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.

User’s Manual

V850E/RS1^{TM}

32-/16-bit Single-Chip Microcontroller with CAN Interface

Hardware

µPD70F3402, µPD70F3403, µPD70F3403A

Document No. U16702EE3V2UD00
Date Published April 2006

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 **INPUT OF SIGNAL DURING POWER OFF STATE**

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

- **The information in this document is current as of April, 2006. 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

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielski Strasse 166 B
30177 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. 52180
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Limburglaan 5
5616 HR Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
TEL: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

Seoul Branch
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

G05.11-1A

Preface

Readers

This User's Document is intended for users who want to understand the functions of the V850E/RS1.

Purpose

This User's Document presents the hardware manual of V850E/RS1.

Organization

This system specification describes the following sections:

- Pin function
- CPU function
- Internal peripheral function
- Flash memory
- Target Electrical Specification
- Package Drawing
- Recommended Soldering Conditions

Legend

Symbols and notation are used as follows:

Weight in data notation	:	Left is high-order column, right is low order column
Active low notation	:	$\overline{\text{xxx}}$ (pin or signal name is over-scored) or /xxx (slash before signal name)
Memory map address:	:	High order at high stage and low order at low stage
Note	:	Explanation of (Note) in the text
Caution	:	Item deserving extra attention
Remark	:	Supplementary explanation to the text
Numeric notation	:	Binary... xxxx or xxxB Decimal... xxxx Hexadecimal... xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity)

K (kilo): 2^{10} = 1024

M (mega): 2^{20} = 1024^2 = 1,048,576

G (giga): 2^{30} = 1024^3 = 1,073,741,824

Table of Contents

Preface 5

Chapter 1 Introduction. 29
1.1 Outline. 29
1.2 Features 30
1.3 Applications 32
1.4 Ordering Information 32
1.5 Pin Configuration (Top View) 33
1.6 Configuration of Function Block 35
1.6.1 Internal block diagram 35
1.6.2 On-chip units. 36

Chapter 2 Pin Functions 39
2.1 List of Pin Functions 39
2.2 Pin States 44
2.3 Description of Pin Functions 45
2.4 Pin I/O Circuit Types, I/O Buffer Power Supply and Handling of Unused Pins 57

Chapter 3 CPU Function 61
3.1 Features 61
3.2 CPU Register Set 62
3.2.1 Program register set 63
3.2.2 System register set 64
3.2.3 Special registers 70
3.3 Operation Modes 74
3.3.1 Operation modes 74
3.4 Address Space 75
3.4.1 CPU address space 75
3.4.2 Image 76
3.4.3 Wrap-around of CPU address space 77
3.4.4 Memory map 78
3.4.5 Memory areas 81
3.4.6 Recommended use of address space 84
3.4.7 Cautions 86
3.5 Programmable I/O Area 92
3.5.1 Programmable peripheral I/O control register (BPC) 92
3.6 Peripheral I/O Registers 93

Chapter 4 Port Functions 105
4.1 Features 105
4.2 Basic Port Configuration. 105
4.3 Port Configuration 107
4.3.1 Table of port configuration 107
4.3.2 Port function swap control register 111
4.3.3 Port 0 112
4.3.4 Port 1 118
4.3.5 Port 3 124
4.3.6 Port 4 133
4.3.7 Port 5 136
4.3.8 Port 7 141
4.3.9 Port 9 143
4.3.10 Port CM. 155
4.3.11 Port CS 159
4.3.12 Port CT 162
4.3.13 Port DL 166

4.4 Port Function Operation 169
4.4.1 Write to I/O ports. 169
4.4.2 Read from I/O port 169
4.4.3 I/O port calculation 169
4.5 Port Type. 170
4.6 Port Block Types 173
4.6.1 Port block type E-SD1 173
4.6.2 Port block type E-SD4 174
4.6.3 Port block type E-SD7E 175
4.6.4 Port block type E-SDW4. 176
4.6.5 Port block type N-SDW7. 177
4.6.6 Port block type N-SD7 178
4.6.7 Port block type L-SD1. 179
4.6.8 Port block type G-SD7 180
4.6.9 Port block type U-SD8 181
4.6.10 Port block type N-SD2 182
4.6.11 Port block type E-SDW10. 183
4.6.12 Port block diagram E-SD1L 184
4.6.13 Port block type W-SDW11 185
4.6.14 Port block type E-SD7 186
4.6.15 Port block type A-1 187
4.6.16 Port block type U-SDW11E 188
4.6.17 Port block type G-SD7A 189
4.6.18 Port block type G-SDW8E 190
4.6.19 Port block type G-SD6 191
4.6.20 Port block type G-SDJ2 192
4.6.21 Port block type G-SDJ5 193
4.6.22 Port block type G-SDJ8E 194
4.6.23 Port block type E-DWJ4 195
4.6.24 Port block type U-SDW11. 196
4.6.25 Port block type W-SD11 197
4.6.26 Port block type W-SD12E. 198
4.6.27 Port block type U-SDW10. 199
4.6.28 Port Block type G-SDW6 200
4.6.29 Port block type E-D1. 201
4.6.30 Port block type E-D4. 202
4.6.31 Port block type D-7E. 203
4.6.32 Port block type C-D1. 204
4.6.33 Port block type B. 205

Chapter 5 Bus Control Function. 207
5.1 Features 207
5.2 Bus Control Pins 208
5.2.1 Pin status when internal ROM, internal RAM, or peripheral I/O is accessed . . 208
5.2.2 Pin status in each operation mode 208
5.3 Memory Block Function 209
5.3.1 Chip select control function. 210
5.4 Bus Access. 211
5.4.1 Number of clocks for access. 211
5.4.2 Bus size setting function 212
5.4.3 Access by bus size 213
5.5 Wait Function 219
5.5.1 Programmable wait function 219
5.5.2 External wait function 220
5.5.3 Relationship between programmable wait and external wait 220
5.5.4 Programmable address wait function 221
5.6 Idle State Insertion Function. 222

5.7 Bus Hold Function . . . 223
5.7.1 Functional outline . . . 223
5.7.2 Bus hold procedure. . . 224
5.7.3 Operation in power save mode. . . 224
5.8 Bus Priority. . . 225
5.9 Boundary Operation Conditions . . . 225
5.9.1 Program space . . . 225
5.9.2 Data space . . . 225
5.10 Bus Timing . . . 226

Chapter 6 Clock Generator . . . 233
6.1 Overview. . . 233
6.2 Configuration . . . 234
6.3 Control Registers. . . 235
6.4 PLL Function . . . 243
6.4.1 Control register . . . 245
6.5 Programmable Clock output Function (PCL) . . . 250
6.5.1 Control registers . . . 250
6.6 Usage . . . 252
6.6.1 To use PLL1 . . . 252
6.6.2 To use clock through mode . . . 253
6.6.3 to Use the programmable clock output function (PCL). . . 253

Chapter 7 16-Bit Timer/Event Counter P . . . 255
7.1 Features . . . 255
7.2 Function Outline . . . 255
7.3 Configuration . . . 256
7.4 Control Registers. . . 261
7.5 Operation . . . 270
7.5.1 Anytime write and reload . . . 270
7.5.2 Interval timer mode (TPnMD2 to TPnMD0 = 000) . . . 275
7.5.3 External event counter mode (TPnMD2 to TPnMD0 = 001). . . 278
7.5.4 External trigger pulse mode (TPnMD2 to TPnMD0 = 010). . . 281
7.5.5 One-shot pulse mode (TPnMD2 to TPnMD0 = 011). . . 284
7.5.6 PWM mode (TPnMD2 to TPnMD0 = 110) . . . 287
7.5.7 Free-running mode (TPnMD2 to TPnMD0 = 101) . . . 291
7.5.8 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110) . . . 297
7.6 Timer Synchronization Operation Function . . . 299

Chapter 8 16-Bit Timer/Event Counter Q . . . 303
8.1 Features . . . 303
8.2 Function Outline . . . 303
8.3 Configuration . . . 304
8.4 Control Registers. . . 311
8.5 Operation . . . 319
8.5.1 Anytime write and reload . . . 319
8.5.2 Interval timer mode (TQnMD2 to TQnMD0 = 000) . . . 324
8.5.3 External event counter mode (TQnMD2 to TQnMD0 = 001) . . . 327
8.5.4 External trigger pulse mode (TQnMD2 to TQnMD0 = 010) . . . 331
8.5.5 One-shot pulse mode (TQnMD2 to TQnMD0 = 011) . . . 334
8.5.6 PWM mode (TQnMD2 to TQnMD0 = 110). . . 337
8.5.7 Free-running mode (TQnMD2 to TQnMD0 = 101) . . . 342
8.5.8 Pulse width measurement mode (TQnMD2 to TQnMD0 = 110). . . 349
8.6 Timer Synchronization Operation Function . . . 351

Chapter 9 16-Bit Interval Timer M 355
9.1 Features 355
9.2 Configuration 356
9.3 Control Register 357
9.4 Operation 358
9.4.1 Interval timer mode 358
9.4.2 Clock generator and clock enable timing 358

Chapter 10 Functions of Watchdog Timer 2 359
10.1 Functions 359
10.2 Configuration 360
10.3 Control Registers 360
10.4 Operation 364

Chapter 11 A/D Converter 367
11.1 Functions 367
11.2 Configuration 369
11.3 Control Registers 371
11.4 Operation 380
11.4.1 Basic operation 380
11.4.2 Trigger mode 381
11.4.3 Operation mode 383
11.5 Extended Functions 391
11.5.1 Automatic discharge operation 391
11.5.2 Diagnostic mode 391
11.6 Precautions on Operation 392
11.6.1 Stopping A/D 392
11.6.2 Trigger input during A/D conversion operation 392
11.6.3 External or timer trigger interval 392
11.6.4 Operation in standby modes 392
11.6.5 Compare match interrupt in timer trigger mode (External trigger mode) 393
11.6.6 Timing that makes the A/D conversion result undefined 393
11.7 Cautions 395
11.8 How to Read A/D Converter Characteristics Table 398

Chapter 12 Asynchronous Serial Interface A (UARTA) 403
12.1 Features 403
12.2 Configuration 404
12.2.1 Control registers 406
12.3 Control Registers 407
12.4 Interrupt Request Signals 415
12.5 Operation 416
12.5.1 Data format 416
12.5.2 UART transmission 417
12.5.3 Procedure of continuous transmission 418
12.5.4 UART reception 420
12.5.5 Reception errors 421
12.5.6 Types and operation of parity 423
12.5.7 Noise filter of receive data 424
12.6 Dedicated Baud Rate Generator 425
12.7 Caution for Use in On-chip Debug Mode 430

Chapter 13 3-Wire Serial Interface (CSIB) 431
13.1 Features 431
13.2 Configuration 431
13.3 Control Registers Overview 433
13.4 Control Registers Description 434
13.5 Transfer Data Length Change Function 441

13.6 Interrupt Request Signals . . . 442
13.7 Operation . . . 443
13.7.1 Single transfer mode (master mode, transmission/reception mode) . . . 443
13.7.2 Single transfer mode (master mode, reception mode) . . . 444
13.7.3 Continuous mode (master mode, transmission/reception mode) . . . 445
13.7.4 Continuous mode (master mode, reception mode) . . . 446
13.7.5 Continuous reception mode (error) . . . 447
13.7.6 Continuous mode (slave mode, transmission/reception mode) . . . 448
13.7.7 Continuous mode (slave mode, reception mode) . . . 449
13.7.8 Clock timing . . . 450
13.7.9 Output pin status with operation disabled . . . 452
13.8 Operation Flow . . . 453
13.9 Prescaler 3 . . . 461
13.9.1 Control registers of prescaler 3 . . . 461
13.9.2 Generation of count clock . . . 462

Chapter 14 Queued CSI (CSI30, CSI31) . . . 463
14.1 Features . . . 463
14.1.1 Queued CSI Block Diagram . . . 464
14.1.2 Input/Output Pins . . . 464
14.2 Queued CSI Control Registers . . . 465
14.3 Explanation of Queued CSI Functions . . . 476
14.3.1 Transmit Buffer . . . 476
14.3.2 Serial Data Direction Select Function . . . 477
14.3.3 Data Length Select Function . . . 478
14.3.4 Slave Mode . . . 479
14.3.5 Master Mode . . . 479
14.3.6 Transmission Clock Select Function . . . 480
14.3.7 Description of the Single Buffer Transfer Mode . . . 481
14.3.8 Description of the FIFO Buffer Transfer Mode . . . 483
14.3.9 Description of the Operation Modes . . . 485
14.3.10 Additional Timing and Delay Selections . . . 486
14.3.11 Default Pin Levels . . . 489
14.3.12 Transmit Buffer Overflow Interrupt Signal (INTC3nO) . . . 490
14.4 Operating Procedure . . . 491

Chapter 15 DMA Functions (DMA Controller) . . . 503
15.1 Features . . . 503
15.2 Configuration . . . 504
15.3 Control Registers . . . 505
15.4 DMA Transfer Rules . . . 520
15.4.1 Transfer targets . . . 520
15.4.2 DMA channel priority . . . 520
15.4.3 DMA transfer start factors . . . 520
15.4.4 DMA transfer end . . . 521
15.5 Transfer Modes . . . 522
15.5.1 Single transfer mode . . . 522
15.5.2 Fixed channel transfer mode . . . 523
15.5.3 Block transfer mode . . . 524
15.5.4 Summary on the transfer modes . . . 524

Chapter 16 FCAN Controller . . . 525

16.1 Overview . . . 525

16.1.1 Features . . . 525

16.1.2 Overview of Functions . . . 526

16.1.3 Configuration . . . 527

16.2 CAN Protocol . . . 528

16.2.1 Frame format . . . 528

16.2.2 Frame types . . . 529

16.2.3 Data frame and remote frame. . . . 529

16.2.4 Error frame . . . 537

16.2.5 Overload frame . . . 538

16.3 Functions . . . 539

16.3.1 Determining bus priority . . . 539

16.3.2 Bit stuffing . . . 539

16.3.3 Multi masters . . . 540

16.3.4 Multi cast. . . . 540

16.3.5 CAN sleep mode/CAN stop mode function. . . . 540

16.3.6 Error control function . . . 540

16.3.7 Baud rate control function . . . 547

16.4 Connection with Target System . . . 551

16.5 Internal Registers of CAN controller . . . 552

16.5.1 CAN controller configuration. . . . 552

16.5.2 Register access type . . . 553

16.5.3 Register bit configuration . . . 586

16.6 Control Registers . . . 588

16.7 Bit Set/Clear Function . . . 624

16.8 CAN Controller Initialization . . . 626

16.8.1 Initialization of CAN module . . . 626

16.8.2 Initialization of message buffer . . . 626

16.8.3 Redefinition of message buffer . . . 626

16.8.4 Transition from initialization mode to operation mode . . . 627

16.8.5 Resetting error counter CnERC of CAN module . . . 628

16.9 Message Reception . . . 629

16.9.1 Message reception . . . 629

16.9.2 Receive history list function . . . 630

16.9.3 Mask function . . . 632

16.9.4 Multi buffer receive block function . . . 634

16.9.5 Remote frame reception . . . 635

16.10 Message Transmission . . . 636

16.10.1 Message transmission . . . 636

16.10.2 Transmit history list function . . . 638

16.10.3 Automatic block transmission (ABT) . . . 639

16.10.4 Transmission abort process . . . 641

16.10.5 Remote frame transmission . . . 641

16.11 Power Saving Modes . . . 642

16.11.1 CAN sleep mode. . . . 642

16.11.2 CAN stop mode . . . 644

16.11.3 Example of using power saving modes . . . 646

16.12 Interrupt Function . . . 647

16.13 Diagnosis Functions and Special Operational Modes . . . 648

16.13.1 Receive-only mode . . . 648

16.13.2 Single-shot mode . . . 649

16.13.3 Self-test mode. . . . 650

16.14 Time Stamp Function. . . . 651

16.14.1 Time stamp function . . . 651

16.15 Baud Rate Settings . . . 652

16.15.1 Bit rate setting conditions . . . 652

16.15.2 Representative examples of baud rate settings . . . 656

16.16 Operation of CAN Controller . . . 660

Chapter 17 Interrupt/Exception Processing Function . . . 685
17.1 Features . . . 685
17.2 Non-Maskable Interrupts . . . 688
17.2.1 Operation . . . 690
17.2.2 Restore . . . 691
17.2.3 NP flag . . . 692
17.2.4 Eliminating noise on NMI pin . . . 692
17.2.5 Function to detect edge of NMI pin . . . 693
17.3 Maskable Interrupts . . . 694
17.3.1 Operation . . . 694
17.3.2 Restore . . . 696
17.3.3 Priorities of maskable interrupts . . . 697
17.3.4 Interrupt control register (xxICn) . . . 701
17.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3) . . . 704
17.3.6 In-service priority register (ISPR) . . . 705
17.3.7 Maskable interrupt status flag . . . 706
17.3.8 Watchdog timer mode register 2 (WDTM2) . . . 707
17.3.9 Eliminating noise on INTP0 to INTP7 pins . . . 708
17.3.10 Function to detect edge of INTP0 to INTP7 pins . . . 708
17.4 Software Exception . . . 714
17.4.1 Operation . . . 714
17.4.2 Restore . . . 715
17.4.3 Exception status flag (EP) . . . 716
17.5 Exception Trap . . . 717
17.5.1 Illegal opcode definition . . . 717
17.5.2 Debug trap . . . 719
17.6 Interrupt Acknowledge Time of CPU . . . 721
17.7 Periods in Which Interrupts Are Not Acknowledged by CPU . . . 722

Chapter 18 Standby Function . . . 723
18.1 Overview . . . 723
18.2 Status Transition . . . 724
18.3 HALT Mode . . . 727
18.3.1 Setting and operation status . . . 727
18.3.2 Releasing HALT mode . . . 727
18.4 IDLE1 Mode . . . 729
18.4.1 Setting and operation status . . . 729
18.4.2 Releasing IDLE1 mode . . . 729
18.5 IDLE2 Mode . . . 731
18.5.1 Setting and operation status . . . 731
18.5.2 Releasing IDLE2 mode . . . 731
18.5.3 Securing setup time when releasing IDLE2 mode . . . 733
18.6 STOP Mode . . . 734
18.6.1 Setting and operation status . . . 734
18.6.2 Releasing STOP mode . . . 734
18.7 Securing Oscillation Stabilization Time . . . 736
18.8 Control Registers . . . 737

Chapter 19 RESET Function . . . 739
19.1 Overview . . . 739
19.2 Registers to Check Reset Source . . . 740
19.3 Operation . . . 741
19.3.1 Reset operation by $\overline{\text{RESET}}$ pin . . . 741
19.3.2 Reset operation by WDT2RES signal . . . 743
19.3.3 Reset operation by Low Voltage Detector and Power On Clear . . . 745

Chapter 20 Regulator. 747
20.1 Outline. 747
20.2 Operation 748

Chapter 21 Flash Memory 749
21.1 Features 749
21.2 Erasure Unit 750
21.2.1 Address assignment in the memory map 751
21.3 Writing with Flash Programmer 752
21.4 Programming Environment. 752
21.5 Communication Mode 753
21.6 Pin Connection. 756
21.6.1 FLMD0 pin 756
21.6.2 FLMD1 pin 757
21.6.3 Serial interface pin 758
21.6.4 $\overline{\text{RESET}}$ pin 760
21.6.5 Port pins (including NMI) 760
21.6.6 Other signal pins. 760
21.6.7 Power supply 760
21.7 Recommended Circuit Example of the Flash Write Mode 761
21.8 Programming Method 762
21.8.1 Flash memory control 762
21.8.2 Flash memory programming mode. 763
21.8.3 Selection of communication mode 764
21.8.4 Communication command 764

Chapter 22 On-Chip Debug Function. 767
22.1 Functional Outline 767
22.1.1 Type of on-chip debug unit 767
22.1.2 Debug functions 767
22.2 Security Function. 770
22.3 Control Register. 771
22.4 Operation of On-Chip Debug Function 773
22.5 Connection to N-Wire Emulator 774
22.5.1 KEL connector 774
22.6 Restrictions and Cautions on On-Chip Debug Function .778

Chapter 23 Power-On-Clear Circuit 779
23.1 Functions of Power-on-Clear Circuit 779
23.2 Configuration of Power-on-Clear Circuit. 780
23.3 Operation of Power-on-Clear Circuit 780

Chapter 24 Low-Voltage Detector 781
24.1 Functions of Low-Voltage Detector 781
24.2 Configuration of Low-Voltage Detector 781
24.3 Registers Controlling Low-Voltage Detector. 782
24.4 Operation of Low-Voltage Detector 785
24.4.1 To use for internal reset signal 785
24.4.2 To use for interrupt 787
24.5 RAM Retention Voltage Detection Operation 788

Chapter 25 Clock Monitor 789
25.1 Functions of Clock Monitor. 789
25.2 Configuration of Clock Monitor 789
25.3 Register Controlling Clock Monitor 790
25.4 Operation of Clock Monitor. 791

Chapter 26 CRC Function 793
26.1 Functions 793
26.2 Configuration 793
26.3 Registers. 794
26.4 Operation 795
26.5 Usage Method 796

Chapter 27 Electrical Specification 799
27.1 General 799
27.2 Absolute Maximum Ratings 799
27.2.1 Absolute maximum ratings (μPD70F3402). 800
27.2.2 Absolute maximum ratings (μPD70F3403, μPD70F3403A) 800
27.3 Capacitance 800
27.4 Recommended Operating Conditions 801
27.4.1 Recommended operating conditions (μPD70F3402) 801
27.4.2 Recommended operating conditions (μPD70F3403, μPD70F3403A) 801
27.5 Oscillator Characteristics 802
27.5.1 General condition 802
27.5.2 Oscillator timing and recommended oscillator connection 802
27.6 Voltage Regulator Characteristics 803
27.6.1 General condition 803
27.6.2 Regulator 0 803
27.6.3 Regulator 1 803
27.7 DC Characteristics. 804
27.7.1 General condition 804
27.7.2 DC input/output level 804
27.7.3 DC power supply current (μPD70F3402) 806
27.7.4 DC power supply current (μPD70F3403, μPD70F3403A) 806
27.8 AC Characteristics. 807
27.8.1 General condition 807
27.8.2 AC test input waveform. 807
27.8.3 Input waveform 808
27.8.4 Output waveform 808
27.8.5 $\overline{\text{RESET}}$, NMI, Interrupt and FLMD0 timing 809
27.8.6 Timer P and Timer Q input/output 810
27.8.7 Bus interface timing (μPD70F3403, μPD70F3403A only) 811
27.8.8 CSIBn timing. 815
27.8.9 UARTAn timing. 816
27.8.10 CAN Interface timing (μPD70F3402) 817
27.8.11 CAN Interface timing (μPD70F3403, μPD70F3403A) 817
27.8.12 CAN Interface timing diagram. 817
27.8.13 CSI3n timing 818
27.8.14 A/D Converter characteristics. 824
27.8.15 Power-on-clear (POC) 825
27.8.16 Low-Voltage Indicator (LVI) 826
27.9 Power on Sequence. 827
27.10 Flash Memory Characteristics 828
27.10.1 General condition 828
27.10.2 Basic Flash characteristics (μPD70F3402). 828
27.10.3 Basic Flash characteristics (μPD70F3403, μPD70F3403A). 828
27.10.4 External Flash programmer serial write operation characteristics 829
27.10.5 Flash programming characteristics. 830

Chapter 28 Package Drawing 831

Chapter 29 Recommended Soldering Conditions 833

Appendix A Instruction Set List 835

Appendix B Index 843

Appendix C Revision History 849

List of Tables

Figure 1-1: Pin Configuration 33
Figure 1-2: Internal Block Diagram 35
Figure 2-1: Pin I/O Circuits 60
Figure 3-1: CPU Register Set 62
Figure 3-2: Program Counter (PC) Format 63
Figure 3-3: Interrupt Status Saving Registers (EIPC and EIPSW) Format 65
Figure 3-4: NMI Status Saving Registers (FEPC and FEPSW) Format 66
Figure 3-5: Interrupt Source Register (ECR) Format 66
Figure 3-6: Program Status Word (PSW) Format (1/2) 67
Figure 3-7: CALLT Execution Status Saving Registers (CTPC and CTPSW) Format 68
Figure 3-8: Exception/Debug Trap Status Saving Registers (DBPC and DBPSW) Format 69
Figure 3-9: CALLT Base Pointer (CTBP) Format 69
Figure 3-10: Command Register (PRCMD) Format 72
Figure 3-11: System Status Register (SYS) Format 73
Figure 3-12: CPU Address Space 75
Figure 3-13: Image on Address Space 76
Figure 3-14: Program Space 77
Figure 3-15: Data Space 77
Figure 3-16: Data Memory Map (Physical Addresses) 78
Figure 3-17: Program Memory Map 79
Figure 3-18: Memory Map Area for µPD70F3402, µPD70F3403 and µPD70F3403A 80
Figure 3-19: Internal ROM Area 81
Figure 3-20: Internal RAM Area (10 KB) 82
Figure 3-21: Internal RAM Area (16 KB) 82
Figure 3-22: Internal Peripheral I/O Area 83
Figure 3-23: Wrap-Around Using µPD70F3402 / µPD70F3403 / µPD70F3403A 84
Figure 3-24: Recommended Memory Map 85
Figure 3-25: System Wait Control Register (VSWC) Format 86
Figure 3-26: On-Chip Debug Mode Register (OCDM) Format 87
Figure 3-27: Timing Chart When On-Chip Debug Function Is Not Used 88
Figure 3-28: Timing Chart of Transition to Normal Operation Mode 88
Figure 3-29: Timing Chart of Transition to On-Chip Debug Mode 89
Figure 3-30: Programmable Peripheral I/O Control Register (BPC) Format 92
Figure 4-1: Port Configuration Diagram 105
Figure 4-2: Port Function Swap Control Register Format 111
Figure 4-3: Port Register 0 (P0) Format 112
Figure 4-4: Port Mode Register 0 (PM0) Format 113
Figure 4-5: Port Mode Control Register 0 (PMC0) Format (1/2) 113
Figure 4-6: Port Function Control Register 0 (PFC0) Format 114
Figure 4-7: Pull-up Resistor Option Register 0 (PU0) Format 115
Figure 4-8: Pull-down Resistor Option Register 0 (PD0) Format 115
Figure 4-9: External Interrupt Falling Edge Specification Register 0 (INTF0) Format 116
Figure 4-10: External Interrupt Rising Edge Specification Register 0 (INTR0) Format 117
Figure 4-11: Port Register 1 (P1) Format 118
Figure 4-12: Port Mode Register 1 (PM1) Format 119
Figure 4-13: Port Mode Control Register 1 (PMC1) Format 119
Figure 4-14: Port Function Control Register 1 (PFC1) Format 120
Figure 4-15: Port Function Control Extended Register 1 (PFCE1) Format 120
Figure 4-16: Pull-up Resistor Option Register 1 (PU1) Format 121
Figure 4-17: Pull-down Resistor Option Register 1 (PD1) Format 121
Figure 4-18: External Interrupt Falling Edge Specification Register 1 (INTF1) Format 122
Figure 4-19: External Interrupt Rising Edge Specification Register 1 (INTR1) Format 123
Figure 4-20: Port Register 3 (P3) Format 125
Figure 4-21: Port Mode Register 3 (PM3) Format 126
Figure 4-22: Port Mode Control Register 3 (PMC3) Format (1/2) 127

Figure 4-23: Port Function Control Register 3 (PFC3) Format 128
Figure 4-24: Port Function Control Expansion Register 3 (PFCE3) Format 129
Figure 4-25: Pull-up Resistor Option Register 3 (PU3) Format 130
Figure 4-26: Pull-down Resistor Option Register 3 (PD3) Format 130
Figure 4-27: External Interrupt Falling Edge Specification Register 3 (INTF3) Format 131
Figure 4-28: External Interrupt Rising Edge Specification Register 3 (INTR3) Format 132
Figure 4-29: Port Register 4 (P4) Format 133
Figure 4-30: Port Mode Register 4 (PM4) Format 134
Figure 4-31: Port Mode Control Register 4 (PMC4) Format 134
Figure 4-32: Pull-up Resistor Option Register 4 (PU4) Format 135
Figure 4-33: Pull-down Resistor Option Register 4 (PD4) Format 135
Figure 4-34: Port Register 5 (P5) Format 136
Figure 4-35: Port Mode Register 5 (PM5) Format 137
Figure 4-36: Port Mode Control Register 5 (PMC5) Format (1/2) 137
Figure 4-37: Port Function Control Register 5 (PFC5) Format 138
Figure 4-38: Pull-up Resistor Option Register 5 (PU5) Format 139
Figure 4-39: Pull-down Resistor Option Register 5 (PD5) Format 140
Figure 4-40: Port Register 7H, Port Register 7L (P7H, P7L)Format 141
Figure 4-41: Port Mode Register 7H, Port Mode Register 7L (PM7H, PM7L)Format 142
Figure 4-42: Port Register 9 (P9) Format 144
Figure 4-43: Port Mode Register 9 (PM9) Format 145
Figure 4-44: Port Mode Control Register 9 (PMC9) Format (1/3) 145
Figure 4-45: Port Function Control Register 9 (PFC9) Format 148
Figure 4-46: Port Function Control Expansion Register 9 (PFCE9) Format 148
Figure 4-47: Pull-up Resistor Option Register 9 (PU9) Format 151
Figure 4-48: Pull-down Resistor Option Register 9 (PD9) Format 152
Figure 4-49: External Interrupt Falling Edge Specification Register 9H (INTF9H) Format 153
Figure 4-50: External Interrupt Rising Edge Specification Register 9H (INTR9H) Format 154
Figure 4-51: Port Register CM (PCM) Format 155
Figure 4-52: Port Mode Register CM (PMCM) Format 156
Figure 4-53: Port Mode Control Register CM (PMCCM) Format 157
Figure 4-54: Pull-up Resistor Option Register CM (PUCM) Format 158
Figure 4-55: Pull-down Resistor Option Register CM (PDCM) Format 158
Figure 4-56: Port Register CS (PCS) Format 159
Figure 4-57: Port Mode Register CS (PMCS) Format 160
Figure 4-58: Port Mode Control Register CS (PMCCS) Format 160
Figure 4-59: Pull-up Resistor Option Register CS (PUCS) Format 161
Figure 4-60: Pull-down Resistor Option Register CS (PDCS) Format 161
Figure 4-61: Port Register CT (PCT) Format 162
Figure 4-62: Port Mode Register CT (PMCT) Format 163
Figure 4-63: Port Mode Control Register CT (PMCCT) Format 164
Figure 4-64: Pull-up Resistor Option Register CT (PUCT) Format 165
Figure 4-65: Pull-down Resistor Option Register CT (PDCT) Format 165
Figure 4-66: Port Register DL (PDL) Format 166
Figure 4-67: Port Mode Register DL (PMDL) Format 167
Figure 4-68: Port Mode Control Register DL (PMCDL) Format 168
Figure 4-69: Type E-SD1 Block Diagram..... 173
Figure 4-70: Type E-SD4 Block Diagram..... 174
Figure 4-71: Type E-SD7E Block Diagram 175
Figure 4-72: Type E-SDW4 Block Diagram 176
Figure 4-73: Type N-SDW7 Block Diagram 177
Figure 4-74: Type N-SD7 Block Diagram 178
Figure 4-75: Type L-SD1 Block Diagram..... 179
Figure 4-76: Type G-SD7 Block Diagram 180
Figure 4-77: Type U-SD8 Block Diagram 181
Figure 4-78: Type N-SD2 Block Diagram 182
Figure 4-79: Type E-SDW10 Block diagram..... 183
Figure 4-80: Type E-SD1L Block Diagram..... 184

Figure 4-81: Type W-SDW11 Block Diagram ... 185
Figure 4-82: Type E-SD7 Block Diagram ... 186
Figure 4-83: Type A-1 Block Diagram ... 187
Figure 4-84: Type U-SDW11E Block Diagram ... 188
Figure 4-85: Type G-SD7A Block Diagram ... 189
Figure 4-86: Type G-SDW8E Block Diagram ... 190
Figure 4-87: Type G-SD6 Block Diagram ... 191
Figure 4-88: Type G-SDJ2 Block Diagram ... 192
Figure 4-89: Type G-SDJ5 Block Diagram ... 193
Figure 4-90: Type G-SDJ8E Block Diagram ... 194
Figure 4-91: Type E-DWJ4 Block Diagram ... 195
Figure 4-92: Type U-SDW11 Block Diagram ... 196
Figure 4-93: Type W-SD11 Block Diagram ... 197
Figure 4-94: Type W-SD12E Block Diagram ... 198
Figure 4-95: Type U-SDW10 Block Diagram ... 199
Figure 4-96: Type G-SDW6 Block Diagram ... 200
Figure 4-97: Type E-D1 Block Diagram ... 201
Figure 4-98: Type E-D4 Block Diagram ... 202
Figure 4-99: Type D-7E Block Diagram ... 203
Figure 4-100: Type C-D1 Block Diagram ... 204
Figure 4-101: Type B Block Diagram ... 205
Figure 5-1: Data Memory Map ... 209
Figure 5-2: Bus Size Configuration Register (BSC) Format ... 212
Figure 5-3: Little-Endian Address in Word ... 213
Figure 5-4: Data Wait Control Register 0 (DWC0) Format ... 219
Figure 5-5: Example of Inserting Wait States in Separate Bus Mode ... 220
Figure 5-6: Address Wait Control Register (AWC) Format ... 221
Figure 5-7: Bus Cycle Control Register (BCC) Format ... 222
Figure 5-8: Bus Hold Status Transition Procedure ... 224
Figure 5-9: Bus Read Timing (Bus Size: 16 bit, 16-bit Access) ... 226
Figure 5-10: Bus Read Timing (Bus Size: 8 bit) ... 227
Figure 5-11: Bus Write Timing (Bus Size: 16 bit, 16-bit Access) ... 228
Figure 5-12: Bus Write Timing (Bus Size: 8 bit) ... 229
Figure 5-13: Address Wait Timing (Bus Size: 16 bit) ... 230
Figure 5-14: Bus Hold Timing (Bus Size: 16 bit) ... 231
Figure 6-1: Clock Generator ... 234
Figure 6-2: Main Peripheral Clock Control Register (MPCCTL) Format (1/2) ... 235
Figure 6-3: Extension Clock Select Register (EXCKSEL) Format ... 237
Figure 6-4: Power Save Mode Register (PSMR) Format ... 238
Figure 6-5: Processor Clock Control Register (PCC) Format ... 239
Figure 6-6: CPU Operation Clock Status Register (CCLS) Format ... 240
Figure 6-7: Ring-OSC Mode Register (RCM) Format ... 240
Figure 6-8: Oscillation Stabilization Time Select Register (OSTS) Format ... 241
Figure 6-9: Clock Selection Register 2 (OCKS2) Format ... 242
Figure 6-10: PLL Control Register (PLLCTL0) Format ... 245
Figure 6-11: PLL Control Register (PLLCTL1) Format ... 245
Figure 6-12: PLL Lockup Time Specification Register (PLLS) ... 246
Figure 6-13: Clock Control Register (CKC) Format ... 247
Figure 6-14: Clock Selection Register 0 (OCKS0) Format ... 248
Figure 6-15: Clock Selection Register 1 (OCKS1) Format ... 249
Figure 6-16: Programmable Clock Mode Register (PCLM) Format ... 250
Figure 6-17: Clock Selection Register 3 (OCKS3) Format ... 251
Figure 7-1: Block Diagram of Timer P ... 257
Figure 7-2: Capture/Compare Register 0 (TPnCCR0) Format ... 258
Figure 7-3: Capture/Compare Register 1 (TPnCCR1) Format ... 259
Figure 7-4: TMPn Timer Read Buffer Register (TPnCNT) Format ... 260
Figure 7-5: TMPn Control Register 0 (TPnCTL0) Format ... 261
Figure 7-6: TMPn Control Register 1 (TPnCTL1) Format (1/2) ... 262

Figure 7-7: TMPn Dedicated I/O Control Register 0 (TPnIOC0) Format 264
Figure 7-8: TMPn Dedicated I/O Control Register 1 (TPnIOC1) Format 265
Figure 7-9: TMPn Dedicated I/O Control Register 2 (TPnIOC2) Format 266
Figure 7-10: TMPn Option Register 0 (TPnOPT0) Format 267
Figure 7-11: Selector Operation Control Register 0 (SELCNT0) Format 268
Figure 7-12: TIPnm Pin Noise Elimination Control Register n (PnmNFC) Format 269
Figure 7-13: Flowchart of Basic Operation for Anytime Write.. 271
Figure 7-14: Timing Diagram for Anytime Write... 272
Figure 7-15: Flowchart of Basic Operation for Reload... 273
Figure 7-16: Timing Chart for Reload 274
Figure 7-17: Flowchart of Basic Operation in Interval Timer Mode.. 275
Figure 7-18: Basic Operation Timing in Interval Timer Mode (1/2).. 276
Figure 7-19: Flowchart of Basic Operation in External Event Counter Mode 278
Figure 7-20: Basic Operation Timing in External Event Counter Mode (1/2)................................. 279
Figure 7-21: Flowchart of Basic Operation in External Trigger Pulse Output Mode 282
Figure 7-22: Basic Operation Timing in External Trigger Pulse Output Mode
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 283
Figure 7-23: Flowchart of Basic Operation in One-Shot Pulse Mode 285
Figure 7-24: Timing of Basic Operation in One-Shot Pulse Mode
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 286
Figure 7-25: Flowchart of Basic Operation in PWM Mode (1/2) 287
Figure 7-26: Basic Operation Timing in PWM Mode (1/2) 289
Figure 7-27: Flowchart of Basic Operation in Free-Running Mode.. 292
Figure 7-28: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 0)
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 293
Figure 7-29: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 1)
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 294
Figure 7-30: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 0)
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 295
Figure 7-31: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 1)
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0) 296
Figure 7-32: Flowchart of Basic Operation in Pulse Width Measurement Mode 297
Figure 7-33: Basic Operation Timing in Pulse Width Measurement Mode
(TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 0) 298
Figure 7-34: Tuned Operation Image (TMP2, TMP3, TMQ0).. 301
Figure 7-35: Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)................. 302
Figure 8-1: Block Diagram of Timer Q 305
Figure 8-2: Capture/Compare Register 0 (TQnCCR0) Format 306
Figure 8-3: Capture/Compare Register 1 (TQnCCR1) Format 307
Figure 8-4: Capture/Compare Register 2 (TQnCCR2) Format 308
Figure 8-5: Capture/Compare Register 3 (TQnCCR3) Format 309
Figure 8-6: Timer Read Buffer Register (TQnCNT) Format 310
Figure 8-7: Timer Qn Control Register 0 (TQnCTL0) Format 311
Figure 8-8: Timer Q Control Register 1 (TQnCTL1) Format (1/2) 312
Figure 8-9: Timer Q Dedicated I/O Control Register 0 (TQnIOC0) Format 314
Figure 8-10: Timer Q Dedicated I/O Control Register 1 (TQnIOC1) Format 315
Figure 8-11: Timer Q Dedicated I/O Control Register 2 (TQnIOC2) Format 316
Figure 8-12: Timer Q Option Register 0 (TQnOPT0) Format 317
Figure 8-13: TIQnm Pin Noise Elimination Control Register n (QnmNFC) Format 318
Figure 8-14: Flowchart of Basic Operation for Anytime Write.. 320
Figure 8-15: Timing Chart of Any Time Write 321
Figure 8-16: Flowchart of Basic Operation for Reload... 322
Figure 8-17: Timing Chart of Reload.. 323
Figure 8-18: Flowchart of Basic Operation in Interval Timer Mode.. 324
Figure 8-19: Basic Operation Timing in Interval Timer Mode (1/2).. 325
Figure 8-20: Flowchart of Basic Operation in External Event Counter Mode 328
Figure 8-21: Basic Operation Timing in External Event Counter Mode (1/2)................................. 329
Figure 8-22: Flowchart of Basic Operation in External Trigger Pulse Output Mode 332

Figure 8-23: Basic Operation Timing in External Trigger Pulse Output Mode ... 333
Figure 8-24: Flowchart of Basic Operation in One-Shot Pulse Mode ... 335
Figure 8-25: Timing of Basic Operation in One-Shot Pulse Mode ... 336
Figure 8-26: Flowchart of Basic Operation in PWM Mode (1/2) ... 338
Figure 8-27: Basic Operation Timing in PWM Mode (1/2) ... 340
Figure 8-28: Flowchart of Basic Operation in Free-Running Mode ... 343
Figure 8-29: Basic Operation Timing in Free-Running Mode (1/4) ... 345
Figure 8-30: Flowchart of Basic Operation in Pulse Width Measurement Mode ... 349
Figure 8-31: Basic Operation Timing in Pulse Width Measurement Mode ... 350
Figure 8-32: Tuned Operation Image (TMP2, TMP3, TMQ0) ... 353
Figure 8-33: Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0) ... 354
Figure 9-1: Block Diagram of TMM0 ... 356
Figure 9-2: TMM0 Compare Register 0 (TMnCMP0) Format ... 356
Figure 9-3: TMM0 Timer Control Register (TM0CTL0) Format ... 357
Figure 9-4: Interval Timer Mode Timing ... 358
Figure 9-5: Count Operation Start Timing ... 358
Figure 10-1: Block Diagram of Watchdog Timer 2 ... 359
Figure 10-2: Oscillation Stabilization Time Select Register (OSTS) Format ... 360
Figure 10-3: Watchdog Timer Mode Register 2 (WDTM2) Format ... 361
Figure 10-4: Watchdog Timer Enable Register (WDTE) Format ... 363
Figure 10-5: Oscillation Stabilization Time Selection Function ... 364
Figure 11-1: Block Diagram of A/D Converter ... 368
Figure 11-2: A/D Conversion Result Register n
(ADA0CRn, ADA0CRDD, ADA0CRSS) Format (1/2) ... 371
Figure 11-3: Relationship Between Analog Input Voltages and A/D Converter Results ... 373
Figure 11-4: A/D Converter Mode Register 0H (ADSCM0H) Format ... 374
Figure 11-5: A/D Converter Mode Register 0L (ADSCM0L) Format (1/2) ... 375
Figure 11-6: AD Converter Mode Register 1H (ADSCM1H) Format ... 377
Figure 11-7: AD Converter Extended Mode Register (ADVMS0) ... 378
Figure 11-8: AD Converter External Trigger Selection Register (SELCNT1) Format ... 379
Figure 11-9: A/D Converter Basic Operation ... 380
Figure 11-10: Example of Select Mode Operation (ANI2) (1/2) ... 383
Figure 11-11: Example of Single Scan Mode Operation (4-Channel Scan (ANI2 to ANI5)) (1/2) ... 385
Figure 11-12: Example of Continuous Select Mode Operation (ANI2) (1/2) ... 387
Figure 11-13: Example of Continuous Scan Mode Operation (ANI2 to ANI5) (1/2) ... 389
Figure 11-14: Conversion Result Read Timing (when Conversion Result is undefined) ... 393
Figure 11-15: Conversion Result Read Timing (When Conversion Result Is Normal) ... 394
Figure 11-16: Processing of Analog Input Pin ... 395
Figure 11-17: Generation Timing of A/D Conversion End Interrupt Request ... 396
Figure 11-18: AV_{REF0} Pin Processing Example ... 397
Figure 11-19: Overall Error ... 398
Figure 11-20: Quantization Error ... 399
Figure 11-21: Zero-Scale Error ... 399
Figure 11-22: Full-Scale Error ... 400
Figure 11-23: Differential Linearity Error ... 400
Figure 11-24: Integral Linearity Error ... 401
Figure 11-25: Sampling Time ... 401
Figure 12-1: Block Diagram of Asynchronous Serial Interface A ... 405
Figure 12-2: UARTAn Control Register 0 (UAnCTL0) Format (1/2) ... 407
Figure 12-3: UARTAn Control Register 1 (UAnCTL1) Format ... 409
Figure 12-4: UARTAn Control Register 2 (UAnCTL2) Format ... 410
Figure 12-5: UARTAn Option Control Register 0 (UAnOPT0) ... 411
Figure 12-6: UARTAn Status Register (UAnSTR) Format (1/2) ... 412
Figure 12-7: UARTAn Receive Data Register (UAnRX) Format ... 414
Figure 12-8: UARTAn Transmit Data Register (UAnTX) Format ... 414
Figure 12-9: Format of Transmit/Receive Data of UARTA (1/2) ... 416
Figure 12-10: UART Transmission ... 417
Figure 12-11: Processing Flow of Continuous Transfer ... 418

Figure 12-12: Timing of Continuous Transmission Operation 419
Figure 12-13: UART Reception........ 420
Figure 12-14: Receive Data Read Flow........ 421
Figure 12-15: Noise Filter Circuit 424
Figure 12-16: Configuration of Baud Rate Generator........ 425
Figure 12-17: Permissible Baud Rate Range for Reception........ 428
Figure 12-18: Transfer Rate for Continuous Transmission........ 430
Figure 13-1: Block Diagram of 3-Wire Serial Interface 432
Figure 13-2: CSIBn Control Register 0 (CBnCTL0) Format (1/2) 434
Figure 13-3: CSIBn Control Register 1 (CBnCTL1) Format (1/2) 436
Figure 13-4: CSIBn Control Register 2 (CBnCTL2) Format 438
Figure 13-5: CSIBn Status Register (CBnSTR) Format 439
Figure 13-6: CSIBn Receive Data Register (CBnRX) Format 440
Figure 13-7: CSIBn Transmit Data Register (CBnTX) Format 440
Figure 13-8: Changing Transfer Data Length 441
Figure 13-9: Single Transfer Timing (Master Mode, Transmission/Reception Mode) 443
Figure 13-10: Single Transfer Timing (Master Mode, Reception Mode)........ 444
Figure 13-11: Continuous Transfer Timing (Master Mode, Transmission/Reception Mode) 445
Figure 13-12: Continuous Transfer Timing (Master Mode, Reception Mode)........ 446
Figure 13-13: Continuous Transfer Timing (Error)........ 447
Figure 13-14: Continuous Transfer Timing (Slave Mode, Transmission/Reception Mode) 448
Figure 13-15: Continuous Transfer Timing (Slave Mode, Reception Mode)........ 449
Figure 13-16: Clock Timing (1/2) 450
Figure 13-17: Single Transmission Flow........ 453
Figure 13-18: Single Reception Flow (Master) 454
Figure 13-19: Single Transmission/Reception Flow (Master)........ 455
Figure 13-20: Single Reception Flow (Slave) 456
Figure 13-21: Continuous Transmission Flow 457
Figure 13-22: Continuous Reception Flow (Master) 458
Figure 13-23: Continuous Transmission/Reception Flow (Master)........ 459
Figure 13-24: Continuous Reception Flow (Slave) 460
Figure 13-25: Prescaler Mode Register 0 (PRSM0) Format 461
Figure 13-26: Prescaler Compare Register 0 (PRSCM0) Format 462
Figure 14-1: Queued CSI Block Diagram 464
Figure 14-2: Queued CSI Operation Mode Registers (CSIM0, CSIM1) Format (1/2) 466
Figure 14-3: Queued CSI Clock Selection Registers (CSIC0, CSIC1) Format (1/2) 468
Figure 14-4: Queued CSI Baud Rate Block Diagram 470
Figure 14-5: Receive Data Buffer Registers (SIRB0, SIRB1) Format 471
Figure 14-6: Chip Select Data Buffer Registers (SFCS0, SFCS1) Format 471
Figure 14-7: Transmission Data Buffer Registers (SFDB0, SFDB1) Format 472
Figure 14-8: FIFO Buffer Status Registers (SFA0, SFA1) Format (1/2) 472
Figure 14-9: Queued CSI Data Length Selection Registers (CSIL0, CSIL1) Format 474
Figure 14-10: Queued CSI Transfer Number Selection Registers (SFN0, SFN1) Format 475
Figure 14-11: Transmit Buffer........ 476
Figure 14-12: Serial Data Direction Select Function........ 477
Figure 14-13: Data Length Select Function 478
Figure 14-14: Slave Mode........ 479
Figure 14-15: Master Mode........ 479
Figure 14-16: Transfer Clock Select Function 480
Figure 14-17: Single Buffer Transfer Mode Data Handling........ 481
Figure 14-18: Single Buffer Transfer Mode (Master, Transmit/Receive) Timing 482
Figure 14-19: FIFO Buffer Transfer Mode Data Handling........ 483
Figure 14-20: FIFO Buffer Transfer Mode (Master, Transmit/Receive) Timing 484
Figure 14-21: Delay Selection of Receive Termination Interrupt (INTC3nI) 486
Figure 14-22: Selection of Transmit Wait Enable/Disable 487
Figure 14-23: Selection of Chip-Select Mode 488
Figure 14-24: Transmit Buffer Overflow Interrupt Signal (INTC3nO)........ 490
Figure 14-25: Single Buffer Transfer Mode (Master, Transmit Only) Timing........ 491

Figure 14-26: Single Buffer Transfer Mode (Master, Receive Only) Timing 492
Figure 14-27: Single Buffer Transfer Mode (Master, Transmit/Receive) Timing 493
Figure 14-28: Single Buffer Transfer Mode (Slave, Transmit Only) Timing 494
Figure 14-29: Single Buffer Transfer Mode (Slave, Receive Only) Timing 495
Figure 14-30: Single Buffer Transfer Mode (Slave, Transmit/Receive) Timing 496
Figure 14-31: FIFO Buffer Transfer Mode (Master, Transmit Only) Timing 497
Figure 14-32: FIFO Buffer Transfer Mode (Master, Receive Only) Timing 498
Figure 14-33: FIFO Buffer Transfer Mode (Master, Transmit/Receive) Timing 499
Figure 14-34: FIFO Buffer Transfer Mode (Slave, Transmit Only) Timing 500
Figure 14-35: FIFO Buffer Transfer Mode (Slave, Receive Only) Timing 501
Figure 14-36: FIFO Buffer Transfer Mode (Slave, Transmit/Receive) Timing 502
Figure 15-1: DMA Block Diagram 504
Figure 15-2: DMA Control Register (DMC) Format 505
Figure 15-3: IDMEN Bit and NMI Handling 506
Figure 15-4: DMA Channel Status Flag Register (DMSF) Format 507
Figure 15-5: DMA Source Address Register (DMSAn) Format (1/2) 507
Figure 15-6: DMA Destination Address Register (DMDAn) Format (1/2) 509
Figure 15-7: DMA Transfer Count Register (DMBCn) Format 510
Figure 15-8: DMA Addressing Control Register (DMADCn) Format (1/2) 511
Figure 15-9: Effect of TDIR Flag on DMA Transfer 512
Figure 15-10: DMA Channel Control Register (DMCHCn) Format (1/4) 513
Figure 15-11: DMA Trigger Factor Register (DTFRn) Format 517
Figure 15-12: TCS Bit and INTDMAn Generation 521
Figure 15-13: Single Transfer Mode Example (1 Channel) 522
Figure 15-14: Single Transfer Mode Example (3 Channels) 522
Figure 15-15: Fixed Channel Transfer Example (2 Channels) 523
Figure 15-16: Fixed Channel Transfer Mode Example (3 Channels) 523
Figure 15-17: Block Transfer Example (2 Channels) 524
Figure 16-1: Block Diagram of CAN Module 527
Figure 16-2: Composition of Layers 528
Figure 16-3: Data Frame 529
Figure 16-4: Remote Frame 530
Figure 16-5: Start of Frame (SOF) 530
Figure 16-6: Arbitration Field (in Standard Format Mode) 531
Figure 16-7: Arbitration Field (in Extended Format Mode) 531
Figure 16-8: Control Field 532
Figure 16-9: Data Field 533
Figure 16-10: CRC Field 533
Figure 16-11: ACK Field 534
Figure 16-12: End of Frame (EOF) 534
Figure 16-13: Interframe Space (Error Active Node) 535
Figure 16-14: Interframe Space (Error Passive Node) 536
Figure 16-15: Error Frame 537
Figure 16-16: Overload Frame 538
Figure 16-17: Recovery from Bus-off State Through Normal Recovery Sequence 545
Figure 16-18: Segment Setting 547
Figure 16-19: Reference: Configuration of Data Bit Time Defined by CAN Specification 548
Figure 16-20: Adjusting Synchronization of Data Bit 549
Figure 16-21: Resynchronization 550
Figure 16-22: Connection to CAN Bus 551
Figure 16-23: CAN Global Control Register (CnGMCTRL) Format (1/2) 588
Figure 16-24: CAN Global Clock Selection Register (CnGMCS) Format 590
Figure 16-25: CAN Global Automatic Block Transmission Control Register (CnGMABT) Format (1/2) 591
Figure 16-26: CAN Global Automatic Block Transmission Delay Register (CnGMABTD) Format . 593
Figure 16-27: CAN Module Mask Control Register (CnMASKaL, CnMASKaH) (a = 1, 2, 3, or 4) Format (1/2) 594
Figure 16-28: CAN Module Control Register (CnCTRL) Format (1/4) 596

Figure 16-29: CAN Module Last Error Information Register (CnLEC) Format 600
Figure 16-30: CAN Module Information Register (CnINFO) Format 601
Figure 16-31: CAN Module Error Counter Register (CnERC) Format 602
Figure 16-32: CAN Module Interrupt Enable Register (CnIE) Format (1/2) 603
Figure 16-33: CAN Module Interrupt Status Register (CnINTS) Format (1/2) 605
Figure 16-34: CAN Module Bit Rate Prescaler Register (CnBRP) Format 606
Figure 16-35: CAN Module Clock 607
Figure 16-36: CAN Module Bit Rate Register (CnBTR) Format (1/2) 608
Figure 16-37: CAN Module Last In-Pointer Register (CnLIPT) Format 610
Figure 16-38: CAN Module Receive History List Register (CnRGPT) Format (1/2) 611
Figure 16-39: CAN Module Last Out-Pointer Register (CnLOPT) Format 612
Figure 16-40: CAN Module Transmit History List Register (CnTGPT) Format (1/2)........................ 613
Figure 16-41: CAN Module Time Stamp Register (CnTS) Format (1/2).. 614
Figure 16-42: CAN Message Data Byte Register (CnMDATAxm) (x = 0 to 7) Format (1/2) 616
Figure 16-43: CAN Message Data Length Register m (CnMDLCm) Format 618
Figure 16-44: CAN Message Configuration Register (CnMCONFm) Format (1/2) 619
Figure 16-45: CAN message ID register m (CnMIDLm, CnMIDHm) Format 620
Figure 16-46: CAN Message Control Register m (CnMCTRLm) Format (1/3) 621
Figure 16-47: Example of Bit Setting/Clearing Operations 625
Figure 16-48: Bit Status After Bit Setting/Clearing Operations 625
Figure 16-49: Setting Transmission Request (TRQ) to Transmit Message Buffer
After Redefinition 627
Figure 16-50: Transition to Operation Modes 628
Figure 16-51: Receive History List 631
Figure 16-52: Message Processing Example 636
Figure 16-53: Transmit History List 639
Figure 16-54: CAN Module Terminal Connection in Receive-Only Mode.. 648
Figure 16-55: CAN Module Terminal Connection in Self-Test Mode... 650
Figure 16-56: Timing Diagram of Capture Signal TSOUT 651
Figure 16-57: Initialization.. 660
Figure 16-58: Re-initialization 661
Figure 16-59: Message Buffer Initialization.. 662
Figure 16-60: Message Buffer Redefinition 663
Figure 16-61: Transmitting Message Buffer Redefinition.. 664
Figure 16-62: Message Transmit Processing (Normal Operation Mode) 665
Figure 16-63: Message Transmit Processing (Normal Operation Mode with ABT) 666
Figure 16-64: Transmission via Interrupt (Using CnLOPT register).. 667
Figure 16-65: Transmission via Interrupt (Using CnTGPT Register) 668
Figure 16-66: Transmission via Software Polling.. 669
Figure 16-67: Transmission Abort Processing (except Normal Operation Mode with ABT) 670
Figure 16-68: Transmission Abort Processing Except for ABT Transmission
(Normal Operation Mode with ABT) 671
Figure 16-69: Transmission Abort Processing (Normal Operation Mode with ABT)........................ 672
Figure 16-70: Transmission Request Abort Processing (Normal Operation Mode with ABT) 673
Figure 16-71: Reception via Interrupt (Using CnLIPT Register) 674
Figure 16-72: Reception via Interrupt (Using CnRGPT Register).. 675
Figure 16-73: Reception via Software Polling.. 676
Figure 16-74: Setting CAN Sleep Mode/Stop Mode 677
Figure 16-75: Clear CAN Sleep/Stop Mode.. 678
Figure 16-76: Bus-Off Recovery 679
Figure 16-77: Normal Shutdown Process... 680
Figure 16-78: Forced Shutdown Process 680
Figure 16-79: Error Handling 681
Figure 16-80: Setting CPU Standby (from CAN Sleep Mode) 682
Figure 16-81: Setting CPU Standby (from CAN Stop Mode).. 683
Figure 17-1: Non-Maskable Interrupt Request Acknowledgement Operation (1/2) 688
Figure 17-2: Servicing Configuration of Non-Maskable Interrupt 690
Figure 17-3: RETI Instruction Processing 691

Figure 17-4: NP Flag Format ... 692
Figure 17-5: NMI Mode Register (NMIM) Format ... 693
Figure 17-6: Maskable Interrupt Servicing ... 695
Figure 17-7: RETI Instruction Processing ... 696
Figure 17-8: Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2) ... 698
Figure 17-9: Example of Servicing Interrupt Requests Simultaneously Generated ... 700
Figure 17-10: Interrupt Control Register (xxICn) Format ... 701
Figure 17-11: Interrupt Mask Registers 0 to 3 (IMR0 to IMR3) Format ... 704
Figure 17-12: In-Service Priority Register (ISPR) Format ... 705
Figure 17-13: Maskable Interrupt Status Flag Format ... 706
Figure 17-14: Watchdog Timer Mode Register 2 (WDTM2) Format ... 707
Figure 17-15: External Interrupt Falling Edge Specification Register 0 (INTF0) Format ... 709
Figure 17-16: External Interrupt Rising Edge Specification Register 0 (INTR0) Format ... 709
Figure 17-17: External Interrupt Falling Edge Specification Register 1 (INTF1) Format ... 710
Figure 17-18: External Interrupt Rising Edge Specification Register 1 (INTR1) Format ... 710
Figure 17-19: External Interrupt Falling Edge Specification Register 3 (INTF3) Format ... 711
Figure 17-20: External Interrupt Rising Edge Specification Register 3 (INTR3) Format ... 711
Figure 17-21: External Interrupt Falling Edge Specification Register 9H (INTF9H) Format ... 712
Figure 17-22: External Interrupt Rising Edge Specification Register 9H (INTR9H) Format ... 712
Figure 17-23: Noise Elimination Control Register Format ... 713
Figure 17-24: Software Exception Processing ... 714
Figure 17-25: RETI Instruction Processing ... 715
Figure 17-26: Exception Status Flag (EP) Format ... 716
Figure 17-27: Illegal Opcode Definition ... 717
Figure 17-28: Exception Trap Processing ... 718
Figure 17-29: Restore Processing from Exception Trap ... 718
Figure 17-30: Debug Trap Processing Format ... 719
Figure 17-31: Processing Format of Restoration from Debug Trap ... 720
Figure 17-32: Pipeline Operation at Interrupt Request Acknowledgement (Outline) ... 721
Figure 18-1: Status Transition ... 724
Figure 18-2: Standby Transition from PLL Operation (PLL = ON) ... 725
Figure 18-3: Standby Transition from X1 Through Mode (PLL = ON) ... 725
Figure 18-4: Standby Transition from X1 Through Mode (PLL = OFF) ... 726
Figure 18-5: IDLE Mode Timing ... 733
Figure 18-6: Oscillation Stabilization Time ... 736
Figure 18-7: Power Save Control Register (PSC) Format ... 737
Figure 18-8: Power Save Mode Register (PSMR) Format ... 738
Figure 19-1: Reset Source Flag Register (RESF) Format ... 740
Figure 19-2: Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input ... 742
Figure 19-3: Timing of Power-on Reset Operation ... 742
Figure 19-4: Timing of Reset Operation by WDT2RES Signal Generation ... 744
Figure 19-5: Timing of Reset Operation by Low-Voltage Detector ... 746
Figure 20-1: Regulator Block Diagram ... 747
Figure 20-2: REGC Pin Connection (REGC = Capacity) ... 748
Figure 21-1: Address Assignment of Flash Blocks for V850E/RS1 ... 751
Figure 21-2: Environment Required for Writing Programs to Flash Memory ... 752
Figure 21-3: Communication with Dedicated Flash Programmer (UARTA0) ... 753
Figure 21-4: Communication with Dedicated Flash Programmer (CSIB0) ... 754
Figure 21-5: Communication with Dedicated Flash Programmer (CSIB0 + HS) ... 754
Figure 21-6: FLMD0 Pin Connection Example ... 756
Figure 21-7: FLMD1 Pin Connection Example ... 757
Figure 21-8: Signal Conflict (Input Pin of Serial Interface) ... 758
Figure 21-9: Abnormal Operation of Other Device ... 759
Figure 21-10: Signal Conflict ($\overline{\text{RESET}}$ Pin) ... 760
Figure 21-11: Recommended Circuit Example ... 761
Figure 21-12: Procedure for Manipulating Flash Memory ... 762
Figure 21-13: Flash Memory Programming Mode ... 763

Figure 21-14: Communication Command 764
Figure 22-1: Block Diagram of On-chip Debug Function 769
Figure 22-2: On-Chip Debug Mode Register (OCDM) Format 771
Figure 22-3: P911/ $\overline{\text{DRST}}$ Built-in Pull Down Resistor 772
Figure 22-4: Timing Chart of Selecting Normal Operation Mode 773
Figure 22-5: Timing Chart of Selecting On-Chip Debug Mode 773
Figure 22-6: Connection to N-Wire Emulator (NEC Electronics IE-V850E1-CD-NW: N-Wire Card) 774
Figure 22-7: Pin Configuration of Connector for Emulator Connection (Target System Side)........ 775
Figure 22-8: Example of Recommended Emulator Connection Circuit 777
Figure 23-1: Block Diagram of Power-on-Clear Circuit........ 780
Figure 23-2: Timing of Internal Reset Signal Generation by Power-on-Clear Circuit 780
Figure 24-1: Block Diagram of Low-Voltage Detector........ 781
Figure 24-2: Low-Voltage Detection Register (LVIM) Format 782
Figure 24-3: Low-Voltage Detection Level Selection Register (LVIS) Format 783
Figure 24-4: Internal RAM Data Status Register (RAMS) Format 783
Figure 24-5: Peripheral Emulation Register 1 (PEMU1) 784
Figure 24-6: Operation Timing of Low-Voltage Detector (LVIMD = 1)........ 786
Figure 24-7: Operation Timing of Low-Voltage Detector (LVIMD = 0)........ 787
Figure 24-8: Operation Timing of RAM Retention Voltage Detection Function 788
Figure 25-1: Block Diagram of Clock Monitor 789
Figure 25-2: Clock Monitor Mode Register (CLM) Format 790
Figure 25-3: When Oscillation of Main Clock Is Stopped........ 791
Figure 25-4: Operation in Software STOP Mode or After Software STOP Mode Is Released 792
Figure 26-1: Block Diagram of CRC Register........ 793
Figure 26-2: CRC Input Register (CRCIN) Format 794
Figure 26-3: CRC Data Register (CRCD) Format 794
Figure 26-4: CRC Operation Circuit Operation Example (LSB First)........ 795
Figure 26-5: CRC Operation Flow 796
Figure 26-6: CRC Transmission Example 797
Figure 27-1: Recommended Oscillator Connection (Ceramic or Crystal Resonator) 802
Figure 27-2: Voltage Regulator Startup Timing 803
Figure 27-3: AC Test Conditions........ 807
Figure 27-4: Input Rise and Fall Time........ 808
Figure 27-5: Output Rise and Fall Time........ 808
Figure 27-6: RESET, Interrupt, NMI and FLMD0 Timing 809
Figure 27-7: Read Cycle (CLKOUT Asynchronous, 1 Wait)........ 812
Figure 27-8: Write Cycle (CLKOUT Asynchronous, 1 $\overline{\text{WAIT}}$) 813
Figure 27-9: Bus Hold........ 814
Figure 27-10: CSIBn Timing (CBnCKP=0, CBnDAP=0)........ 816
Figure 27-11: CAN Internal Timing 817
Figure 27-12: CSI3n Timings (1/2)........ 819
Figure 27-13: CS3n3 - CS3n0 Pins Timings (1/3) 821
Figure 27-14: Power-on-clear (POC) Waveform........ 825
Figure 27-15: Low-Voltage Indicator (LVI) Waveform........ 826
Figure 27-16: Power on Sequence 827
Figure 27-17: Flash EPROM Serial Programming Operation Characteristics 829
Figure 28-1: Package Drawing 831

List of Tables

Table 1-1: Product Versions 30
Table 1-2: Port Functions and Control Function 38
Table 2-1: Pin of Power Supplies 39
Table 2-2: Port Pins 39
Table 2-3: Non-port pins 41
Table 2-4: Pin Operation States in Various Modes 44
Table 2-5: Pin I/O Circuit Types 57
Table 3-1: Program Registers 63
Table 3-2: System Register Numbers 64
Table 3-3: Access Conditions 90
Table 3-4: Peripheral I/O Registers 93
Table 4-1: I/O Buffer Power Supplies for Pins 106
Table 4-2: Control Register Setting 107
Table 4-3: Port Configuration 110
Table 4-4: Valid Edge Specification 117
Table 4-5: Valid Edge Specification 123
Table 4-6: Valid Edge Specification 132
Table 4-7: Valid Edge Specification 154
Table 4-8: Port Type 170
Table 5-1: Bus Control Pins (Multiplexed Bus) 208
Table 5-2: Pin Status When Internal ROM, Internal RAM, or Peripheral I/O Is Accessed 208
Table 5-3: Allocation of the Memory Blocks 210
Table 5-4: Bus Priority 225
Table 6-1: Divide and PLL0 Time Value 252
Table 6-2: Divide and PLL1 Time Value 252
Table 6-3: Divide Value of f_{PLL} Frequency and f_{PCL} Frequency 254
Table 7-1: Configuration of TMP0 to TMP3 256
Table 7-2: TMP Pin List 256
Table 7-3: Tuned Operation Mode of Timer 299
Table 7-4: Timer Modes Usable in Tuned Operation Mode 299
Table 7-5: Timer Output Functions 300
Table 8-1: TMQ Configuration 304
Table 8-2: TMQ Pin List 304
Table 8-3: Tuned Operation Mode of Timer 351
Table 8-4: Timer Modes Usable in Tuned Operation Mode 351
Table 8-5: Timer Output Functions 352
Table 9-1: Configuration of TMM 356
Table 10-1: Configuration of Watchdog Timer 2 360
Table 10-2: Watchdog Timer 2 Clock Selection 362
Table 11-1: Configuration of A/D Converter 369
Table 11-2: Software Trigger Mode (ADSCM0H Register Configuration) 381
Table 11-3: External Trigger Mode (ADSCM0H and SELCNT1 Registers Configuration) 381
Table 12-1: Configuration of UARTA0 and UARTA1 404
Table 12-2: List of Pins of Asynchronous Serial Interface A 404
Table 12-3: Interrupts and Their Default Priority 415
Table 12-4: Reception Error Causes 421
Table 12-5: Baud Rate Generator Set Data 427
Table 12-6: Permissible Maximum/Minimum Baud Rate Error 429
Table 13-1: Configuration of CSIB0, CSIB1 431
Table 13-2: List of 3-Wire Serial Interface Pins 432
Table 13-3: Interrupts and their Default Priority 442
Table 14-1: Input/Output Pins of the CSI3 464
Table 14-2: CSI30 465
Table 14-3: CSI31 465
Table 15-1: Interrupt Source for DMA Trigger Factor Register (DTFRn) 518

Table 15-2: Transfer Targets ... 520
Table 15-3: Comparison of DMA Transfer Modes ... 524
Table 16-1: Overview of Functions ... 526
Table 16-2: Frame Types ... 529
Table 16-3: RTR Frame Settings ... 531
Table 16-4: Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits ... 532
Table 16-5: Data Length Setting ... 532
Table 16-6: Operation in Error Status ... 536
Table 16-7: Definition of Error Frame Fields ... 537
Table 16-8: Definition of Overload Frame Fields ... 538
Table 16-9: Determining Bus Priority ... 539
Table 16-10: Bit Stuffing ... 539
Table 16-11: Error Types ... 540
Table 16-12: Output Timing of Error Frame ... 541
Table 16-13: Types of Error States ... 542
Table 16-14: Error Counter ... 543
Table 16-15: List of CAN Controller Registers ... 552
Table 16-16: Register Access Type ... 553
Table 16-17: CAN Module Register Bit Configuration ... 584
Table 16-18: CAN Global Register Bit Configuration ... 586
Table 16-19: Message Buffer Register Bit Configuration ... 587
Table 16-20: List of CAN Module Interrupt Sources ... 647
Table 16-21: Settable Bit Rate Combinations ... 653
Table 16-22: Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) ... 656
Table 16-23: Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) ... 658
Table 17-1: Interrupt/Exception Source List ... 685
Table 17-2: Interrupt Control Register (xxICn) ... 702
Table 17-3: Watchdog Timer 2 Clock Selection ... 708
Table 17-4: Valid Edge Specification ... 709
Table 17-5: Valid Edge Specification ... 710
Table 17-6: Valid Edge Specification ... 711
Table 17-7: Valid Edge Specification ... 712
Table 18-1: Standby Modes ... 723
Table 18-2: Operation After Releasing HALT Mode by Interrupt Request ... 727
Table 18-3: Operation Status in HALT Mode ... 728
Table 18-4: Operation After Releasing IDLE1 Mode by Interrupt Request ... 730
Table 18-5: Operation Status in IDLE1 Mode ... 730
Table 18-6: Operation After Releasing IDLE2 Mode by Interrupt Request ... 732
Table 18-7: Operation Status in IDLE2 Mode ... 732
Table 18-8: Operation After Releasing STOP Mode by Interrupt Request ... 735
Table 18-9: Operation Status in STOP Mode ... 735
Table 19-1: Hardware Status on $\overline{\text{RESET}}$ Pin Input ... 741
Table 19-2: Hardware Status During WDT2RES Signal Generation ... 743
Table 19-3: Hardware Status During Reset Operation by Low-Voltage Detector ... 745
Table 21-1: Signal Generation of Dedicated Flash Programmer (PG-FP4) ... 755
Table 21-2: Relationship Between FLMD0 and FLMD1 Pins and Operation Mode ... 757
Table 21-3: Pins Used by each Serial Interface ... 758
Table 21-4: List of Communication Modes ... 764
Table 21-5: Flash Memory Control Command ... 765
Table 21-6: Response Commands ... 765
Table 22-1: ID Code ... 770
Table 22-2: Pin Functions of Connector for Emulator Connection (Target System Side) ... 776
Table 25-1: Configuration of Clock Monitor ... 789
Table 25-2: Operation Status of Clock Monitor (When CLM.CLME Bit = 1, During Ring-OSC Operation) (CKSEL Connected to Ring-OSC) ... 791
Table 26-1: CRC Configuration ... 793
Table 29-1: Soldering Conditions ... 833

Chapter 1 Introduction

The V850E/RS1 is a product in the V850 Series of NEC Electronics Corporation single-chip microcontrollers. This chapter provides an overview of the V850E/RS1.

1.1 Outline

The V850E/RS1single chip microcontroller is a member of NEC's V850 32-bit RISC family, which match the performance gains attainable with RISC-based controllers to the needs of embedded control applications. The V850E CPU offers easy pipeline handling and programming, resulting in compact code size comparable to 16-bit CISC CPUs.

The V850E/RS1 offers an excellent combination of general purpose peripheral functions, like serial communication interfaces (UART, clocked SI, clocked SI with translation buffer), timers and measurement inputs (A/D converter), with dedicated CAN network support. The device offers power-saving modes to manage the power consumption effectively under varying conditions. Thus equipped, the V850E/RS1 is ideally suited for automotive applications.

(1) Implementation of V850E CPU

The V850E CPU supports a RISC instruction set in which instruction execution speeds are increased greatly through the use of basic instructions that execute one instruction per clock and optimized pipelines.
Object code efficiency is increased in the C compiler by using 2-byte length basic instructions and instructions corresponding to high-level languages, which makes a program compact.
Furthermore, since interrupt response time including processing by the on-chip interrupt controller also is fast, this CPU is suited to the realm of advanced real-time control.

(2) Peripheral I/O function

The V850E/RS1 includes various function Timer (TMQ, TMP, TMM) and, Serial interface (CSI3, CSIB, UART) and CAN controller. CSI3 includes translation buffer (20 bits x 16 depth).

(3) External bus interface function (µPD70F3403 and µPD70F3403A)

The µPD70F3403 and µPD70F3403A are provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

Cautions: 1. Do not set up any external bus interface configuration to the µPD70F3402.

2. External bus interface functions are only described for the µPD70F3403 and µPD70F3403A microcontrollers.

(4) On-chip flash memory

The µPD70F3402, µPD70F3403 and µPD70F3403A have on-chip an high speed flash memory, which is able to fetch one instruction within one clock cycle. It is possible to program the user application directly on the target board, on which the V850E/RS1 is mounted. In such case, system development time can be reduced and system maintainability after shipping can be markedly improved.

(5) A full range of development environment products

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

1.2 Features

- CPU
  - core : V850E
  - Number of instructions : 83
  - Minimum instruction execution time : 31.25 ns (during 32 MHz operations) for 3402
    : 25 ns (during 40 MHz operations) for 3403 and 3403A
  - General purpose registers : 32 bits × 32 registers

- Instruction set optimized for control operations:
  - V850E
  - Signed multiplication
    (16 bits × 16 bits→ 32 bits or 32 bits × 32 bits → 64 bits): 1 to 2 clocks
  - Saturated operation instructions (with overflow/underflow detection function)
  - 32-bit shift instructions: 1 clock
  - Bit manipulation instructions
  - Load/store instructions with long/short format
  - Signed load instructions

- Internal memory

Table 1-1: Product Versions

Part Number	Internal Memory		CAN Interface (ch)	CPU speed
	Flash Memory (KB)	RAM (KB)		
µPD70F3402	128	10	2	32 MHz (max.)
µPD70F3403	256	16	2	40 MHz (max.)
µPD70F3403A				

- Power saving function
  - 2 × IDLE (IDLE1/IDLE2), HALT, and STOP

- Clock generator
  - $f_{XX} = f_X$ (f_X = 4 to 8 MHz) : direct mode
  - $f_{XX} = f_{PLL}$ = 24 MHz, 32 MHz : PLL mode for µPD70F3402
  - $f_{XX} = f_{PLL}$ = 24 MHz, 32 MHz, 40 MHz : PLL mode for µPD70F3403 and µPD70F34033A

- Power supply voltage range
  - 4.0 V to 5.5 V : 24 MHz and 32 MHz max. for µPD70F3402
  - 4.0 V to 5.5 V : 24 MHz, 32 MHz and 40 MHz max. for µPD70F3403 and µPD70F3403A

- Operating temperature
  - µPD70F3403 and µPD70F3403A : T_A = –40°C to +85°C @ f_{XX} = 40 MHz max.
  - µPD70F3402 : T_A = –40°C to +110°C @ f_{XX} = 32 MHz max.

- Real-time pulse units:
  - 16-bit timer/counter (Type TMQ) : 2 channels
  - 16-bit timer/counter (Type TMP) : 4 channels
  - 16-bit interval timer (type TMM) : 1 channel
  - Watchdog timer : 1 channel

- Serial interface:
  - UART/LIN : 2 channels
  - CSIB : 2 channels
  - CSI3 : 2 channels

• CAN (2.0B active)	: 2 channels × 32 message buffers Note
• A/D converter	
- 10-bit resolution	: 16 inputs (with auto discharge function and diagnostic mode)
• Interrupts	: 60 vectored interrupts
- Non-maskable interrupts	: 2 sources (NMI, Watchdog)
- Maskable interrupts	: External: 8, Internal: 51 sources

- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 2 sources (illegal opcode exception)

• DMA transfer function	: 6 independent channels

- On-chip power on reset
- On-chip low voltage indicator
- On-chip ring oscillator for security (Type. 220 kHz)
- On-chip debug function
- On-chip main clock monitor

• Package	: 100-pin plastic LQFP (14 × 14)

Note: The CAN macro of this device fulfils the requirements according ISO 11898. Additionally the CAN macro was tested according to the test procedures required by ISO 16845. The CAN macro successfully passed all test patterns. Beyond these test patterns, other tests like robustness tests and processor interface tests as recommended by C&S/FH Wolfenbuettel have successfully been issued.

1.3 Applications

The V850E/RS1 is a device designed for car manufacturers. It is ideally suited for automotive applications, like Safety System. It is also an excellent choice for other applications where a combination of sophisticated peripheral functions with CAN network support is required like Body Electronics Application.

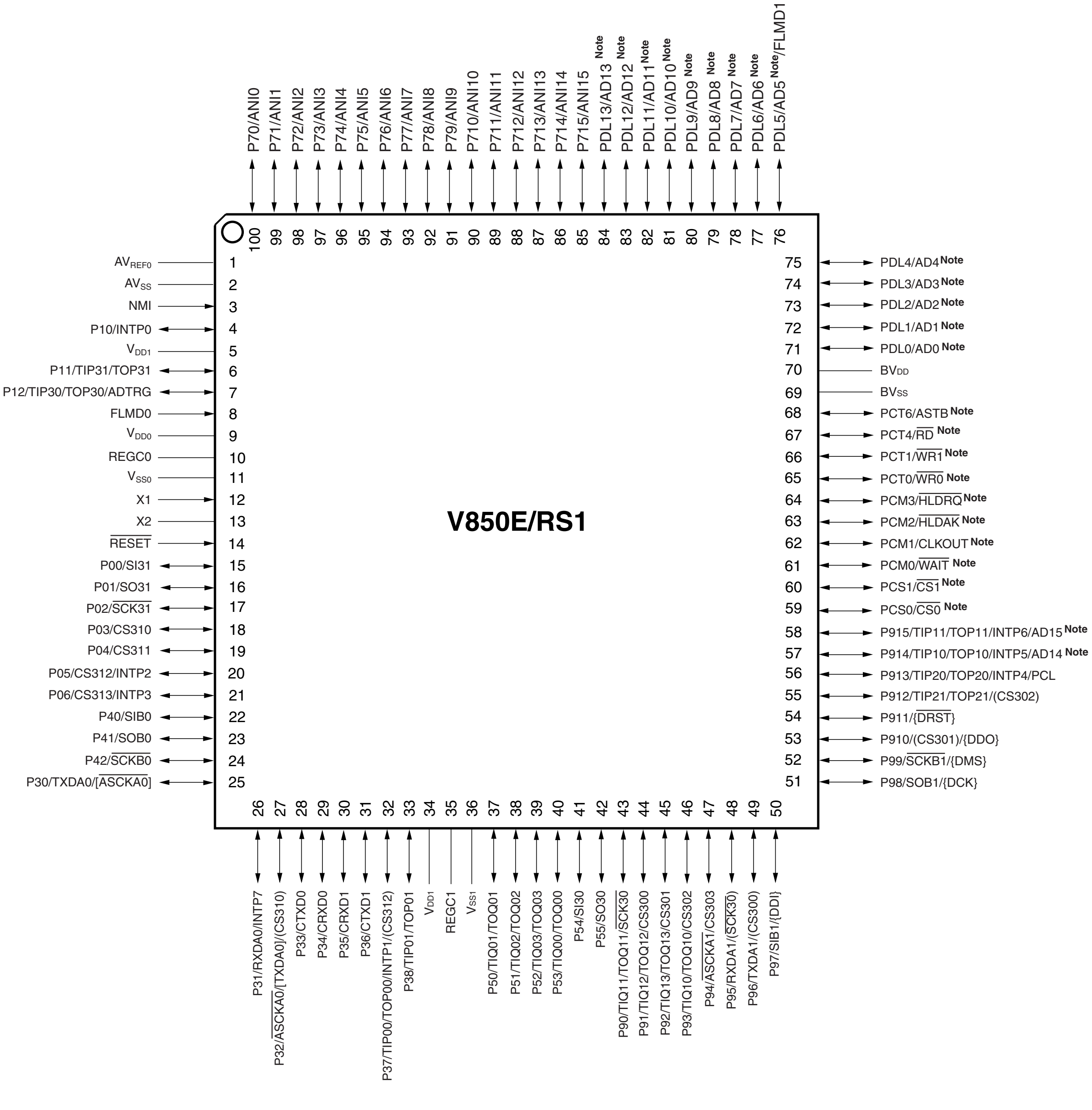
1.4 Ordering Information

Part No.	Package	Quality Grade
µPD70F3402GC(A1)-8EA	100-pin plastic LQFP (14 × 14)	Special
µPD70F3403GC(A)-8EA	100-pin plastic LQFP (14 × 14)	Special
µPD70F3403AGC(A)-8EA	100-pin plastic LQFP (14 × 14)	Special

1.5 Pin Configuration (Top View)

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

Figure 1-1: Pin Configuration



Note: External bus interface is available only for 3403 and 3403A

Pin Identification

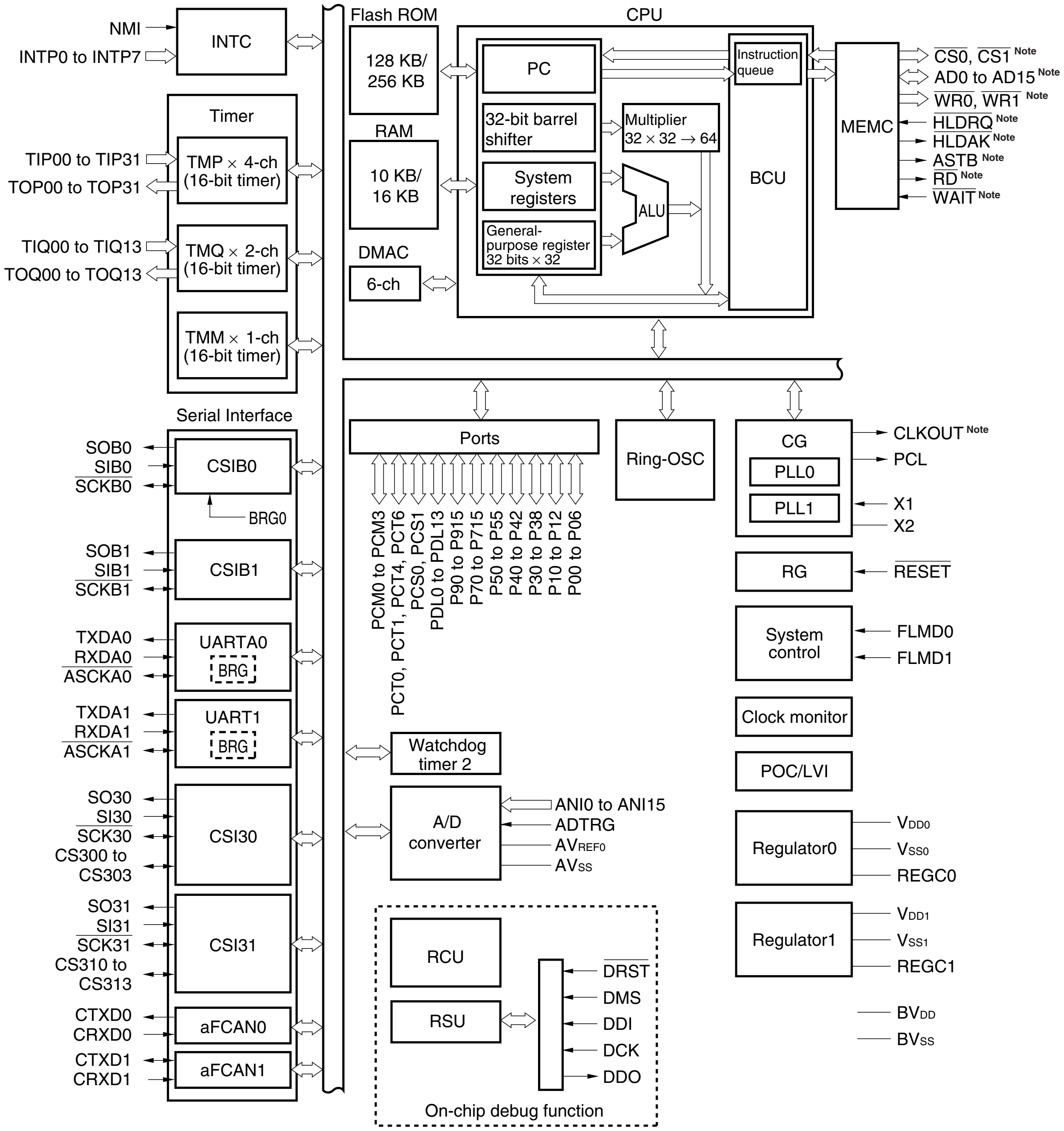
AD0 to AD15[Note]	Address/data bus	PCL	Programmable clock output
ADTRG	A/D trigger input	PCM0 to PCM3	Port CM
ANI0 to ANI15	Analog input	PCT0, PCT1,	Port CT
$\overline{\text{ASCKA0}}$ to $\overline{\text{ASCKA1}}$	Asynchronous serial clock	PCT4, PCT6	
ASTB[Note]	Address strobe	PDL0 to PDL13	Port DL
AV_{REF0}	Analog reference voltage	$\overline{\text{RD}}$[Note]	Read strobe
AV_{SS}	Analog ground	REGC0, REGC1	Regulator control
BV_{DD}	Power supply for bus interface	$\overline{\text{RESET}}$	Reset
BV_{SS}	Ground for bus interface	RXDA0 to RXDA1	Receive data
CLKOUT[Note]	Clock output	$\overline{\text{SCK30}}$, $\overline{\text{SCK31}}$	Serial clock
CTXD0, CTXD1	Transmit data for controller area network	$\overline{\text{SCKB0}}$ to $\overline{\text{SCKB1}}$	Serial clock
CRXD0, CRXD1	Receive data for controller area network	SI30, SI31	Serial input
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$[Note]	Chip select	SIB0 to SIB1	Serial input
CS300 to CS303	Serial Chip Select	SO30, SO31	Serial output
CS310 to CS313	Serial Chip Select	SOB0 to SOB1	Serial output
DCK	Debug clock	TIP00, TIP01,	Timer input
DDI	Debug data input	TIP10, TIP11,	
DDO	Debug data output	TIP20, TIP21,	
DMS	Debug mode select	TIP30, TIP31,	
DRST	Debug reset	TIQ00 to TIQ03,	
FLMD0, FLMD1	Flash programming mode	TIQ10 to TIQ13	
$\overline{\text{HLDAK}}$[Note]	Hold acknowledge	TOP00, TOP01,	Timer output
$\overline{\text{HLDRQ}}$[Note]	Hold request	TOP10, TOP11,	
INTP0 to INTP7	Interrupt request from peripherals	TOP20, TOP21,	
		TOP30, TOP31,	
NMI	Non-maskable interrupt request	TOQ00 to TOQ03,	
P00 to P06	Port 0	TOQ10 to TOQ13	
P10 to P12	Port 1	TXDA0 to TXDA1	Transmit data
P30 to P38	Port 3	V_{DD0}, V_{DD1}	Power supply
P40 to P42	Port 4	V_{SS0}, V_{SS1}	Ground
P50 to P55	Port 5	$\overline{\text{WAIT}}$[Note]	Wait
P70 to P715	Port 7	$\overline{\text{WR0}}$[Note]	Write strobe low level data
P90 to P915	Port 9	$\overline{\text{WR1}}$[Note]	Write strobe high level data
PCS0, PCS1	Port CS	X1, X2	Crystal for main clock

Note: External bus interface is available only for 3403 and 3403A

1.6 Configuration of Function Block

1.6.1 Internal block diagram

Figure 1-2: Internal Block Diagram



Note: External bus interface is available only for 3403 and 3403A

1.6.2 On-chip units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.
Other dedicated on-chip hardware, such as the multiplier (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits), barrel shifter (32-bit), and other dedicated hardware are on-chip to accelerate complex instruction processing.

(2) Bus control unit (BCU)

The BCU starts a required external bus cycle based on a physical address obtained from the CPU. If there is no bus cycle start request from the CPU when fetching an instruction from an external memory area, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is fetched into the internal instruction queue of the CPU.

(3) Memory controller (MEMC)

The MEMC controls SRAM, ROM, and various I/O for external memory expansion.

(4) DMA controller (DMAC)

A 6-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.

(5) ROM

The µPD70F3402 has an on-chip flash memory of 128 Kbytes.
The µPD70F3403 and µPD70F3403A have an on-chip flash memory of 256 Kbytes.
The 128 KB or 256 KB flash memory are mapped to the address spaces from 0000000H to 001FFFFH, or 0000000H to 003FFFFH, respectively.
During instruction fetch, flash memory can be accessed from the CPU in 1-clock cycle.

(6) RAM

This consists of a 10 KB or 16 KB RAM mapped to the address spaces from 3FFC800H to 3FFEFFFH or 3FFB000H to 3FFEFFFH, respectively.
During instruction fetch or data access RAM memory can be accessed from the CPU in 1-clock cycle.

(7) Interrupt controller (INTC)

The INTC services hardware interrupts requests from on-chip peripheral I/O and external sources. Eight levels of interrupt priorities can be specified for this interrupt requests, and multiprocessing controls against the interrupt sources can be performed.

(8) Clock generator (CG)

This clock generator supplies frequencies for the CPU and the built-in peripherals. As the input clock, an external oscillator is connected to pins X1 and X2.
The clock generator has 2 PLL which could be independently assigned either to the peripherals or to the CPU. It generates seven types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{ring}), and supplies one of them as the operating clock for the CPU (f_{CPU}).

(9) Ring-OSC

A Ring-OSC is provided on chip. The oscillation frequency is 220 kHz (TYP.). This Ring-OSC supplies the clock for the watchdog timer 2 and timer TMM.

(10) Timer/counter

A two-channel 16-bit timer/event counter (TMQ), a four-channel 16-bit timer/event counter (TMP), and a one-channel 16-bit interval timer (TMM), are provided on chip.

(11) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the Ring-OSC, the main clock can be selected as the source clock. Watchdog timer 2 generates a non-maskable interrupt request (INTWDT) or a reset signal after an overflow occurs.

(12) Serial interface

The V850E/RS1 includes four kinds of serial interfaces: asynchronous serial interface A (UARTA), and 3-wire variable-length serial interface B (CSIB), 3 wire variable length serial interface 3 with transmit and receive buffer (CSI3) and maximum two-channel CAN are provided as serial interfaces.
In the case of UARTA, data is transferred via the TXDAn pins and RXDAn pins (n = 0, 1).
In the case of CSIB, data is transferred via the SOBm pins, SIBm pins, and SCKBm pins
(m = 0, 1).
In the case of CSI3, data is transferred via the SO3m pins, SIm pins, and SCK3m pins (m = 0, 1).
A dedicated baud rate generator is provided on chip for CSIB, CSI3 and UARTA.
In the case of CAN, data is transferred via the CTXDn pins and CRXDn pins (n = 0, 1).

(13) A/D converter

This high-speed, high-resolution 10-bit A/D converter includes 16 analog input pins with auto-discharge function and diagnostic mode. Conversion is performed using the successive approximation method.

(14) CRC function

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

(15) On-chip debug function

An on-chip debug function via an N-wire-type in-circuit emulator 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 on-chip debug mode setting register (OCDM).

(16) Port

There are general-purpose port functions and control pin functions, as listed below.

Table 1-2: Port Functions and Control Function

Port	I/O	Port Function	Control Function
P0	7-bit I/O	General-purpose port	Serial interface I/O, external interrupt
P1	3-bit I/O		External interrupt, Timer I/O
P3	9-bit I/O		External interrupt, serial interface I/O, timer I/O
P4	3-bit I/O		Serial interface I/O
P5	6-bit I/O		Timer I/O, Serial interface I/O
P7	16-bit I/O		A/D converter analog input
P9	16-bit I/O		Serial interface I/O, timer I/O, external interrupt, clock output, on chip debug I/O
PCS	2-bit I/O		Chip select signal
PCM	4-bit I/O		External bus interface
PCT	4-bit I/O		External bus interface
PDL	14-bit I/O		External address/data bus

Chapter 2 Pin Functions

2.1 List of Pin Functions

The names and functions of V850E/RS1 pins are described below. These pins can be divided into port pins and non-port pins according to their functions. Table 2-1 shows the relationship between power supplies and the pins below.

Table 2-1: Pin of Power Supplies

Power Supply	Corresponding Pin
V_{DD0},V_{SS0}, REGC0	Regulator for Clock Generator and PLL
AV_{REF0}, AV_{SS}	Reference voltage and ground potential for A/D converter. Port 7
BV_{DD}, BV_{SS}	Port CM, Port CS, port CT, port DL
V_{DD1}, V_{SS1}, REGC1	Port 0, Port 1, Port 3, Port 4, Port 5, Port 9, NMI, $\overline{\text{RESET}}$ and Regulator for Internal circuit (CPU, RAM and FLASH memory).

(1) Port pins

Table 2-2: Port Pins (1/2)

Pin Name	I/O	Function	Alternate Function
P00	I/O	Port 0 7-bit I/O port Input/output can be specified in 1-bit units.	SI31
P01			SO31
P02			$\overline{\text{SCK31}}$
P03			CS310
P04			CS311
P05			CS312/INTP2
P06			CS313/INTP3
P10	I/O	Port 1 3-bit I/O port Input/output can be specified in 1-bit units.	INTP0
P11			TIP31/TOP31
P12			TIP30/TOP30/ADTRG
P30	I/O	Port 3 9-bit I/O port Input/output can be specified in 1-bit units.	TXDA0/[$\overline{\text{ASCKA0}}$]
P31			RXDA0/INTP7
P32			$\overline{\text{ASCKA0}}$/[TXDA0]/(CS310)
P33			CTXD0
P34			CRXD0
P35			CRXD1
P36			CTXD1
P37			TIP00/TOP00/INTP1/(CS312)
P38			TIP01/TOP01
P40	I/O	Port 4 3-bit I/O port Input/output can be specified in 1-bit units.	SIB0
P41			SOB0
P42			$\overline{\text{SCKB0}}$

Table 2-2: Port Pins (2/2)

Pin Name	I/O	Function	Alternate Function
P50	I/O	Port 5 6-bit I/O port Input/output can be specified in 1-bit units.	TIQ01/TOQ01
P51			TIQ02/TOQ02
P52			TIQ03/TOQ03
P53			TIQ00/TOQ00
P54			SI30
P55			SO30
P70 to P715	I/O	Port 7 16-bit I/O port Input/output can be specified in 1-bit units.	ANI0 to ANI15
P90	I/O	Port 9 16-bit I/O port Input/output can be specified in 1-bit units.	TIQ11/TOQ11/$\overline{\text{SCK30}}$
P91			TIQ12/TOQ12/CS300
P92			TIQ13/TOQ13/CS301
P93			TIQ10/TOQ10/CS302
P94			$\overline{\text{ASCKA1}}$/CS303
P95			RXDA1/($\overline{\text{SCK30}}$)
P96			TXDA1/(CS300)
P97			SIB1/[DDI]
P98			SOB1/[DCK]
P99			$\overline{\text{SCKB1}}$/[DMS]
P910			(CS301)/[DDO]
P911			[$\overline{\text{DRST}}$]
P912			TIP21/TOP21/(CS302)
P913			TIP20/TOP20/INTP4/PCL
P914			TIP10/TOP10/INTP5/AD14 **Note**
P915			TIP11/TOP11/INTP6/AD15 **Note**
PCM0	I/O	Port CM 4-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WAIT}}$ **Note**
PCM1			CLKOUT **Note**
PCM2			$\overline{\text{HLDAK}}$ **Note**
PCM3			$\overline{\text{HLDRQ}}$ **Note**
PCS0, PCS1	I/O	Port CS 2-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$ **Note**
PCT0	I/O	Port CT 4-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WR0}}$ **Note**
PCT1			$\overline{\text{WR1}}$ **Note**
PCT4			$\overline{\text{RD}}$ **Note**
PCT6			ASTB **Note**
PDL0 to PDL4	I/O	Port DL 14-bit I/O port Input/output can be specified in 1-bit units.	AD0 to AD4 **Note**
PDL5			AD5/FLMD1 **Note**
PDL6 to PDL13			AD6 to AD13 **Note**

Note: External bus function port pins are only supported by the µPD70F3403 and µPD70F3403A

(2) Non-port pins

Table 2-3: Non-port pins (1/3)

Pin Name	I/O	Function	Alternate Function
NMI	Input	External interrupt (non-maskable, analog, noise elimination)	–
INTP0	Input	External interrupt request input (maskable, analog, noise elimination)	P10
INTP1			P37/TIP00/TOP00/(CS312)
INTP2			P05/CS312
INTP3			P06/CS13
INTP4			P913/TIP20/TOP20/PCL
INTP5			P914/TIP10/TOP10/AD14
INTP6			P915/TIP11/TOP11/AD15
INTP7			P31/RXDA0
TIP00	Input	External event/clock input (TMP00)	P37/TOP00/INTP1/(CS312)
TIP01		External event/clock input (TMP01)	P38/TOP01
TIP10		External event/clock input (TMP10)	P914/TOP10/INTP5/AD14
TIP11		External event/clock input (TMP11)	P915TOP11/INTP6/AD15
TIP20		External event/clock input (TMP20)	P913/TOP20/INTP4/PCL
TIP21		External event/clock input (TMP21)	P912/TOP21/(CS302)
TIP30		External event/clock input (TMP30)	P11/TOP31
TIP31		External event/clock input (TMP31)	P12/TOP30/ADTRG
TOP00	Output	Timer output (TMP00)	P37/TIP00/INTP1/(CS312)
TOP01		Timer output (TMP01)	P38/TIP01
TOP10		Timer output (TMP10)	P914/TIP10/INTP5/AD14
TOP11		Timer output (TMP11)	P915/TIP11/INTP6/AD15
TOP20		Timer output (TMP20)	P913/TIP20/INTP4/PCL
TOP21		Timer output (TMP21)	P912 /TIP21/(CS302)
TOP30		Timer output (TMP30)	P11/TIP31
TOP31		Timer output (TMP31)	P12/TIP30/ADTRG
TIQ00	Input	External event/clock input (TMQ00)	P53/ TOQ00
TIQ01		External event/clock input (TMQ01)	P50/TOQ01
TIQ02		External event/clock input (TMQ02)	P51/TOQ02
TIQ03		External event/clock input (TMQ03)	P52/TOQ03
TIQ10		External event/clock input (TMQ10)	P93/CS302/TOQ10
TIQ11		External event/clock input (TMQ11)	P90/$\overline{\text{SCK30}}$/TOQ11
TIQ12		External event/clock input (TMQ12)	P91/CS300/TOQ12
TIQ13		External event/clock input (TMQ03)	P92/CS301/TOQ13

Table 2-3: Non-port pins (2/3)

Pin Name	I/O	Function	Alternate Function
TOQ00	Output	Timer output (TMQ00)	P53/TIQ00
TOQ01		Timer output (TMQ01)	P50/TIQ01
TOQ02		Timer output (TMQ02)	P51/TIQ02
TOQ03		Timer output (TMQ03)	P52/TIQ03
TOQ10		Timer output (TMQ10)	P93/CS302/TIQ10
TOQ11		Timer output (TMQ11)	P90/$\overline{\text{SCK30}}$/TIQ11
TOQ12		Timer output (TMQ12)	P91/CS300/TIQ12
TOQ13		Timer output (TMQ13)	P92/CS301/TIQ13
SIB0	Input	Serial receive data input (CSIB0)	P40
SIB1		Serial receive data input (CSIB1)	P97/[DDI]
SI30		Serial receive data input (CSI30)	P54
SI31		Serial receive data input (CSI31)	P00
SOB0	Output	Serial transmit data output (CSIB0)	P41
SOB1		Serial transmit data output (CSIB1)	P98/[DCK]
SO30		Serial transmit data output (CSI30)	P55
SO31		Serial transmit data output (CSI31)	P01
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99/[DMS]
$\overline{\text{SCK30}}$		Serial clock I/O (CSI30)	P90/TIQ11/TOQ11, P95/RXDA1
$\overline{\text{SCK31}}$		Serial clock I/O (CSI31)	P02
CS300	Output	Serial chip select output (CSI30)	P91/TIQ12/TOQ12, P96/TXDA1
CS301		Serial chip select output (CSI30)	P92/TIQ13/TOQ13
CS302		Serial chip select output (CSI30)	P93/TIQ10/TOQ10
CS303		Serial chip select output (CSI30)	P94/$\overline{\text{ASCKA1}}$
CS310		Serial chip select output (CSI31)	P03, P32
CS311		Serial chip select output (CSI31)	P04
CS312		Serial chip select output (CSI31)	P05/INTP2, P37/TIP00/TOP00/ INTP1
CS313		Serial chip select output (CSI31)	P06/INTP3
RXDA0	Input	Serial receive data input (UARTA0)	P31/INTP7
RXDA1		Serial receive data input (UARTA1)	P95/($\overline{\text{SCK30}}$)
TXDA0	Output	Serial transmit data output (UARTA0)	P30/[$\overline{\text{ASCKA0}}$]
TXDA1		Serial transmit data output (UARTA1)	P96/(CS300)
$\overline{\text{ASCKA0}}$	I/O	UARTA0 baud rate clock input	P32/[TXDA0]/(CS310), P30/ TXDA0
$\overline{\text{ASCKA1}}$		UARTA1 baud rate clock input	P94/CS303
CRXD0	Input	CAN serial receive data input (CAN0)	P34
CRXD1		CAN serial receive data input (CAN1)	P35
CTXD0	Output	CAN serial transmit data output (CAN0)	P33
CTXD1		CAN serial transmit data output (CAN1)	P36
ANI0 to ANI15	Input	Analog voltage input for A/D converter	P70 to P715

Table 2-3: Non-port pins (3/3)

Pin Name	I/O	Function	Alternate Function
AV_{REF0}	Input	Reference voltage input for A/D converter (same potential as V_{DD})	–
AV_{SS}	-	Ground potential for A/D converter (same potential as V_{SS})	–
ADTRG	Input	A/D converter external trigger input	P12/TIP30/TOP30
DDI	Input	Debug data input	P97/SIB1
DCK	Input	Debug clock input	P98/SOB1
DMS	Input	Debug mode select	P99/$\overline{\text{SCKB1}}$
DDO	Output	Debug data output	P910/(CS301)
DRST	Input	DCU reset input	P911
FLMD0	Input	Flash programming mode setting pin	-
FLMD1			PDL5/AD5
CLKOUT	Output	Internal system clock output	PCM1
PCL	Output	Clock output (for trimming of X1 input clock, subsystem clock)	P913/TIP20/TOP20/INTP4
REGC0	-	Connection of regulator output stabilization capacitance	–
REGC1	-	Connection of regulator output stabilization capacitance	–
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Connection of resonator for main clock	–
X2	–		–
V_{DD0}	–	Positive power supply for internal	–
V_{DD1}	–	Positive power supply for internal	
V_{SS}	–	Ground potential for internal	–
BV_{DD}	–	Positive power supply for bus interface and alternate ports	–
BV_{SS}	–	Ground potential for bus interface and alternate ports	–

2.2 Pin States

The operation states of pins in various mode are described in Table 2-4.

Table 2-4: Pin Operation States in Various Modes

Bus Control Pin	Reset	Halt Mode During DMA Transfer	External Bus Interface Mode	Idle Mode Stop Mode	Idle State **Note 2**	Bus Hold
AD0 to AD15	HI-Z **Note 1**	Operating		Hi-Z	Held	Hi-Z
$\overline{\text{WAIT}}$			-	-	-	-
CLKOUT			Operating	L	Operating	Operating
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$			Operating	-	-	-
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$			H	H	H	Hi-Z
$\overline{\text{RD}}$						
ASTB						
$\overline{\text{HLDAK}}$						L
$\overline{\text{HLDRQ}}$			-	-	-	Operating

Notes: 1. The bus control pin is shared with the port pin, so it is initialized to the input mode (port mode).

2. The state of the pins in the idle state inserted following T3 state is shown.

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 Description of Pin Functions

(1) P00 to P06 (Port 0)

Port 0 is a 7-bit I/O port for which input and output can be specified in 1-bit units.
In addition to I/O port pins, P00 to P06 can also be used as CS30, CSI31 and external interrupt request inputs. The port/control mode can be selected for each bit, and the pin's valid edge for P05 and P06 is specified by the INTR0 and INTF0 registers.
An on-chip pull-up and pull-down resistor can be specified for use in 1-bit units using pull-up/pull-down resistor option register 0 (PU0, PD0).

(a) Port mode

P00 to P06 can be set to input or output in 1-bit units using port mode register 0 (PM0).

(b) Control mode

P00 to P06 can be set to control modes in 1-bit units using port mode control register 0 (PMC0) and port mode function register 0 (PFC0).

- SI31 (serial input) … Input
  This is the serial receive data input pin for CSI31.

- SO31 (serial output) … Output
  This is the serial transmit data output pin for CSI31.

- $\overline{\text{SCK31}}$(serial clock) … Output
  This is the serial clock I/O pin for CSI31.

- CS310 to CS313
  These are the chip select pin for CSI31

- INTP2, INTP3 (interrupt request from peripherals) … Input
  These are external interrupt request input pins.

(2) P10, P11, P12 (Port 1)

Port 1 is 3-bit I/O port for which input and output can be specified in 1-bit units.
In addition to I/O pins, P10 to P12 can also be used as external interrupt request inputs, timer/ counter I/O in the control mode and the external A/D conversion start trigger. The port/control mode can be selected for each bit, and the pin's valid edge for P10 is specified by the INTR1 and INTF1 registers.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register 1 (PU1, PD1).

(a) Port mode

P10 and P12 can be set to input or output in 1-bit units using port mode register 1 (PM1).

(b) Control mode

P10 to P12 can be set to control modes in 1-bit units using port mode control register 3 (PMC1), port mode function register 1 (PFC1) and port mode function expand register 1 (PFCE1)

- INTP0 (interrupt request from peripherals) … Input
  These are external interrupt request input pins.

- TIP30, TIP31 (timer input) … Input
  These are an external capture input pins for TMP3.

- TOP30, TOP31 timer output) … Input
  These are the pulse signal output pins for TMP3.

- ADTRG the external A/D conversion start trigger … Input
  This is the external trigger pins for ADC.

(3) P30 to P38 (Port 3)

Port 3 is a 9-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O port pins, P30 to P38 can also be used as external interrupt request inputs, serial interface I/O, CAN I/F I/O and timer/counter I/O in the control mode. The port/control mode can be selected for each bit, and the pin valid edge for P31 and P37 is specified by the INTR3 and INTF3 registers.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register 3 (PU3, PD3).

(a) Port mode

P30 to P38 can be set to input or output in 1-bit units using port mode register 3 (PM3).

(b) Control mode

P30 to P38 can be set to control modes in 1-bit units using port mode control register 3 (PMC3), port mode function register 3 (PFC3) and port mode function expand register 3 (PFCE3)

- RXDA0 (receive data) … Input
  These are the serial receive transmit input pins for UARTA0.

- TXDA0 (transmit data) … Output
  These are the serial send data output pins for UARTA0.

- $\overline{\text{ASCKA0}}$ (asynchronous serial clock 0) … Input
  This is the serial baud rate input pin for UARTA0.

- INTP1,INTP7 (interrupt request from peripherals) … Input
  This is the external interrupt request input pin.

- TIP00, TIP01 (timer input) … Input
  These are the external count clock input pins for TMP0.

- TOP00, TOP01 (timer output) … Input
  These are the pulse signal output pins for timers TMP0.

- CRXD0, CRXD1 (receive data for controller area network) … Input
  These are the CAN serial receive data input pins for CAN0 and CAN1.

- CTXD0, CTXD1 (transmit data for controller area network) … Output
  These are the CAN serial transmit data output pins for CAN0 and CAN1.

- CS310, CS312
  These are the serial chip select output for CSI31.

(4) P40 to P42 (Port 4) … Bi-directional I/O

Port 4 is a 3-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O port pins, P40 to P42 can also be used as serial interface I/O in the control mode.
The port/control mode can be selected for each bit.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register 4 (PU4, PD4).

(a) Port mode

P40 to P42 can be set to input or output in 1-bit units using port mode register 4 (PM4).

(b) Control mode

P40 to P42 can be set to control modes in 1-bit units using port mode control register 4 (PMC4)

- SIB0 (serial input) … Input
  This is the serial receive data input pin for CSIB0.

- SOB0 (serial output) … Output
  This is the serial transmit data output pin for CSIB0.

- $\overline{\text{SCKB0}}$ (serial clock) … Bi-directional I/O
  This is the serial clock I/O pin for CSIB0.

(5) P50 to P55 (Port 5)

Port 5 is a 6-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O port pins, P50 to P55 can also be used as timer/counter I/O and CSI3 in the control mode. The port/control mode can be selected for each bit.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register 5 (PU5, PD5).

(a) Port mode

P50 to P55 can be set to input or output in 1-bit units using port mode register 5 (PM5).

(b) Control mode

P50 to P55 can be set to control modes in 1-bit units using port mode control register 5 (PMC5) and port mode function register 5 (PFC5).

- TIQ00, TIQ01, TIQ02, TIQ03 (timer input) … Input
  These are the input/capture pins for TMQ0.

- TOQ00, TOQ01, TOQ02, TOQ03 (timer output) … Output
  These are the output/compare pins for TMQ0.

- SI30 (serial input) … Input
  This is the serial receive data input pin for CSI30.

- SO30 (serial output) … Output
  This is the serial transmit data output pin for CSI30.

(6) P70 to P715 (Port 7) … Bi-directional I/O

Port 7 is a 16-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O port pins, P70 to P715 can also be used as analog output pins for the A/D converter in the control mode. However, they cannot be switched between input port and analog output pin.

(a) Port mode

P70 to P715 can be set to input or output in 1-bit units using port mode register 7 (PM7).

(b) Control mode

P70 to P715 can be set to control modes in 1-bit units using port mode control register 7 (PMC7)

- ANI0 to ANI15 (analog inputs 0 to 15) … Input
  These are the analog input pins for the A/D converter.

(7) P90 to P915 (Port 9) … Bi-directional I/O

Port 9 is a 16-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O port pins, P90 to P915 can also be used as serial interface I/O, timer/counter I/O, external interrupt request inputs, clock output, and on chip debug function in the control mode. The port/control mode can be selected for each bit, and the pin's valid edge for P913 to P915 is specified by the INTR9H and INTF9H registers.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register 9 (PU9, PD9).

(a) Port mode

P90 to P915 can be set to input or output in 1-bit units using port mode register 9 (PM9).

(b) Control mode

P90 to P915 can be set to control modes in 1-bit units using port mode control register 9 (PMC9), port mode function register 9 (PFC9) and port mode function expand register 9 (PFCE9).

- SIB1 (serial input)...input
  These are the serial receive data input pins for CSIB1.

- SOB1 (serial output)...output
  These are the serial transmit data input pins for CSIB1.

- $\overline{\text{SCKB1}}$, $\overline{\text{SCK30}}$ (serial clock) … Bi-directional I/O
  This is the serial clock I/O pin for CSIB1 and CSI30

- CS300 to CS303 (Chip select) … Output
  These are the Chip select pins for CSI30

- RXDA1 (receive data) … Input
  This is the serial receive data input pins for UARTA1.

- TXDA1 (transmit data) … Output
  This is the serial transmit data output pin for UARTA1.

- TIP20, TIP21 (timer input) … Input
  These are the external count clock input pins for timer TMP2.

- TOP20, TOP21 (timer output) … Output
  These are the pulse signal output pins for timer TMP2.

- PCL (clock output) … Output
  This is the clock output pin.

- INTP4 to INTP6 (interrupt request from peripherals) … Input
  These are external interrupt request input pins.

- DDI, DCK,DMS, DDO, $\overline{\text{DRST.}}$
  These are signals for the N-Wire interface (JTAG).

(8) PCM0 to PCM3 (Port CM) … Bi-directional I/O

Port CM is a 4-bit I/O port for which input and output can be set in 1-bit units.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register CM (PUCM, PDCM).

(a) Port mode

PCM0 to PCM3 can be set to input or output in 1-bit units using port mode register CM (PMCM).

(b) Control mode

PCM0 to PCM3 can be set to control modes in 1-bit units using port mode control register CM (PMCCM).

- $\overline{\text{HLDAK}}$ (hold acknowledge) … Output
  This is an output pin for the acknowledge signal that indicates the high-impedance status for the address bus, data bus, and control bus when V850E/RS1 receives a bus hold request. The address bus, data bus, and control bus are in high impedance while this signal is active.

- $\overline{\text{HLDRQ}}$ (hold request) … Input
  This is an input pin by which an external device requests the V850E/RS1 to release the address bus, data bus, and control bus release requests. This pin accepts asynchronous input for CLKOUT. When this pin is made active, V850E/RS1 sets the address bus, data bus, and control bus to high impedance upon the end of the bus cycle currently being executed, or immediately if no bus cycle is being executed, and the HLDAK signal is then made active and the bus is released.

- CLKOUT (clock out) … Output
  This pin outputs internally generated bus clocks.

- $\overline{\text{WAIT}}$ (wait) … Input
  This is the input pin for the control signal used to insert waits into the bus cycle. This pin is sampled at the falling edge of the clock during the T2 or TW state of the bus cycle. ON/OFF switching of the wait function is performed with port mode control register CM (PMCCM).

(9) PCS0, PCS1 (Port CS) … Bi-directional I/O

Port CS is a 2-bit I/O port for which input and output can be set in 1-bit units.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register CS (PUCS, PDCS).

(a) Port mode

PCS0 and PCS1 can be set to input or output in 1-bit units using port mode register CS (PMCS).

(b) Control mode

PCS0 and PCS1 can be set to control modes in 1-bit units using port mode control register CS (PMCCS).

- $\overline{\text{CS0}}$, $\overline{\text{CS1}}$ (chip select input) … Input
  This is the chip select signal for external SRAM, external ROM, or external peripheral I/O.

  The signal $\overline{\text{CSn}}$ is assigned to memory block n (n = 0, 1).
  This is active for the period during which a bus cycle that accesses the corresponding memory block is activated. It is inactive in an idle state (TI).

(10) PCT0, PCT1, PCT4, PCT6 (Port CT) … Bi-directional I/O

Port CT is a 4-bit I/O port for which input and output can be set in 1-bit units.
In addition to I/O pins, PCT0, PCT1, PCT4, PCT6 can also be used as control signal output pins for external memory expansion in the control mode.
The port/control mode can be selected for each bit.
An on-chip pull-up and Pull-down resistor can be specified for use in 1-bit units using pull-up/ pull-down resistor option register CT (PUCT, PDCT).

(a) Port mode

PCT0, PCT1, PCT4, PCT6 can be set to input or output in 1-bit units using port CT mode register (PMCT).

(b) Control mode

PCT0, PCT1, PCT4, PCT6 can be set to control modes in 1-bit units using port mode control register CT (PMCCT).

- $\overline{\text{WR0}}$ (write strobe low-level data) … Output
  This is the write strobe signal output pin for the lower data of the external 16-bit data bus.

- $\overline{\text{WR1}}$ (write strobe high-level data) … Output
  This is the write strobe signal output pin for the higher data of the external 16-bit data bus.

- $\overline{\text{RD}}$ (read strobe) … Output
  This is the read strobe signal output pin for the external 16-bit data bus.

- ASTB (address strobe) … Output
  This is the output pin for the latch strobe signal for the external address bus. Output becomes low level in synchronization with the falling edge of the clock during the T1 state of the bus cycle, and becomes high level in synchronization with the falling edge of the clock during the T3 state of the bus cycle. Output becomes high level when the bus cycle is inactive.

(11) PDL0 to PDL13 (port DL) … Bi-directional I/O

Port DL is a 14-bit I/O port for which input and output can be set in 1-bit units.
During flash memory programming (corresponding to input of high level to FLMD0), PDL5 functions as the FLMD1 pin. At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL13 can be set to input or output in 1-bit units with the port DL mode register (PMDL).

(b) Control mode

PDL0 to PDL13 can be set to control modes in 1-bit units using port mode control register DL (PMCDL).

- AD0 to AD13 (address/data bus 0 to 15) … Bi-directional I/O
  These form an address/data multiplexed bus during external access. In the multiplexed bus mode, they function as address output or data I/O, and in the separate bus mode, they function as data I/O.

(12) FLMD0(flash programming mode) Input

This is the input pin that specifies the operation mode. Fix this pin to low level so that its level does not change during operation.
The internal circuit samples the status of the FLMD0 pin at the rising edge of the $\overline{\text{RESET}}$ pin.
In the normal operation mode, connect this pin to V_{SS}.

(13) $\overline{\text{RESET}}$ (reset) ... Input

$\overline{\text{RESET}}$ is a signal that is input asynchronously and has a low level width regardless of the status of the operating clock. When this signal is input, system reset is executed with higher priority than all other operations.
In addition to being used for ordinary initializations/start operations, this signal can also be used to cancel a standby mode (STOP).

(14) X1, X2 (crystal for main clock)

These pins are used to connect the resonator that generates the system clock.

(15) AV_{SS} (analog V_{SS})

This is the ground pin for the A/D converter and alternate ports.

(16) AV_{REF0} (analog reference voltage) ... Input

This is the reference voltage supply pin for the A/D converter.

(17) V_{DD0}, V_{DD1} (power supply)

This is the positive power supply pin. Connect all the V_{DD} pins to a positive power supply.

(18) V_{SS0}, V_{SS1} (ground)

This is a ground pin. Connect all the V_{SS} pins to ground.

(19) FLMD1 (flash programming mode)

This is a control pin for the flash memory programming mode when connected to the dedicated flash programming tool. Always input a low signal to this pin during flash programming.

(20) BV_{DD} (power supply for I/O)

This is the positive power supply pin for I/O.

(21) BV_{SS} (ground for I/O)

This is the ground pin for the for I/O.

2.4 Pin I/O Circuit Types, I/O Buffer Power Supply and Handling of Unused Pins

Table 2-5: Pin I/O Circuit Types (1/3)

Pin Name	I/O Circuit Types	Recommended Connection
P00/SI31	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P01/SO31		
P02/$\overline{\text{SCK31}}$		
P03/CS310		
P04/CS311		
P05/CS312/INTP2		
P06/CS313/INTP3		
P10/INTP0	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P11/TIP31/TOP31		
P12/TIP30/TOP30/ADTRG		
P30/TXDA0/[$\overline{\text{ASCKA0}}$]	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P31/RXDA0/INTP7		
P32/$\overline{\text{ASCKA0}}$/[TXDA0]/ (CS310)		
P33/CTXD0		
P34/CRXD0		
P35/CRXD1		
P36/CTXD1		
P37/TIP00/TOP00/INTP1/ (CS312)		
P38/TIP01/TOP01		
P40/SIB0	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P41/SOB0		
P42/$\overline{\text{SCKB0}}$		
P50/TIQ01/TOQ01	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P51/TIQ02/TOQ02		
P52/TIQ03/TOQ03		
P53/TIQ00/TOQ00		
P54/SI30		
P55/SO30		
P70 to P715 /ANI0 to ANI15	11-G	Connect it to AV_{REF0} or AV_{SS0} through independent resistor

Table 2-5: Pin I/O Circuit Types (2/3)

Pin Name	I/O Circuit Types	Recommended Connection
P90/TIQ11/TOQ11/$\overline{\text{SCK30}}$	5-AM	Connect it to V_{DD1} or V_{SS1} through independent resistor.
P91/TIQ12/TOQ12/CS300	5-AM	
P92/TIQ13/TOQ13/CS301	5-AM	
P93/TIQ10/TOQ10/CS302	5-AM	
P94/$\overline{\text{ASCKA1}}$/CS303	5-AM	
P95/RXDA1/($\overline{\text{SCK30}}$)	5-AM	
P96/TXDA1/(CS300)	5-AM	
P97/SIB1/{DDI}	5-AM	
P98/SOB1/{DCK}	5-AM	
P99/$\overline{\text{SCKB1}}$/{DMS}	5-AM	Connect it to BV_{DD} or BV_{SS} through independent resistor.
P910/CS301/{DDO}	5-AE	
P911/{$\overline{\text{DRST}}$}	5-AM	
P912//TIP21/TOP21/CS302	5-AM	
P913/TIP20/TOP20/INTP4/ PCL	5-AM	
P914/TIP10/TOP10/INTP5/ AD14	5-AM	Connect it to BV_{DD} or BV_{SS} through independent resistor.
P915/TIP11/TOP11/INTP6/ AD15	5-AM	
PCM0/$\overline{\text{WAIT}}$	5-AM	Connect it to BV_{DD} or BV_{SS} through independent resistor.
PCM1/CLKOUT		
PCM2/$\overline{\text{HLDAK}}$		
PCM3/$\overline{\text{HLDRQ}}$		
PCS0/$\overline{\text{CS0}}$	5-AM	Connect it to BV_{DD} or BV_{SS} through independent resistor.
PCS1/$\overline{\text{CS1}}$		
PCT0/$\overline{\text{WR0}}$	5-AM	Connect it to BV_{DD} or BV_{SS} through independent resistor.
PCT1/$\overline{\text{WR1}}$		
PCT4/$\overline{\text{RD}}$		
PCT6/ASTB		

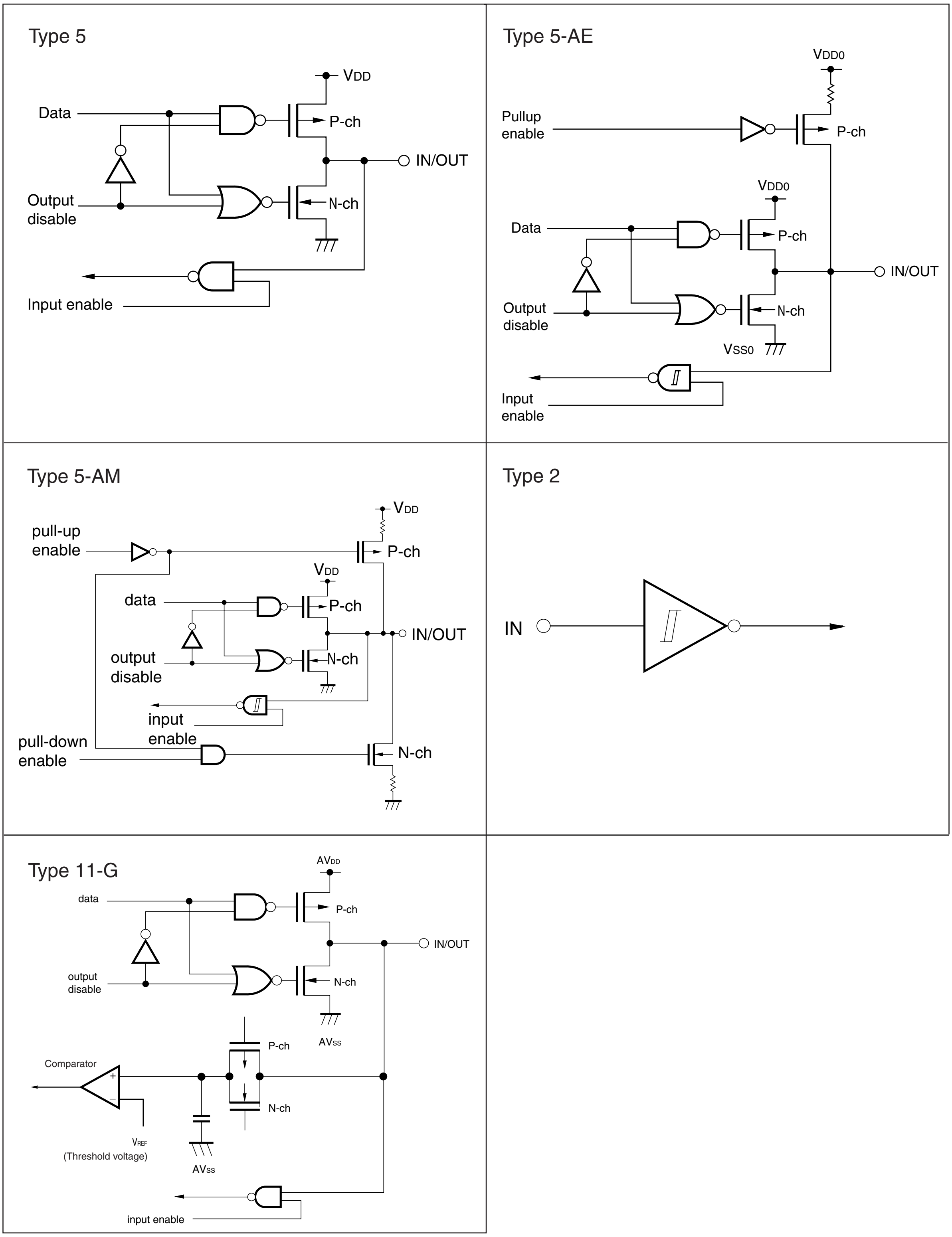
Table 2-5: Pin I/O Circuit Types (3/3)

Pin Name	I/O Circuit Types	Recommended Connection
PDL0/AD0	5	Connect it to BV_{DD} or BV_{SS} through independent resistor.
PDL1/AD1		
PDL2/AD2		
PDL3/AD3		
PDL4/AD4		
PDL5/AD5/FLMD1		
PDL6/AD6		
PDL7/AD7		
PDL8/AD8		
PDL9/AD9		
PDL10/AD10		
PDL11/AD11		
PDL12/AD12		
PDL13/AD13		
AV_{REF0}	-	Analog voltage reference for A/D converter.
AV_{SS}	-	Analog Ground.
FLMD0	-	See 21.6.1 ”FLMD0 pin” on page 756.
$\overline{\text{RESET}}$	2	External system reset input.
V_{DD0}	-	Power supply pin for PLL.
V_{SS0}	-	Ground potential pin for PLL.
X1	-	Connection of resonator for main clock.
X2	-	
V_{DD1}	-	Power supply pin for Flash, CPU and I/O buffers.
V_{SS1}	-	Ground potential pin for Flash, CPU and I/O buffers.
BV_{DD}	-	Power supply pin for I/O buffers.
BV_{SS}	-	Ground potential pin for I/O buffers.
REGC0	-	Connect to V_{SS} via a capacitor. **Note 1**
REGC1	-	Connect to V_{SS} via a capacitor. **Note 2**
NMI	-	NMI

Notes: 1. NEC specifies to connect 1µF to REGC0.

2. NEC specifies to connect 4.7 µF to REGC1.

Figure 2-1: Pin I/O Circuits



Chapter 3 CPU Function

The CPU of the V850E/RS1, which is based on RISC architecture, executes almost all instructions in one clock cycle due to its five-stage pipeline control.

3.1 Features

- Number of instructions: 81
- Minimum instruction execution time: 41.6 ns @ 24 MHz, 31.25 ns @ 32 MHz,
25 ns @ 40 MHz only for µPD70F3403 and µPD70F3403A
- Memory space
- Program area: 64 MB linear address space
- Data area: 4 GB linear address space
- General-purpose registers: 32 bits × 32 registers
- Internal 32-bit architecture
- 5-stage pipeline control
- Hardware interlock on register/flag hazards
- Instruction set
    - Upwardly compatible with V850 CPU
    - Multiplication/division instruction
    - Saturated calculation instructions (with overflow/underflow detection function)
    - 32-bit shift instructions: 1 clock
    - Non blocking Load/store instruction with long/short format
    - Multiplication can be performed in 1 or 2 clocks due to on-chip hardware multiplier:
        - 16 bits × 16 bits → 32 bits
        - 32 bits × 32 bits → 32 bits or 64 bits in one clock cycle.
    - Four types of bit manipulation instructions:
        - Set
        - Clear
        - Not
        - Test

3.2 CPU Register Set

The CPU registers of the V850E/RS1 can be classified into general purpose register set, which are used by programs, and system register set, which are used to control the execution environment. This chapter describe also specific registers which can be read or written using the LDSR and STSR instructions. All the registers have 32-bit width.

For details, refer to V850E1 User's Manual Architecture.
(Document No. U14559EJ2V0UM00 (2nd edition))

Figure 3-1: CPU Register Set

(1) Program register set

31 0

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

31 0

PC	(Program counter)

(2) System register set

31 0

EIPC	(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

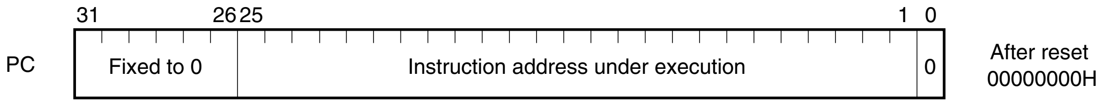
Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer	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

(2) Program counter (PC)

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

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

Figure 3-2: Program Counter (PC) Format



3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.
Read from and write to system registers are performed by setting the system register numbers shown below with the system register load/store instructions (LDSR, STSR instructions).

Table 3-2: System Register Numbers

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)	√	√
3	NMI status saving register (FEPSW)	√	√
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)	√ Note 2	√
17	CALLT execution status saving register (CTPSW)	√ Note 2	√
18	Exception/debug trap status saving register (DBPC)	√ Note 3	√
19	Exception/debug trap status saving register (DBPSW)	√ Note 3	√
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×

Notes: 1. Since only one set of these registers is available, the program must save the contents of these registers when multiple interrupt servicing is permitted.

2. Since only one set of these registers is available, the program must save the contents of these registers when CALLT instructions nesting are used.

3. These registers can be accessed only when the DBTRAP instruction is executed.

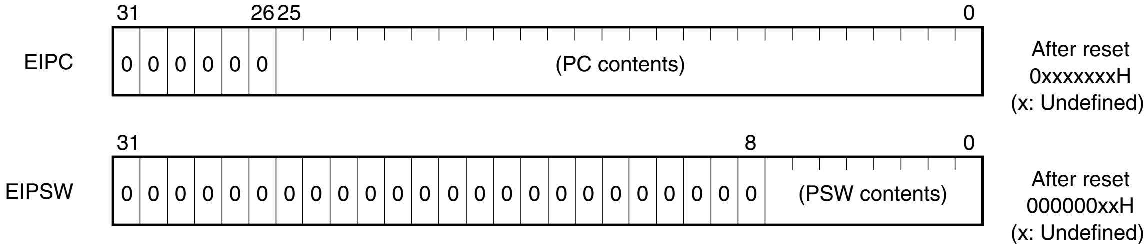
Caution: Even if EIPC or FEPC, or bit 0 of CTPC is set to 1 by the LDSR instruction, bit 0 is ignored when execution is returned to the main routine by the RETI instruction after interrupt servicing (this is because bit 0 of the PC is fixed to 0). Set an even value to EIPC, FEPC, and CTPC (bit 0 = 0).

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 one of the instruction under execution, except some instructions, 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).

Figure 3-3: Interrupt Status Saving Registers (EIPC and EIPSW) Format

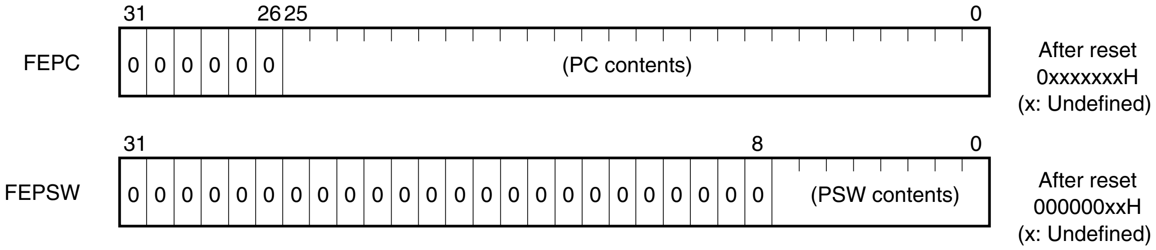


The values of EIPC and EIPSW are restored to PC and PSW during execution of a RETI instruction.

(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).

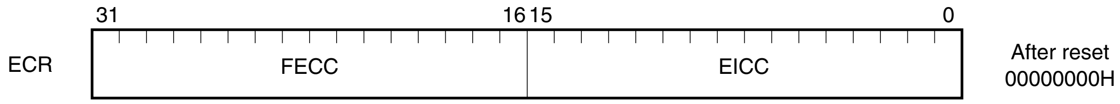
Figure 3-4: NMI Status Saving Registers (FEPC and FEPSW) Format



(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.

Figure 3-5: Interrupt Source Register (ECR) Format



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

The list of exception codes is tabulated in **Table 17-1, “Interrupt/Exception Source List,” on page 685**.

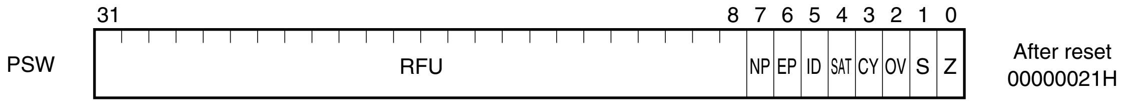
(4) Program status word (PSW)

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

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of the LDSR instruction execution. However, if the ID flag is set to 1, interrupt request acknowledgement during LDSR instruction execution is prohibited.

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

Figure 3-6: Program Status Word (PSW) Format (1/2)



Bit position	Flag name	Meaning
31 to 8	RFU	Reserved field. Fixed to 0.
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.

Note: During saturated operation, the saturated operation results are determined by the contents of the OV flag and S flag. The SAT flag is set to 1 only when the OV flag is set to 1 during saturated operation. This is explained on the following table.

Figure 3-6: Program Status Word (PSW) Format (2/2)

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.

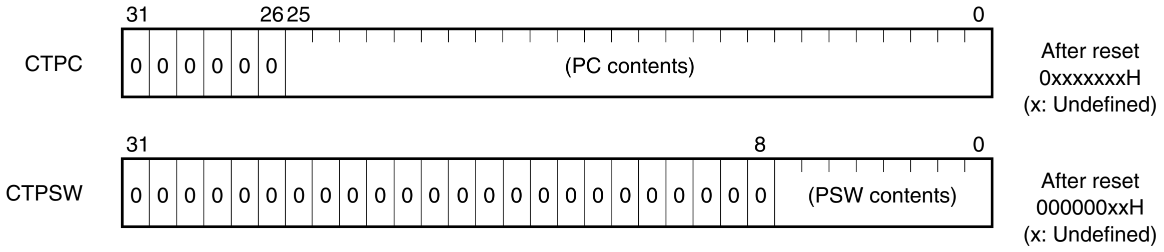
When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and those of the program status word (PSW) are saved to CTPSW.

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 (fixed to 0) for future function expansion.

Figure 3-7: CALLT Execution Status Saving Registers (CTPC and CTPSW) Format



The values of CTPC and CTPSW are restored to PC and PSW during execution of the CTRET instruction.

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

DBPC and DBPSW are exception/debug trap status registers.

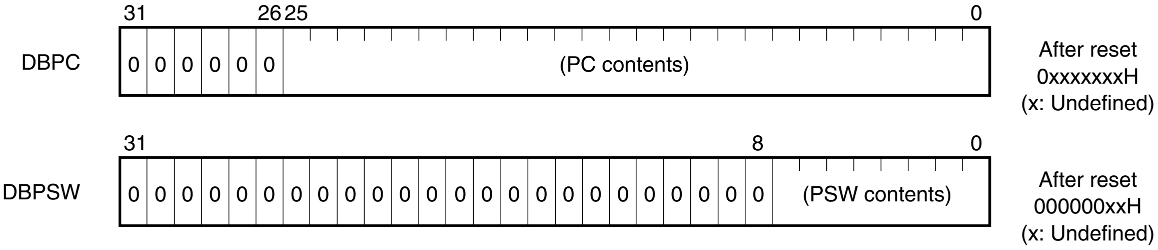
If an exception trap or debug trap occurs, the contents of the program counter (PC) are saved to DBPC, and those of the program status word (PSW) are saved to DBPSW.

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.

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

Figure 3-8: Exception/Debug Trap Status Saving Registers (DBPC and DBPSW) Format



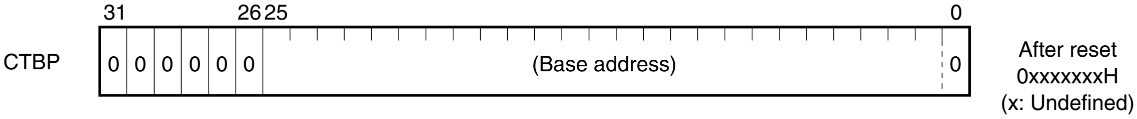
The values of DBPC and DBPSW are restored to PC and PSW during execution of the DBRET instruction.

(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 (fixed to 0) for future function expansion.

Figure 3-9: CALLT Base Pointer (CTBP) Format



3.2.3 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up.
The V850E/RS1 has the following special registers.

- Power save control register (PSC)
- Clock control register (CKC)
- Main peripheral clock control register (MPCCTL)
- Processor clock control register (PCC)
- Clock monitor mode register (CLM)
- Reset control flag register (RESF)
- Low-voltage detection register (LVIM)
- Internal RAM data status register (RAMS)
- On-chip debug mode setting register (OCDM)

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

(1) Setting data to special registers

Set data to the special registers in the following sequence.

<1> 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 command register (PRCMD).
<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).
<10> Enable DMA operation if necessary.

[Example] With PSC register

```
       ST.B   r11, PSMR[r0]      ; Set PSMR register.
<1>    CLR1   0, DCHCn[r0]       ; Disable DMA operation. n = 0 to 5
<2>    MOV    0x02, r10
<3>    ST.B   r10, PRCMD[r0]     ; Write PRCMD register.
<4>    ST.B   r10, PSC[r0]       ; Set PSC register.
<5>    NOP;   Dummy instruction
<6>    NOP;   Dummy instruction
<7>    NOP;   Dummy instruction
<8>    NOP;   Dummy instruction
<9>    NOP;   Dummy instruction
<10>   SET1   0, DCHCn[r0]; Enable DMA operation. n = 0 to 5
(next instruction)
```

There is no special sequence to read a special register.

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 and an error to be stored in the PERR bit of the SYS register.

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.

3. Five NOP instructions or more must be inserted immediately after setting the IDLE mode or software STOP mode (by setting the STP bit of the PSC register to 1).

(2) Command register (PRCMD)

The command register (PRCMD) is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program hang-up. The first write access to a special register (power save control register (PSC)) 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.

PRCMD register must be written with store instruction execution by CPU only (not with DMA transfer). If an illegal store operation to a special register takes place, it can be checked by the PERR flag of the system status register (SYS).

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

Figure 3-10: Command Register (PRCMD) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PRCMD	8 bits Registration Code								FFFFF1FCH	undefined
R/W	W	W	W	W	W	W	W	W		

Remark: Registration Code is any 8-bit data.

Caution: PRCMD, PSC, MPCCTL, CKC, PCC, CLM, RESF, LVIM, RAMS, OCDM registers must be written with store instruction execution by CPU only. If an illegal store operation to a special register takes place, it can be checked by the PERR flag of the system status register (SYS).

(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.

Figure 3-11: System Status Register (SYS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
SYS	0	0	0	0	0	0	0	PRERR	FFFFFB02H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PRERR	Detects protection error
0	Protection error did not occur
1	Protection error occurred

Operation conditions of PERR flag.

(a) Set condition (PRERR = 1)

- When data is written to a special register without writing anything to the PRCMD register (when <4> is executed without executing <3> in 3.2.3 (1)"Setting data to special registers" on page 71
- When data is written to a 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 3.4.8 (1) Setting special register is not the setting of a special register)

(b) Clear condition (PRERR = 0)

- When 0 is written to the PRERR flag of the SYS register
- 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.

3.3 Operation Modes

3.3.1 Operation modes

The V850E/RS1 has the following operation modes.

(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. By setting the PMCDH, PMCDL, PMCCM, PMCCS, and PMCCT registers to the control mode using instructions, an external device can be connected to the external memory area.

(2) Flash memory programming mode

In this mode, the internal flash memory can be programmed by using a flash programmer.
For details refer to **Chapter 21 ”Flash Memory” on page 749**.

(3) On-chip debug mode

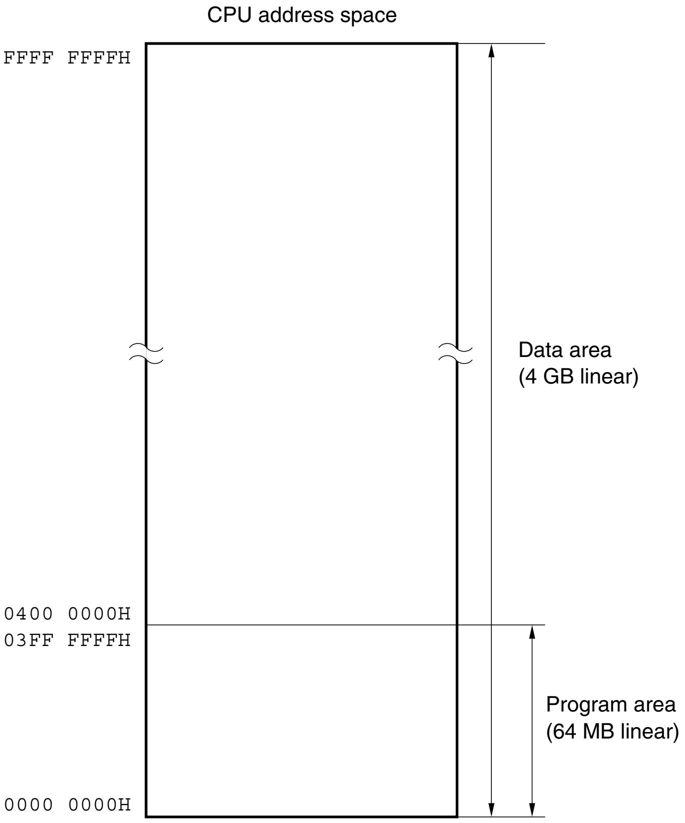
With on-chip debug mode, the user program is able to be controlled using the microcomputer that is mounted on the target hardware. Refer to **Chapter 22 ”On-Chip Debug Function” on page 767** for details of operation.

3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/RS1 has 32-bit architecture and supports up to 4 GB of linear address space (data space) for operand addressing (data access). It also supports up to 64 MB of linear address space (program space) for instruction addressing. Note, however, that both the program and data spaces have areas that are prohibited from being used. For details, refer to **Figure 3-13, "Image on Address Space," on page 76**.
Figure 3-12 shows the CPU address space.

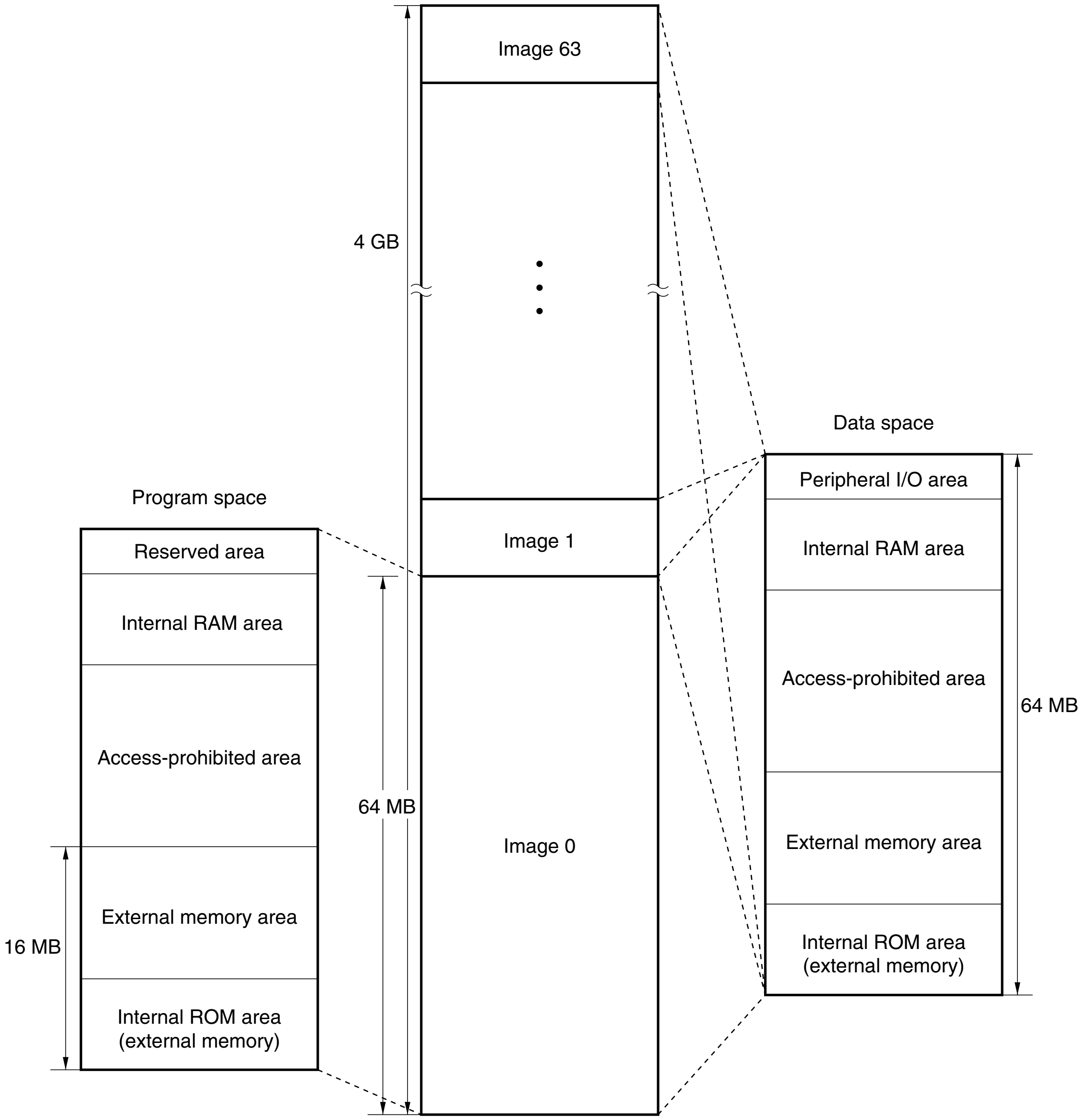
Figure 3-12: CPU Address Space



3.4.2 Image

For instruction addressing, up to 16 MB of linear address space (program space) and an internal RAM area are supported. Up to 4 GB of linear address space (data space) is supported for operand addressing (data access). In the 4 GB address space, it seems that there are sixty-four 64 MB physical address spaces. This means that the same 64 MB physical address space is accessed, regardless of the values of bits 31 to 26.

Figure 3-13: Image on Address Space



3.4.3 Wrap-around of CPU address space

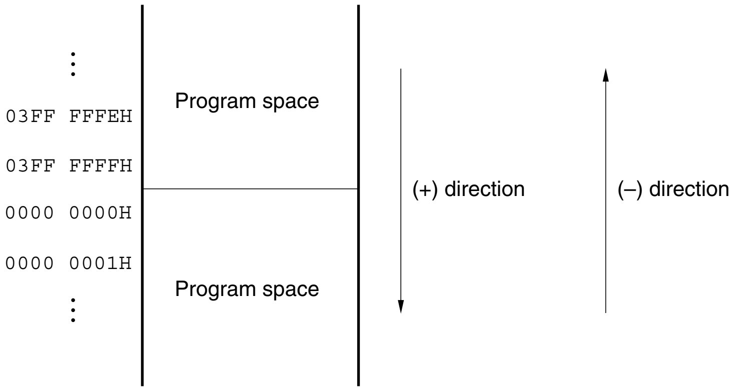
(1) Program space

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

Therefore, the lowest address of the program space, 00000000H, and the highest address, 03FFFFFFH, are contiguous addresses. That the lowest address and the highest address of the program space are contiguous in this way is called wrap-around.

Caution: Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is a peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.

Figure 3-14: Program Space

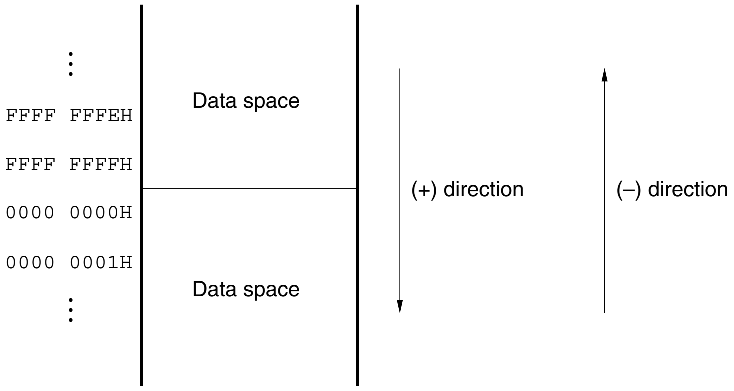


(2) Data space

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

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

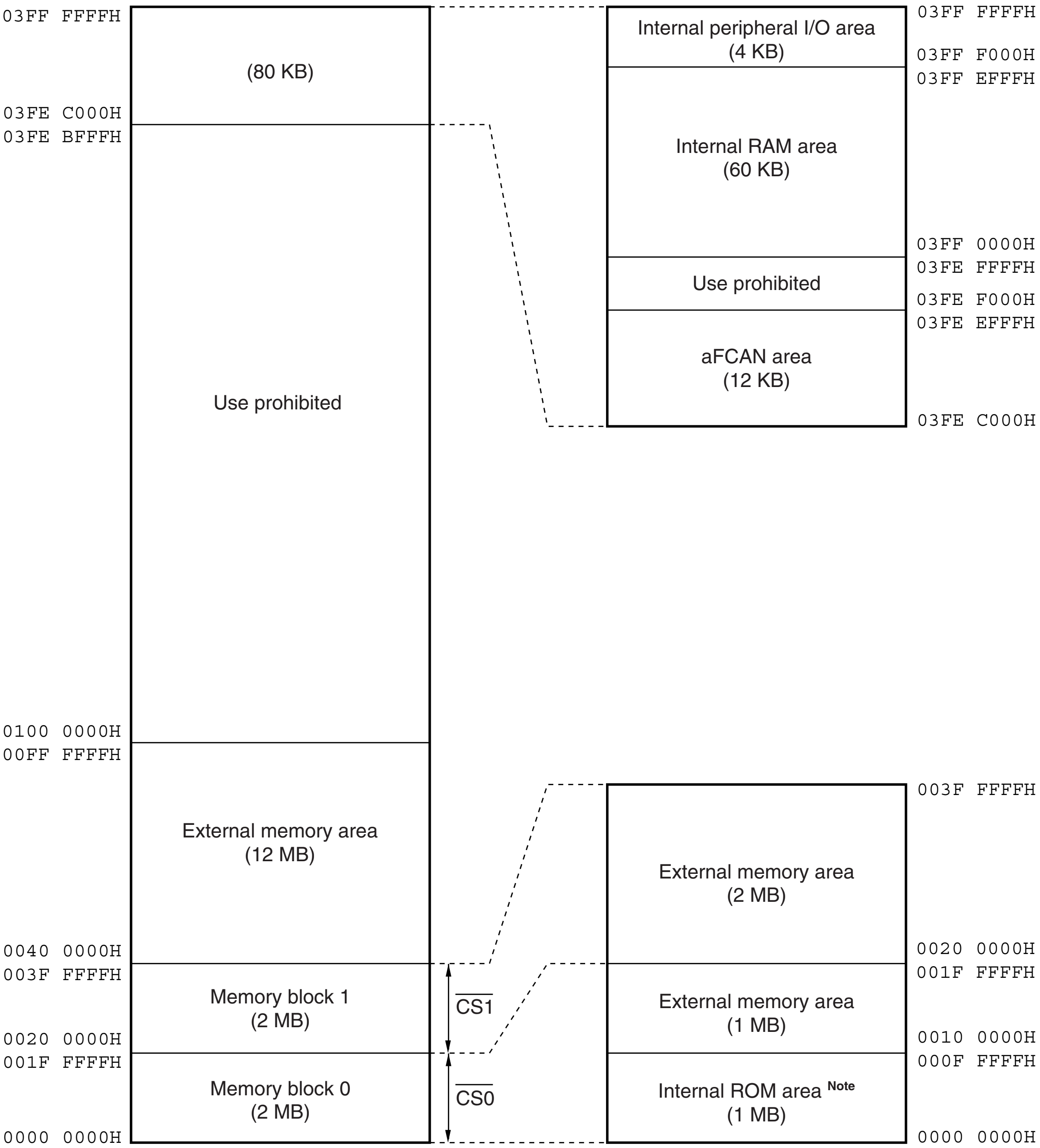
Figure 3-15: Data Space



3.4.4 Memory map

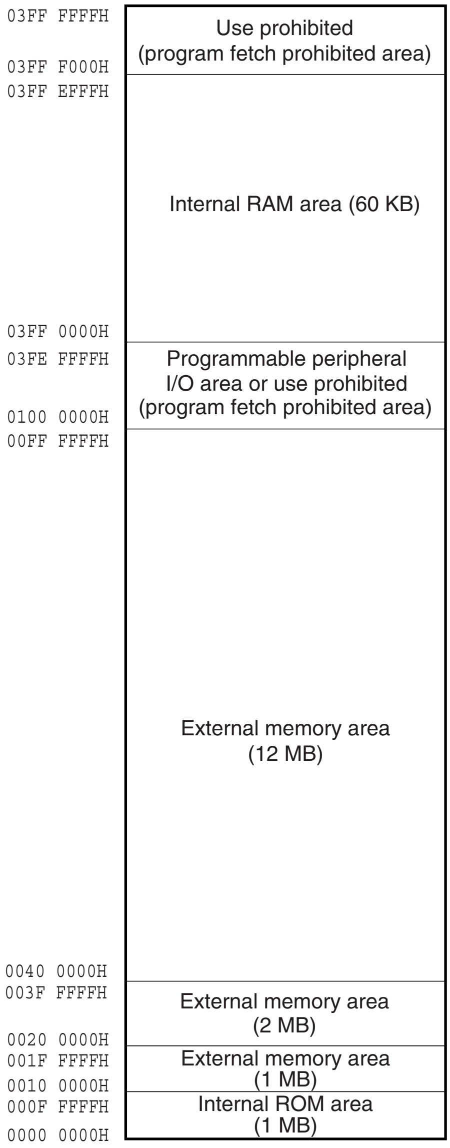
The areas shown in Figure 3-16 are reserved in the V850E/RS1.

Figure 3-16: Data Memory Map (Physical Addresses)



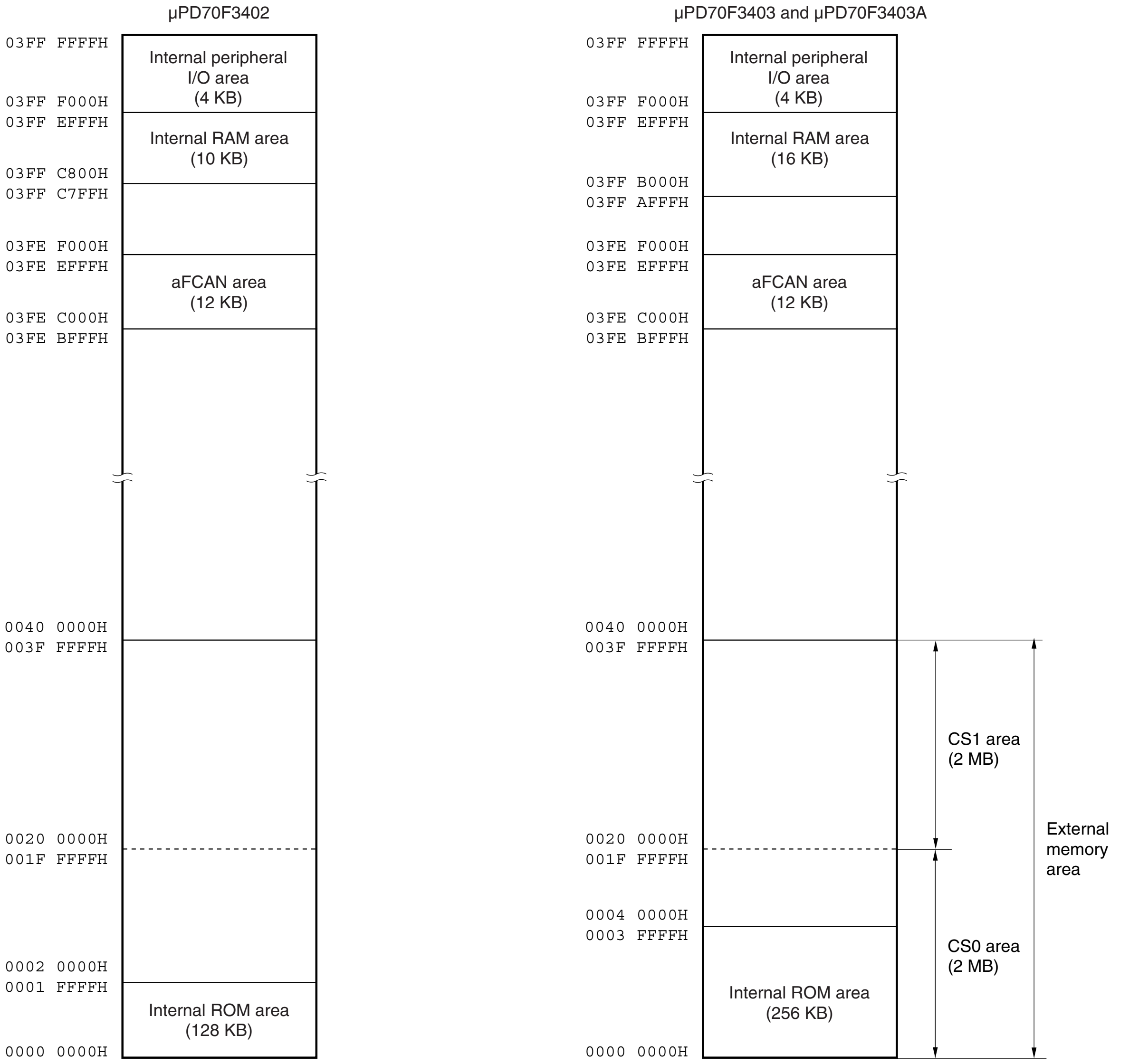
Note: Fetch access and read access to addresses 0000000H to 00FFFFFH is made to the internal ROM area. However, data write access to these addresses is made to the external memory area.

Figure 3-17: Program Memory Map



Remark: Instructions can be executed to the external memory area without execution branching from the internal ROM area to the external memory area.

Figure 3-18: Memory Map Area for μPD70F3402, μPD70F3403 and μPD70F3403A



3.4.5 Memory areas

(1) Internal ROM area

(a) Memory map

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

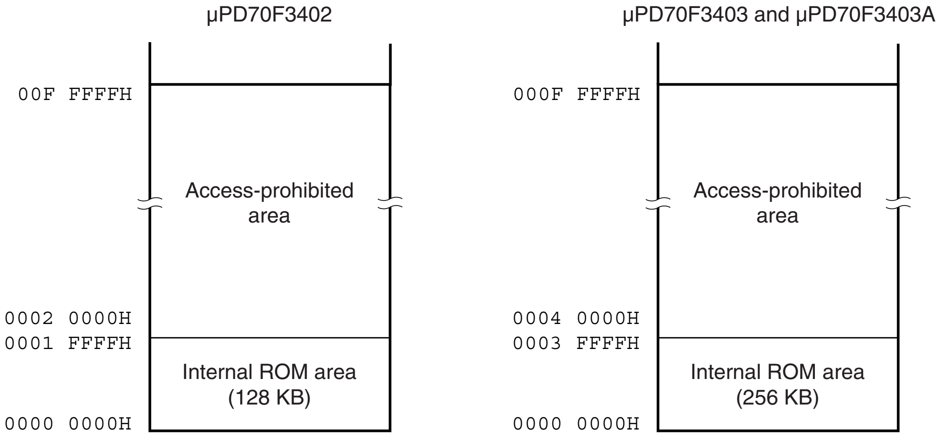
- **µPD70F3402**

128 KB are provided in the following addresses as physical internal ROM (flash memory).
In single-chip mode: Addresses 000000H to 01FFFFH

- **µPD70F3403 and µPD70F3403A**

256 KB are provided in the following addresses as physical internal ROM (flash memory).
In single-chip mode: Addresses 000000H to 03FFFFH

Figure 3-19: Internal ROM Area



(b) Interrupt/exception table

The V850E/RS1 speeds up the interrupt response time by fixing handler addresses corresponding to interrupts/exceptions.
A collection of these handler addresses is called an interrupt/exception table, which is mapped to the internal ROM area. When an interrupt/exception is acknowledged, execution jumps to a handler address and the program in the area starting from that address is executed.
Refer to **Chapter 17 ”Interrupt/Exception Processing Function” on page 685** for a description of the interrupt/exception sources and corresponding addresses.

(2) Internal RAM area

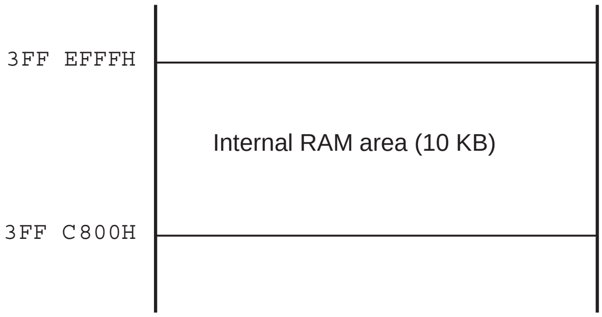
60 KB of addresses 3FF0000H to 3FFEFFFH are reserved as the internal RAM area.

(a) µPD70F3402

10 KB are provided in the following addresses as physical internal RAM.

- Addresses 3FFC800H to 3FFEFFFH

Figure 3-20: Internal RAM Area (10 KB)

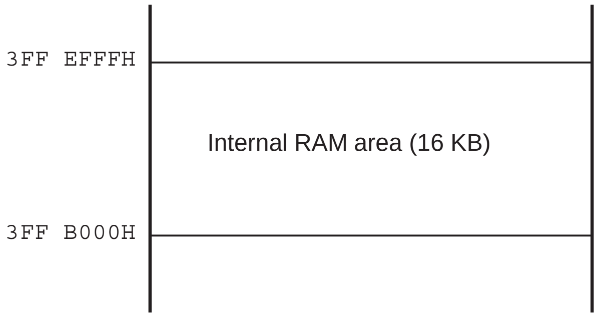


(b) µPD70F3403 and µPD70F3403A

16 KB are provided in the following addresses as physical internal RAM.

- Addresses 3FFB000H to 3FFEFFFH

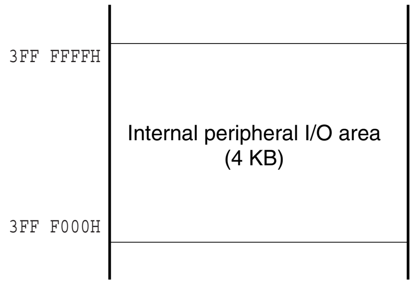
Figure 3-21: Internal RAM Area (16 KB)



(3) Internal peripheral I/O area

4 KB of addresses 3FFF000H to 3FFFFFFH are allocated as the internal peripheral I/O area.

Figure 3-22: Internal Peripheral I/O Area



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

Cautions: 1. When a register is accessed in word units, a word area is accessed twice in half-word units in the order of lower area and higher area, with the lower 2 bits of the address ignored.

2. If a register that can be accessed in byte units is accessed in half word units, the higher 8 bits are undefined when the register is read, and data is written to the lower 8 bits.

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

3.4.6 Recommended use of address space

The architecture of the V850E/RS1 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

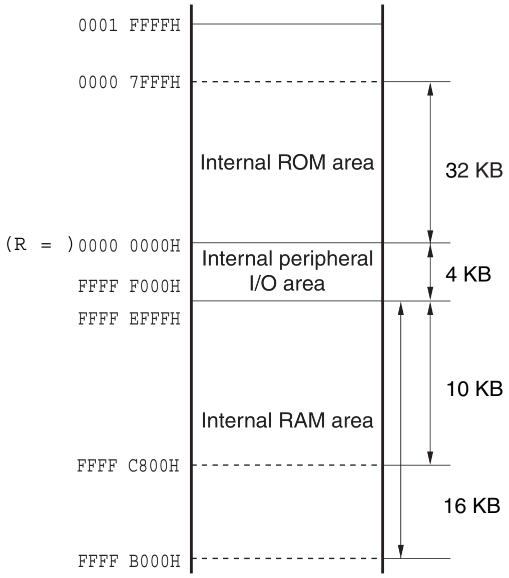
Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Regarding the program space, therefore, a 64 MB space of contiguous addresses starting from 00000000H unconditionally corresponds to the memory map.
To use the internal RAM area as the program space, access addresses 3FFC000H to 3FFEFFFH.

(2) Data space

With the V850E/RS1, 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.

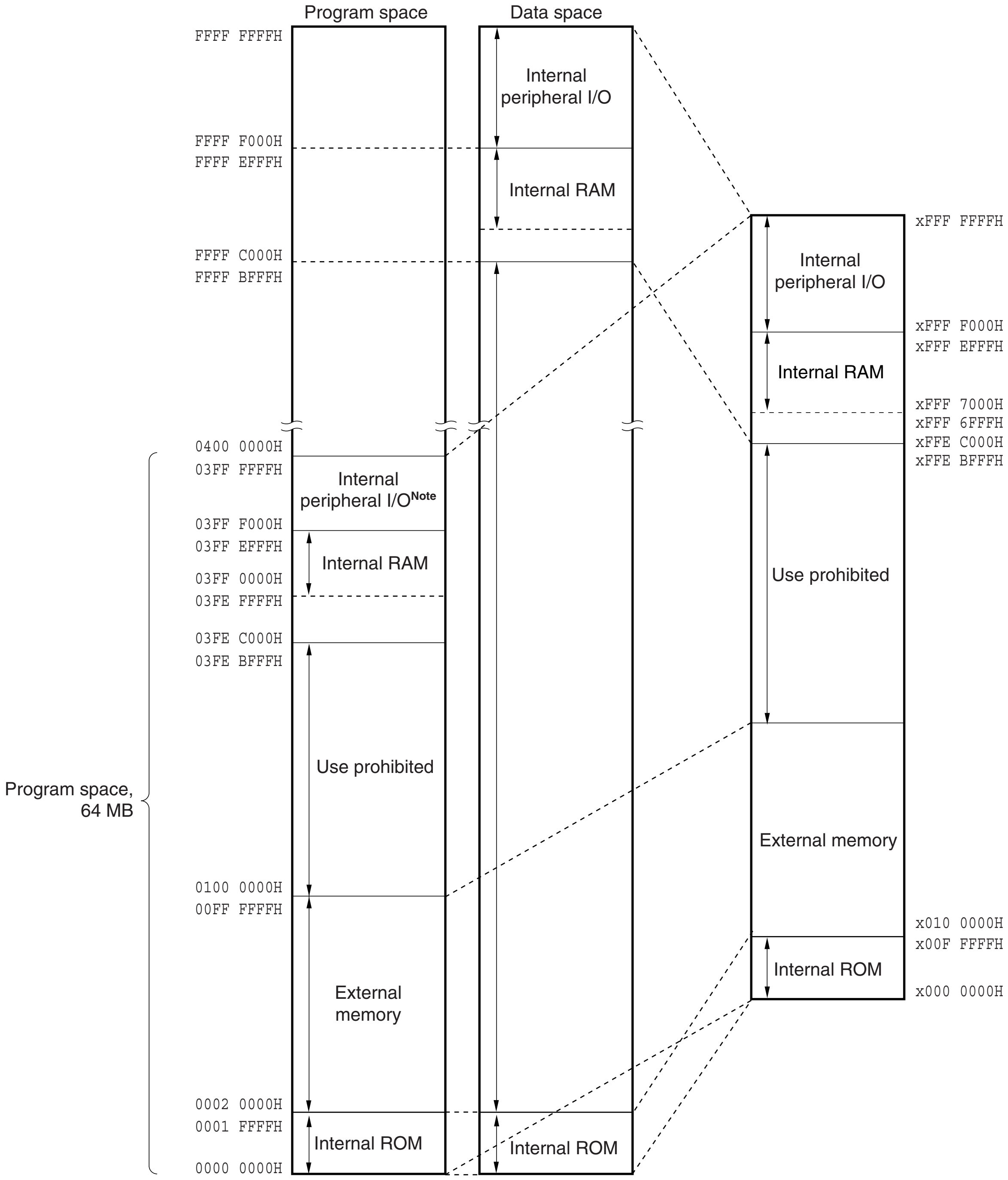
Example: An application example of wrap-around is shown below (μPD70F3402 / μPD70F3403 / μPD70F3403A).

Figure 3-23: Wrap-Around Using μPD70F3402 / μPD70F3403 / μPD70F3403A



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 3-24: Recommended Memory Map



Note: Access to this area is prohibited. To access the internal I/O in this area, specify addresses FFFF000H to FFFFFFFH.

Remark: indicates the recommended area.

3.4.7 Cautions

(1) Registers to be set first

Be sure to set the following registers first when using the V850E/RS1.

- System wait control register (VSWC)
- On-chip Debug Mode control 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.

(2) Area access time (VSWC) and On chip debug mode control register (OCDM)

(a) System wait control register (VSWC)

The system wait control register (VSWC) controls wait of bus access to the internal peripheral I/O registers.
Three clocks are required to access an internal peripheral I/O register (without a wait cycle). The V850E/RS1 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.

Figure 3-25: System Wait Control Register (VSWC) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
VSWC	0	SUWL2	SUWL1	SUWL0	0	VSWL2	VSWL1	VSWL0	FFFFF06EH	77H
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W		

Operation Frequency (f_{XX})	VSWC setting
4 MHz ≤ ϕ ≤ 25 MHz	11H
25 MHz ≤ ϕ ≤ 33 MHz	12H
33 MHz ≤ ϕ ≤ 40 MHz	14H

Caution: Only select a value that corresponds to a supported operating frequency.

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

This register is used to switch between the normal operation mode and the on-chip debug mode. The OCDM register is a special register (refer to 3.2.3 ”Special registers” on page 70). Writing is possible only using a specific sequence so that its contents cannot be rewritten by mistake in case of inadvertent program loops.

When OCDM0 bit is set to 1 and $\overline{\text{DRST}}$ pin input is high level, the on-chip debug mode is selected. Since after reset the initial value of OCDM0 bit is 1, when not using the on-chip debug function, it is necessary to clear the OCDM0 bit and maintain the $\overline{\text{DRST}}$ pin at low level until the OCDM0 bit is cleared.

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

Figure 3-26: On-Chip Debug Mode Register (OCDM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCDM	0	0	0	0	0	0	0	OCDM0	FFFFF9FCH	01H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

OCDM0	Operation mode
0	Normal operation mode
1	$\overline{\text{DRST}}$ pin = low level: Normal operation mode $\overline{\text{DRST}}$ pin = high level: On-chip debug mode

Notes: 1. On input to RESET pin (external reset): OCDM0 = 1
On power-on reset: OCDM0 = 0
On occurrence of internal reset (other than power-on clear): OCDM register holds the value before occurrence of reset.

2. P97/SIB1/{DDI}
P98/$\overline{\text{SOB1}}$/{DCK}
P99/$\overline{\text{SCKB1}}$/{DMS}
P910/$\overline{\text{CS301}}$/{DDO}
P911/{$\overline{\text{DRST}}$}

Caution: A pull-down resistor function is also associated with the OCDM register setting. When OCDM0=1, the port pull-down resistor is enabled, likewise when OCDM0=0 the pull-down is disabled. Refer to Chapter 4 ”Port Functions” on page 105 for more information.

Figure 3-27: Timing Chart When On-Chip Debug Function Is Not Used

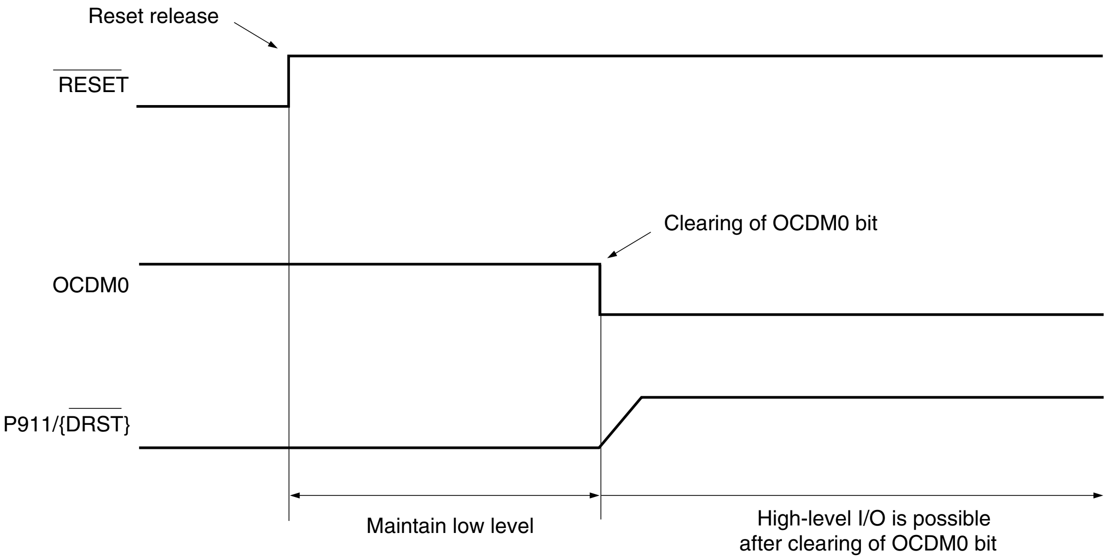


Figure 3-28: Timing Chart of Transition to Normal Operation Mode

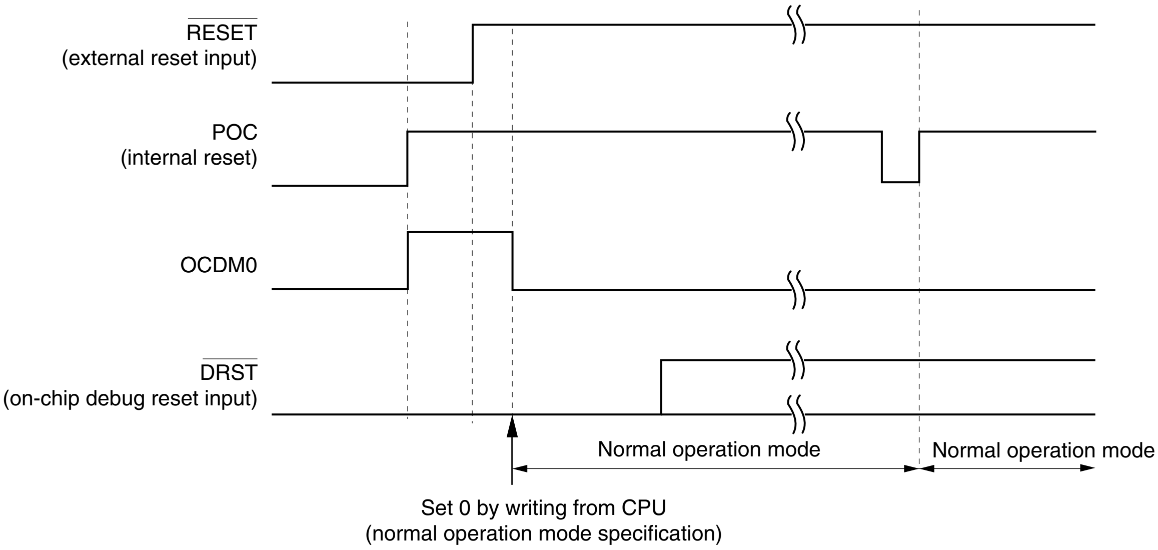
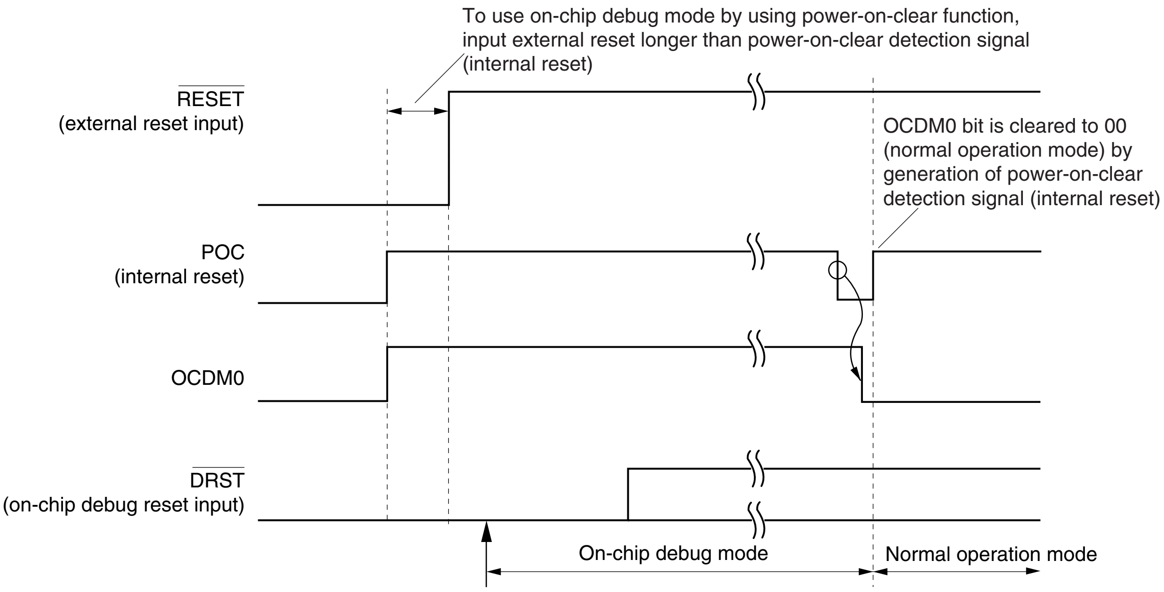


Figure 3-29: Timing Chart of Transition to On-Chip Debug Mode


To use on-chip debug mode by using power-on-clear function, input external reset longer than power-on-clear detection signal (internal reset)
RESET
(external reset input)
OCDM0 bit is cleared to 00 (normal operation mode) by generation of power-on-clear detection signal (internal reset)
POC
(internal reset)
OCDM0
DRST
(on-chip debug reset input)
On-chip debug mode
Normal operation mode

Caution: To use the on-chip debug function of a product with a power-on clear function, input a low level to the $\overline{\text{RESET}}$ input pin for 2000 ms or longer after power application.

(c) Watchdog timer mode register 2 (WDTM2)

The WDTM2 register sets the overflow time and the operation clock of the watchdog timer 2.
The watchdog timer 2 automatically starts in the reset mode after reset is released. Write the WDTM2 register to activate this operation.
For details, refer to **Chapter 10 ”Functions of Watchdog Timer 2” on page 359**.

(3) 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 status. If this wait status occurs, the number of clocks required to execute an instruction increases by the number of wait clocks shown below.
This must be taken into consideration if real-time processing is required.
When 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.

Table 3-3: Access Conditions (1/2)

Peripheral Function	Register Name	Access	k
16-bit timer/ event counter P (TMP) (n = 0 to 4)	TPnCNT	Read	1 or 2
	TPnCCR0	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
	TPnCCR1	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
16-bit timer/ event counter Q (TMQ) (m = 0, 1)	TQmCNT	Read	1 or 2
	TQmCCR0	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
	TQmCCR1	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
	TQmCCR2	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	0 or 2
	TQmCCR3	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
Watchdog timer 2 (WDT2)	WDTM2	Write (when WDT2 operating)	3
A/D converter	ADSCM0	Read	1 or 2
	ADA0CR0 to ADA0CR15	Read	1 or 2
CRC	CRCD	Write	1

Table 3-3: Access Conditions (2/2)

Peripheral Function	Register Name	Access	k
CAN controller (n = 0, 1, m = 0 to 31, a = 1 to 4)	CnGMCTRL, CnGMCS, CnGMABT, CnGMABTD, CnMASKaL, CnMASKaH, CnCTRL, CnLEC, CnINFO, CnERC, CnIE, CnINTS, CnBRP, CnBTR, CnTS	Read/write	$(f_{XX}/f_{CANMODE} + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 2 + 1) \times (1/(2 + j))$ (MAX.)[Note]
	CnRGPT, CnTGPT	Write	$(f_{XX}/f_{CANMODE} + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 2 + 1) \times (1/(2 + j))$ (MAX.)[Note]
		Read	$(f_{XX}/f_{CANMODE} \times 3 + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 4 + 1) \times (1/(2 + j))$ (MAX.)[Note]
	CnLIPT, CnLOPT	Read	$(f_{XX}/f_{CANMODE} \times 3 + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 4 + 1) \times (1/(2 + j))$ (MAX.)[Note]
	CnMDATA01m, CnMDATA0m, CnMDATA1m, CnMDATA23m, CnMDATA2m, CnMDATA3m, CnMDATA45m, CnMDATA4m, CnMDATA5m, CnMDATA67m, CnMDATA6m, CnMDATA7m, CnMDLCm, CnMCONFm, CnMIDLm, CnMIDHm, CnMCTRLm	Write (8 bits)	$(f_{XX}/f_{CANMODE} \times 4 + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 5 + 1) \times (1/(2 + j))$ (MAX.)[Note]
		Write (16 bits)	$(f_{XX}/f_{CANMODE} \times 2 + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 3 + 1) \times (1/(2 + j))$ (MAX.)[Note]
		Read (8/16 bits)	$(f_{XX}/f_{CANMODE} \times 3 + 1) \times (1/(2 + j))$ (MIN.)[Note] $(f_{XX}/f_{CANMODE} \times 4 + 1) \times (1/(2 + j))$ (MAX.)[Note]

Note: Digits below the decimal point are rounded up.

Remark:

f_{XX}:	Main clock frequency = f_{XX}
$f_{CANMODE}$:	CAN module system clock
i:	Values (0) of higher 4 bits of VSWC register
j:	Values (0 or 1) of lower 4 bits of VSWC register

3.5 Programmable I/O Area

The V850E/RS1 includes an additional memory area for the control of on-chip peripherals. The base address of this area is located in the external memory space.
A control register setting is required to enable this additional memory.

Control registers for the following on-chip peripherals are implemented in the Programmable I/O area.

- CAN

3.5.1 Programmable peripheral I/O control register (BPC)

This 16-bit register specifies selection of the Programmable I/O area.
The BPC register can be read or written only in 16-bit units.
RESET input clears BPC to 0000H.
To use the CAN built-in interfaces, set PA15 to 1 by writing the BPC register with a 16-bit memory manipulation instruction.
To disable access to the CAN RAM and CAN registers, clear PA15 to 0 by writing 0000H to the BPC register with a 16-bit memory manipulation instruction.
The mapping of the CAN RAM and registers can be shown in section **3.4.4 "Memory map" on page 78**.
For example, if BPC = 0x8FFB, the programmable area is set to 3FEC000H.

Figure 3-30: Programmable Peripheral I/O Control Register (BPC) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
BPC	PA15	0	PA13	PA12	PA11	PA10	PA09	PA08	FFFFF064H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

7	6	5	4	3	2	1	0
PA07	PA06	PA05	PA04	PA03	PA02	PA01	PA00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15</td><td>PA15</td><td>Enables/disables usage of programmable peripheral I/O area
<table>
<tr><th>PA15</th><th>Usage of programmable peripheral I/O area</th></tr>
<tr><td>0</td><td>Usage of programmable peripheral I/O area is disabled</td></tr>
<tr><td>1</td><td>Usage of programmable peripheral I/O area is enabled</td></tr>
</table></td></tr>
<tr><td>0</td><td>PA13 to PA00</td><td>Specify an address in programmable peripheral I/O area
(correspond to A27 to A14 respectively)</td></tr>
</table>

Caution: When using the CAN controller (PA15 = 1), be sure to set 8FFBH to this register. When not using the CAN controller (PA15 = 0), be sure to set 0000H to this register.

Remark: The programmable peripheral I/O area is fixed by hardware in the V850E core in 3FEC000H to 3FEEFFFH. However, be sure to set 8FFBH to this register when using the CAN controller because it is possible to write it in bit 13 to bit 0 of this register in tool (ICE).

3.6 Peripheral I/O Registers

Table 3-4: Peripheral I/O Registers (1/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF004H	Port DL	PDL	R/W			×	undefined
FFFFF004H	Port DLL	PDLL	R/W	×	×		undefined
FFFFF005H	Port DLH	PDLH	R/W	×	×		undefined
FFFFF008H	Port CS	PCS	R/W	×	×		undefined
FFFFF00AH	Port CT	PCT	R/W	×	×		undefined
FFFFF00CH	Port CM	PCM	R/W	×	×		undefined
FFFFF024H	Port DL mode register	PMDL	R/W			×	FFFFH
FFFFF024H	Port DL mode register L	PMDLL	R/W	×	×		FFH
FFFFF025H	Port DL mode register H	PMDLH	R/W	×	×		FFH
FFFFF028H	Port CS mode register	PMCS	R/W	×	×		FFH
FFFFF02AH	Port CT mode register	PMCT	R/W	×	×		FFH
FFFFF02CH	Port CM mode register	PMCM	R/W	×	×		FFH
FFFFF044H	Port DL mode control register	PMCDL	R/W			×	0000H
FFFFF044H	Port DL mode control register L	PMCDLL	R/W	×	×		00H
FFFFF045H	Port DL mode control register H	PMCDLH	R/W	×	×		00H
FFFFF048H	Port CS mode control register	PMCCS	R/W	×	×		00H
FFFFF04AH	Port CT mode control register	PMCCT	R/W	×	×		00H
FFFFF04CH	Port CM mode control register	PMCCM	R/W	×	×		00H
FFFFF064H	Peripheral area selection control register	BPC	R/W			×	0000H
FFFFF066H	Bus size configuration register	BSC	R/W			×	5555H
FFFFF06EH	NPB strobe wait control register	VSWC	R/W	×	×		77H
FFFFF100H	Interrupt mask register 0	IMR0	R/W			×	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L	R/W	×	×		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H	R/W	×	×		FFH
FFFFF102H	Interrupt mask register 1	IMR1	R/W			×	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L	R/W	×	×		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H	R/W	×	×		FFH
FFFFF104H	Interrupt mask register 2	IMR2	R/W			×	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L	R/W	×	×		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H	R/W	×	×		FFH
FFFFF106H	Interrupt mask register 3	IMR3	R/W			×	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L	R/W	×	×		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H	R/W	×	×		FFH
FFFFF110H	Interrupt control register	LVIIC	R/W	×	×		47H
FFFFF112H	Interrupt control register	PIC0	R/W	×	×		47H
FFFFF114H	Interrupt control register	PIC1	R/W	×	×		47H
FFFFF116H	Interrupt control register	PIC2	R/W	×	×		47H

Table 3-4: Peripheral I/O Registers (2/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF118H	Interrupt control register	PIC3	R/W	×	×		47H
FFFFF11AH	Interrupt control register	PIC4	R/W	×	×		47H
FFFFF11CH	Interrupt control register	PIC5	R/W	×	×		47H
FFFFF11EH	Interrupt control register	PIC6	R/W	×	×		47H
FFFFF120H	Interrupt control register	PIC7	R/W	×	×		47H
FFFFF122H	Interrupt control register	TQ0OVIC	R/W	×	×		47H
FFFFF124H	Interrupt control register	TQ0CCIC0	R/W	×	×		47H
FFFFF126H	Interrupt control register	TQ0CCIC1	R/W	×	×		47H
FFFFF128H	Interrupt control register	TQ0CCIC2	R/W	×	×		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3	R/W	×	×		47H
FFFFF12CH	Interrupt control register	TP0OVIC	R/W	×	×		47H
FFFFF12EH	Interrupt control register	TP0CCIC0	R/W	×	×		47H
FFFFF130H	Interrupt control register	TP0CCIC1	R/W	×	×		47H
FFFFF132H	Interrupt control register	TP1OVIC	R/W	×	×		47H
FFFFF134H	Interrupt control register	TP1CCIC0	R/W	×	×		47H
FFFFF136H	Interrupt control register	TP1CCIC1	R/W	×	×		47H
FFFFF138H	Interrupt control register	TP2OVIC	R/W	×	×		47H
FFFFF13AH	Interrupt control register	TP2CCIC0	R/W	×	×		47H
FFFFF13CH	Interrupt control register	TP2CCIC1	R/W	×	×		47H
FFFFF13EH	Interrupt control register	TP3OVIC	R/W	×	×		47H
FFFFF140H	Interrupt control register	TP3CCIC0	R/W	×	×		47H
FFFFF142H	Interrupt control register	TP3CCIC1	R/W	×	×		47H
FFFFF144H	Interrupt control register	TM0EQIC0	R/W	×	×		47H
FFFFF146H	Interrupt control register	CB0RIC	R/W	×	×		47H
FFFFF148H	Interrupt control register	CB0TIC	R/W	×	×		47H
FFFFF14AH	Interrupt control register	CB1RIC	R/W	×	×		47H
FFFFF14CH	Interrupt control register	CB1TIC	R/W	×	×		47H
FFFFF14EH	Interrupt control register	UA0RIC	R/W	×	×		47H
FFFFF150H	Interrupt control register	UA0TIC	R/W	×	×		47H
FFFFF152H	Interrupt control register	UA1RIC	R/W	×	×		47H
FFFFF154H	Interrupt control register	UA1TIC	R/W	×	×		47H
FFFFF156H	Interrupt control register	ADIC	R/W	×	×		47H
FFFFF158H	Interrupt control register	C0ERRIC	R/W	×	×		47H
FFFFF15AH	Interrupt control register	C0WUPIC	R/W	×	×		47H
FFFFF15CH	Interrupt control register	C0RECIC	R/W	×	×		47H
FFFFF15EH	Interrupt control register	C0TRXIC	R/W	×	×		47H
FFFFF160H	Interrupt control register	C30IC	R/W	×	×		47H
FFFFF162H	Interrupt control register	C30OC	R/W	×	×		47H
FFFFF164H	Interrupt control register	C31IC	R/W	×	×		47H
FFFFF166H	Interrupt control register	C31OC	R/W	×	×		47H

Table 3-4: Peripheral I/O Registers (3/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF168H	Interrupt control register	CB2RIC	R/W	×	×		47H
FFFFF16AH	Interrupt control register	CB2TIC	R/W	×	×		47H
FFFFF16CH	Interrupt control register	TQ1OVIC	R/W	×	×		47H
FFFFF16EH	Interrupt control register	TQ1CCIC0	R/W	×	×		47H
FFFFF170H	Interrupt control register	TQ1CCIC1	R/W	×	×		47H
FFFFF172H	Interrupt control register	TQ1CCIC2	R/W	×	×		47H
FFFFF174H	Interrupt control register	TQ1CCIC3	R/W	×	×		47H
FFFFF17AH	Interrupt control register	C1ERRIC	R/W	×	×		47H
FFFFF17CH	Interrupt control register	C1WUPIC	R/W	×	×		47H
FFFFF17EH	Interrupt control register	C1RECIC	R/W	×	×		47H
FFFFF180H	Interrupt control register	C1TRXIC	R/W	×	×		47H
FFFFF182H	Interrupt control register	DMAIC0	R/W	×	×		47H
FFFFF184H	Interrupt control register	DMAIC1	R/W	×	×		47H
FFFFF186H	Interrupt control register	DMAIC2	R/W	×	×		47H
FFFFF188H	Interrupt control register	DMAIC3	R/W	×	×		47H
FFFFF18AH	Interrupt control register	DMAIC4	R/W	×	×		47H
FFFFF18CH	Interrupt control register	DMAIC5	R/W	×	×		47H
FFFFF1FAH	In-service priority register	ISPR	R	×	×		00H
FFFFF1FCH	Command register	PRCMD	W		×		undefined
FFFFF1FEH	Power save control register	PSC	R/W	×	×		00H
FFFFF200H	A/D converter scan mode register 0	ADSCM0	R/W			√	0000H
FFFFF200H	A/D converter scan mode register 0L	ADSCM0L	R/W	×	×		00H
FFFFF201H	A/D converter scan mode register 0H	ADSCM0H	R/W	×	×		00H
FFFFF203H	A/D converter scan mode register 1H	ADSCM1H	R/W	×	×		00H
FFFFF204H	A/D extended mode control register	ADVMS0	R/W	×	×		01H
FFFFF220H	A/D conversion result register 0	ADA0CR0	R			×	0000H
FFFFF222H	A/D conversion result register 1	ADA0CR1	R			×	0000H
FFFFF224H	A/D conversion result register 2	ADA0CR2	R			×	0000H
FFFFF226H	A/D conversion result register 3	ADA0CR3	R			×	0000H
FFFFF228H	A/D conversion result register 4	ADA0CR4	R			×	0000H
FFFFF22AH	A/D conversion result register 5	ADA0CR5	R			×	0000H
FFFFF22CH	A/D conversion result register 6	ADA0CR6	R			×	0000H
FFFFF22EH	A/D conversion result register 7	ADA0CR7	R			×	0000H
FFFFF230H	A/D conversion result register 8	ADA0CR8	R			×	0000H
FFFFF232H	A/D conversion result register 9	ADA0CR9	R			×	0000H
FFFFF234H	A/D conversion result register 10	ADA0CR10	R			×	0000H
FFFFF236H	A/D conversion result register 11	ADA0CR11	R			×	0000H
FFFFF238H	A/D conversion result register 12	ADA0CR12	R			×	0000H
FFFFF23AH	A/D conversion result register 13	ADA0CR13	R			×	0000H
FFFFF23CH	A/D conversion result register 14	ADA0CR14	R			×	0000H

Table 3-4: Peripheral I/O Registers (4/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF23EH	A/D conversion result register 15	ADA0CR15	R			×	0000H
FFFFF240H	A/D conversion result register DD	ADA0CRDD	R			×	0000H
FFFFF242H	A/D conversion result register SS	ADA0CRSS	R			×	0000H
FFFFF308H	Selector operation control register 0	SELCNT0	R/W	×	×		00H
FFFFF30AH	Selector operation control register 1	SELCNT1	R/W	×	×		00H
FFFFF30CH	Extension clock select register	EXCKSEL	R/W	×	×		00H
FFFFF310H	CRC input register	CRCIN	R/W		×		00H
FFFFF312H	CRC data register	CRCD	R/W			×	0000H
FFFFF318H	Noise filter control register	NFC	R/W	×	×		00H
FFFFF400H	Port 0	P0	R/W	×	×		undefined
FFFFF402H	Port 1	P1	R/W	×	×		undefined
FFFFF406H	Port 3	P3	R/W			×	undefined
FFFFF406H	Port 3L	P3L	R/W	×	×		undefined
FFFFF407H	Port 3H	P3H	R/W	×	×		undefined
FFFFF408H	Port 4	P4	R/W	×	×		undefined
FFFFF40AH	Port 5	P5	R/W	×	×		undefined
FFFFF40EH	Port 7	P7	R/W			×	undefined
FFFFF40EH	Port 7L	P7L	R/W	×	×		undefined
FFFFF40FH	Port 7H	P7H	R/W	×	×		undefined
FFFFF412H	Port 9	P9	R/W			×	undefined
FFFFF412H	Port 9L	P9L	R/W	×	×		undefined
FFFFF413H	Port 9H	P9H	R/W	×	×		undefined
FFFFF420H	Port 0 mode register	PM0	R/W	×	×		FFH
FFFFF422H	Port 1 mode register	PM1	R/W	×	×		FFH
FFFFF426H	Port 3 mode register	PM3	R/W			×	FFFFH
FFFFF426H	Port 3 mode register L	PM3L	R/W	×	×		FFH
FFFFF427H	Port 3 mode register H	PM3H	R/W	×	×		FFH
FFFFF428H	Port 4 mode register	PM4	R/W	×	×		FFH
FFFFF42AH	Port 5 mode register	PM5	R/W	×	×		FFH
FFFFF42EH	Port 7 mode register	PM7	R/W			×	FFFFH
FFFFF42EH	Port 7 mode register L	PM7L	R/W	×			FFH
FFFFF42FH	Port 7 mode register H	PM7H	R/W	×	×		FFH
FFFFF432H	Port 9 mode register	PM9	R/W			×	FFFFH
FFFFF432H	Port 9 mode register L	PM9L	R/W	×	×		FFH
FFFFF433H	Port 9 mode register H	PM9H	R/W	×	×		FFH
FFFFF440H	Port 0 mode control register	PMC0	R/W	×	×		00H
FFFFF442H	Port 1 mode control register	PMC1	R/W	×	×		00H
FFFFF446H	Port 3 mode control register	PMC3	R/W			×	0000H
FFFFF446H	Port 3 mode control register L	PMC3L	R/W	×	×		00H
FFFFF447H	Port 3 mode control register H	PMC3H	R/W	×	×		00H

Table 3-4: Peripheral I/O Registers (5/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF448H	Port 4 mode control register	PMC4	R/W	×	×		00H
FFFFF44AH	Port 5 mode control register	PMC5	R/W	×	×		00H
FFFFF452H	Port 9 mode control register	PMC9	R/W			×	0000H
FFFFF452H	Port 9 mode control register L	PMC9L	R/W	×	×		00H
FFFFF453H	Port 9 mode control register H	PMC9H	R/W	×	×		00H
FFFFF460H	Port 0 function control register	PFC0	R/W	×	×		00H
FFFFF462H	Port 1 function control register	PFC1	R/W	×	×		00H
FFFFF466H	Port 3 function control register	PFC3	R/W			×	0000H
FFFFF466H	Port 3 function control register L	PFC3L	R/W	×	×		00H
FFFFF467H	Port 3 function control register H	PFC3H	R/W	×	×		00H
FFFFF46AH	Port 5 function control register	PFC5	R/W	×	×		00H
FFFFF472H	Port 9 function control register	PFC9	R/W			×	0000H
FFFFF472H	Port 9 function control register L	PFC9L	R/W	×	×		00H
FFFFF473H	Port 9 function control register H	PFC9H	R/W	×	×		00H
FFFFF484H	Data wait control register 0	DWC0	R/W			×	7777H
FFFFF488H	Address wait control register	AWC	R/W			×	FFFFH
FFFFF48AH	Bus cycle control register	BCC	R/W			×	AAAAH
FFFFF540H	TMQ0 timer control register 0	TQ0CTL0	R/W	×	×		00H
FFFFF541H	TMQ0 timer control register 1	TQ0CTL1	R/W	×	×		00H
FFFFF542H	TMQ0 timer-specific I/O control register 0	TQ0IOC0	R/W	×	×		00H
FFFFF543H	TMQ0 timer-specific I/O control register 1	TQ0IOC1	R/W	×	×		00H
FFFFF544H	TMQ0 timer-specific I/O control register 2	TQ0IOC2	R/W	×	×		00H
FFFFF545H	TMQ0 timer option register	TQ0OPT0	R/W	×	×		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0	R/W			×	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1	R/W			×	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2	R/W			×	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3	R/W			×	0000H
FFFFF54EH	TMQ0 timer read buffer register	TQ0CNT	R			×	0000H
FFFFF590H	TMP0 timer control register 0	TP0CTL0	R/W	×	×		00H
FFFFF591H	TMP0 timer control register 1	TP0CTL1	R/W	×	×		00H
FFFFF592H	TMP0 timer-specific I/O control register 0	TP0IOC0	R/W	×	×		00H
FFFFF593H	TMP0 timer-specific I/O control register 1	TP0IOC1	R/W	×	×		00H
FFFFF594H	TMP0 timer-specific I/O control register 2	TP0IOC2	R/W	×	×		00H
FFFFF595H	TMP0 timer option register	TP0OPT0	R/W	×	×		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0	R/W			×	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1	R/W			×	0000H
FFFFF59AH	TMP0 count register	TP0CNT	R			×	0000H
FFFFF5A0H	TMP1 timer control register 0	TP1CTL0	R/W	×	×		00H
FFFFF5A1H	TMP1 timer control register 1	TP1CTL1	R/W	×	×		00H
FFFFF5A2H	TMP1 timer-specific I/O control register 0	TP1IOC0	R/W	×	×		00H

Table 3-4: Peripheral I/O Registers (6/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF5A3H	TMP1 timer-specific I/O control register 1	TP1IOC1	R/W	×	×		00H
FFFFF5A4H	TMP1 timer-specific I/O control register 2	TP1IOC2	R/W	×	×		00H
FFFFF5A5H	TMP1 timer option register	TP1OPT0	R/W	×	×		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0	R/W			×	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1	R/W			×	0000H
FFFFF5AAH	TMP1 count register	TP1CNT	R/W			×	0000H
FFFFF5B0H	TMP2 timer control register 0	TP2CTL0	R/W	×	×		00H
FFFFF5B1H	TMP2 timer control register 1	TP2CTL1	R/W	×	×		00H
FFFFF5B2H	TMP2 timer-specific I/O control register 0	TP2IOC0	R/W	×	×		00H
FFFFF5B3H	TMP2 timer-specific I/O control register 1	TP2IOC1	R/W	×	×		00H
FFFFF5B4H	TMP2 timer-specific I/O control register 2	TP2IOC2	R/W	×	×		00H
FFFFF5B5H	TMP2 timer option register	TP2OPT0	R/W	×	×		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0	R/W			×	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1	R/W			×	0000H
FFFFF5BAH	TMP2 count register	TP2CNT	R			×	0000H
FFFFF5C0H	TMP3 timer control register 0	TP3CTL0	R/W	×	×		00H
FFFFF5C1H	TMP3 timer control register 1	TP3CTL1	R/W	×	×		00H
FFFFF5C2H	TMP3 timer-specific I/O control register 0	TP3IOC0	R/W	×	×		00H
FFFFF5C3H	TMP3 timer-specific I/O control register 1	TP3IOC1	R/W	×	×		00H
FFFFF5C4H	TMP3 timer-specific I/O control register 2	TP3IOC2	R/W	×	×		00H
FFFFF5C5H	TMP3 timer option register	TP3OPT0	R/W	×	×		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0	R/W			×	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1	R/W			×	0000H
FFFFF5CAH	TMP3 count register	TP3CNT	R			×	0000H
FFFFF610H	TMQ1 timer control register 0	TQ1CTL0	R/W	×	×		00H
FFFFF611H	TMQ1 timer control register 1	TQ1CTL1	R/W	×	×		00H
FFFFF612H	TMQ1 timer-specific I/O control register 0	TQ1IOC0	R/W	×	×		00H
FFFFF613H	TMQ1 timer-specific I/O control register 1	TQ1IOC1	R/W	×	×		00H
FFFFF614H	TMQ1 timer-specific I/O control register 2	TQ1IOC2	R/W	×	×		00H
FFFFF615H	TMQ1 timer option register	TQ1OPT0	R/W	×	×		00H
FFFFF616H	TMQ1 capture/compare register 0	TQ1CCR0	R/W			×	0000H
FFFFF618H	TMQ1 capture/compare register 1	TQ1CCR1	R/W			×	0000H
FFFFF61AH	TMQ1 capture/compare register 2	TQ1CCR2	R/W			×	0000H
FFFFF61CH	TMQ1 capture/compare register 3	TQ1CCR3	R/W			×	0000H
FFFFF61EH	TMQ1 timer read buffer register	TQ1CNT	R			×	0000H
FFFFF690H	TMM0 timer control register 0	TM0CTL0	R/W	×	×		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0	R/W			×	0000H
FFFFF6C0H	Oscillation stabilization time selection register	OSTS	R/W		×		03H
FFFFF6C1H	PLL lockup time specification register	PLLS	R/W		×		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2	R/W		×		67H

Table 3-4: Peripheral I/O Registers (7/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFF6D1H	Watchdog timer enable register	WDTE	R/W		×		9AH
FFFFF702H	Port 1 function control expansion register	PFCE1	R/W	×	×		00H
FFFFF706H	Port 3 function control expansion register	PFCE3	R/W	×	×		00H
FFFFF712H	Port 9 function control expansion register	PFCE9	R/W			×	0000H
FFFFF712H	Port 9 function control expansion register L	PFCE9L	R/W	×	×		00H
FFFFF713H	Port 9 function control expansion register H	PFCE9H	R/W	×	×		00H
FFFFF802H	System status register	SYS	R/W	×	×		00H
FFFFF80CH	Ring-OSC mode register	RCM	R/W	×	×		00H
FFFFF820H	Power save mode register	PSMR	R/W	×	×		00H
FFFFF822H	Clock control register	CKC	R/W	×	×		03H
FFFFF824H	PLL lock status register	LOCKR	R	×	×		02H
FFFFF828H	Processor clock control register	PCC	R/W	×	×		00H
FFFFF82CH	PLL control register 0	PLLCTL0	R/W	×	×		00H
FFFFF82DH	PLL control register 1	PLLCTL1	R/W	×	×		00H
FFFFF82EH	CPU operation clock status register	CCLS	R	×	×		00H Note
FFFFF82FH	Programmable clock mode register	PCLM	R/W	×	×		01H
FFFFF860H	Clock selection register 0	OCKS0	R/W		×		11H
FFFFF864H	Clock selection register 1	OCKS1	R/W		×		10H
FFFFF868H	Clock selection register 2	OCKS2	R/W		×		00H
FFFFF86CH	Clock selection register 3	OCKS3	R/W		×		00H
FFFFF870H	Clock monitor mode register	CLM	R/W	×	×		00H
FFFFF87AH	Port Function Swap control register	PSWAP	R/W	×	×		00H
FFFFF888H	Reset status flag register	RESF	R/W	×	×		00H
FFFFF890H	Low-voltage detection register	LVIM	R/W	×	×		00H
FFFFF891H	Low-voltage detection level selection register	LVIS	R/W		×		00H
FFFFF892H	Internal RAM data status register	RAMS	R/W	×	×		01H
FFFFF8B0H	BRG0 prescaler mode register	PRSM0	R/W		×		00H
FFFFF8B1H	BRG0 precaler compare register	PRSCM0	R/W		×		00H
FFFFF9FCH	On-chip debug shared port setting register	OCDM	R/W	×	×		01H
FFFFF9FEH	Peripheral emulation register	PEUM1	R/W	×	×		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0	R/W	×	×		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1	R/W		×		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2	R/W		×		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0	R/W	×	×		14H
FFFFFA04H	UARTA0 status register	UA0STR	R/W	×	×		00H
FFFFFA06H	UARTA0 reception data register	UA0RX	R		×		FFH
FFFFFA07H	UARTA0 transmission data register	UA0TX	R/W		×		FFH

Note: After reset release, when CPU starts operation by the ring-OSC for some reason or accident, it is set to 01H.

Table 3-4: Peripheral I/O Registers (8/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFFA10H	UARTA1 control register 0	UA1CTL0	R/W	×	×		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1	R/W		×		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2	R/W		×		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0	R/W	×	×		14H
FFFFFA14H	UARTA1 status register	UA1STR	R/W	×	×		00H
FFFFFA16H	UARTA1 reception data register	UA1RX	R		×		FFH
FFFFFA17H	UARTA1 transmission data register	UA1TX	R/W		×		FFH
FFFFFA48H	Port CS pull-up resistor option register	PUCS	R/W	×	×		00H
FFFFFA4AH	Port CT pull-up resistor option register	PUCT	R/W	×	×		00H
FFFFFA4CH	Port CM pull-up resistor option register	PUCM	R/W	×	×		00H
FFFFFA68H	Port CS pull-down resistor option register	PDCS	R/W	×	×		00H
FFFFFA6AH	Port CT pull-down resistor option register	PDCT	R/W	×	×		00H
FFFFFA6CH	Port CM pull-down resistor option register	PDCM	R/W	×	×		00H
FFFFFB00H	TIP00 noise filter circuit control register	P00NFC	R/W	×	×		00H
FFFFFB04H	TIP01 noise filter circuit control register	P01NFC	R/W	×	×		00H
FFFFFB08H	TIP10 noise filter circuit control register	P10NFC	R/W	×	×		00H
FFFFFB0CH	TIP11 noise filter circuit control register	P11NFC	R/W	×	×		00H
FFFFFB10H	TIP20 noise filter circuit control register	P20NFC	R/W	×	×		00H
FFFFFB14H	TIP21 noise filter circuit control register	P21NFC	R/W	×	×		00H
FFFFFB18H	TIP30 noise filter circuit control register	P30NFC	R/W	×	×		00H
FFFFFB1CH	TIP31 noise filter circuit control register	P31NFC	R/W	×	×		00H
FFFFFB50H	TIQ00 noise filter circuit control register	Q00NFC	R/W	×	×		00H
FFFFFB54H	TIQ01 noise filter circuit control register	Q01NFC	R/W	×	×		00H
FFFFFB58H	TIQ02 noise filter circuit control register	Q02NFC	R/W	×	×		00H
FFFFFB5CH	TIQ03 noise filter circuit control register	Q03NFC	R/W	×	×		00H
FFFFFB60H	TIQ10 noise filter circuit control register	Q10NFC	R/W	×	×		00H
FFFFFB64H	TIQ11 noise filter circuit control register	Q11NFC	R/W	×	×		00H
FFFFFB68H	TIQ12 noise filter circuit control register	Q12NFC	R/W	×	×		00H
FFFFFB6CH	TIQ13 noise filter circuit control register	Q13NFC	R/W	×	×		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0	R/W	×	×		00H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1	R/W	×	×		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3	R/W	×	×		00H
FFFFFC08H	NMI mode register	NMIM	R/W	×	×		03H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H	R/W	×	×		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0	R/W	×	×		00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1	R/W	×	×		00H

Table 3-4: Peripheral I/O Registers (9/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFFC26H	External interrupt rising edge specification register 3	INTR3	R/W	×	×		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H	R/W	×	×		00H
FFFFFC40H	Pull-up resistor option register 0	PU0	R/W	×	×		00H
FFFFFC42H	Pull-up resistor option register 0	PU1	R/W	×	×		00H
FFFFFC46H	Pull-up resistor option register 3	PU3	R/W	×	×	×	0000H
FFFFFC46H	Pull-up resistor option register 3L	PU3L	R/W	×	×		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H	R/W	×	×		00H
FFFFFC48H	Pull-up resistor option register 4	PU4	R/W	×	×		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5	R/W	×	×		00H
FFFFFC52H	Pull-up resistor option register 9	PU9	R/W			×	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L	R/W	×	×		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H	R/W	×	×		00H
FFFFFCA0H	Pull-down resistor option register 0	PD0	R/W	×	×		00H
FFFFFCA2H	Pull-down resistor option register 0	PD1	R/W	×	×		00H
FFFFFCA6H	Pull-down resistor option register 3	PD3	R/W			×	0000H
FFFFFC46H	Pull-down resistor option register 3L	PD3L	R/W	×	×		00H
FFFFFC47H	Pull-down resistor option register 3H	PD3H	R/W	×	×		00H
FFFFFCA8H	Pull-down resistor option register 4	PD4	R/W	×	×		00H
FFFFFCAAH	Pull-down resistor option register 5	PD5	R/W	×	×		00H
FFFFFCB2H	Pull-down resistor option register 9	PD9	R/W			×	0000H
FFFFFCB2H	Pull-down resistor option register 9L	PD9L	R/W	×	×		00H
FFFFFCB3H	Pull-down resistor option register 9H	PD9H	R/W	×	×		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0	R/W	×	×		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1	R/W	×	×		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2	R/W		×		00H
FFFFFD03H	CSIB0 status register	CB0STR	R/W	×	×		00H
FFFFFD04H	CSIB0 reception data register	CB0RX	R			×	0000H
FFFFFD04H	CSIB0 reception data register L	CB0RXL	R		×		00H
FFFFFD06H	CSIB0 transmission data register	CB0TX	R/W			×	0000H
FFFFFD06H	CSIB0 transmission data register L	CB0TXL	R/W		×		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0	R/W	×	×		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1	R/W	×	×		00H
FFFFFD12H	CSIB1 control register 2	CB1CTL2	R/W		×		00H
FFFFFD13H	CSIB1 status register	CB1STR	R/W	×	×		00H
FFFFFD14H	CSIB1 reception data register	CB1RX	R			×	0000H
FFFFFD14H	CSIB1 reception data register L	CB1RXL	R		×		00H
FFFFFD16H	CSIB1 transmission data register	CB1TX	R/W			×	0000H
FFFFFD16H	CSIB1 transmission data register L	CB1TXL	R/W		×		00H
FFFFFD40H	CSI30 operation mode register	CSIM0	R/W	×	×		00H

Table 3-4: Peripheral I/O Registers (10/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFFD41H	CSI30 clock selection register	CSIC0	R/W	×	×		07H
FFFFFD42H	CSI30 receive data buffer	SIRB0	R			×	0000H
FFFFFD42H	CSI30 receive data buffer L	SIRB0L	R		×		00H
FFFFFD43H	CSI30 receive data buffer H	SIRB0H	R		×		00H
FFFFFD44H	CSI30 chip select FIFO buffer	SFCS0	R/W			×	FFFFH
FFFFFD44H	CSI30 chip select FIFO buffer L	SFCS0L	R/W		×		FFH
FFFFFD45H	CSI30 chip select FIFO buffer H	SFCS0H	R/W		×		FFH
FFFFFD46H	CSI30 transmit data FIFO buffer	SFDB0	R/W			×	0000H
FFFFFD46H	CSI30 transmit data FIFO buffer L	SFDB0L	R/W		×		00H
FFFFFD47H	CSI30 transmit data FIFO buffer H	SFDB0H	R/W		×		00H
FFFFFD48H	CSI30 FIFO buffer status register	SFA0	R/W	×	×		20H
FFFFFD49H	CSI30 transmit data length select register	CSIL0	R/W	×	×		00H
FFFFFD4CH	CSI30 transmit data number select register	SFN0	R/W	×	×		00H
FFFFFD60H	CSI31 operation mode register	CSIM1	R/W	×	×		00H
FFFFFD61H	CSI31 clock selection register	CSIC1	R/W	×	×		07H
FFFFFD62H	CSI31 receive data buffer	SIRB1	R			×	0000H
FFFFFD62H	CSI31 receive data buffer L	SIRB1L	R		×		00H
FFFFFD63H	CSI31 receive data buffer H	SIRB1H	R		×		00H
FFFFFD64H	CSI31 chip select FIFO buffer	SFCS1	R/W			×	FFFFH
FFFFFD64H	CSI31 chip select FIFO buffer L	SFCS1L	R/W		×		FFH
FFFFFD65H	CSI31 chip select FIFO buffer H	SFCS1H	R/W		×		FFH
FFFFFD66H	CSI31 transmit data FIFO buffer	SFDB1	R/W			×	0000H
FFFFFD66H	CSI31 transmit data FIFO buffer L	SFDB1L	R/W		×		00H
FFFFFD67H	CSI31 transmit data FIFO buffer H	SFDB1H	R/W		×		00H
FFFFFD68H	CSI31 FIFO buffer status register	SFA1	R/W	×	×		20H
FFFFFD69H	CSI31 transmit data length select register	CSIL1	R/W	×	×		00H
FFFFFD6CH	CSI31 transmit data number select register	SFN1	R/W	×	×		00H
FFFFFDE0H	MDMA Channel control register 0	DMCHC0	R/W	×	×		
FFFFFDE1H	MDMA Channel control register 1	DMCHC1	R/W	×	×		
FFFFFDE2H	MDMA Channel control register 2	DMCHC2	R/W	×	×		
FFFFFDE3H	MDMA Channel control register 3	DMCHC3	R/W	×	×		
FFFFFDE4H	MDMA Channel control register 4	DMCHC4	R/W	×	×		
FFFFFDE5H	MDMA Channel control register 5	DMCHC5	R/W	×	×		
FFFFFE00H	MDMA Control register	DMC	R/W	×	×		
FFFFFE02H	MDMA built-in BSC register	DMBSC	R			×	
FFFFFE04H	MDMA channel status flag register	DMSF	R			×	
FFFFFE08H	MDMA ch0 source address register	DMSA0	R/W				×
FFFFFE08H	MDMA ch0 source address register L	DMSA0L	RW			×	
FFFFFE0AH	MDMA ch0 source address register H	DMSA0H	R/W			×	

Table 3-4: Peripheral I/O Registers (11/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFFE0CH	MDMA ch0 destination address register	DMDA0	R/W				×
FFFFFE0CH	MDMA ch0 destination address register L	DMDA0L	R/W			×	
FFFFFE0EH	MDMA ch0 destination address register H	DMDA0H	R/W			×	
FFFFFE10H	MDMA ch0 transfer count register	DMBC0	R/W			×	
FFFFFE12H	MDMA ch0 addressing control register	DMADC0	R/W			×	
FFFFFE12H	MDMA ch0 addressing control register L	DMADC0L	R/W		×		
FFFFFE13H	MDMA ch0 addressing control register H	DMADC0H	R/W		×		
FFFFFE14H	MDMA ch1 source address register	DMSA1	R/W				×
FFFFFE14H	MDMA ch1 source address register L	DMSA1L	R/W			×	
FFFFFE16H	MDMA ch1 source address register H	DMSA1H	R/W			×	
FFFFFE18H	MDMA ch1 destination address register	DMDA1	R/W				×
FFFFFE18H	MDMA ch1 destination address register L	DMDA1L	R/W			×	
FFFFFE1AH	MDMA ch1 destination address register H	DMDA1H	R/W			×	
FFFFFE1CH	MDMA ch1 transfer count register	DMBC1	R/W			×	
FFFFFE1EH	MDMA ch1 addressing control register	DMADC1	R/W			×	
FFFFFE1EH	MDMA ch1 addressing control register L	DMADC1L	R/W		×		
FFFFFE1FH	MDMA ch1 addressing control register H	DMADC1H	R/W		×		
FFFFFE20H	MDMA ch2 source address register	DMSA2	R/W				×
FFFFFE20H	MDMA ch2 source address register L	DMSA2L	R/W			×	
FFFFFE22H	MDMA ch2 source address register H	DMSA2H	R/W			×	
FFFFFE24H	MDMA ch2 destination address register	DMDA2	R/W				×
FFFFFE24H	MDMA ch2 destination address register L	DMDA2L	R/W			×	
FFFFFE26H	MDMA ch2 destination address register H	DMDA2H	R/W			×	
FFFFFE28H	MDMA ch2 transfer count register	DMBC2	R/W			×	
FFFFFE2AH	MDMA ch2 addressing control register	DMADC2	R/W			×	
FFFFFE2AH	MDMA ch2 addressing control register L	DMADC2L	R/W		×		
FFFFFE2BH	MDMA ch2 addressing control register H	DMADC2H	R/W		×		
FFFFFE2CH	MDMA ch3 source address register	DMSA3	R/W				×
FFFFFE2CH	MDMA ch3 source address register L	DMSA3L	R/W			×	
FFFFFE2EH	MDMA ch3 source address register H	DMSA3H	R/W			×	
FFFFFE30H	MDMA ch3 destination address register	DMDA3	R/W				×
FFFFFE30H	MDMA ch3 destination address register L	DMDA3L	R/W			×	
FFFFFE32H	MDMA ch3 destination address register H	DMDA3H	R/W			×	
FFFFFE34H	MDMA ch3 transfer count register	DMBC3	R/W			×	
FFFFFE36H	MDMA ch3 addressing control register	DMADC3	R/W			×	
FFFFFE36H	MDMA ch3 addressing control register L	DMADC3L	R/W		×		
FFFFFE37H	MDMA ch3 addressing control register H	DMADC3H	R/W		×		
FFFFFE38H	MDMA ch4 source address register	DMSA4	R/W				×
FFFFFE38H	MDMA ch4 source address register L	DMSA4L	R/W			×	
FFFFFE3AH	MDMA ch4 source address register H	DMSA4H	R/W			×	

Table 3-4: Peripheral I/O Registers (12/12)

Address	Description	Symbol	R/W	Manipulatable bits			Default value
				1-bit	8-bit	16-bit	
FFFFFE3CH	MDMA ch4 destination address register	DMDA4	R/W				×
FFFFFE3CH	MDMA ch4 destination address register L	DMDA4L	R/W			×	
FFFFFE3EH	MDMA ch4 destination address register H	DMDA4H	R/W			×	
FFFFFE40H	MDMA ch4 transfer count register	DMBC4	R/W			×	
FFFFFE42H	MDMA ch4 addressing control register	DMADC4	R/W			×	
FFFFFE42H	MDMA ch4 addressing control register L	DMADC4L	R/W		×		
FFFFFE43H	MDMA ch4 addressing control register H	DMADC4H	R/W		×		
FFFFFE44H	MDMA ch5 source address register	DMSA5	R/W				×
FFFFFE44H	MDMA ch5 source address register L	DMSA5L	R/W			×	
FFFFFE46H	MDMA ch5 source address register H	DMSA5H	R/W			×	
FFFFFE48H	MDMA ch5 destination address register	DMDA5	R/W				×
FFFFFE48H	MDMA ch5 destination address register L	DMDA5L	R/W			×	
FFFFFE4AH	MDMA ch5 destination address register H	DMDA5H	R/W			×	
FFFFFE4CH	MDMA ch5 transfer count register	DMBC5	R/W			×	
FFFFFE4EH	MDMA ch5 addressing control register	DMADC5	R/W			×	
FFFFFE4EH	MDMA ch5 addressing control register L	DMADC5L	R/W		×		
FFFFFE4FH	MDMA ch5 addressing control register H	DMADC5H	R/W		×		
FFFFFF40H	DMA trigger factor register 0	DTFR0	R/W	×	×		
FFFFFF42H	DMA trigger factor register 1	DTFR1	R/W	×	×		
FFFFFF44H	DMA trigger factor register 2	DTFR2	R/W	×	×		
FFFFFF46H	DMA trigger factor register 3	DTFR3	R/W	×	×		
FFFFFF48H	DMA trigger factor register 4	DTFR4	R/W	×	×		
FFFFFF4AH	DMA trigger factor register 5	DTFR5	R/W	×	×		

Chapter 4 Port Functions

4.1 Features

- I/O ports: 84
- Other peripheral function I/O pins can be alternatively used
- Input/output specifiable in 1-bit units

4.2 Basic Port Configuration

The V850E/RS1 features a total of 84 I/O ports consisting of the ports 0, 1, 3, 4, 5, 7, 9, CM, CS, CT, and DL.
The port configuration is shown below.

Figure 4-1: Port Configuration Diagram

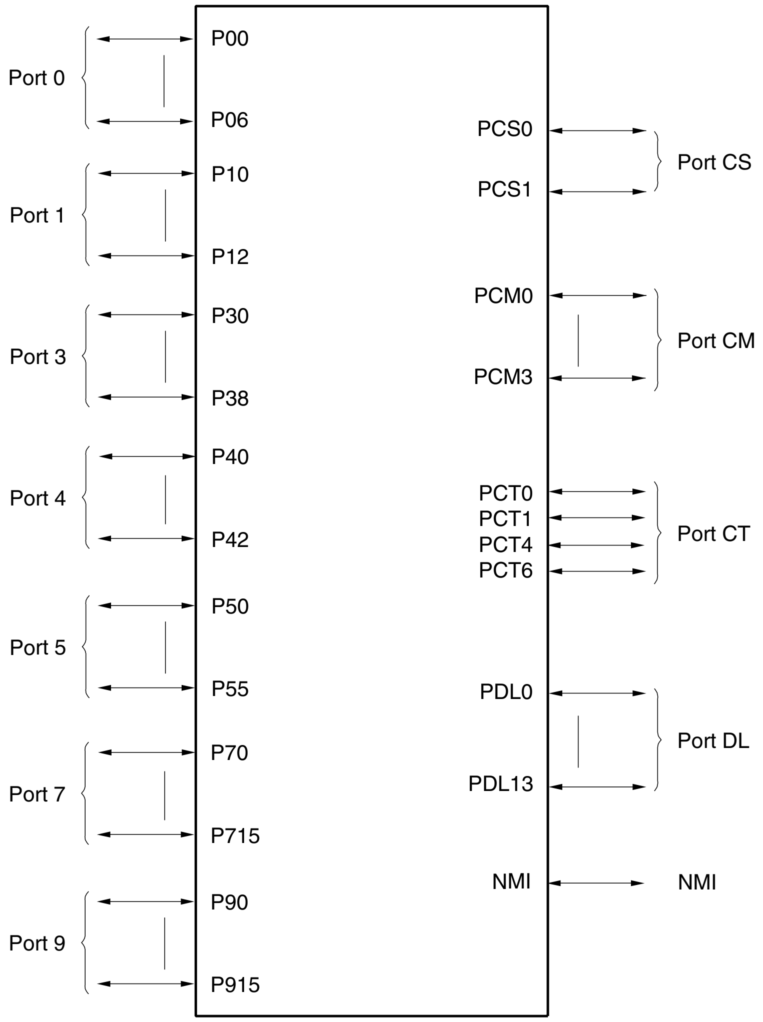


Table 4-1: I/O Buffer Power Supplies for Pins

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
BV_{DD1}	Port CM, Port CS, port CT, port DL
V_{DD1}	Port 0, Port 1, Port 3, Port 4, Port 5, Port 9, $\overline{\text{RESET}}$, NMI

Remark: NMI and $\overline{\text{RESET}}$ pins have no Port function.

4.3 Port Configuration

4.3.1 Table of port configuration

Table 4-2: Control Register Setting (1/4)

Pin No.	Pin Name	Control Register	Value								Other
		PMC	0	1				1	1	1	
		PFC	x	0				1	0	1	
		PFCE	x	0				0	1	1	
		PSWAP	xx	00	01	10	11	xx	xx	xx	
1	AV$_{\text{REF0}}$										
2	AV$_{\text{SS}}$										
3	NMI		NMI								
4	P10/INTP0		P10	INTP0				x	x	x	
5	V$_{\text{DD1}}$										
6	P11/TIP31/TOP31		P11	TIP31				TOP31	x	x	
7	P12/TIP30/TOP30/ ADTRG		P12	TIP30				TOP30	ADTRG	-	
8	FLMD0										FLMD0
9	V$_{\text{DD0}}$										
10	REGC0										
11	V$_{\text{SS0}}$										
12	X1										
13	X2										
14	$\overline{\text{RESET}}$		$\overline{\text{RESET}}$								
15	P00/SI31		P00	SI31				x	x	x	
16	P01/SO31		P01	SO31				x	x	x	
17	P02/$\overline{\text{SCK31}}$		P02	$\overline{\text{SCK31}}$				x	x	x	
18	P03/CS310		P03	CS310			No-Func	x	x	x	
19	P04/CS311		P04	CS311				x	x	x	
20	P05/CS312/INTP2		P05	CS312			No-Func	INTP2	x	x	
21	P06/CS313/INTP3		P06	CS313				INTP3	x	x	
22	P40/SIB0		P40	SIB0				x	x	x	
23	P41/SOB0		P41	SOB0				x	x	x	
24	P42/$\overline{\text{SCKB0}}$		P42	$\overline{\text{SCKB0}}$				x	x	x	
25	P30/TXDA0/[$\overline{\text{ASCKA0}}$]		P30	TXDA0	$\overline{\text{ASCKA0}}$	TXDA0		x	x	x	
26	P31/RXDA0/INTP7		P31	RXDA0				INTP7	x	x	
27	P32/$\overline{\text{ASCKA0}}$/[TXDA0]/ (CS310)		P32	$\overline{\text{ASCKA0}}$	TXDA0	$\overline{\text{ASCKA0}}$	CS310	x	x	x	
28	P33/CTXD0		P33	CTXD0				x	x	x	

Remarks: 1. x: Don't care.

2. No-Func: No function

Table 4-2: Control Register Setting (2/4)

Pin No.	Pin Name	Control Register	Value								Other
		PMC	0	1				1	1	1	
		PFC	x	0				1	0	1	
		PFCE	x	0				0	1	1	
		PSWAP	xx	00	01	10	11	xx	xx	xx	
29	P34/CRXD0		P34	CRXD0				x	x	x	
30	P35/CRXD1		P35	CRXD1				x	x	x	
31	P36/CTXD1		P36	CTXD1				x	x	x	
32	P37/TIP00/TOP00/ INTP1/(CS312)		P37	TIP00			CS312	TOP00	INTP1	-	
33	P38/TIP01/TOP01		P38	TIP01				TOP01	x	x	
34	V_{DD1}										
35	REGC1										
36	V_{SS1}										
37	P50/TIQ01/TOQ01		P50	TIQ01				TOQ01	x	x	
38	P51/TIQ02/TOQ02		P51	TIQ02				TOQ02	x	x	
39	P52/TIQ03/TOQ03		P52	TIQ03				TOQ03	x	x	
40	P53/TIQ00/TOQ00		P53	TIQ00				TOQ00	x	x	
41	P54/SI30		P54	SI30				x	x	x	
42	P55/SO30		P55	SO30				x	x	x	
43	P90/TIQ11/TOQ11/ $\overline{SCK30}$		P90	$\overline{SCK30}$		No-Func		TOQ11	TIQ11	-	
44	P91/TIQ12/TOQ12/ CS300		P91	CS300		No-Func		TOQ12	TIQ12	-	
45	P92/TIQ13/TOQ13/ CS301		P92	CS301		No-Func		TOQ13	TIQ13	-	
46	P93/TIQ10/TOQ10/ CS302		P93	CS302		No-Func		TOQ10	TIQ10	-	
47	P94/$\overline{ASCKA1}$/CS303		P94	CS303				$\overline{ASCKA1}$	x	x	
48	P95/RXDA1/($\overline{SCK30}$)		P95	RXDA1		$\overline{SCK30}$		RXDA1	x	x	
49	P96/TXDA1/(CS300)		P96	TXDA1		CS300		TXDA1	x	x	
50	P97/SIB1/{DDI}		P97	SIB1					x	x	DDI
51	P98/SOB1/{DCK}		P98	SOB1					x	x	DCK
52	P99/$\overline{SCKB1}$/{DMS}		P99	$\overline{SCKB1}$					x	x	DMS
53	P910/CS301/{DDO}		P910			CS301		x	x	x	DDO
54	P911/{$\overline{DRST}$}		P911			No-Func		x	x	x	$\overline{DRST}$
55	P912//TIP21/TOP21/ CS302		P912			CS302		TOP21	TIP21	-	
56	P913/TIP20/TOP20/ INTP4/PCL		P913	INTP4				TOP20	TIP20	PCL	
57	P914/TIP10/TOP10/ INTP5/AD14		P914	INTP5				TOP10	TIP10	AD14	

Remarks: 1. x: Don't care.

2. No-Func: No function

Table 4-2: Control Register Setting (3/4)

Pin No.	Pin Name	Control Register	Value								Other
		PMC	0	1				1	1	1	
		PFC	x	0				1	0	1	
		PFCE	x	0				0	1	1	
		PSWAP	xx	00	01	10	11	xx	xx	xx	
58	P915/TIP11/TOP11/INTP6/AD15		P915	INTP6				TOP11	TIP11	AD15	
59	PCS0/$\overline{\text{CS0}}$		PCS0	$\overline{\text{CS0}}$				x	x	x	
60	PCS1/$\overline{\text{CS1}}$		PCS1	$\overline{\text{CS1}}$				x	x	x	
61	PCM0/$\overline{\text{WAIT}}$		PCM0	$\overline{\text{WAIT}}$				x	x	x	
62	PCM1/CLKOUT		PCM1	CLKOUT				x	x	x	
63	PCM2/$\overline{\text{HLDAK}}$		PCM2	$\overline{\text{HLDAK}}$				x	x	x	
64	PCM3/$\overline{\text{HLDRQ}}$		PCM3	$\overline{\text{HLDRQ}}$				x	x	x	
65	PCT0/$\overline{\text{WR0}}$		PCT0	$\overline{\text{WR0}}$				x	x	x	
66	PCT1/$\overline{\text{WR1}}$		PCT1	$\overline{\text{WR1}}$				x	x	x	
67	PCT4/$\overline{\text{RD}}$		PCT4	$\overline{\text{RD}}$				x	x	x	
68	PCT6/ASTB		PCT6	ASTB				x	x	x	
69	BV_{SS}										
70	BV_{DD}										
71	PDL0/AD0		PDL0	AD0				x	x	x	
72	PDL1/AD1		PDL1	AD1				x	x	x	
73	PDL2/AD2		PDL2	AD2				x	x	x	
74	PDL3/AD3		PDL3	AD3				x	x	x	
75	PDL4/AD4		PDL4	AD4				x	x	x	
76	PDL5/AD5/FLMD1		PDL5	AD5				x	x	x	FLMD1
77	PDL6/AD6		PDL6	AD6				x	x	x	
78	PDL7/AD7		PDL7	AD7				x	x	x	
79	PDL8/AD8		PDL8	AD8				x	x	x	
80	PDL9/AD9		PDL9	AD9				x	x	x	
81	PDL10/AD10		PDL10	AD10				x	x	x	
82	PDL11/AD11		PDL11	AD11				x	x	x	
83	PDL12/AD12		PDL12	AD12				x	x	x	
84	PDL13/AD13		PDL13	AD13				x	x	x	
85	P715/ANI15		P715	ANI15				x	x	x	
86	P714/ANI14		P714	ANI14				x	x	x	
87	P713/ANI13		P713	ANI13				x	x	x	
88	P712/ANI12		P712	ANI12				x	x	x	
89	P711/ANI11		P711	ANI11				x	x	x	
90	P710/ANI10		P710	ANI10				x	x	x	
91	P79/ANI9		P79	ANI9				x	x	x	

Remarks: **1.** x: Don’t care.

2. No-Func: No function

Table 4-2: Control Register Setting (4/4)

Pin No.	Pin Name	Control Register	Value								Other
		PMC	0	1				1	1	1	
		PFC	x	0				1	0	1	
		PFCE	x	0				0	1	1	
		PSWAP	xx	00	01	10	11	xx	xx	xx	
92	P78/ANI8		P78	ANI8				x	x	x	
93	P77/ANI7		P77	ANI7				x	x	x	
94	P76/ANI6		P76	ANI6				x	x	x	
95	P75/ANI5		P75	ANI5				x	x	x	
96	P74/ANI4		P74	ANI4				x	x	x	
97	P73/ANI3		P73	ANI3				x	x	x	
98	P72/ANI2		P72	ANI2				x	x	x	
99	P71/ANI1		P71	ANI1				x	x	x	
100	P70/ANI0		P70	ANI0				x	x	x	

Remarks: 1. x: Don’t care.

2. No-Func: No function

Table 4-3: Port Configuration

Item	Configuration
Control register	Port mode register (PMn: n = 0, 1, 3, 4, 5, 7H, 7L, 9, CM, CS, CT, DL)
	Port mode control register (PMCn: n = 0, 1, 3, 4, 5, 7H, 7L, 9, CM, CS, CT, DL)
	Port function control register (PFCn: n = 0, 1, 3, 5, 9)
	Port function control expansion register (PFCEn: n = 3, 5, 9)
	Pull-up resistor option register (PUn: n = 0, 1, 3, 4, 5, 9, CM, CS, CT)
	Pull-down resistor option register (PDn: n = 0, 1, 3, 4, 5, 9, CM, CS, CT)
	External interrupt falling edge specification register (INTFn: n = 0, 1, 3)
	External interrupt rising edge specification register (INTRn: n = 0, 1, 3)
Ports	I/O: 84

4.3.2 Port function swap control register

This device has a special purpose register for swapping port function by setting control bit. The Swap function is implemented to allow flexible configuration.
More detail of the settings by control bit is shown in Table 4-2, “Control Register Setting,” on page 107.

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-2: Port Function Swap Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PSWAP	0	0	0	0	0	0	PSWAP1	PSWAP0	FFFFF87AH	00H
R/W										

PSWAP1	PSWAP0	Function
0	0	See Table 4-2, “Control Register Setting,” on page 107
0	1	
1	0	
1	1	

4.3.3 Port 0

Port 0 is a 7-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 0 functions

- 7bits I/O port
- Port input/output specifiable in 1-bit units by port mode register 0 (PM0)
- Port mode/control mode specifiable in 1-bit units by port mode control register 0 (PMC0)
- Control mode 1/Control mode 2 specifiable in 1-bit units by port function control register 0 (PFC0)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 0 (PU0)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 0 (PD0)
- Valid edge of external interrupts specifiable in 1-bit units by external interrupt falling edge specification register 0 (INTF0) and external interrupt rising edge specification register 0 (INTR0)

See Table 4-8, “Port Type,” on page 170 for alternate functions of Port 0.

(2) Registers

(a) Port register 0 (P0)

Port register 0 (P0) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-3: Port Register 0 (P0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P0	0	P06	P05	P04	P03	P02	P01	P00	FFFFF400H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

P0	Output data control (in output mode) (n = 0 to 6)
0	Outputs 0
1	Outputs 1

Remarks: 1. Input mode: When port 0 (P0) is read, the pin levels at this time are read. During write, the data written to P0 is written. This does not affect the input pins.

Output mode: When port 0 (P0) is read, the value of P0 is read. During write, the value is written to P0 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P0 is read in the input mode. When P0 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 0 (PM0)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-4: Port Mode Register 0 (PM0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFFFF420H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

(c) Port mode control register 0 (PMC0)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Figure 4-5: Port Mode Control Register 0 (PMC0) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC0	0	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00	FFFFF440H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMC06	P06 pin operation mode specification
0	I/O port
1	CS313/INTP3 (Chip select3 output for CSI31 / External interrupt3 input)

PMC05	P05 pin operation mode specification
0	I/O port
1	CS312/INTP2 (Chip select2 output for CSI31 / External interrupt2 input)

PMC04	P04 pin operation mode specification
0	I/O port
1	CS311 (Chip select1 output for CSI31)

Figure 4-5: Port Mode Control Register 0 (PMC0) Format (2/2)

PMC03	P03 pin operation mode specification
0	I/O port
1	CS310 (Chip select0 output for CSI31)

PMC02	P02 pin operation mode specification
0	I/O port
1	$\overline{\text{SCK31}}$ (Serial clock input/output for CSI31)

PMC01	P01 pin operation mode specification
0	I/O port
1	SO31 (Serial output for CSI31)

PMC00	P00 pin operation mode specification
0	I/O port
1	SI31 (Serial input for CSI31)

(d) Port function control register 0 (PFC0)

This is an 8-bit register used to specify control mode 1/control mode 2.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-6: Port Function Control Register 0 (PFC0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFC0	0	PFC06	PFC05	0	0	0	0	0	FFFFF460H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PFC06	Specification of P06 pin operation mode in control mode
0	CS313 (Chip select3 output for CSI31)
1	INTP3 (External interrupt1 input)

PFC05	Specification of P05 pin operation mode in control mode
0	CS312 (Chip select2 output for CSI31)
1	INTP2 (External interrupt2 input)

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

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-7: Pull-up Resistor Option Register 0 (PU0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	FFFFFC40H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU0n	Connection of a pull-up resistor control (n = 0 to 6)
0	Not connected
1	Connected

(f) Pull-down resistor option register 0 (PD0)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-8: Pull-down Resistor Option Register 0 (PD0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD0	0	PD06	PD05	PD04	PD03	PD02	PD01	PD00	FFFFFCA0H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PD00	Connection of a pull-down resistor control
0	Not connected
1	Connected

(g) External interrupt falling edge specification register 0 (INTF0)

This is an 8-bit register used to specify detection of the falling edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF0n = INTR0n bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-9: External Interrupt Falling Edge Specification Register 0 (INTF0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTF0	0	INTF06	INTF05	0	0	0	0	0	FFFFFC00H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-4, “Valid Edge Specification,” on page 117.

(h) External interrupt rising edge specification register 0 (INTR0)

This is an 8-bit register used to specify detection of the rising edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF0n = INTR0n bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-10: External Interrupt Rising Edge Specification Register 0 (INTR0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTR0	0	INTR06	INTR05	0	0	0	0	0	FFFFFC20H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-4, “Valid Edge Specification,” on page 117.

Table 4-4: Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification
0	0	No edge specified
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark: n = 5: INTP2
n = 6: INTP3

4.3.4 Port 1

Port 1 is a 3-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 1 functions

- 3-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 1 (PM1)
- Port mode/control mode specifiable in 1-bit units by port mode control register 1 (PMC1)
- Control mode 1, 2/control mode 3 specifiable in 1-bit units by port function control register 1 (PFC1, PFCE1)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 1 (PU1)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 1 (PD1)
- Valid edge of external interrupts specifiable in 1-bit units by external interrupt falling edge specification register 0 (INTF1) and external interrupt rising edge specification register 0 (INTR1).

See Table 4-8, “Port Type,” on page 170 for alternate functions.

(2) Registers

(a) Port register 1 (P1)

Port register 1 (P1) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-11: Port Register 1 (P1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P1	0	0	0	0	0	P12	P11	P10	FFFFF402H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

P1n	Output data control (in output mode) (n = 0 to 2)
0	Outputs 0
1	Outputs 1

Remarks: 1. Input mode: When port 1 (P1) is read, the pin levels at this time are read. During write, the data written to P1 is written. This does not affect the input pins.

Output mode: When port 1 (P1) is read, the value of P1 is read. During write, the value is written to P1 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P1 is read in the input mode. When P1 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 1 (PM1)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-12: Port Mode Register 1 (PM1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM1	1	1	1	1	1	PM12	PM11	PM10	FFFFF422H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PM1n	I/O mode control (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port mode control register 1 (PMC1)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Figure 4-13: Port Mode Control Register 1 (PMC1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC1	0	0	0	0	0	PMC12	PMC11	PMC10	FFFFF442H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMC12	P12 pin operation mode specification
0	I/O port
1	TIP30/TOP30/ADTRG (TMP3 input0 / TMP3 output0 / ADC external trigger input)

PMC11	P11 pin operation mode specification
0	I/O port
1	TIP31/TOP31 (TMP3 input1 / TMP3 output1)

PMC10	P10 pin operation mode specification
0	I/O port
1	INTP0 (External interrupt0 input)

(d) Port function control register 1 (PFC1)

This is an 8-bit register used to specify control mode 1, 2 and 3 with PFCE1.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-14: Port Function Control Register 1 (PFC1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFC1	0	0	0	0	0	PFC12	PFC11	0	FFFFF462H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification refer to **4.3.4 (2) (f) P1 pin control mode settings**.

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

This is an 8-bit register used to specify control mode 1,2 and 3 with PFC1.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-15: Port Function Control Extended Register 1 (PFCE1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFCE1	0	0	0	0	0	PFCE12	0	0	FFFFF702H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification refer to **4.3.4 (2) (f) P1 pin control mode settings**.

(f) P1 pin control mode settings

PFC11	Specification of P11 pin operation mode in control mode
0	TIP31 (TMP3 input 1)
1	TOP31 (TMP3 output 1)

PFCE12	PFC12	Specification of P12 pin operation mode in control mode
0	0	TIP30 (TMP3 input 0)
0	1	TOP30 (TMP3 output 0)
1	0	ADTRG (ADC external trigger input)
1	1	Setting prohibited

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

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-16: Pull-up Resistor Option Register 1 (PU1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU1	0	0	0	0	0	PU12	PU11	PU10	FFFFFC42H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU1n	Connection of a pull-up resistor control (n = 0 to 2)
0	Not connected
1	Connected

(h) Pull-down resistor option register 1 (PD1)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-17: Pull-down Resistor Option Register 1 (PD1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD1	0	0	0	0	0	PD12	PD11	PD10	FFFFFCA2H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PD1n	Connection of a pull-down resistor control (n = 0 to 2)
0	Not connected
1	Connected

(i) External interrupt falling edge specification register 1 (INTF1)

This is an 8-bit register used to specify detection of the falling edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF1n = INTR1n bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-18: External Interrupt Falling Edge Specification Register 1 (INTF1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTF1	0	0	0	0	0	0	0	INTF10	FFFFFC02H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to **Table 4-5, “Valid Edge Specification,” on page 123**.

(j) External interrupt rising edge specification register 1 (INTR1)

This is an 8-bit register used to specify detection of the rising edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF10 = INTR10 bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-19: External Interrupt Rising Edge Specification Register 1 (INTR1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTR1	0	0	0	0	0	0	0	INTR10	FFFFFC22H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to **Table 4-5, “Valid Edge Specification,” on page 123**.

Table 4-5: Valid Edge Specification

INTF10	INTR10	Valid Edge Specification
0	0	No edge specified
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

4.3.5 Port 3

Port 3 is a 9-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 3 functions

- 9-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 3 (PM3)
- Port mode/control mode specifiable in 1-bit units by port mode control register 3 (PMC3)
- Control mode 1,2/control mode 3 specifiable in 1-bit units by port function control register 3 (PFC3, PFCE3)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 3 (PU3)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 3 (PD3)
- Valid edge of external interrupts specifiable in 1-bit units by external interrupt falling edge specification register 3 (INTF3) and external interrupt rising edge specification register 3 (INTR3).

See **Table 4-8, “Port Type,” on page 170** for alternate functions.

(2) Registers

(a) Port register 3 (P3)

Port register 3 (P3) is a 16-bit register used to control pin level read and output level write.
This register can be read and written in16-bit, 8-bit and 1-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 becomes two 8-bit registers for which I/O can be manipulated in 8-bit or 1-bit units.

Figure 4-20: Port Register 3 (P3) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
P3H	0	0	0	0	0	0	0	P38	FFFFF407H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P3L	P37	P36	P35	P34	P33	P32	P31	P30	FFFFF406H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Remarks: **1.** Input mode: When port 3 (P3) is read, the pin levels at this time are read. During write, the data written to P3 is written. This does not affect the input pins.

Output mode: When port 3 (P3) is read, the value of P3 is read. During write, the value is written to P3 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P3 is read in the input mode. When P3 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 3 (PM3)

This is a 16-bit register used to specify the input mode/output mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-21: Port Mode Register 3 (PM3) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PM3H	1	1	1	1	1	1	1	PM38	FFFFF427H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM3L	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFFFF426H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

(c) Port mode control register 3 (PMC3)

This is a 16-bit register used to specify the port mode/control mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-22: Port Mode Control Register 3 (PMC3) Format (1/2)

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PMC3H	0	0	0	0	0	0	0	PMC38	FFFFF447H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC3L	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF446H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMC38	P38 pin operation mode specification
0	I/O port
1	TIP01/TOP01 (TMP0 input1 / TMP0 output1)

PMC37	P37 pin operation mode specification
0	I/O port
1	TIP00/TOP00/INTP1/(CS312) (TMP0 input0 / TMP0 output0 / External interrupt 1 / Chip select2 for CSI31)

PMC36	P36 pin operation mode specification
0	I/O port
1	CTXD1 (Serial transmit data output for aFCAN1)

PMC35	P35 pin operation mode specification
0	I/O port
1	CRXD1 (Serial receive data input for aFCAN1)

PMC34	P34 pin operation mode specification
0	I/O port
1	CRXD0 (Serial receive data input for aFCAN0)

Figure 4-22: Port Mode Control Register 3 (PMC3) Format (2/2)

PMC33	P33 pin operation mode specification
0	I/O port
1	CTXD0 (Serial transmit data output for aFCAN0)

PMC32	P32 pin operation mode specification
0	I/O port
1	$\overline{\text{ASCKA0}}$ /[TXDA0]/(CS312) (Serial clock input for UARTA0 / Serial output for UARTA0 / Chip select0 for CSI31)

PMC31	P31 pin operation mode specification
0	I/O port
1	RXDA0/INTP7 (Serial input for UARTA0 / External interrupt 7 input)

PMC30	P30 pin operation mode specification
0	I/O port
1	TXDA0/[$\overline{\text{ASCKA0}}$] (Serial output for UARTA0 / Serial clock input for UARTA0)

(d) Port function control register 3 (PFC3)

This is a 16/8-bit register used to specify control mode 1/control mode 2.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-23: Port Function Control Register 3 (PFC3) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PFC3H	0	0	0	0	0	0	0	PFC38	FFFFF467H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFC3L	PFC37	0	0	0	0	0	PFC31	0	FFFFF466H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification refer to **(f) P3 pin control mode settings**.

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

This is an 8-bit register used to specify control mode 1, 2 / control mode 3.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-24: Port Function Control Expansion Register 3 (PFCE3) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFCE3	PFCE37	0	0	0	0	0	0	0	FFFFF706H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification refer to **(f) P3 pin control mode settings**.

(f) P3 pin control mode settings

PFC38	P38 Pin Control Mode Specification
0	TIP01 input (TMP0 input1)
1	TOP01 output (TMP0 output1)

PFCE37	PFC37	P37 Pin Control Mode Specification
0	0	TIP00/CS312 (TMP0 input0 / Chip select2 for CSI31)
0	1	TOP00 output (TMP0 output 0)
1	0	INTP1 (External interrupt 1)
1	1	Setting prohibited

PFC31	P31 Pin Control Mode Specification
0	RXDA0 (Serial input for UARTA0)
1	INTP7 (External Interrupt7)

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

This is a 16-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-25: Pull-up Resistor Option Register 3 (PU3) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PU3H	0	0	0	0	0	0	0	PU38	FFFFFC47H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU3L	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	FFFFFC46H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU3n	Connection of a pull-up resistor control (n = 0 to 8)
0	Not connected
1	Connected

(h) Pull-down resistor option register 3 (PD3)

This is a 16-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written only in 16-bit units.
However, when using the higher 8 bits of the PD3 register as the PD3H register and the lower 8 bits as the PD3L register, PD3 can be read and written in 8-bit and 1-bit units.

Figure 4-26: Pull-down Resistor Option Register 3 (PD3) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PD3H	0	0	0	0	0	0	0	PD38	FFFFFCA7H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD3L	PD37	PD36	PD35	PD34	PD33	PD32	PD31	PD30	FFFFFCA6H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

(i) External interrupt falling edge specification register 3 (INTF3)

This is an 8-bit register used to specify detection of the falling edge for the external interrupt pin. This register can be read and written only in 16-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF3 = INTR3 bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-27: External Interrupt Falling Edge Specification Register 3 (INTF3) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTF3	INTF37	0	0	0	0	0	INTF31	0	FFFFFC06H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-6.

(j) External interrupt rising edge specification register 3 (INTR3)

This is an 8-bit register used to specify detection of the rising edge for the external interrupt pin. This register can be read and written only in 16-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF3 = INTR3 bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-28: External Interrupt Rising Edge Specification Register 3 (INTR3) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTR3	INTR37	0	0	0	0	0	INTR31	0	FFFFFC26H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-6.

Table 4-6: Valid Edge Specification

INTF3n	INTR3n	Valid Edge Specification (n = 1, 7)
0	0	No edge specified
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

4.3.6 Port 4

Port 4 is a 3-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 4 functions

- 3-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 4 (PM4)
- Port mode/control mode specifiable in 1-bit units by port mode control register 4 (PMC4)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 4 (PU4)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 4 (PD4)
- N-ch Open drain specifiable in 1-bit units by port function register 4 (PF4).

See **Table 4-8, "Port Type," on page 170** for alternate functions.

(2) Registers

(a) Port register 4 (P4)

Port register 4 (P4) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-29: Port Register 4 (P4) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P4	0	0	0	0	0	P42	P41	P40	FFFFF408H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Remarks: 1. Input mode: When port 4 (P4) is read, the pin levels at this time are read. During write, the data written to P4 is written. This does not affect the input pins.

Output mode: When port 4 (P4) is read, the value of P4 is read. During write, the value is written to P4 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P4 is read in the input mode. When P4 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 4 (PM4)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-30: Port Mode Register 4 (PM4) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM4	1	1	1	1	1	PM42	PM41	PM40	FFFFF428H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

(c) Port mode control register 4 (PMC4)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Figure 4-31: Port Mode Control Register 4 (PMC4) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40	FFFFF448H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMC42	P42 pin operation mode specification
0	I/O port
1	$\overline{\text{SCKB0}}$ Input (Serial clock input/output for CSIB0)

PMC41	P41 pin operation mode specification
0	I/O port
1	SOB0 output (Serial output for CSIB0)

PMC40	P40 pin operation mode specification
0	I/O port
1	SIB0 input (Serial input for CSIB0)

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

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-32: Pull-up Resistor Option Register 4 (PU4) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU4	0	0	0	0	0	PU42	PU41	PU40	FFFFFC48H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU4n	Connection of a pull-up resistor control (n = 0 to 2)
0	Not connected
1	Connected

(e) Pull-down resistor option register 4 (PD4)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-33: Pull-down Resistor Option Register 4 (PD4) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD4	0	0	0	0	0	PD42	PD41	PD40	FFFFFCA8H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PD4n	Connection of a pull-down resistor control (n = 0 to 2)
0	Not connected
1	Connected

4.3.7 Port 5

Port 5 is a 6-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 5 functions

- 6-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 5 (PM5)
- Port mode/control mode specifiable in 1-bit units by port mode control register 5 (PMC5)
- Control mode 1/control mode 2 specifiable in 1-bit units by port function control register 5 (PFC5)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 5 (PU5)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 5 (PD5).

See **Table 4-8, “Port Type,” on page 170** for alternate functions.

(2) Registers

(a) Port register 5 (P5)

Port register 5 (P5) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-34: Port Register 5 (P5) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P5	0	0	P55	P54	P53	P52	P51	P50	FFFFF40AH	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

P5n	Output data control (in output mode) (n = 0 to 5)
0	Outputs 0
1	Outputs 1

Remarks: 1. Input mode: When port 5 (P5) is read, the pin levels at this time are read. During write, the data written to P5 is written. This does not affect the input pins.

Output mode: When port 5 (P5) is read, the value of P5 is read. During write, the value is written to P5 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P5 is read in the input mode. When P5 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 5 (PM5)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-35: Port Mode Register 5 (PM5) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFFFF42AH	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

(c) Port mode control register 5 (PMC5)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Figure 4-36: Port Mode Control Register 5 (PMC5) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC5	0	0	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50	FFFFF44AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMC55	P55 pin operation mode specification
0	I/O port
1	SO30 (Serial output for CSI30)

PMC54	P54 pin operation mode specification
0	I/O port
1	SI30 (Serial input for CSI30)

PMC53	P53 pin operation mode specification
0	I/O port
1	TIQ00/TOQ00 (TMQ0 input0 / TMQ0 output0)

Figure 4-36: Port Mode Control Register 5 (PMC5) Format (2/2)

PMC52	P52 pin operation mode specification
0	I/O port
1	TIQ03/TOQ03 (TMQ0 input3 / TMQ0 output3)

PMC51	P51 pin operation mode specification
0	I/O port
1	TIQ02/TOQ02 (TMQ0 input2 / TMQ0 output2)

PMC50	P50 pin operation mode specification
0	I/O port
1	TIQ01/TOQ01 (TMQ0 input1 / TMQ0 output1)

(d) Port function control register 5 (PFC5)

This is an 8-bit register used to specify control mode 1/control mode 2.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-37: Port Function Control Register 5 (PFC5) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFC5	0	0	0	0	PFC53	PFC52	PFC51	PFC50	FFFFF46AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification,; refer to **(e) P5 pin control mode settings.**

(e) P5 pin control mode settings

PFC53	P53 Pin Control Mode Specification
0	TIQ00 input (TMQ0 input0)
1	TOQ00 output (TMQ0 output0)

PFC52	P52 Pin Control Mode Specification
0	TIQ03 input (TMQ0 input3)
1	TOQ03 output (TMQ0 output3)

PFC51	P51 Pin Control Mode Specification
0	TIQ02 input (TMQ0 input2)
1	TOQ02 output (TMQ0 output2)

PFC50	P50 Pin Control Mode Specification
0	TIQ01 input (TMQ0 input1)
1	TOQ01 output (TMQ0 output1)

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

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-38: Pull-up Resistor Option Register 5 (PU5) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	FFFFFC4AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU5n	Connection of a pull-up resistor control (n = 0 to 5)
0	Not connected
1	Connected

(g) Pull-down resistor option register 5 (PD5)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-39: Pull-down Resistor Option Register 5 (PD5) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD5	0	0	PD55	PD54	PD53	PD52	PD51	PD50	FFFFFCAAH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PD5n	Connection of a pull-down resistor control (n = 0 to 5)
0	Not connected
1	Connected

4.3.8 Port 7

Port 7 is a 16-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 7 functions

- 16-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 7 (PM7)

See **Table 4-8, “Port Type,” on page 170** for alternate functions.

(2) Registers

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

Port register 7H and port register 7L (P7H, P7L) are 8-bit registers used to control pin level read and output level write. These registers can be read and written in 8-bit or 1-bit units.
16-bit access is not possible.

Figure 4-40: Port Register 7H, Port Register 7L (P7H, P7L)Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P7H	P715	P714	P713	P712	P711	P710	P79	P78	FFFFF40FH	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P7L	P77	P76	P75	P74	P73	P72	P71	P70	FFFFF40EH	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Caution: Do not read the P7H and P7L registers during A/D conversion.

Remarks: 1. Input mode: When port 7H (P7H) and port 7L (P7L) are read, the pin levels at this are read. During write, the data written to P7H and P7L are written. This does not affect the input pins.

Output mode: When port 7H (P7H) and port 7L (P7L) are read, the values of P7H and P7L are read. During write, the values are written to P7H and P7L and the written values are immediately output.

2. An undefined value (pin input level) is read for the value after reset when P7H and P7L are read in the input mode. When P7H and P7L are read in the output mode, 00H (output latch value) is output.

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

These are 8-bit registers used to specify the input mode/output mode.
These registers can be read and written in 8-bit or 1-bit units.
16-bit access is not possible.

Figure 4-41: Port Mode Register 7H, Port Mode Register 7L (PM7H, PM7L)Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM7H	PM715	PM714	PM713	PM712	PM711	PM710	PM79	PM78	FFFFF42FH	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFFFF42EH	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Caution: When using P70 to P715 as alternate functions (ANI0 to ANI15), set PM7H register and PM7L register = FFH.

4.3.9 Port 9

Port 9 is a 16-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port 9 functions

- 16-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register 9 (PM9)
- Port mode/control mode specifiable in 1-bit units by port mode control register 9 (PMC9)
- Control mode 1, 2, 3 and 4 specifiable in 1-bit units by port function control register 9 (PFC9, PFCE9)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register 9 (PU9)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register 9 (PD9)
- Valid edge of external interrupts (alternate function) specifiable in 1-bit units.
  Specification done with external interrupt falling edge specification register 9H (INTF9H) and external interrupt rising edge specification register 9H (INTR9H).

See **Table 4-8, “Port Type,” on page 170** for alternate functions.

(2) Registers

(a) Port register 9 (P9)

Port register 9 (P9) is a 16-bit register used to control pin level read and output level write.
This register can be read and written in 8-bit and 1-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 becomes two 8-bit registers for which I/O can be manipulated in 8-bit or 1-bit units.

Figure 4-42: Port Register 9 (P9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
P9H	P915	P914	P913	P912	P911	P910	P99	P98	FFFFF413H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
P9L	P97	P96	P95	P94	P93	P92	P91	P90	FFFFF412H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Remarks: 1. Input mode: When port 9 (P9) is read, the pin levels at this time are read. During write, the data written to P9 is written. This does not affect the input pins.

Output mode: When port 9 (P9) is read, the value of P9 is read. During write, the value is written to P9 and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when P9 is read in the input mode. When P9 is read in the output mode, 00H (output latch value) is output.

(b) Port mode register 9 (PM9)

This is a 16-bit register used to specify the input mode/output mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-43: Port Mode Register 9 (PM9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PM9H	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98	FFFFF433H	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM9L	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFFFF432H	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

(c) Port mode control register 9 (PMC9)

This is a 16-bit register used to specify the port mode/control mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-44: Port Mode Control Register 9 (PMC9) Format (1/3)

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PMC9H	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98	FFFFF453H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMC9L	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90	FFFFF452H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Figure 4-44: Port Mode Control Register 9 (PMC9) Format (2/3)

PMC915	P915 pin operation mode specification
0	I/O port
1	TIP11/TOP11/INTP6/AD15 [Note] (TMP1 input1 / TMP1 output1 / External interrupt input6 / Address/Data bus I/O[Note])

PMC914	P914 pin operation mode specification
0	I/O port
1	TIP10/TOP10/INTP5/AD14[Note] (TMP1 input0 / TMP1 output0 / External interrupt input5 / Address/Data bus I/O[Note])

PMC913	P913 pin operation mode specification
0	I/O port
1	TIP20/TOP20/INTP4/PCL (TMP2 input0 / TMP2 output0 / External interrupt input4 Programmable clock output)

PMC912	P912 pin operation mode specification
0	I/O port
1	TIP21/TOP21/CS302 (TMP2 input1 / TMP2 output1 / Chip select2 output forCSI30)

PMC911	P911 pin operation mode specification
0	I/O port
1	[$\overline{\text{DRST}}$] (OCD mode reset input)

PMC910	P910 pin operation mode specification
0	I/O port
1	(CS301)/[DDO] (Chip select1 for CSI30/OCD mode data output)

PMC99	P99 pin operation mode specification
0	I/O port
1	$\overline{\text{SCKB1}}$/DMS (Serial clock input/output for CSIB1 /OCD mode set input)

PMC98	P98 pin operation mode specification
0	I/O port
1	SOB1/[DCK] (Serial output for CSIB1 / OCD mode clock input)

PMC97	P97 pin operation mode specification
0	I/O port
1	SIB1/[DDI] (Serial input for CSIB1 / OCD mode data input)

Note: AD14, AD15 and Address/Data bus I/O are only for the µPD70F3403 and µPD70F3403A

Figure 4-44: Port Mode Control Register 9 (PMC9) Format (3/3)

PMC96	P96 pin operation mode specification
0	I/O port
1	TXDA1/(CS300) (Serial output for UARTA1 / Chip select 0 for CSI30)

PMC95	P95 pin operation mode specification
0	I/O port
1	RXDA1/($\overline{\text{SCK30}}$) (Serial input for UARTA1 / Serial clock input/output for CSI30)

PMC94	P94 pin operation mode specification
0	I/O port
1	$\overline{\text{ASCKA1}}$ /CS303 (Serial clock input for UARTA1 / Chip select3 for CSI30)

PMC93	P93 pin operation mode specification
0	I/O port
1	TIQ10/TOQ10/CS302 (TMQ1 input0 / TMQ1 output0 /Chip select2 for CSI30)

PMC92	P92 pin operation mode specification
0	I/O port
1	IQ13/TOQ13/CS301 (TMQ1 input3 / TMQ1 output3 / Chip select1 for CSI30)

PMC91	P91 pin operation mode specification
0	I/O port
1	TIQ12/TOQ12/CS300 (TMQ1 input2 / TMQ1 output2 / Chip select0 for CSI30)

PMC90	P90 pin operation mode specification
0	I/O port
1	IQ11/TOQ11/$\overline{\text{SCK30}}$ (TMQ1 input1 / TMQ1 output1 / Clock input/output for CSI30)

(d) Port function control register 9 (PFC9)

This is a 16-bit register used to specify control mode 1, 2, 3 and 4 with PFCE9.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-45: Port Function Control Register 9 (PFC9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PFC9H	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98	FFFFF473H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFC9L	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90	FFFFF472H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification,; refer to **Table f, “P9 pin control mode settings,” on page 149**.

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

This is a 16-bit register used to specify control mode 1, 2, 3 and 4 with PFC9.
This register can be read and written only in 16-bit units.
However, when using the higher 8 bits of the PFCE9 register as the PFCE9H register and the lower 8 bits as the PFCE9L register, PFCE9 can be read and written in 8-bit and 1-bit units.

Figure 4-46: Port Function Control Expansion Register 9 (PFCE9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PFCE9H	PFCE915	PFCE914	PFCE913	PFCE912	0	0	0	0	FFFFF713H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PFCE9L	0	0	0	0	PFCE93	PFCE92	PFCE91	PFCE90	FFFFF712H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on control mode specification,; refer to **(f) 'P9 pin control mode settings' on page 149**.

(f) P9 pin control mode settings

PFCE915	PFC915	P915 Pin Control Mode Specification
0	0	INTP6
0	1	TOP11
1	0	TIP11
1	1	AD15[Note]

PFCE914	PFC914	P914 Pin Control Mode Specification
0	0	INTP5
0	1	TOP10
1	0	TIP10
1	1	AD14[Note]

Note: AD14 and AD15 are only for the µPD70F3403 and µPD70F3403A

PFCE913	PFC913	P913 Pin Control Mode Specification
0	0	INTP4
0	1	TOP20
1	0	TIP20
1	1	PCL

PFCE912	PFC912	P912 Pin Control Mode Specification
0	0	CS302
0	1	TOP21
1	0	TIP21
1	1	Setting prohibited

PFC99	P99 Pin Control Mode Specification
0	$\overline{\text{SCKB1}}$
1	[DMS]

PFC98	P98 Pin Control Mode Specification
0	SOB1
1	[DCK]

PFC97	P97 Pin Control Mode Specification
0	SIB1
1	[DDI]

PFC96	P96 Pin Control Mode Specification
0	TxDA1/(CS300)
1	TxDA1

PFC95	P95 Pin Control Mode Specification
0	RxDA1/($\overline{\text{SCK30}}$)
1	RxDA1

PFC94	P95 Pin Control Mode Specification
0	CS303
1	$\overline{\text{ASCKA1}}$

PFCE93	PFC93	P93 Pin Control Mode Specification
0	0	CS302
0	1	TOQ10
1	0	TIQ10
1	1	Setting prohibited

PFCE92	PFC92	P92 Pin Control Mode Specification
0	0	CS301
0	1	TOQ13
1	0	TIQ13
1	1	Setting prohibited

PFCE91	PFC91	P91 Pin Control Mode Specification
0	0	CS300
0	1	TOQ12
1	0	TIQ12
1	1	Setting prohibited

PFCE90	PFC90	P90 Pin Control Mode Specification
0	0	$\overline{\text{SCK30}}$
0	1	TOQ11
1	0	TIQ11
1	1	Setting prohibited

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

This is a 16-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-47: Pull-up Resistor Option Register 9 (PU9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PU9H	PU915	PU914	PU913	PU912	0	PU910	PU99	PU98	FFFFFC53H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PU9L	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	FFFFFC52H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PU9n	Connection of a pull-up resistor control (n = 0 to 15)
0	Not connected
1	Connected

(h) Pull-down resistor option register 9 (PD9)

This is a 16-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written only in 16-bit units.
However, when using the higher 8 bits of the PD9 register as the PD9H register and the lower 8 bits as the PUD9L register, PD9 can be read and written in 8-bit and 1-bit units.

Figure 4-48: Pull-down Resistor Option Register 9 (PD9) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PD9H	PD915	PD914	PD913	PD912	PD911[Note]	0	PD99	PD98	FFFFFCB3H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PD9L	PD97	PD96	PD95	PD94	PD93	PD92	PD91	PD90	FFFFFCB2H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PD9n	Connection of a pull-down resistor control (n = 0 to 15)
0	Not connected
1	Connected

Note: 0 (Default). While OCDM = 01H the built-in pull-down function is enabled for P911, but this state is not indicated in bit PD911 at that time. Refer to **Chapter 22 'On-Chip Debug Function' on page 767**.

(i) External interrupt falling edge specification register 9H (INTF9H)

This is an 8-bit register used to specify detection of the falling edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF9n = INTR9n bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-49: External Interrupt Falling Edge Specification Register 9H (INTF9H) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0	FFFFFC13H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-7, “Valid Edge Specification,” on page 154.

(j) External interrupt rising edge specification register 9H (INTR9H)

This is an 8-bit register used to specify detection of the rising edge for the external interrupt pin. This register can be read and written in 8-bit or 1-bit units.

Cautions: 1. When switching from the external interrupt function (alternate function) to the port function, edge detection may occur. Therefore, set the port mode after setting INTF9n = INTR9n bit = 0.

2. An on-chip circuit for eliminating noise through analog delay is provided for external interrupt input.

Figure 4-50: External Interrupt Rising Edge Specification Register 9H (INTR9H) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0	FFFFFC33H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Remark: For details on valid edge specification, refer to Table 4-7, “Valid Edge Specification,” on page 154.

Table 4-7: Valid Edge Specification

INTF9n	INTR9n	Valid Edge Specification (n = 13 to 15)
0	0	No edge specified
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark: n = 13 to 15: Control of INTP4 to INTP6 pins

4.3.10 Port CM

Port CM is a 4-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port CM functions

- 4-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register CM (PMCM)
- Port mode/control mode specifiable in 1-bit units by port mode control register CM (PMCCM)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register CM (PUCM)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register CM (PDCM)

See **Table 4-8, “Port Type,” on page 170** for alternate functions.

(2) Registers

(a) Port register CM (PCM)

Port register CM (PCM) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-51: Port Register CM (PCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PCM	0	0	0[Note]	0[Note]	PCM3	PCM2	PCM1	PCM0	FFFFF00CH	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Note: PCM4, PCM5: please set 0 any time.

Remarks: 1. Input mode: When port CM (PCM) is read, the pin levels at this time are read. During write, the data written to PCM is written. This does not affect the input pins.

Output mode: When port CM (PCM) is read, the value of PCM is read. During write the value is written to PCM and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when PCM is read in the input mode. When PCM is read in the output mode, 00H (output latch value) is output.

(b) Port mode register CM (PMCM)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-52: Port Mode Register CM (PMCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCM	1	1	0[Note]	0[Note]	PMCM3	PMCM2	PMCM1	PMCM0	FFFFF02CH	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Note: PMCM4, PMCM5: please set 0 any time.

(c) Port mode control register CM (PMCCM)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Remark: External bus functions are only valid for the µPD70F3403 and µPD70F3403A

Figure 4-53: Port Mode Control Register CM (PMCCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCCM	0	0	0$^{\text{Note}}$	0$^{\text{Note}}$	PMCCM3	PMCCM2	PMCCM1	PMCCM0	FFFFF04CH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMCCM3	PCM3 pin operation mode specification
0	I/O port
1	$\overline{\text{HLDRQ}}$ input (Bus hold request signal input)

PMCCM2	PCM2 pin operation mode specification
0	I/O port
1	$\overline{\text{HLDAK}}$ output (Bus hold acknowledge signal output)

PMCCM1	PCM1 pin operation mode specification
0	I/O port
1	CLKOUT output (External bus base clock output)

PMCCM0	PCM0 pin operation mode specification
0	I/O port
1	$\overline{\text{WAIT}}$ input (External wait input for bus cycle)

Note: PMCCM4, PMCCM5: please set 0 any time.

(d) Pull-up resistor option register CM (PUCM)

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-54: Pull-up Resistor Option Register CM (PUCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PUCM	0	0	0	0	PUCM3	PUCM2	PUCM1	PUCM0	FFFFFA4CH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PUCMn	Connection of a pull-up resistor control (n = 0 to 3)
0	Not connected
1	Connected

(e) Pull-down resistor option register CM (PDCM)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-55: Pull-down Resistor Option Register CM (PDCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PDCM	0	0	0	0	PDCM3	PDCM2	PDCM1	PDCM0	FFFFFA6CH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PDCMn	Connection of a pull-down resistor control (n = 0 to 3)
0	Not connected
1	Connected

4.3.11 Port CS

Port CS is a 2-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port CS functions

- 2-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register CS (PMCS)
- Port mode/control mode specifiable in 1-bit units by port mode control register CS (PMCCS)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register CS (PUCS)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register CS (PDCS)

See **Table 4-8, "Port Type," on page 170** for alternate functions.

(2) Registers

(a) Port register CS (PCS)

Port register CS (PCS) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-56: Port Register CS (PCS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PCS	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	PCS1	PCS0	FFFFF008H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Note: PCS7 to PCS2: please set 0 any time.

Remarks: 1. Input mode: When port CS (PCS) is read, the pin levels at this time are read. During write, the data written to PCS is written. This does not affect the input pins.

Output mode: When port CS (PCS) is read, the value of PCS is read. During write, the value is written to PCS and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when PCS is read in the input mode. When PCS is read in the output mode, 00H (output latch value) is output.

(b) Port mode register CS (PMCS)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-57: Port Mode Register CS (PMCS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCS	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	PMCS1	PMCS0	FFFFF028H	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Note: PCMS7 to PMCS2: please set 0 any time.

(c) Port mode control register CS (PMCCS)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Remark: External bus functions are only valid for the µPD70F3403 and µPD70F3403A

Figure 4-58: Port Mode Control Register CS (PMCCS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCCS	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	0[Note]	PMCCS1	PMMCS0	FFFFF048H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMCCS1	PCS1 pin operation mode specification
0	I/O port
1	$\overline{\text{CS1}}$ output (Chip select1 signal output)

PMCCS0	PCS0 pin operation mode specification
0	I/O port
1	$\overline{\text{CS0}}$ output (Chip select0 signal output)

Note: PMCCS7 to PMCCS2: please set 0 any time.

(d) Pull-up resistor option register CS (PUCS)

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-59: Pull-up Resistor Option Register CS (PUCS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PUCS	0	0	0	0	0	0	PUCS1	PUCS0	FFFFFA48H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PUCSn	Connection of a pull-up resistor control (n = 0, 1)
0	Not connected
1	Connected

(e) Pull-down resistor option register CS (PDCS)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-60: Pull-down Resistor Option Register CS (PDCS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PDCS	0	0	0	0	0	0	PDCS1	PDCS0	FFFFFA68H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PDCSn	Connection of a pull-down resistor control (n = 0, 1)
0	Not connected
1	Connected

4.3.12 Port CT

Port CT is a 4-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port CT functions

- 4-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register CT (PMCT)
- Port mode/control mode specifiable in 1-bit units by port mode control register CT (PMCCT)
- On-chip pull-up resistor specifiable in 1-bit units by pull-up resistor option register CT (PUCT)
- On-chip pull-down resistor specifiable in 1-bit units by pull-down resistor option register CT (PDCT)

See **Table 4-8, "Port Type," on page 170** for alternate functions.

(2) Registers

(a) Port register CT (PCT)

Port register CT (PCT) is an 8-bit register that controls pin level read, output level write. It can be read and written in 8-bit or 1-bit units.

Figure 4-61: Port Register CT (PCT) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PCT	0[Note]	PCT6	0[Note]	PCT4	0[Note]	0[Note]	PCT1	PCT0	FFFFF00AH	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

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

Note: PCT7, PCT5, PCT3, PCT2: please set 0 any time.

Remarks: 1. Input mode: When port CT (PCT) is read, the pin levels at this time are read. During write, the data written to PCT is written. This does not affect the input pins.

Output mode: When port CT (PCT) is read, the value of PCT is read. During write, the

value is written to PCT and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when PCT is read in the input mode. When PCT is read in the output mode, 00H (output latch value) is output.

(b) Port mode register CT (PMCT)

This is an 8-bit register used to specify the input mode/output mode.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-62: Port Mode Register CT (PMCT) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCT	0[Note]	PMCT6	0[Note]	PMCT4	0[Note]	0[Note]	PMCT1	PMCT0	FFFFF02AH	FFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMCTn	I/O mode control (n = 0, 1, 4, 6)
0	Output mode
1	Input mode

Note: PMCT7, PMCT5, PMCT3, PMCT2: please set 0 any time.

(c) Port mode control register CT (PMCCT)

This is an 8-bit register used to specify the port mode/control mode.
It can be read and written in 8-bit or 1-bit units.

Remark: External bus functions are only valid for the µPD70F3403 and µPD70F3403A

Figure 4-63: Port Mode Control Register CT (PMCCT) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCCT	0[Note]	PMCCT6	0[Note]	PMCCT4	0[Note]	0[Note]	PMCCT1	PMCCT0	FFFFF04AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMCCT6	PCT6 pin operation mode specification
0	I/O port
1	ASTB output (Address strobe output)

PMCCT4	PCT4 pin operation mode specification
0	I/O port
1	$\overline{\text{RD}}$ output (Read strobe output)

PMCCT1	PCT1 pin operation mode specification
0	I/O port
1	$\overline{\text{WR1}}$ output (Higher byte Write strobe output)

PMCCT0	PCT0 pin operation mode specification
0	I/O port
1	$\overline{\text{WR0}}$ output (Lower byte Write strobe output)

Note: PMCCT7, PMCCT5, PMCCT3, PMCCT2: please set 0 any time.

(d) Pull-up resistor option register CT (PUCT)

This is an 8-bit register used to set whether to use an on-chip pull-up resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-64: Pull-up Resistor Option Register CT (PUCT) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PUCT	0	PUCT6	0	PUCT4	0	0	PUCT1	PUCT0	FFFFFA4AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PUCTn	Connection of a pull-up resistor control (n = 0, 1, 4, 6)
0	Not connected
1	Connected

(e) Pull-down resistor option register CT (PDCT)

This is an 8-bit register used to set whether to use an on-chip pull-down resistor.
This register can be read and written in 8-bit or 1-bit units.

Figure 4-65: Pull-down Resistor Option Register CT (PDCT) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PDCT	0	PDCT6	0	PDCT4	0	0	PDCT1	PDCT0	FFFFFA6AH	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PDCTn	Connection of a pull-down resistor control (n = 0, 1, 4, 6)
0	Not connected
1	Connected

4.3.13 Port DL

Port DL is a 14-bit port for which I/O settings can be controlled in 1-bit units.

(1) Port DL functions

- 14-bit I/O port
- Port input/output specifiable in 1-bit units by port mode register DL (PMDL)
- Port mode/control mode specifiable in 1-bit units by port mode control register DL (PMCDL)

See **Table 4-8, "Port Type," on page 170** for alternate functions.

(2) Registers

(a) Port register DL (PDL)

Port register DL (PDL) is a 16-bit register used to control pin level read and output level write.
This register can be read and written in 8-bit and 1-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 becomes two 8-bit registers for which I/O can be manipulated in 8-bit or 1-bit units.

Figure 4-66: Port Register DL (PDL) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PDLH	0	0	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8	FFFFF005H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PDLL	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0	FFFFF004H	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PDLn	Output data control (in output mode) (n = 0 to 13)
0	Outputs 0
1	Outputs 1

Remarks: 1. Input mode: When port DL (PDL) is read, the pin levels at this time are read. During write, the data written to PDL is written. This does not affect the input pins.

Output mode: When port DL (PDL) is read, the value of PDL is read. During write, the value is written to PDL and the written value is immediately output.

2. An undefined value (pin input level) is read for the value after reset when PDL is read in the input mode. When PDL is read in the output mode, 00H (output latch value) is output.

(b) Port mode register DL (PMDL)

This is a 16-bit register used to specify the input mode/output mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Figure 4-67: Port Mode Register DL (PMDL) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PMDLH	0[Note]	0[Note]	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8	FFFFF025H	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMDLL	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0	FFFFF024H	FFFFH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMDLn	I/O mode control (n = 0 to 13)
0	Output mode
1	Input mode

Note: PMDL14 and PMDL15: please set 0 any time.

(c) Port mode control register (PMCDL)

This is a 16-bit register used to specify the port mode/control mode.
This register can be read and written only 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 and written in 8-bit and 1-bit units.

Remark: External bus functions are only valid for the µPD70F3403 and µPD70F3403A

Figure 4-68: Port Mode Control Register DL (PMCDL) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
PMCDLH	0	0	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8	FFFFF045H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PMCDLL	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0	FFFFF044H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

PMCDLn	PDLn pin operation mode specification (n = 0 to 13)
0	I/O port
1	ADn I/O (address/data bus I/O)

4.4 Port Function Operation

The port operation differs according to the I/O mode settings, as follows.

4.4.1 Write to I/O ports

(1) Output mode

Values are written to output latches using transfer instructions. Moreover, the output latch contents are output to the pin. Once data has been written to an output latch, it is held until data is newly written to that output latch.

(2) Input mode

Values are written to output latches using transfer instructions. However, since the output buffer is off, the status of the pin does not change.

Once data has been written to an output latch, it is held until data is newly written to that output latch.

Caution: In the case of a 1-bit memory manipulation instruction, the manipulation target is 1 bit, but the port is accessed in 8-bit units. Therefore, for ports where input and output are mixed, the contents of output latches of pins specified for input other than the manipulation target bit become undefined.

4.4.2 Read from I/O port

(1) Output mode

The contents of the output latch are read using a transfer instruction. The contents of the output latch do not change.

(2) Input mode

The pin status is read using a transfer instruction. The contents of the output latch do not change.

4.4.3 I/O port calculation

(1) Output port

Calculation for the output latch contents is performed and the result is written to the output latch. The output latch contents are output to the pin.

Once data has been written to an output latch, it is held until data is newly written to that output latch.

(2) Input mode

The output latch contents become undefined. However, since the output buffer is off, the pin status does not change.

Caution: In the case of a 1-bit memory manipulation instruction, the manipulation target is 1 bit, but the port is accessed in 8-bit units. Therefore, for ports where input and output are mixed, the contents of output latches of pins specified for input other than the manipulation target bit become undefined.

4.5 Port Type

Table 4-8: Port Type (1/3)

Pin name		Alternate Function	I/O	Port block Type Note 1
Port 0	P00	SI31	I/O	E-SD1
	P01	SO31		E-SD4
	P02	$\overline{\text{SCK31}}$		E-SD7E
	P03	CS310		E-SDW4
	P04	CS311		E-SD4
	P05	CS312/INTP2		N-SDW7
	P06	CS313/INTP3		N-SD7
Port 1	P10	INTP0	I/O	L-SD1
	P11	TIP31/TOP31		G-SD7
	P12	TIP30/TOP30/ADTRG		U-SD8
Port 3	P30	TXDA0/[$\overline{\text{ASCKA0}}$]	I/O	E-SDW7
	P31	RXDA0/INTP7		N-SD2
	P32	$\overline{\text{ASCKA0}}$/[TXDA0]/(CS310)		E-SDW10
	P33	CTXD0		E-SD4
	P34	CRXD0		E-SD1L
	P35	CRXD1		E-SD1L
	P36	CTXD1		E-SD4
	P37	TIP00/TOP00/INTP1/(CS312)		W-SDW11
	P38	TIP01/TOP01		G-SD7
Port 4	P40	SIB0	I/O	E-SD1
	P41	SOB0		E-SD4
	P42	$\overline{\text{SCKB0}}$		E-SD7
Port 5	P50	TIQ01/TOQ01	I/O	G-SD7
	P51	TIQ02/TOQ02		G-SD7
	P52	TIQ03/TOQ03		G-SD7
	P53	TIQ00/TOQ00		G-SD7
	P54	SI30		E-SD1
	P55	SO30		E-SD4
Port 7	P70 - P715	ANI0-ANI15	I/O	A-1

Notes: 1. Refer to **4.6 Port Block Types**

2. The P911 pin's alternate functions are pins for on-chip debugging. After reset, the P911/SOB2/[DRST] pins are initialized to the on-chip debugging pin DRST. When using the P911 pin as a port and not as an on-chip debugging pin, the following handling is required.

<1> Write "0" to OCDM0 bit of OCDM register.
<2> P911/SOB2/[DRST] pin input to "L" until handling <1> above is completed.

Setting DRST = "H" before the above handling is performed when not using on-chip debugging will cause malfunction (CPU deadlock). Although P911 pin has built-in pull down resister, be careful about the handling of the P911 pin.

3. Externally pulling down the P911 pin is recommended.

4. Since the pin AD5/FLMD1 is set in the flash programming mode, it need not be manipulated with the port control register. For details, refer to CHAPTER 17 FLASH MEMORY.

Table 4-8: Port Type (2/3)

Pin name		Alternate Function	I/O	Port block Type Note 1
Port 9	P90	TIQ11/TOQ11/$\overline{\text{SCK30}}$	I/O	U-SDW11E
	P91	TIQ12/TOQ12/CS300		U-SDW10
	P92	TIQ13/TOQ13/CS301		U-SDW10
	P93	TIQ10/TOQ10/CS302		U-SDW10
	P94	$\overline{\text{ASCKA1}}$/CS303		G-SD7A
	P95	RXDA1/($\overline{\text{SCK30}}$)		G-SDW8E
	P96	TXDA1/(CS300)		G-SDW6
	P97	SIB1/{DDI}		G-SDJ2
	P98	SOB1/{DCK}		G-SDJ5
	P99	$\overline{\text{SCKB1}}$/{DMS}		G-SDJ8E
	P910	{CS301}/{DDO}		E-SWJ7
	P911	{$\overline{\text{DRST}}$} Notes 2, 3		E-DWJ4
	P912	TIP21/TOP21/(CS302)		U-SDW11
	P913	TIP20/TOP20/INTP4/PCL		W-SD11
	P914	TIP10/TOP10/INTP5/(AD14)		W-SD12E
	P915	TIP11/TOP11/INTP6/(AD15)		W-SD12E
Port CM	PCM0	$\overline{\text{WAIT}}$	I/O	E-D1/C-D1(70F3402)
	PCM1	CLKOUT		E-D4/C-D1(70F3402)
	PCM2	$\overline{\text{HLDAK}}$		E-D4/C-D1(70F3402)
	PCM3	$\overline{\text{HLDAQ}}$		E-D1/C-D1(70F3402)
Port CS	PCS0	$\overline{\text{CS0}}$	I/O	E-D4/C-D1(70F3402)
	PCS1	$\overline{\text{CS1}}$		E-D4/C-D1(70F3402)
Port CT	PCT0	$\overline{\text{WR0}}$	I/O	E-D4/C-D1(70F3402)
	PCT1	$\overline{\text{WR1}}$		E-D4/C-D1(70F3402)
	PCT4	$\overline{\text{RD}}$		E-D4/C-D1(70F3402)
	PCT6	ASTB		E-D4/C-D1(70F3402)

Notes: 1. Refer to **4.6 Port Block Types**

2. The P911 pin's alternate functions are pins for on-chip debugging. After reset, the P911/SOB2/[DRST] pins are initialized to the on-chip debugging pin DRST. When using the P911 pin as a port and not as an on-chip debugging pin, the following handling is required.

<1> Write "0" to OCDM0 bit of OCDM register.
<2> P911/SOB2/[DRST] pin input to "L" until handling <1> above is completed.

Setting DRST = "H" before the above handling is performed when not using on-chip debugging will cause malfunction (CPU deadlock). Although P911 pin has built-in pull down resister, be careful about the handling of the P911 pin.

3. Externally pulling down the P911 pin is recommended.

4. Since the pin AD5/FLMD1 is set in the flash programming mode, it need not be manipulated with the port control register. For details, refer to CHAPTER 17 FLASH MEMORY.

Table 4-8: Port Type (3/3)

Pin name		Alternate Function	I/O	Port block Type Note 1
Port DL	PDL0	AD0	I/O	D-7E/B (70F3402)
	PDL1	AD1		D-7E/B (70F3402)
	PDL2	AD2		D-7E/B (70F3402)
	PDL3	AD3		D-7E/B (70F3402)
	PDL4	AD4		D-7E/B (70F3402)
	PDL5	AD5/FLMD1 Note 4		D-7E/B (70F3402)
	PDL6	AD6		D-7E/B (70F3402)
	PDL7	AD7		D-7E/B (70F3402)
	PDL8	AD8		D-7E/B (70F3402)
	PDL9	AD9		D-7E/B (70F3402)
	PDL10	AD10		D-7E/B (70F3402)
	PDL11	AD11		D-7E/B (70F3402)
	PDL12	AD12		D-7E/B (70F3402)
	PDL13	AD13		D-7E/B (70F3402)
	AV_{REF0}			-
	AV_{SS}			-
	FLMD0			-
	REGC			-
	$\overline{\text{RESET}}$			-
	X1			-
	X2			-
	V_{DD0}			-
	V_{SS0}			-
	V_{DD1}			-
	V_{SS1}			-
	BV_{DD}			-
	BV_{SS}			-
	REGC0			-
	REGC1			-

Notes: 1. Refer to **4.6 Port Block Types**

2. The P911 pin's alternate functions are pins for on-chip debugging. After reset, the P911/SOB2/[DRST] pins are initialized to the on-chip debugging pin DRST. When using the P911 pin as a port and not as an on-chip debugging pin, the following handling is required.

<1> Write "0" to OCDM0 bit of OCDM register.
<2> P911/SOB2/[DRST] pin input to "L" until handling <1> above is completed.

Setting DRST = "H" before the above handling is performed when not using on-chip debugging will cause malfunction (CPU deadlock). Although P911 pin has built-in pull down resister, be careful about the handling of the P911 pin.

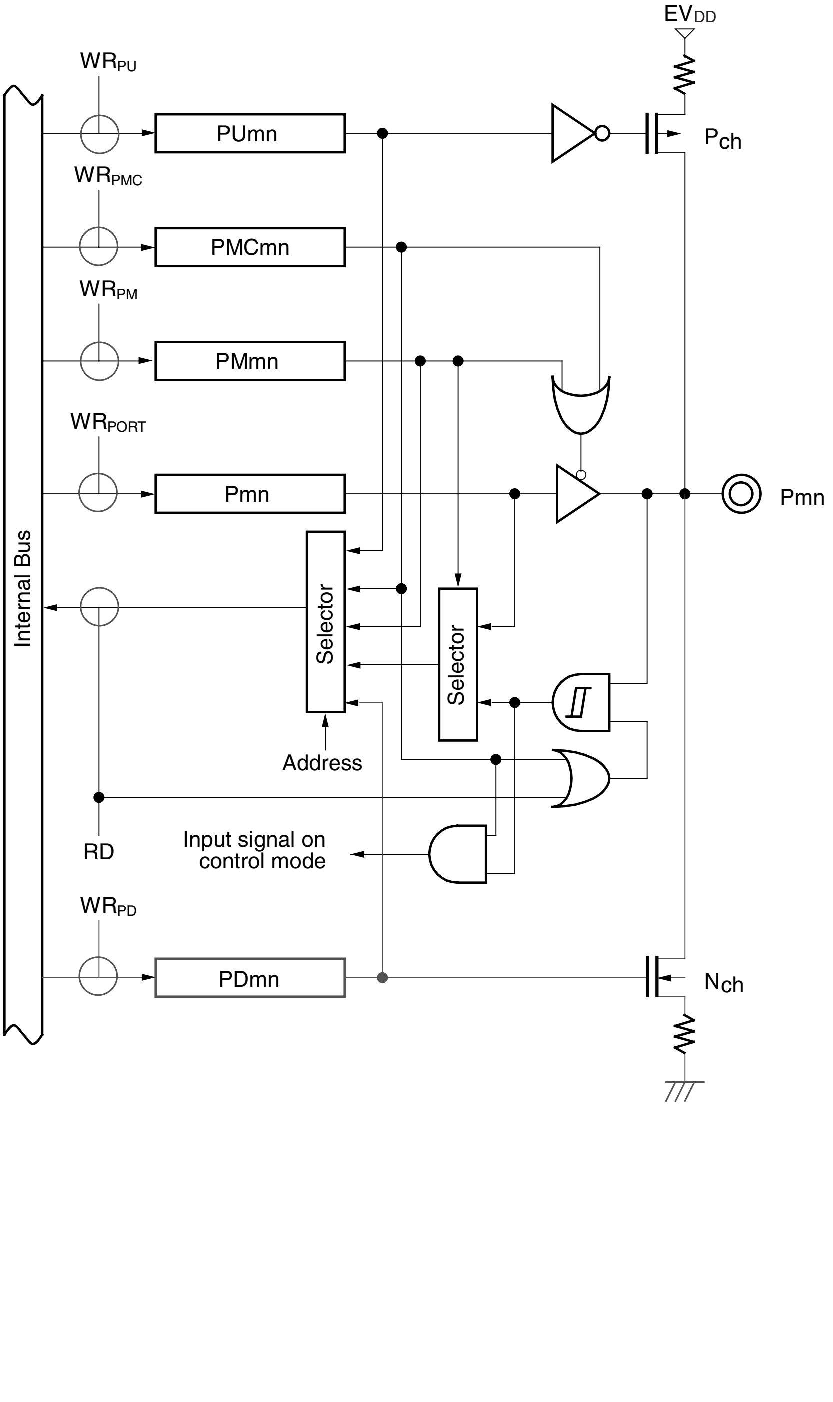
3. Externally pulling down the P911 pin is recommended.

4. Since the pin AD5/FLMD1 is set in the flash programming mode, it need not be manipulated with the port control register. For details, refer to CHAPTER 17 FLASH MEMORY.

4.6 Port Block Types

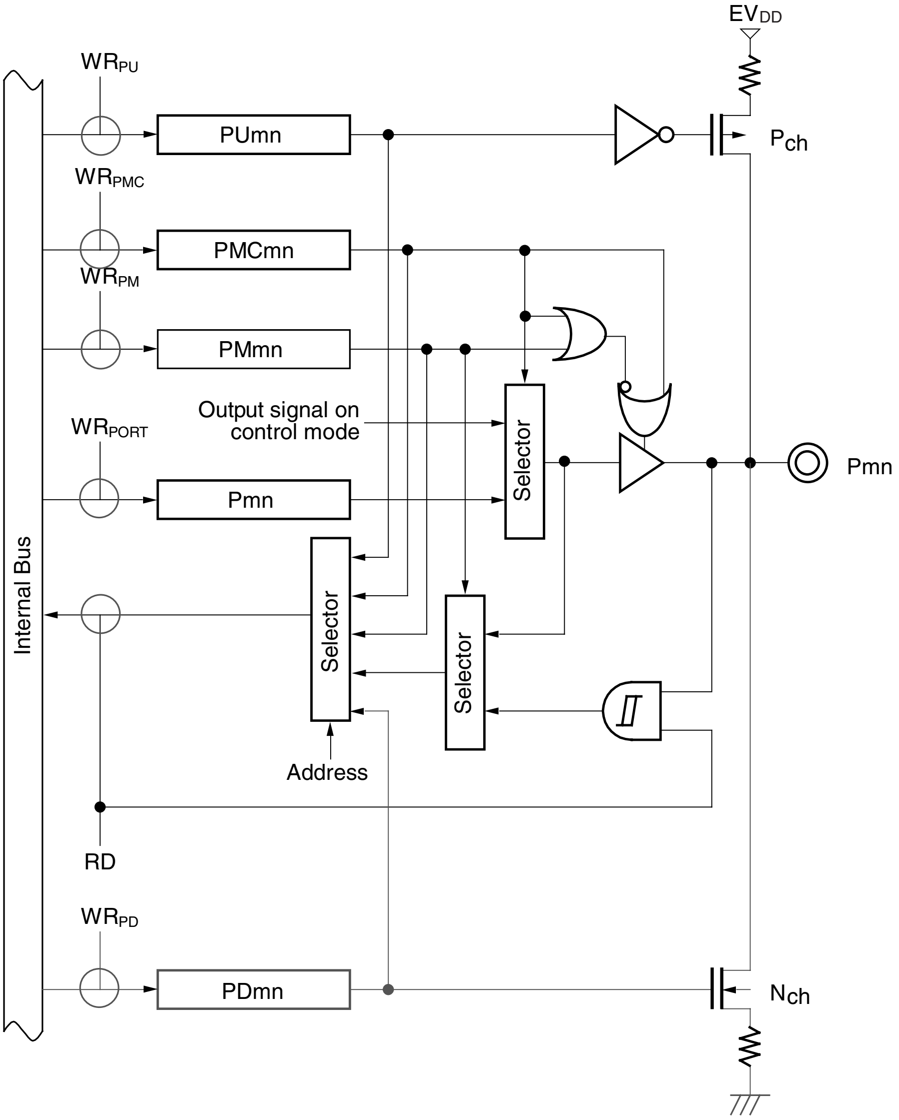
4.6.1 Port block type E-SD1

Figure 4-69: Type E-SD1 Block Diagram



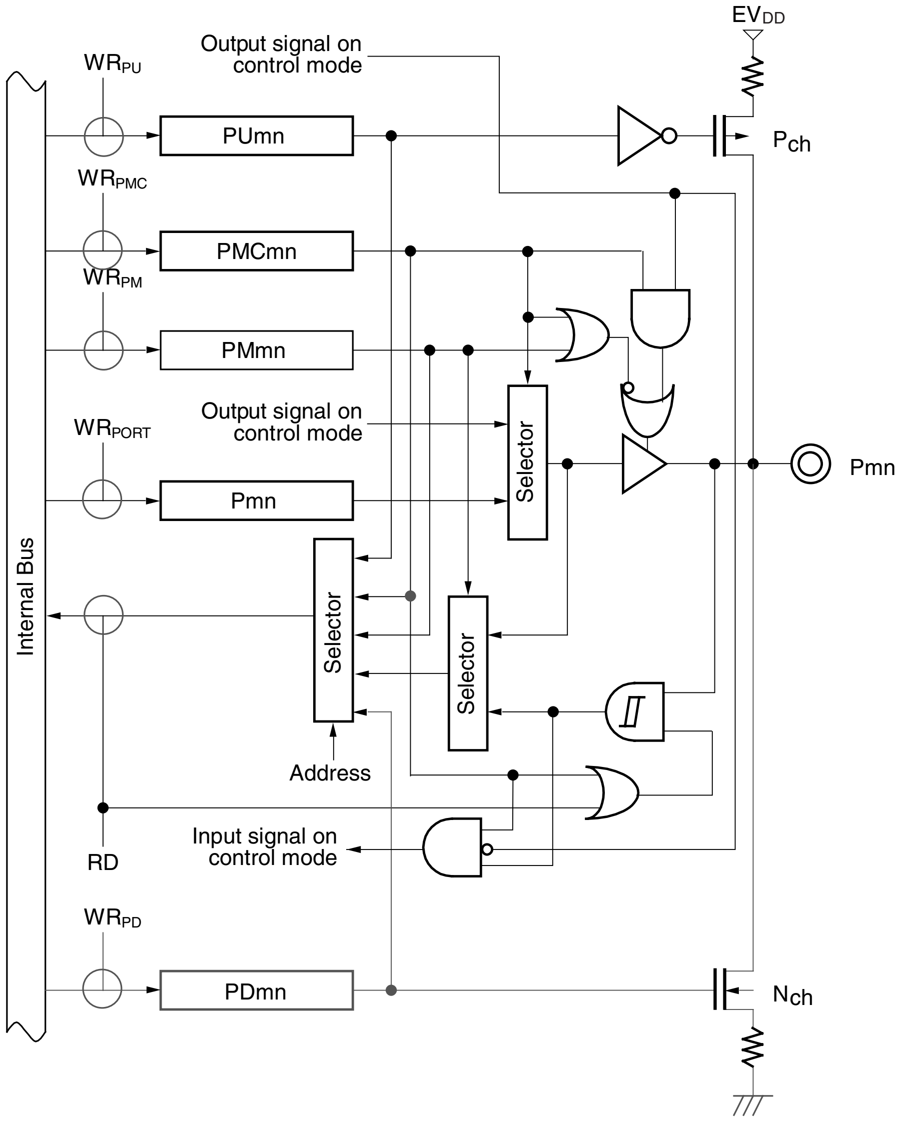
4.6.2 Port block type E-SD4

Figure 4-70: Type E-SD4 Block Diagram



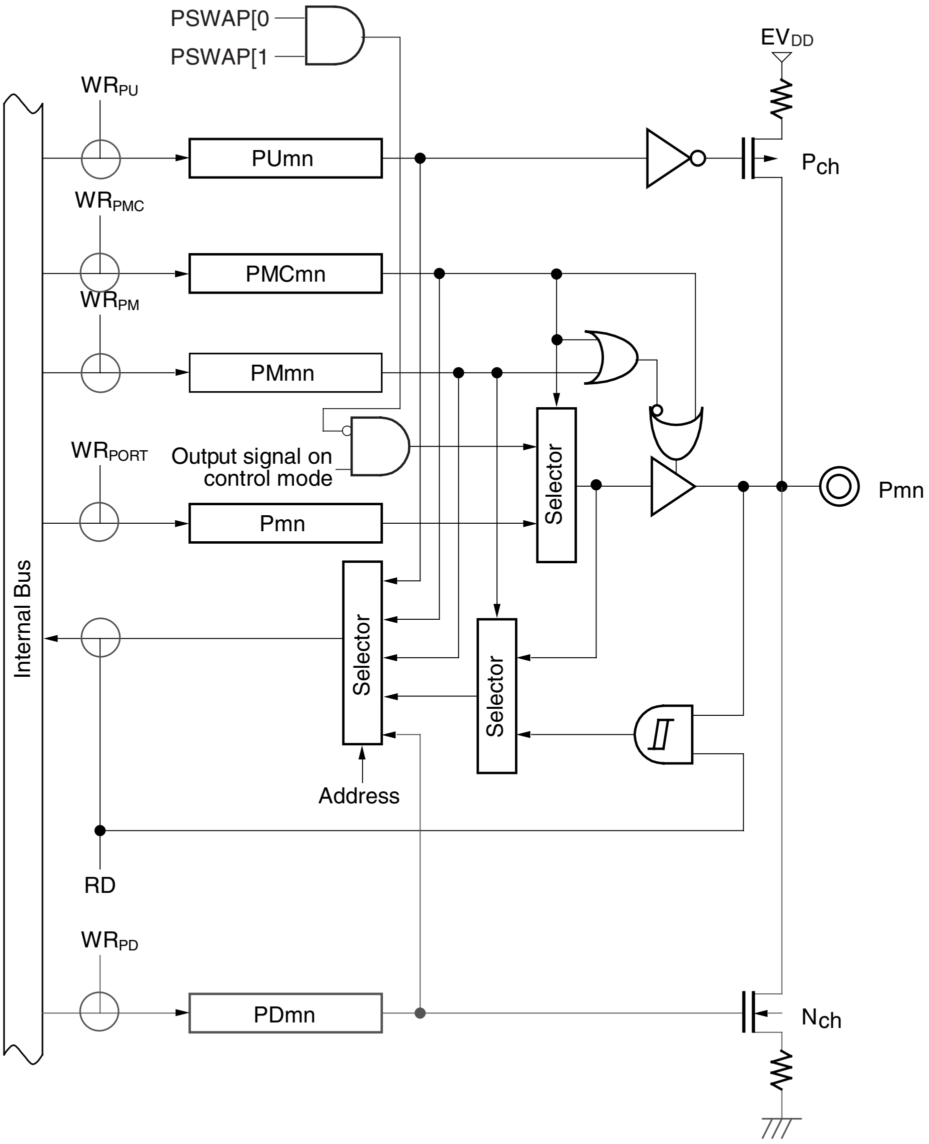
4.6.3 Port block type E-SD7E

Figure 4-71: Type E-SD7E Block Diagram



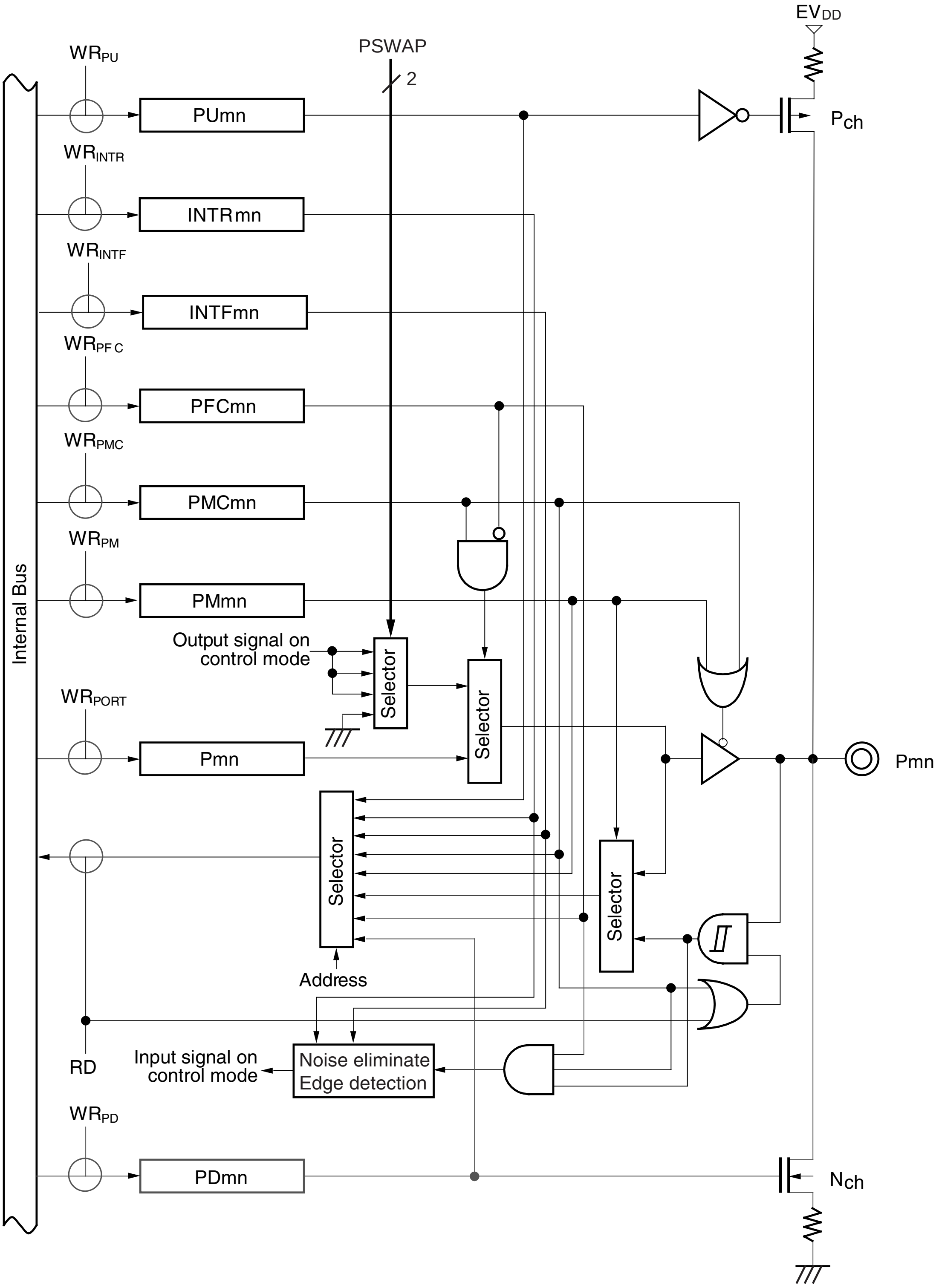
4.6.4 Port block type E-SDW4

Figure 4-72: Type E-SDW4 Block Diagram



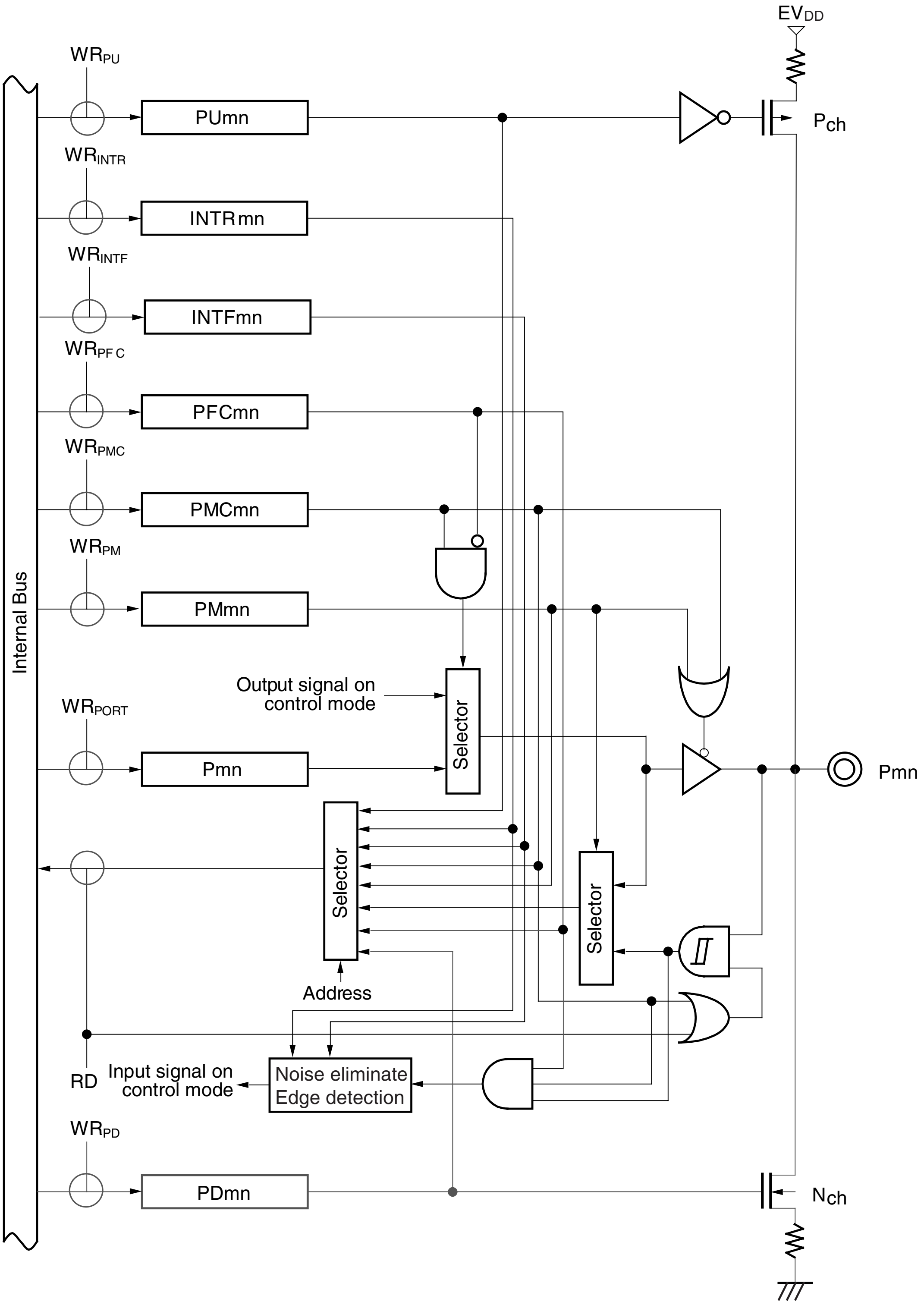
4.6.5 Port block type N-SDW7

Figure 4-73: Type N-SDW7 Block Diagram



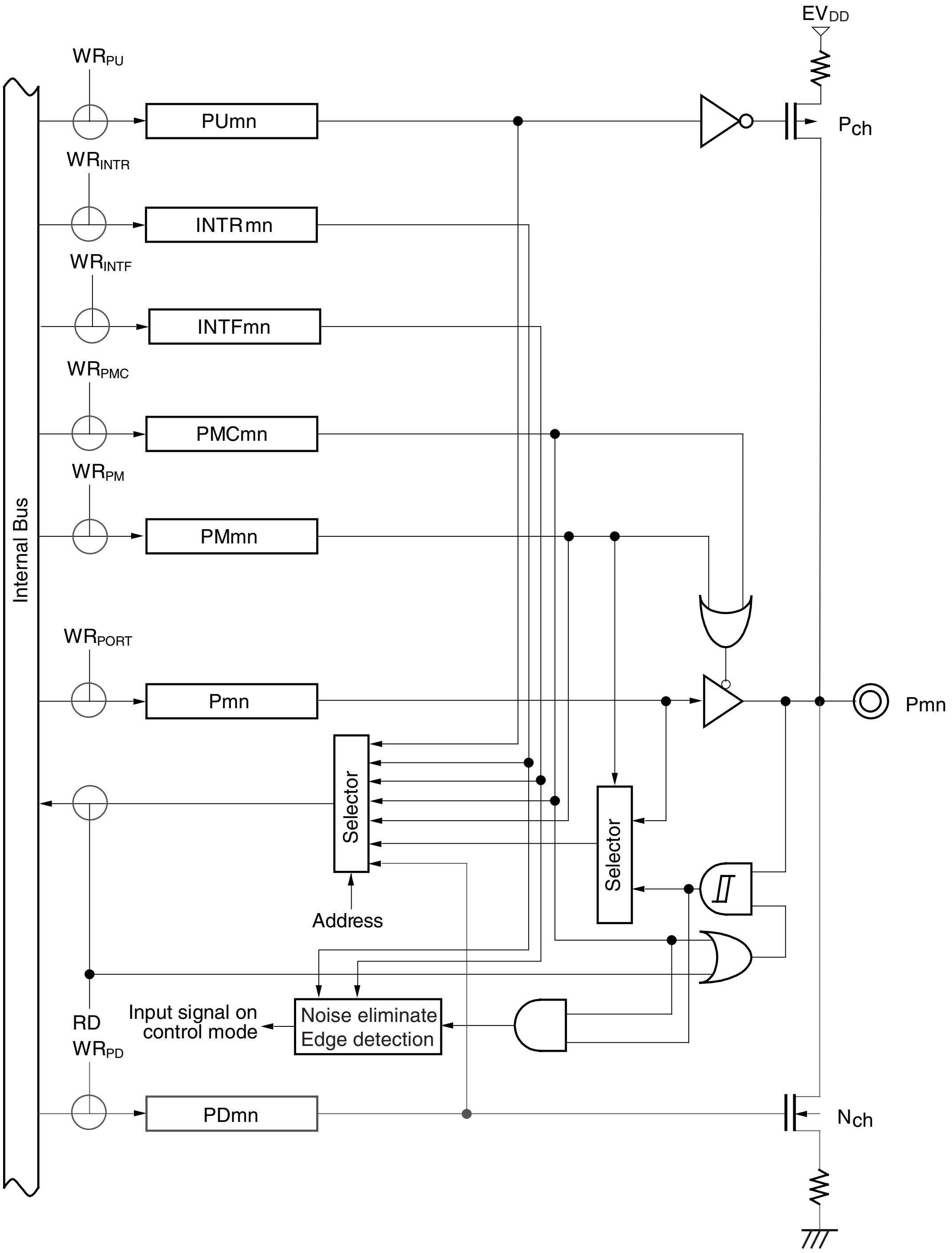
4.6.6 Port block type N-SD7

Figure 4-74: Type N-SD7 Block Diagram



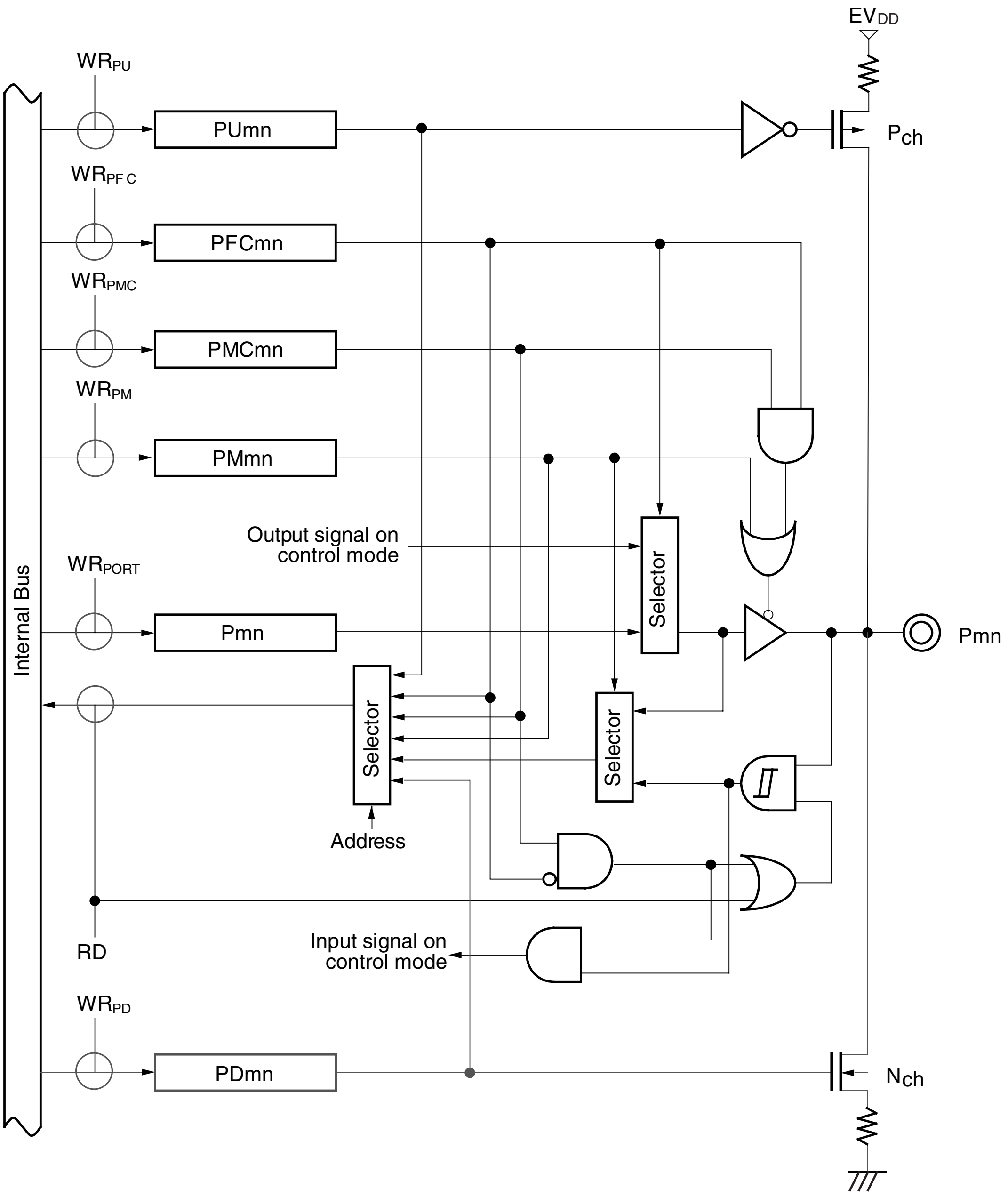
4.6.7 Port block type L-SD1

Figure 4-75: Type L-SD1 Block Diagram



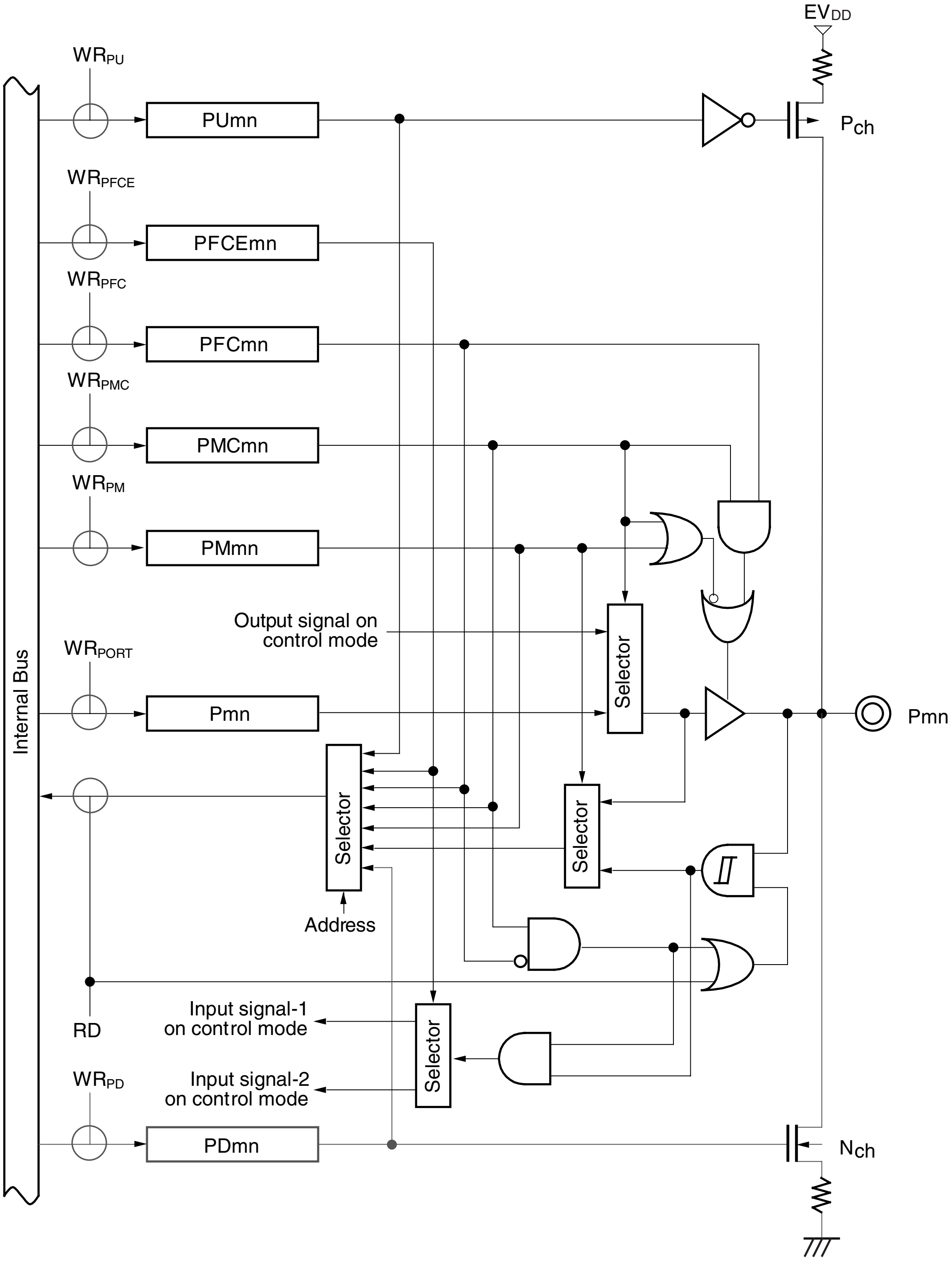
4.6.8 Port block type G-SD7

Figure 4-76: Type G-SD7 Block Diagram



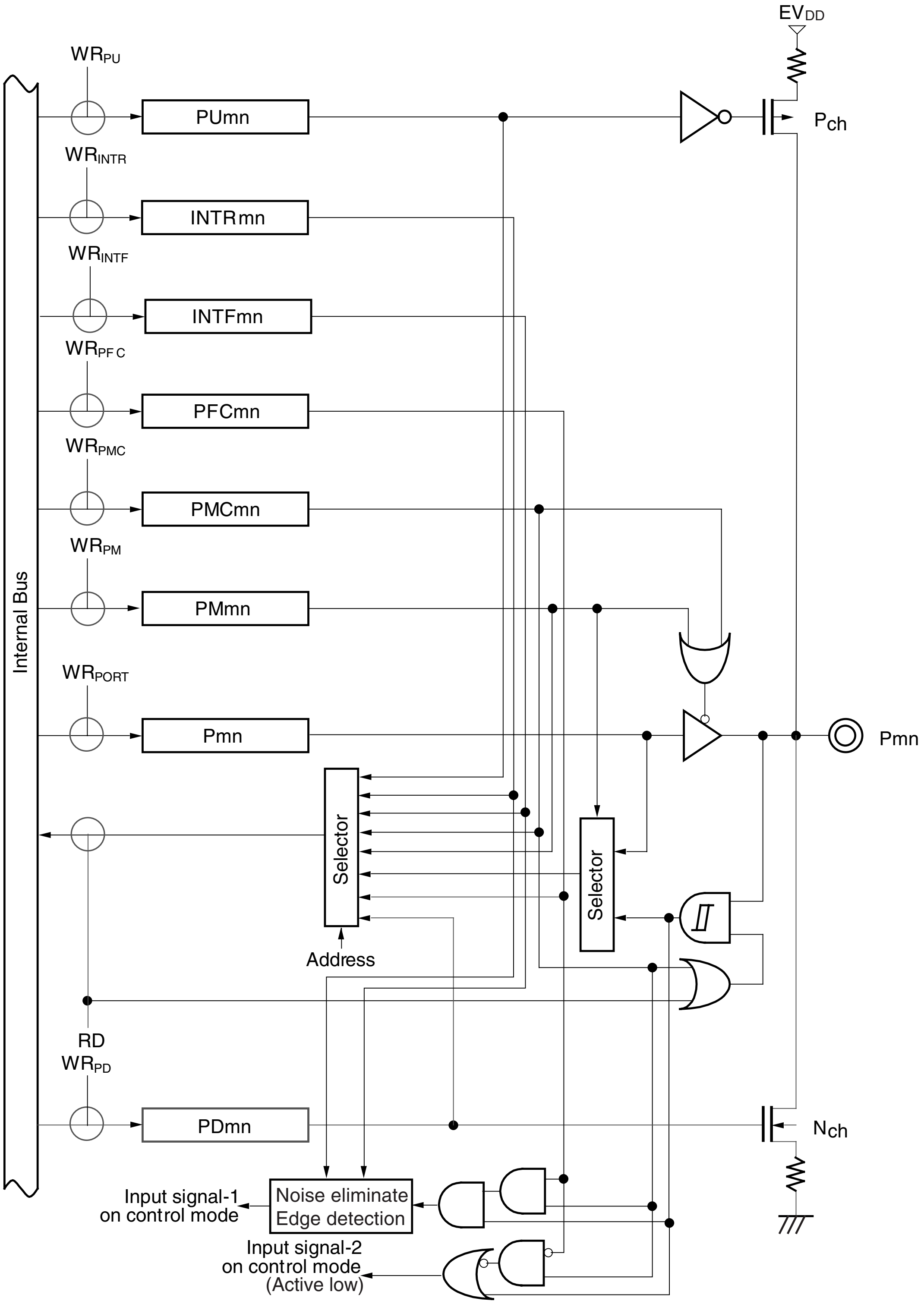
4.6.9 Port block type U-SD8

Figure 4-77: Type U-SD8 Block Diagram



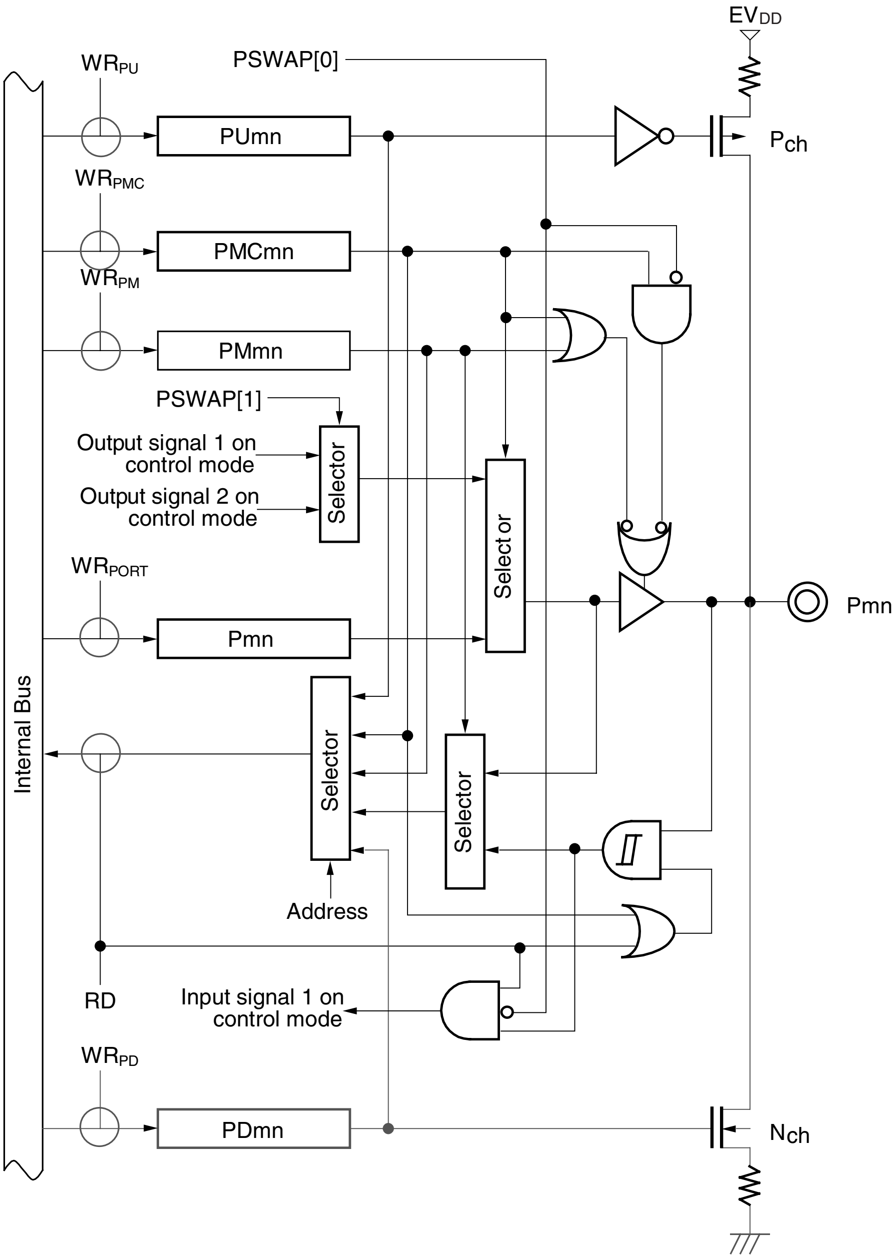
4.6.10 Port block type N-SD2

Figure 4-78: Type N-SD2 Block Diagram



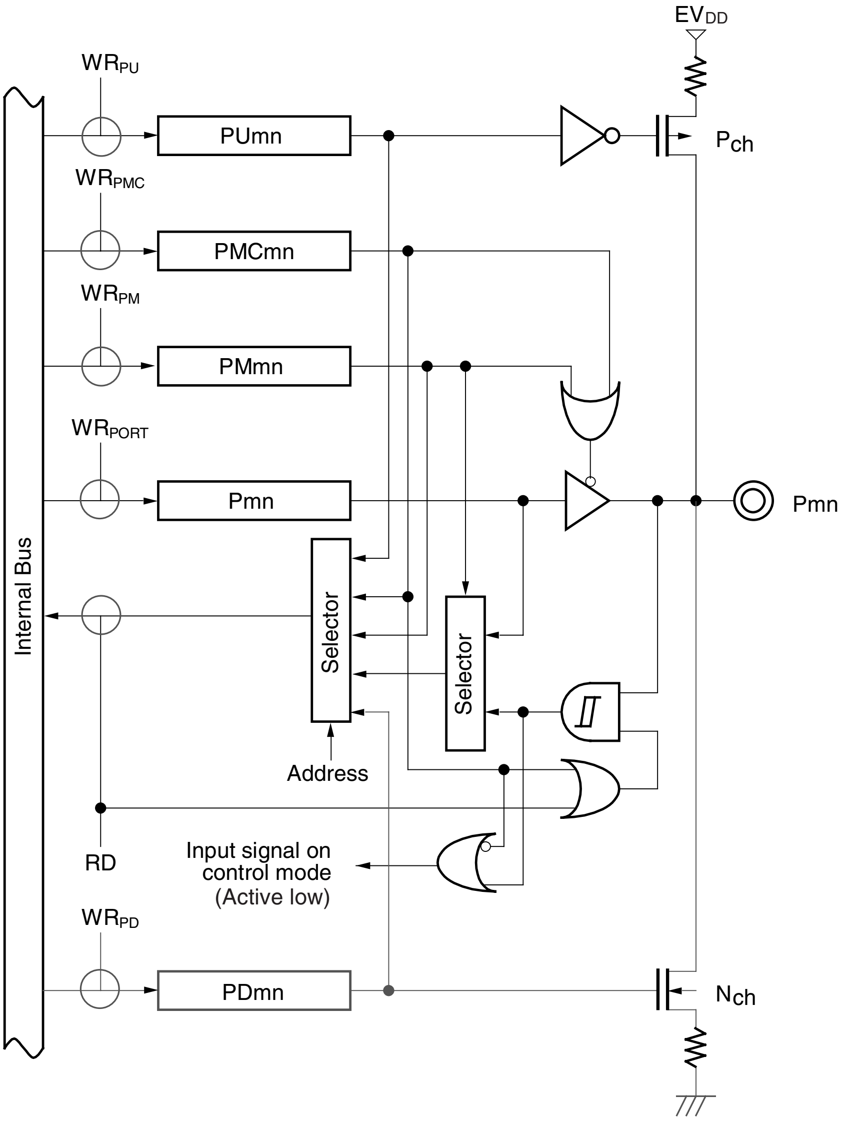
4.6.11 Port block type E-SDW10

Figure 4-79: Type E-SDW10 Block diagram



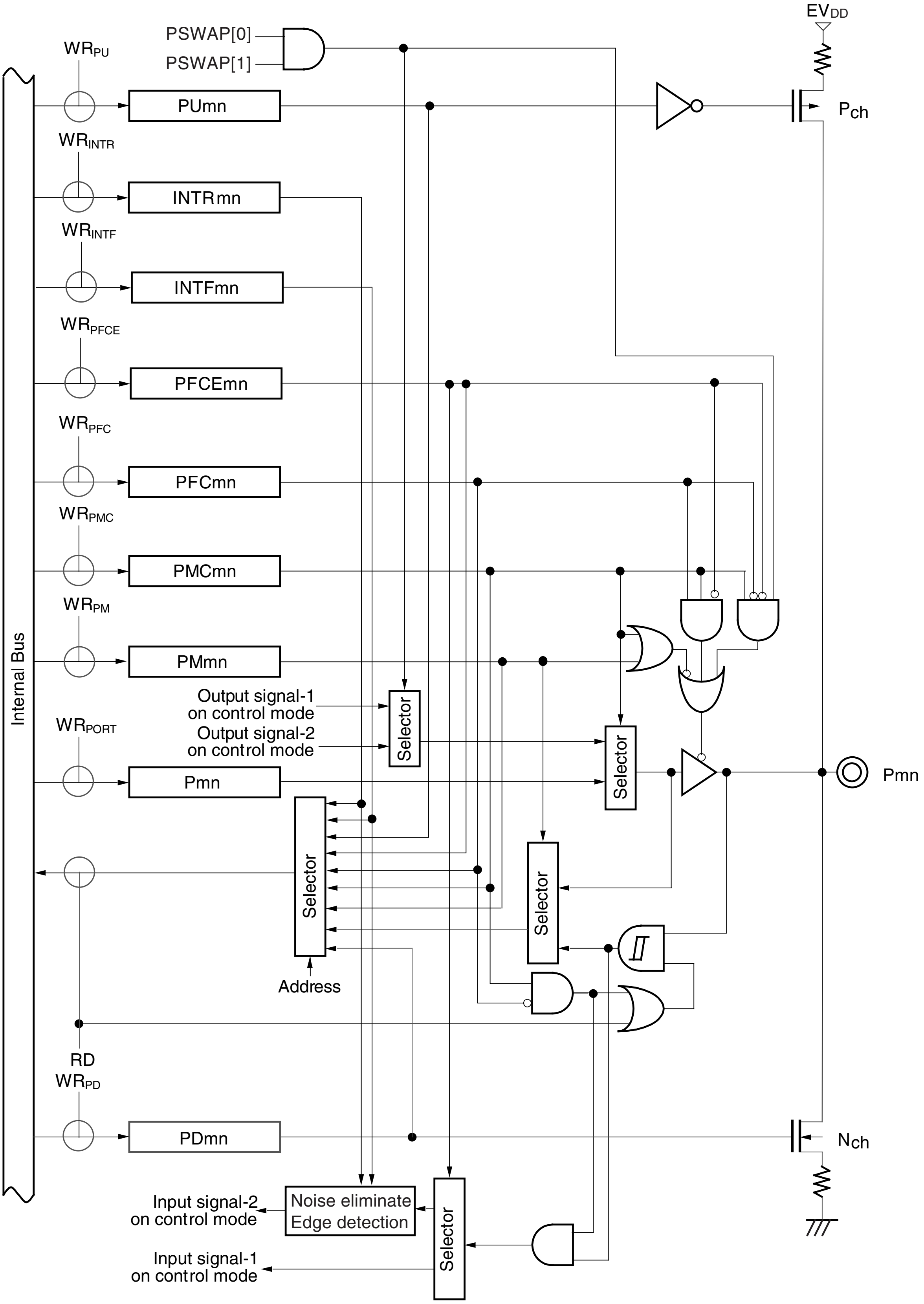
4.6.12 Port block diagram E-SD1L

Figure 4-80: Type E-SD1L Block Diagram



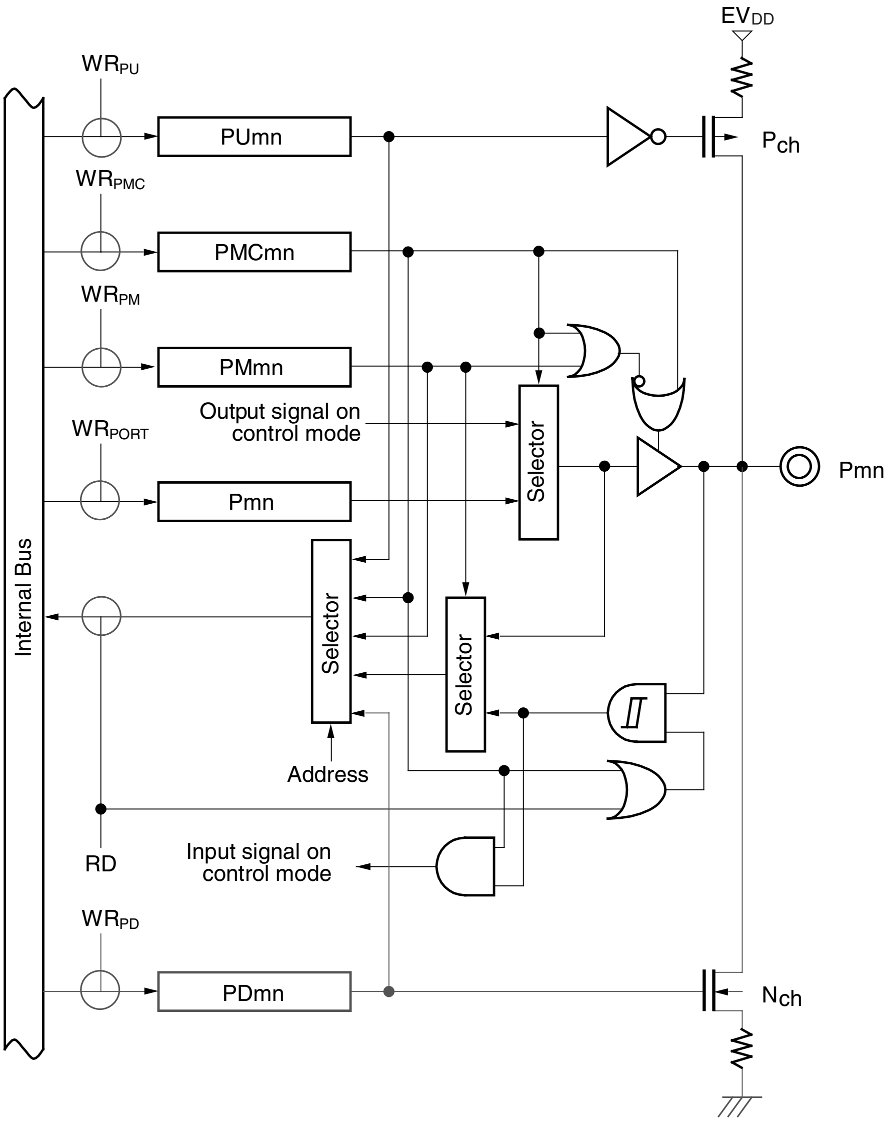
4.6.13 Port block type W-SDW11

Figure 4-81: Type W-SDW11 Block Diagram



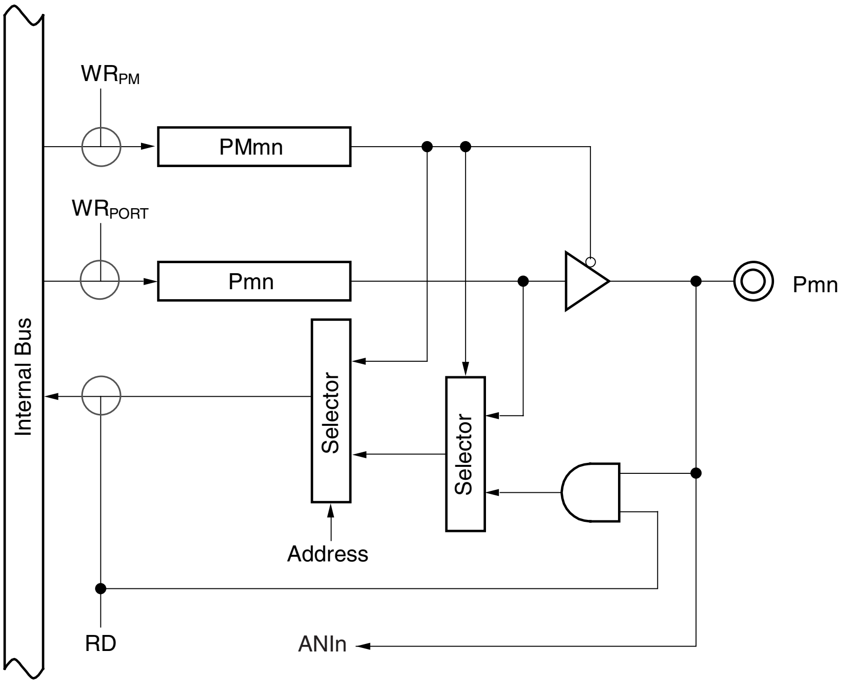
4.6.14 Port block type E-SD7

Figure 4-82: Type E-SD7 Block Diagram



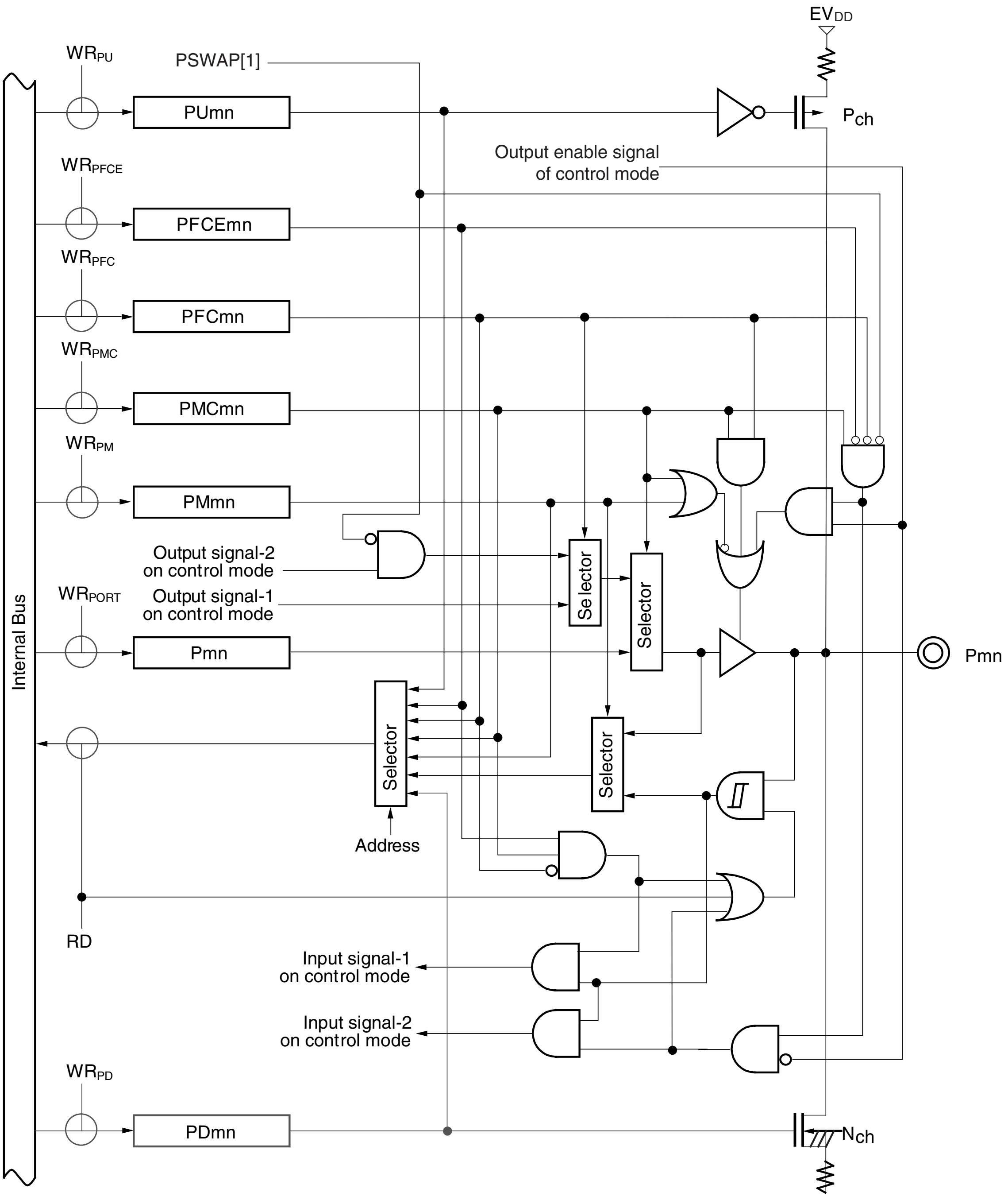
4.6.15 Port block type A-1

Figure 4-83: Type A-1 Block Diagram



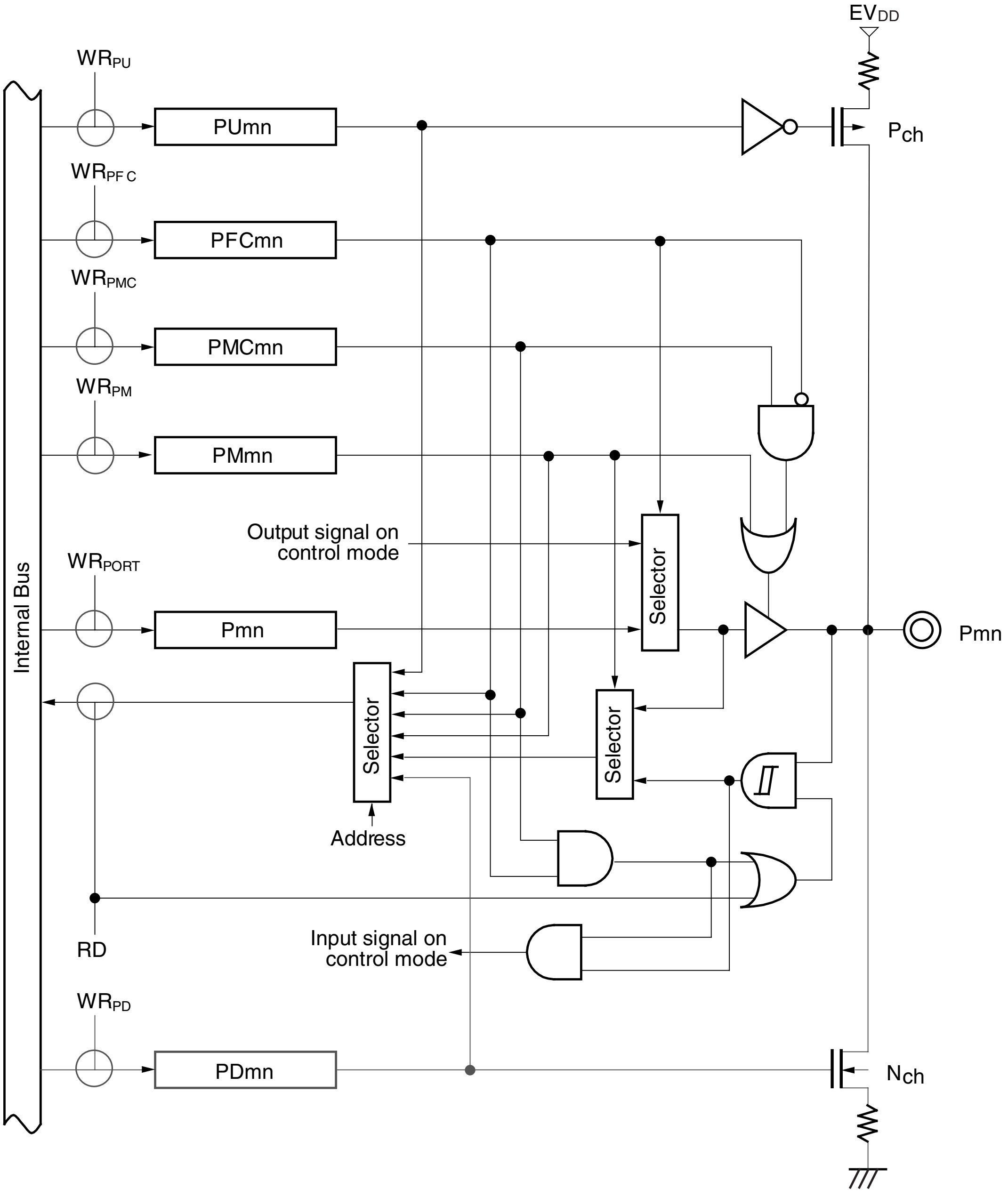
4.6.16 Port block type U-SDW11E

Figure 4-84: Type U-SDW11E Block Diagram



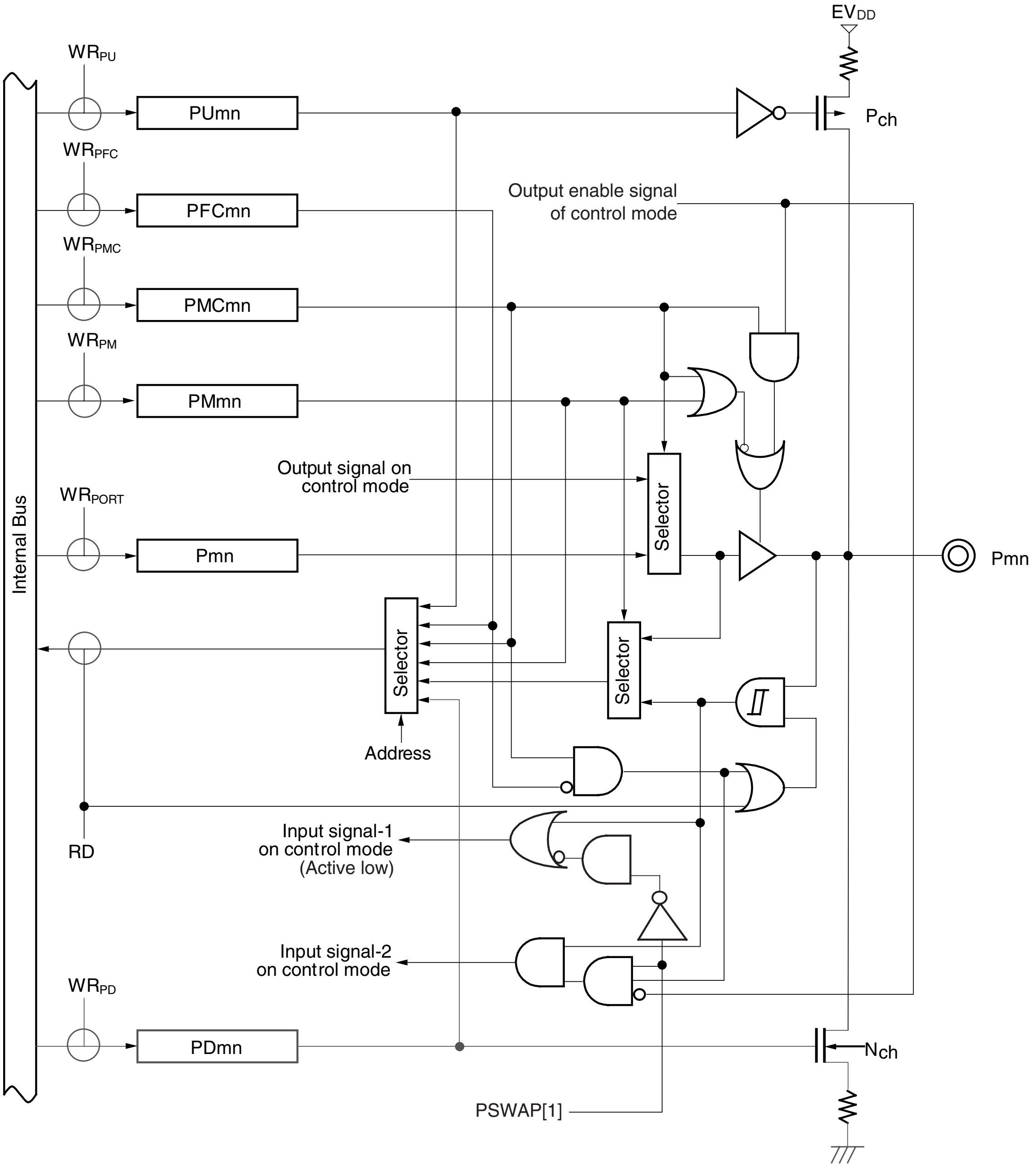
4.6.17 Port block type G-SD7A

Figure 4-85: Type G-SD7A Block Diagram



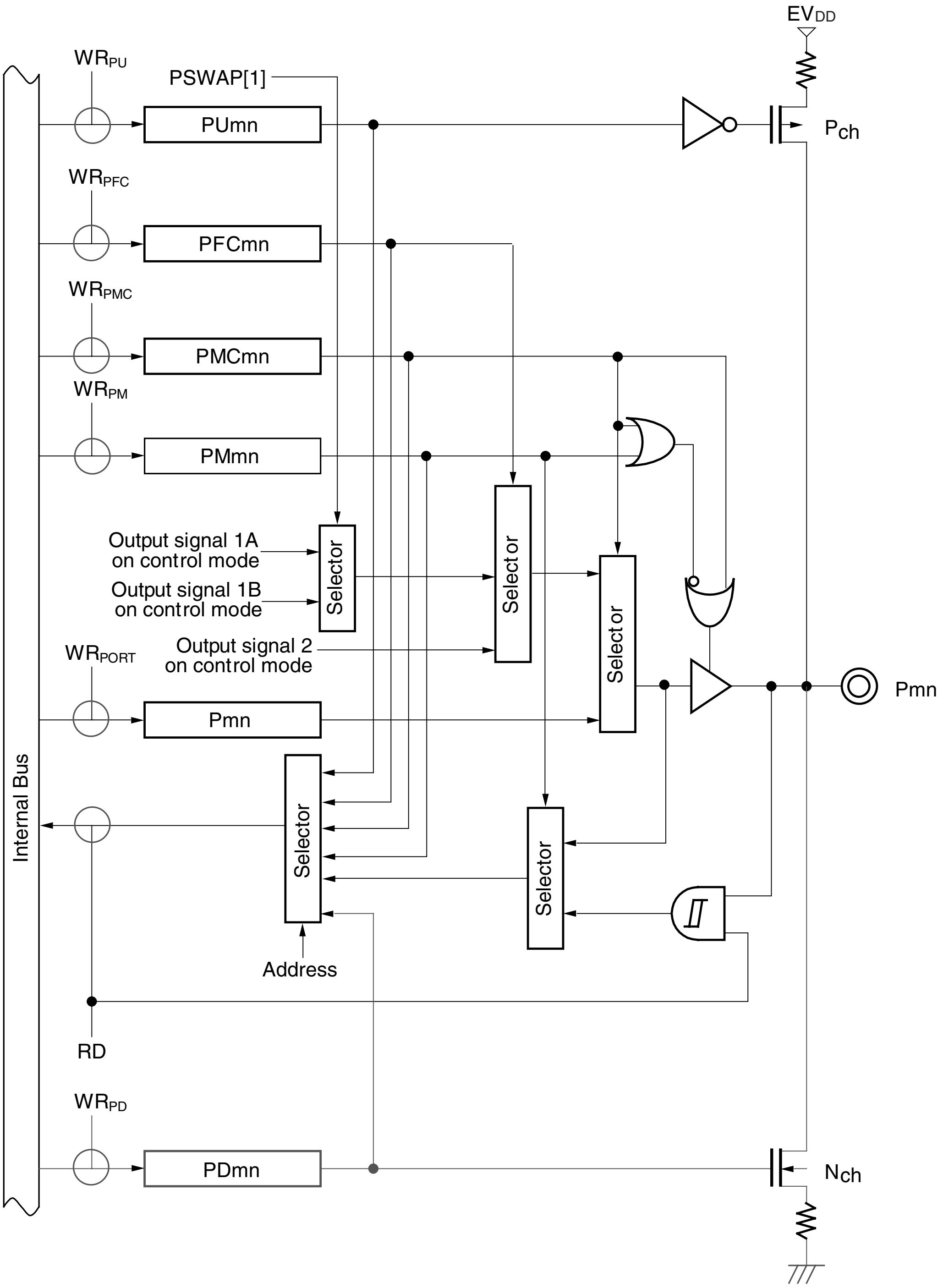
4.6.18 Port block type G-SDW8E

Figure 4-86: Type G-SDW8E Block Diagram



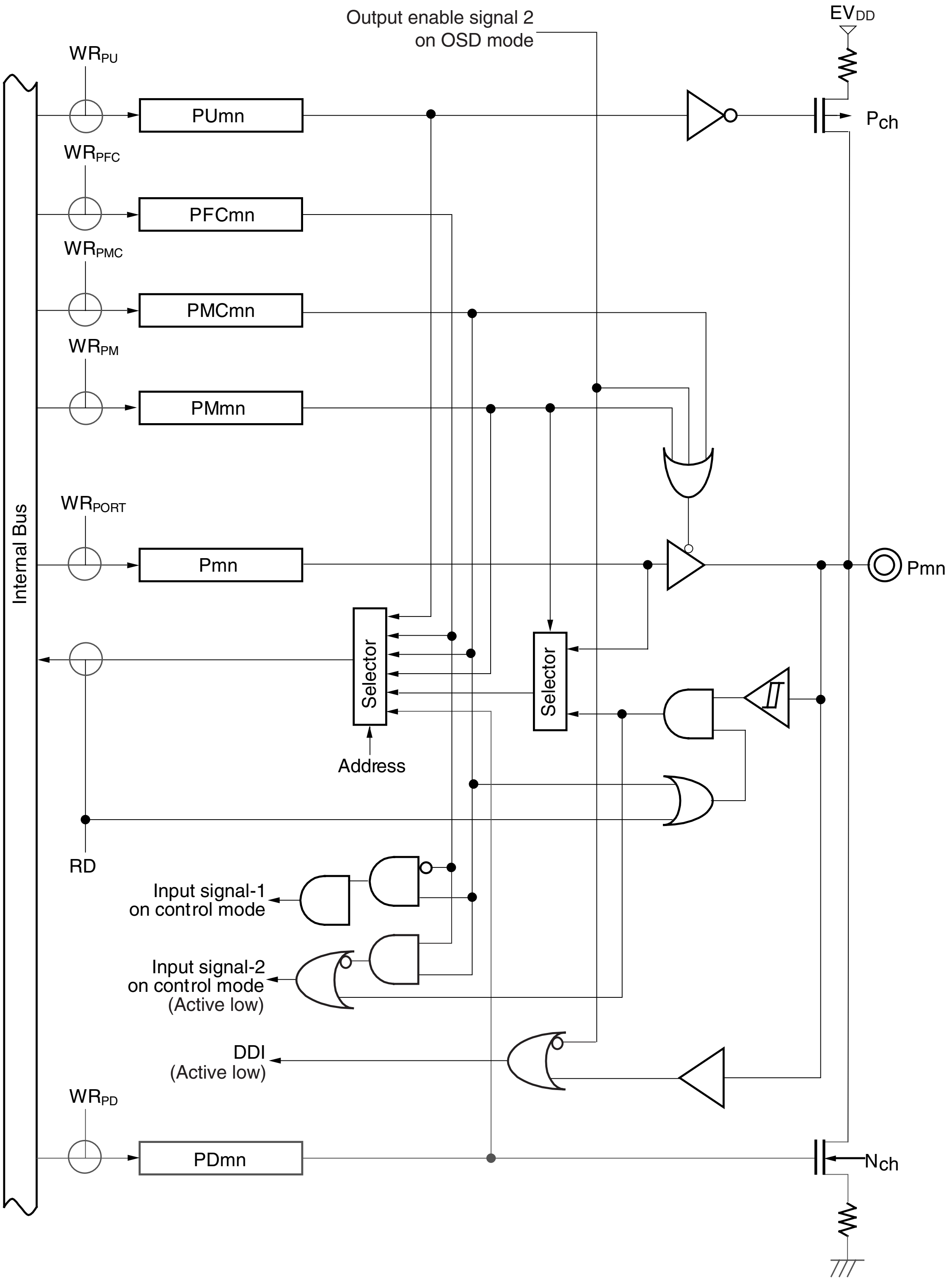
4.6.19 Port block type G-SD6

Figure 4-87: Type G-SD6 Block Diagram



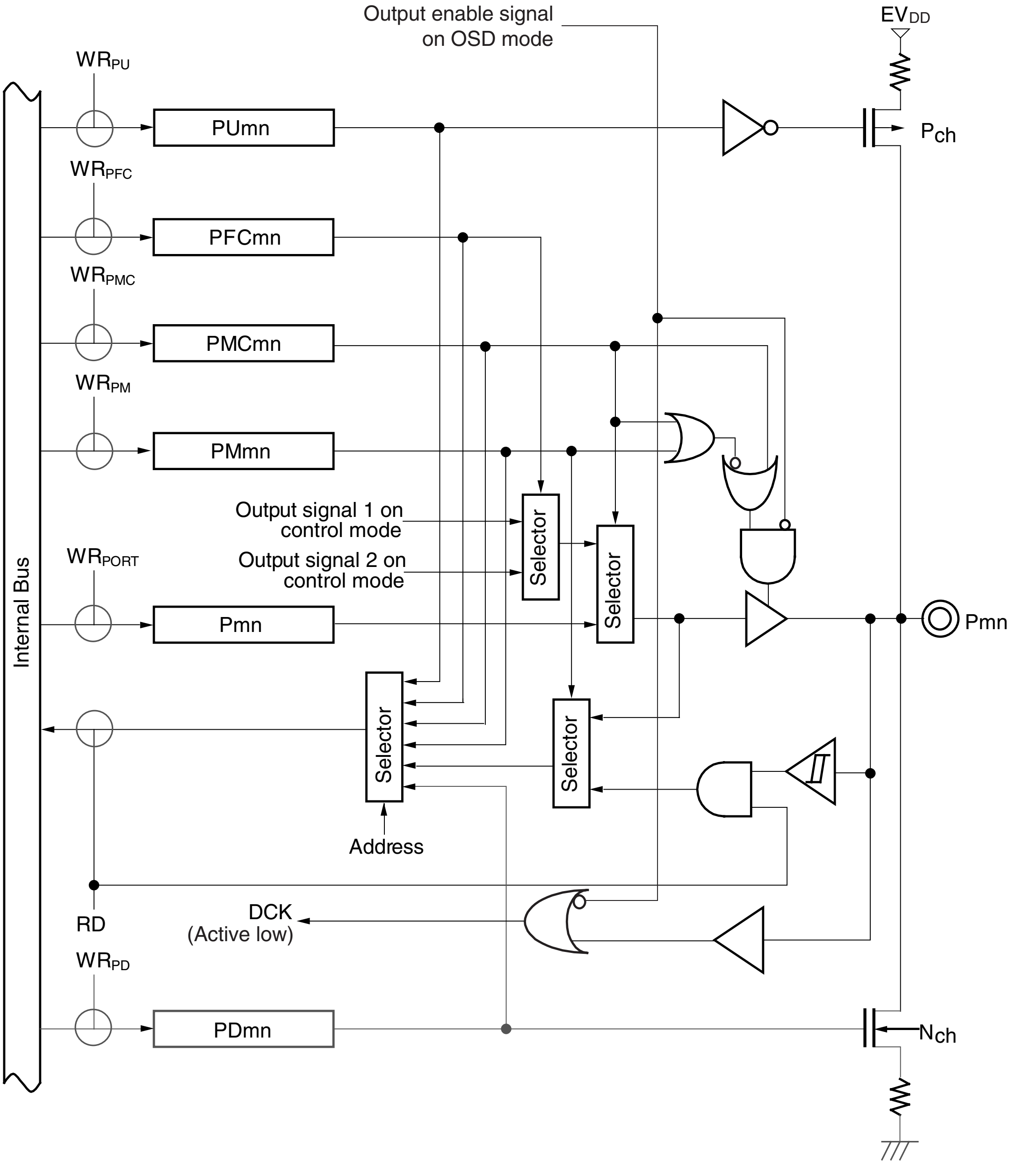
4.6.20 Port block type G-SDJ2

Figure 4-88: Type G-SDJ2 Block Diagram



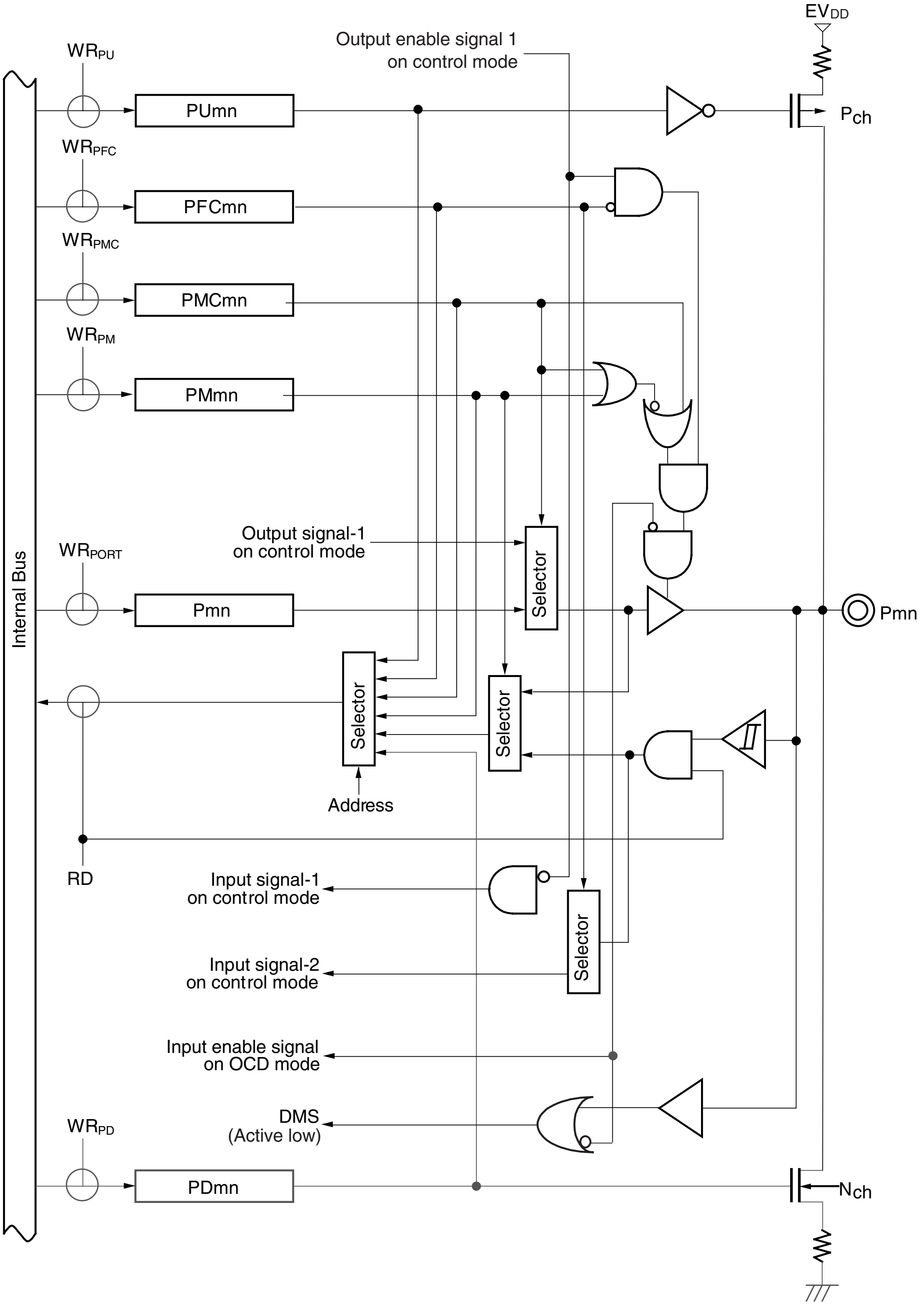
4.6.21 Port block type G-SDJ5

Figure 4-89: Type G-SDJ5 Block Diagram



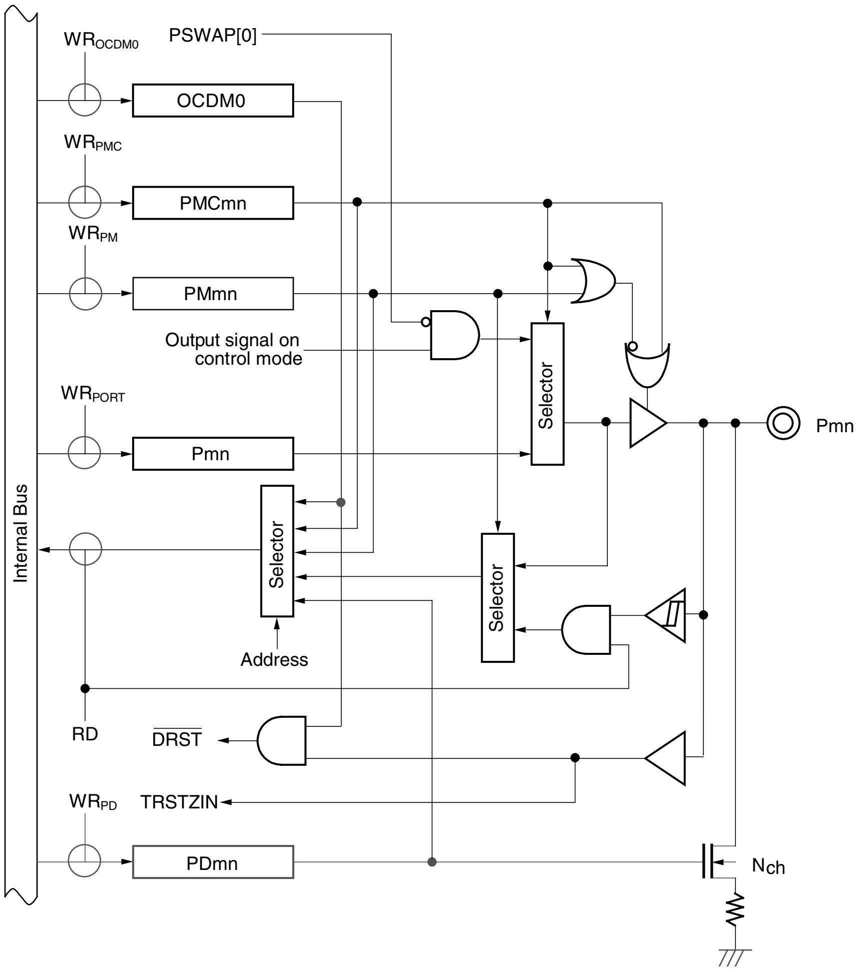
4.6.22 Port block type G-SDJ8E

Figure 4-90: Type G-SDJ8E Block Diagram



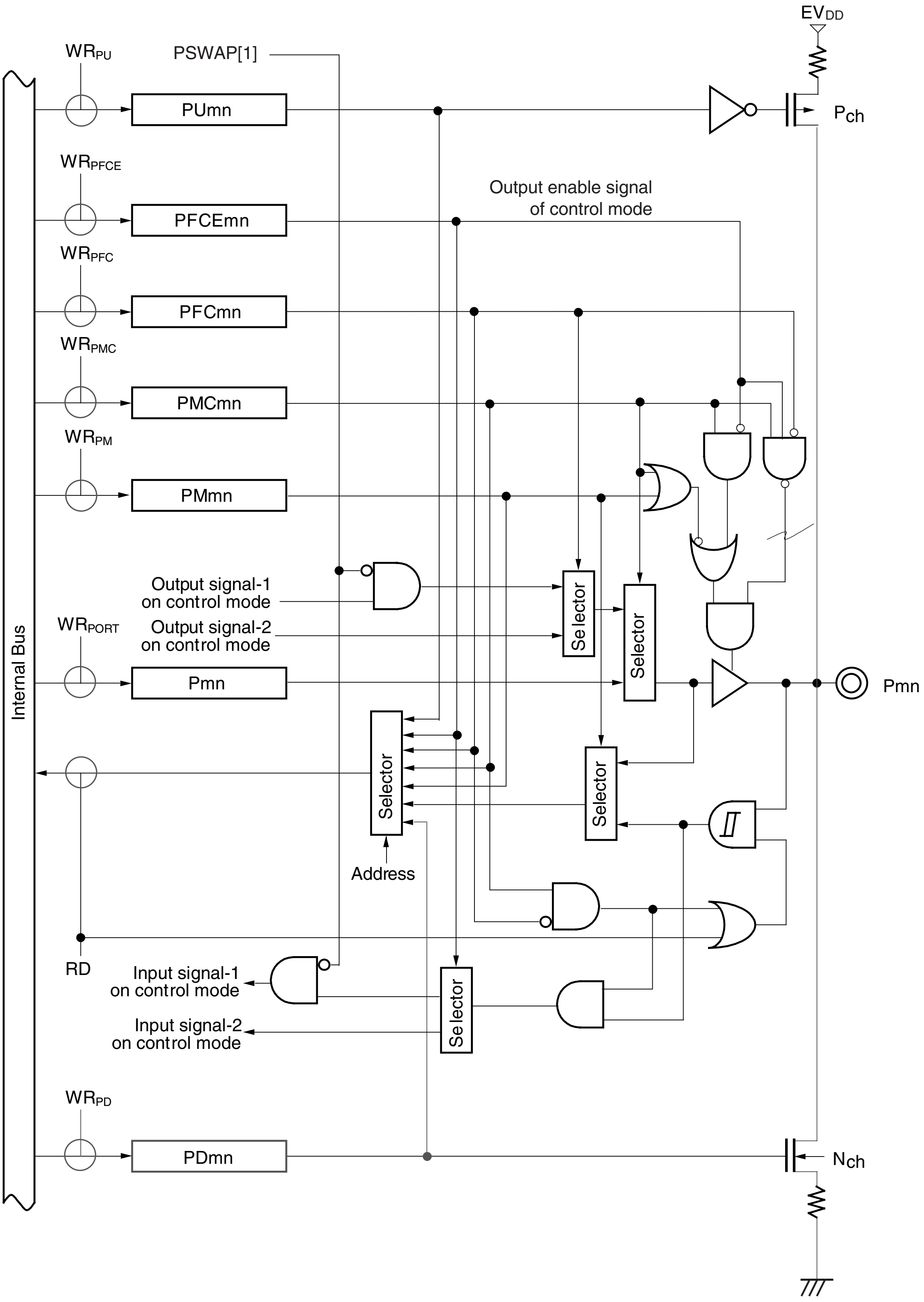
4.6.23 Port block type E-DWJ4

Figure 4-91: Type E-DWJ4 Block Diagram



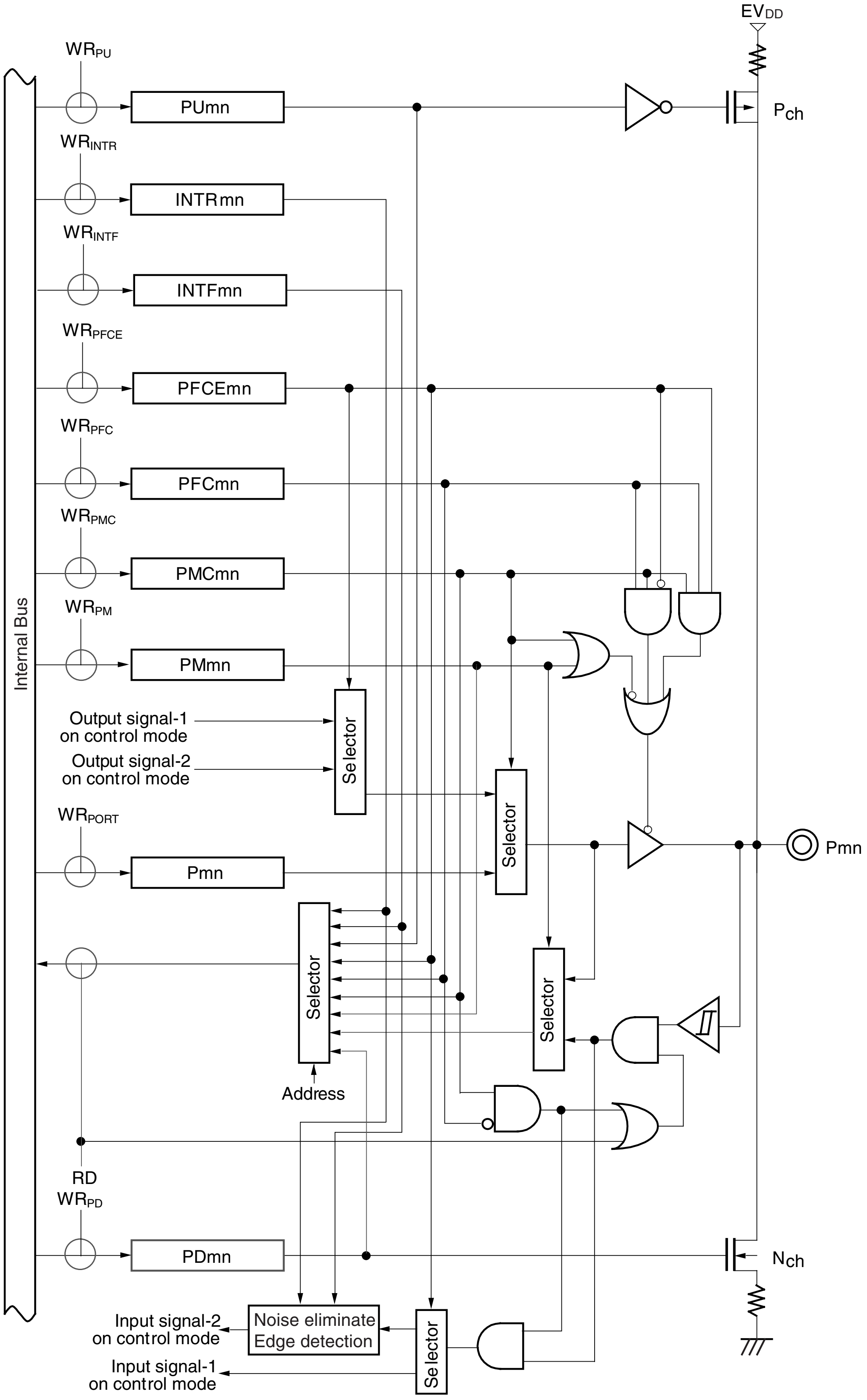
4.6.24 Port block type U-SDW11

Figure 4-92: Type U-SDW11 Block Diagram



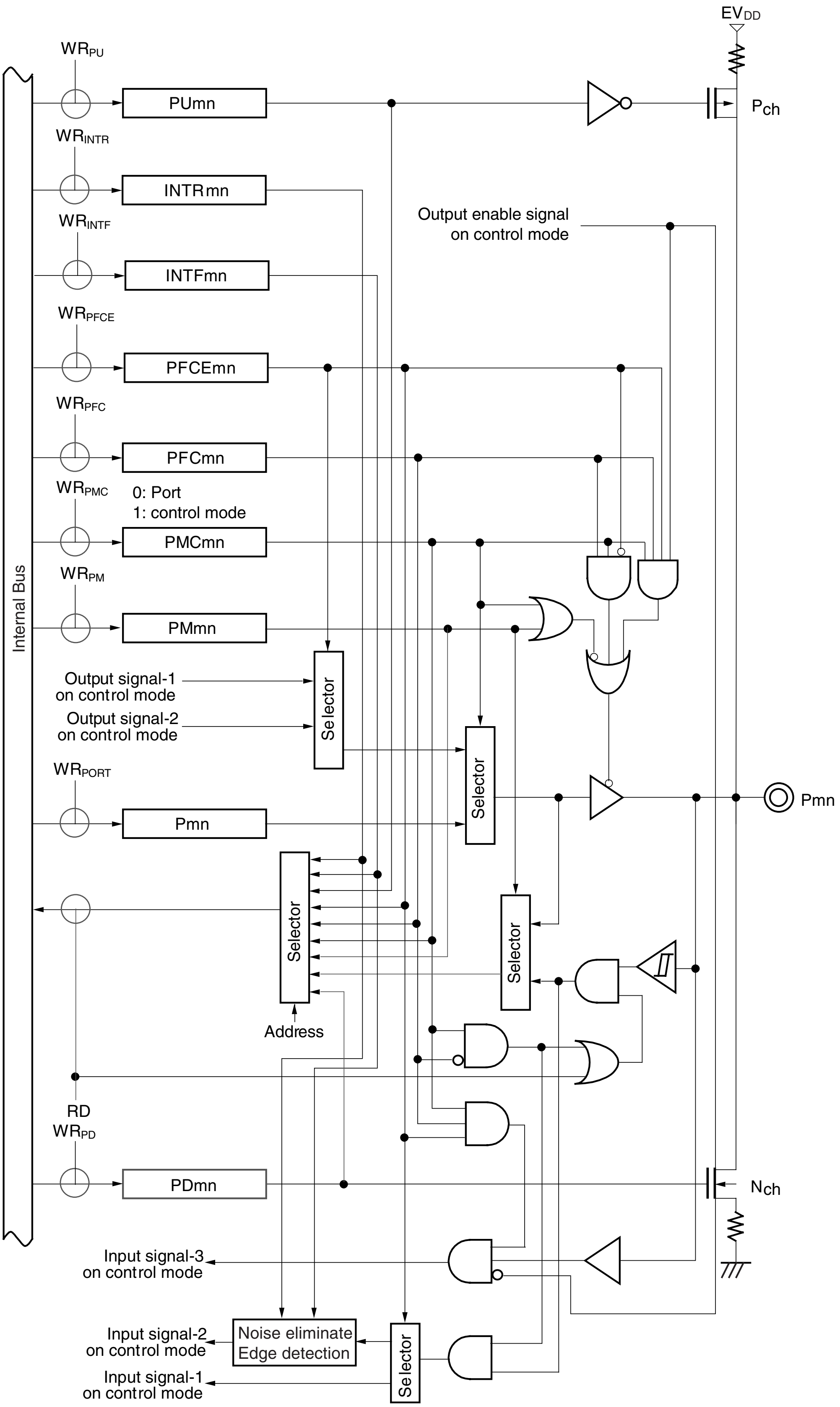
4.6.25 Port block type W-SD11

Figure 4-93: Type W-SD11 Block Diagram



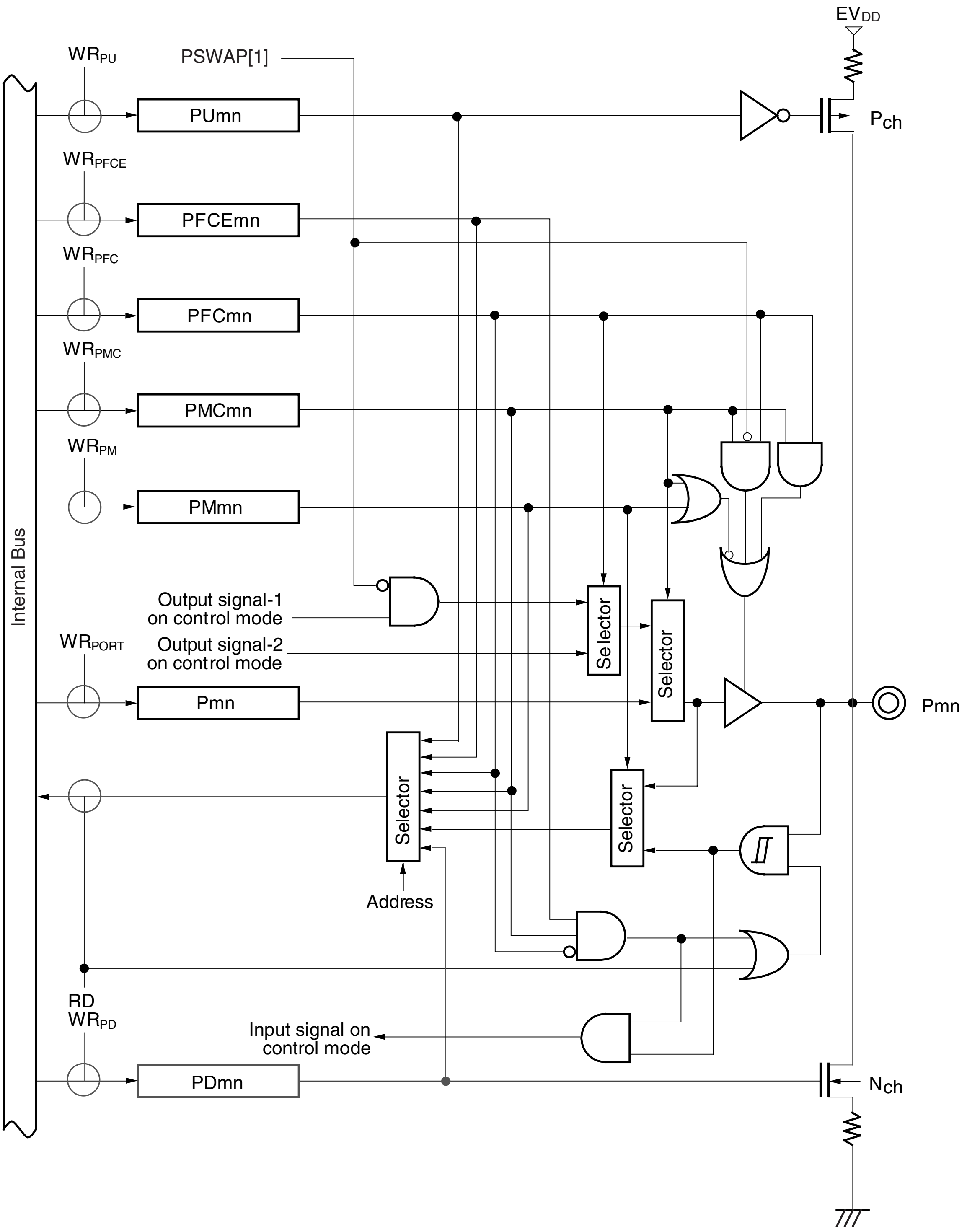
4.6.26 Port block type W-SD12E

Figure 4-94: Type W-SD12E Block Diagram



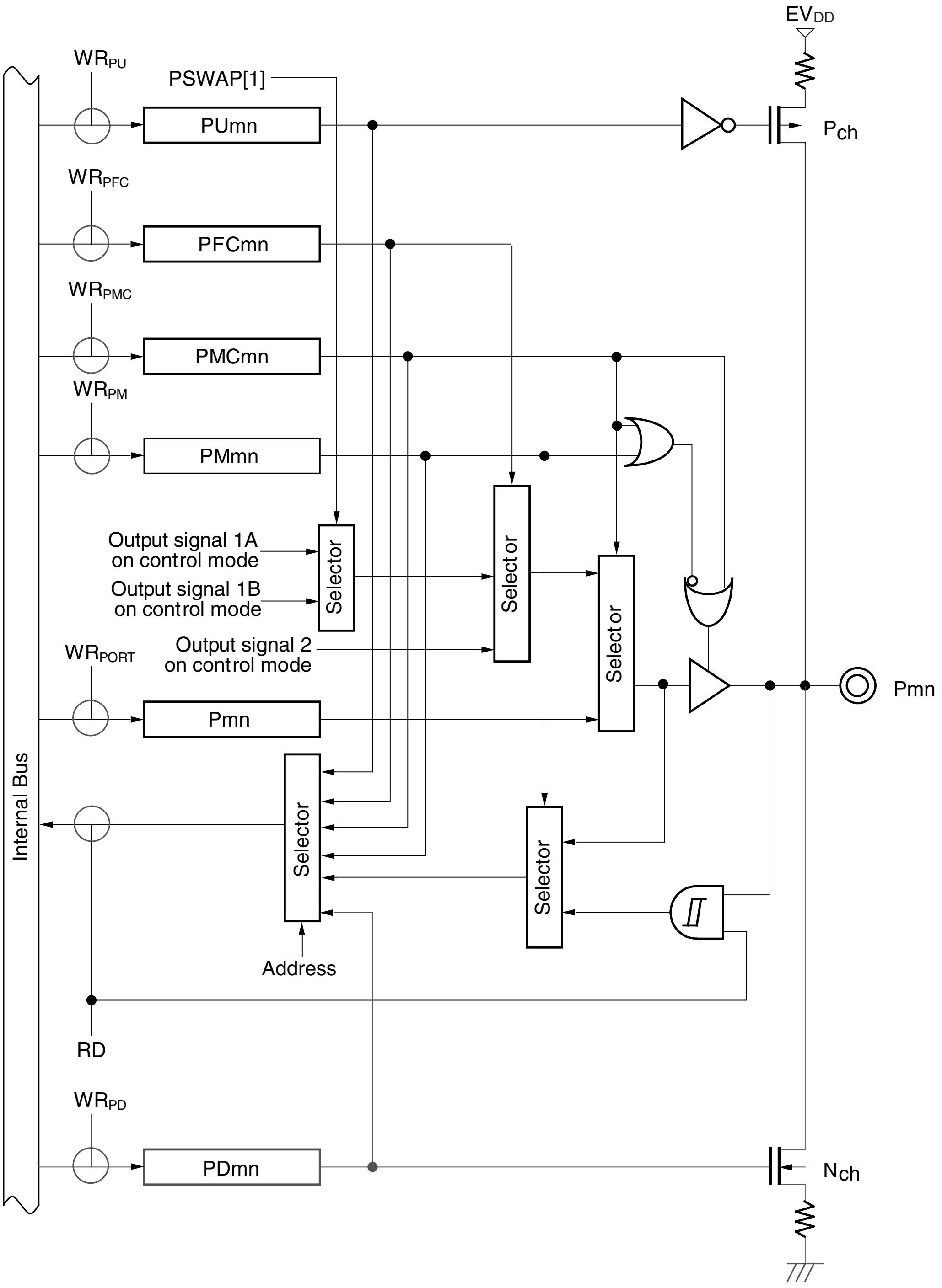
4.6.27 Port block type U-SDW10

Figure 4-95: Type U-SDW10 Block Diagram



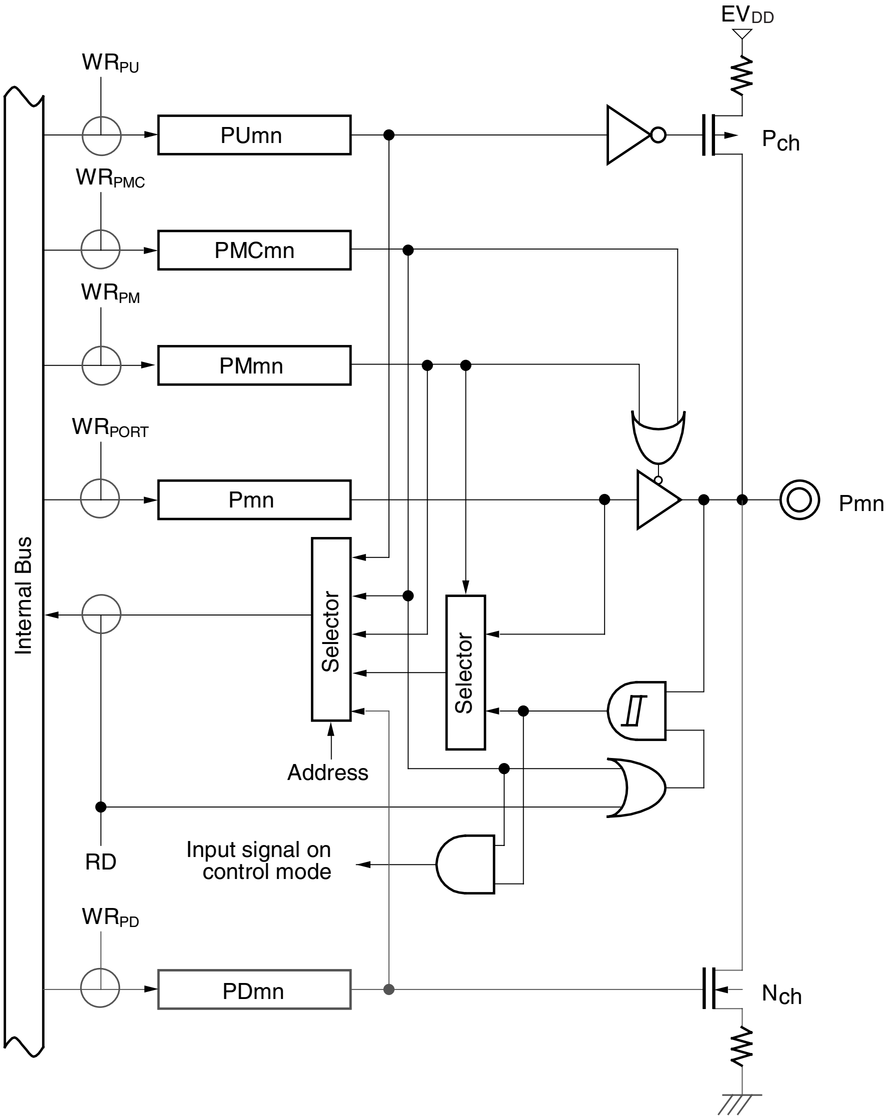
4.6.28 Port Block type G-SDW6

Figure 4-96: Type G-SDW6 Block Diagram



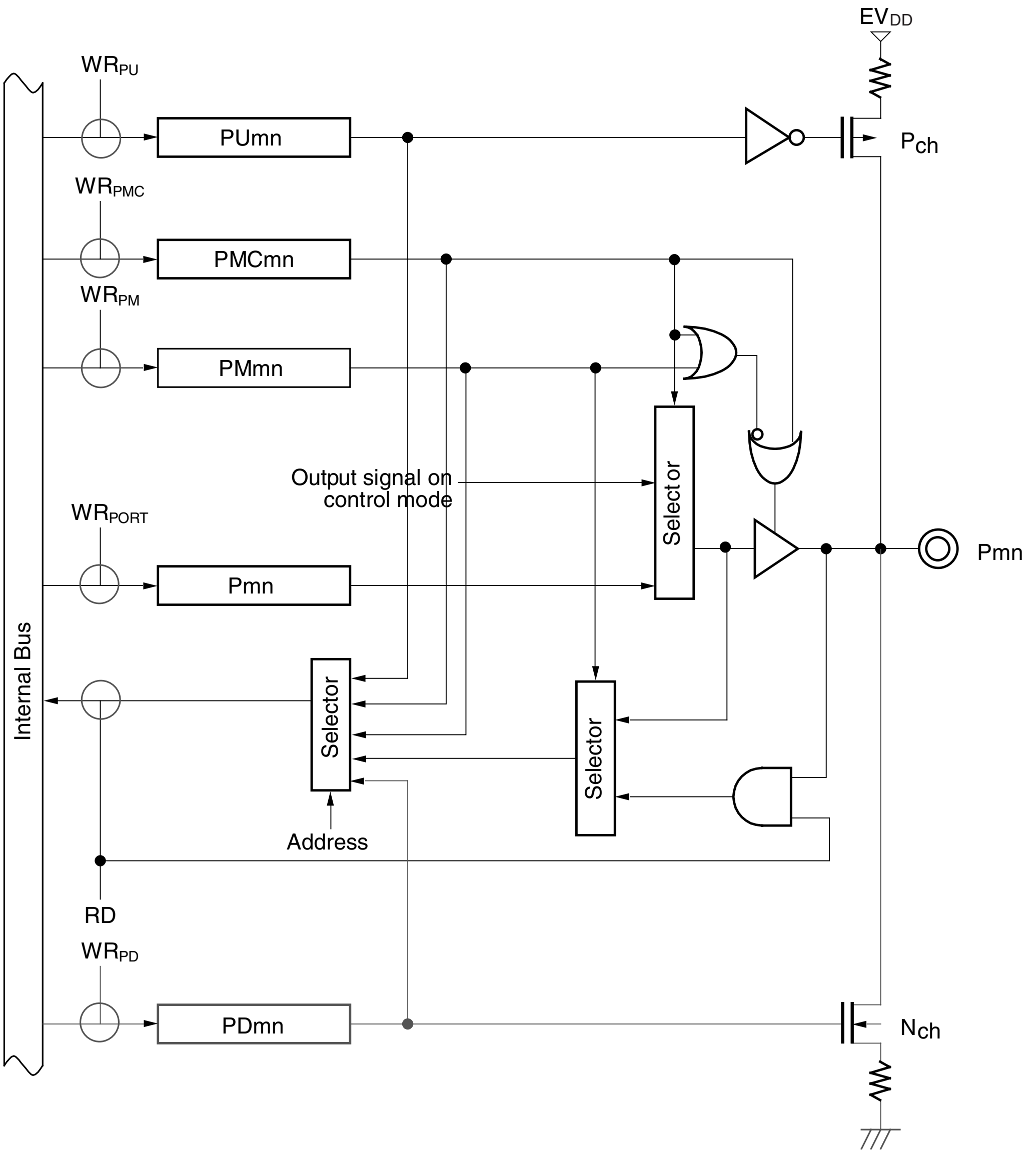
4.6.29 Port block type E-D1

Figure 4-97: Type E-D1 Block Diagram



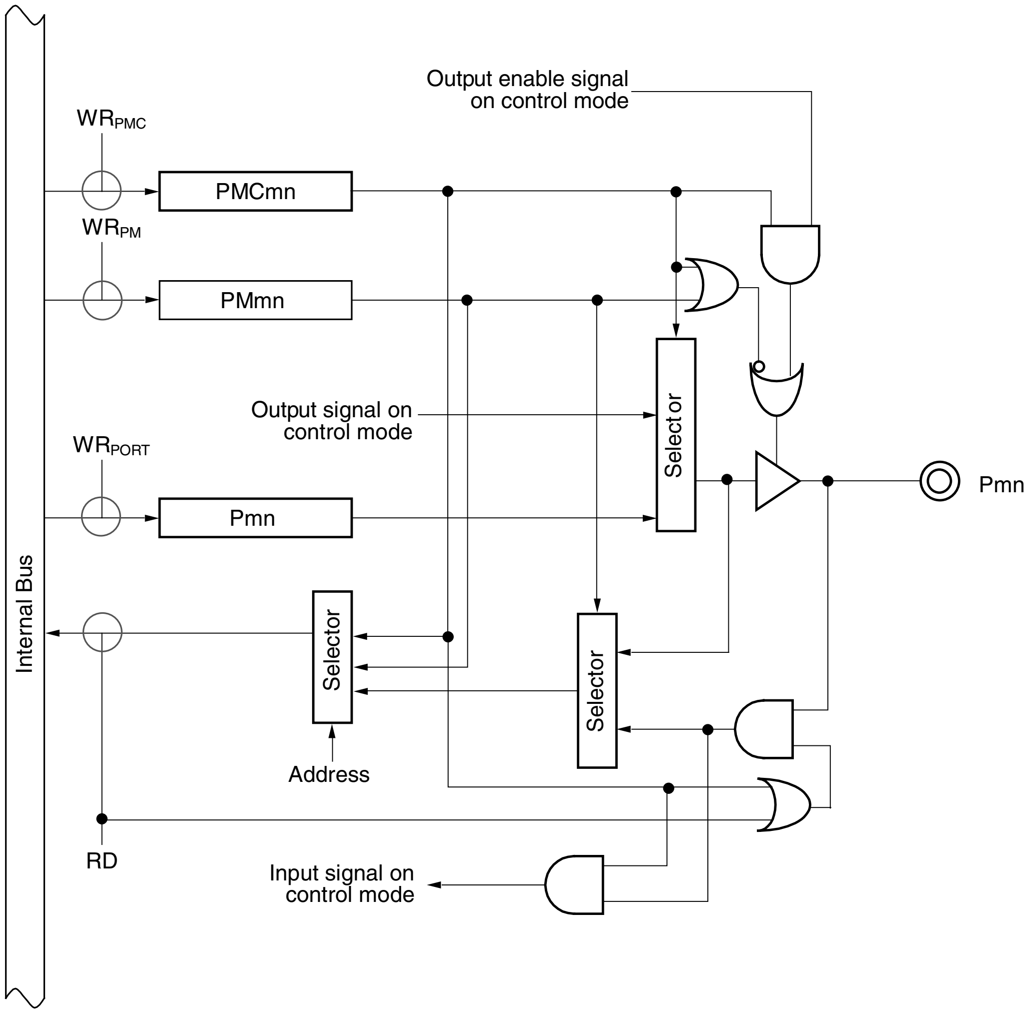
4.6.30 Port block type E-D4

Figure 4-98: Type E-D4 Block Diagram



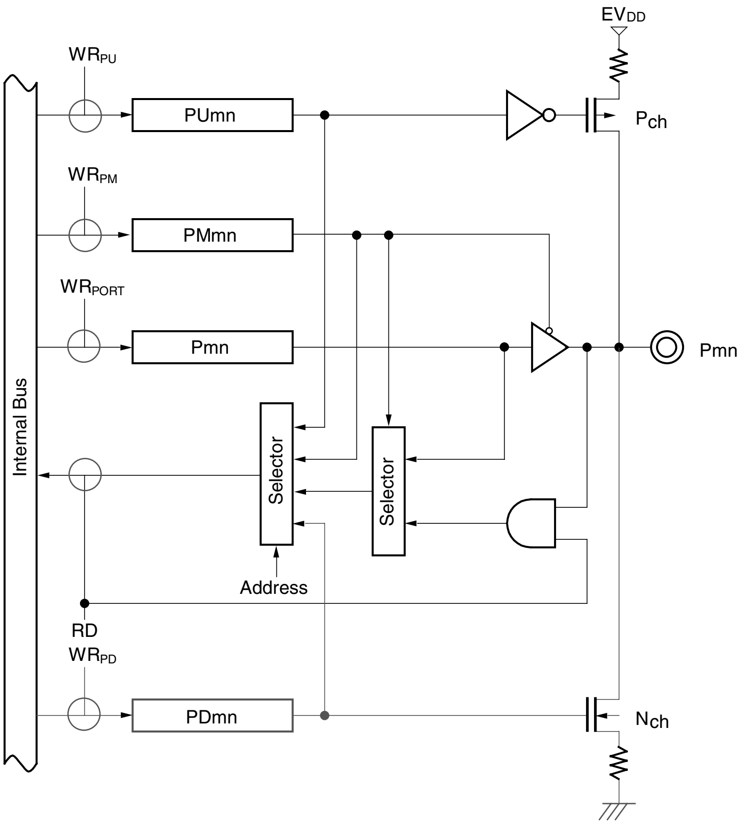
4.6.31 Port block type D-7E

Figure 4-99: Type D-7E Block Diagram



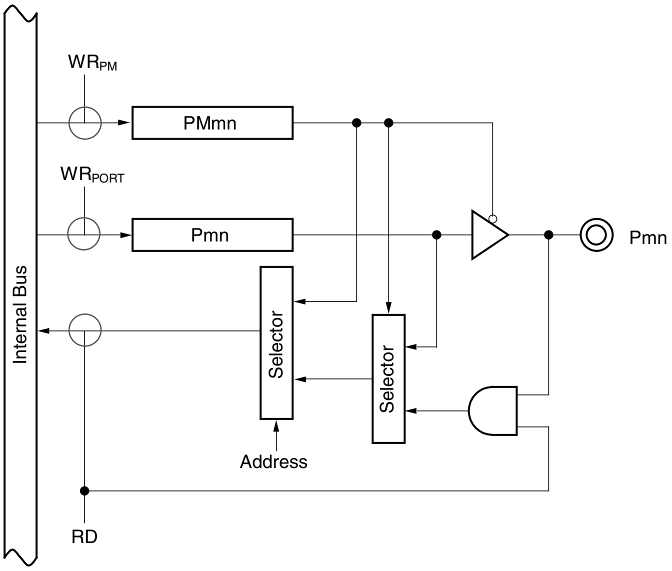
4.6.32 Port block type C-D1

Figure 4-100: Type C-D1 Block Diagram



4.6.33 Port block type B

Figure 4-101: Type B Block Diagram



[MEMO]

Chapter 5 Bus Control Function

The µPD70F3403 and µPD70F3403A are provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

Cautions: 1. Do not set up any external bus interface configuration to the µPD70F3402.

2. External bus interface functions are only described for the µPD70F3403 and µPD70F3403A microcontrollers.

5.1 Features

- Output is selectable from a 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
- Little-Endian format
- address misalign function
- Chip Select output function ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$).

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
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	PCS0, PCS1	Output	Chip select signal
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	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	

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

Table 5-2: Pin Status When Internal ROM, Internal RAM, or Peripheral I/O Is Accessed

Access Destination	Address Bus	Data Bus	Control Signal
AD0 to AD15	undefined	Hi-Z	Inactive

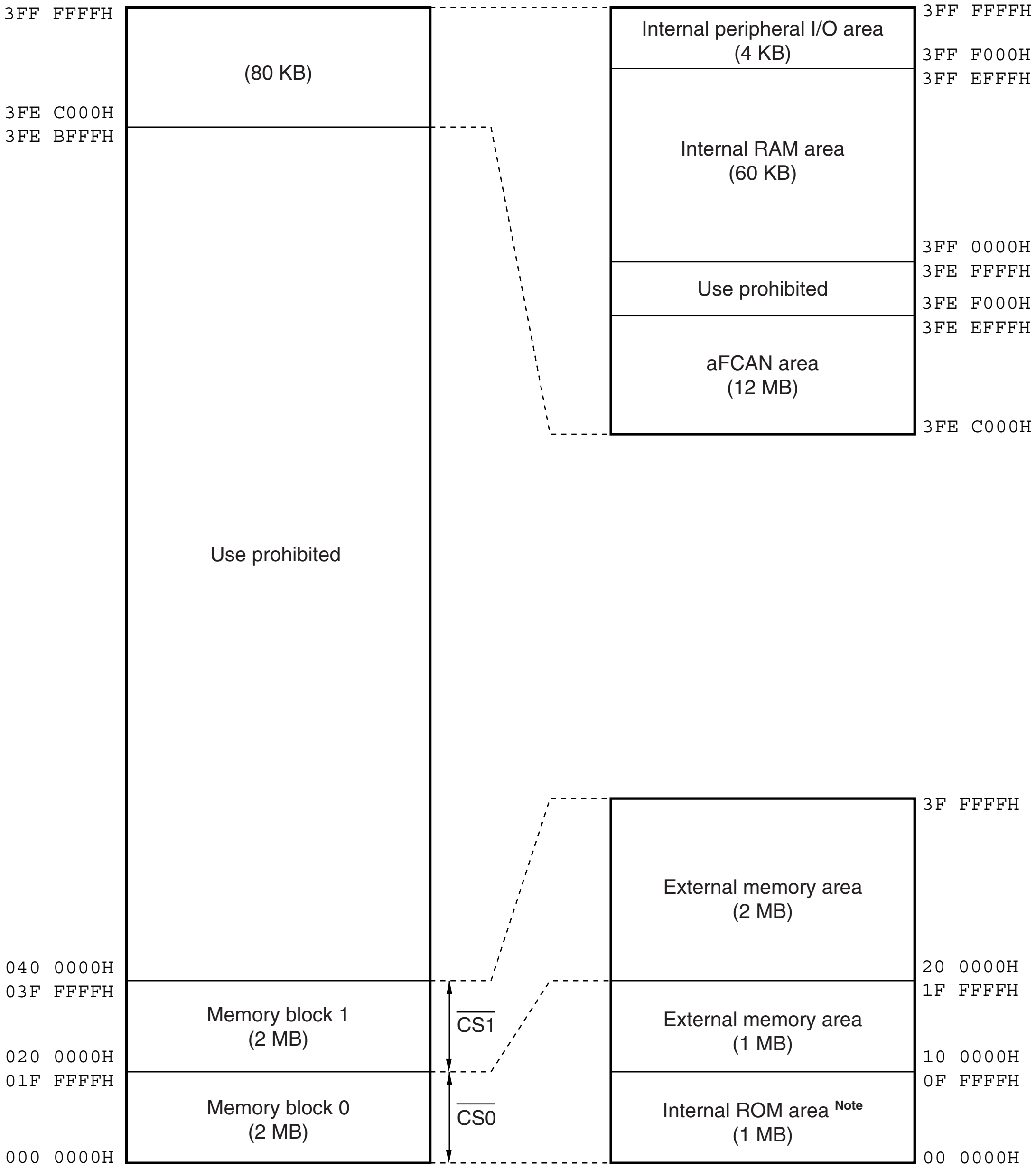
5.2.2 Pin status in each operation mode

For the pin status of the V850E/RS1 in each operation mode, refer to **2.3 ”Description of Pin Functions” on page 45**.

5.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of 2 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



Note: This area is an external memory area in the case of a data write access.

5.3.1 Chip select control function

Of the 64 MB (linear) address space, the lower 4 MB (0000000H to 03FFFFFH) include two chip select control functions, $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$. The areas that can be selected by $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$ are fixed. as shown in Table 5-3. By using these chip select control functions, the memory block can be divided to enable effective use of the memory space.
However, since the V850E/RS1 has sixteen address pins (PDL0/AD0 to PDL15/AD15), 64 KB addresses can be selected linearly.

Table 5-3: Allocation of the Memory Blocks

	V850E/RS1
$\overline{\text{CS0}}$	0000000H to 01FFFFFH (2 MB)
$\overline{\text{CS1}}$	0200000H to 03FFFFFH (2 MB)

5.4 Bus Access

5.4.1 Number of clocks for access

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

Area (Bus Width) Area (Bus Width) Bus Cycle Type	Internal ROM (32 bits)	Internal RAM (32 bits)	External Memory (16 bits)
Instruction fetch (normal access)	1	1[Note 1]	3 + n [Note 2]
Instruction fetch (branch)	2	1	3 + n [Note 2]
Operand data access	3	1	3 + n [Note 2]

Notes: 1. 2 if a conflict with a data access occurs.

2. n: Number of wait states.

Remark: Unit: Clocks/access

5.4.2 Bus size setting function

The bus size of each external memory area selected by $\overline{\text{CSn}}$ can be set (to 8 bits or 16 bits) by using the BSC register.
The external memory area of the V850E/RS1 (0000000H to 03FFFFFH) is selected by $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$.

(1) Bus size configuration register (BSC)

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

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

Figure 5-2: Bus Size Configuration Register (BSC) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
BSC	0	1	0	1	0	1	0	1	FFFFF066H	5555H
R/W										
	7	6	5	4	3	2	1	0		
	0	1	0	1	0	BS10	0	BS00		
$\overline{\text{CSn}}$	signal					$\overline{\text{CS1}}$		$\overline{\text{CS0}}$		

BSn0	Data bus width of CSn space (n = 0 to 1)
0	8 bits
1	16 bits

Caution: Be sure to set bits 14, 12, 10, 8, 6, 4 to 1, and clear bits 15, 13, 11, 9, 7, 5, 3, 1 to 0.

5.4.3 Access by bus size

The V850E/RS1 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 V850E/RS1 supports only the Little-Endian format.

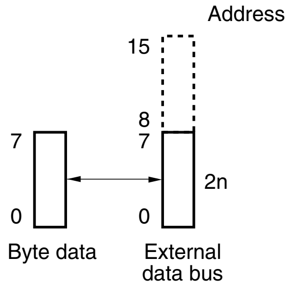
Figure 5-3: Little-Endian Address in Word

31 24	23 16	17 8	7 0
000BH	000AH	0009H	0008H
0007H	0006H	0005H	0004H
0003H	0002H	0001H	0000H

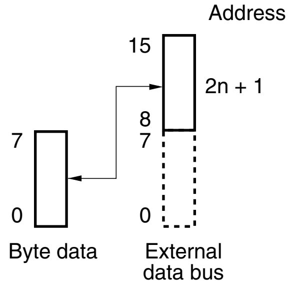
(1) Byte access (8 bits)

(a) 16-bit data bus width

<1> Access to even address (2n)

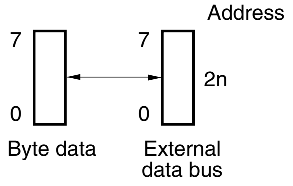


<2> Access to odd address (2n + 1)

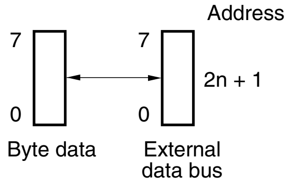


(b) 8-bit data bus width

<1> Access to even address (2n)



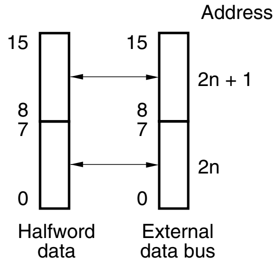
<2> Access to odd address (2n + 1)



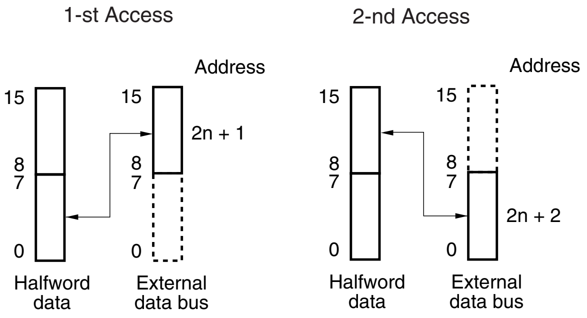
(2) Halfword access (16 bits)

(a) With 16-bit data bus width

<1> Access to even address (2n)

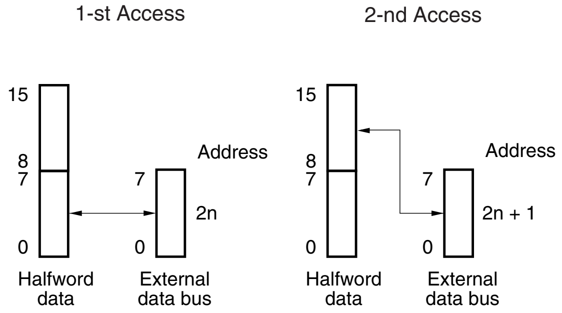


<2> Access to odd address (2n + 1)

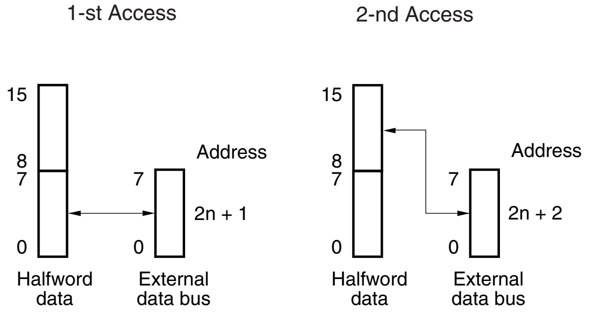


(b) 8-bit data bus width

<1> Access to even address (2n)



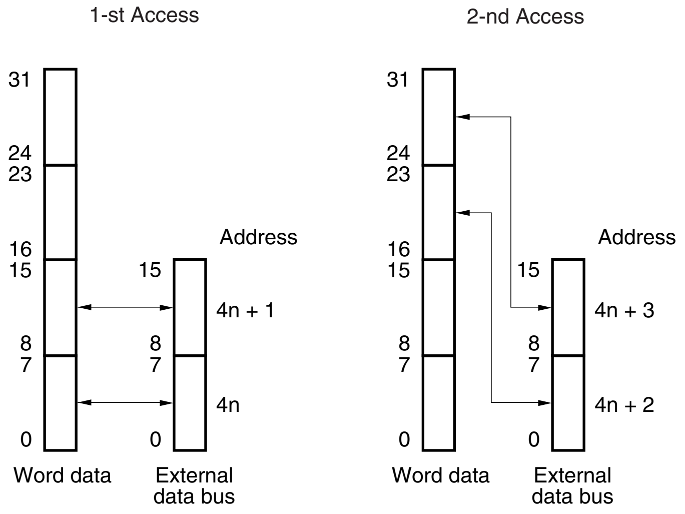
<2> Access to odd address (2n + 1)



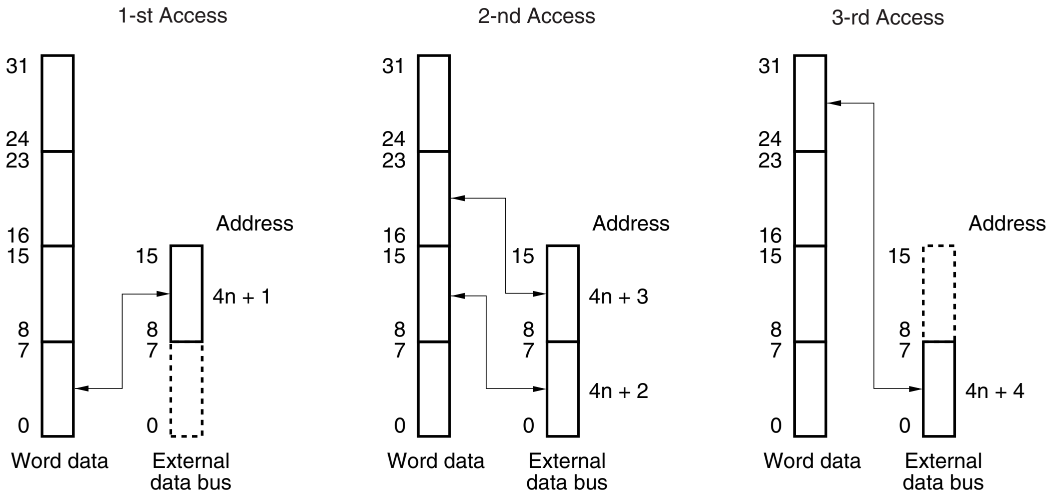
(3) Word access (32 bits)

(a) 16-bit data bus width (1/2)

<1> Access to address (4n)

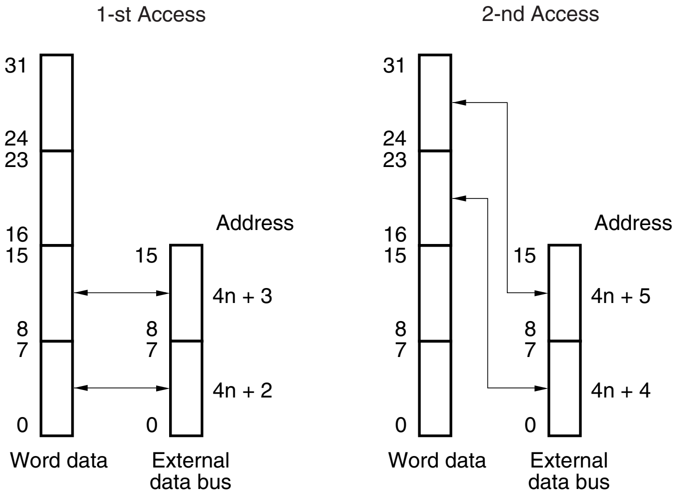


<2> Access to address (4n + 1)

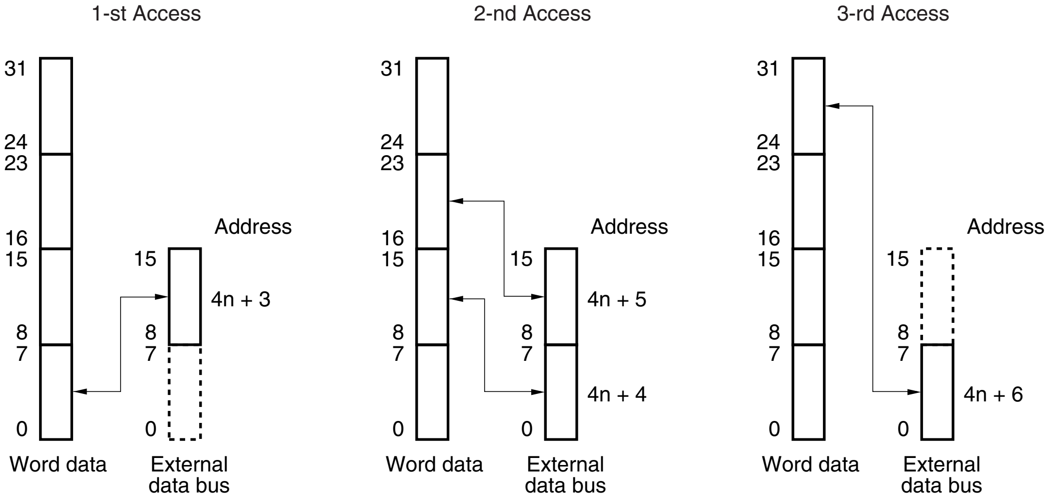


(a) 16-bit data bus width (2/2)

<3> Access to address (4n + 2)

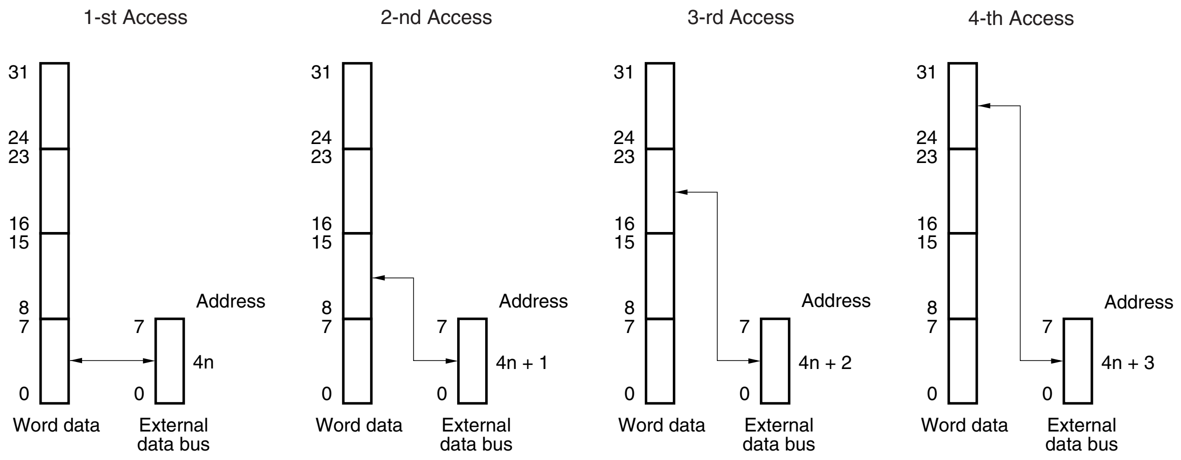


<4> Access to address (4n + 3)

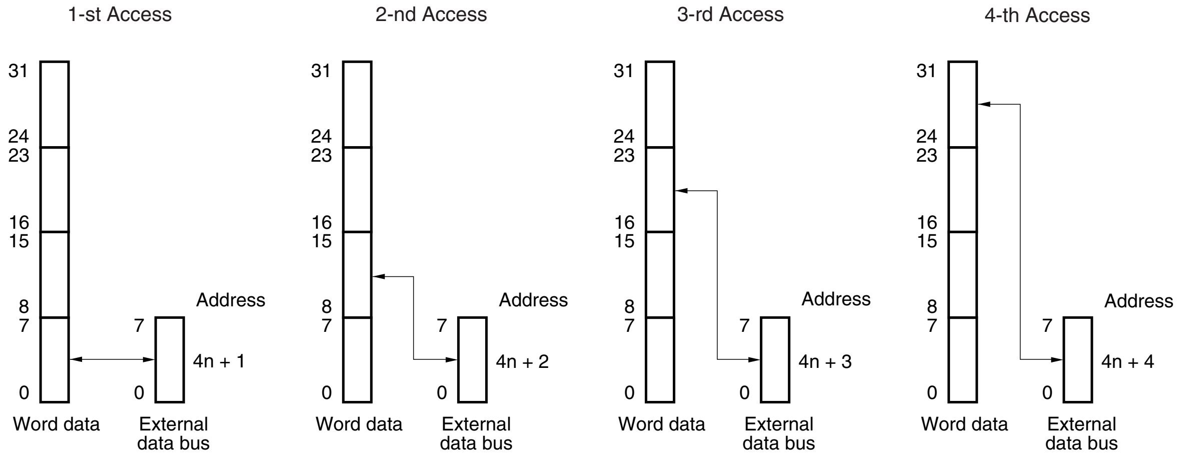


(b) 8-bit data bus width (1/2)

<1> Access to address (4n)

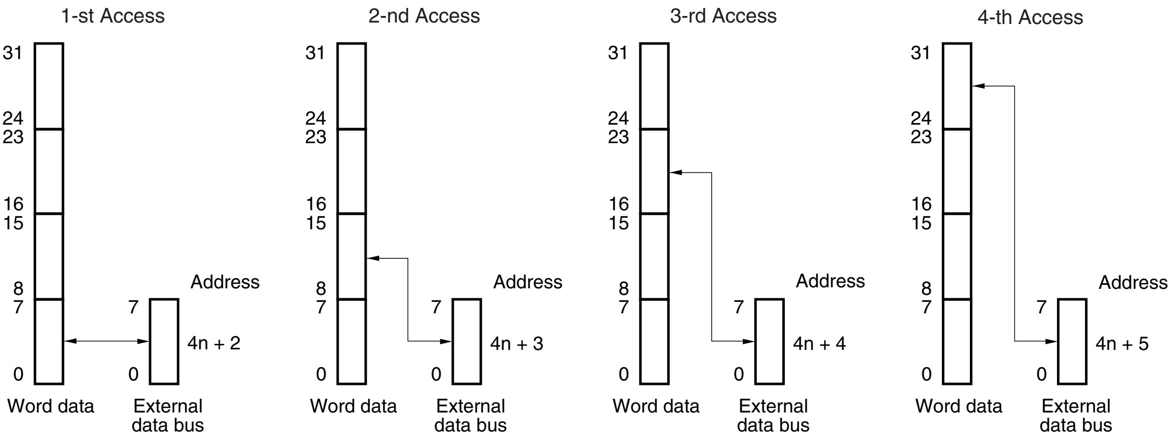


<2> Access to address (4n + 1)

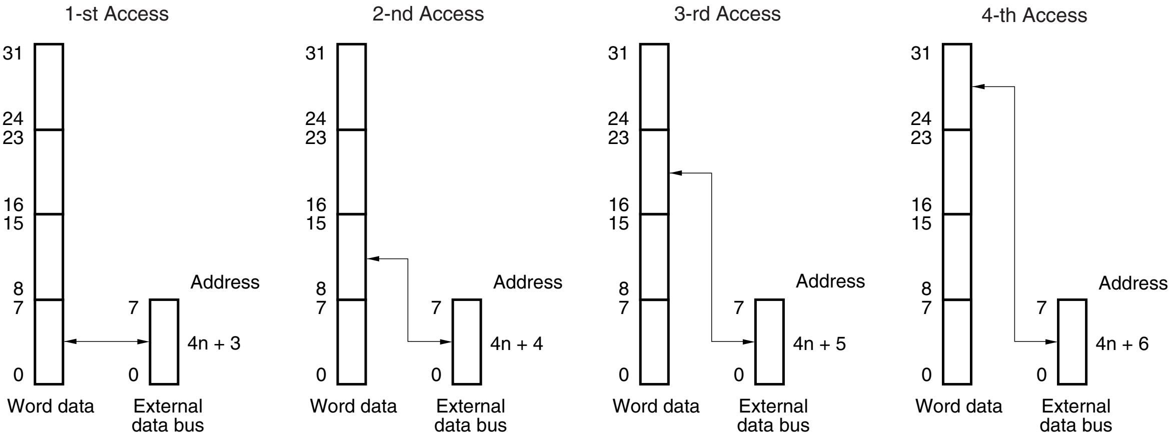


(b) 8-bit data bus width (2/2)

<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 for each chip select area ($\overline{CS0}$, $\overline{CS1}$) by using data wait control register 0 (DWC0). 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.

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 other than the one for this initialization routine until the initial settings of the DWC0 register are complete. However, external memory areas whose initial settings are complete may be accessed.

Figure 5-4: Data Wait Control Register 0 (DWC0) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
DWC0	0	1	1	1	0	1	1	1	FFFFF484H	7777H
R/W										

	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
$\overline{CSn}$ signal		$\overline{CS1}$				$\overline{CS0}$		

DWn2	DWn1	DWn0	Number of wait states inserted in memory block n space (n = 0, 1)
0	0	0	None
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

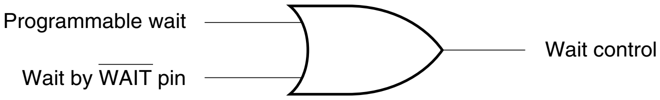
Caution: Be sure to set to 1 bits 14 to 12 and bits 10 to 8.
Be sure to clear to 0 bits 15, 11, 7, and 3.

5.5.2 External wait function

To synchronize an extremely slow external device, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).
Access to each area of the internal ROM, internal RAM, and on-chip peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.
The $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT, and is sampled at the falling edge of the clock in the T2 and TW states of the bus cycle in the multiplexed bus mode. In the separate bus mode, it is sampled at the rising edge of the clock immediately after the T1 and TW states of the bus cycle. If the setup/hold time of the sampling timing is not satisfied, a wait state is inserted in the next state, or not inserted at all.

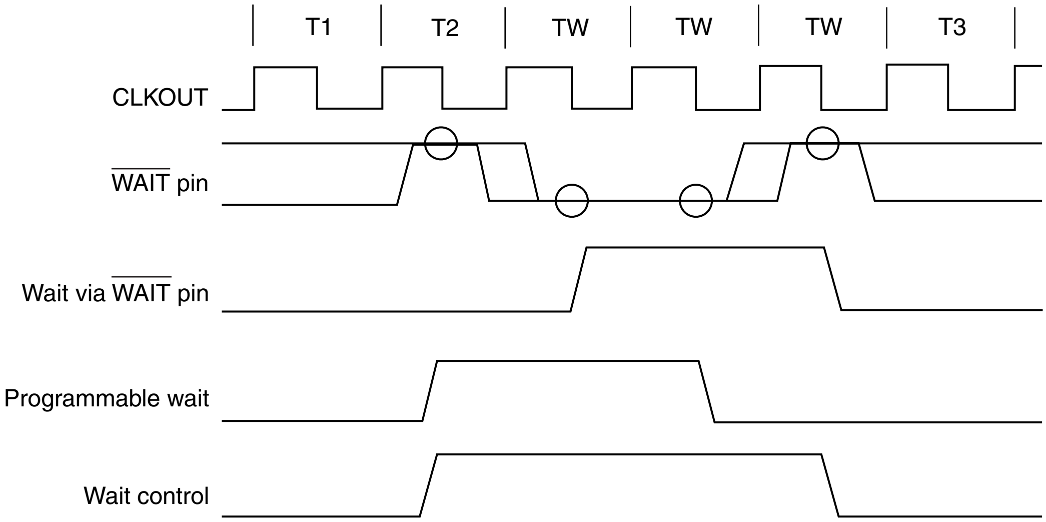
5.5.3 Relationship between programmable wait and external wait

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. In other words, the number of wait cycles is determined by the side with the greatest number of cycles.



For example, if the timing of the programmable wait and the $\overline{\text{WAIT}}$ pin signal is as illustrated below, three wait states will be inserted in the bus cycle.

Figure 5-5: Example of Inserting Wait States in Separate Bus Mode


T1
T2
TW
TW
TW
T3
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait via $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control

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 address wait control register (AWC). Address wait insertion is set for each chip select area ($\overline{CS0}$, $\overline{CS1}$).
If an address setup wait is inserted, it seems that the high-clock period of T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

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

Figure 5-6: Address Wait Control Register (AWC) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
AWC	1	1	1	1	1	1	1	1	FFFFF488H	FFFFH
R/W										
	7	6	5	4	3	2	1	0		
	1	1	1	1	AHW1	ASW1	AHW0	ASW0		
$\overline{CSn}$	signal				$\overline{CS1}$		$\overline{CS0}$			

AHWn	Specifies insertion of address hold wait (n = 0, 1)
0	Not inserted
1	Inserted

ASWn	Specifies insertion of address hold wait (n = 0, 1)
0	8 bits
1	16 bits

Caution: Be sure to set bits 15 to 4 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 in the multiplexed address/data bus mode. In the separate bus mode, one idle state (TI) can be inserted after the T2 state. 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 bus cycle control register (BCC).

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control register (BCC)

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

Cautions: 1. The internal ROM, internal RAM, and internal 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 other than the one for this initialization routine until the initial settings of the BCC register are complete. However, external memory areas whose initial settings are complete may be accessed.

Figure 5-7: Bus Cycle Control Register (BCC) Format

Symbol	15	14	13	12	11	10	9	8	Address	After reset
BCC	1	0	1	0	1	0	1	0	FFFFF48AH	AAAAH
R/W										

	7	6	5	4	3	2	1	0
	1	0	1	0	BC11	0	BC01	0
$\overline{\text{CSn}}$ signal					$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

BCn1	Specifies insertion of idle state (n = 0, 1)
0	Not inserted
1	Inserted

Caution: Be sure to set bits 15, 13, 11, 9, 7, and 5 to 1, and clear bits 14, 12, 10, 8, 6, 4, 2, and 0 to 0.

5.7 Bus Hold Function

5.7.1 Functional outline

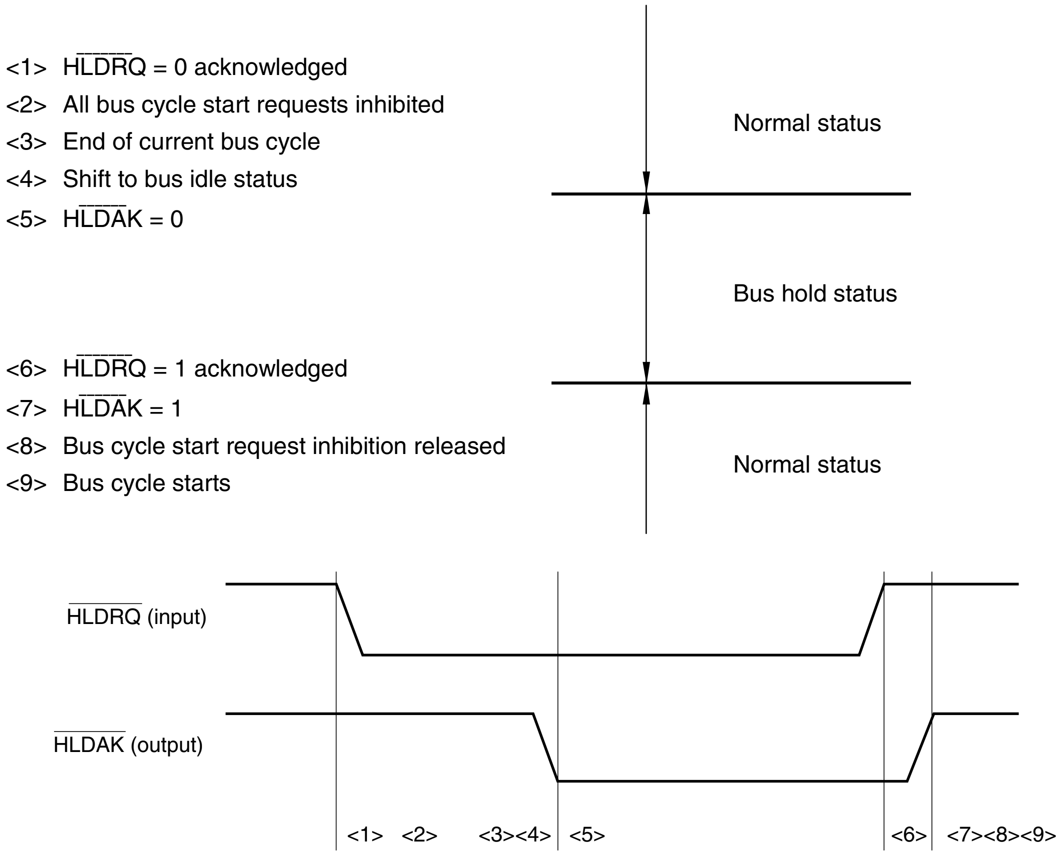
The HLDAK and HLDRQ functions are valid if the PCM2 and PCM3 pins are set in the control mode.
When the HLDRQ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus goes into a high-impedance state and is released (bus hold status). If the request for the bus mastership is cleared and the 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 a peripheral I/O register or the external memory is accessed.
The bus hold status is indicated by assertion of the HLDAK pin (low level). The bus hold function enables the configuration 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 in 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.

Figure 5-8: Bus Hold Status Transition Procedure



<1> $\overline{\text{HLDRQ}}$ = 0 acknowledged
<2> All bus cycle start requests inhibited
<3> End of current bus cycle
<4> Shift to bus idle status
<5> $\overline{\text{HLDAK}}$ = 0

<6> $\overline{\text{HLDRQ}}$ = 1 acknowledged
<7> $\overline{\text{HLDAK}}$ = 1
<8> Bus cycle start request inhibition released
<9> Bus cycle starts

5.7.3 Operation in power save mode

Because the internal system clock is stopped in the software STOP and IDLE modes, the bus hold status is not entered even if the $\overline{\text{HLDRQ}}$ pin is asserted.
In the HALT mode, the $\overline{\text{HLDAK}}$ pin is asserted as soon as the $\overline{\text{HLDRQ}}$ pin 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, instruction fetch (branch), instruction fetch (successive), and operand data accesses are executed in the external bus cycle.

Bus hold has the highest priority, followed by 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 5-4: Bus Priority

Priority	External Bus Cycle	Bus Master
High	Bus hold	External device
↕	DMA transfer	DMAC
	Operand data access	CPU
	Instruction fetch (branch)	CPU
Low	Instruction fetch (successive)	CPU

5.9 Boundary Operation Conditions

5.9.1 Program space

(1) If a branch instruction exists at the upper limit of the internal RAM area, a prefetch operation straddling over the internal peripheral I/O area (invalid fetch) does not occur.

(2) Instruction execution to the external memory area cannot be continued without a branch from the internal ROM area to the external memory area.

5.9.2 Data space

The V850E/RS1 has an address misalign function.

With this function, data can be placed at all addresses, regardless of the format of the data (word data or halfword data). However, if the word data or halfword data is not aligned at the boundary, a bus cycle is generated at least twice, causing the bus efficiency to drop.

(1) Halfword-length data access

A byte-length bus cycle is generated twice if the least significant bit of the address is 1.

(2) Word-length data access

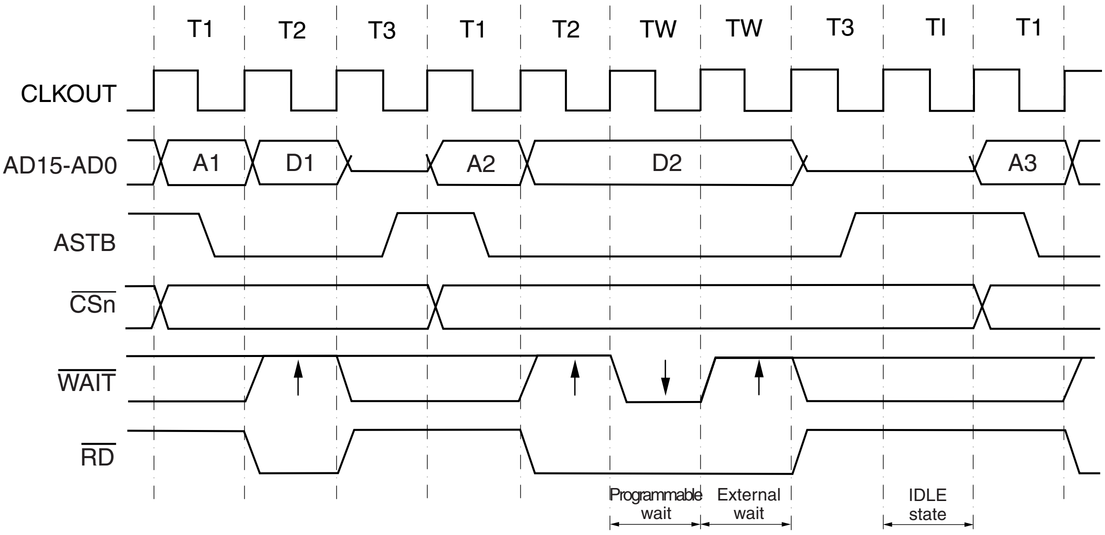
(a) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(b) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

5.10 Bus Timing

(1) Read cycle

Figure 5-9: Bus Read Timing (Bus Size: 16 bit, 16-bit Access)



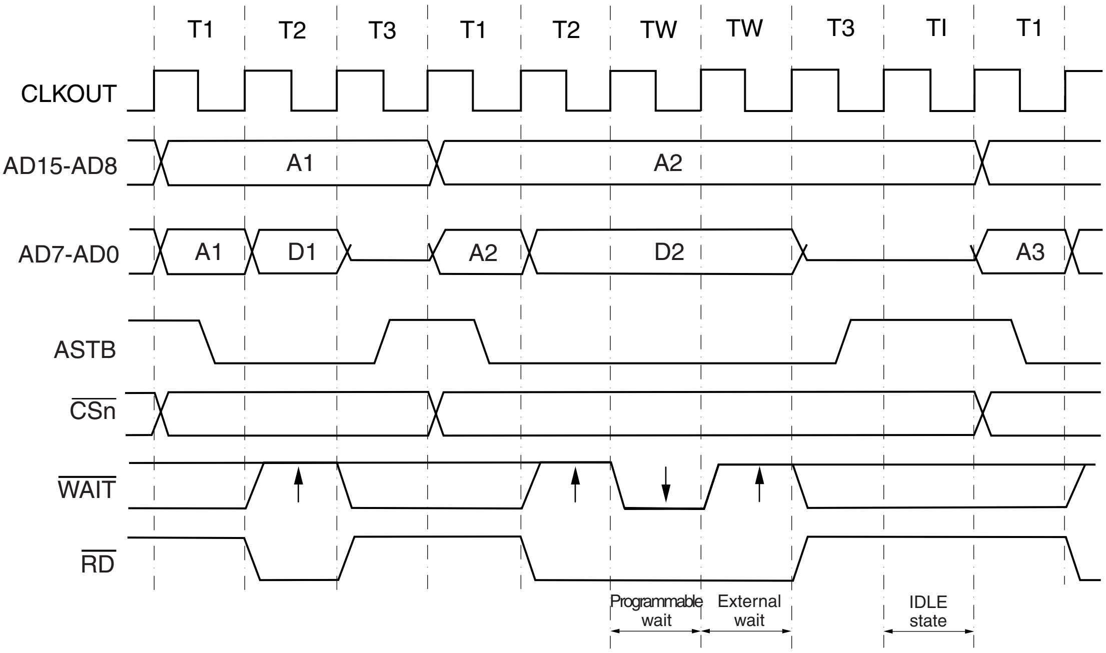
In the case of 8-bit access	Odd address	Even address
AD15 to AD8	Data	-
AD7 to AD0	-	Data

Note: AD15-8 hold the data when we access to odd address with 8-bit access.

Remarks: 1. The circles indicate the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

Figure 5-10: Bus Read Timing (Bus Size: 8 bit)

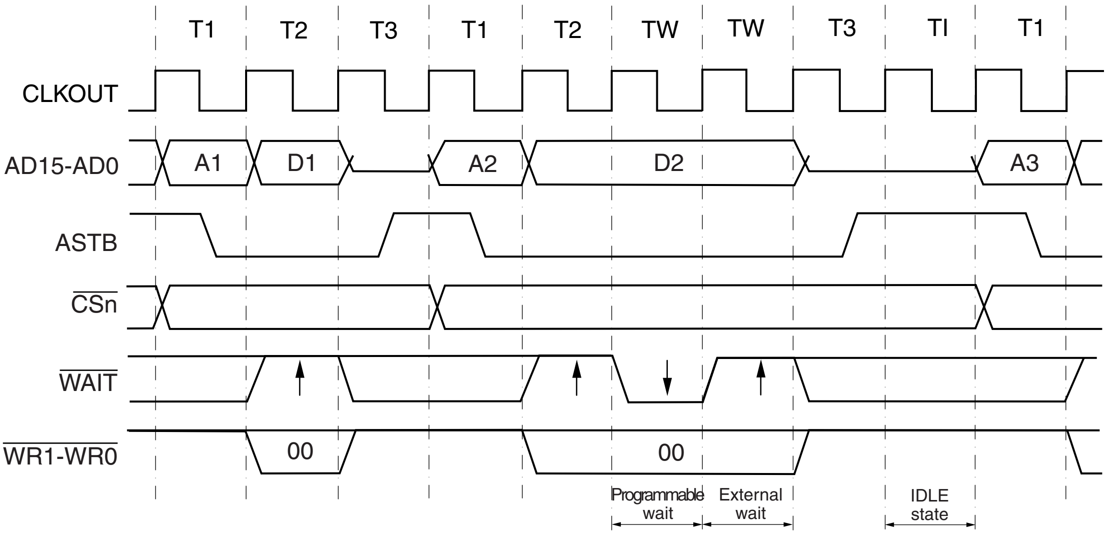


Remarks: 1. The circles indicate the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

(2) Write cycle

Figure 5-11: Bus Write Timing (Bus Size: 16 bit, 16-bit Access)



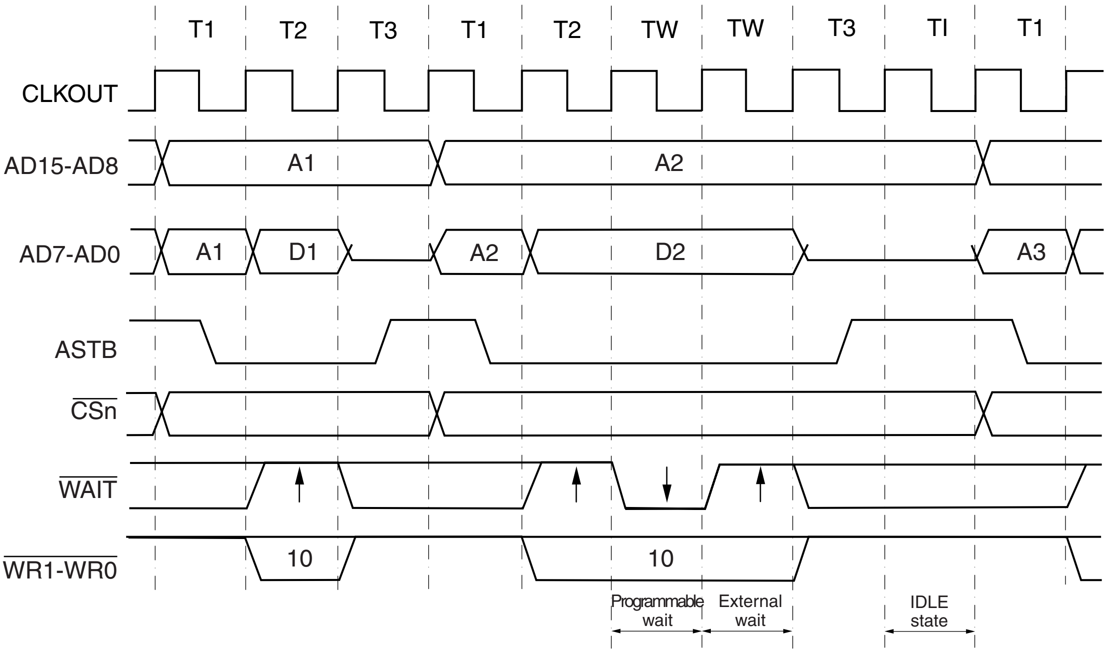
In the case of 8-bit access	Odd address	Even address
AD15 to AD8	Data	Undefined
AD7 to AD0	Undefined	Data
$\overline{\text{WR1}}$, $\overline{\text{WR0}}$	01	10

Note: $\overline{\text{WR0}}$ and $\overline{\text{WR1}}$ output a low level as shown in the above timing chart when target data access is performed. At all other times, these pins output a high level.

Remarks: 1. The circles indicate the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

Figure 5-12: Bus Write Timing (Bus Size: 8 bit)

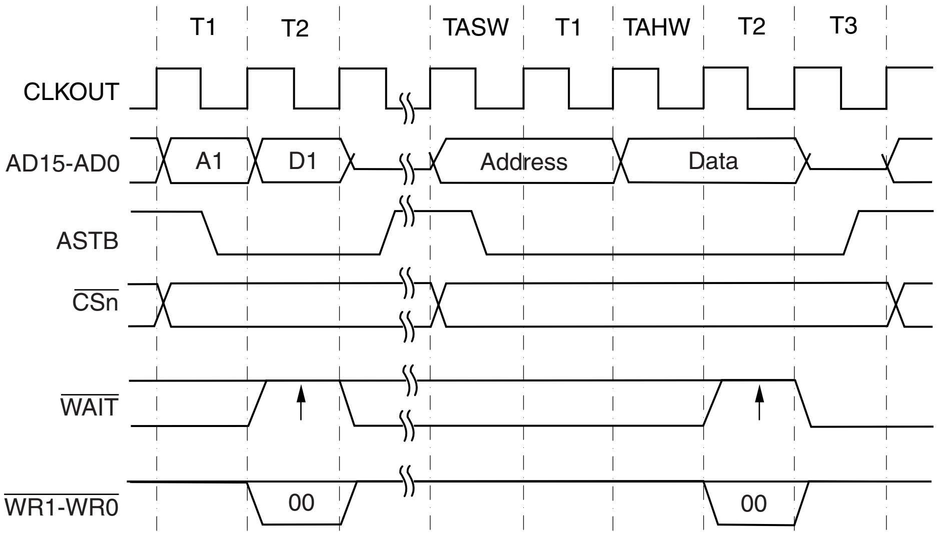


Note: $\overline{\text{WR0}}$ and $\overline{\text{WR1}}$ output a low level as shown in the above timing chart when target data access is performed. At all other times, these pins output a high level.

Remarks: **1.** The circles indicate the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

Figure 5-13: Address Wait Timing (Bus Size: 16 bit)



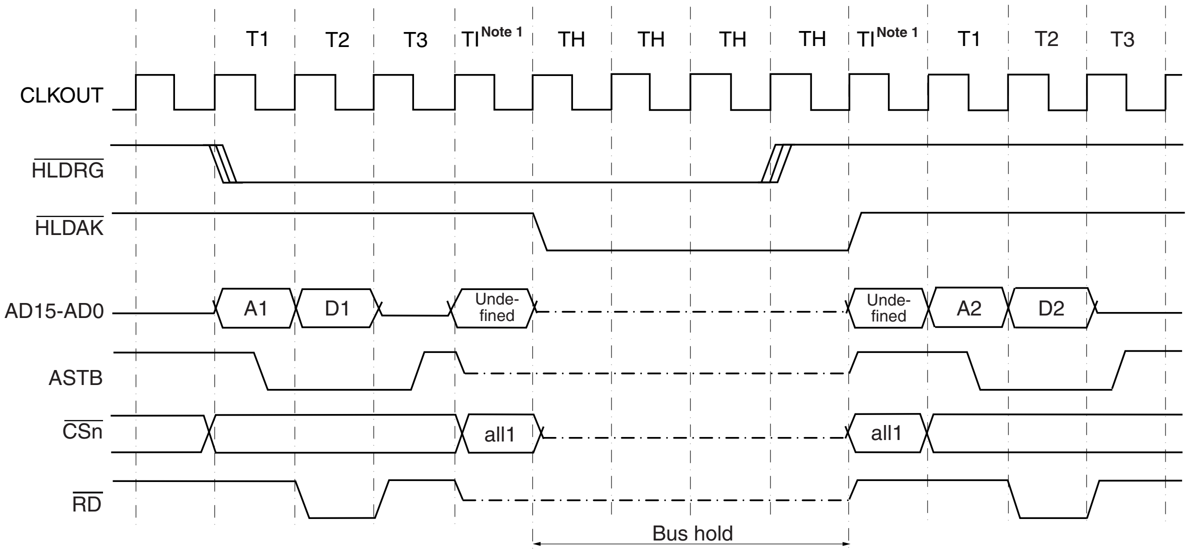
Note: WR0 and WR1 output a low level as shown in the above timing chart when target data access is performed. At all other times, these pins output a high level.

Remarks: 1. The circle indicates the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

(3) Bus hold cycle

Figure 5-14: Bus Hold Timing (Bus Size: 16 bit)



Notes: 1. Upon detection of a low level in the T2 and T3 states of HLDRQ (sampling timing), the operation moves on to the bus hold cycle after the T3 state ends. Thereafter, upon detection of a low level or high level in the TH state (sampling timing), the bus hold status is maintained after the TH state ends, or the bus cycle is restarted.

2. WR0 and WR1 output a low level as shown in the above timing chart when target data access is performed. At all other times, these pins output a high level.

3. This idle state (TI) does not depend on the BCC register settings.

Remarks: 1. The circles indicate the sampling timing when 0 is set for the programmable wait.

2. The broken line indicates high impedance.

[MEMO]

Chapter 6 Clock Generator

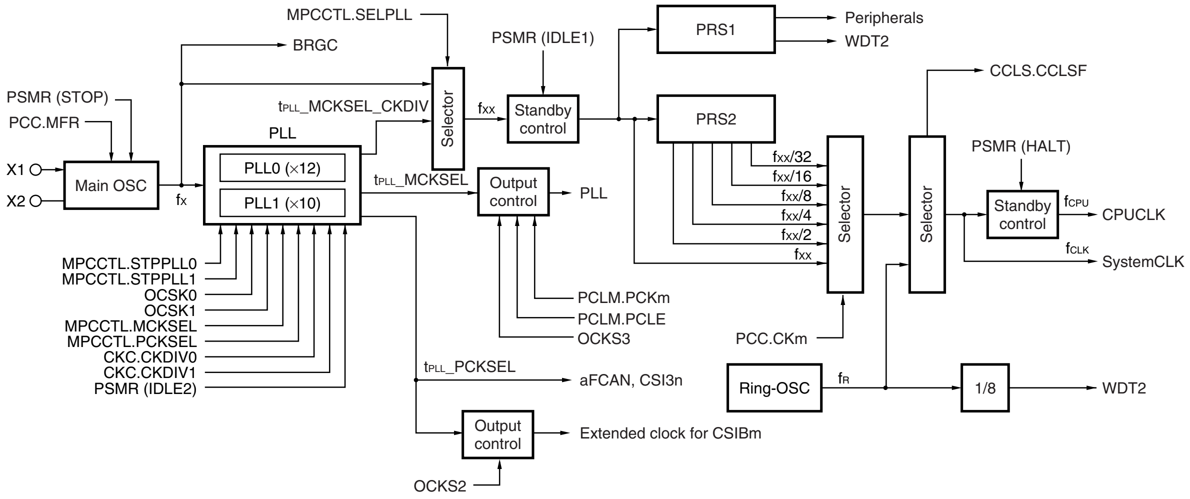
6.1 Overview

The following clock generation functions are available.

- Main clock oscillator
  - In clock-through mode
    f_X = 4 to 8 MHz (internal f_{XX} = 4 to 8 MHz)
  - In PLL mode
    f_X = 4 to 8 MHz (internal f_{XX} = 4 to 40 MHz (µPD70F3403 and µPD70F3403A)
    = 4 to 32 MHz (µPD70F3402))
- Multiply function by PLL (Phase Locked Loop)
  - Clock-through mode/PLL mode selectable
- Ring OSC
  - f_R = 100 to 400 kHz
- Internal system clock generation
  - 7 steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{RING})
- Peripheral clock generation
- Clock output function
- Programmable clock output function

6.2 Configuration

Figure 6-1: Clock Generator



(1) Main clock oscillator (Main OSC)

The main resonator oscillates the following frequencies (f_X).

- In clock-through mode
  f_X = 4 to 8 MHz (internal f_{XX} = 4 to 8 MHz)
- In PLL mode
  f_X = 4 to 8 MHz (internal f_{XX} = 4 to 40 MHz (µPD70F3403 and µPD70F3403A)
  = 4 to 32 MHz (µPD70F3402))

(2) PLL

This circuit multiplies the clock generated by the main clock oscillator (f_X).
It operates in two modes: clock-through mode in which f_X is output as is, and PLL mode in which a multiplied clock is output. These modes can be selected by using the SELPLL bit of the Main peripheral clock control register (MPCCTL). Operation of the PLL can be started or stopped by the STPPLL bit of MPCCTL register.

(3) Ring-OSC

Outputs a frequency (f_R) of 100 to 400 kHz.

(4) Prescaler 1 (PRS1)

This prescaler divides the clock to be supplied to the WDT2 and the on-chip peripheral functions.

(5) Prescaler 2 (PRS2)

This circuit divides main clock f_{XX} (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$) to provide the CPU ($f_{CPU)}$ clock and system clock (f_{CLK}).
f_{CLK} is supplied to the interrupt controller INTC, ROM controller, ROM and RAM blocks. It can be output from the CLKOUT pin.
f_{CPU} is the clock supplied to CPU.

6.3 Control Registers

(1) Main peripheral clock control register (MPCCTL)

This is an 8-bit register that selects the internal clock. This register can be read or written in 8-bit or 1-bit units.

Data can be written to this register only in combination of specific sequences (refer to **3.2.3 ”Special registers” on page 70**). The clock generator flexibility allows to use either simultaneously both PLL or to use only one PLL. Both PLLs can be assigned respectively to the CPU or to the peripherals.

Figure 6-2: Main Peripheral Clock Control Register (MPCCTL) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
MPCCTL	SELPLL	0	0	0	MCKSEL	PCKSEL	STPPLL1	STPPLL0	FFFFF82BH	80H
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W		

SELPLL	f_{XX} clock select
0	Main-OSC clock ($f_{XX} = f_X$)
1	PLL clock ($f_{XX} = f_{PLL\_MCKSEL\_CKDIV}$)

Caution: When using an 8-bit manipulation instruction, do not change the SELPLL bit with other bit.

MCKSEL	Master clock selection register (f_{PLL_MCKSEL})
0	Main clock ($f_{PLL\_MCKSEL)}$ = PLL0 (Default)
1	Main clock ($f_{PLL\_MCKSEL)}$ = PLL1

Caution: When using PLL clock for CPU, do not change the MCKSEL bit value.

PCKSEL	Peripheral clock selection register (f_{PLL_PCKSEL})
0	Peripheral clock = PLL0 (Default)
1	Peripheral clock = PLL1

Caution: When EXCKSEL register value is 01h to 3Fh, do not change PCKSEL bit value.
f_{PLL_PCKSEL} clock is provided only to aFCANn, CSI3n and CSIBn (n = 0,1).

STPPLL1	PLL 1 execution stop register
0	PLL1 executable (Default)
1	PLL1 stop

Cautions: 1. If this bit is set to “1”, it is impossible to set to “0” by register writing. Only $\overline{\text{RESET}}$ input can be set to “0”.

2. When using PLL1 clock for peripheral functions, do not set to 1 this bit.

Figure 6-2: Main Peripheral Clock Control Register (MPCCTL) Format (2/2)

STPPLL0	PLL 0 execution stop register
0	PLL0 executable (Default)
1	PLL0 stop

Cautions: 1. If this bit is set to “1”, it is impossible to set to “0” by register writing. Only $\overline{\text{RESET}}$ input can be set to “0”.

2. When using PLL0 clock for peripheral functions, do not set to 1 this bit.

(2) Extension clock select register (EXCKSEL)

This is an 8-bit register that selects the internal clock. This register can be read or written in 8-bit or 1-bit units.

Figure 6-3: Extension Clock Select Register (EXCKSEL) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
EXCKSEL	0	0	CB1CKSEL	CB0CKSEL	CS31CKSEL	CS30CKSEL	AF1CKSEL	AF0CKSEL	FFFFF30CH	00H
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

CB1CKSEL	CSIB1 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

CB0CKSEL	CSIB0 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

CS31CKSEL	CSI31 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

CS30CKSEL	CSI30 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

AF1CKSEL	CAN1 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

AF0CKSEL	CAN0 Clock selection
0	Normal clock selection (f_{XX})
1	Extended clock selection (f_{PLL_PCKSEL})

Cautions: 1. Before modifying EXCKSEL register value, corresponding peripheral I/O function have to be stopped.

2. When set to 1 either bit 5 or 4 of EXCKSEL register, be sure to set bit 4 of OCKS2 register (OCKSEN2 bit) to 1.

(3) Power save mode register (PSMR)

This 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. Be sure to clear bits 2 to 7 of the PSMR register to 0.

Figure 6-4: Power Save Mode Register (PSMR) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PSMR	0	0	0	0	0	0	PSM1	PSM0	FFFFF820H	00H
R/W	R/W	R	R/W	R/W	R	R	R/W	R/W		

PSM1	PSM0	Software standby mode select
0	0	IDLE1 mode
0	1	STOP mode
1	0	IDLE2 mode
1	1	Setting prohibited

Caution: The PSM0 and PSM1 bits are valid only when the STP bit of the PSC register is 1.

Remark: IDLE1: In this mode, all operations except the oscillator operation, flash memory, and PLL are stopped. After the IDLE1 mode is released, the normal mode do not need to wait the lapse of the oscillation stabilization time.

IDLE2: In this mode, all operations except the oscillator and flash memory operation are stopped. After the IDLE2 mode is released, the normal mode is returned to following the lapse of the setup time (flash memory, PLL) specified by the OSTS register.

STOP: In this mode, all operations are stopped, except the ring oscillator operation. After the STOP mode is released, the normal mode is returned to following the lapse of the oscillation stabilization time specified by the OSTS register.

Note: Refer to **Chapter 18 ”Standby Function” on page 723** to get information about standby mode release.

(4) Processor clock control register (PCC)

The processor clock control register (PCC) is a special register. This register can be read or written in 8-bit or 1-bit units. Data can be written to this register only in combination of specific sequences (refer to **3.2.3 ”Special registers” on page 70**).

Figure 6-5: Processor Clock Control Register (PCC) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PCC	0	0	MFRC	0	0	CK2	CK1	CK0	FFFFF828H	00H
R/W	R	R	R/W	R	R	R/W	R/W	R/W		

MFRC	Use of Main clock on –chip feedback register
0	Used
1	Not Used

CK2	CK1	CK0	Clock selection (for f_{CPU})
0	0	0	$f_{CPU} = f_{XX}$
0	0	1	$f_{CPU} = f_{XX}$ /2
0	1	0	$f_{CPU} = f_{XX}$ /4
0	1	1	$f_{CPU} = f_{XX}$ /8
1	0	0	$f_{CPU} = f_{XX}$ /16
1	0	1	$f_{CPU} = f_{XX}$ /32
1	1	0	Setting prohibited
1	1	1	

Caution: When operating the CPU and peripherals by PLL, changing the PCC register is prohibited.

(5) CPU operation clock status register (CCLS)

Status flags that indicate the operation status of the CPU operating clock.
This register can be read in 8-bit or 1-bit units.

Figure 6-6: CPU Operation Clock Status Register (CCLS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
CCLS	0	0	0	0	0	0	0	CCLSF	FFFFF82EH	00H
R/W	R	R	R	R	R	R	R	R		

CCLSF	Status of operating clock
0	Operating with main system clock (f_X)
1	Operating with ring OSC (f_R)

(6) Ring-OSC mode register (RCM)

This is an 8-bit register that sets the operation mode of the ring oscillator (Ring-OSC).
This register can be read or written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 6-7: Ring-OSC Mode Register (RCM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
RCM	0	0	0	0	0	0	0	RSTOP	FFFFF80CH	00H
R/W	R	R	R	R	R	R	R	R/W		

RSTOP	Ring-OSC operation select
0	Ring-OSC Operate
1	Ring-OSC Stop

(7) Oscillation stabilization time select register (OSTS)

This 8-bit register selects the oscillation stabilization time following reset or release of the STOP mode.

The OSTS register is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 03H.

Figure 6-8: Oscillation Stabilization Time Select Register (OSTS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFFF6C0H	03H
R/W	R	R	R	R	R	R/W	R/W	R/W		

OSTS2	OSTS1	OSTS0	Selection of stabilization time/setup time[Note]			
				f_X = 4 MHz	f_X = 6 MHz	f_X = 8 MHz
0	0	0	$2^{10}/f_X$	0.256 ms	0.170 ms	0.128 ms
0	0	1	$2^{11}/f_X$	0.512 ms	0.341 ms	0.256 ms
0	1	0	$2^{12}/f_X$	1.024 ms	0.683 ms	0.512 ms
0	1	1	$2^{13}/f_X$	2.048 ms	1.365 ms	1.024 ms
1	0	0	$2^{14}/f_X$	4.096 ms	2.731 ms	2.048 ms
1	0	1	$2^{15}/f_X$	8.192 ms	5.461 ms	4.096 ms
1	1	0	$2^{16}/f_X$	16.38 ms	10.92 ms	8.192 ms
1	1	1	Setting prohibited			

Note: The oscillation stabilization time and setup time are required when the stop mode and idle mode are released respectively.

(8) Clock selection register 2 (OCKS2)

This is an 8-bit register that controls the operation enable and clock selection for CSIBn (n = 0, 1).

Figure 6-9: Clock Selection Register 2 (OCKS2) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCKS2	0	0	0	OCKSEN2	OCKSTH2	0	OCKS21	OCKS20	FFFFF868H	00H
R/W	R	R	R	R/W	R/W	R	R/W	R/W		

OCKSEN2	Specified for execution enable
0	Operation Disable
1	Operation Enable

OCKSTH2	Specified for output clock through or divide
0	Output clock is divided clock by setting OCKS21 and OCKS20
1	Output clock is through (Extended clock = f_{PLL_PCKSEL})

OCKS21	OCKS20	Specified for divider factor
0	0	Extended clock = f_{PLL_PCKSEL} / 2
0	1	Extended clock = f_{PLL_PCKSEL} / 3
1	0	Extended clock = f_{PLL_PCKSEL} / 4
1	1	Extended clock = f_{PLL_PCKSEL} / 5

Caution: When PLL mode operation is enabled, OCKS2 register value must not be changed.

6.4 PLL Function

The PLL function is used to output the operating clock of the CPU and peripheral macro at a frequency 10 times (PLL1) / 12 times (PLL0) higher than the oscillation frequency, and selects the clock-through mode.

• When PLL function is used:	Input clock:	4 to 8 MHz
	Output:	4 to 40 MHz (μPD70F3403 and μPD70F3403A) 4 to 32 MHz (μPD70F3402)
• Clock-through mode:	Input clock:	4 to 8 MHz
	Output:	4 to 8 MHz

PLL0 can generate below frequencies from f_X signal and register setting:

Base clock (f_X)	OCKS0 register divide value	PLL0 Time value	CKC Register divide value	PLL0 Output (f_{PLL0})Note
f_X	1	12	1	$f_{PLL0} = f_X \times 12$
			2	$f_{PLL0} = f_X \times 6$
			4	$f_{PLL0} = f_X \times 3$
	2		1	$f_{PLL0} = f_X \times 6$
			2	$f_{PLL0} = f_X \times 3$
			4	$f_{PLL0} = f_X \times 1.5$
	3		1	$f_{PLL0} = f_X \times 4$
			2	$f_{PLL0} = f_X \times 2$
			4	$f_{PLL0} = f_X \times 1$
	4		1	$f_{PLL0} = f_X \times 3$
			2	$f_{PLL0} = f_X \times 1.5$
			4	$f_{PLL0} = f_X \times 0.75$
	5		1	$f_{PLL0} = f_X \times 2.4$
			2	$f_{PLL0} = f_X \times 1.2$
			4	$f_{PLL0} = f_X \times 0.6$

Note: Set the f_{PLL0} as 24 MHz or 32 MHz when f_{PLL0} is used as CPU clock.

PLL1 can generate below frequencies from f_X signal and register setting.

Base clock (f_X)	OCKS1 register divide value	PLL1 Time value	CKC Register divide value	PLL1 Output (f_{PLL1})Note
f_X	1	10	1	$f_{PLL1} = f_X \times 10$
			2	$f_{PLL1} = f_X \times 5$
			4	$f_{PLL1} = f_X \times 2.5$
	2		1	$f_{PLL1} = f_X \times 5$
			2	$f_{PLL1} = f_X \times 2.5$
			4	$f_{PLL1} = f_X \times 1.25$
	3		1	$f_{PLL1} = f_X \times 3.3$
			2	$f_{PLL1} = f_X \times 1.67$
			4	$f_{PLL1} = f_X \times 0.83$
	4		1	$f_{PLL1} = f_X \times 2.5$
			2	$f_{PLL1} = f_X \times 1.25$
			4	$f_{PLL1} = f_X \times 0.63$
	5		1	$f_{PLL1} = f_X \times 2$
			2	$f_{PLL1} = f_X \times 1$
			4	$f_{PLL1} = f_X \times 0.5$

Note: Set the f_{PLL1} as 40 MHz when f_{PLL1} is used as CPU clock.

6.4.1 Control register

(1) PLL0 control register 0 (PLLCTL0)

This is an 8-bit register that enables PLL0 operation. This register can be written in 8-bit units.
After reset, an firmware initializes this register to 00H so that the PLL0 is enabled and in a locked status.
Always write 00H to this register. However, the user is not required to initialize due to internal firmware processing.

Figure 6-10: PLL Control Register (PLLCTL0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PLLCTL0	-	-	-	-	-	-	-	-	FFFFF82CH	00H
R/W	W	W	W	W	W	W	W	W		

(2) PLL1 control register 1 (PLLCTL1)

This is an 8-bit register that enables PLL1 operation. This register can be written in 8-bit units.
Before PLL1 can be used (before MCKSEL=1 and SELPLL=1), this register must be initialized to 00H. Always write 00H to this register.

Figure 6-11: PLL Control Register (PLLCTL1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PLLCTL1	-	-	-	-	-	-	-	-	FFFFF82DH	00H
R/W	W	W	W	W	W	W	W	W		

(3) PLL lockup time specification register (PLLS)

The PLLS register is an 8-bit register used to select the PLL lockup time when the PLL is initialized by writing the value 00H to the PLLCTLn register, and when the software STOP mode is released. This register can be read or written in 8-bit units.
Reset input sets this register to 03H.

Figure 6-12: PLL Lockup Time Specification Register (PLLS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PLLS	0	0	0	0	0	0	PLLS1	PLLS0	FFFFF6C1H	03H
R/W	R	R	R	R	R	R	R/W	R/W		

PLLS1	PLLS0	PLL lockup time selector
0	0	$2^{10}/f_X$ (128 µs)
0	1	$2^{11}/f_X$ (256 µs)
1	0	$2^{12}/f_X$ (512 µs)
1	1	$2^{13}/f_X$ (1024 µs) (default)

Remark: The value in parentheses assumes an 8 MHz external oscillator frequency (f_X).

Caution: Always set PLLS so that the interval time is 800 µs or longer.

(4) Clock control register (CKC)

This is an 8-bit register that controls system clock (f_{XX}) when it operates on PLL mode (SELPLL bit of MPCCTL register is set to 1). This register can be read or written in 8-bit or 1-bit units.
Be sure to clear bits 2 to 7 of the CKC register to 0. In other case the operation is not guaranteed.
Data can be written to this register only in combination of specific sequences (refer to **3.2.3 ”Special registers” on page 70**).

Figure 6-13: Clock Control Register (CKC) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
CKC	0	0	0	0	0	0	CKDIV1	CKDIV0	FFFFF822H	03H
R/W	R	R	R	R	R	R	R/W	R/W		

CKDIV1	CKDIV0	System clock (f_{XX}) select on PLL mode
0	0	Setting prohibited
0	1	$f_{PLL\_MCKSEL\_CKDIV} = f_{PLL}$(PLL internal clock) / 4
1	0	$f_{PLL\_MCKSEL\_CKDIV} = f_{PLL}$(PLL internal clock) / 2
1	1	$f_{PLL\_MCKSEL\_CKDIV} = f_{PLL}$(PLL internal clock)

Caution: When operating the CPU and the peripherals by PLL mode, changing the CKC register is prohibited.

(5) Clock selection register 0 (OCKS0)

This is an 8-bit register that controls the operation enable and clock input selection for PLL0.

Figure 6-14: Clock Selection Register 0 (OCKS0) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCKS0	0	0	0	OCKSEN0	OCKSTH0	0	OCKS01	OCKS00	FFFFF860H	11H
R/W	R	R	R	R/W	R/W	R	R/W	R/W		

OCKSEN0	Specified for execution enable
0	PLL0 operation Disable
1	PLL0 operation Enable

OCKSTH0	Specified for output clock through or divide
0	Output clock is divided clock by setting OCKS01 & OCKS00
1	Output clock is through

OCKS01	OCKS00	Specified for divider factor
0	0	$f_X/2$
0	1	$f_X/3$
1	0	$f_X/4$
1	1	$f_X/5$

Caution: When PLL mode operation is enabled, OCKS0 register value must not be changed.

(6) Clock selection register 1 (OCKS1)

This is an 8-bit register that controls the operation enable and clock input selection for PLL1.

Figure 6-15: Clock Selection Register 1 (OCKS1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCKS1	0	0	0	OCKSEN1	OCKSTH1	0	OCKS11	OCKS10	FFFFF864H	10H
R/W	R	R	R	R/W	R/W	R	R/W	R/W		

OCKSEN1	Specified for execution enable
0	PLL1 operation Disable
1	PLL1 operation Enable

OCKSTH1	Specified for output clock through or divide
0	Output clock is divided clock by setting OCKS11 & OCKS10
1	Output clock is through

OCKS11	OCKS10	Specified for divider factor
0	0	$f_X/2$
0	1	$f_X/3$
1	0	$f_X/4$
1	1	$f_X/5$

Caution: When PLL mode operation is enabled, OCKS1 register value must not be changed.

6.5 Programmable Clock output Function (PCL)

It is possible to output a clock independent from CPU clock on pin P913/TIP20/TOP20/INTP4/PCL. The programmable clock output frequency is equal to f_{PLL_MCKSEL} divided by two prescalers (PCLM and OCKS3 registers). Corresponding ports registers have to be set accordingly (Refer to section **4.3 ”Port Configuration” on page 107**).

6.5.1 Control registers

(1) Programmable clock mode register (PCLM)

This is an 8-bit register that controls the PCL output. This register can be read or written in 8-bit or 1-bit units.

Figure 6-16: Programmable Clock Mode Register (PCLM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PCLM	0	0	0	PCLE	0	PCK2	PCK1	PCK0	FFFFF82FH	01H
R/W	R	R	R	R/W	R	R/W	R/W	R/W		

PCLE	PCL operation selection
0	PCL output disable (PCL output level is low)
1	PCL output enable

Caution: PCLE can set to 1 after setting the port control registers (PM, PMC, PFC, PFCE). This bit can set to 1 on PLL operation. And this bit has to be cleared to 0 before stopping the PLL.

PCK2	PCK1	PCK0	PLL output select
0	0	0	$f_{PCL1} = f_{PLL\_MCKSEL}$
0	0	1	$f_{PCL1} = f_{PLL\_MCKSEL} / 2$
0	1	0	$f_{PCL1} = f_{PLL\_MCKSEL} / 4$
0	1	1	$f_{PCL1} = f_{PLL\_MCKSEL} / 8$
1	0	0	$f_{PCL1} = f_{PLL\_MCKSEL} / 16$
1	0	1	Setting prohibited
1	1	0	
1	1	1	

Caution: Before modifying the output selection clock, PCL output has to be stopped (PCLE bit cleared to 0).

(2) Clock selection register 3 (OCKS3)

This is an 8-bit register that controls the operation enable and clock selection for PCL output.

Figure 6-17: Clock Selection Register 3 (OCKS3) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCKS3	0	0	0	OCKSEN3	OCKSTH3	0	OCKS31	OCKS30	FFFFF86CH	00H
R/W	R	R	R	R/W	R/W	R	R/W	R/W		

OCKSEN3	Specified for execution enable
0	Operation Disable
1	Operation Enable

OCKSTH3	Specified for output clock through or divide
0	Output clock is divided clock by setting OCKS31 and OCKS30
1	Output clock is through (f_{PCL1})

OCKS31	OCKS30	Specified for divider factor
0	0	$f_{PCL} = f_{PCL1} /2$
0	1	$f_{PCL} = f_{PCL1} /3$
1	0	$f_{PCL} = f_{PCL1} /4$
1	1	$f_{PCL} = f_{PCL1} /5$

Remark: OCKSEN3 bit disables only the divider factor function. If OCKSEN3 bit is set to 0 and OCKSTH3 bit is set to 1, the PCL outputs f_{PCL1} frequency.

6.6 Usage

6.6.1 To use PLL1

After $\overline{\text{RESET}}$ has been released, the default mode is PLL0 mode (F_{PLL_MCKSEL}= 4 × f_X). When operating mode change to PLL1 mode, register access must keep below mentioned order.

- (When use PLL1 function as master clock and PLL0 for peripheral clock)

1) Setting OCKS1 register
2) Setting CKC register
3) PLLCTL1 register = 00H write
4) MPCCTL register = 00H write (MPCCTL: SELPLL bit = 0)
4) MPCCTL register = 08H write (MPCCTL: MCKSEL bit = 1)
5) MPCCTL register <= 88H write (MPCCTL: SELPLL bit = MCKSEL bit =1)

- (When use PLL1 function as master clock and peripheral clock)

1) Setting OCKS1 register
2) Setting CKC register
3) PLLCTL1 register = 00H write
4) MPCCTL register = 00H write (MPCCTL: SELPLL bit = 0)
4) MPCCTL register = 08H write (MPCCTL: MCKSEL bit = 1)
5) MPCCTL register <= 8DH write (MPCCTL: SELPLL bit = MCKSEL bit = PCKSEL bit = STPPLL0 bit = 1)

Caution: Before using the STOP/IDLE mode for power saving operation, the user must first reduce CPU operating frequency to 20 MHz using PCC register.

The follow table shows the divide and PLL0 time value.

Table 6-1: Divide and PLL0 Time Value

OCKS0 register (h)	CKC register (h)	PLL time value	f_X (MHz)		
			Xtal = 4 MHz	Xtal = 6 MHz	Xtal = 8 MHz
12h	03h	3	12	18	24
11h (Default)	03h (Default)	4	16	24	32
18h	02h	6	24	36	Setting prohibited

The follow table shows the divide and PLL1 time value.

Table 6-2: Divide and PLL1 Time Value

OCKS1 register (h)	CKC register (h)	PLL time value	f_X (MHz)		
			Xtal = 4 MHz	Xtal = 6 MHz	Xtal = 8 MHz
	03h	2	8	12	16
10h (Default)	03h	5	20	30	40
18h	03h	10	40	Setting prohibited	Setting prohibited

6.6.2 To use clock through mode

When operating mode change to clock through mode, register access must keep the following order:

1) MPCCTL = 00H,(MPCCTL: SELPLL bit = 0
2) MPCCTL = 03H,MPCCTL: STPPLL0 bit = STPPLL1 bit = 0

6.6.3 to Use the programmable clock output function (PCL)

When use the PCL function, register setting have 1 sequence.

- After Reset timing

1) Setting OCKS3 register
2) Setting PCLM: PCK2 - PCK0 bit
3) Setting Port 9 port mode register, PM9
4) Setting Port 9 port mode control register, PMC9
5) Setting Port 9 port mode function register, PFC9
6) Setting Port 9 expanded function register, PFCE9
7) Set PCLM: PCLE bit

- PCL output period change timing

1) Clear PCLE bit to 0 (PCLM register)
2) Setting OCKS3 register
3) Setting PCLM: PCK2 - PCK0 bit
4) Set PCLE bit to 1 (PCLM register)

The follow table shows the divide value of f_{PLL} frequency and f_{PCL} frequency.

Table 6-3: Divide Value of f_{PLL} Frequency and f_{PCL} Frequency

PCLM register (h)	OCKS3 register (h)	PCL output signal period
1(10h)	1 (18h)	$f_{PCL} = f_{PLL\_MCKSEL} / 1$
	2 (10h)	$f_{PCL} = f_{PLL\_MCKSEL} / 2$
	3 (11h)	$f_{PCL} = f_{PLL\_MCKSEL} / 3$
	4 (12h)	$f_{PCL} = f_{PLL\_MCKSEL} / 4$
	5 (13h)	$f_{PCL} = f_{PLL\_MCKSEL} / 5$
2(11h)	1 (18h)	$f_{PCL} = f_{PLL\_MCKSEL} / 2$
	2 (10h)	$f_{PCL} = f_{PLL\_MCKSEL} / 4$
	3 (11h)	$f_{PCL} = f_{PLL\_MCKSEL} / 6$
	4 (12h)	$f_{PCL} = f_{PLL\_MCKSEL} / 8$
	5 (13h)	$f_{PCL} = f_{PLL\_MCKSEL} / 10$
4(12h)	1 (18h)	$f_{PCL} = f_{PLL\_MCKSEL} / 4$
	2 (10h)	$f_{PCL} = f_{PLL\_MCKSEL} / 8$
	3 (11h)	$f_{PCL} = f_{PLL\_MCKSEL} / 12$
	4 (12h)	$f_{PCL} = f_{PLL\_MCKSEL} / 16$
	5 (13h)	$f_{PCL} = f_{PLL\_MCKSEL} / 20$
8(13h)	1 (18h)	$f_{PCL} = f_{PLL\_MCKSEL} / 8$
	2 (10h)	$f_{PCL} = f_{PLL\_MCKSEL} / 16$
	3 (11h)	$f_{PCL} = f_{PLL\_MCKSEL} / 24$
	4 (12h)	$f_{PCL} = f_{PLL\_MCKSEL} / 32$
	5 (13h)	$f_{PCL} = f_{PLL\_MCKSEL} / 40$
16(14h)	1 (18h)	$f_{PCL} = f_{PLL\_MCKSEL} / 16$
	2 (10h)	$f_{PCL} = f_{PLL\_MCKSEL} / 32$
	3 (11h)	$f_{PCL} = f_{PLL\_MCKSEL} / 48$
	4 (12h)	$f_{PCL} = f_{PLL\_MCKSEL} / 64$
	5 (13h)	$f_{PCL} = f_{PLL\_MCKSEL} / 80$

Chapter 7 16-Bit Timer/Event Counter P

The V850E/RS1 includes four channels Timer P (TMP0, TMP1, TMP2 and TMP3).

7.1 Features

Timer P (TMP) is a 16-bit timer/event counter provided with general-purpose functions.
TMP can perform the following operations.

- 16-bit-accuracy PWM output timer
- Interval timer
- External event counter function (operation not possible when clock is stopped)
- Timer synchronised operation function
- One-shot pulse output
- Pulse interval and frequency measurement counter
- Free running function
- External trigger pulse output function

7.2 Function Outline

- Capture trigger input signal × 2
- External trigger input signal × 1
- Clock select × 8
- External event count input × 1
- Readable counter × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer output (TOPn0, TOPn1) × 2

Remark: n = 0 to 3

7.3 Configuration

TMP includes the following hardware.

Table 7-1: Configuration of TMP0 to TMP3

Item	Configuration
Timer register	16-bit counter
Registers	TMPn timer capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn timer read buffer register (TPnCNT) CCR0 buffer register, CCR1 buffer register
Timer output	TOPn1, TOPn0
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option registers 0 (TPnOPT0) Selector operation control register 0,1 (SELCNT0, SELCNT1) TIPm pin noise elimination control register (PnmNFC)

Remarks: 1. n = 0 to 3

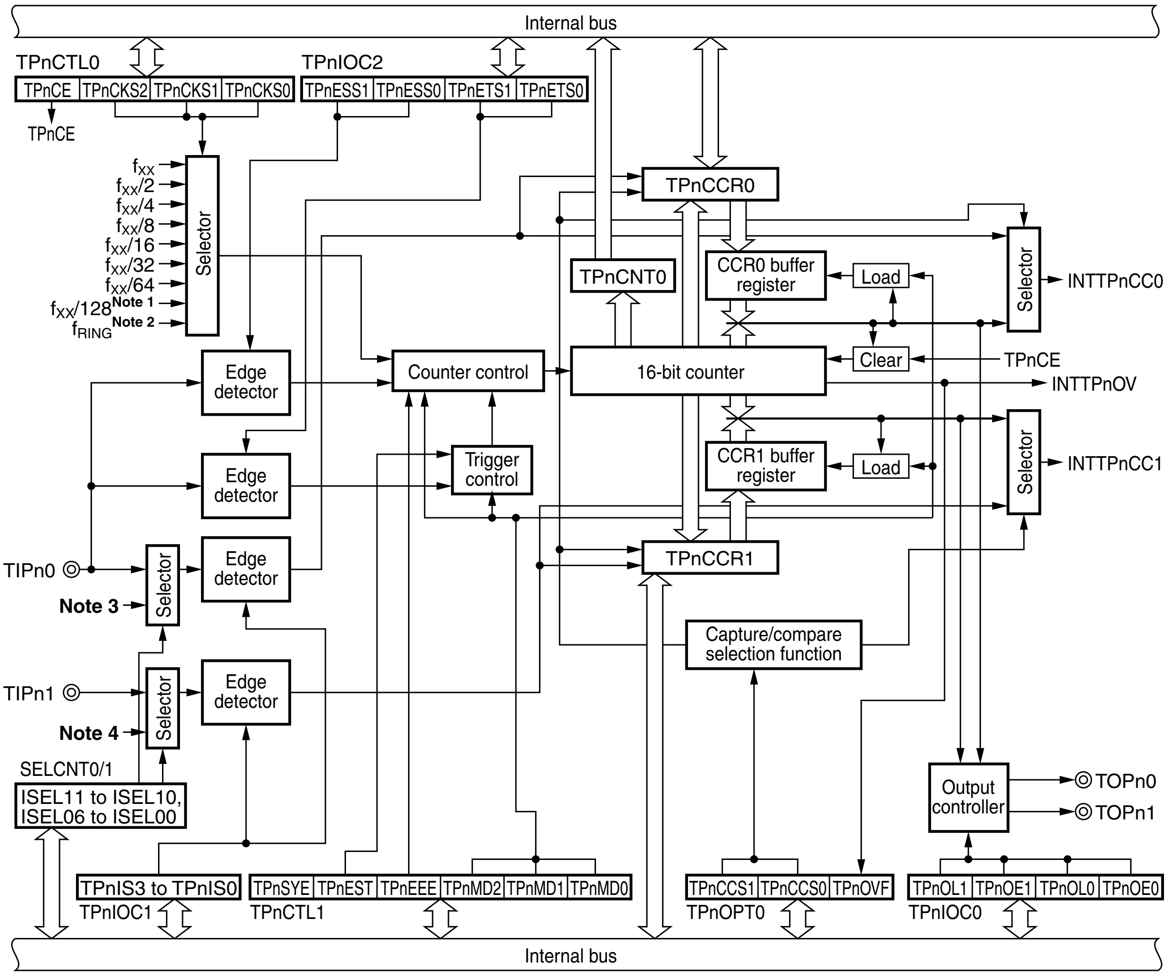
2. m = 0,1

Timer P (TMP) pins are alternate function of port pins. For how to set the alternate function, refer to the description of the registers in **Chapter 4 ”Port Functions” on page 105**.

Table 7-2: TMP Pin List

Pin Name	I/O	Function	Alternate Function
TIP00	Input	External event/clock input (TMP00)	P37/TOP00/INTP1/(CS312)
TIP01		External event/clock input (TMP01)	P38/TOP01
TIP10		External event/clock input (TMP10)	P914/TOP10/INTP5/AD14
TIP11		External event/clock input (TMP11)	P915TOP11/INTP6/AD15
TIP20		External event/clock input (TMP20)	P913/TOP20/INTP4/PCL
TIP21		External event/clock input (TMP21)	P912/TOP21/(CS302)
TIP30		External event/clock input (TMP30)	P11/TOP31
TIP31		External event/clock input (TMP31)	P12/TOP30/ADTRG
TOP00	Output	Timer output (TMP00)	P37/TIP00/INTP1/(CS312)
TOP01		Timer output (TMP01)	P38/TIP01
TOP10		Timer output (TMP10)	P914/TIP10/INTP5/AD14
TOP11		Timer output (TMP11)	P915/TIP11/INTP6/AD15
TOP20		Timer output (TMP20)	P913/TIP20/INTP4/PCL
TOP21		Timer output (TMP21)	P912 /TIP21/(CS302)
TOP30		Timer output (TMP30)	P11/TIP31
TOP31		Timer output (TMP31)	P12/TIP30/ADTRG

Figure 7-1: Block Diagram of Timer P



Notes: 1. TMP0, TMP2

2. TMP1, TMP3

3. TMP0: TSOUT signal of CAN0 block
TMP1: TSOUT signal of CAN1 block
TMP2: RXDA0 pin
TMP3: RXDA1 pin

4. TMP0: INTTM0EQ0 interrupt of TMM0 block

(1) TMP capture/compare register 0 (TPnCCR0)

The TPnCCR0 register is a 16-bit register that operates either as capture register or as a compare register.
Whether this register is used as a capture register or as a compare register can be specified with the TPnCCS1 and TPnCCSS0 bits of the TMPn option register 0 (TPnOPT0 register), but only in the free-running mode.
In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)
In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.
After a $\overline{\text{RESET}}$, TPnCCR0 register default status is compare register.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Caution: When external event counter mode is used, do not set TPnCCR0 register to 0000H.

Figure 7-2: Capture/Compare Register 0 (TPnCCR0) Format

Address: TP0CCR0: FFFFF596H, TP1CCR0: FFFFF5A6H,
TP2CCR0: FFFFF5B6H, TP3CCR0: FFFFF5C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TPnCCR0																	FFFF F596H to FFFF F5C6H	R/W	0000H

Remark: n = 0 to 3

- When used as a compare register

TPnCCR0 can be rewritten when TPnCE = 1, as below mentioned.

TMP operation mode	TPnCCR0 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because it is only used as capture register

- When used as capture register

Count value is stored in TPnCCR0 upon capture trigger (TIPn0) input edge detection.

Note: The value of TPnCCR0 register is read when TPnCE bit of TMPn control register 0 (TPnCTL0) equals 1.

(2) TMP capture/compare register 1 (TPnCCR1)

The TPnCCR1 register is a 16-bit register that operates either both as a capture register or as a compare register.
Whether this register is used as a capture register or as a compare register can be specified with the TPnCCS1 and TPnCCS0 bits of the TMPn option register 0 (TPnOPT0 register), but only in free-running mode.
In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)
In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.
After $\overline{\text{RESET}}$, TPnCCR1 register default status is compare register.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Caution: When external event counter mode is used, do not set TPnCCR1 register to 0000H.

Figure 7-3: Capture/Compare Register 1 (TPnCCR1) Format

Address: TP0CCR1: FFFFF598H, TP1CCR1: FFFFF5A8H
TP2CCR1: FFFFF5B8H, TP3CCR1: FFFFF5C8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TPnCCR1																	FFFF F598H to FFFF F5C8H	R/W	0000H

Remark: n = 0 to 3

- When used as a compare register

TPnCCR1 can be rewritten when TPnCE = 1, as below mentioned.

TMP operation mode	TPnCCR0 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because it is only used as capture register

- When used as a capture register

Count value is stored in TPnCCR1 upon capture trigger (TIPn1) input edge detection.

Note: The value of TPnCCR1 register is read when TPnCE bit of TMPn control register 0 (TPnCTL0) equals 1.

(3) TMPn counter read buffer register (TPnCNT)

TPnCNT register is a read buffer register that can read 16-bit counter values.
This register is read-only, using a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to FFFFH.
Although the hardware status is FFFFH when TPnCE bit of TPnCTL0 equals 0, 0000H is read from this register.
The value of the register is read when TPnCE bit = 1.

Figure 7-4: TMPn Timer Read Buffer Register (TPnCNT) Format

Address: TP0CNT: FFFFF59AH, TP1CNT: FFFFF5AAH
TP2CNT: FFFFF5BAH, TP3CNT: FFFFF5CAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TPnCNT																	FFFF F59AH to FFFF F5CAH	R	FFFFH

Remark: n = 0 to 3

7.4 Control Registers

(1) TMPn control register 0 (TPnCTL0)

TMPn control register 0 is an 8-bit register that controls the operation of timer P.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.
The same value can always be written to the TPnCTL0 register by software.

Figure 7-5: TMPn Control Register 0 (TPnCTL0) Format

Address: TP0CTL0: FFFFF590H, TP1CTL0: FFFFF5A0H
TP2CTL0: FFFFF5B0H, TP3CTL0: FFFFF5C0H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TPnCTL0	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0	FFFFF590H to FFFFF5C0H	R/W	00H

TPnCE	Timer Pn operation control
0	Disable internal operating clock operation (TMPn is asynchronously reseted)
1	Enable internal operating clock operation
Internal operating clock control and TMPn asynchronous reset are performed with the TPnCE bit. When TPnCE bit is cleared to 0, the internal operating clock of TMPn stops (fixed to low level) and TMPn is reset asynchronously. When the TPnCE bit is set to 1, the internal operating clock is enabled within 2 input clocks, and TMPn counts up.	

TPnCKS2	TPnCKS1	TPnCKS0	Internal count clock selection	
			n = 0, 2	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_{RING}

Caution: Set bits TPnCKS2 to TPnCKS0 only when TPnCE = 0.
When TPnCE bit setting is changed from 0 to 1, TPnCKS2 to TPnCKS0 bits can be set simultaneously.

Remark: n = 0 to 3
f_{RING}: Ring oscillator frequency

(2) TMPn timer control register 1 (TPnCTL1)

TMPn control register 1 is an 8-bit register that controls the operation of timer P.
This register can be read and written in 8-bit or 1-bit units.
16-bit access is not possible.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-6: TMPn Control Register 1 (TPnCTL1) Format (1/2)

Address:TP0CTL1: FFFFF591H, TP1CTL1: FFFFF5A1H
TP2CTL1: FFFFF5B1H, TP3CTL1: FFFFF5C1H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TPnCTL1	TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0	FFFFF591H to FFFFF5C1H	R/W	00H

<table>
<tr><th>TPnSYE</th><th>Tuned operation mode enable control</th></tr>
<tr><td>0</td><td>Independent operation mode (asynchronous operation mode)</td></tr>
<tr><td>1</td><td>Tuned operation mode (specification of slave operation)
In this mode, timer P can operate in synchronization with a master timer.
<table>
<tr><th>Master timer</th><th colspan="2">Slave timer</th></tr>
<tr><td>TMP0</td><td>TMP1</td><td>-</td></tr>
<tr><td>TMP2</td><td>TMP3</td><td>TMQ0</td></tr>
</table>
For the tuned operation mode, refer to <b>7.6 Timer Synchronized Operation Function</b>.
<b>Caution: Be sure to clear the TP0SYE and TP2SYE bits to 0 (master timer) and TP1SYE, TP3SYE and TQ0SYE bits to 1 when respectively used as slave timers.</b></td></tr>
</table>

Caution: In the synchronous operation mode, the master macro can be set only in the PWM mode, external trigger pulse output mode, pulse output mode, and free-running mode.
The slave macros can be set only in the free-running mode.
Setting the external event count mode, one-shot pulse mode, and pulse width measurement mode is prohibited.

<table>
<tr><th>TPnEST</th><th>Software trigger control</th></tr>
<tr><td>0</td><td>No operation</td></tr>
<tr><td rowspan="2">1</td><td>In one-shot pulse mode: One-shot pulse software trigger</td></tr>
<tr><td>In external trigger pulse output mode: Pulse output software trigger</td></tr>
<tr><td colspan="2">The TPnEST bit functions as a software trigger in the one-shot pulse mode or external trigger pulse output mode (this bit is invalid in any other mode). By setting TPnEST to 1 when TPnCE = 1, a software trigger is issued. Therefore, be sure to set TPnEST to 1 when TPnCE = 1. The TIPn0 pin is used for an external trigger.
The read value of the TPnEST bit is always 0.</td></tr>
</table>

Figure 7-6: TMPn Control Register 1 (TPnCTL1) Format (2/2)

TPnEEE	Count clock selection
0	Use the internal clock (clock selected with TPnCKS2 to TPnCKS0 bits of TPnCTL0 register)
1	Use external clock (TIPn0 input edge)
The valid edge is specified with TPnEES1 and TPnEES0 bits when TPnEEE bit = 1 (external clock TIPn0).	

TPnMD2	TPnMD1	TPnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event counter mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse mode
1	0	0	PWM mode
1	0	1	Free-running mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Caution: Set bits TPnEEE and TPnMD2 to TPnMD0 when TPnCE = 0. (The same value can be written when TPnCE = 1.) The operation is not guaranteed when rewriting is performed when TPnCE = 1. If rewriting was mistakenly performed, clear TPnCE to 0 and then set the bits again.

Remark: n = 0 to 3

(3) TMPn dedicated I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer output.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-7: TMPn Dedicated I/O Control Register 0 (TPnIOC0) Format

Address: TP0IOC0: FFFFF592H, TP1IOC0: FFFFF5A2H
TP2IOC0: FFFFF5B2H, TP3IOC0: FFFFF5C2H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0	FFFFF592H to FFFFF5C2H	R/W	00H

TPnOLm	Timer output level setting
0	Normal output
1	Inverted output
This bit can be used to invert the timer output	

TPnOEm	Timer output setting
0	Timer output is disabled (Low level and high level are output from TOPn1 pin when TPnOL = 0 and TPnOL = 1, respectively.)
1	Timer output is enabled (TOPnm pin outputs pulses.)

Cautions: 1. Rewrite bits TPnOLm and TPnOEm when TPnCE = 0 (the same value can be written when TPnCE = 1.). If rewriting was mistakenly performed, clear TPnCE to 0 and then set the bits again.

2. To enable the timer output, be sure to set the corresponding alternate-function pins TPnIS3 to TPnIS0 of the TPnIOC1 register to "No edge detection" and invalidate the capture operation. Then set the corresponding alternate-function port to output mode.

Remark: n = 0 to 3
m = 0, 1

(4) TMPn dedicated I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge for the external input signals (TIPn0 and TIPn1).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-8: TMPn Dedicated I/O Control Register 1 (TPnIOC1) Format

Address: TP0IOC1: FFFFF593H, TP1IOC1: FFFFF5A3H
TP2IOC1: FFFFF5B3H, TP3IOC1: FFFFF5C3H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0	FFFFF593H to FFFFF5C3H	R/W	00H

TPnIS3	TPnIS2	Capture input (TIPn1) valid edge setting
0	0	No edge detection (capture operation is invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnIS1	TPnIS0	Capture input (TIPn0) valid edge detection
0	0	No edge detection (capture operation is invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions: 1. Rewrite bits TPnIS3 to TPnIS0 when TPnCE = 0 (the same value can be written when TPnCE = 1.). If rewriting was mistakenly performed, clear TPnCE to 0 and then set the bits again.

2. Bits TPnIS3 to TPnIS0 are valid only in the free-running mode and pulse width measurement mode. In all the other modes, capture operation is not performed.

3. If used as the capture input, be sure to set the corresponding alternate-function pins TPnOE1 and TPnOE0 of the TPnIOC0 register to “Timer output is disabled” and set the capture input valid edge. Then set the corresponding alternate-function port to input mode.

Remark: n = 0 to 3

(5) TMP I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge for external event count input signals (TIPn0) and external trigger input signal (TIPn0).
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-9: TMPn Dedicated I/O Control Register 2 (TPnIOC2) Format

Address: TP0IOC2: FFFFF594H, TP1IOC2: FFFFF5A4H
TP2IOC2: FFFFF5B4H, TP3IOC2: FFFFF5C4H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0	FFFFF594H to FFFFF5C4H	R/W	00H

TPnEES1	TPnEES0	External event count input valid edge setting (TIPn0)
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnETS1	TPnETS0	External trigger input valid edge detection (TIPn0)
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions: 1. Rewrite TPnEES1, TPnEES0, TPnETS1, and TPnETS0 bits when TPnCE = 0 (the same value can be written when TPnCE = 1.). If rewriting was mistakenly performed, clear TPnCE to 0 and then set the bits again.

2. TPnEES1 and TPnEES0 bits are valid only when TPnEEE = 1 or when the external event count mode has been set (TPnMD2 to TPnMD0 of TPnCTL1 register = 001).

Remark: n = 0 to 3

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register used to set the capture/compare operation and detect overflow.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-10: TMPn Option Register 0 (TPnOPT0) Format

Address: TP0OPT0: FFFFF595H, TP1OPT0: FFFFF5A5H
TP2OPT0: FFFFF5B5H, TP3OPT0: FFFFF5C5H

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
TPnOPT0	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF	FFFFF595H to FFFFF5C5H	R/W	00H

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TPnCCS1 bit setting is valid only in the free-running mode.	

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TPnCCS0 bit setting is valid only in the free-running mode.	

TPnOVF	Timer P overflow detection
Set (1)	Overflow occurrence
Reset (0)	TPnOVF bit write or TPnCE = 0
• The TPnOVF bit is reset when the 16-bit counter value overflows from FFFFH to 0000H in the free-running mode and the pulse width measurement mode. • An interrupt request signal (INTTPnOV) is generated as soon as TPnOVF bit is set (1). The INTTPnOV signal is not generated in any mode other than free-running mode and the pulse width measurement mode. • The TPnOVF bit is not cleared even when the TPnOVF bit and the TPnOPT0 register are read when TPnOVF = 1. • The TPnOVF bit can be both read and written, but 1 cannot be written to the TPnOVF bit from the CPU. Writing 1 has no influence on timer P operation.	

Caution: Rewrite TPnCCS1 and TPnCCS0 bits when TPnCE = 0 (the same value can be written when TPnCE = 1.). If rewriting was mistakenly performed, clear TPnCE to 0 and then set the bits again.

Remark: n = 0 to 3

(7) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TMPn.
This register can be read or written in 8-bit or 1-bit units.
RESET input clears this register to 00H.

Figure 7-11: Selector Operation Control Register 0 (SELCNT0) Format

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
SELCNT0	ISEL07	ISEL06	ISEL05	ISEL04	ISEL03	ISEL02	ISEL01	ISEL00	FFFFF308H	R/W	00H

ISEL07	Selection of TIP31 input signal (TMP3)
0	TIP31 pin input
1	Not permitted

ISEL06	Selection of TIP30 input signal (TMP3)
0	TIP30 pin input
1	RXDA1 pin input

ISEL05	Selection of TIP21 input signal (TMP2)
0	TIP21 pin input
1	Not permitted

ISEL04	Selection of TIP20 input signal (TMP2)
0	TIP20 pin input
1	RXDA0 pin input

ISEL03	Selection of TIP11 input signal (TMP1)
0	TIP11 pin input
1	Not permitted

ISEL02	Selection of TIP10 input signal (TMP1)
0	TIP10 pin input
1	TSOUT signal of CAN1

ISEL01[Note]	Selection of TIP01 input signal (TMP0)
0	TIP01 pin input
1	INTTM0EQ0 interrupt of TMM0

ISEL00	Selection of TIP00 input signal (TMP0)
0	TIP00 pin input
1	TSOUT signal of CAN0

Caution: To set ISEL07 to ISEL00 bits to 1 when permitted, set the corresponding pin in the capture input mode.

Note: When INTTM0EQ0 is used as TIP01 input signal, be sure that TMM operation clock cycle >= TMP operation clock cycle × 4.

(8) TIPnm pin noise elimination control register n (PnmNFC)

The PnmNFC register is an 8-bit register that sets the digital noise filter of the timer P input pin.
This register can be read or written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-12: TIPnm Pin Noise Elimination Control Register n (PnmNFC) Format

Address: P00NFC: FFFFFB00H (TIP00 pin)
P01NFC: FFFFFB04H (TIP01 pin)
P10NFC: FFFFFB08H (TIP10 pin)
P11NFC: FFFFFB0CH (TIP11 pin)
P20NFC: FFFFFB10H (TIP20 pin)
P21NFC: FFFFFB14H (TIP21 pin)
P30NFC: FFFFFB18H (TIP30 pin)
P31NFC: FFFFFB1CH (TIP31 pin)

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
PnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0	FFFFFB00H to FFFFFB1CH	R/W	00H

NFSTS	Selection of sampling times number for digital noise filtering
0	3 times
1	2 times

NFC2	NFC1	NFC0	Sampling clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
Other than above			Setting prohibited

Cautions: 1. Be sure to clear bits 3 to 5 and 7 to 0.

2. A signal input to the timer input pin (TIPnm) before the PnmNFC register is set is output with digital noise eliminated. Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the PnmNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation.

Remarks: 1. The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a width narrower than this may cause a miscount if it is synchronized with the sampling clock.

2. n: Number of timer channels (0 to 3)
m: Number of input pins (0, 1)

7.5 Operation

Timer P can perform the following operations.

Operation	TPnEST Software trigger bit	TIPn0 External trigger input	TPnEEE Count clock selection	Capture/Compare Selection	Compare Write
Interval timer mode	Invalid	Invalid	Internal/external	Compare only	Any time write
External event counter mode**Note 1**	Invalid	Invalid	External only	Compare only	Any time write
External trigger pulse output mode**Note 2**	Valid	Valid	Internal only	Compare only	Reload
One-shot pulse output mode**Note 2**	Valid	Valid	Internal only	Compare only	Any time write
PWM mode	Invalid	Invalid	Internal/external	Compare only	Reload
Free-running mode	Invalid	Invalid	Internal/external	Capture/compare selectable	Any time write
Pulse width measurement mode**Note 2**	Invalid	Invalid	Internal only	Capture only	Not applicable

Notes: 1. To use the external event counter function, specify that the input edge of the TIPn0 pin is not detected by clearing the TPnIS1 and TPnIS0 bits of the TPnIOC1 register to “00”.

2. To use the external trigger pulse output mode, one-shot pulse mode, or pulse width measurement mode, select a count clock by clearing the TPnEEE bit of the TPnCTL1 register to 0.

7.5.1 Anytime write and reload

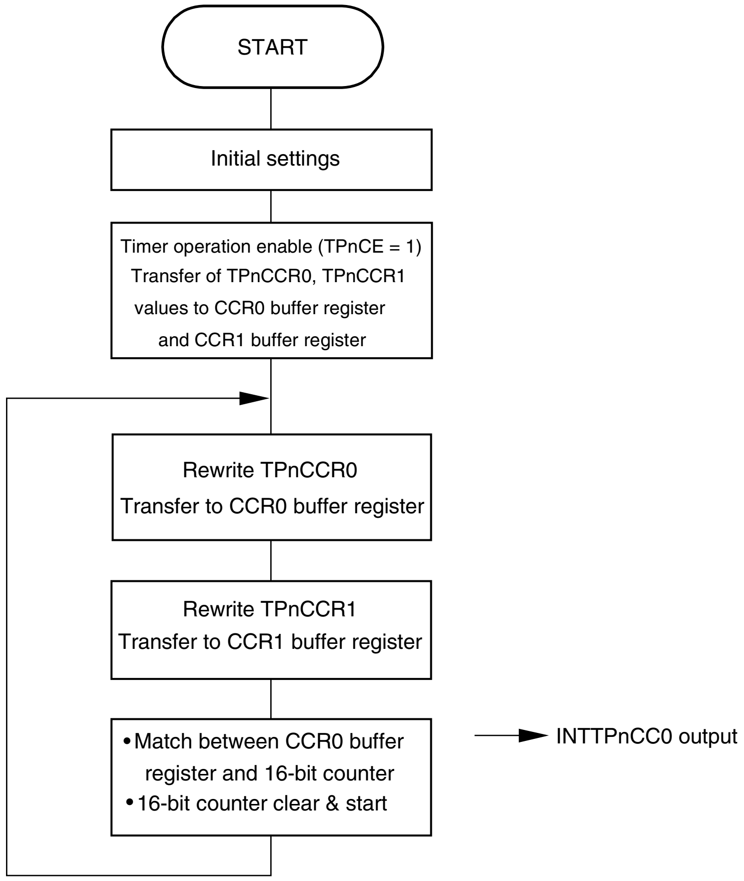
TPnCCR0 and TPnCCR1 register rewrite is possible for timer P during timer operation (TPnCE = 1), but the write method (any time write, reload) differs depending on the mode.

(1) Anytime write

When data is written to the TPnCCRm register during timer operation, it is transferred at any time to CCRm buffer register and used as the 16-bit counter comparison value.

Remark: n = 0 to 3
m = 0, 1

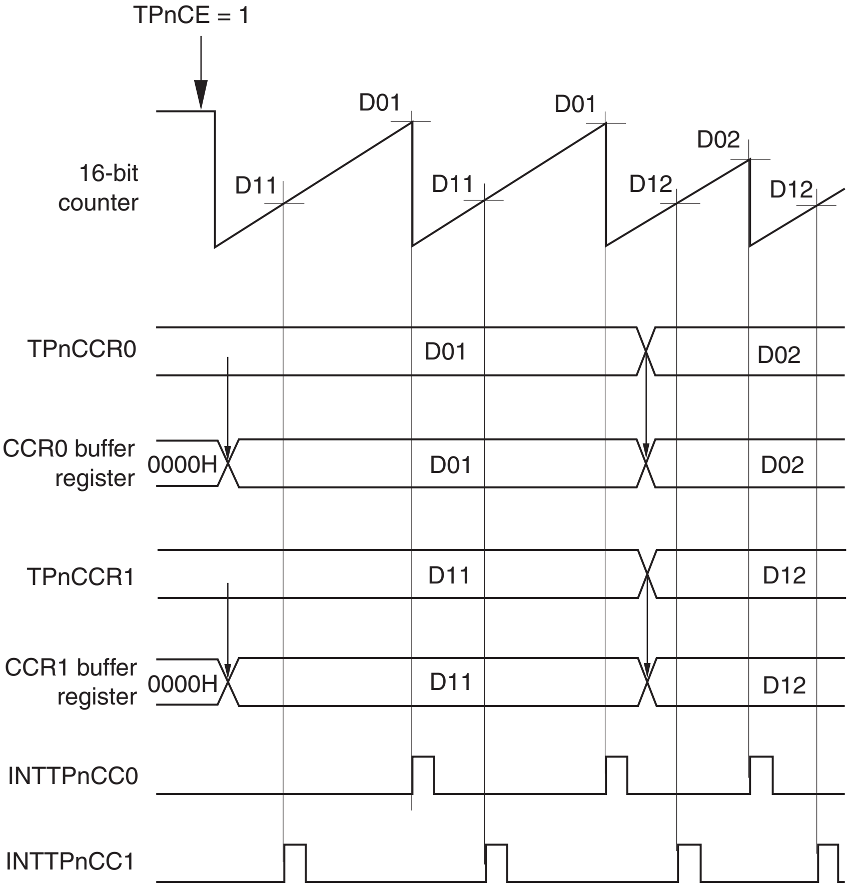
Figure 7-13: Flowchart of Basic Operation for Anytime Write


START
Initial settings
Timer operation enable (TPnCE = 1)
Transfer of TPnCCR0, TPnCCR1
values to CCR0 buffer register
and CCR1 buffer register
Rewrite TPnCCR0
Transfer to CCR0 buffer register
Rewrite TPnCCR1
Transfer to CCR1 buffer register
• Match between CCR0 buffer
register and 16-bit counter
• 16-bit counter clear & start
INTTPnCC0 output

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

2. n = 0 to 3

Figure 7-14: Timing Diagram for Anytime Write



Remarks: 1. D01, D02: Setting values of TPnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TPnCCR1 register (0000H to FFFFH)

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

3. n = 0 to 3

(2) Reload

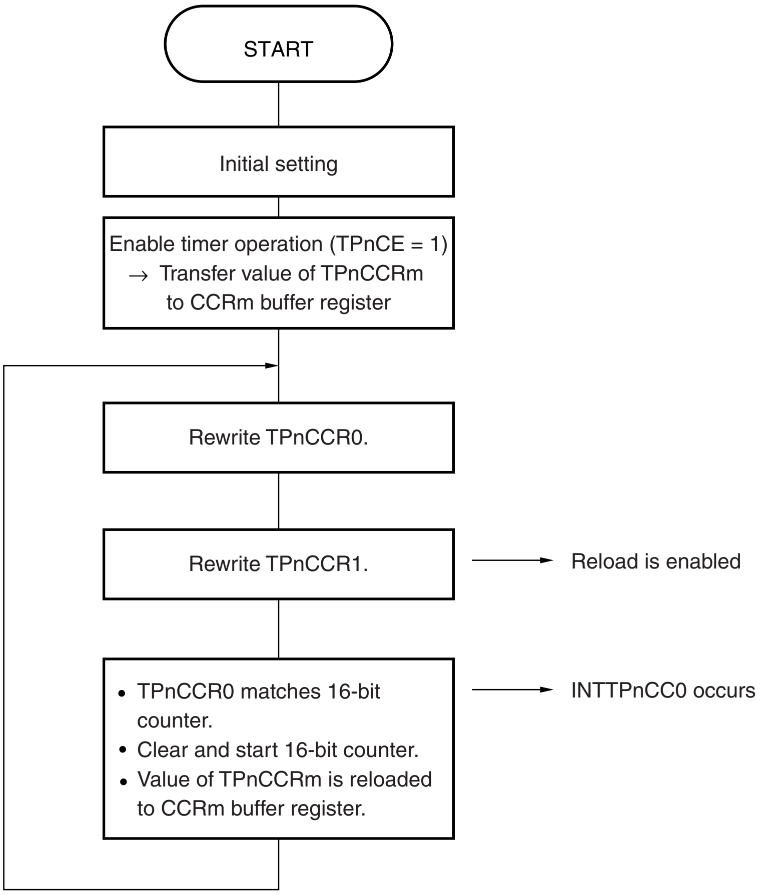
When data is written to the TPnCCR0 and TPnCCR1 registers during timer operation, it is compared with the value of the 16-bit counter via the CCRm buffer register. The values of the TPnCCR0 and TPnCCR1 registers can be rewritten when TPnCE = 1.

So that the set values of the TPnCCR0 and TPnCCR1 registers are compared with the value of the 16-bit counter (the set values are reloaded to the CCRm buffer register), the value of the TPnCCR0 register must be rewritten and then a value must be written to the TPnCCR1 register before the value of the 16-bit counter

matches the value of TPnCCR0. When the value of the TPnCCR0 register matches the value of the 16-bit counter, the values of the TPnCCR0 and TPnCCR1 registers are reloaded.

Whether the next reload timing is made valid or not is controlled by writing to the TPnCCR1 register. Therefore, write the same value to the TPnCCR1 register when it is necessary to rewrite the value of only the TPnCCR0 register.

Figure 7-15: Flowchart of Basic Operation for Reload

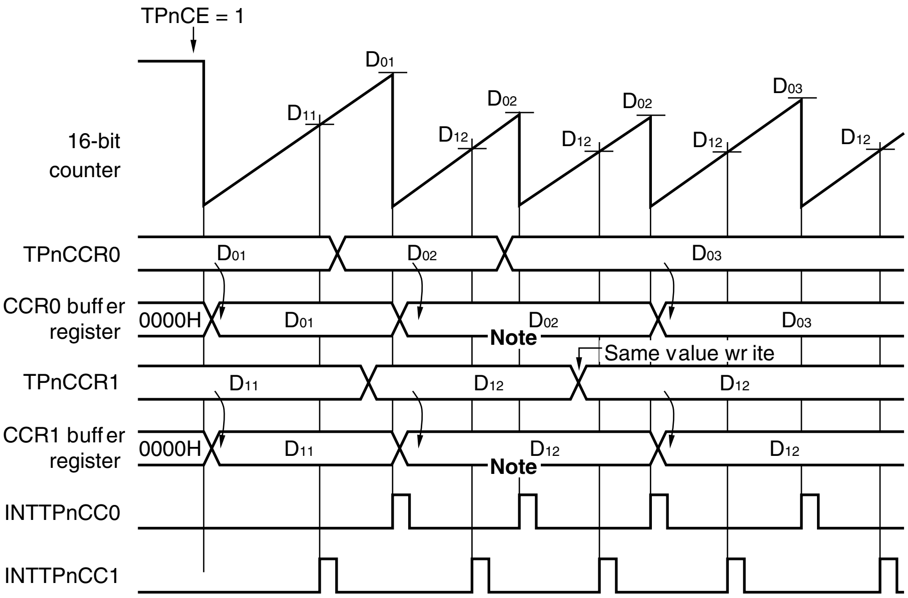

START
Initial setting
Enable timer operation (TPnCE = 1)
→ Transfer value of TPnCCRm
to CCRm buffer register
Rewrite TPnCCR0.
Rewrite TPnCCR1.
Reload is enabled
• TPnCCR0 matches 16-bit counter.
• Clear and start 16-bit counter.
• Value of TPnCCRm is reloaded to CCRm buffer register.
INTTPnCC0 occurs

Caution: Writing to the TPnCCR1 register includes an operation to enable reload. Therefore, rewrite the TPnCCR1 register after rewriting the TPnCCR0 register.

Remarks: **1.** Above flowchart illustrates an example of the PWM mode operation.

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

Figure 7-16: Timing Chart for Reload



Note: Reload is not performed because TPnCCR1 register is not written.

Remarks: 1. D01, D02, D03: Setting value of TPnCCR0 register (0000H to FFFFH)
D11, D12: Setting value of TPnCCR1 register (0000H to FFFFH)

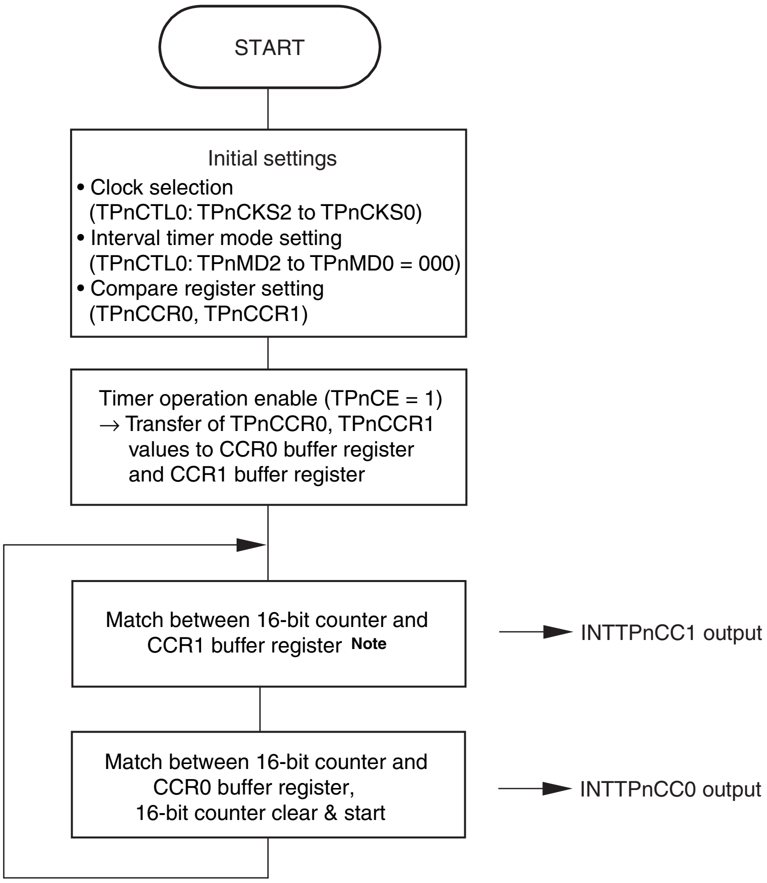
2. Above flowchart illustrates PWM mode operation.

3. n = 0 to 3

7.5.2 Interval timer mode (TPnMD2 to TPnMD0 = 000)

In the interval timer mode, an interrupt request signal (INTTPnCC0) is generated upon a match between the setting value of the TPnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared. The TPnCCR0 register can be rewritten when TPnCE = 1, and when a value is set to TPnCCR0 with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write mode, and is compared with the 16-bit counter value.
In the interval timer mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.
16-bit counter clearing using the TPnCCR1 register is not performed. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.
Moreover, TOP1n pin output is also possible by setting the TPnOE1 bit to 1.
When the TPnCCR1 register is not used, it is recommended to set FFFFH as the setting value for the TPnCCR1 register.
When performing timer output with the TOPn1 pin, set the same values to the TPnCCR0 register and the TPnCCR1 register since the 16-bit timer counter cannot be cleared with the TPnCCR1 register.

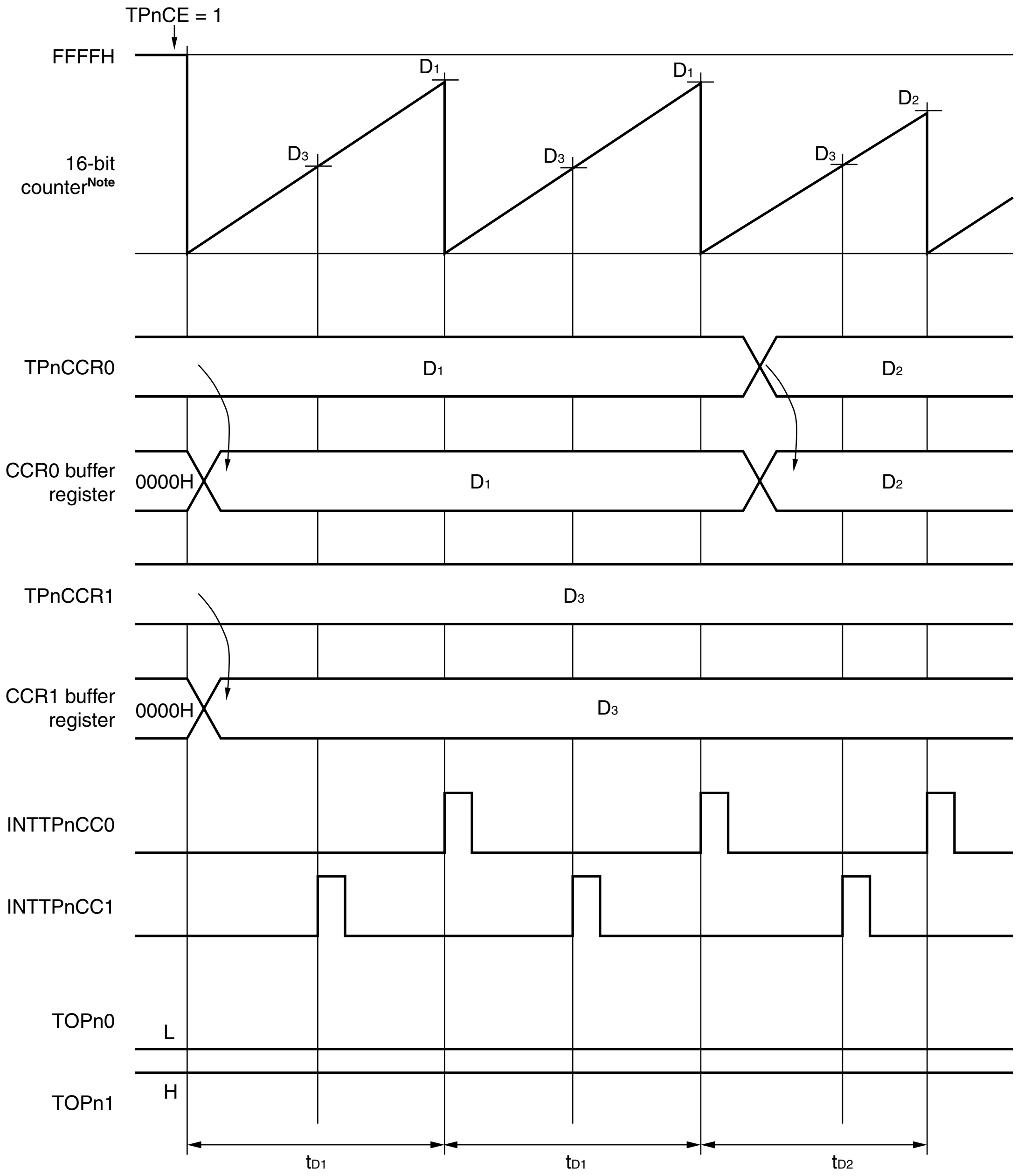
Figure 7-17: Flowchart of Basic Operation in Interval Timer Mode


START
Initial settings
• Clock selection
(TPnCTL0: TPnCKS2 to TPnCKS0)
• Interval timer mode setting
(TPnCTL0: TPnMD2 to TPnMD0 = 000)
• Compare register setting
(TPnCCR0, TPnCCR1)
Timer operation enable (TPnCE = 1)
→ Transfer of TPnCCR0, TPnCCR1 values to CCR0 buffer register and CCR1 buffer register
Match between 16-bit counter and CCR1 buffer register Note
INTTPnCC1 output
Match between 16-bit counter and CCR0 buffer register, 16-bit counter clear & start
INTTPnCC0 output

Note: The 16-bit counter is not cleared when its value matches the value of TPnCCR1.

Figure 7-18: Basic Operation Timing in Interval Timer Mode (1/2)

(a) When D1 > D2 > D3; only TPnCCR0 register value is written, and TOPn0 and TOPn1 are not output (TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 1)



Note: The 16-bit counter is not cleared when its value matches the value of TPnCCR1.

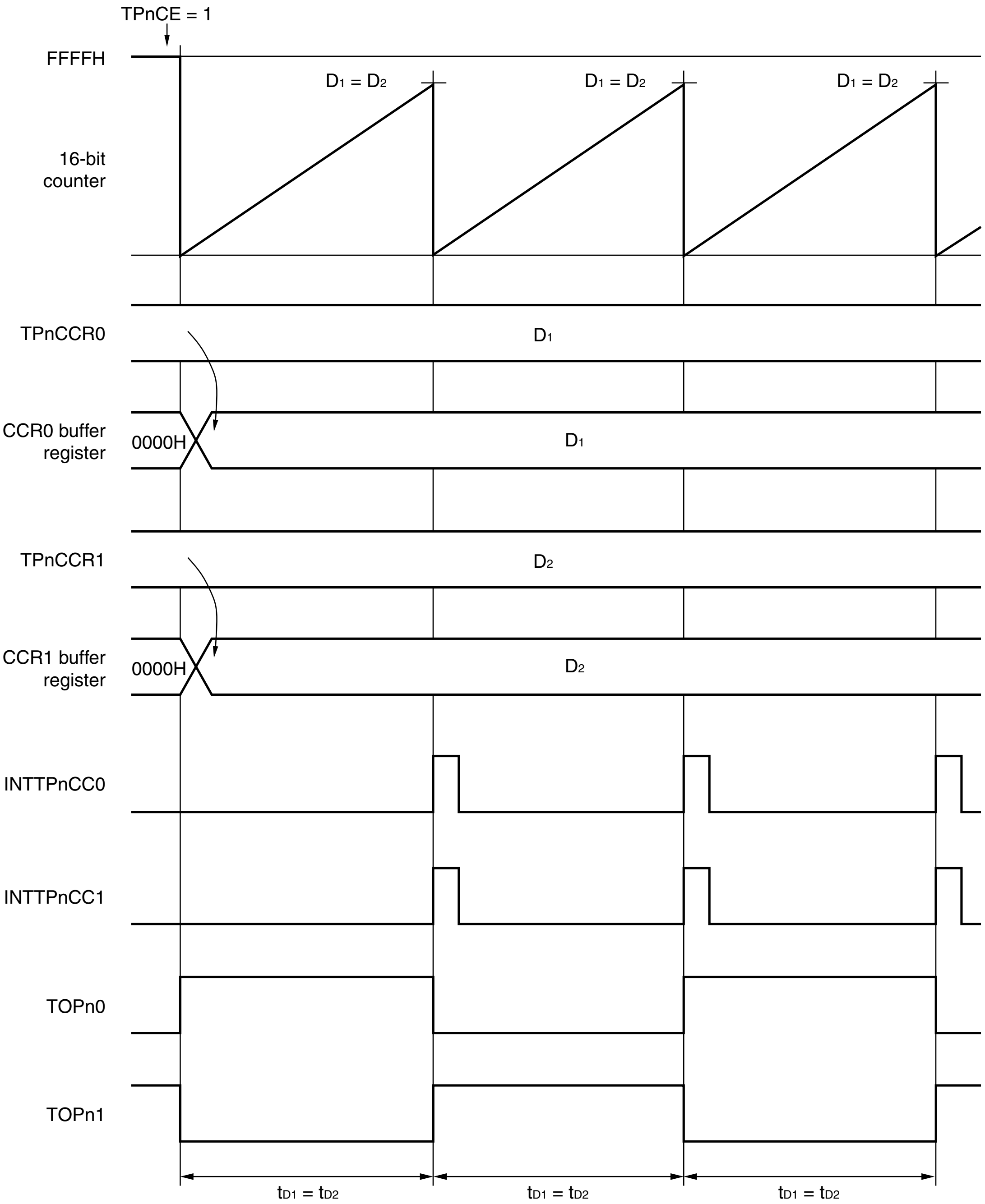
Remarks: 1. D1, D2: Setting values of TPnCCR0 register (0000H to FFFFH)
D3: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Interval time (t_{Dn})= (Dn + 1) × (count clock cycle)

3. n = 0 to 3

Figure 7-18: Basic Operation Timing in Interval Timer Mode (2/2)

(b) When D1 = D2; TPnCCR0 and TPnCCR1 are not rewritten, and TOPn0 and TOPn1 are output (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)



Remarks: 1. D1: Setting value of TPnCCR0 register (0000H to FFFFH)
D2: Setting value of TPnCCR1 register (0000H to FFFFH)

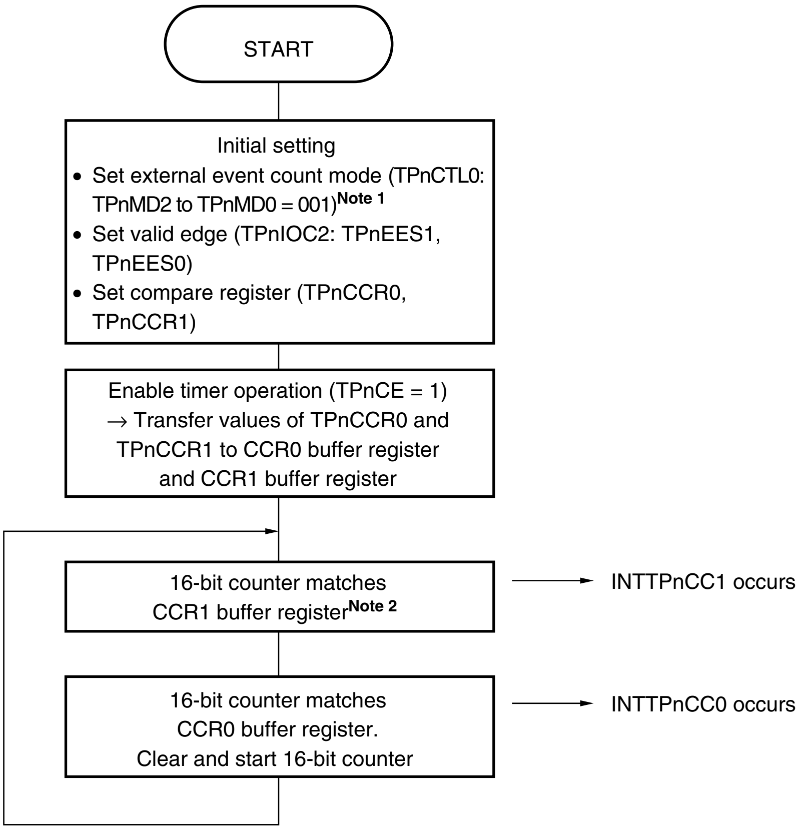
2. Interval time (t_{Dn}) = (Dn + 1) × (count clock cycle)

3. n = 0 to 3

7.5.3 External event counter mode (TPnMD2 to TPnMD0 = 001)

In the external event count mode, the external event count input (TIPn0 pin input) is used as a count-up signal. Regardless of the setting of the TPnEEE bit of the TPnCTL0 register, 16-bit timer/event counter P counts up the external event count input (TIPn0 pin input) when it is set in the external event count mode. In the external event count mode, an interrupt request (INTTPnCC0) is generated when the set value of the TPnCCR0 register matches the value of the 16-bit counter, and the value of the 16-bit counter is cleared.
When a value is set to the TPnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write, and is compared with the 16-bit counter value.
In the external event counter mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.
The 16-bit counter can not be cleared using TPnCCR1 register. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.
Moreover, TOPnm pin output is also possible by setting the TPnOEm bit to 1.
When performing timer output with the TOPn1 pin, set the same values to TPnCCR0 register and TPnCCR1 register since the 16-bit counter cannot be cleared with CCR1 buffer register.
The TPnCCR0 register can be rewritten when TPnCE = 1. When TPnCCR1 register is not used, it is recommended to set TPnCCR1 register to FFFFH.

Figure 7-19: Flowchart of Basic Operation in External Event Counter Mode


START
Initial setting
• Set external event count mode (TPnCTL0: TPnMD2 to TPnMD0 = 001)Note 1
• Set valid edge (TPnIOC2: TPnEES1, TPnEES0)
• Set compare register (TPnCCR0, TPnCCR1)
Enable timer operation (TPnCE = 1)
→ Transfer values of TPnCCR0 and TPnCCR1 to CCR0 buffer register and CCR1 buffer register
16-bit counter matches CCR1 buffer registerNote 2
INTTPnCC1 occurs
16-bit counter matches CCR0 buffer register. Clear and start 16-bit counter
INTTPnCC0 occurs

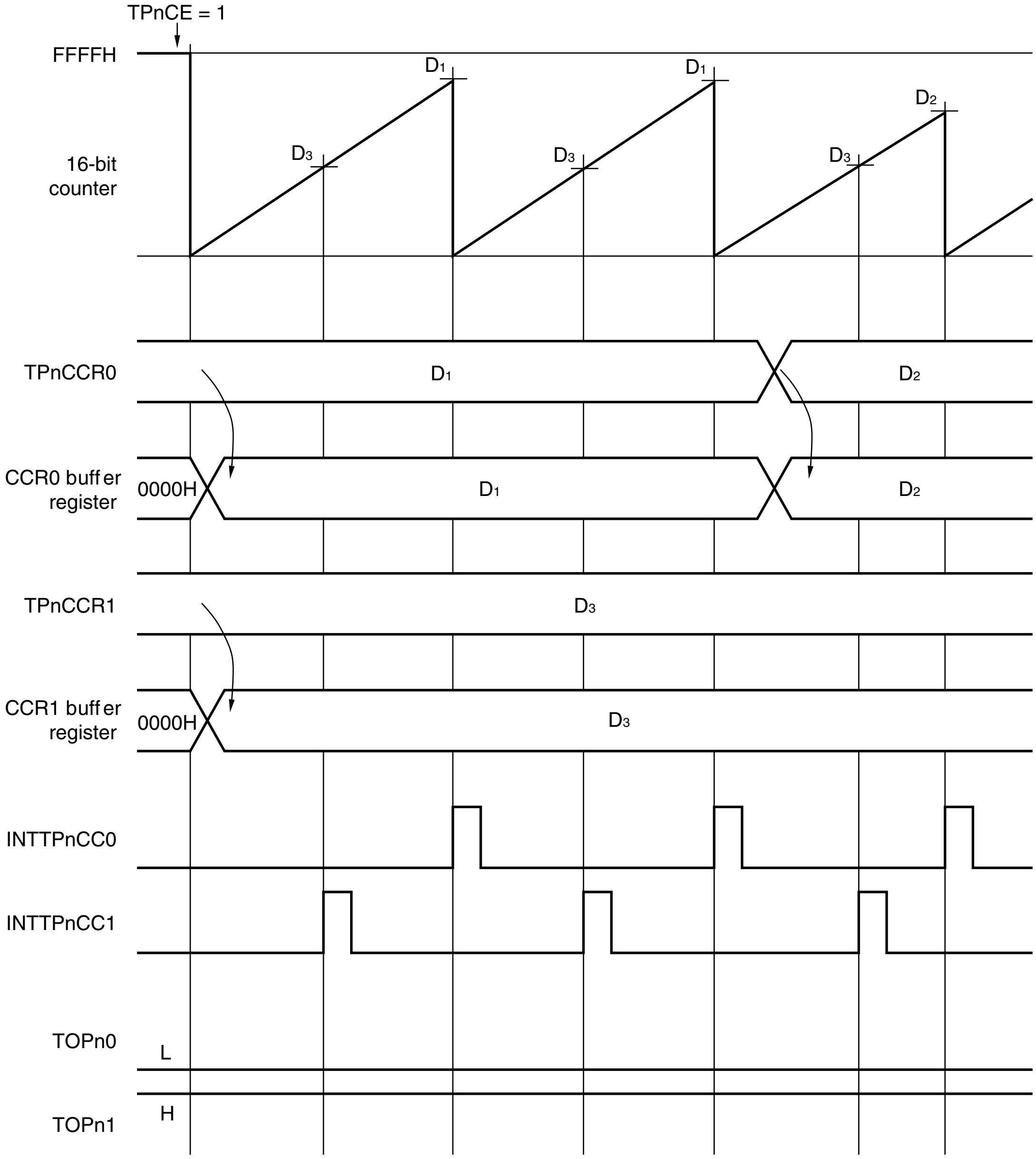
Notes: 1. Selection of the TPnEEE bit has no influence.

2. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Remark: n = 0 to 3
m = 0, 1

Figure 7-20: Basic Operation Timing in External Event Counter Mode (1/2)

(a) When D1 > D2 > D3; rewrite TPnCCR0 only; TOPn1 and TOPn0 are not output (TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 1)



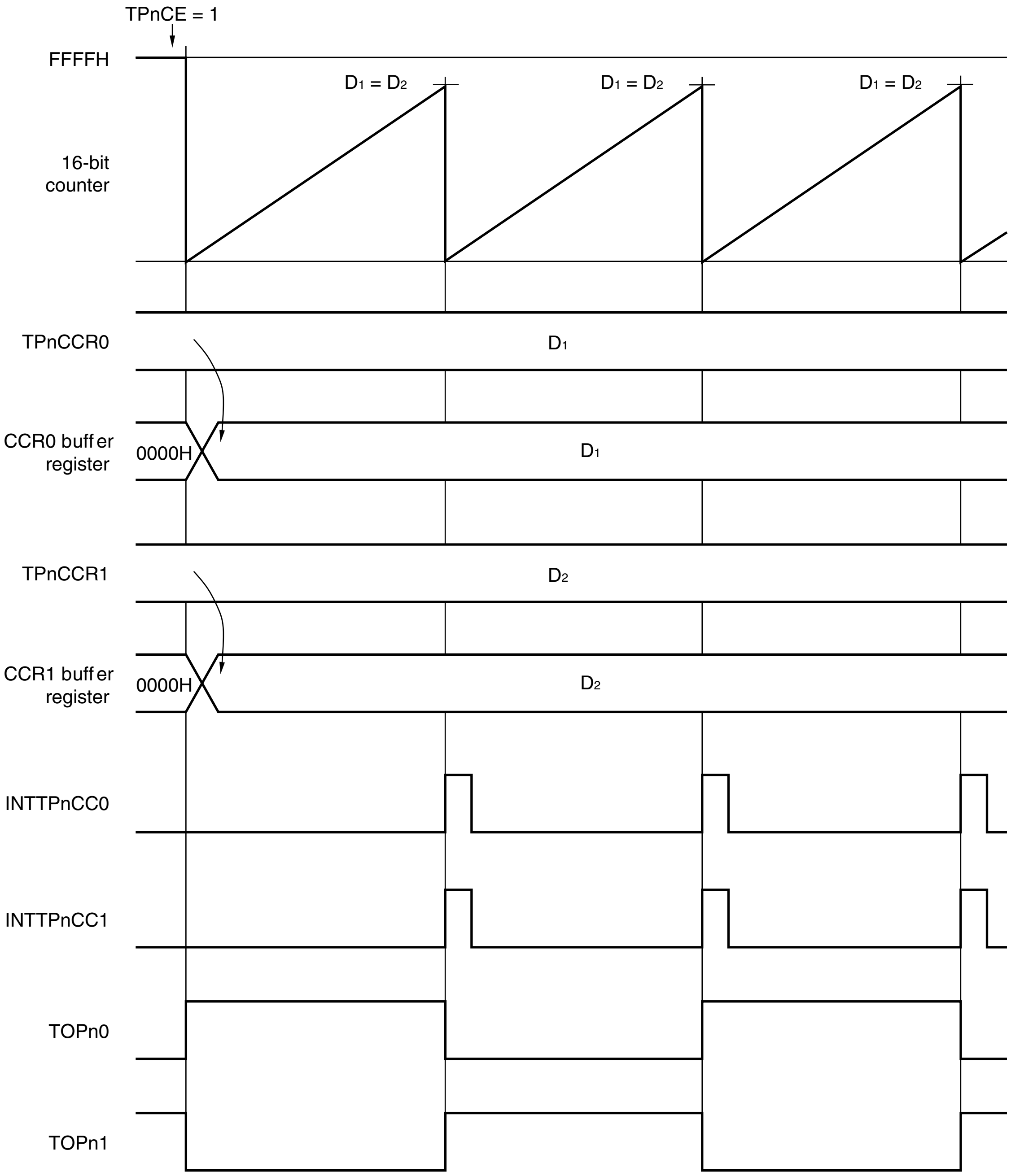
Remarks: 1. D1, D2: Setting values of TPnCCR0 register (0000H to FFFFH)
D3: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dn + 1)

3. n = 0 to 3

Figure 7-20: Basic Operation Timing in External Event Counter Mode (2/2)

(b) When D1 = D2; TPnCCR0 and TPnCCR1 are not rewritten, and TOPn0 and TOPn1 are output (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)



Remarks: 1. D1: Setting value of TPnCCR0 register (0000H to FFFFH)
D2: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Number of event count = (Dn + 1)

3. n = 0 to 3

7.5.4 External trigger pulse mode (TPnMD2 to TPnMD0 = 010)

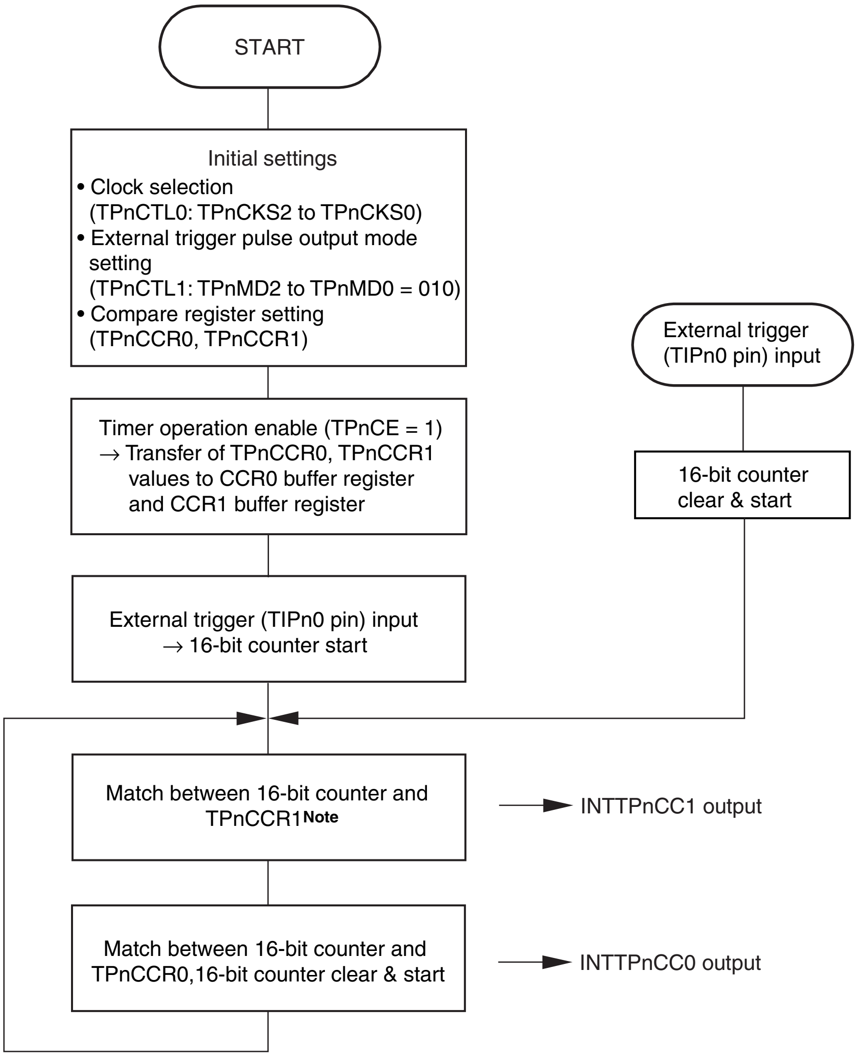
When TPnCE = 1 in the external trigger pulse mode, the 16-bit counter stops at FFFFH and waits for input of an external trigger (TIPn0 pin input). When the counter detects the edge of the external trigger (TIPn0 pin input), it starts counting up.
The duty factor of the signal output from the TOPn1 pin is set by a reload register (TPnCCR1) and the period is set by a compare register (TPnCCR0).
Rewriting the TPnCCR0 and TPnCCR1 registers is enabled when TPnCE = 1.
So that the set values of the TPnCCR0 and TPnCCR1 registers after rewriting are compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), the TPnCCR0 register must be rewritten and then a value is written to the TPnCCR1 register before the value of the 16-bit counter matches the value of the TPnCCR0 register.
When the value of the TPnCCR0 register later matches the value of the 16-bit counter, the values of the TPnCCR0 and TPnCCR1 registers are reloaded to the CCRm buffer register.
Whether the next reload timing is made valid or not is controlled by writing to the TPnCCR1 register. Therefore, write the same value to the TPnCCR1 register when it is necessary to rewrite the value of only the TPnCCR0 register.
Reload is invalid when only the TPnCCR0 register is rewritten. To stop timer P, clear TPnCE to 0. If the edge of the external trigger (TIPn0 pin input) is detected more than once in the external trigger pulse mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. To realize the same function as the external trigger pulse mode by using a software trigger instead of the external trigger input (TIPn0 pin input) (software trigger pulse mode), a software trigger is generated by setting the TPnEST bit of the TPnCTL1 register to 1. The waveform of the external trigger pulse is output from TOPn1. A toggle output is produced from the TOPn0 pin when the value of the TPnCCR0 register matches the value of the 16-bit counter.
In the external trigger pulse mode, the capture function of the TPnCCR0 and TPnCCR1 registers cannot be used because these registers can be used only as compare registers.

Caution: In the external trigger pulse mode, select the internal clock (TPnEEE bit of TPnCTL1 register = 0) for the count clock.

Remarks: 1. For the reload operation when TPnCCR0 and TPnCCR1 are rewritten during timer operation, refer to section **7.5.6 PWM mode** (TPnMD2 to TPnMD0 = 100).

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

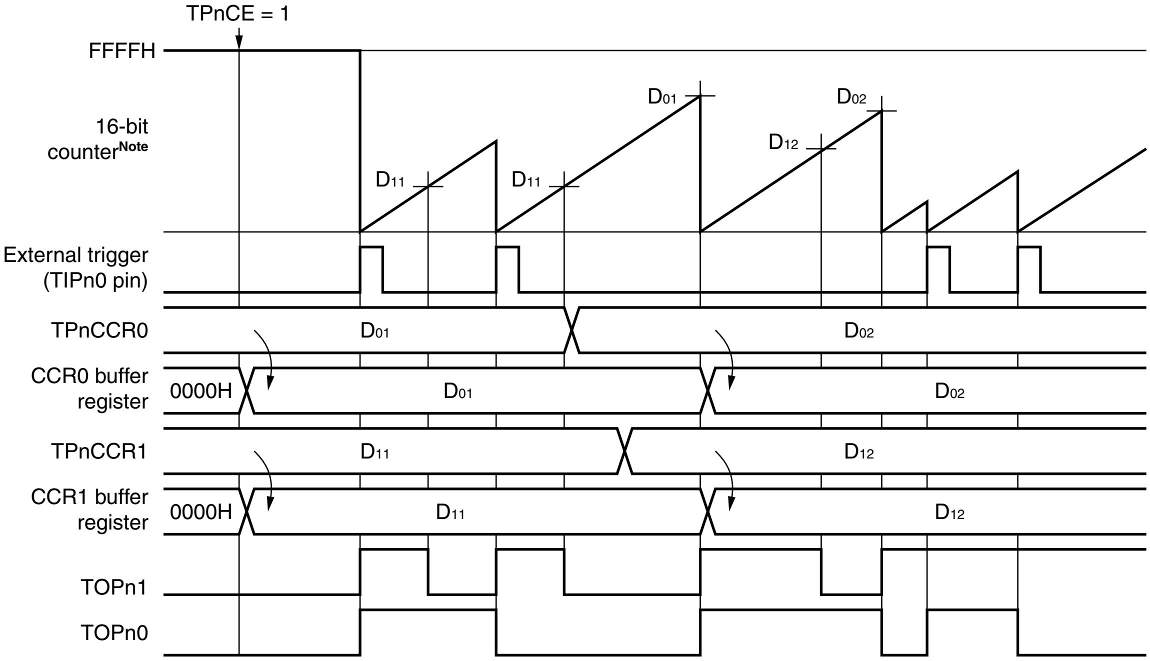
Figure 7-21: Flowchart of Basic Operation in External Trigger Pulse Output Mode


START
Initial settings
• Clock selection
(TPnCTL0: TPnCKS2 to TPnCKS0)
• External trigger pulse output mode setting
(TPnCTL1: TPnMD2 to TPnMD0 = 010)
• Compare register setting
(TPnCCR0, TPnCCR1)
Timer operation enable (TPnCE = 1)
→ Transfer of TPnCCR0, TPnCCR1 values to CCR0 buffer register and CCR1 buffer register
External trigger (TIPn0 pin) input
→ 16-bit counter start
External trigger
(TIPn0 pin) input
16-bit counter
clear & start
Match between 16-bit counter and TPnCCR1Note
INTTPnCC1 output
Match between 16-bit counter and TPnCCR0,16-bit counter clear & start
INTTPnCC0 output

Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Remark: n = 0 to 3

Figure 7-22: Basic Operation Timing in External Trigger Pulse Output Mode (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Note: The 16-bit counter is not cleared when it matches the CCR1 buffer register.

Remarks: 1. D01, D02: Setting value of TPnCCR0 register (0000H to FFFFH)
D11, D12: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (Set value of TPnCCR1 register) / (Set value of TP0CCR0 register)
Cycle of TOPn1 output = (Set value of TPnCCR0 register) Þ (Count clock cycle)

3. n = 0 to 3

7.5.5 One-shot pulse mode (TPnMD2 to TPnMD0 = 011)

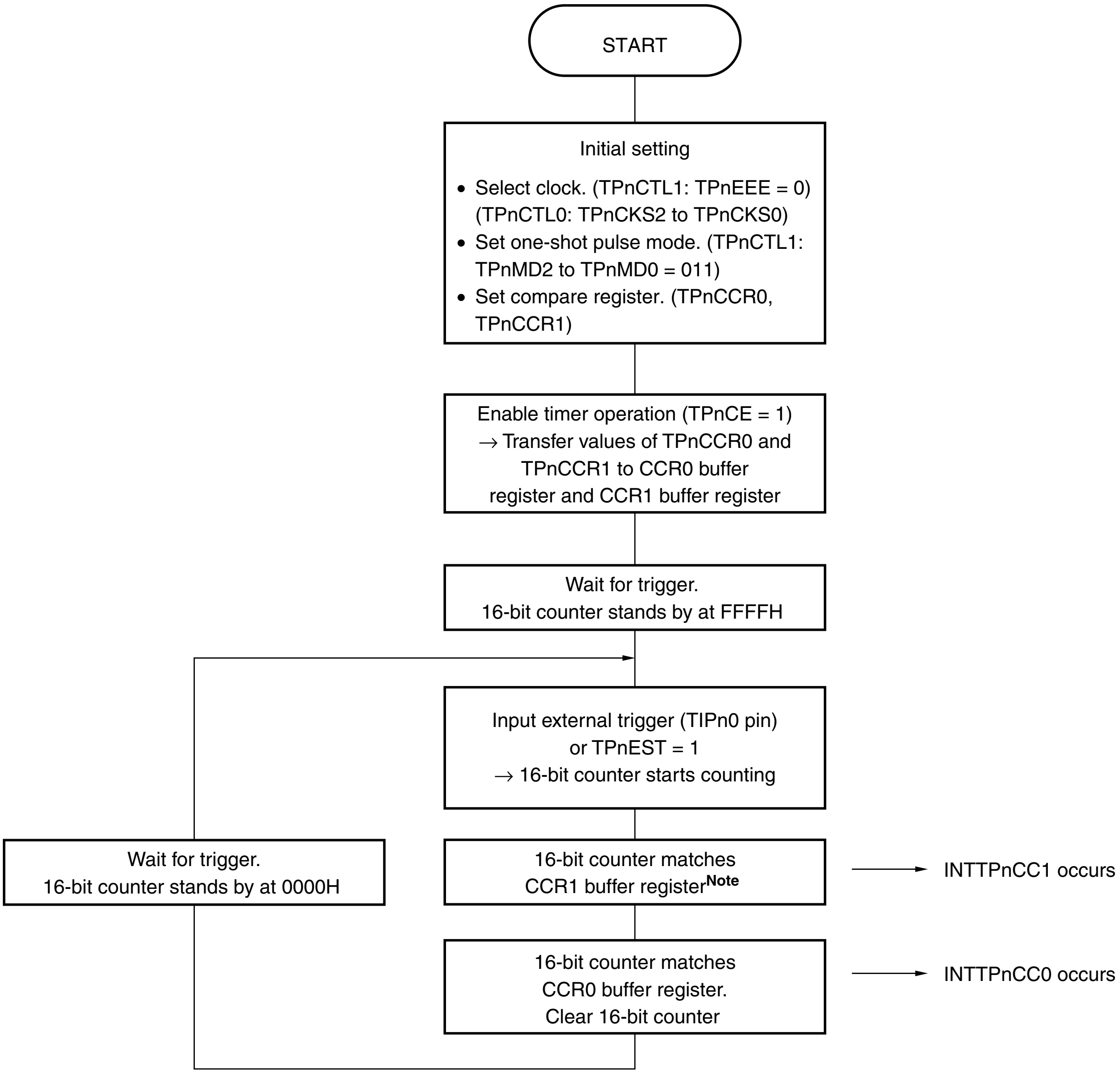
When TPnCE is set to 1 in the one-shot pulse mode, the 16-bit counter waits for the setting of the TPnEST bit (to 1) or a trigger that is input when the edge of the TIPn0 pin is detected, while holding FFFFH. When the trigger is input, the 16-bit counter starts counting up. When the value of the 16-bit counter matches the value of the CCR1 buffer register that has been transferred from the TPnCCR1 register, TOPn1 goes high. When the value of the 16-bit counter
matches the value of the CCR0 buffer register that has been transferred from the TPnCCR0 register, TOPn1 goes low, and the 16-bit counter is cleared to 0000H and stops. Input of a second or
subsequent trigger is ignored while the 16-bit counter is operating. Be sure to input a second trigger while the 16-bit counter is stopped at 0000H. In the one shot pulse mode, rewriting the TPnCCR0 and TPnCCR1 registers is enabled when TPnCE = 1. The set values of the TPnCCR0 and TPnCCR1 registers become valid after a write instruction from the CPU is executed. They are then transferred to the CCR0 and CCR1 buffer registers, and compared with the value of the 16-bit counter. The waveform of the one-shot pulse is output from the TOPn1 pin. The TOPn0 pin produces a toggle output when the value of the 16-bit counter matches the value of the TPnCCR0 register. In the one-shot pulse mode, the TPnCCR0 and TPnCCR1 registers function only as compare registers. They cannot be used as capture registers.

Cautions: 1. In the one-shot pulse mode, select the internal clock (TPnEEE bit of TPnCTL1 register = 0) as the count clock.

2. In the one-shot pulse mode, it is prohibited to set the TPnCCR1 register to 0000H.

Remark: n = 0 to 3

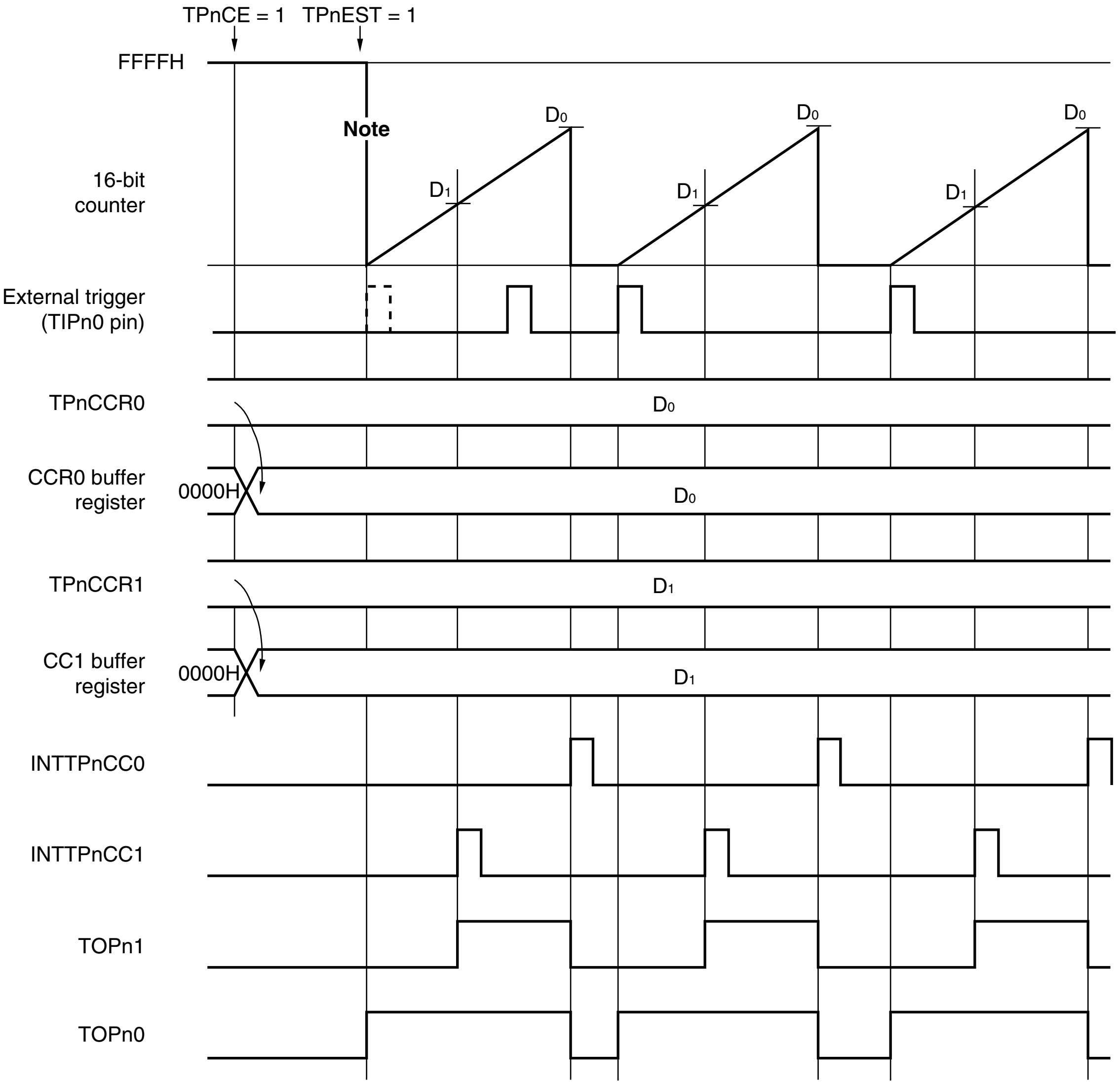
Figure 7-23: Flowchart of Basic Operation in One-Shot Pulse Mode


START
Initial setting
• Select clock. (TPnCTL1: TPnEEE = 0) (TPnCTL0: TPnCKS2 to TPnCKS0)
• Set one-shot pulse mode. (TPnCTL1: TPnMD2 to TPnMD0 = 011)
• Set compare register. (TPnCCR0, TPnCCR1)
Enable timer operation (TPnCE = 1) → Transfer values of TPnCCR0 and TPnCCR1 to CCR0 buffer register and CCR1 buffer register
Wait for trigger. 16-bit counter stands by at FFFFH
Input external trigger (TIPn0 pin) or TPnEST = 1 → 16-bit counter starts counting
Wait for trigger. 16-bit counter stands by at 0000H
16-bit counter matches CCR1 buffer registerNote
INTTPnCC1 occurs
16-bit counter matches CCR0 buffer register. Clear 16-bit counter
INTTPnCC0 occurs

Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Caution: The 16-bit counter is not cleared when a trigger input is performed during the count-up operation of the 16-bit counter.

Figure 7-24: Timing of Basic Operation in One-Shot Pulse Mode (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Note: The 16-bit counter starts counting up when either TPnEST = 1 is set or TPTTRG is input.

Remarks: 1. D0: Setting value of TPnCCR0 register (0000H to FFFFH)
D1: Setting value of TPnCCR1 register (0000H to FFFFH)

2. n = 0 to 3

7.5.6 PWM mode (TPnMD2 to TPnMD0 = 110)

In the PWM mode, TMPn capture/compare register 1 (TPnCCR1) is used to set the duty factor and TMPn capture/compare register 0 (TPnCCR0) is used to set the cycle.
By using these two registers and operating the timer, variable-duty PWM is output.
Rewriting the TPnCCR0 and TPnCCR1 registers is enabled when TPnCE = 1.
So that the set values of the TPnCCR0 and TPnCCR1 registers are compared with the value of the 16-bit counter (reloaded to the CCR0 and CCR1 buffer registers), the TPnCCR0 register must be rewritten and then a value must be written to the TPnCCR1 register before the value of the 16-bit counter matches the value of the TPnCCR0 register.
The values of the TPnCCR0 and TPnCCR1 registers are reloaded when the value of the TPnCCR0 register later matches the value of the 16-bit counter. Whether the next reload timing is made valid or not is controlled by writing to the TPnCCR1 register. Therefore, write the same value to the TPnCCR1 register even when only the value of the TPnCCR0 register needs to be rewritten. Reload is invalid when only the value of the TPnCCR0 register is rewritten. To stop timer P, clear TPnCE to 0. The waveform of PWM is output from the TOPn1 pin. The TOPn0 pin produces a toggle output when the 16-bit counter matches the TPnCCR0 register.
In the PWM mode, the TPnCCR0 and TPnCCR1 registers are used only as compare registers. They cannot be used as capture registers.

Figure 7-25: Flowchart of Basic Operation in PWM Mode (1/2)

(a) When values of TPnCCR0, TPnCCR1 registers are not rewritten during timer operation

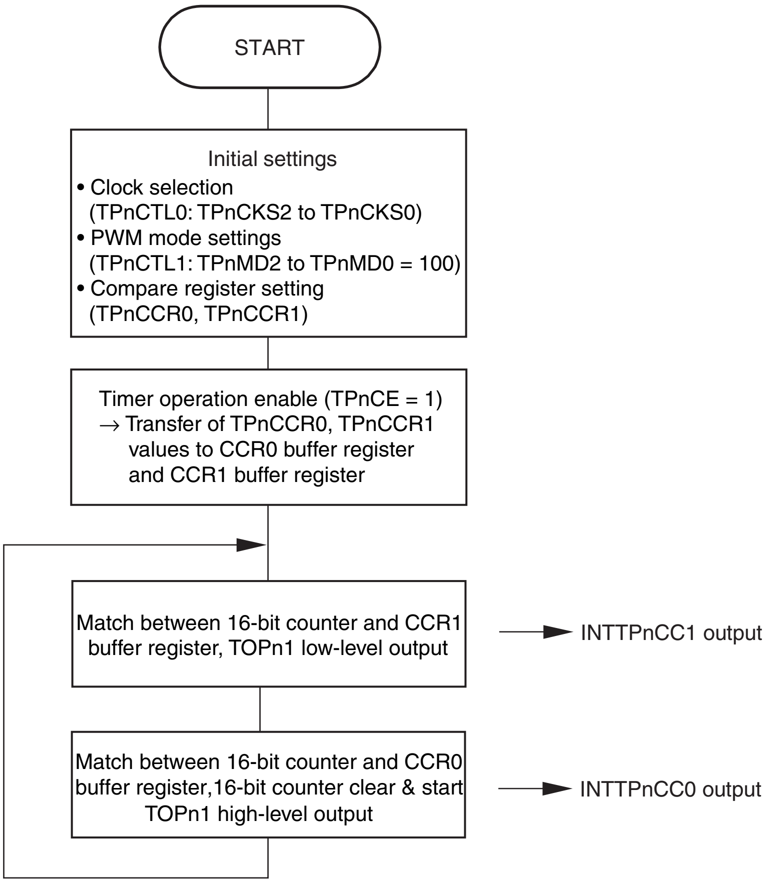
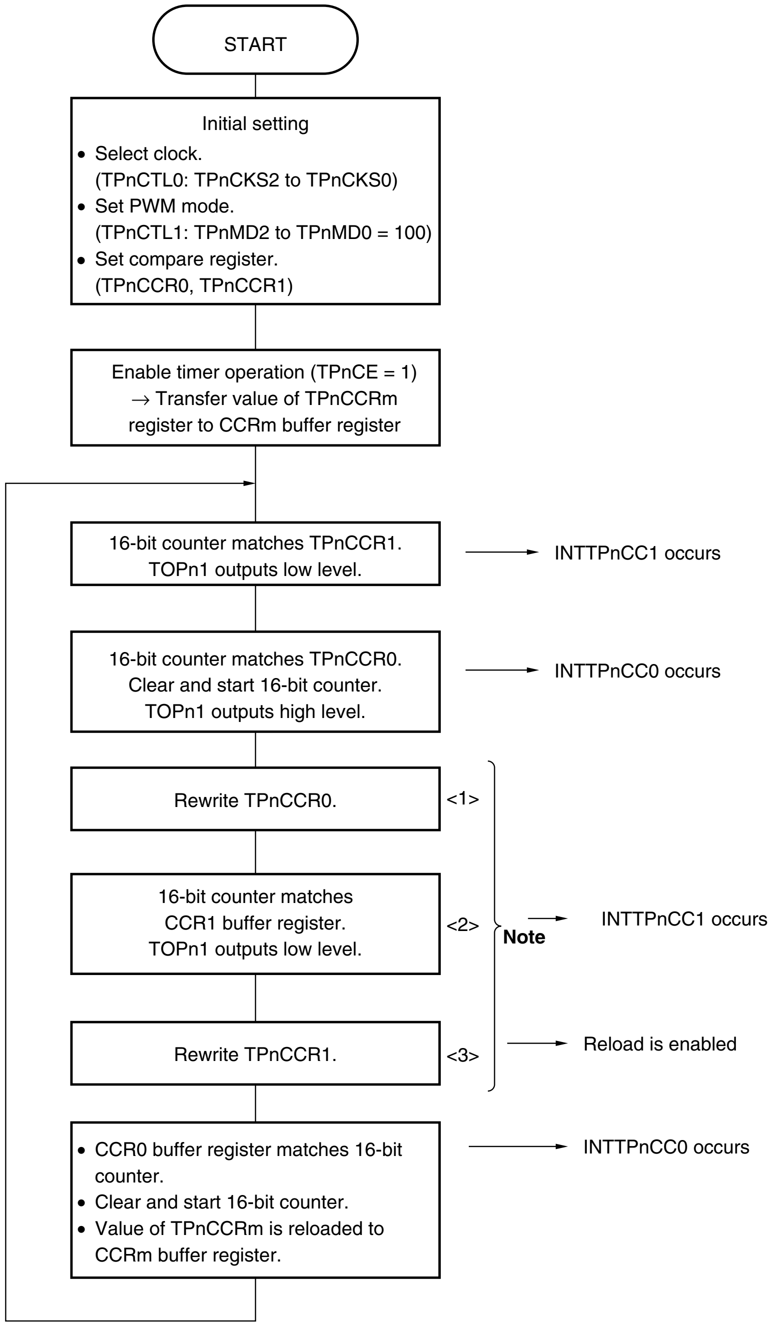

START
Initial settings
• Clock selection
(TPnCTL0: TPnCKS2 to TPnCKS0)
• PWM mode settings
(TPnCTL1: TPnMD2 to TPnMD0 = 100)
• Compare register setting
(TPnCCR0, TPnCCR1)
Timer operation enable (TPnCE = 1)
→ Transfer of TPnCCR0, TPnCCR1 values to CCR0 buffer register and CCR1 buffer register
Match between 16-bit counter and CCR1 buffer register, TOPn1 low-level output
INTTPnCC1 output
Match between 16-bit counter and CCR0 buffer register,16-bit counter clear & start TOPn1 high-level output
INTTPnCC0 output

Figure 7-25: Flowchart of Basic Operation in PWM Mode (2/2)

(b) When values of TPnCCR0, TPnCCR1 registers are rewritten during timer operation

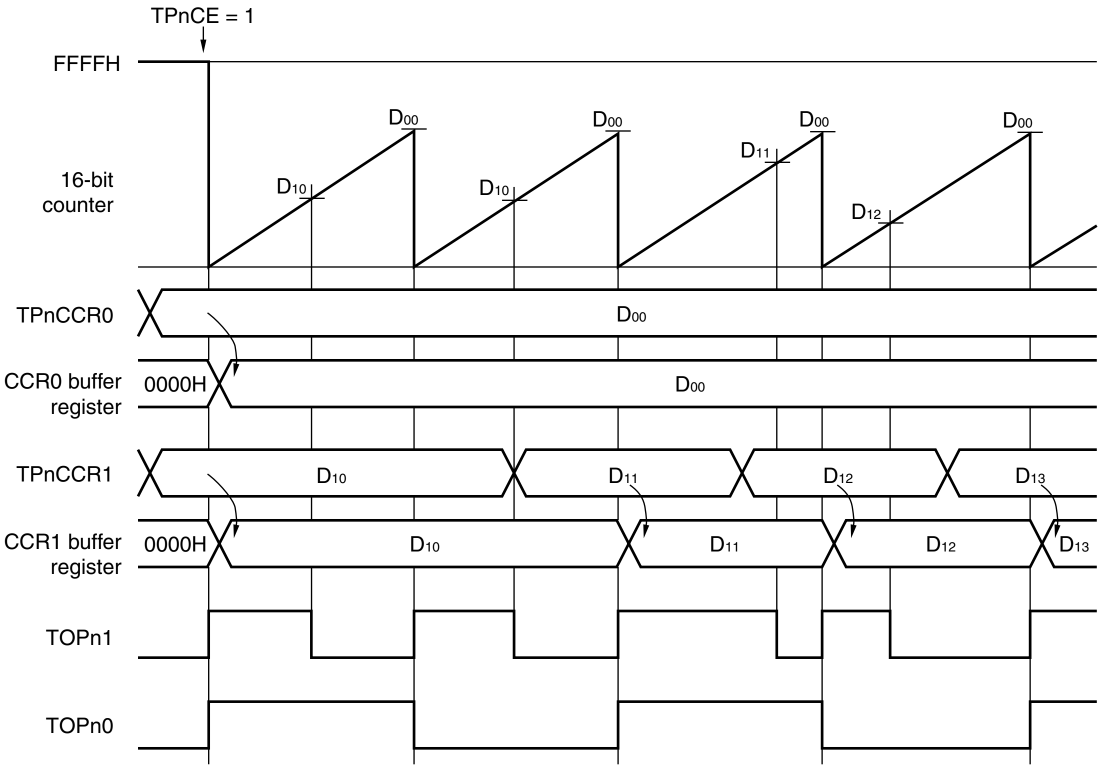

START
Initial setting
• Select clock.
(TPnCTL0: TPnCKS2 to TPnCKS0)
• Set PWM mode.
(TPnCTL1: TPnMD2 to TPnMD0 = 100)
• Set compare register.
(TPnCCR0, TPnCCR1)
Enable timer operation (TPnCE = 1)
→ Transfer value of TPnCCRm register to CCRm buffer register
16-bit counter matches TPnCCR1. TOPn1 outputs low level.
INTTPnCC1 occurs
16-bit counter matches TPnCCR0. Clear and start 16-bit counter. TOPn1 outputs high level.
INTTPnCC0 occurs
Rewrite TPnCCR0.
<1>
16-bit counter matches CCR1 buffer register. TOPn1 outputs low level.
<2>
Note
INTTPnCC1 occurs
Rewrite TPnCCR1.
<3>
Reload is enabled
• CCR0 buffer register matches 16-bit counter.
• Clear and start 16-bit counter.
• Value of TPnCCRm is reloaded to CCRm buffer register.
INTTPnCC0 occurs

Note: The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TPnCCR1, but make sure that step <3> comes after step <1>.

Remark: n = 0 to 3
m = 0, 1

Figure 7-26: Basic Operation Timing in PWM Mode (1/2)

(a) When rewriting TPnCCR1 value (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



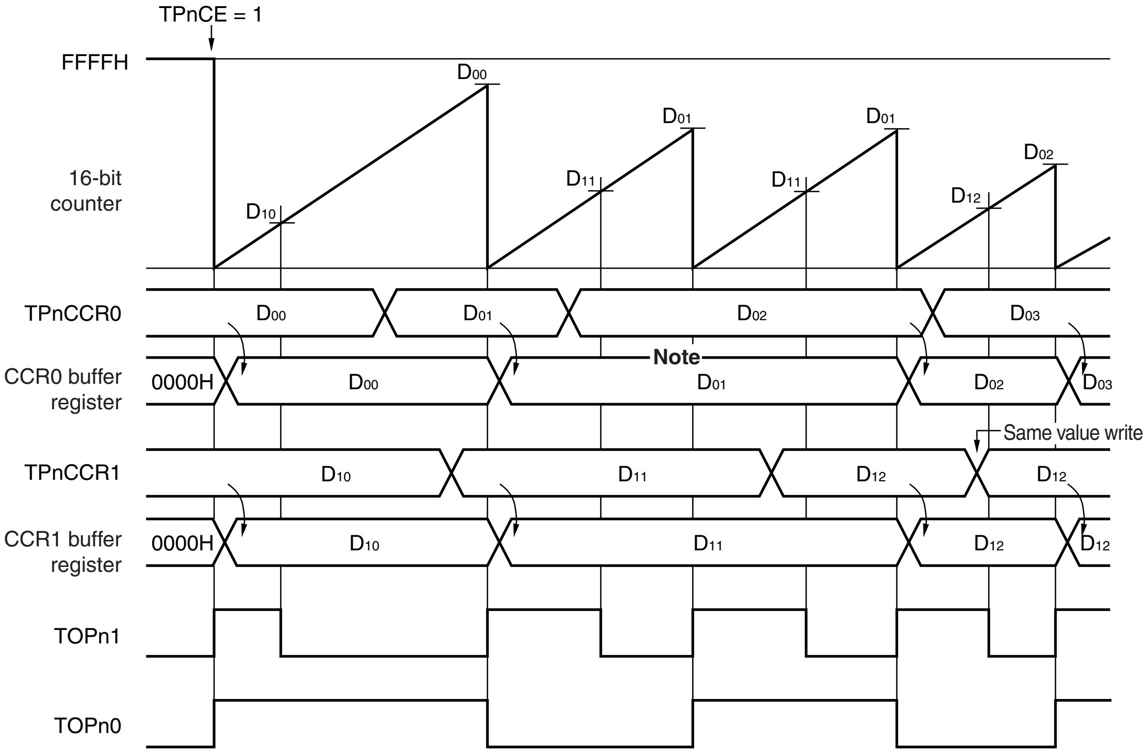
Remarks: 1. D00: Set value of TPnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Set value of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (Set value of TPnCCR1 register) / (Set value of TP0CCR0 register)
Cycle of TOPn1 output = (Set value of TPnCCR0 register) Þ (Count clock cycle)
Toggle width of TOPn0 output = (Set value of TPnCCR0 register + 1) Þ (Count clock period)

3. n = 0 to 3

Figure 7-26: Basic Operation Timing in PWM Mode (2/2)

(b) When TPnCCR0, TPnCCR1 values are rewritten (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Note: Reload is not performed because the TPnCCR1 register was not rewritten.

Remarks: 1. D00, D01, D02, D03: Setting values of TPnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Setting values of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (Set value of TPnCCR1 register) / (Set value of TP0CCR0 register)
Cycle of TOPn1 output = (Set value of TPnCCR0 register) x (Count clock cycle)
Toggle width of TOPn0 output = (Set value of TPnCCR0 register + 1) x (Count clock cycle)

3. n = 0 to 3

7.5.7 Free-running mode (TPnMD2 to TPnMD0 = 101)

In the free-running mode, both the interval function and the compare function can be realized by operating the 16-bit counter as a free-running counter and selecting capture/compare operation with the TPnCCS1 and TPnCCS0 bits.
The settings of the TPnCCS1 and TPnCCS0 bits of the TPnOPT0 register are valid only in the free-running mode.

TPnCCS1	Operation
0	Use TPnCCR1 register as compare register
1	Use TPnCCR1 register as capture register

TPnCCS0	Operation
0	Use TPnCCR0 register as compare register
1	Use TPnCCR0 register as capture register

- Using TPnCCR1 register as compare register

An interrupt is output upon a match between the 16-bit counter and the CCR1 buffer register in the free-running mode (interval function).
Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)
When timer output (TOPn1) has been enabled, TOPn1 performs toggle output upon a match between the 16-bit counter and the CCR1 buffer register.

- Using TPnCCR1 register as capture register

The value of the 16-bit counter is saved to the TPnCCR1 register upon TIPn1 pin edge detection.

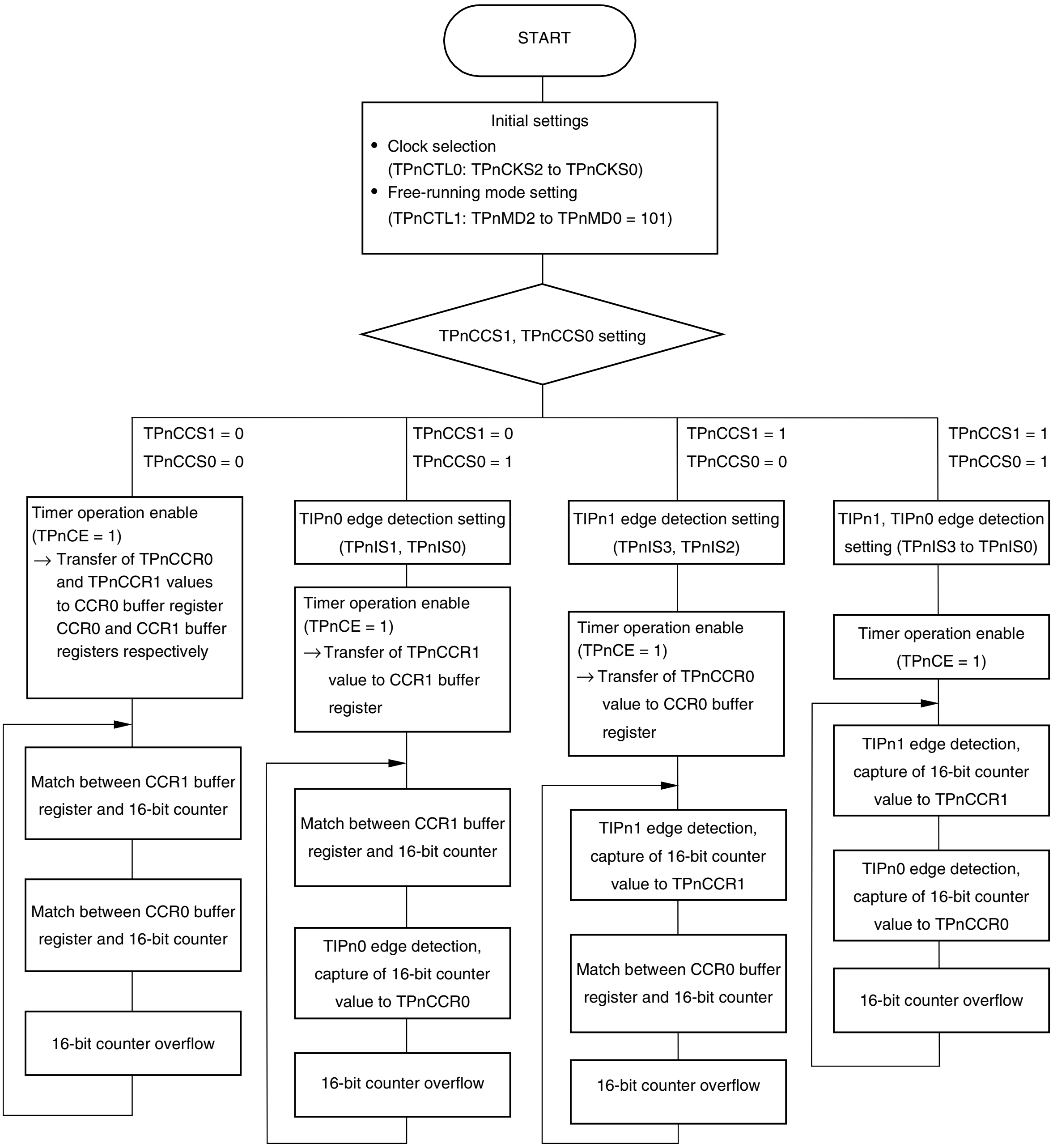
- Using TPnCCR0 register as compare register

An interrupt is output upon a match between the 16-bit counter and the CCR0 buffer register in the free-running mode (interval function).
Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)
When timer output (TOPn0) has been enabled, TOPn0 performs toggle output upon a match between the 16-bit counter and the CCR0 buffer register.

- Using TPnCCR0 register as capture register

The value of the 16-bit counter is saved to the TPnCCR0 register upon TIPn0 pin edge detection.

Figure 7-27: Flowchart of Basic Operation in Free-Running Mode

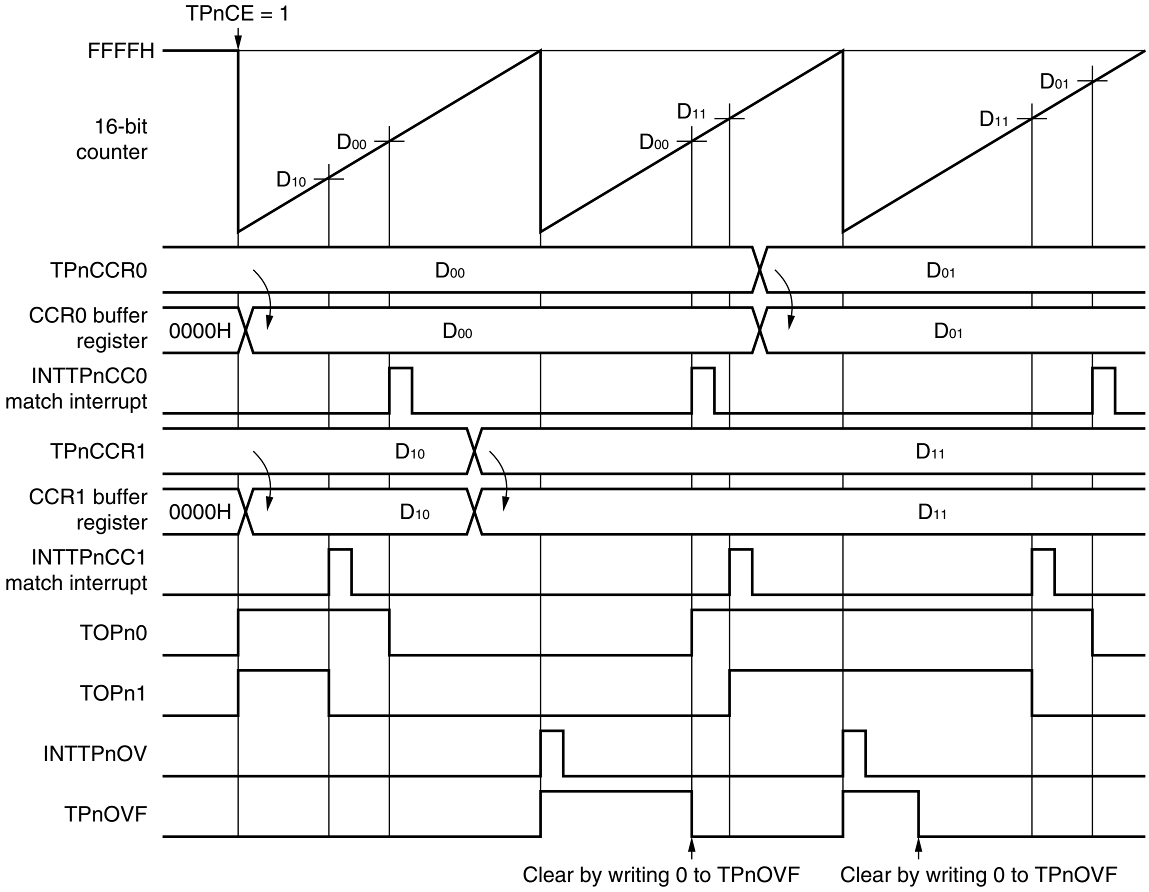

START
Initial settings
• Clock selection
(TPnCTL0: TPnCKS2 to TPnCKS0)
• Free-running mode setting
(TPnCTL1: TPnMD2 to TPnMD0 = 101)
TPnCCS1, TPnCCS0 setting
TPnCCS1 = 0
TPnCCS0 = 0
TPnCCS1 = 0
TPnCCS0 = 1
TPnCCS1 = 1
TPnCCS0 = 0
TPnCCS1 = 1
TPnCCS0 = 1
Timer operation enable (TPnCE = 1) → Transfer of TPnCCR0 and TPnCCR1 values to CCR0 buffer register CCR0 and CCR1 buffer registers respectively
Match between CCR1 buffer register and 16-bit counter
Match between CCR0 buffer register and 16-bit counter
16-bit counter overflow
TIPn0 edge detection setting (TPnIS1, TPnIS0)
Timer operation enable (TPnCE = 1) → Transfer of TPnCCR1 value to CCR1 buffer register
Match between CCR1 buffer register and 16-bit counter
TIPn0 edge detection, capture of 16-bit counter value to TPnCCR0
16-bit counter overflow
TIPn1 edge detection setting (TPnIS3, TPnIS2)
Timer operation enable (TPnCE = 1) → Transfer of TPnCCR0 value to CCR0 buffer register
TIPn1 edge detection, capture of 16-bit counter value to TPnCCR1
Match between CCR0 buffer register and 16-bit counter
16-bit counter overflow
TIPn1, TIPn0 edge detection setting (TPnIS3 to TPnIS0)
Timer operation enable (TPnCE = 1)
TIPn1 edge detection, capture of 16-bit counter value to TPnCCR1
TIPn0 edge detection, capture of 16-bit counter value to TPnCCR0
16-bit counter overflow

Remark: n = 0 to 3

(1) When TPnCCS1 = 0, and TPnCCS0 = 0 settings (interval function description, compare function)

When TPnCE = 1 is set, the 16-bit counter counts from 0000H to FFFFH and the free-running count-up operation continues until TPnCE = 0 is set. In this mode, when a value is written to the TPnCCR0 and TPnCCR1 registers, they are transferred to the CCR0 buffer register and the CCR1 buffer register (any time write mode). In this mode, no one-shot pulse is output even when an one-shot pulse trigger is input. Moreover, when TPnOEm = 1 is set, TOPnm performs toggle output upon a match between the 16-bit counter and the CCRm buffer register.

Figure 7-28: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 0) (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks: 1. D00, D01: Setting values of TPnCCR0 register (0000H to FFFFH)
D10, D11: Setting values of TPnCCR1 register (0000H to FFFFH)

2. Toggle width of TOPn0 output = (Set value of TPnCCR0 register) x (Count clock cycle)
Toggle width of TOPn1 output = (Set value of TPnCCR1 register) x (Count clock cycle)

3. TOPnm output goes high when counting is started.

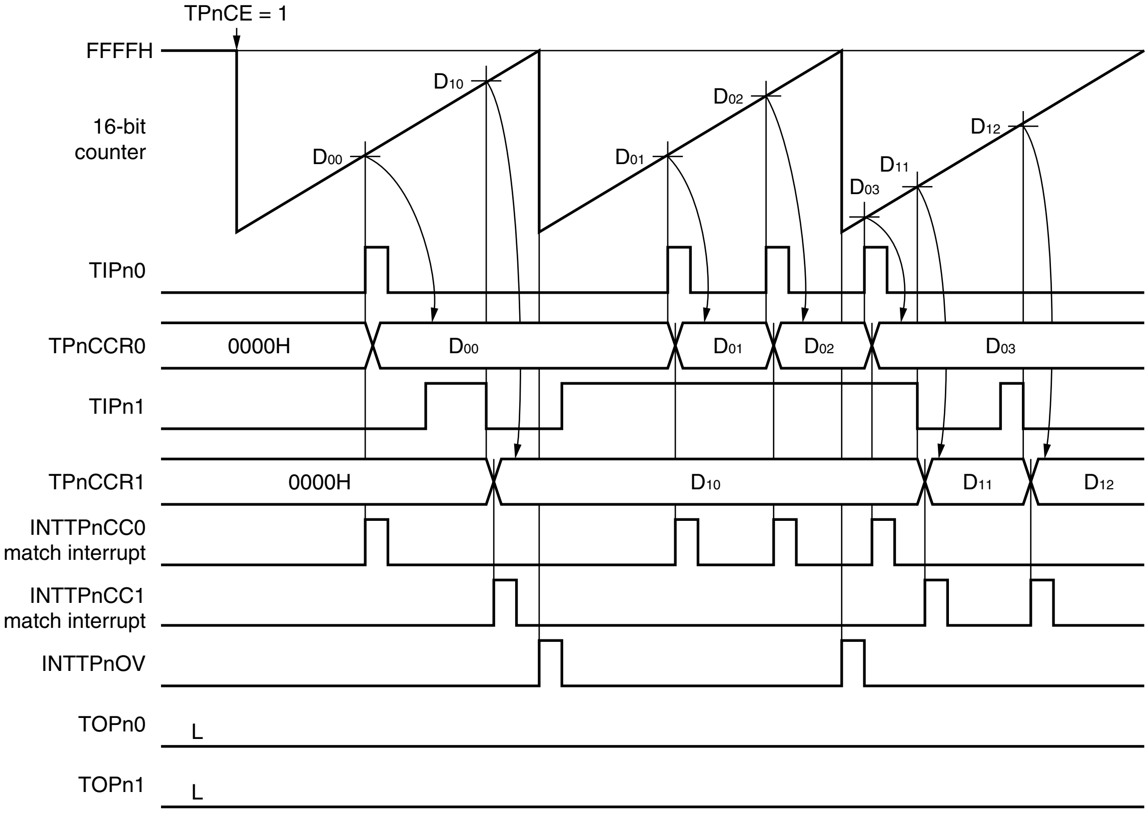
4. n = 0 to 3
m = 0, 1

(2) When TPnCCS1 = 1 and TPnCCS0 = 1 settings (capture function description)

When TPnCE = 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. During this time, values are captured by capture trigger operation and are written to the TPnCCR0 and TPnCCR1 registers.

Regarding capture close to the overflow (FFFFH), judgment is made using the overflow flag (TPnOVF). However, if overflow occurs twice (two or more free-running cycles), the capture trigger interval cannot be judged with the TPnOVF flag. In this case, the system should be revised.

Figure 7-29: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 1) (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks: 1. D00, D01, D02, D03: Values captured to TPnCCR0 register (0000H to FFFFH)
D10, D11, D12: Values captured to TPnCCR1 register (0000H to FFFFH)

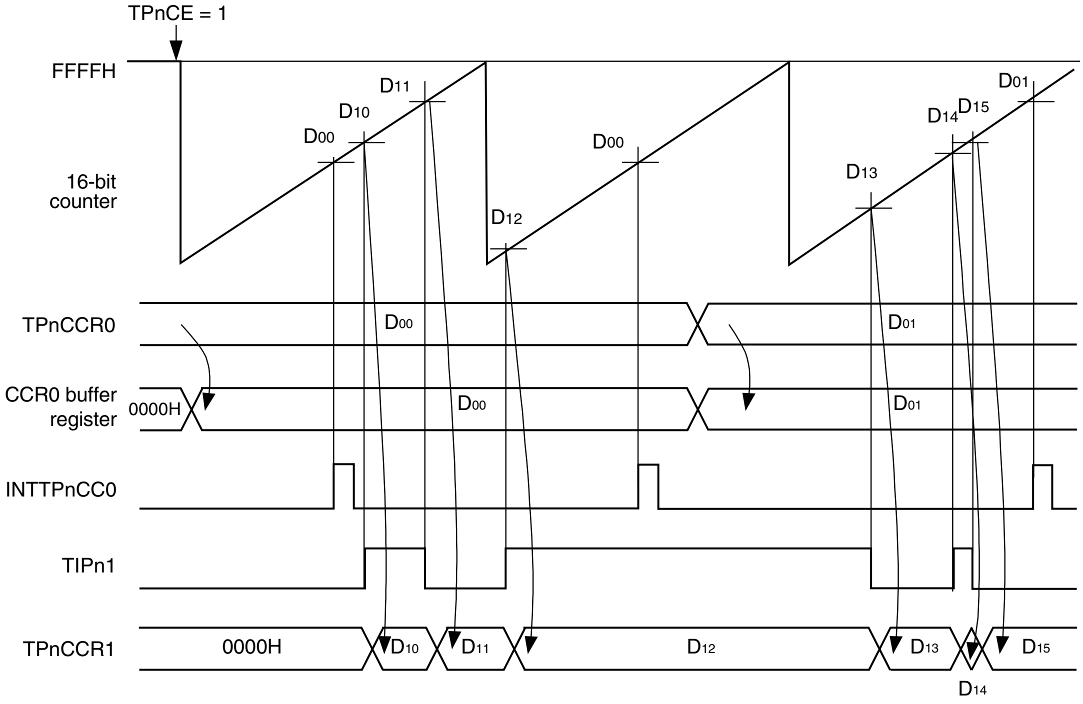
2. TIPn0: Set to rising edge detection (TPnIS1, TPnIS0 = 01)
TIPn1: Set to falling edge detection (TPnIS3, TPnIS2 = 10)

3. n = 0 to 3

(3) When TPnCCS1 = 1 and TPnCCS0 = 0

When TPnCE = 1 is set, the counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. The TPnCCR0 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value transferred to the CCR0 buffer register from the TPnCCR0 register as an interval function. Even if TPnOE1 = 1 to realize the output function, TPnCCR1 register cannot control TOPn1 because it is used as capture register.

Figure 7-30: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 0) (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks: 1. D00, D01: Setting compare values of TPnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13, D14, D15: Values captured to TPnCCR1 register (0000H to FFFFH)

2. TIPn1: Set to detection of both rising and falling edges (TPnIS3, TPnIS2 = 11)

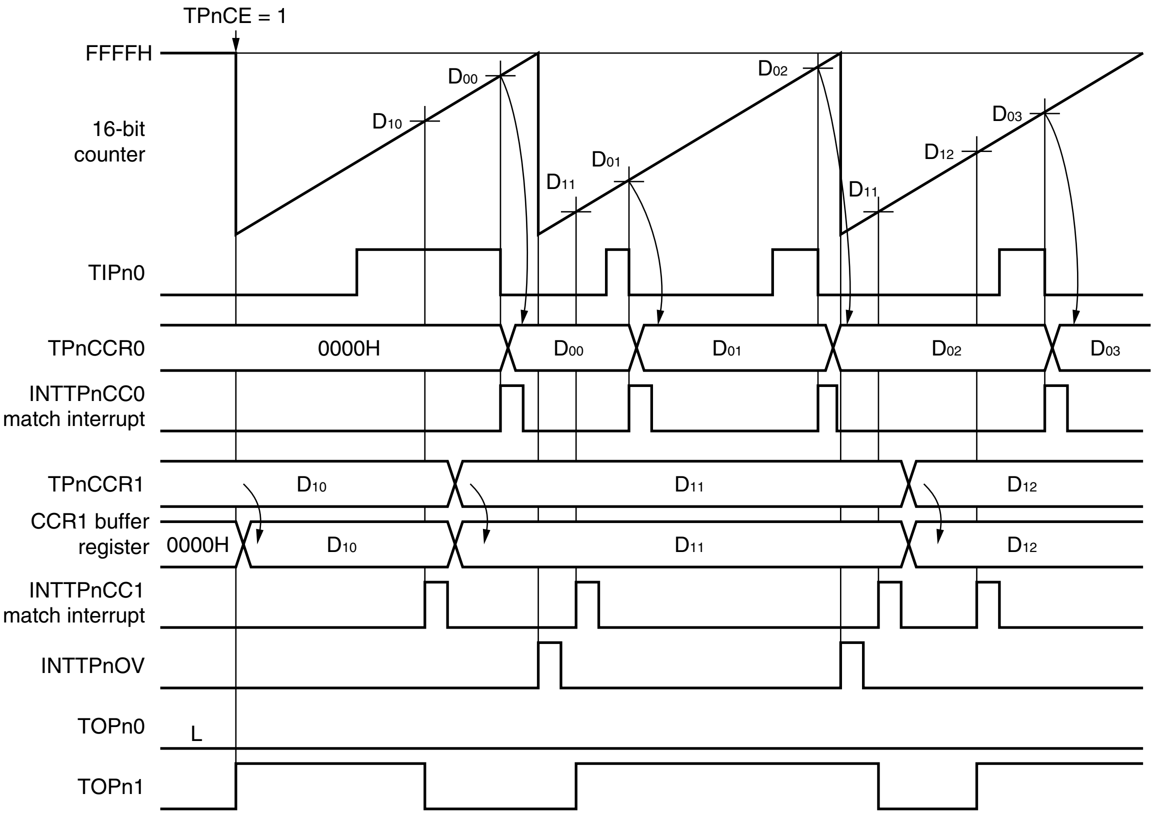
3. n = 0 to 3

(4) When TPnCCS1 = 0 and TPnCS0 = 1

When TPnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. The TPnCCR1 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value of the TPnCCR1 register as an interval function. When TPnOE1 = 1 is set, TOPn1 performs toggle output upon mach between the value of the 16-bit counter and the setting value of the TPnCCR1 register.

Even if TPnOE0 = 1 to realize the output function, TPnCCR0 register cannot control TOPn0 because it is used as capture register.

Figure 7-31: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 1) (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks: 1. D00, D01, D02, D03: Values captured to TPnCCR0 register (0000H to FFFFH)
D10, D11, D12: Setting compare value of TPnCCR1 register (0000H to FFFFH)

2. TIPn0: Set to falling edge detection (TPnIS1, TPnIS0 = 10)

3. n = 0 to 3

(5) Overflow flag

When the counter overflows from FFFFH to 0000H in the free-running mode, the overflow flag (TPnOVF) is set to 1 and an overflow interrupt (INTTPnOV) is output.

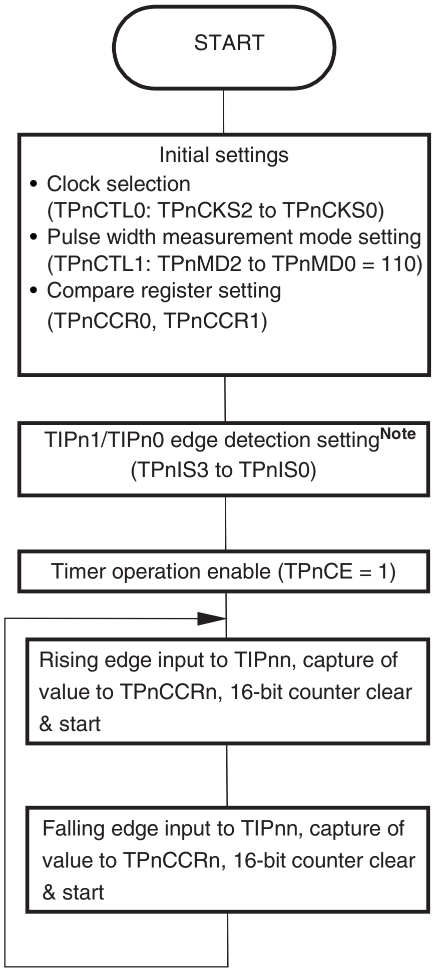
The overflow flag is cleared by the CPU when writing 0 to it.

7.5.8 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110)

In the pulse width measurement mode, free-running count is performed, and upon detection of both the rising and falling edges of TIPn0, the 16-bit counter value is saved to capture register 0 (TPnCCR0) and the 16-bit counter is cleared to 0000H. The external input pulse width can be measured as a result.
However, when measuring a large pulse width that exceeds 16-bit counter overflow, perform judgment with the overflow flag. Since measurement of pulses for which overflow occurs twice or more is not possible, adjust the operating frequency of the 16-bit counter. The value of the 16-bit counter is also saved to capture register 1 (TPnCCR1) and the 16-bit counter cleared upon detection of the TIPn1 edge.

Caution: In the pulse width measurement mode, select the internal clock (TPnEEE of TPnCTL1 register = 0).

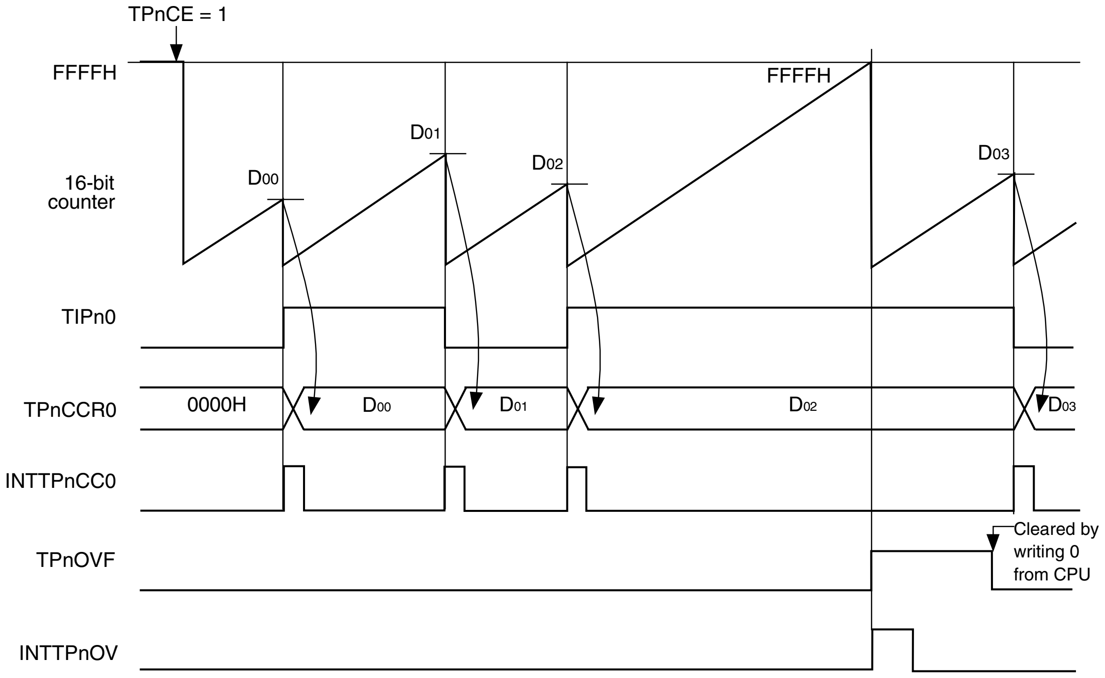
Figure 7-32: Flowchart of Basic Operation in Pulse Width Measurement Mode


START
Initial settings
• Clock selection (TPnCTL0: TPnCKS2 to TPnCKS0)
• Pulse width measurement mode setting (TPnCTL1: TPnMD2 to TPnMD0 = 110)
• Compare register setting (TPnCCR0, TPnCCR1)
TIPn1/TIPn0 edge detection setting Note (TPnIS3 to TPnIS0)
Timer operation enable (TPnCE = 1)
Rising edge input to TIPnn, capture of value to TPnCCRn, 16-bit counter clear & start
Falling edge input to TIPnn, capture of value to TPnCCRn, 16-bit counter clear & start

Note: An external pulse can be input from either TIPn0 or TIPn1. Only one of them can be used. Specify that both the rising and falling edges are detected. Specify that the input edge of an external pulse input that is not used is not detected.

Remark: n = 0 to 3
m = 0, 1

Figure 7-33: Basic Operation Timing in Pulse Width Measurement Mode (TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 0)



Remarks: 1. D00, D01, D02, D03: Values captured to TPnCCR0 register (0000H to FFFFH)

2. TIPn0: both rising and falling edges are detected (TPnIS1, TPnIS0 = 11)

3. n = 0 to 3

7.6 Timer Synchronization Operation Function

Timer P and timer Q have a timer synchronized operation function (tuned operation mode).
The timers that can be synchronized are listed in Table 7-3 (√□: Settable, ×: Not settable).

Table 7-3: Tuned Operation Mode of Timer

Master Timer	Slave Timer		V850E/RS1
TMP0	TMP1	-	√
TMp2	TMP3	TMQ0	√

Cautions: 1. The tuned operation mode is enabled or disabled by the TPnSYE bit of the TPnCTL1 register and TQnSYE bit of the TQnCTL1 register. For TMP2, either or both TMP3 and TMQ0 can be specified as slaves.

2. Set the tuned operation mode using the following procedure.

- **Set the TPnSYE bit of the TPnCTL1 register and the TQnSYE bit of the TQnCTL1 register of the slave timer to enable the tuned operation. Set the TPnMD2 to TPnMD0 bits of the TPnCTL1 register and TPnMD2 to TPnMD0 bits of the TQnCTL1 register of the slave timer to the free-running mode.**
- **Set the timer mode by using the TPnMD2 to TPnMD0 bits of the TPnCTL1 register and the TPnMD2 to TPnMD0 bits of the TQnCTL1 register. At this time, do not set the TPnSYE bit of the TPnCTL1 register and the TQnSYE bit of the TQnCTL1 register of the master timer.**
- **Set the compare register value of the master and slave timers.**
- **Set the TPnCE bit of the TPnCTL0 register and the TQnCE bit of the TQnCTL0 register of the slave timer to enable operation on the internal operating clock.**
- **Set the TPnCE bit of the TPnCTL0 register and the TQnCE bit of the TQnCTL0 register of the master timer to enable operation on the internal operating clock.**

Tables 7-4 and 7-5 show the timer modes that can be used in the tuned operation mode
(√□: Settable, ×: Not settable).

Table 7-4: Timer Modes Usable in Tuned Operation Mode

Master Timer	Free-Running Mode	PWM Mode	Triangular Wave PWM Mode
TMP0	√	√	×
TMP2	√	√	×
TMQ1	√	√	√

Table 7-5: Timer Output Functions

Tuned Channel	Timer	Pin	Free-Running Mode		PWM Mode		Triangular Wave PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
Ch0	TMP0 (master)	TOP00	PPG	←	Toggle	←	N/A	←
		TOP01	PPG	←	PWM	←	N/A	←
	TMP1 (slave)	TOP10	PGP	←	Toggle	PWM	N/A	←
		TOP11	PPG	←	PWM	←	N/A	←
Ch1	TMP2 (master)	TOP20	PPG	←	Toggle	←	N/A	←
		TOP21	PPG	←	PWM	←	N/A	←
	TMP3 (slave)	TOP30	PPG	←	Toggle	PWM	N/A	←
		TOP31	PPG	←	PWM	←	N/A	←
Ch1	TMQ0 (slave)	TOQ00	PPG	←	Toggle	PWM	Toggle	N/A
		TOQ01 to TOQ03	PPG	←	PWM	←	Triangular wave PWM	N/A
Ch2	TMQ1 (master)	TOQ10	PPG	←	Toggle	←	Toggle	←
		TOQ11 to TOQ13	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 TOPn0 and TOQm0 (n = 0 to 3, m = 0 to 1)

Figure 7-34: Tuned Operation Image (TMP2, TMP3, TMQ0)

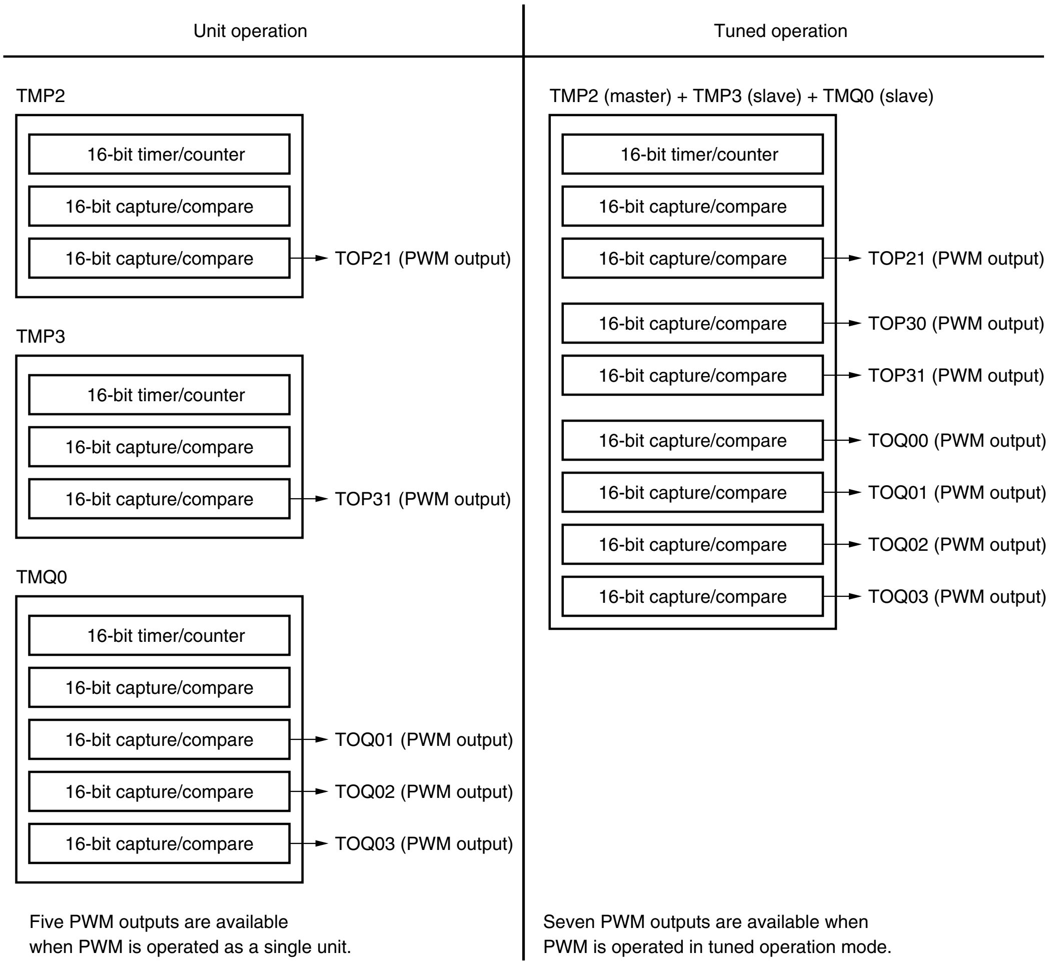
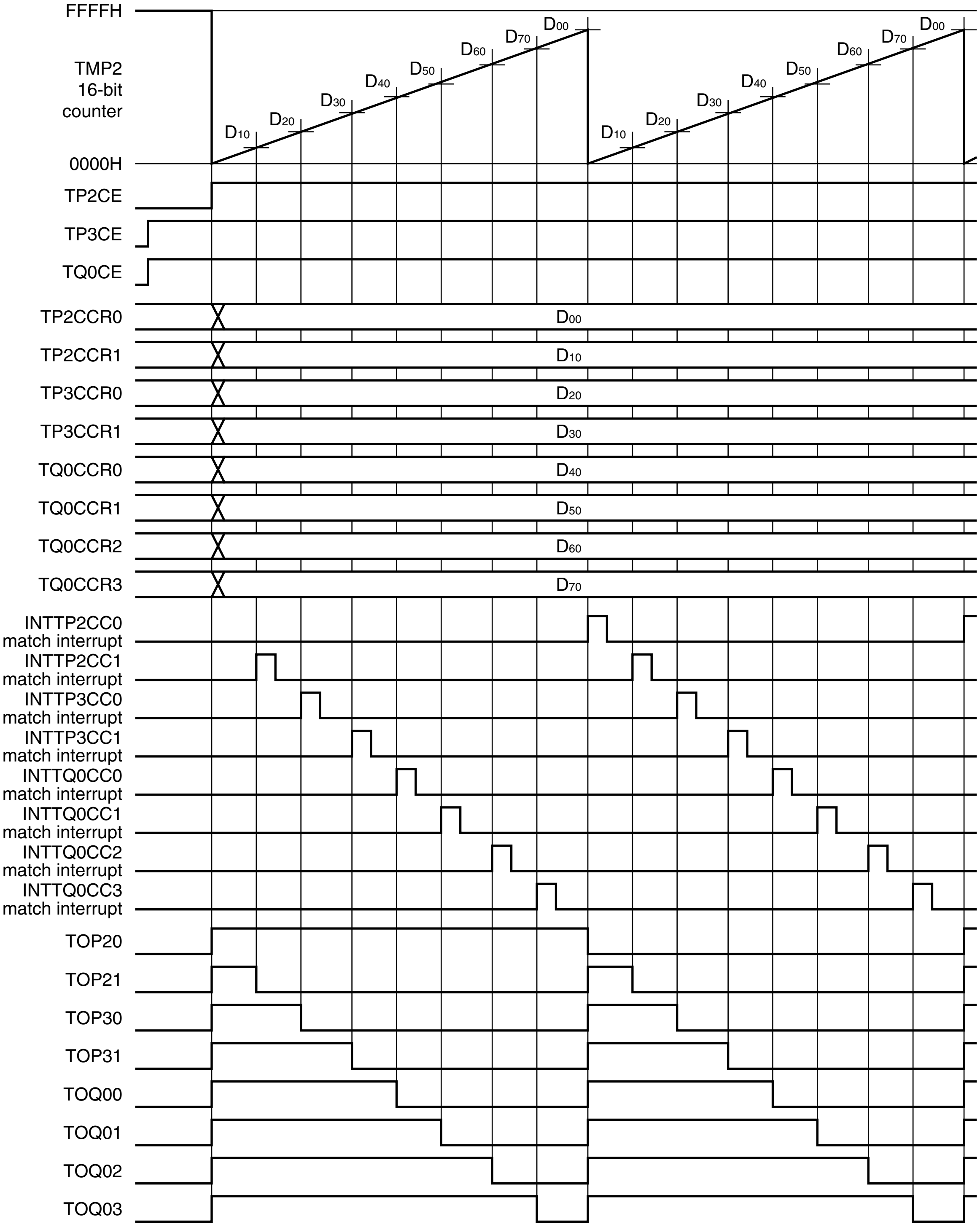


Figure 7-35: Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)


FFFFH
TMP2 16-bit counter
0000H
D10
D20
D30
D40
D50
D60
D70
D00
TP2CE
TP3CE
TQ0CE
TP2CCR0
TP2CCR1
TP3CCR0
TP3CCR1
TQ0CCR0
TQ0CCR1
TQ0CCR2
TQ0CCR3
INTTP2CC0 match interrupt
INTTP2CC1 match interrupt
INTTP3CC0 match interrupt
INTTP3CC1 match interrupt
INTTQ0CC0 match interrupt
INTTQ0CC1 match interrupt
INTTQ0CC2 match interrupt
INTTQ0CC3 match interrupt
TOP20
TOP21
TOP30
TOP31
TOQ00
TOQ01
TOQ02
TOQ03

Chapter 8 16-Bit Timer/Event Counter Q

The V850E/RS1 includes two channels 16 bit timer/event counter Q (TMQ0, TMQ1)

8.1 Features

Timer Q (TMQ) is a 16-bit timer/event counter provided with general-purpose functions.
TMQ can perform the following operations.

- 16-bit-accuracy PWM output
- Interval timer
- External event counter (operation not possible when clock is stopped)
- Timer synchronised operation function
- One-shot pulse output
- Pulse width measurement function
- Triangular wave PWM output

8.2 Function Outline

- Capture trigger input signal × 4
- External trigger input signal × 1
- Clock select × 8
- External event input × 1
- Readable counter × 1
- Capture/compare reload register × 4
- Capture/compare match interrupt × 4
- Timer output (TOQn0 to TOQn4) × 4

8.3 Configuration

TMQ includes the following hardware.

Table 8-1: TMQ Configuration

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMQn capture/compare registers 0 to 3 (TQnCCR0 to TQnCCR3) TMQn counter read buffer register (TQnCNT) CCR0 to CCR3 buffers registers
Timer input	4 (TIQn0^{Note} to TIQn3)
Timer output	3 (TOQn1 toTOQn3)
Control registers	TMQn control registers 0, 1 (TQnCTL0, TQnCTL1) TMQn dedicated I/O control registers 0 to 2 (TQnIOC0 to TQnIOC2) TMQn option registers 0 (TQnOPT0)

Note: TIQn0 functions alternately as a capture trigger input signal, external trigger input signal, and external event input signal.

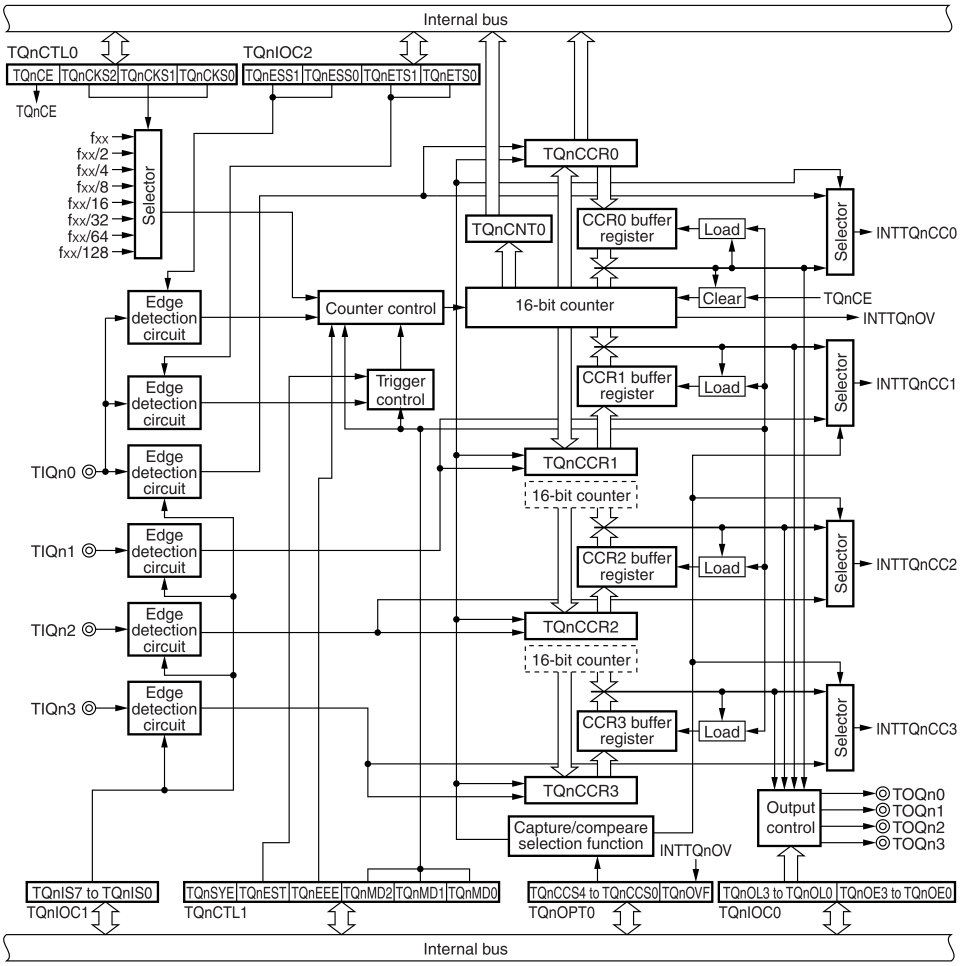
Remark: n = 0 to 1; the notation of n is maintained throughout Section 8 unless otherwise noted.

Timer Q (TMQ) pins are alternate function of port pins. For how to set the alternate function, refer to the description of the registers in **Chapter 4 ”Port Functions” on page 105**.

Table 8-2: TMQ Pin List

Pin Name	I/O	Function	Alternate Function
TIQ00	Input	External event/clock input (TMQ00)	P53/ TOQ00
TIQ01		External event/clock input (TMQ01)	P50/TOQ01
TIQ02		External event/clock input (TMQ02)	P51/TOQ02
TIQ03		External event/clock input (TMQ03)	P52/TOQ03
TIQ10		External event/clock input (TMQ10)	P93/CS302/TOQ10
TIQ11		External event/clock input (TMQ11)	P90/$\overline{\text{SCK30}}$/TOQ11
TIQ12		External event/clock input (TMQ12)	P91/CS300/TOQ12
TIQ13		External event/clock input (TMQ03)	P92/CS301/TOQ13
TOQ00	Output	Timer output (TMQ00)	P53/TIQ00
TOQ01		Timer output (TMQ01)	P50/TIQ01
TOQ02		Timer output (TMQ02)	P51/TIQ02
TOQ03		Timer output (TMQ03)	P52/TIQ03
TOQ10		Timer output (TMQ10)	P93/CS302/TIQ10
TOQ11		Timer output (TMQ11)	P90/$\overline{\text{SCK30}}$/TIQ11
TOQ12		Timer output (TMQ12)	P91/CS300/TIQ12
TOQ13		Timer output (TMQ13)	P92/CS301/TIQ13

Figure 8-1: Block Diagram of Timer Q



(1) Capture/compare register 0 (TQnCCR0)

The TQnCCR0 register is a 16-bit register that has a capture function and a compare function. Whether this register is used as a capture register or a compare register can be specified by using the TQnCCS0 bit of the TQnOPT0 register, but only in the free-running mode.
In the pulse width measurement mode, this register can be used only as a capture register (it cannot be used as a compare register).
In all the modes other than the free-running mode and pulse width measurement mode, this register functions as a compare register.
In the default status, the TQnCCR0 register functions as a compare register.
This register can be read or written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 8-2: Capture/Compare Register 0 (TQnCCR0) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ0CCR0																	FFFF F546H	R/W	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ1CCR0																	FFFF F616H	R/W	0000H

- When used as a compare register

TQnCCR0 can be rewritten when TQnCE = 1 as below mentioned:

TMQ Operation Mode	Method of Writing TQnCCR0 Register
PWM output mode, external trigger pulse output mode, or triangular wave PWM mode	Reload
Free-running mode, external event count mode, one-shot pulse mode or interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because used only as capture register

- When used as capture register

The count value is stored in TQnCCR0 upon capture trigger (TIQn0) input edge detection.

(2) Capture/compare register 1 (TQnCCR1)

The TQnCCR1 register is a 16-bit register that has a capture function and a compare function.
Whether this register is used as a capture register or a compare register can be specified by using the TQnCCS1 bit of the TQnOPT0 register, but only in the free-running mode.
In the pulse width measurement mode, this register can be used only as a capture register (it cannot be used as a compare register).
In the all the modes other than the free-running mode and pulse width measurement mode, this register functions as a compare register.
In the default status, the TQnCCR1 register functions as a reload register.
This register can be read or written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Caution: In the one-shot pulse mode, it is prohibited to set the TQnCCR1 register to 0000H.

Figure 8-3: Capture/Compare Register 1 (TQnCCR1) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ0CCR1																	FFFF F548H	R/W	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ1CCR1																	FFFF F618H	R/W	0000H

- When used as a compare register

TQnCCR1 can be rewritten when TQnCE = 1, as below mentioned:

TMQ Operation Mode	Method of Writing TQnCCR1 Register
PWM output mode, external trigger pulse output mode, or triangular wave PWM mode	Reload
Free-running mode, external event count mode, one-shot pulse mode or interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because used only as capture register

- When used as a capture register

The count value is stored in TQnCCR1 upon capture trigger (TIQn1) input edge detection.

(3) Capture/compare register 2 (TQnCCR2)

The TQnCCR2 register is a 16-bit register that has a capture function and a compare function. Whether this register is used as a capture register or a compare register can be specified by using the TQnCCS2 bit of the TQnOPT0 register, but only in the free-running mode.
In the pulse width measurement mode, this register can be used only as a capture register (it cannot be used as a compare register).
In all the modes other than the free-running mode and pulse width measurement mode, this register functions as a compare register.
In the default status, the TQnCCR2 register functions as a compare register.
This register can be read or written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 8-4: Capture/Compare Register 2 (TQnCCR2) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ0CCR2																	FFFF F54AH	R/W	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ1CCR2																	FFFF F61AH	R/W	0000H

- When used as a compare register

TQnCCR2 can be rewritten when TQnCE = 1, as below mentioned:

TMQ Operation Mode	Method of Writing TQnCCR2 Register
PWM output mode, external trigger pulse output mode, or triangular wave PWM mode	Reload
Free-running mode, external event count mode, one-shot pulse mode or interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because used only as capture register

- When used as capture register

The count value is stored in TQnCCR2 upon capture trigger (TIQn2) input edge detection.

(4) Capture/compare register 3 (TQnCCR3)

The TQnCCR3 register is a 16-bit register that has a capture function and a compare function. Whether this register is used as a capture register or a compare register can be specified by using the TQnCCS3 bit of the TQnOPT0 register, but only in the free-running mode.

In the pulse width measurement mode, this register can be used only as a capture register (it cannot be used as a compare register).

In all the modes other than the free-running mode and pulse width measurement mode, this register functions as a compare register.

In the default status, the TQnCCR3 register functions as a compare register.

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

$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 8-5: Capture/Compare Register 3 (TQnCCR3) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ0CCR3																	FFFF F54CH	R/W	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ1CCR3																	FFFF F61CH	R/W	0000H

- When used as a compare register

TQnCCR3 can be rewritten when TQnCE = 1, as below mentioned:

TMQ Operation Mode	Method of Writing TQnCCR2 Register
PWM output mode, external trigger pulse output mode, or triangular wave PWM mode	Reload
Free-running mode, external event count mode, one-shot pulse mode or interval timer mode	Any time write
Pulse width measurement mode	Cannot be used because used only as capture register

- When used as capture register

The count value is stored in TQnCCR3 upon capture trigger (TIQn3) input edge detection.

(5) Timer read buffer register (TQnCNT)

The TQnCNT register is a timer read buffer register that can read 16-bit counter values.
This register is read-only using a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to FFFFH.
When TQnCE bit of TQnCTL0 register = 0, the hardware status is FFFFH, but a value of 0000H is returned when this register is read.
The value of this register is read when TQnCE bit = 1.

Figure 8-6: Timer Read Buffer Register (TQnCNT) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ0CNT																	FFFF F54EH	R/W	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TQ1CNT																	FFFF F61EH	R/W	0000H

8.4 Control Registers

(1) Timer Q0 control register 0 (TQnCTL0)

Timer Q0 control register 0 is an 8-bit register that controls the operation of timer Q.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.
The same value can always be written to the TQnCTL0 register by software.

Figure 8-7: Timer Qn Control Register 0 (TQnCTL0) Format

Symbol	<7>	6	5	4	3	2	1	0	Address	R/W	After reset
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0	FFFFF540H	R/W	00H

Symbol	<7>	6	5	4	3	2	1	0	Address	R/W	After reset
TQ1CTL0	TQ1CE	0	0	0	0	TQ1CKS2	TQ1CKS1	TQ1CKS0	FFFFF610H	R/W	00H

TQnCE	Timer Qn operation control
0	Disable internal operating clock operation (asynchronously reset TMQn).
1	Enable internal operating clock operation.
The TQnCE bit controls the internal operating clock and asynchronously resets TMQn. When this bit is cleared to 0, the internal operating clock of TMQn is stopped (fixed to the low level), and TMQn is asynchronously reset. When the TQnCE bit is set to 1, the internal operating clock is enabled within 2 input clocks, and TMQn counts up.	

TQnCKS2	TQnCKS1	TQnCKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

Caution: Set bits TQnCKS2 to TQnCKS0 when TQnCE = 0.
When the value of the TQnCE bit is changed from 0 to 1, bits TQnCKS2 to TQnCKS0 can be set simultaneously.

Remark: n = 0, 1

(2) Timer Q control register 1 (TQnCTL1)

The TQnCTL1 register is an 8-bit register that controls the operation of timer Q.
This register can be read or written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-8: Timer Q Control Register 1 (TQnCTL1) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ0CTL1	TQ0SYE	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0	FFFFF541H	R/W	00H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ1CTL1	TQ1SYE	TQ1EST	TQ1EEE	0	0	TQ1MD2	TQ1MD1	TQ1MD0	FFFFF611H	R/W	00H

<table>
<tr><th>TQnSYE</th><th>Tuned operation mode enable control</th></tr>
<tr><td>0</td><td>Independent operation mode (asynchronous operation mode)</td></tr>
<tr><td>1</td><td>Tuned operation mode (specification of slave operation)
In this mode, timer Q can operate in synchronization with a master timer.
<table>
<tr><th>Master timer</th><th colspan="2">Slave timer</th></tr>
<tr><td>TMQ0</td><td>TMQ1</td><td>-</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
For the tuned operation mode, refer to <b>8.6 Timer Synchronized Operation Function</b>.
<b>Caution: Be sure to clear the TQ0SYE bit to 0 (master timer) and TQ1SYE bit to 1 (slave timer)</b></td></tr>
</table>

Caution: In the synchronous operation mode, the master macro can be set only in the PWM mode, external trigger pulse output mode, pulse output mode, and free-running mode.
The slave macros can be set only in the free-running mode.
Setting the external event count mode, one-shot pulse mode, and pulse width measurement mode is prohibited.

<table>
<tr><th>TQnEST</th><th>Software trigger control</th></tr>
<tr><td>0</td><td>No operation</td></tr>
<tr><td rowspan="2">1</td><td>In one-shot pulse mode: One-shot pulse software trigger</td></tr>
<tr><td>In external trigger pulse output mode: Pulse output software trigger</td></tr>
<tr><td colspan="2">The TQnEST bit functions as a software trigger in the one-shot pulse mode or external trigger pulse output mode (this bit is invalid in any other mode). By setting TQnEST to 1 when TQnCE = 1, a software trigger is issued. Therefore, be sure to set TQnEST to 1 when TQnCE = 1.
The TIQn0 pin is used for an external trigger. The read value of the TQnEST bit is always 0.</td></tr>
</table>

Figure 8-8: Timer Q Control Register 1 (TQnCTL1) Format (2/2)

TQnEEE	Count clock selection
0	Use the internal clock (clock selected with bits TQnCKS2 to TQnCKS0)
1	Use the external clock from the TIQn0 input pin
The valid edge when TQnEEE = 1 (use the external clock from TIQn0 pin) is specified with bits TQnEES1 and TQnEES0.	

TQnMD2	TQnMD1	TQnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event counter mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse mode
1	0	0	PWM mode
1	0	1	Free-running mode
1	1	0	Pulse width measurement mode
1	1	1	Triangular wave PWM mode

Caution: Set bits TQnEEE and TQnMD2 to TQnMD0 when TQnCE = 0. (The same value can be written when TQnCE = 1.) The operation is not guaranteed when rewriting is performed when TQnCE = 1. If rewriting was mistakenly performed, set TQnCE = 0 and then set the bits again.

Remark: n = 0, 1

(3) Timer Q dedicated I/O control register 0 (TQnIOC0)

The TQnIOC0 register is an 8-bit register that controls the timer output.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-9: Timer Q Dedicated I/O Control Register 0 (TQnIOC0) Format

Symbol	7	<6>	5	<4>	3	<2>	1	0	Address	R/W	After reset
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OL1	FFFFF542H	R/W	00H

Symbol	7	<6>	5	<4>	3	<2>	1	0	Address	R/W	After reset
TQ1IOC0	TQ1OL3	TQ1OE3	TQ1OL2	TQ1OE2	TQ1OL1	TQ1OE1	TQ1OL0	TQ1OE0	FFFFF612H	R/W	00H

TQnOLm	Timer output level setting
0	Normal output
1	Inverted output

TQnOEm	Timer output setting
0	Disable timer output (TOQnm pin outputs low level when TQnOLm = 0, and high level when TQnOLm = 1).
1	Enable timer output (TOQnm pin outputs pulses).

Cautions: 1. Rewrite bits TQnOLm and TQnOEm when TQnCE = 0. (The same value can be written when TQnCE = 1.) If rewriting was mistakenly performed, set TQnCE = 0 and then set the bits again.

2. To enable the timer output, be sure to set the corresponding alternate-function pins TQnIS7 to TQnIS0 of the TQnIOC1 register to "Detect no edge" and invalidate the capture operation. Then set the corresponding alternate-function port to output mode.

Remark: n = 0, 1
m = 0, 1, 2, 3

(4) Timer Q dedicated I/O control register 1 (TQnIOC1)

The TQnIOC1 register is an 8-bit register that controls the valid edge of the external input signals (TIQn0 to TIQn3).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-10: Timer Q Dedicated I/O Control Register 1 (TQnIOC1) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0	FFFFF593H	R/W	00H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ1IOC1	TQ1IS7	TQ1IS6	TQ1IS5	TQ1IS4	TQ1IS3	TQ1IS2	TQ1IS1	TQ1IS0	FFFFF613H	R/W	00H

TQnIS7	TQnIS6	Capture input (TIQn3) valid edge setting
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detection of falling edge
1	1	Detection of both edges

TQnIS5	TQnIS4	Capture input (TIQn2) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TQnIS3	TQnIS2	Capture input (TIQn1) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TQnIS1	TQnIS0	Capture input (TIQn0) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions: 1. Rewrite bits TQnIS3 to TQnIS0 when TQnCE = 0. (The same value can be written when TQnCE = 1.) If rewriting was mistakenly performed, set TQnCE = 0 and then set the bits again.

2. The TQnIS7 to TQnIS0 bits are valid only in the free-running mode and pulse width measurement mode. A capture operation is not performed in any other mode.

Remark: n = 0, 1

(5) Timer Q dedicated I/O control register 2 (TQnIOC2)

The TQnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIQn0) and external trigger input signal (TIQn0).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-11: Timer Q Dedicated I/O Control Register 2 (TQnIOC2) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0	FFFFF544H	R/W	00H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TQ1IOC2	0	0	0	0	TQ1EES1	TQ1EES0	TQ1ETS1	TQ1ETS0	FFFFF614H	R/W	00H

TQnEES1	TQnEES0	Setting of valid edge of external event count input (TIQn0)
0	0	Detect no edge (external event count is invalid).
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TQnETS1	TQnETS0	Setting of valid edge of external trigger input (TIQn0)
0	0	Detect no edge (external trigger is invalid).
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions: 1. Rewrite bits TQnEES1, TQnEES0, TQnEST1, and TQnEST0 when TQnCE = 0. (The same value can be written when TQnCE = 1.) If rewriting was mistakenly performed, set TQnCE = 0 and then set the bits again.

2. The TQnEES1 and TQnEES0 bits are valid when TQnEEE = 1 or when the external event count mode is set (TQnMD2 to TQnMD0 of TIQnCTL1 register = 001).

Remark: n = 0, 1

(6) Timer Q option register 0 (TQnOPT0)

The TQnOPT0 register is an 8-bit register that selects a capture or compare operation, and detects an overflow.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-12: Timer Q Option Register 0 (TQnOPT0) Format

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF	FFFFF545H	R/W	00H

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
TQ1OPT0	TQ1CCS3	TQ1CCS2	TQ1CCS1	TQ1CCS0	0	0	0	TQ1OVF	FFFFF615H	R/W	00H

TQnCCSm	Selection of capture or compare operation of TQnCCRm register
0	Compare register
1	Capture register
The TQnCCSm bit setting is valid only in the free-running mode.	

TQnOVF	Detection of Timer Q overflow
Set (1)	Overflow occurrence
Reset (0)	0 written to TQnOVF bit or TQnCE = 0
• The TQnOVF bit is reset when the 16-bit counter value overflows from FFFFH to 0000H in the free-running mode or the pulse width measurement mode. • As soon as the TQnOVF bit has been set to 1, an interrupt request signal (INTTQnOV) is generated. The INTTQnOV signal is not generated in any mode other than the free-running mode and pulse width measurement mode. • The TQnOVF bit is not cleared even when the TQnOVF bit and the TQnOPT0 register are read when TQnOVF = 1. • The TQnOVF bit can be both read and written, but 1 cannot be written to the TQnOVF bit from the CPU. Writing 1 has no influence on the operation of timer Q.	

Cautions: 1. Rewrite bits TQnCCS3 to TQnCCS0 when TQnCE = 0. (The same value can be written when TQnCE = 1.) If rewriting was mistakenly performed, set TQnCE = 0 and then set the bits again.

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

Remark: n = 0, 1
m = 0 to 3

(7) TIQnm pin noise elimination control register n (QnmNFC)

The QnmNFC register is an 8-bit register that sets the digital noise filter of the timer Q input pin.
This register can be read or written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 8-13: TIQnm Pin Noise Elimination Control Register n (QnmNFC) Format

Address: Q00NFC: FFFFFB50H (TIQ00 pin)
Q01NFC: FFFFFB54H (TIQ01 pin)
Q02NFC: FFFFFB58H (TIQ02 pin)
Q03NFC: FFFFFB5CH (TIQ03 pin)
Q10NFC: FFFFFB60H (TIQ10 pin)
Q11NFC: FFFFFB64H (TIQ11 pin)
Q12NFC: FFFFFB68H (TIQ12 pin)
Q13NFC: FFFFFB6CH (TIQ13 pin)

Symbol	7	6	5	4	3	2	1	<0>	Address	R/W	After reset
QnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0	FFFFFB50H to FFFFFB6CH	R/W	00H

NFSTS	Selection of sampling times number for digital noise filtering
0	3 times
1	2 times

NFC2	NFC1	NFC0	Sampling clock selection
0	0	0	fxx
0	0	1	fxx/2
0	1	0	fxx/4
0	1	1	fxx/16
1	0	0	fxx/32
1	0	1	fxx/64
Other than above			Setting prohibited

Cautions: 1. Be sure to clear bits 3 to 5 and 7 to 0.

2. A signal input to the timer input pin (TIQnm) before the QnmNFC register is set is output with digital noise eliminated. Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the QnmNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation.

Remarks: 1. The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a width narrower than this may cause a miscount if it is synchronized with the sampling clock.

2. n: Number of timer channels (0, 1)
m: Number of input pins (0 to 3)

8.5 Operation

Timer Q can perform the following operations.

Operation	TQnEST Software trigger input	TIQn0 External trigger input	TQnEEE Count clock selection	Capture/ Compare Write	Compare Write
Interval timer mode	Invalid	Invalid	Internal/TIQn0 pin	Compare only	Any time write
External event counter mode[Note 1]	Invalid	Invalid	TIQn0 pin only	Compare only	Any time write
External trigger pulse output mode[Note 2]	Valid	Valid	Internal only	Compare only	Reload
One-shot pulse output mode[Note 2]	Valid	Valid	Internal only	Compare only	Any time write
PWM mode	Invalid	Invalid	Internal/TIQn0 pin	Compare only	Reload
Free-running mode	Invalid	Invalid	Internal/TIQn0 pin	Capture/compare switching enabled	Any time write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Internal only	Capture only	Not applicable

Notes: 1. When using the external event count function, set TIQn0 capture input edge detection to “Detect no edge”. (Set TQnIS1 and TQnIS0 bits of TQnIOC1 register to “00”.)

2. When using the external trigger pulse output mode, one-shot pulse mode, or pulse width measurement mode, select the internal clock as the count clock (by setting the TQnEEE bit of the TQnCTL1 register to 1).

Caution: Clearing the TQnCCR1 register to 0000H is prohibited in the one-shot pulse mode.

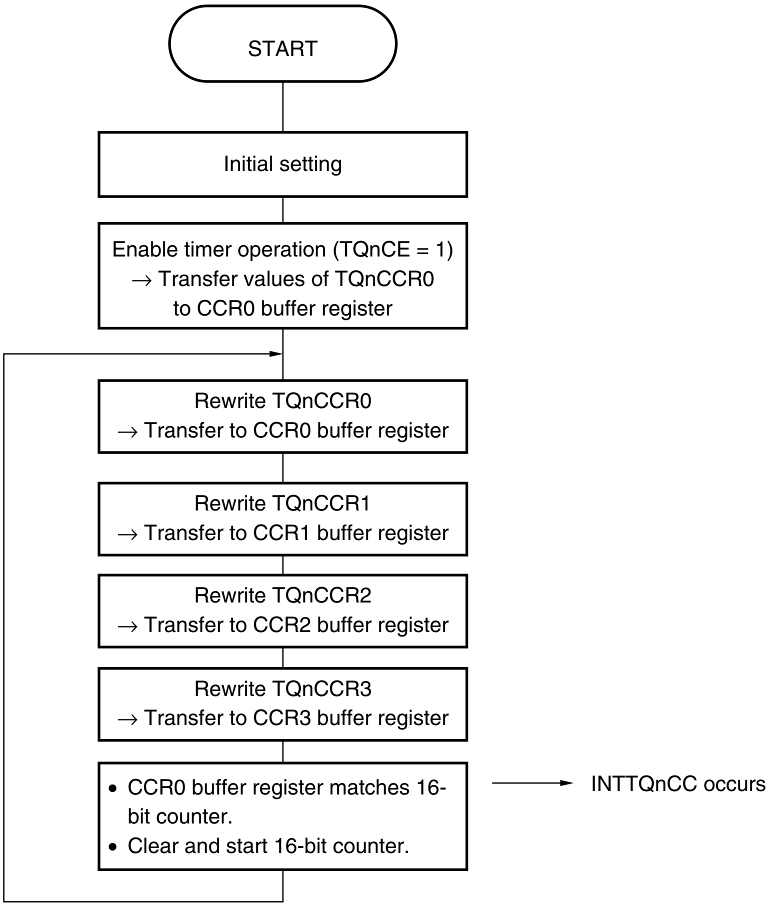
8.5.1 Anytime write and reload

Timer Q allows rewriting of the TQnCCR0 to TQnCCR3 registers while the timer is operating (TQnCE = 1). These registers are written differently (anytime write or reload) depending on the mode.

(1) Anytime write

When data is written to the TQnCCR0 to TQnCCR3 registers during timer operation, it is transferred at any time to the CCR0 buffer register and is compared with the value of the 16-bit counter.

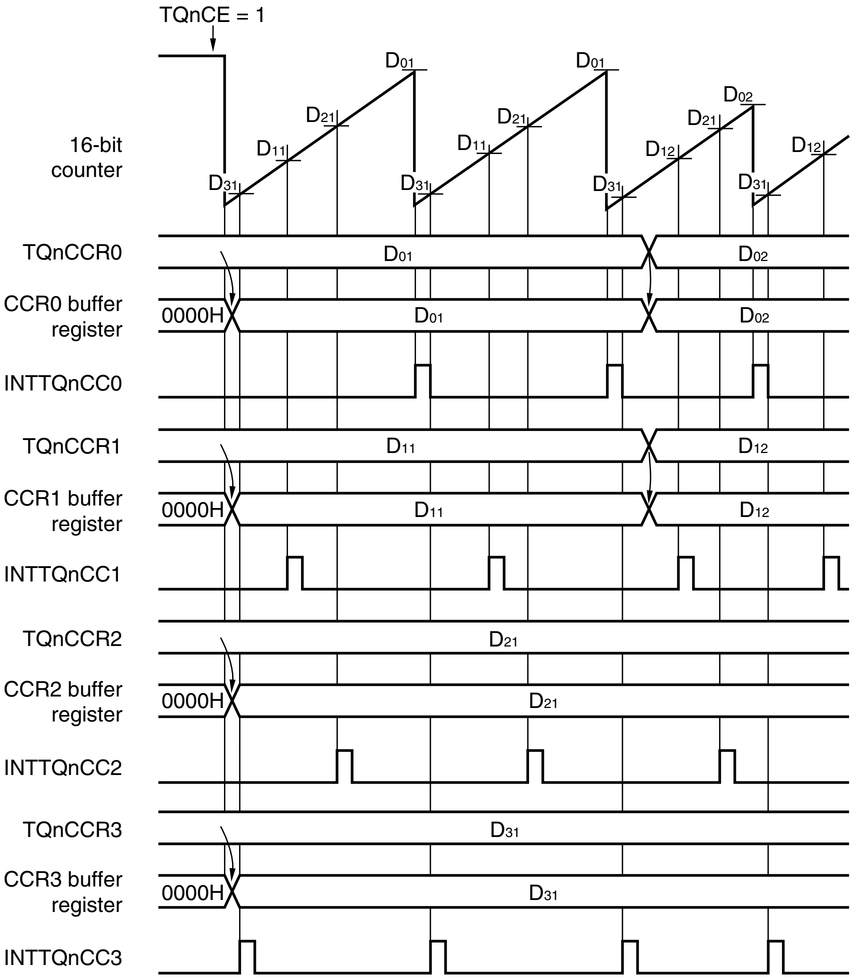
Figure 8-14: Flowchart of Basic Operation for Anytime Write


START
Initial setting
Enable timer operation (TQnCE = 1)
→ Transfer values of TQnCCR0
to CCR0 buffer register
Rewrite TQnCCR0
→ Transfer to CCR0 buffer register
Rewrite TQnCCR1
→ Transfer to CCR1 buffer register
Rewrite TQnCCR2
→ Transfer to CCR2 buffer register
Rewrite TQnCCR3
→ Transfer to CCR3 buffer register
• CCR0 buffer register matches 16-bit counter.
• Clear and start 16-bit counter.
INTTQnCC occurs

Note: n = 0, 1

Remark: The above flowchart illustrates an example of the operation in the interval timer mode.

Figure 8-15: Timing Chart of Any Time Write



Remarks: 1. D01, D02: Setting values of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31: Setting value of TQnCCR3 register (0000H to FFFFH)

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

3. n = 0,1

(2) Reload

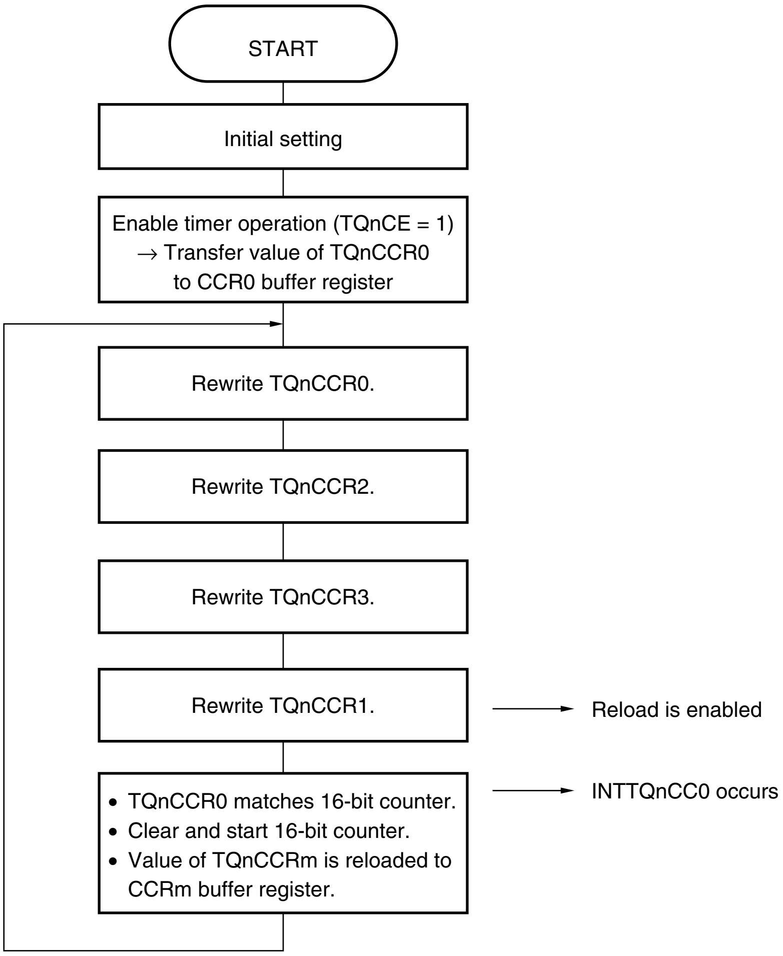
When data is written to the TQnCCRm register during timer operation, it is compared with the value of the 16-bit counter via the CCRm buffer register. The value of the TQnCCRm register can be rewritten when TQnCE = 1.

So that the set values of the TQnCCRm register is compared with the value of the 16-bit counter (the set values are reloaded to the CCRm buffer register), the value of the TQnCCR0 register must be rewritten and then a value must be written to the TQnCCR1 register before the value of the 16-bit counter matches the value of the CCRm buffer register.

When the value of the CCRm buffer register matches the value of the 16-bit counter, the value of the TQnCCRm register is reloaded to the CCRm buffer register.

Whether the next reload timing is made valid or not is controlled by writing to the TQnCCR1 register.

Figure 8-16: Flowchart of Basic Operation for Reload

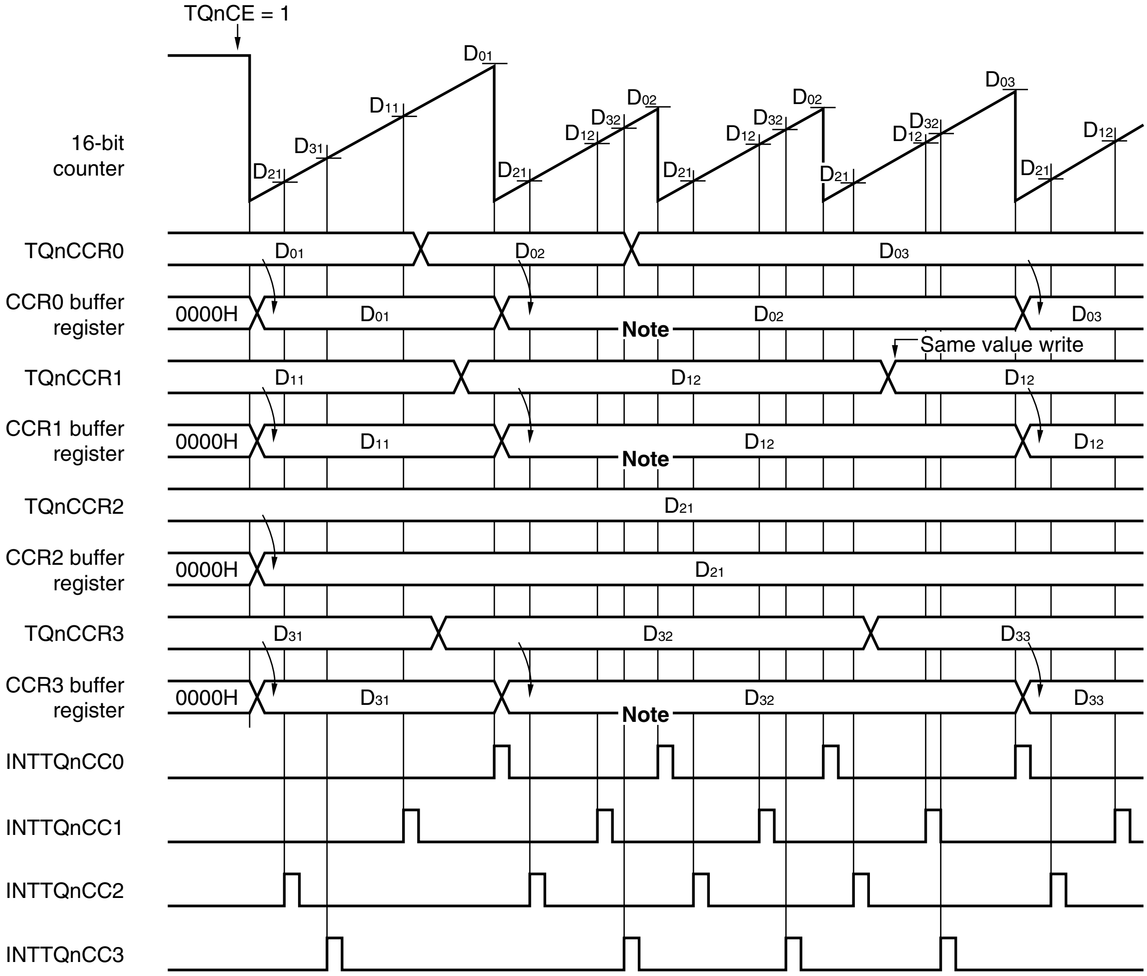

START
Initial setting
Enable timer operation (TQnCE = 1)
→ Transfer value of TQnCCR0
to CCR0 buffer register
Rewrite TQnCCR0.
Rewrite TQnCCR2.
Rewrite TQnCCR3.
Rewrite TQnCCR1.
Reload is enabled
• TQnCCR0 matches 16-bit counter.
• Clear and start 16-bit counter.
• Value of TQnCCRm is reloaded to CCRm buffer register.
INTTQnCC0 occurs

Caution: Writing the TQnCCR1 register includes an operation to enable reload. Therefore, rewrite the TQnCCR1 register after rewriting other TQnCCR registers.

Remarks: 1. The above flowchart illustrates an example of PWM mode operation.

2. n = 0,1; m=0 to 3

Figure 8-17: Timing Chart of Reload



Note: Reload is not performed because TQnCCR1 register is not written.

Remarks: 1. D01, D02, D03: Setting values of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31, D32, D33: Setting values of TQnCCR3 register (0000H to FFFFH)

2. The above flowchart illustrates the operation in the PWM mode operation

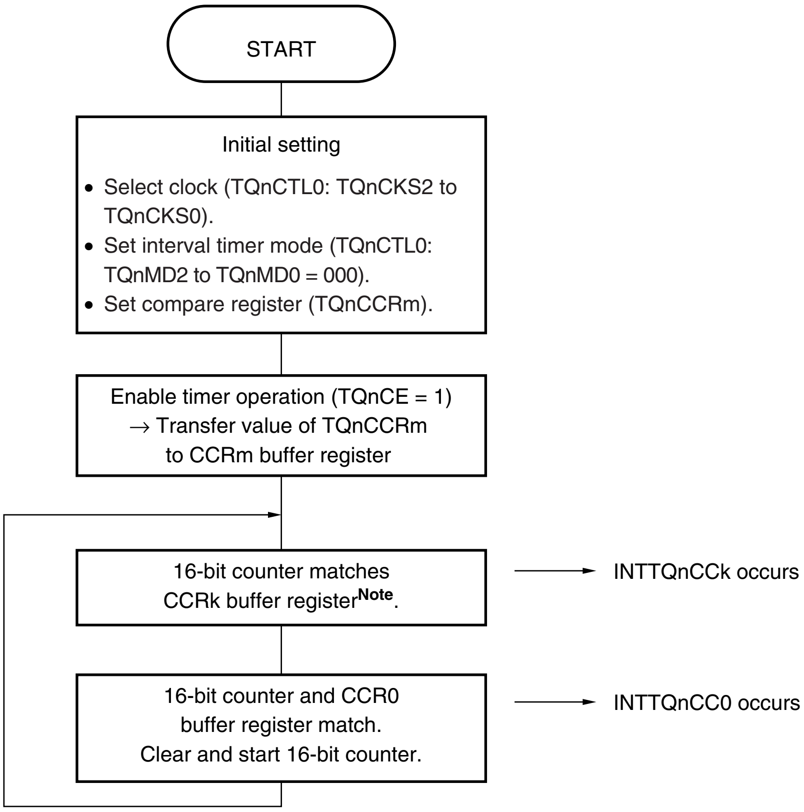
3. n = 0, 1.

8.5.2 Interval timer mode (TQnMD2 to TQnMD0 = 000)

In the interval timer mode, an interrupt request signal (INTTQnCC0) is generated when the set value of the TQnCCR0 register matches the value of the 16-bit counter, and the 16-bit counter is cleared. Rewriting the TQnCCRm register is enabled when TQnCE = 1. When a value is set to the TQnCCRm register by a write instruction from the CPU, it is transferred to the CCRm buffer register by means of anytime write, and is compared with the value of the 16-bit counter.
In the interval timer mode, the 16-bit counter can be cleared only when its value matches the value of the CCR0 buffer register.
The 16-bit counter is not cleared by using the TQnCCRk register.
However, the set value of the TQnCCRk register is transferred to the CCRk buffer register and compared with the value of the 16-bit counter. As a result, an interrupt request (INTTQnCCk) is generated. The value can also be output from the TOQnm pin by setting the TQnOEm bit to 1.
When the TQnCCRk register is not used, it is recommended to set the TQnCCRk register to FFFFH.
When performing timer output with the TOQnk pin, set the same values to the TQnCCR0 register and one of the TQnCCR1 to TQnCCR3 registers since the 16-bit timer counter cannot be cleared with the TQnCCRk register.

Remark: n = 0 to 1,
m = 0 to 3
k = 1 to 3

Figure 8-18: Flowchart of Basic Operation in Interval Timer Mode

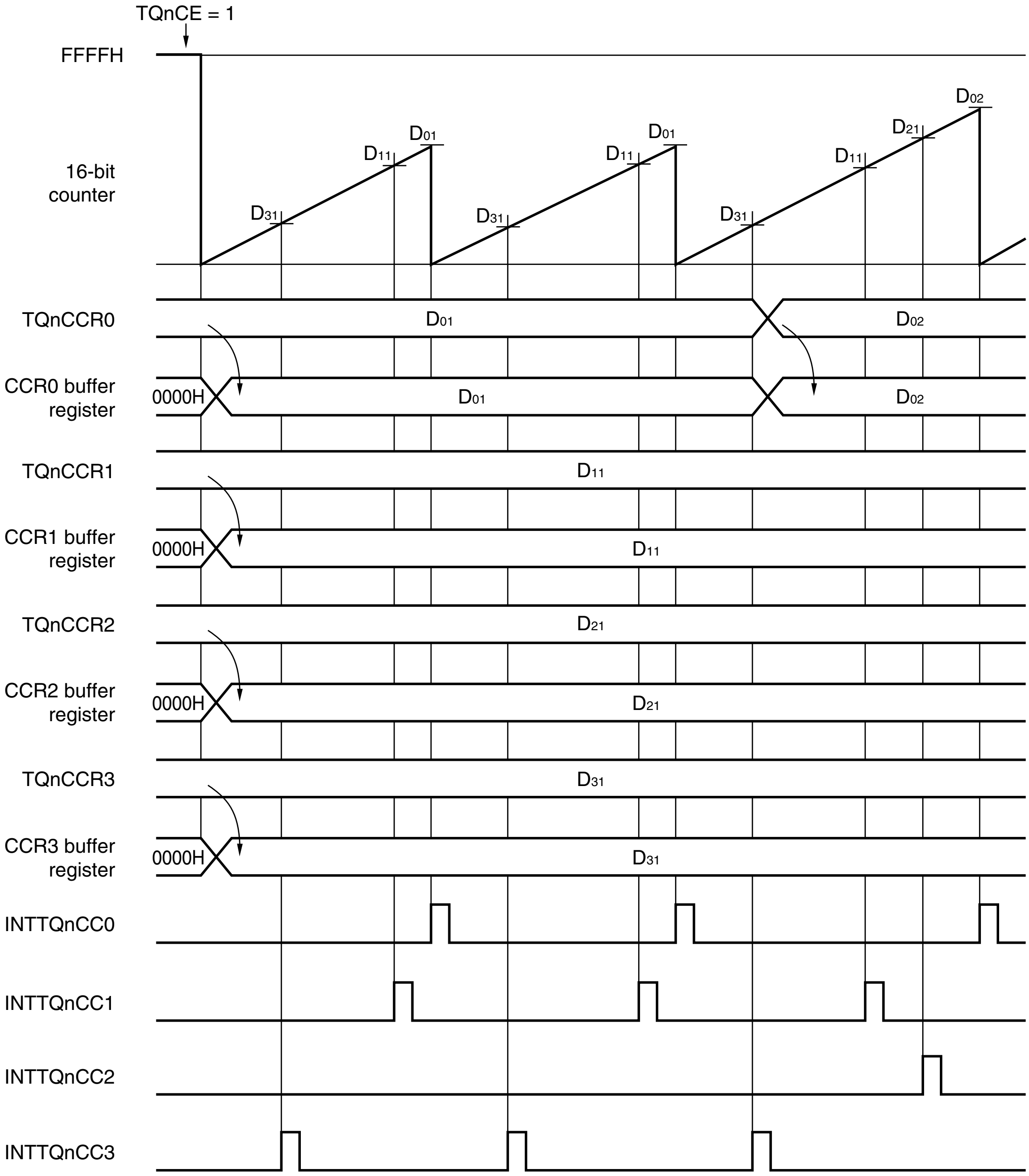

START
Initial setting
• Select clock (TQnCTL0: TQnCKS2 to TQnCKS0).
• Set interval timer mode (TQnCTL0: TQnMD2 to TQnMD0 = 000).
• Set compare register (TQnCCRm).
Enable timer operation (TQnCE = 1) → Transfer value of TQnCCRm to CCRm buffer register
16-bit counter matches CCRk buffer registerNote.
INTTQnCCk occurs
16-bit counter and CCR0 buffer register match. Clear and start 16-bit counter.
INTTQnCC0 occurs

Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and TQnCCRk.

Remark: n = 0,1; m = 0 to 3; k = 1 to 3

Figure 8-19: Basic Operation Timing in Interval Timer Mode (1/2)

(a) When only TQnCCR0 register value is rewritten and TOQnm is not output



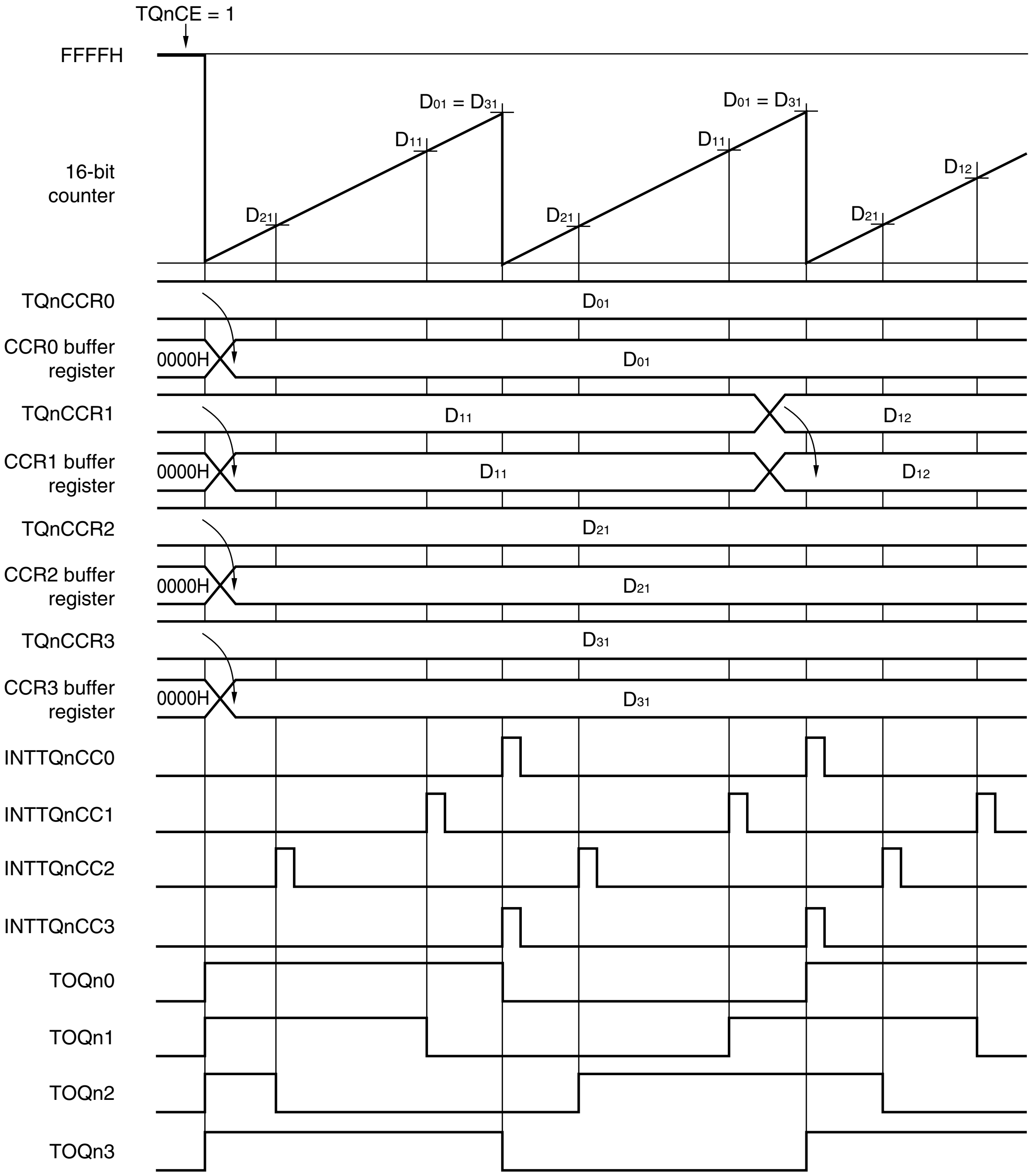
Remarks: 1. D01, D02: Setting values of TQnCCR0 register (0000H to FFFFH)
D11: Setting value of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31: Setting value of TQnCCR3 register (0000H to FFFFH)

2. Interval time = (Dmk + 1) × (count clock cycle)

3. n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-19: Basic Operation Timing in Interval Timer Mode (2/2)

(b) When D01 = D31, only TQnCCR1 register value is rewritten, and TOQnm is output



Remarks: 1. D01: Setting value of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31: Setting value of TQnCCR3 register (0000H to FFFFH)

2. Interval time = (Dmk + 1) × (count clock cycle)

3. n = 0,1; m = 0 to 3; k = 1 to 3

8.5.3 External event counter mode (TQnMD2 to TQnMD0 = 001)

In the external event count mode, the external event count input (TIQn0 pin input) is used as a count-up signal.

Regardless of the setting of the TQnEEE bit of the TQnCTL0 register, 16-bit timer/event counter Q counts up the external event count input (TIQn0 pin input) when it is set in the external event count mode.

In the external event count mode, an interrupt request (INTTQnCC0) is generated when the set value of the TQnCCR0 register matches the value of the 16-bit counter, and the value of the 16-bit counter is cleared.

When a value is set to the TQnCCRm register by a write instruction from the CPU, it is transferred to the CCRm buffer register, and is compared with the value of the 16-bit counter.

In the external event count mode, the 16-bit counter can be cleared only when its value matches the value of the CCR0 buffer register.

The 16-bit counter cannot be cleared by using the TQnCCRk register.

However, the set value of the TQnCCRk register is transferred to the CCRk buffer register and is compared with the value of the 16-bit counter. As a result, an interrupt request (INTTQnCCk) is generated.

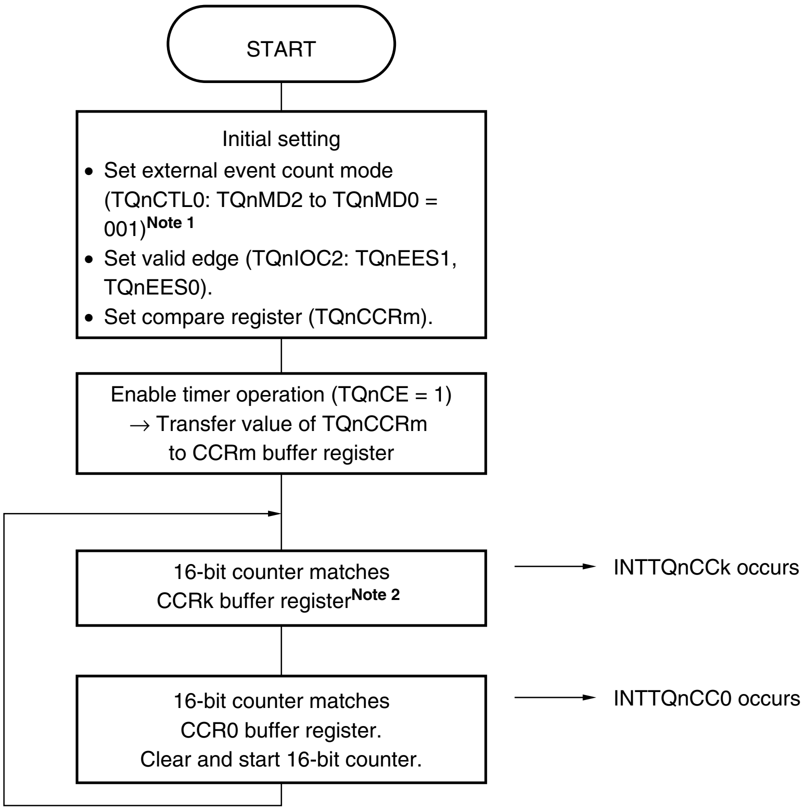
By setting the TQnOEm bit to 1, a signal can be output from the TOQnm pin.

When performing timer output with the TOQnk pin, set the same values to the TQnCCR0 register and the TQnCCRk register since the 16-bit counter cannot be cleared with the CCRk buffer register.

Rewriting the TQnCCR0 register is enabled when TQnCE = 1. When the TQnCCRk register is not used, it is recommended to set TQnCCRk to FFFFH.

Remark: n = 0 to 1,
m = 0 to 3
k = 1 to 3

Figure 8-20: Flowchart of Basic Operation in External Event Counter Mode


START
Initial setting
• Set external event count mode (TQnCTL0: TQnMD2 to TQnMD0 = 001)Note 1
• Set valid edge (TQnIOC2: TQnEES1, TQnEES0).
• Set compare register (TQnCCRm).
Enable timer operation (TQnCE = 1)
→ Transfer value of TQnCCRm to CCRm buffer register
16-bit counter matches CCRk buffer registerNote 2
INTTQnCCk occurs
16-bit counter matches CCR0 buffer register. Clear and start 16-bit counter.
INTTQnCC0 occurs

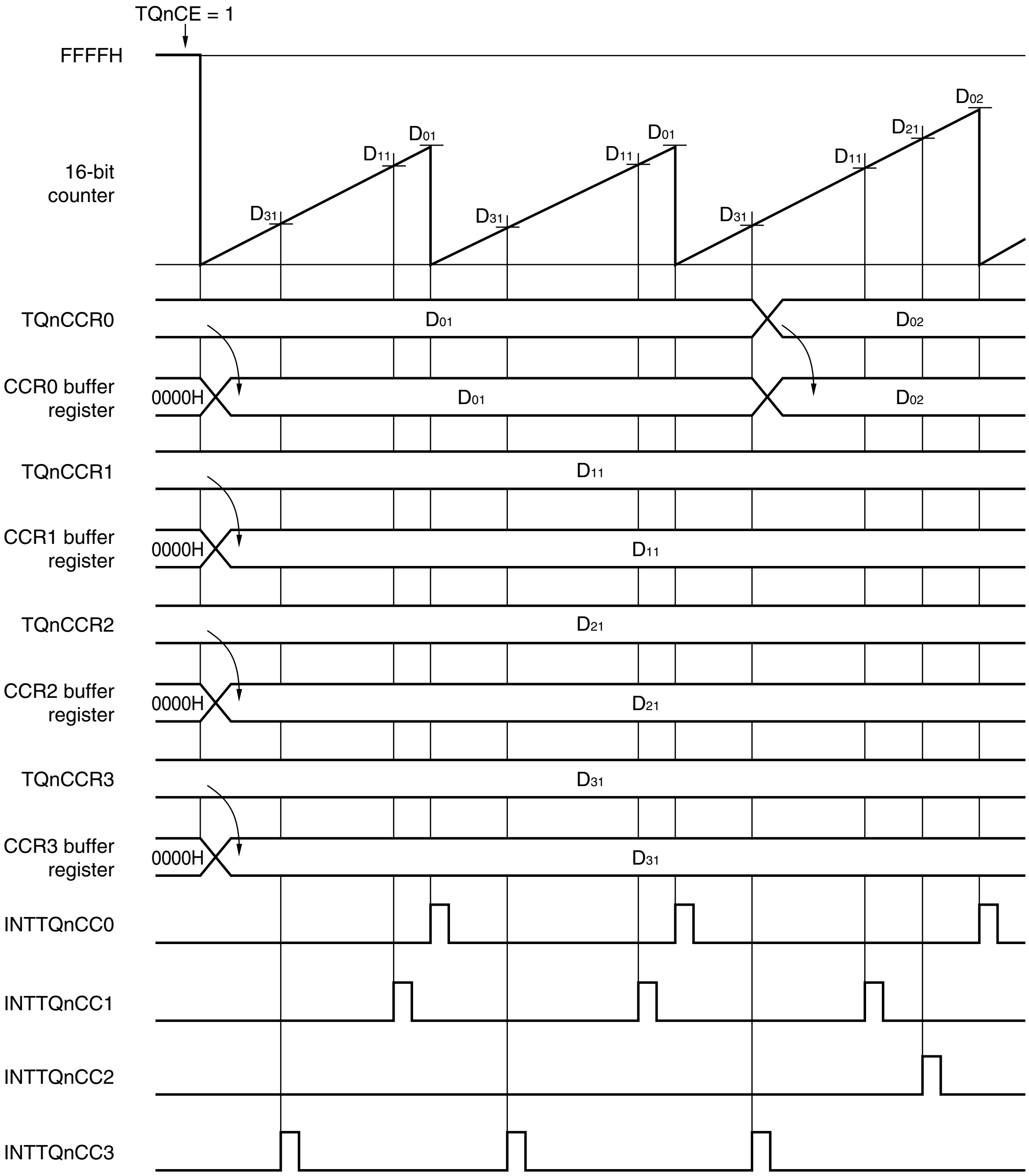
Notes: 1. Selection of the TQnEEE bit has no influence.

2. The 16-bit counter is not cleared when it matches the CCRk buffer register.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-21: Basic Operation Timing in External Event Counter Mode (1/2)

(a) When only TQnCCR0 register value is rewritten and TOQnm is not output



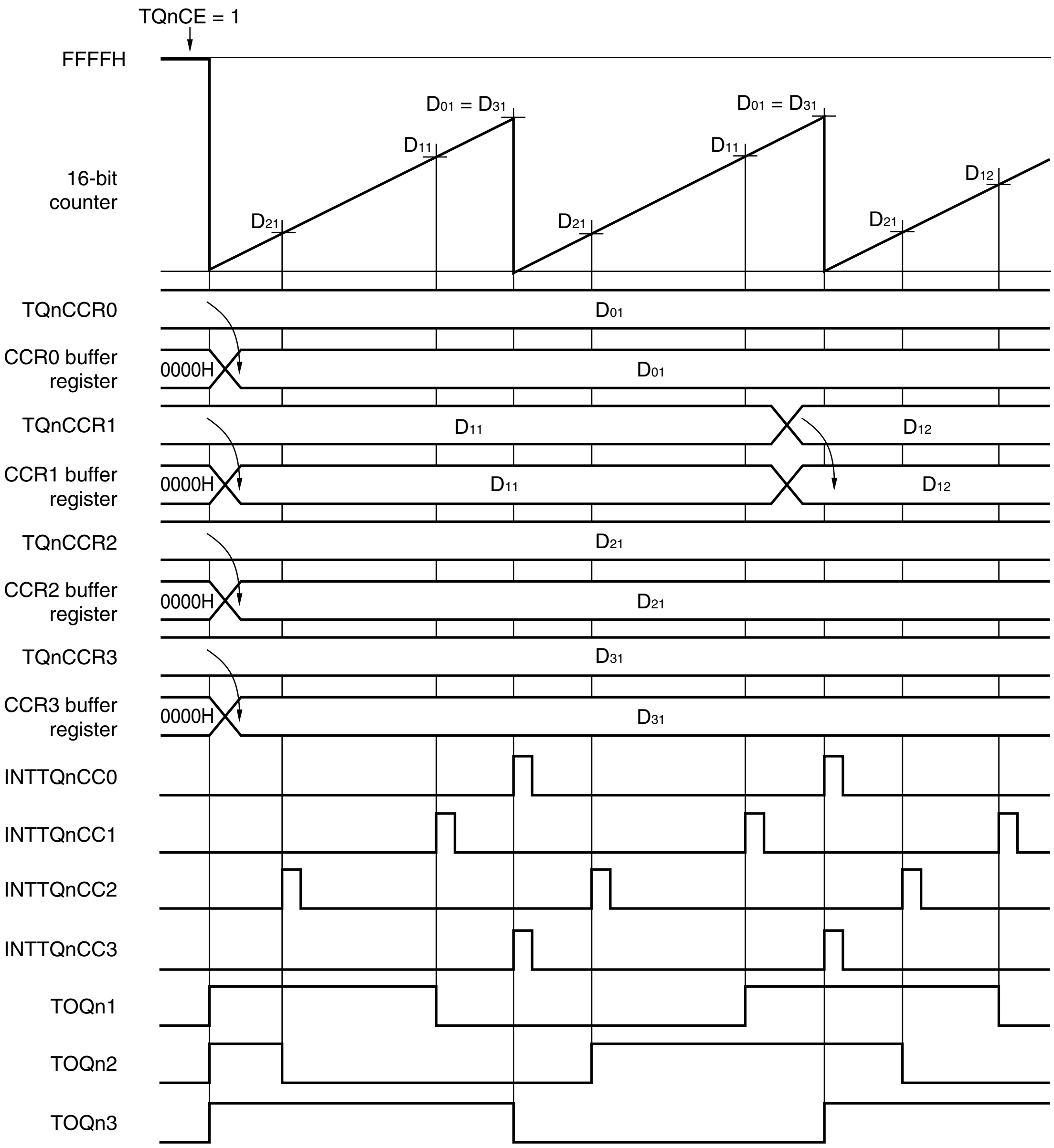
Remarks: 1. D01, D02: Setting values of TQnCCR0 register (0000H to FFFFH)
D11: Setting value of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31: Setting value of TQnCCR3 register (0000H to FFFFH)

2. Interval time = (Dmk + 1) × (count clock cycle)

3. n = 0 to 1, m = 0 to 3, k = 1 to 3

Figure 8-21: Basic Operation Timing in External Event Counter Mode (2/2)

(b) When D01 = D31, only TQnCCR1 register is rewritten, and TOQnm is output



Remarks: 1. D01: Setting value of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31: Setting value of TQnCCR3 register (0000H to FFFFH)

2. Interval time = (Dmk + 1) × (count clock cycle)

3. n = 0 to 1, m = 0 to 3, k = 1 to 3

8.5.4 External trigger pulse mode (TQnMD2 to TQnMD0 = 010)

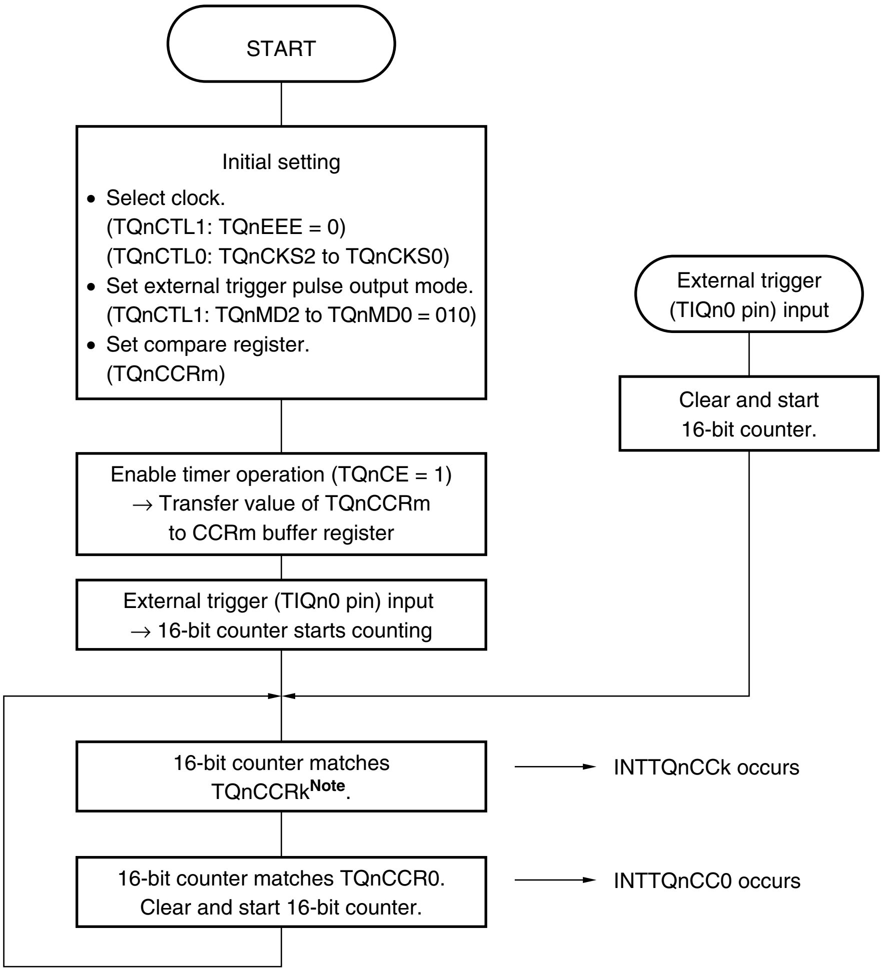
When TQnCE = 1 in the external trigger pulse mode, the 16-bit counter stops at FFFFH and waits for input of an external trigger (TIQn0 pin input). When the counter detects the edge of the external trigger (TIQn0 pin input), it starts counting up.
The duty factor of the signal output from the TOQnk pin is set by a reload register (TQnCCRk) and the period is set by a compare register (TQnCCR0).
Rewriting the TQnCCRm register is enabled when TQnCE = 1. So that the set value of the TQnCCRm register after rewriting is compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), the TQnCCR0 register must be rewritten and then a value is written to the
TQnCCR1 register before the value of the 16-bit counter matches the value of the TQnCCR0 register. When the value of the TQnCCR0 register later matches the value of the 16-bit counter, the value of the TQnCCRm register is reloaded.
Whether the next reload timing is made valid or not is controlled by writing to the TQnCCR1 register. Therefore, write the same value to the TQnCCR1 register when it is necessary to rewrite the value of only the TQnCCR0 register.
Reload is invalid when only the TQnCCR0 register is rewritten.
To stop timer Q, clear TQnCE to 0. If the edge of the external trigger (TIQn0 pin input) is detected more than once in the external trigger pulse mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. To realize the same function as the external trigger pulse mode by using a software trigger instead of the external trigger input (TIQn0 pin input) (software trigger pulse mode), a software trigger is generated by setting the TQnEST bit of the TQnCTL1 register to 1. The waveform of the external trigger pulse is output from TOQnk.
In the external trigger pulse mode, the capture function of the TQnCCRm register cannot be used because this register can be used only as a compare register.

Caution: In the external trigger pulse mode, select the internal clock (TQnEEE bit of TQnCTL1 register = 0) as the count clock.

Remarks: 1. For the reload operation when TQnCCRm is rewritten during timer operation, refer to **8.5.6 PWM mode (TQnMD2 to TQnMD0 = 100)**.

2. n = 0 to 1
m = 0 to 3
k = 1 to 3

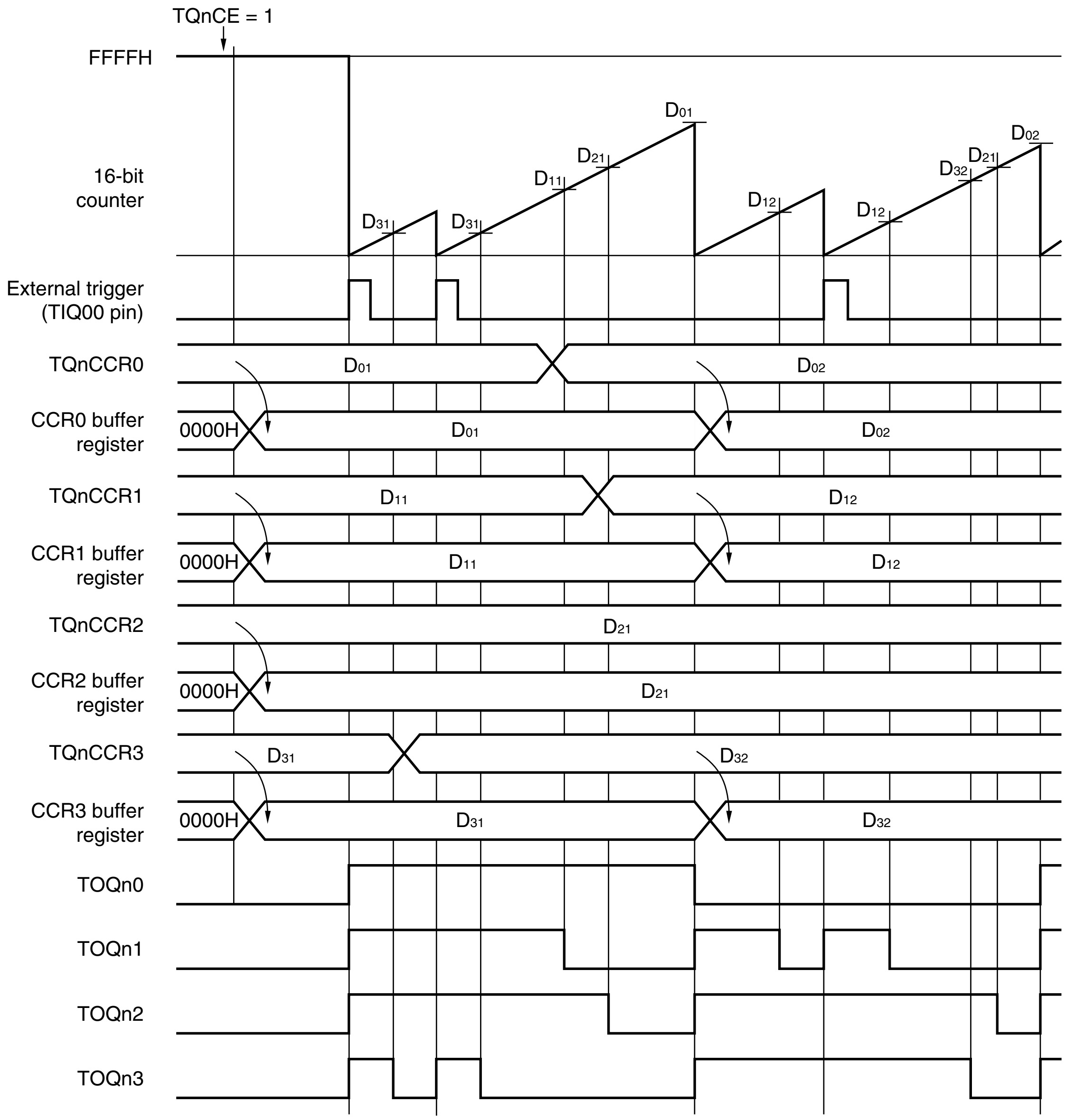
Figure 8-22: Flowchart of Basic Operation in External Trigger Pulse Output Mode


START
Initial setting
• Select clock.
(TQnCTL1: TQnEEE = 0)
(TQnCTL0: TQnCKS2 to TQnCKS0)
• Set external trigger pulse output mode.
(TQnCTL1: TQnMD2 to TQnMD0 = 010)
• Set compare register.
(TQnCCRm)
Enable timer operation (TQnCE = 1)
→ Transfer value of TQnCCRm
to CCRm buffer register
External trigger (TIQn0 pin) input
→ 16-bit counter starts counting
16-bit counter matches
TQnCCRkNote.
INTTQnCCk occurs
16-bit counter matches TQnCCR0.
Clear and start 16-bit counter.
INTTQnCC0 occurs
External trigger
(TIQn0 pin) input
Clear and start
16-bit counter.

Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCRk buffer register.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-23: Basic Operation Timing in External Trigger Pulse Output Mode



Remarks: 1. D01, D02: Setting values of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31, D32: Setting values of TQnCCR3 register (0000H to FFFFH)

2. Duty of TOQnk output = (Set value of TQnCCRk register) / (Set value of TQnCCR0 register)
Cycle of TOQnk output = (Set value of TQnCCR0 register) × (Count clock cycle)

3. n = 0 to 1, m = 0 to 3, k = 1 to 3

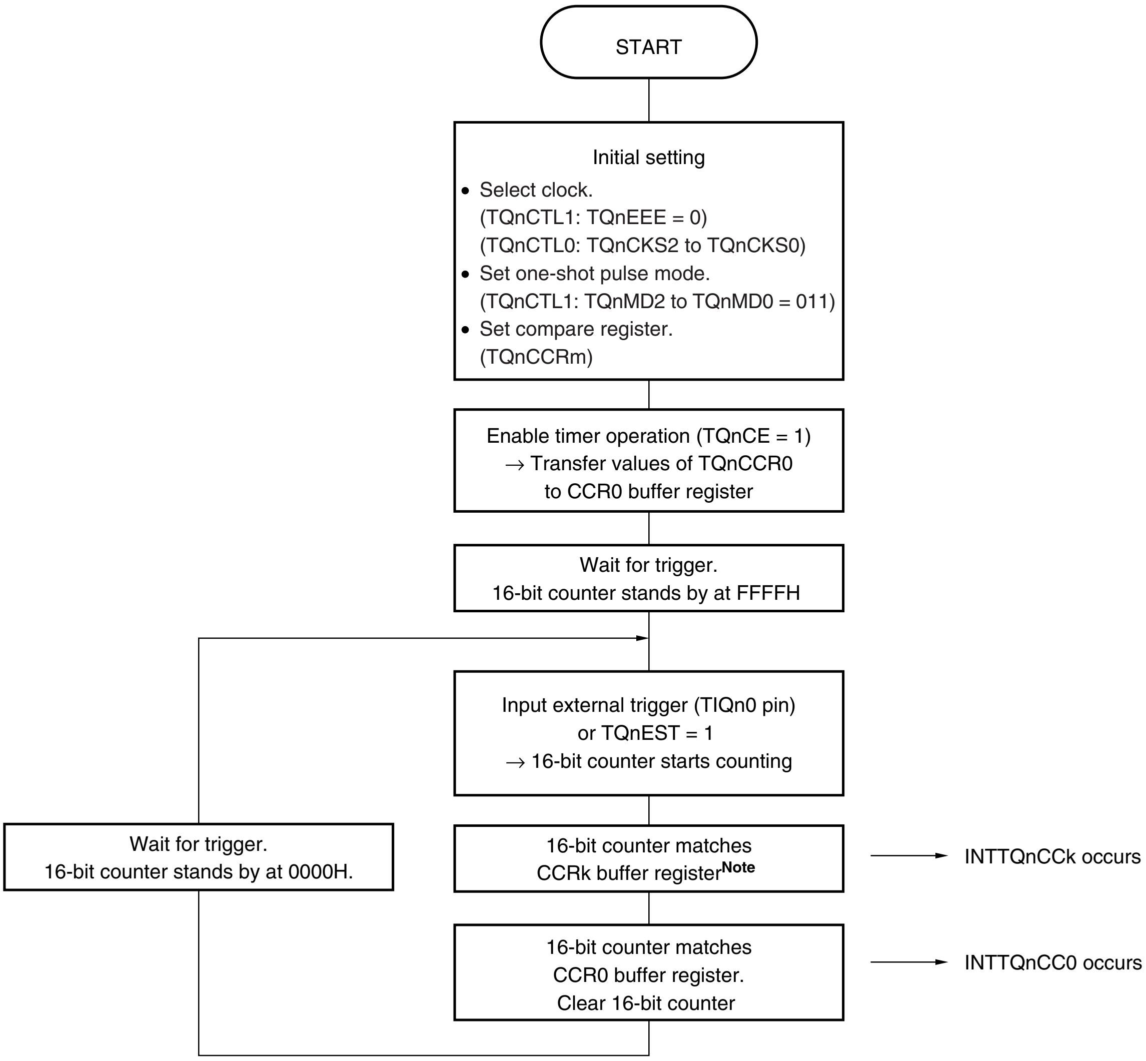
8.5.5 One-shot pulse mode (TQnMD2 to TQnMD0 = 011)

When TQnCE is set to 1 in the one-shot pulse mode, the 16-bit counter waits for the setting of the TQnEST bit (to 1) or a trigger that is input when the edge of the TIQn0 pin is detected, while holding FFFFH. When the trigger is input, the 16-bit counter starts counting up. When the value of the 16-bit counter matches the value of the CCRk buffer register that has been transferred from the TQnCCR0 register, TOQnk goes high. When the value of the 16-bit counter matches the value of the CCR0 buffer register that has been transferred from the TQnCCR0 register, TOQnk goes low, and the 16-bit counter is cleared to 0000H and stops. Input of a second or subsequent trigger is ignored while the 16-bit counter is operating. Be sure to input a second trigger while the 16-bit counter is stopped at 0000H.
In the one-shot pulse mode, rewriting the TQnCCRm register is enabled when TQnCE = 1. The set value of the TQnCCRm register becomes valid after a write instruction from the CPU is executed. They are then transferred to the CCRm buffer register, and compared with the value of the 16-bit counter. The waveform of the one-shot pulse is output from the TOQnk pin. The TOQnm pin produces a toggle output when the value of the 16-bit counter matches the value of the TQnCCR0 register.
In the one-shot pulse mode, the TQnCCRm register function only as a compare register. It cannot be used as a capture register.

Caution: In the one-shot pulse mode, select the internal clock (TQnEEE bit of TQnCTL1 register = 0) for the count clock.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-24: Flowchart of Basic Operation in One-Shot Pulse Mode

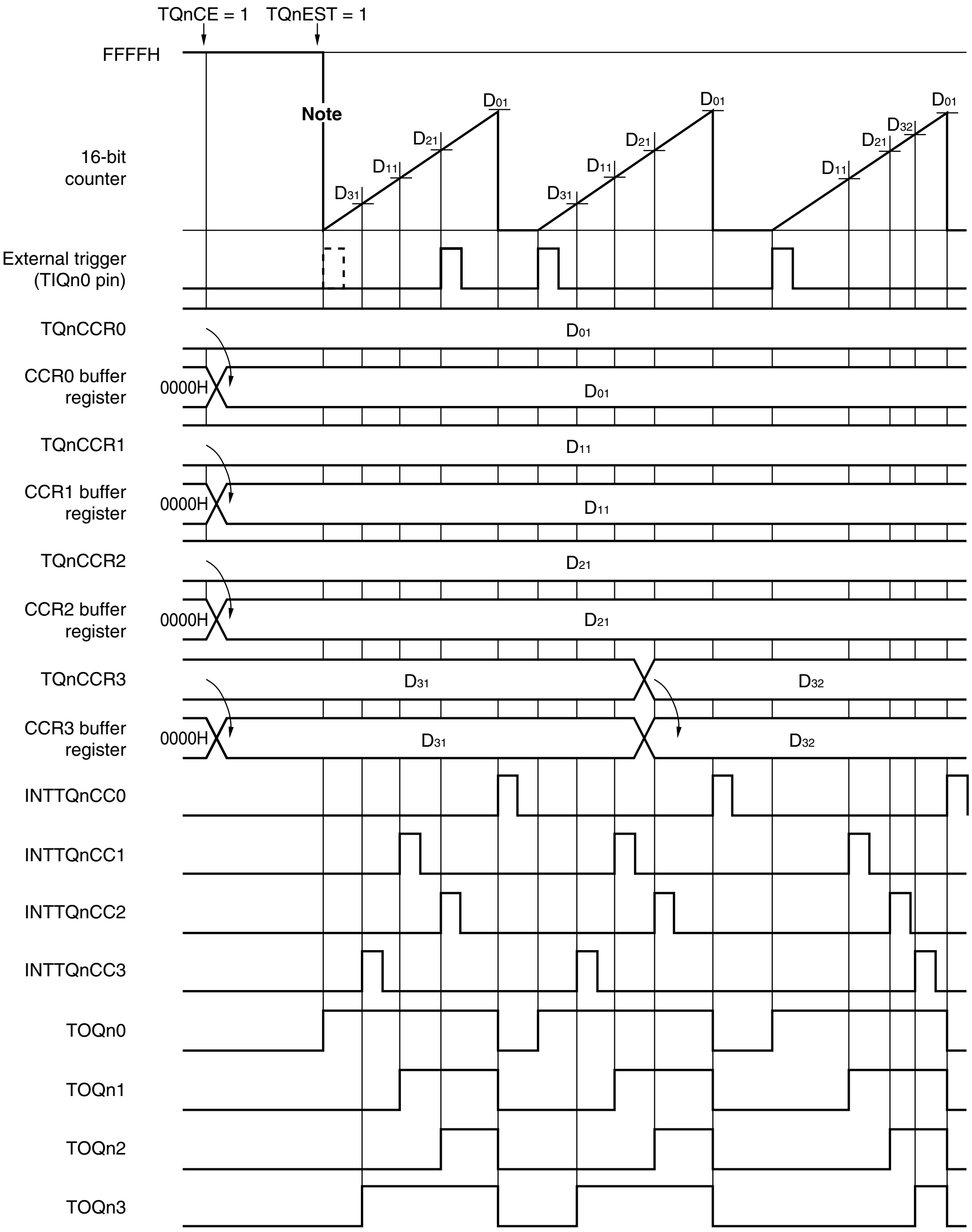

START
Initial setting
• Select clock.
(TQnCTL1: TQnEEE = 0)
(TQnCTL0: TQnCKS2 to TQnCKS0)
• Set one-shot pulse mode.
(TQnCTL1: TQnMD2 to TQnMD0 = 011)
• Set compare register.
(TQnCCRm)
Enable timer operation (TQnCE = 1)
→ Transfer values of TQnCCR0
to CCR0 buffer register
Wait for trigger.
16-bit counter stands by at FFFFH
Input external trigger (TIQn0 pin)
or TQnEST = 1
→ 16-bit counter starts counting
Wait for trigger.
16-bit counter stands by at 0000H.
16-bit counter matches
CCRk buffer register Note
INTTQnCCk occurs
16-bit counter matches
CCR0 buffer register.
Clear 16-bit counter
INTTQnCC0 occurs

Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCRk buffer register.

Caution: The 16-bit counter is not cleared even if the trigger is input while the counter is counting up, and the trigger input is ignored.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-25: Timing of Basic Operation in One-Shot Pulse Mode



Note: The 16-bit counter starts counting up when either TQnEST is set to 1 or external trigger (TIQn0 pin) is input.

Remarks: 1. D01: Setting value of TQnCCR0 register (0000H to FFFFH)
D11: Setting value of TQnCCR1 register (0000H to FFFFH)
D21: Setting value of TQnCCR2 register (0000H to FFFFH)
D31, D32: Setting value of TQnCCR3 register (0000H to FFFFH)

2. n = 0, 1

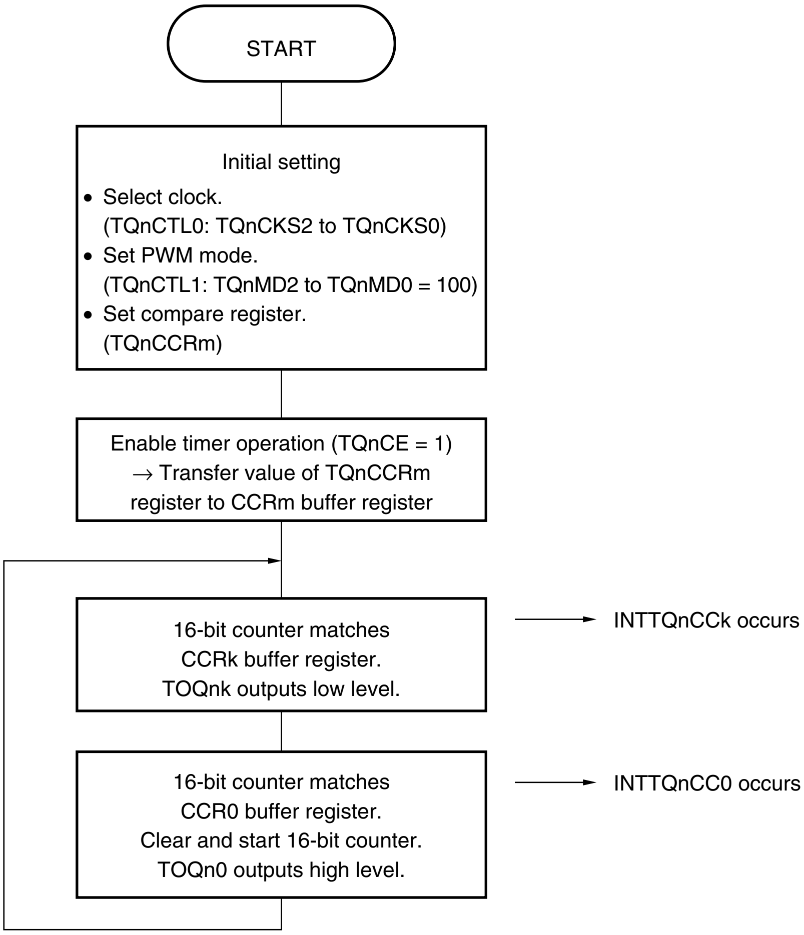
8.5.6 PWM mode (TQnMD2 to TQnMD0 = 110)

In the PWM mode, TMQn capture/compare register k (TQnCCRk) is used to set the duty factor and TMQn capture/compare register 0 (TQnCCR0) is used to set the cycle.
By using these two registers and operating the timer, variable-duty PWM is output.
Rewriting the TQnCCRm register is enabled when TQnCE = 1.
So that the set value of the TQnCCRm register is compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), the TQnCCR0 register must be rewritten and then a value must be written to the TQnCCR1 register before the value of the 16-bit counter matches the value of the TQnCCR0 register. The value of the TQnCCRm register is reloaded when the value of the TQnCCR0 register later matches the value of the 16-bit counter.
Whether the next reload timing is made valid or not is controlled by writing to the TQnCCR1 register. Therefore, write the same value to the TQnCCR1 register even when only the value of the TQnCCR0 register needs to be rewritten. Reload is invalid when only the value of the TQnCCR0 register is rewritten.
To stop timer Q, clear TQnCE to 0. The waveform of PWM is output from the TOQnk pin.
The TOQn0 pin produces a toggle output when the 16-bit counter matches the TQnCCR0 register.
In the PWM mode, the TQnCCRm register is used only as a compare register. It cannot be used as a capture register.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-26: Flowchart of Basic Operation in PWM Mode (1/2)

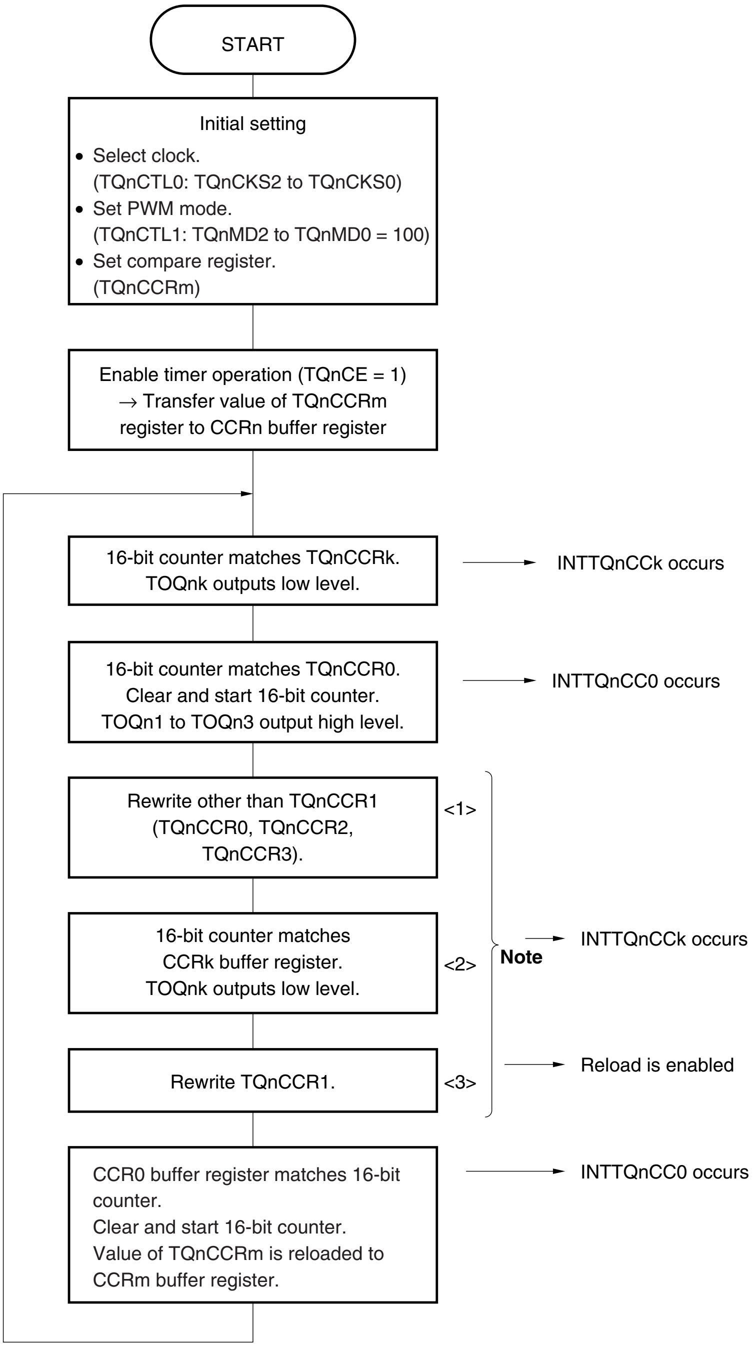
(a) When values of TQnCCRm register is not rewritten during timer operation



Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-26: Flowchart of Basic Operation in PWM Mode (2/2)

(b) Value of TQnCCRm register rewritten during timer operation

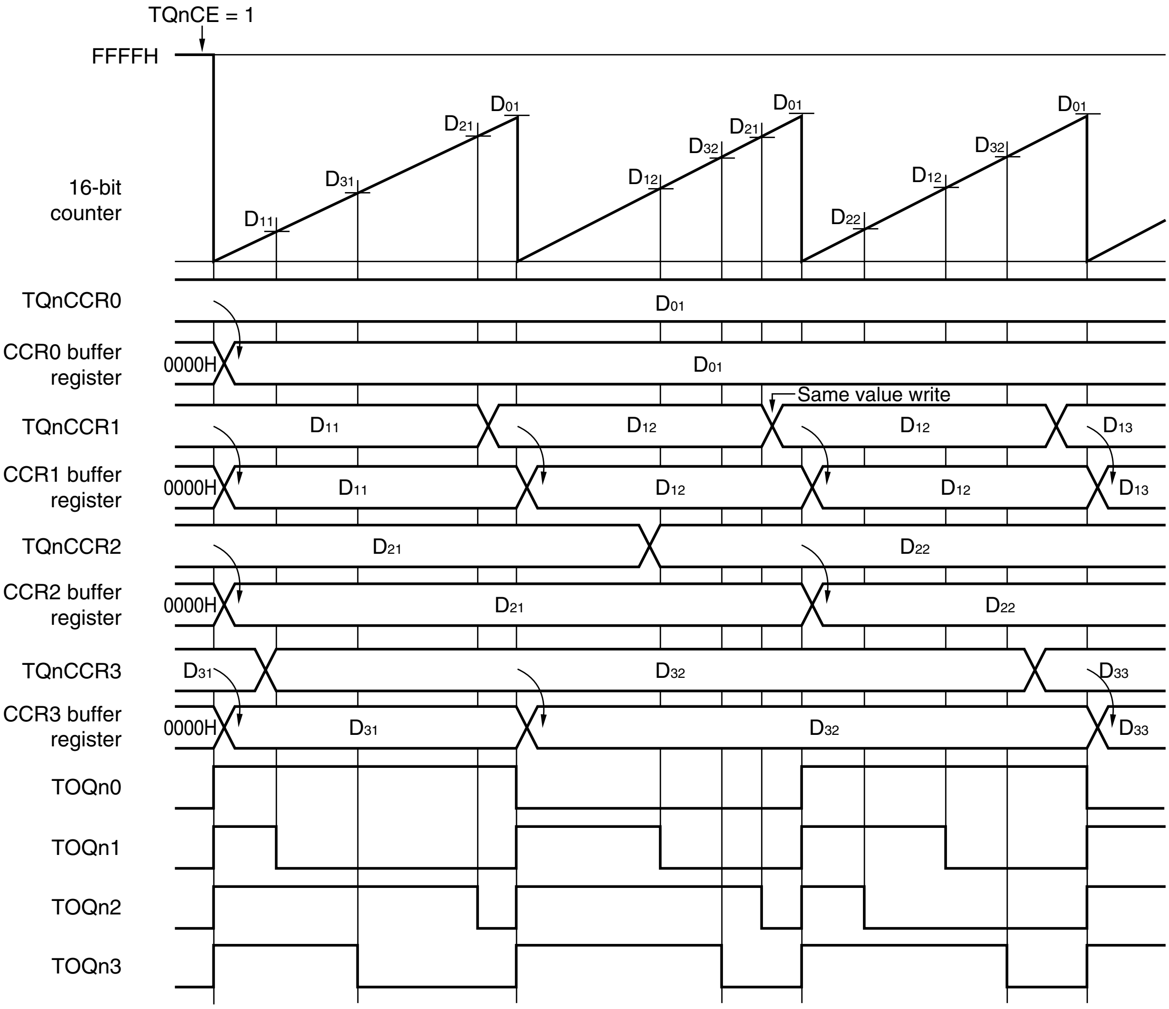

START
Initial setting
• Select clock.
(TQnCTL0: TQnCKS2 to TQnCKS0)
• Set PWM mode.
(TQnCTL1: TQnMD2 to TQnMD0 = 100)
• Set compare register.
(TQnCCRm)
Enable timer operation (TQnCE = 1)
→ Transfer value of TQnCCRm register to CCRn buffer register
16-bit counter matches TQnCCRk.
TOQnk outputs low level.
INTTQnCCk occurs
16-bit counter matches TQnCCR0.
Clear and start 16-bit counter.
TOQn1 to TOQn3 output high level.
INTTQnCC0 occurs
Rewrite other than TQnCCR1 (TQnCCR0, TQnCCR2, TQnCCR3).
<1>
16-bit counter matches CCRk buffer register.
TOQnk outputs low level.
<2>
Note
INTTQnCCk occurs
Rewrite TQnCCR1.
<3>
Reload is enabled
CCR0 buffer register matches 16-bit counter.
Clear and start 16-bit counter.
Value of TQnCCRm is reloaded to CCRm buffer register.
INTTQnCC0 occurs

Note: The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TQnCCRk, but make sure that step <3> comes after step <1>.

Remark: n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-27: Basic Operation Timing in PWM Mode (1/2)

(a) When rewriting values of TQnCCR1 to TQnCCR3 registers



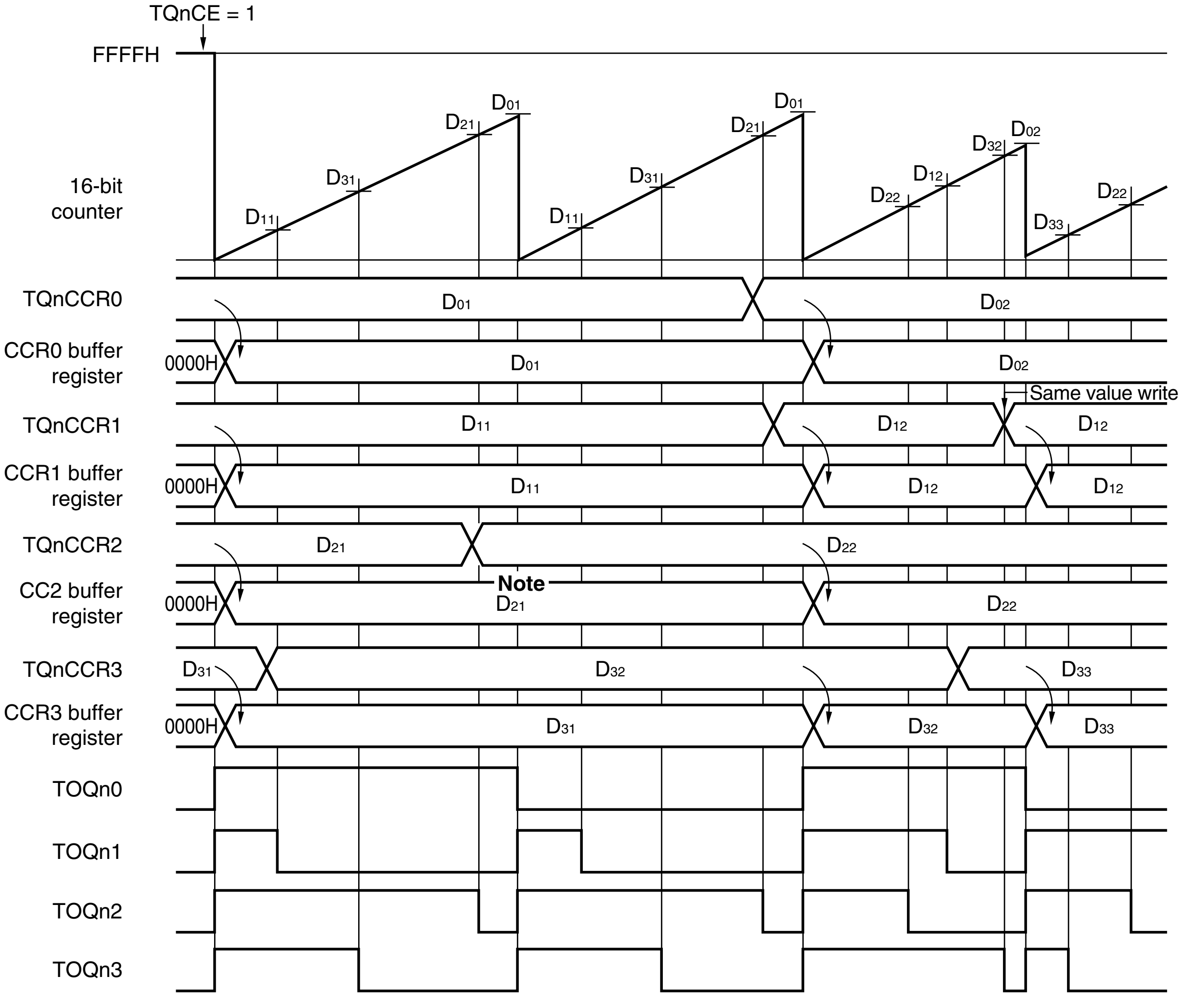
Remarks: 1. D10: Setting value of TQnCCR0 register (0000H to FFFFH)
D11, D12, D13: Setting values of TQnCCR1 register (0000H to FFFFH)
D21, D22: Setting values of TQnCCR2 register (0000H to FFFFH)
D31, D32, D33: Setting values of TQnCCR3 register (0000H to FFFFH)

2. Duty of TOQnk output = (Set value of TQnCCRk register) / (Set value of TQnCCR0 register)
Cycle of TOQnk output = (Set value of TQnCCR0 register) × (Count clock cycle)
Toggle width of TOQn0 output = (Set value of TQnCCR0 register + 1) × (Count clock cycle)

3. n = 0 to 1
m = 0 to 3
k = 1 to 3

Figure 8-27: Basic Operation Timing in PWM Mode (2/2)

(b) When rewriting values of TQnCCR0 to TQnCCR3 registers



Note: Reload is not performed because the TQnCCR1 register was not rewritten.

Remarks: 1. D01, D02: Setting values of TQnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D21, D22: Setting values of TQnCCR2 register (0000H to FFFFH)
D31, D32, D33: Setting values of TQnCCR3 register (0000H to FFFFH)

2. Duty of TOQnk output = (Set value of TQnCCRk register) / (Set value of TQnCCR0 register)
Cycle of TOQnk output = (Set value of TQnCCR0 register) × (Count clock cycle)
Toggle width of TOQn0 output = (Set value of TQnCCR0 register + 1) × (Count clock cycle)

3. n = 0 to 1, k = 1 to 3

8.5.7 Free-running mode (TQnMD2 to TQnMD0 = 101)

In the free-running mode, both the interval function and the compare function can be realized by operating the 16-bit counter as a free-running counter and selecting capture/compare operation with the TQnCCS3 to TQnCCS0 bits of the TQnOPT0 register.
The settings of the TQnCCS3 to TQnCCS0 bits of the TQnOPT0 register are valid only in the free-running mode.

TQnCCSm	Operation
0	Use TQnCCRm register as compare register
1	Use TQnCCRm register as capture register

- When TQnCCRm register is used as compare register

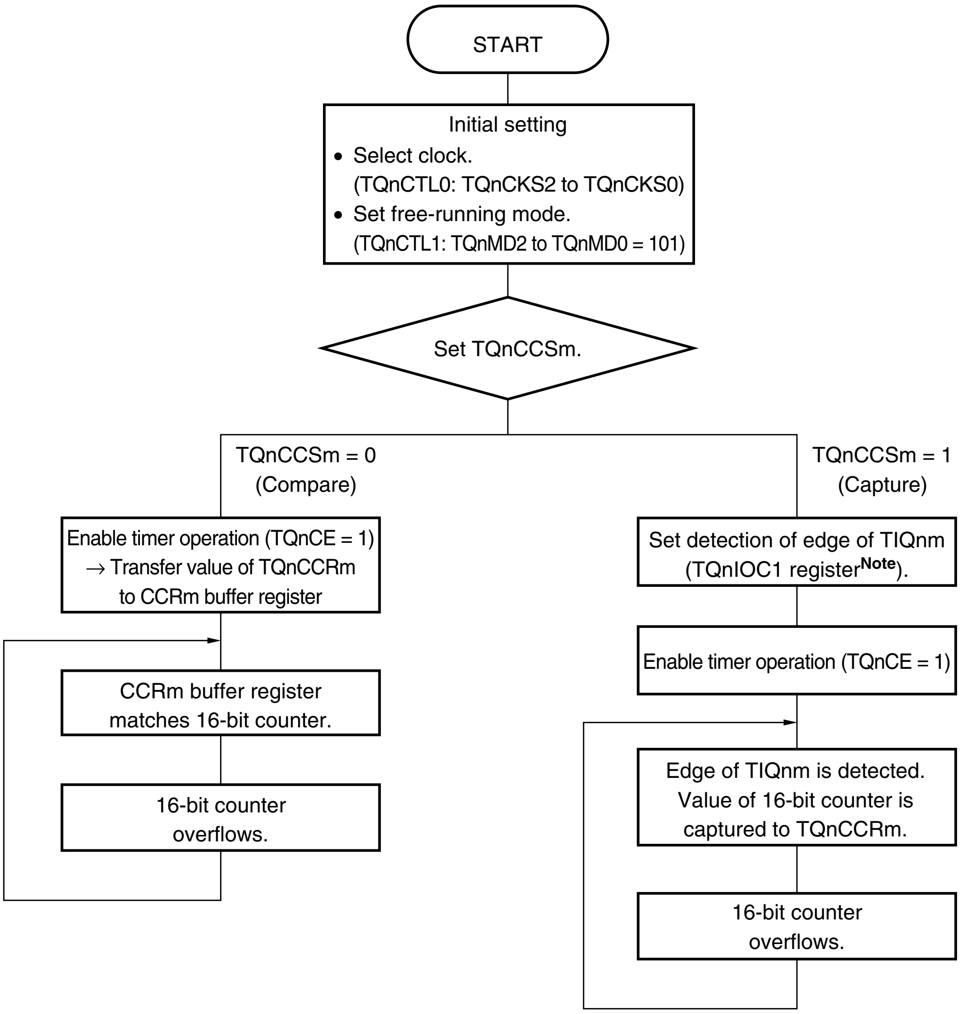
When the value of the 16-bit counter matches the value of the CCRm buffer register in the free-running mode, an interrupt is generated (interval function).
Rewriting the value of the compare register is enabled during timer operation, and a value can be written to the register at any time (when the value to be compared is written to the register, it is synchronized with the internal clock and compared with the value of the 16-bit counter).
If timer output (TOQnm) is enabled, TOQnm produces a toggle output when the value of the 16-bit counter matches the value of the CCRm buffer register.

- When TQnCCRm register is used as capture register

The value of the 16-bit counter is saved to the TQnCCRm register upon TIQnm pin edge detection.

Remark: n = 0 to 1
m = 0 to 3

Figure 8-28: Flowchart of Basic Operation in Free-Running Mode


START
Initial setting
• Select clock.
(TQnCTL0: TQnCKS2 to TQnCKS0)
• Set free-running mode.
(TQnCTL1: TQnMD2 to TQnMD0 = 101)
Set TQnCCSm.
TQnCCSm = 0
(Compare)
TQnCCSm = 1
(Capture)
Enable timer operation (TQnCE = 1)
→ Transfer value of TQnCCRm
to CCRm buffer register
CCRm buffer register
matches 16-bit counter.
16-bit counter
overflows.
Set detection of edge of TIQnm
(TQnIOC1 register[Note]).
Enable timer operation (TQnCE = 1)
Edge of TIQnm is detected.
Value of 16-bit counter is
captured to TQnCCRm.
16-bit counter
overflows.

Note: TQCCR0 edge detection: TQnIS1 and TQnIS0 bits
TQCCR1 edge detection: TQnIS3 and TQnIS2 bits
TQCCR2 edge detection: TQnIS5 and TQnIS4 bits
TQCCR3 edge detection: TQnIS7 and TQnIS6 bits

Remark: n = 0 to 1, m = 0 to 3

(1) When TQnCCSn = 0 setting (compare function)

When TQnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH, and continues counting up in the free-running mode until TQnCE is cleared to 0. If a value is written to the TQnCCRm register in this mode, it is transferred to the CCRm buffer registers (anytime write). Even if an one-shot pulse trigger is input in this mode, an one-shot pulse is not generated. If TQnOEm is set to 1, TOQnm produces a toggle output when the value of the 16-bit counter matches the value of the CCRm buffer register.

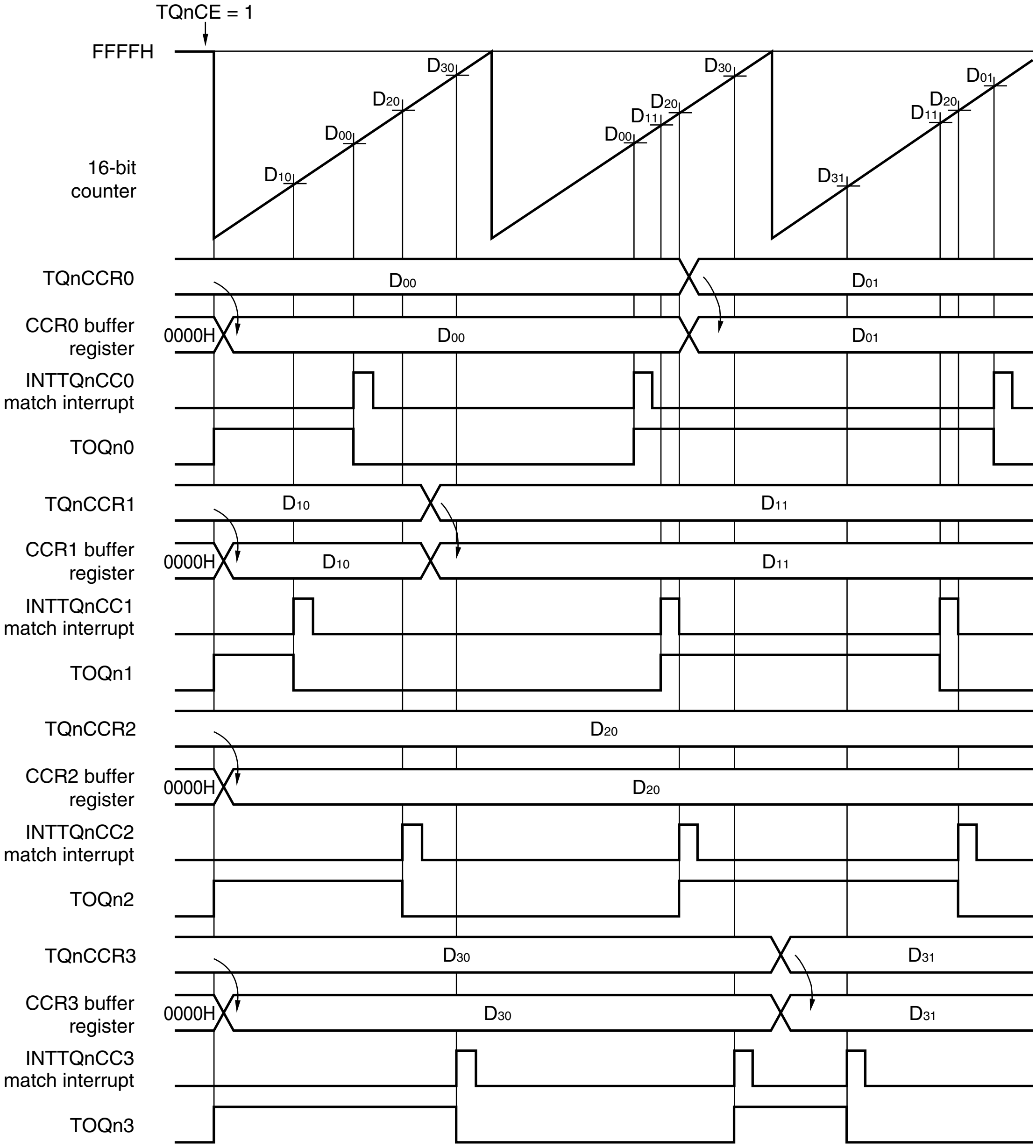
(2) When TQnCCSn = 1 setting (capture function)

When TQnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH, and continues counting up in the free-running mode until TQnCE is cleared to 0. The value captured by a capture trigger is written to the TQnCCRm registers.
Capturing before and after overflow (FFFFH) is judged using the overflow flag (TQnOVF).
However, if the interval of the capture trigger is such that the overflow occurs two times (two periods of more of free-running), the TQnOVF flag cannot be used for judgment.

Remark: n = 0 to 1
m = 0 to 3

Figure 8-29: Basic Operation Timing in Free-Running Mode (1/4)

(a) (TQnCCS3 = 0, TQnCCS2 = 0, TQnCCS1 = 0, TQnCCS0 = 0)



Remarks: 1. D00, D01: Setting values of TQnCCR0 register (0000H to FFFFH)
D10, D11: Setting values of TQnCCR1 register (0000H to FFFFH)
D20: Setting value of TQnCCR2 register (0000H to FFFFH)
D30, D31: Setting values of TQnCCR3 register (0000H to FFFFH)

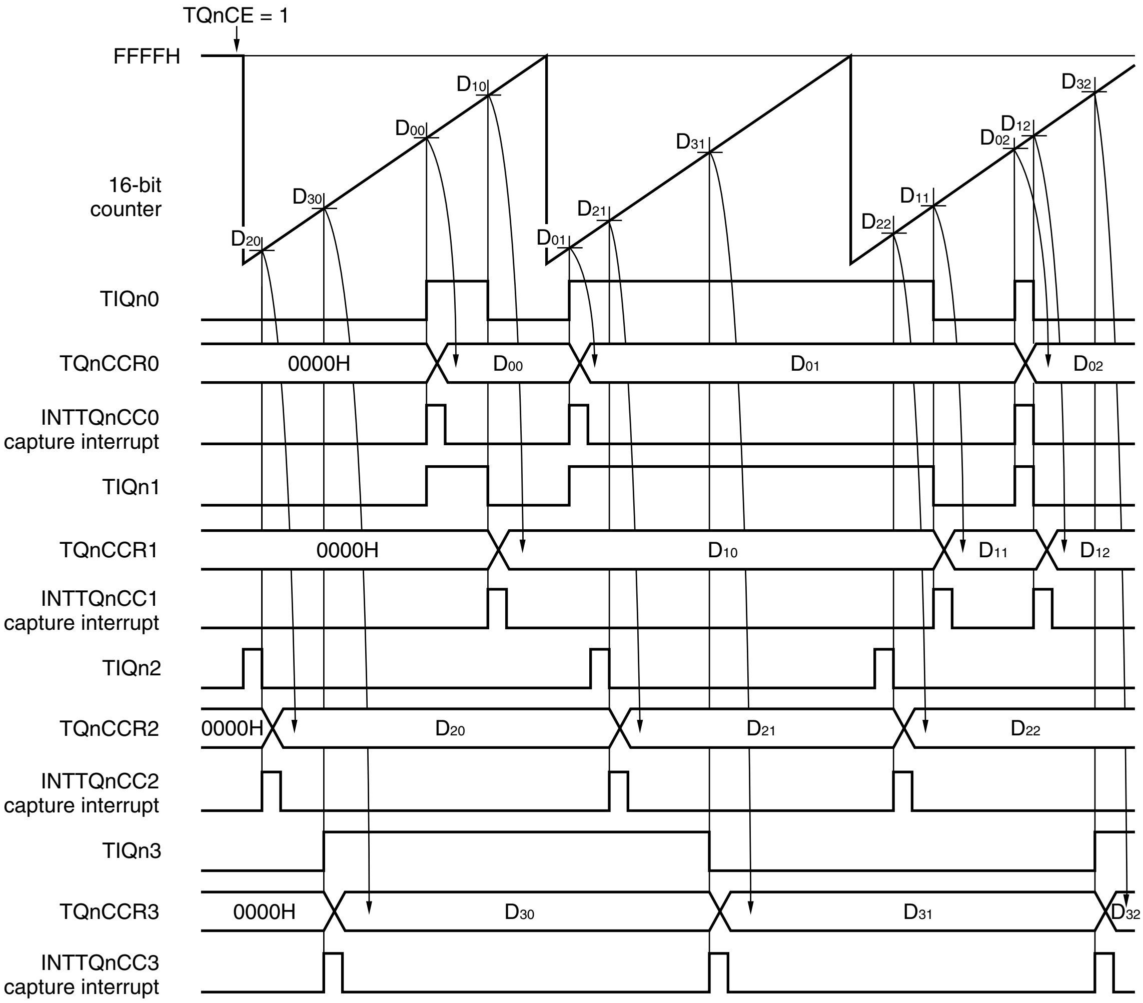
2. Toggle width of TOQnm output = (Set value of TQnCCRm register) × (Count clock cycle)

3. TOQnm output goes high when counting is started.

4. n = 0 to 1, m = 0 to 3

Figure 8-29: Basic Operation Timing in Free-Running Mode (2/4)

(b) (TQnCCS3 = 1, TQnCCS2 = 1, TQnCCS1 = 1, TQnCCS0 = 1)



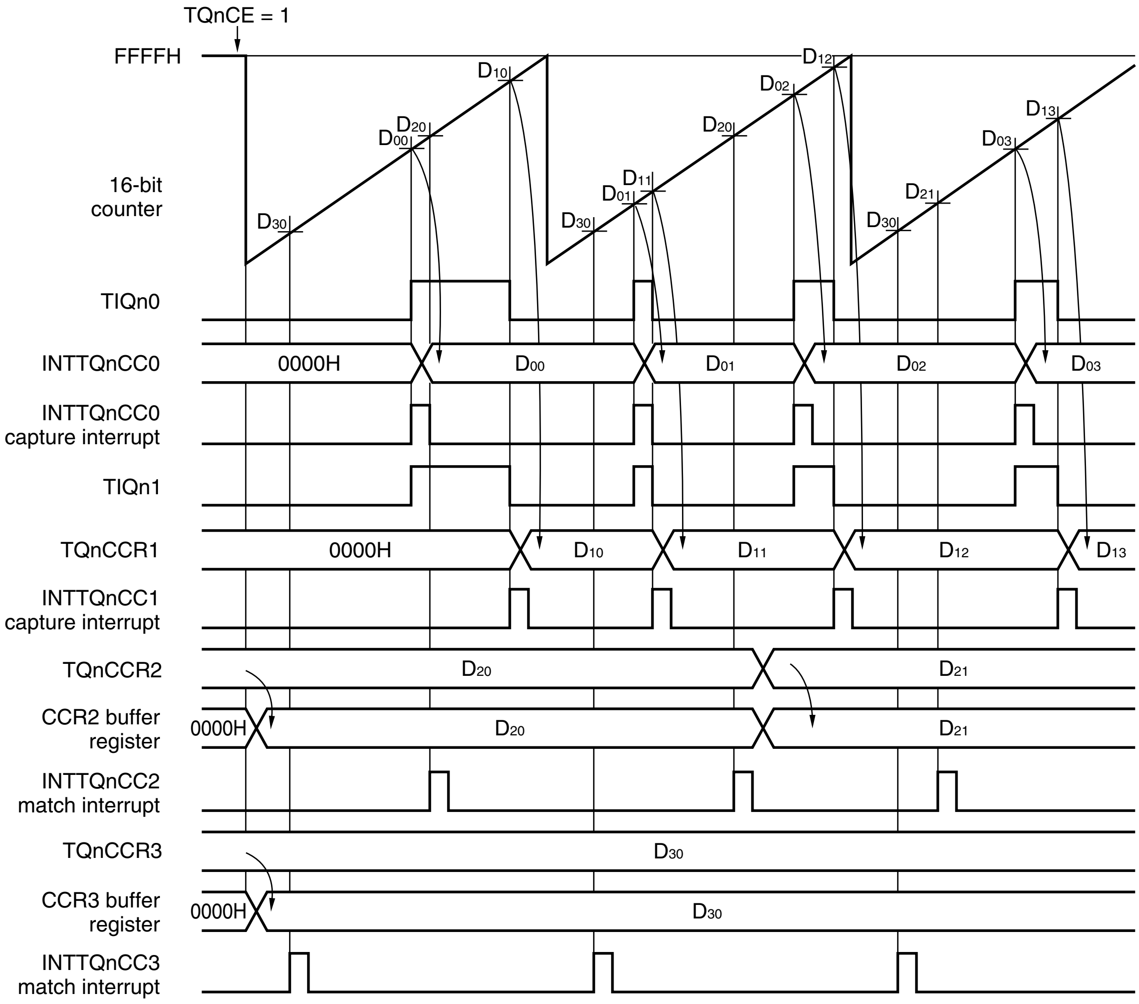
Remarks: 1. D00, D01, D02: Values captured to TQnCCR0 register (0000H to FFFFH)
D10, D11, D12: Values captured to TQnCCR1 register (0000H to FFFFH)
D20, D21, D22: Values captured to TQnCCR2 register (0000H to FFFFH)
D30, D31, D32: Values captured to TQnCCR3 register (0000H to FFFFH)

2. TIQn0: Set to rising edge detection (TQnIS1, TQnIS0 = 01)
TIQn01: Set to falling edge detection (TQnIS3, TQnIS2 = 10)
TIQn2: Set to falling edge detection (TQnIS5, TQnIS4 = 10)
TIQn3: Set to detection of both rising and falling edges (TQnIS7, TQnIS6 = 11)

3. n = 0 to 1

Figure 8-29: Basic Operation Timing in Free-Running Mode (3/4)

(c) (TQnCCS3 = 1, TQnCCS2 = 1, TQnCCS1 = 1, TQnCCS0 = 0)



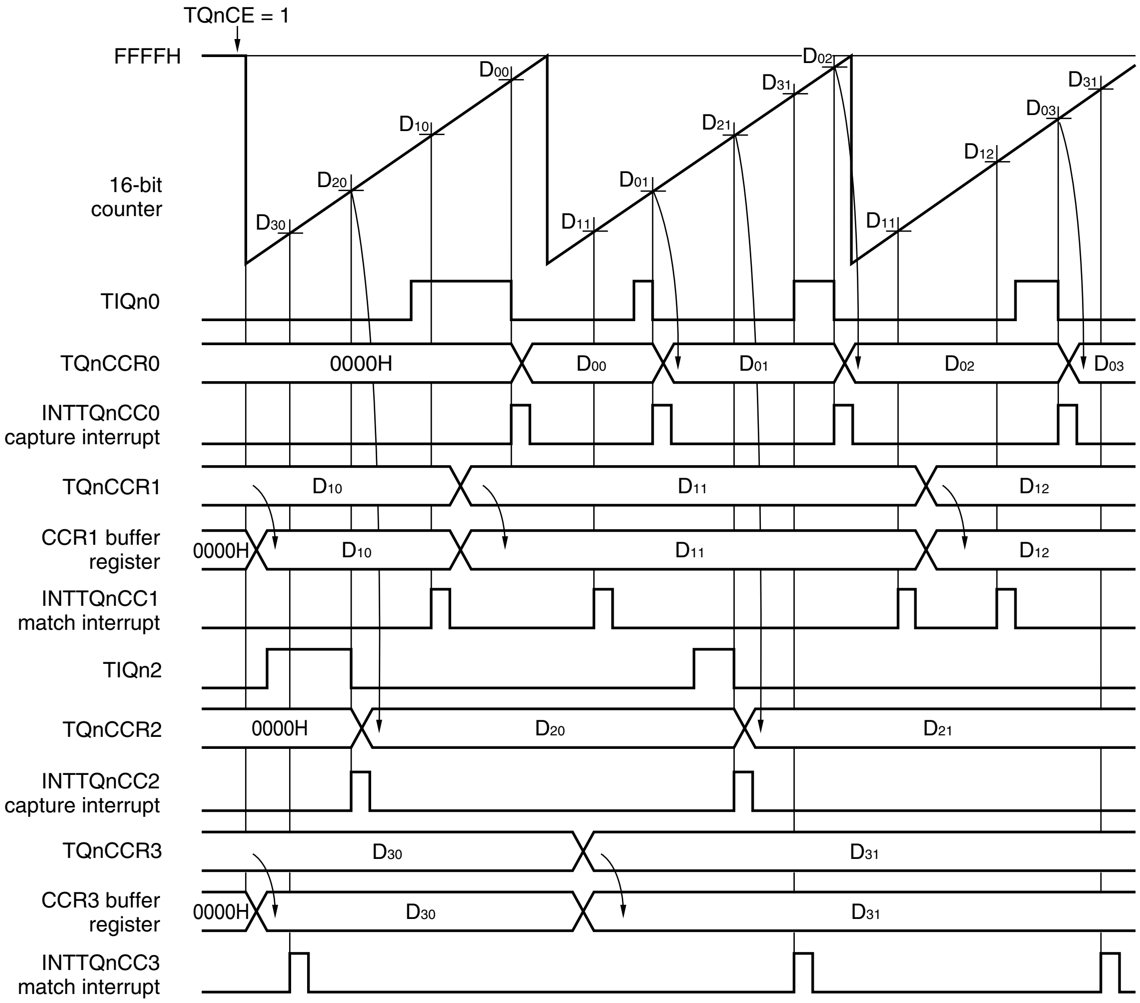
Remarks: 1. D00, D01, D02, D03: Values captured to TQnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Values captured to TQnCCR1 register (0000H to FFFFH)
D20, D21: Setting values of TQnCCR2 register (0000H to FFFFH)
D30: Setting value of TQnCCR3 register (0000H to FFFFH)

2. TIQn0: Set to rising edge detection (TQnIS1, TQnIS0 = 01)
TIQn1: Set to falling edge detection (TQnIS3, TQnIS2 = 10)

3. n = 0 to 1

Figure 8-29: Basic Operation Timing in Free-Running Mode /4/4)

(d) (TQnCCS3 = 0, TQnCCS2 = 1, TQnCCS1 = 0, TQnCCS0 = 1)



Remarks: 1. D00, D01, D02, D03: Values captured to TQnCCR0 register (0000H to FFFFH)
D10, D11, D12: Setting values of TQnCCR1 register (0000H to FFFFH)
D20, D21: Values captured to TQnCCR2 register (0000H to FFFFH)
D30, D31: Setting values of TQnCCR3 register (0000H to FFFFH)

2. TIQn0: Set to falling edge detection (TQnIS1, TQnIS0 = 10)
TIQn2: Set to falling edge detection (TQnIS5, TQnIS4 = 10)

3. n = 0 to 1

(3) Overflow flag

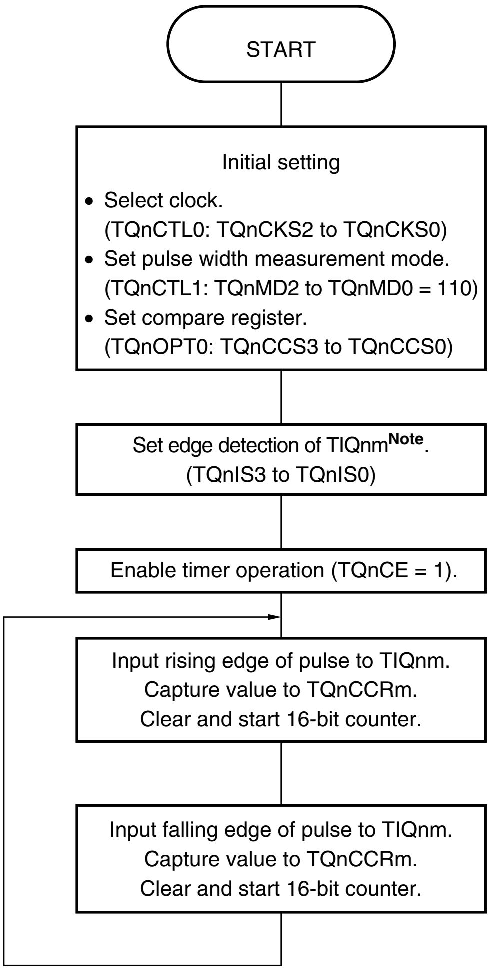
When the counter overflows from FFFFH to 0000H in the free-running mode, the overflow flag (TQnOVF) is set to 1 and an overflow interrupt (INTTQnOV) is output.
The overflow flag is cleared by the CPU writing 0 to it.

8.5.8 Pulse width measurement mode (TQnMD2 to TQnMD0 = 110)

In the pulse width measurement mode, free-running counting is performed. The value of the 16-bit counter is captured to capture register m (TQnCCRm) when both the rising and falling edges of the TIQnm pin are detected, and the 16-bit counter is cleared to 0000H. In this way, the external input pulse width can be measured.
To measure a long pulse width that exceeds the overflow of the 16-bit counter, use the overflow flag for detection. A pulse width that causes overflow to occur twice or more cannot be measured. Adjust the operating frequency of the 16-bit counter.

Caution: In the pulse width measurement mode, select the internal clock (TQnEEE of TQnCTL1 register = 0) as a count clock.

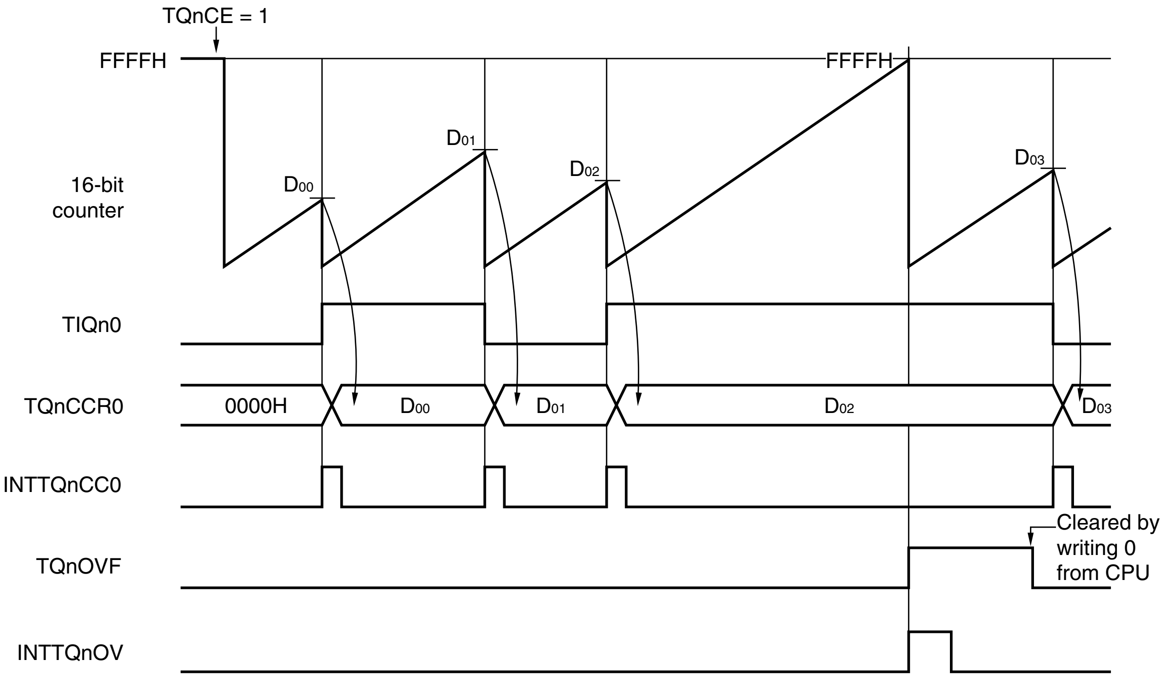
Figure 8-30: Flowchart of Basic Operation in Pulse Width Measurement Mode


START
Initial setting
• Select clock.
(TQnCTL0: TQnCKS2 to TQnCKS0)
• Set pulse width measurement mode.
(TQnCTL1: TQnMD2 to TQnMD0 = 110)
• Set compare register.
(TQnOPT0: TQnCCS3 to TQnCCS0)
Set edge detection of TIQnm^Note.
(TQnIS3 to TQnIS0)
Enable timer operation (TQnCE = 1).
Input rising edge of pulse to TIQnm.
Capture value to TQnCCRm.
Clear and start 16-bit counter.
Input falling edge of pulse to TIQnm.
Capture value to TQnCCRm.
Clear and start 16-bit counter.

Caution: An external pulse can be input from any of TIQn0 to TIQn3 but only one of them can be used. Specify that both the rising and falling edges are detected. Specify that the input edge of an external pulse input that is not used is not detected.

Remark: n = 0 to 1, m = 0 to 3

Figure 8-31: Basic Operation Timing in Pulse Width Measurement Mode



Remarks: 1. D00, D01, D02, D03: Values captured to TQnCCR0 register (0000H to FFFFH)

2. TIQn0: Set to detection of both rising and falling edges

3. n = 0 to 1

8.6 Timer Synchronization Operation Function

Timer P and timer Q have a timer synchronized operation function (tuned operation mode).
The timers that can be synchronized are listed in Table 8-3 (√□: Settable, ×: Not settable).

Table 8-3: Tuned Operation Mode of Timer

Master Timer	Slave Timer		V850E/RS1
TMP0	TMP1	-	√
TMp2	TMP3	TMQ0	√

Cautions: 1. The tuned operation mode is enabled or disabled by the TPnSYE bit of the TPnCTL1 register and TQnSYE bit of the TQnCTL1 register. For TMP2, either or both TMP3 and TMQ0 can be specified as slaves.

2. Set the tuned operation mode using the following procedure.

- **Set the TPnSYE bit of the TPnCTL1 register and the TQnSYE bit of the TQnCTL1 register of the slave timer to enable the tuned operation. Set the TPnMD2 to TPnMD0 bits of the TPnCTL1 register and TPnMD2 to TPnMD0 bits of the TQnCTL1 register of the slave timer to the free-running mode.**
- **Set the timer mode by using the TPnMD2 to TPnMD0 bits of the TPnCTL1 register and the TPnMD2 to TPnMD0 bits of the TQnCTL1 register. At this time, do not set the TPnSYE bit of the TPnCTL1 register and the TQnSYE bit of the TQnCTL1 register of the master timer.**
- **Set the compare register value of the master and slave timers.**
- **Set the TPnCE bit of the TPnCTL0 register and the TQnCE bit of the TQnCTL0 register of the slave timer to enable operation on the internal operating clock.**
- **Set the TPnCE bit of the TPnCTL0 register and the TQnCE bit of the TQnCTL0 register of the master timer to enable operation on the internal operating clock.**

Tables 8-4 and 8-5 show the timer modes that can be used in the tuned operation mode
(√□: Settable, ×: Not settable).

Table 8-4: Timer Modes Usable in Tuned Operation Mode

Master Timer	Free-Running Mode	PWM Mode	Triangular Wave PWM Mode
TMP0	√	√	×
TMP2	√	√	×
TMQ1	√	√	√

Table 8-5: Timer Output Functions

Tuned Channel	Timer	Pin	Free-Running Mode		PWM Mode		Triangular Wave PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
Ch0	TMP0 (master)	TOP00	PPG	←	Toggle	←	N/A	←
		TOP01	PPG	←	PWM	←	N/A	←
	TMP1 (slave)	TOP10	PGP	←	Toggle	PWM	N/A	←
		TOP11	PPG	←	PWM	←	N/A	←
Ch1	TMP2 (master)	TOP20	PPG	←	Toggle	←	N/A	←
		TOP21	PPG	←	PWM	←	N/A	←
	TMP3 (slave)	TOP30	PPG	←	Toggle	PWM	N/A	←
		TOP31	PPG	←	PWM	←	N/A	←
Ch1	TMQ0 (slave)	TOQ00	PPG	←	Toggle	PWM	Toggle	N/A
		TOQ01 to TOQ03	PPG	←	PWM	←	Triangular wave PWM	N/A
Ch2	TMQ1 (master)	TOQ10	PPG	←	Toggle	←	Toggle	←
		TOQ11 to TOQ13	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 TOPn0 and TOQm0 (n = 0 to 3, m = 0 to 1)

Figure 8-32: Tuned Operation Image (TMP2, TMP3, TMQ0)

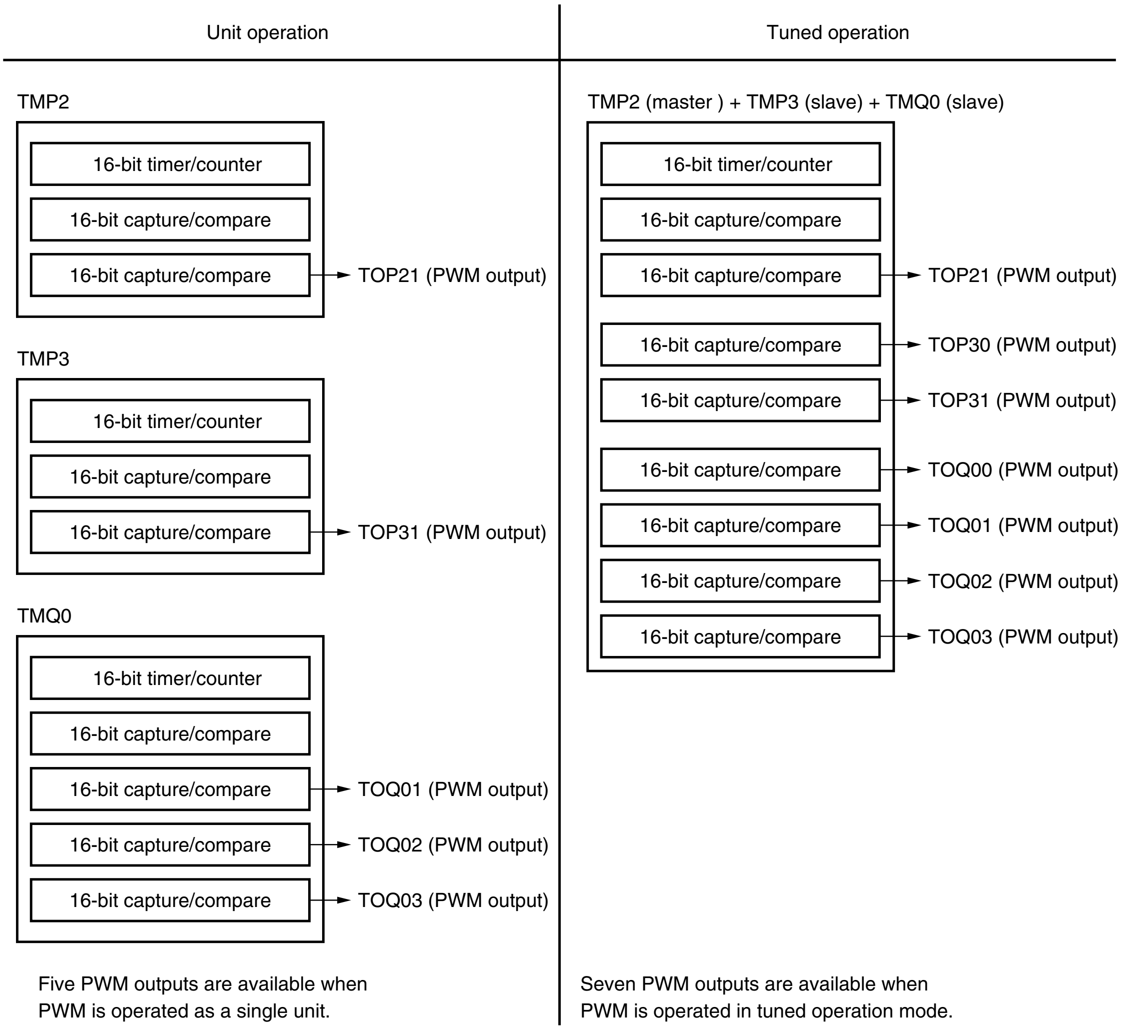
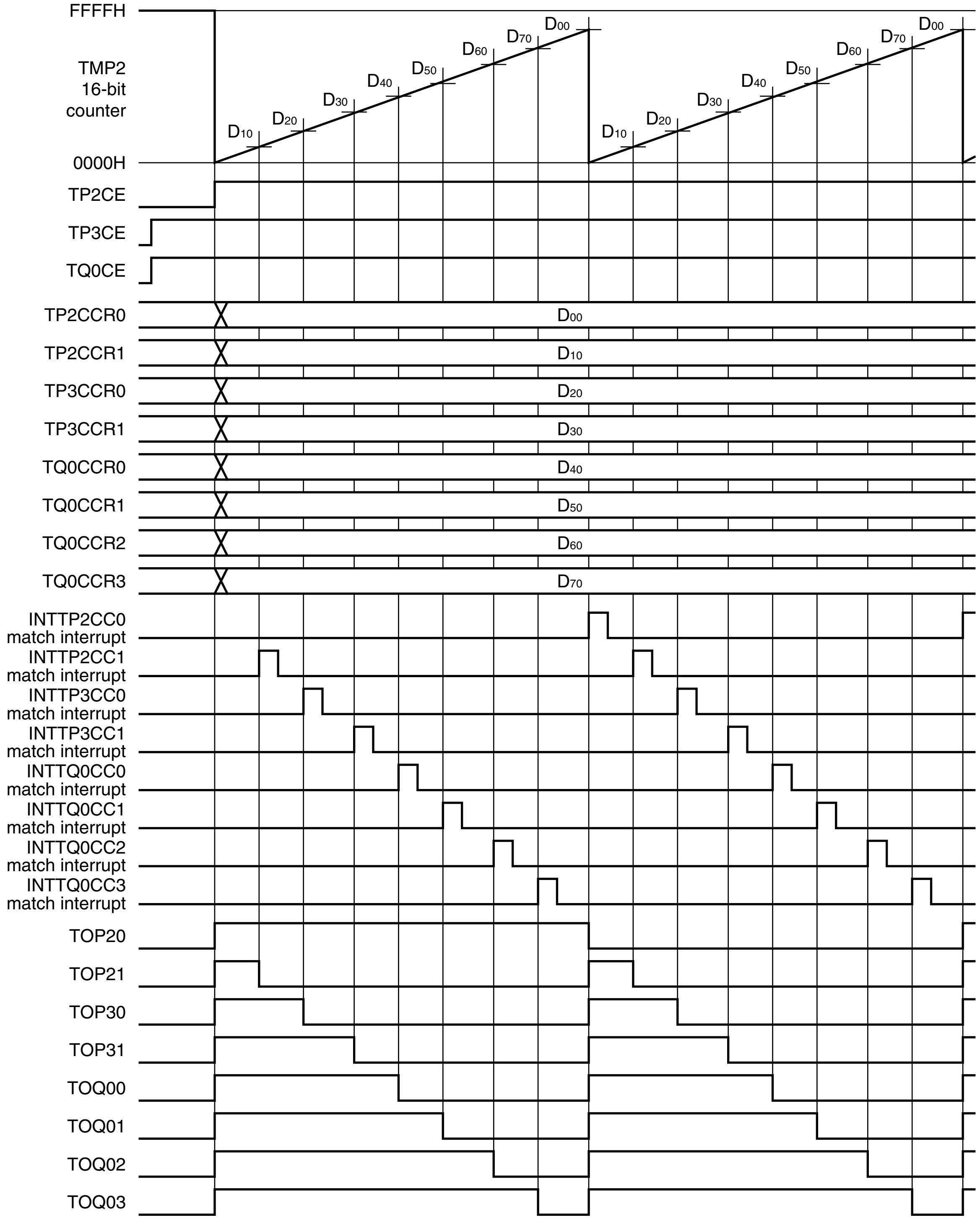


Figure 8-33: Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)


FFFFH
TMP2 16-bit counter
0000H
D10
D20
D30
D40
D50
D60
D70
D00
TP2CE
TP3CE
TQ0CE
TP2CCR0
TP2CCR1
TP3CCR0
TP3CCR1
TQ0CCR0
TQ0CCR1
TQ0CCR2
TQ0CCR3
INTTP2CC0 match interrupt
INTTP2CC1 match interrupt
INTTP3CC0 match interrupt
INTTP3CC1 match interrupt
INTTQ0CC0 match interrupt
INTTQ0CC1 match interrupt
INTTQ0CC2 match interrupt
INTTQ0CC3 match interrupt
TOP20
TOP21
TOP30
TOP31
TOQ00
TOQ01
TOQ02
TOQ03

Chapter 9 16-Bit Interval Timer M

The V850E/RS1 includes one 16-bit interval timer M (TMM0)

9.1 Features

Timer M (TMM0) supports only a clear & start mode. It does not support a free-running mode. To use timer M in a manner equivalent to in the free-running mode, set the compare register to FFFFH and start the 16-bit counter. A match interrupt will occur when the timer overflows.

- Interval function
- 8 clocks selectable
- Simple counter × 1
  (The simple counter is a counter that does not use a counter read buffer and the counter cannot be read during timer count operation.)
- Simple compare × 1
  (Simple compare is a type of compare that does not use a compare write buffer and the compare register cannot be written during timer counter operation.)
- Compare match interrupt × 1

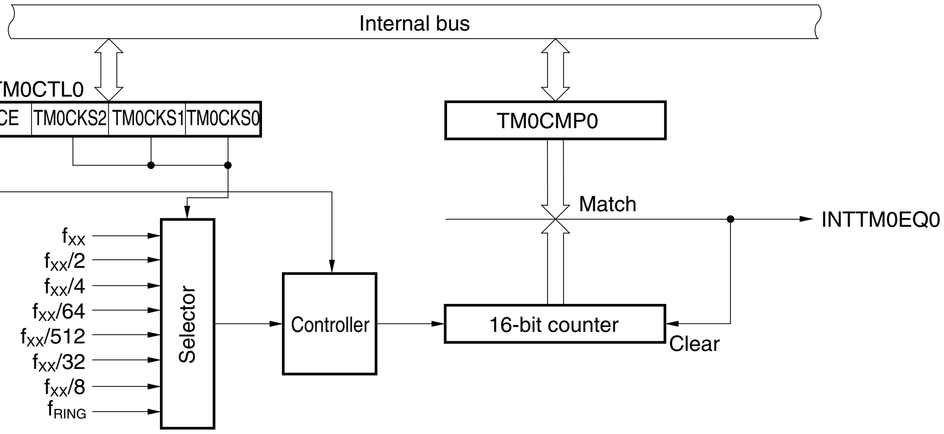
9.2 Configuration

TMM0 includes the following hardware.

Table 9-1: Configuration of TMM

Item	Configuration
Timer register	16-bit counter
Register	TMM0 compare register 0 (TM0CMP0)
Control register	TMM0 timer control register (TM0CTL0)

Figure 9-1: Block Diagram of TMM0



Remark: f_{XX}: Internal system clock frequency
f_R: Ring-OSC clock frequency

(1) 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.
$\overline{\text{RESET}}$ input clears this register to 0000H.
The same value can always be written to the TM0CMP0 register by software.
Rewriting the TM0CMP0 register is prohibited when the TM0CE bit = 1.

Figure 9-2: TMM0 Compare Register 0 (TMnCMP0) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
TM0CMP0																	FFFF F694H	R/W	0000H

9.3 Control Register

(1) TMM0 timer control register (TM0CTL0)

The TMM0 timer control register (TM0CTL0) is an 8-bit register used to control the timer operation.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

The same value can always be written to the TM0CTL0 register by software.

Figure 9-3: TMM0 Timer Control Register (TM0CTL0) Format

Symbol	<7>	6	5	4	3	2	1	0	Address	R/W	After reset
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0	FFFFF690H	R/W	00H

<table>
<tr><th>TM0CE</th><th>Internal clock operation enable/disable specification</th></tr>
<tr><td>0</td><td>Disable internal operating clock operation (asynchronously reset TMM0).</td></tr>
<tr><td>1</td><td>Enable internal operating clock operation.</td></tr>
<tr><td colspan="2">The TM0CE bit controls the internal operating clock and asynchronously resets TMM0.
When this bit is cleared to 0, the internal operating clock of TMM is stopped (fixed to the low level), and TMM0 is asynchronously reset.
When the TM0CE bit is set to 1, the internal operating clock is enabled within two input clocks, and the timer counts up.</td></tr>
</table>

TM0CKS2	TM0CKS1	TM0CKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/64$
1	0	0	$f_{XX}/512$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/8$
1	1	1	f_{RING}

Caution: Bits TM0CKS2 to TM0CKS0 can be rewritten when TM0CE = 0.
However, bits TM0CKS2 to TM0CKS0 and bit TM0CE are mapped to the same register.
Therefore, when changing the value of TM0CE from 0 to 1, it is also possible to change the value of bits TM0CKS2 to TM0CKS0.

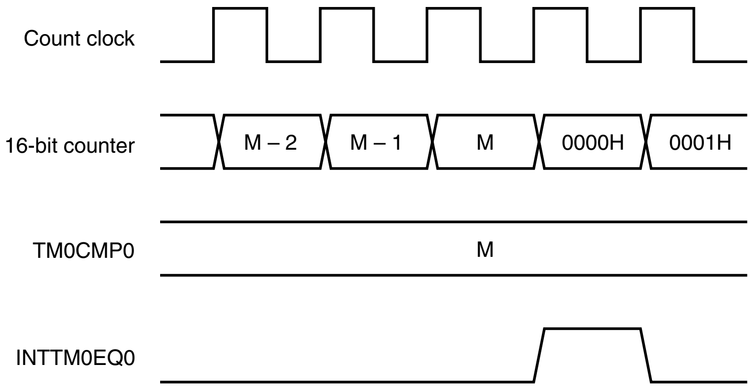
Remark: f_{XX}: Internal system clock frequency
f_{RING}: Ring-OSC frequency

9.4 Operation

9.4.1 Interval timer mode

In the interval timer mode, a match interrupt signal (INTTM0EQ0) is output when the value of the 16-bit counter matches the value of TMM0 compare register 0 (TM0CMP0). At the same time, the counter is cleared to 0000H and starts counting up.
When FFFFH is set to the TM0CMP0 register, timer M performs an operation similar to that in the free-running mode.

Figure 9-4: Interval Timer Mode Timing

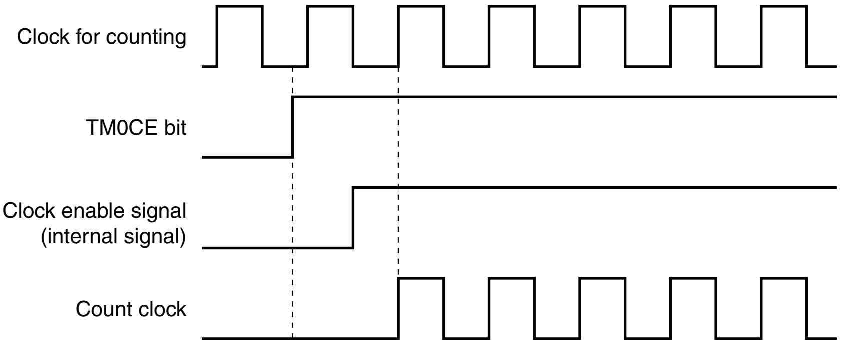


Caution: To set the interval time to M clocks, set M - 1 to the TM0CMP0 register.

9.4.2 Clock generator and clock enable timing

Because the second clock is the first pulse of the timer count-up signal when the TM0CE bit is changed from 0 to 1, the timer counts one clock less.

Figure 9-5: Count Operation Start Timing



Chapter 10 Functions of Watchdog Timer 2

10.1 Functions

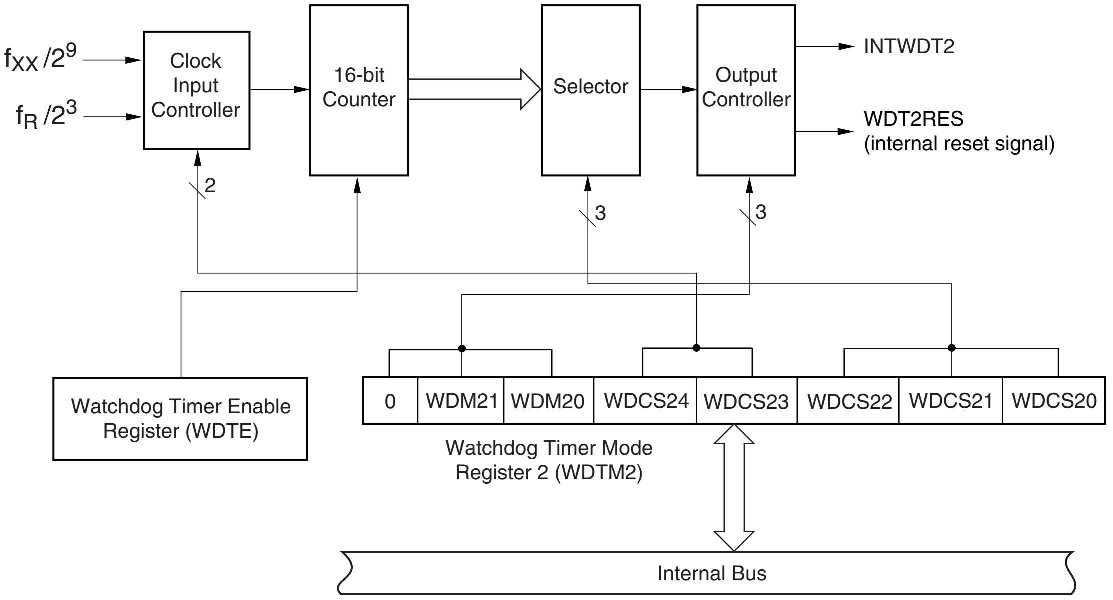
Watchdog Timer 2 has the following functions.

- 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 oscillator and ring oscillator 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 through this function, or clear Watchdog timer 2 once and stop it within the next interval time.
Also, perform write for verification purposes only once, even if the default settings (reset mode, interval time: $f_R/2^{19}$) need not be changed.

2. Restoring using the RETI instruction following non-maskable interrupt servicing due to a non-maskable interrupt request (INTWDT2) is not possible. Therefore, following completion of interrupt servicing, perform a system reset.

Figure 10-1: Block Diagram of Watchdog Timer 2



Remark: f_{XX}:	PLL output oscillation frequency
f_R:	Ring-OSC clock frequency
INTWDT2:	Non-maskable interrupt request signal from Watchdog Timer 2
WDT2RES:	Watchdog Timer 2 reset signal

10.2 Configuration

Watchdog Timer 2 consists of the following hardware.

Table 10-1: Configuration of Watchdog Timer 2

Item	Configuration
Control registers	Oscillation stabilization time select register (OSTS) Watchdog timer mode register 2 (WDTM2) Watchdog timer enable register (WDTE)

10.3 Control Registers

(1) Oscillation stabilization time select register (OSTS)

The OSTS register selects the oscillation stabilization time following reset or release of the stop mode.

This register is set by an 8-bit manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 03H.

Figure 10-2: Oscillation Stabilization Time Select Register (OSTS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFFF6C0H	03H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time Note 1	
				f_X Note 2
			Selected clock	8 MHz
0	0	0	$2^{10}/f_X$	0.128 ms
0	0	1	$2^{11}/f_X$	0.256 ms
0	1	0	$2^{12}/f_X$	0.512 ms
0	1	1	$2^{13}/f_X$	1.024 ms
1	0	0	$2^{14}/f_X$	2.048 ms
1	0	1	$2^{15}/f_X$	4.096 ms
1	1	0	$2^{16}/f_X$	8.192 ms
1	1	1	Setting prohibited	

Notes: 1. The oscillation stabilization time and setup time are required when the software stop mode and idle mode are released, respectively.

2. f_X: Main clock oscillator frequency

(2) 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 or 1-bit units. This register can be read any number of times, but it can be written only once following reset release. Writing to this register will automatically clear the Watchdog timer counter to 0000H.

$\overline{\text{RESET}}$ input sets this register to 67H.

Figure 10-3: Watchdog Timer Mode Register 2 (WDTM2) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	FFFFF6D0H	67H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

WDM21	WDM20	Selection of operation mode of Watchdog Timer 2
0	0	Stops operation
0	1	Non-maskable interrupt request mode (generation of INTWDT2)
1	-	Reset mode (generation of WDTRES2)

Cautions: 1. For details of bits WDCS20 to WDCS24, refer to Table 10-2, “Watchdog Timer 2 Clock Selection,” on page 362.

2. Although Watchdog timer 2 can be stopped just by stopping the operation of Ring-OSC, set the WDTM2 register to 1FH to securely stop the timer (to avoid selection of the main clock due to an erroneous write operation).

3. If the WDTM2 register is rewritten twice after reset, an overflow signal is forcibly generated. But, the overflow signal does not occur, even if the WDTM2 register is written twice after the watch dog timer is suspended.

4. To stop the operation of Watchdog timer 2, set the RSTP bit of the RCM register to 1 (to stop Ring-OSC) and the WDTM2 register to 1FH.

Table 10-2: Watchdog Timer 2 Clock Selection

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected clock	100 kHz (MIN.)	200 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2621.4 ms	1310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5242.9 ms	2621.47 ms	1310.7 ms
						f_{XX} = 24 MHz	f_{XX} = 32 MHz	f_{XX} = 40 MHz
0	1	0	0	0	$2^{18}/f_{XX}$	10.9 ms	8.19 ms	6.6 ms
0	1	0	0	1	$2^{19}/f_{XX}$	21.8 ms	16.4 ms	13.1 ms
0	1	0	1	0	$2^{20}/f_{XX}$	43.7 ms	32.8 ms	26.2 ms
0	1	0	1	1	$2^{21}/f_{XX}$	87.4 ms	65.5 ms	52.2 ms
0	1	1	0	0	$2^{22}/f_{XX}$	174.8 ms	131.1 ms	104.9 ms
0	1	1	0	1	$2^{23}/f_{XX}$	349.5 ms	262.1 ms	209.7 ms
0	1	1	1	0	$2^{24}/f_{XX}$	699.1 ms	524.3 ms	419.4 ms
0	1	1	1	1	$2^{25}/f_{XX}$	1398.1 ms	1048.6 ms	838.9 ms

(3) Watchdog timer enable register (WDTE)

The counter of the Watchdog timer is cleared and counting restarted by writing “ACH” to WDTE. WDTE is set by an 8-bit memory manipulation instruction.
RESET Input sets this register to 9AH.

Figure 10-4: Watchdog Timer Enable Register (WDTE) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
WDTE	RUN2								FFFFF6D1H	9AH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

RUN2	RUN2 Selection of Watchdog timer operation mode[Note]
0	Counting stopped
1	Counter cleared and counting started

Note: Once RUN2 is set to 1 it cannot be cleared to 0 by software. Therefore, counting can be stopped only by RESET input after counting is started.

Cautions: 1. When a value other than “ACH” is written to the WDTE register, an overflow signal is forcibly output when “RUN2” bit was previously set to 1.

2. When an 1-bit memory manipulation instruction is executed for the WDTE register, an overflow signal is forcibly output (an error results in the assembler).

3. The read value of the WDTE register is “9AH” (which differs from written value “ACH”).

10.4 Operation

(1) Oscillation stabilization time selection function

The wait time until the oscillation stabilizes after the software STOP mode is released is controlled by the OSTS register.
The OSTS register can be read or written 8-bit units.
$\overline{\mathrm{RESET}}$ input sets this register to 03H.

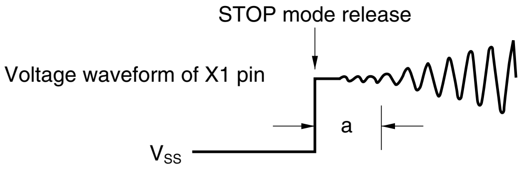
Figure 10-5: Oscillation Stabilization Time Selection Function

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFFF6C0H	03H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time Note	
				f_X
			Selected clock	8 MHz
0	0	0	$2^{10}/f_X$	0.128 ms
0	0	1	$2^{11}/f_X$	0.256 ms
0	1	0	$2^{12}/f_X$	0.512 ms
0	1	1	$2^{13}/f_X$	1.024 ms
1	0	0	$2^{14}/f_X$	2.048 ms
1	0	1	$2^{15}/f_X$	4.096 ms
1	1	0	$2^{16}/f_X$	8.192 ms
1	1	1	Setting prohibited	

Note: The oscillation stabilization time and setup time are required when the software stop mode and idle mode are released, respectively.

Cautions: 1. The wait time following release of the software STOP mode does not include the time until the clock oscillation starts ("a" in the figure below) following release of the software STOP mode, regardless of whether the software STOP mode is released by $\overline{\mathrm{RESET}}$ input or the occurrence of an interrupt request signal.



2. Be sure to clear bits 3 to 7 to 0.

3. The oscillation stabilization time following reset release is $2^{13}/f_X$ (because the initial value of the OSTS register = 03H).

Remark: f_X = Main clock oscillator frequency

(2) Operation as Watchdog timer 2

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 WDM20 bits of the WDTM2 register.
To not use Watchdog timer 2, write 1FH to the WDTM2 register.
If the non-maskable interrupt request mode has been set, restoring using the RETI instruction following non-maskable interrupt servicing is not possible. Therefore, following completion of interrupt servicing, perform a system reset.

Caution: If the set to WDTM2, WDM21 bit = 1 (reset mode), WDT overflow occurred in oscillation stabilization time, after reset or standby release, internal reset is not occurred, CPU clock changing for RING-OSC.

[MEMO]

Chapter 11 A/D Converter

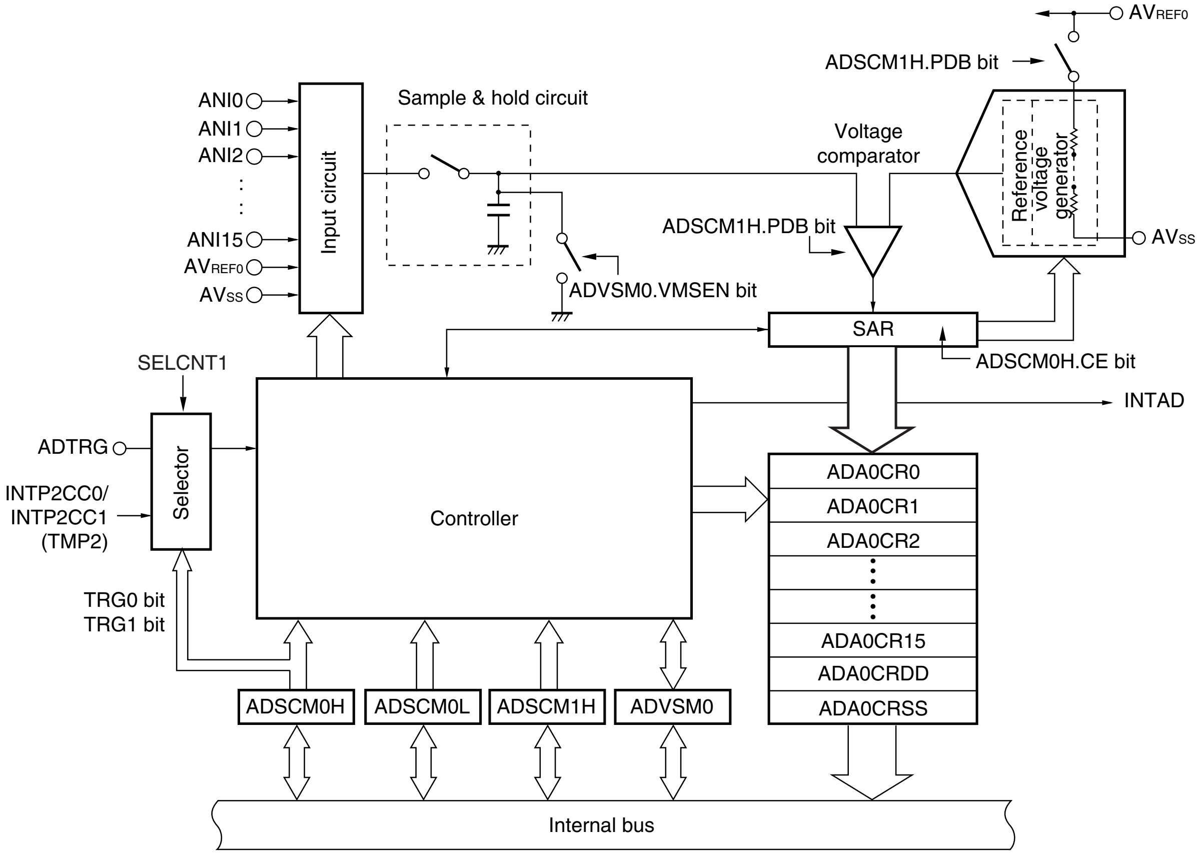
11.1 Functions

The A/D converter converts analog input signals into digital values, has a resolution of 10 bits, and can handle 16 channels of analog input signals (ANI0 to ANI15).
The A/D converter has the following features.

- 10-bit resolution
- 16 channels
- Successive approximation method
- Operating voltage: AV_{REF0} = 4.5 to 5.5 V
- Analog input voltage: AV_{SS} to AV_{REF0}
- The following functions are provided as operation modes.
  - Continuous select mode
  - Continuous scan mode
- The following functions are provided as trigger modes.
  - Software trigger mode
  - External trigger mode
  - Timer trigger mode
- The following extended functions are included
  - Diagnostic mode
  - Discharge operation
- High speed conversion

The block diagram of the A/D converter is shown below.

Figure 11-1: Block Diagram of A/D Converter



11.2 Configuration

The A/D converter includes the following hardware.

Table 11-1: Configuration of A/D Converter

Item	Configuration
Analog inputs	16 channels (ANI0 to ANI15)
Registers	Successive approximation register (SAR) A/D conversion result registers 0 – 15 (ADA0CRn) A/D diagnostic mode conversion result registers (ADA0CRDD, ADA0CRSS)
Control registers	A/D converter mode registers 0 to 1 (ADSCM0H, ADSCM0L, ADSCM1H) A/D converter extended mode control register (ADVMS0) A/D external trigger selection register (SELCNT1)

(1) Successive approximation register (SAR)

This register compares the voltage value of the analog input signal with the voltage tap (compare voltage) value from the series resistor string, 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 has been completed), the contents of the SAR register are transferred to the A/D conversion result ADA0CRn register.

Remark: n = 0 to 15

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals sequentially sent from the input circuit and sends the sampled data to the voltage comparator. This circuit also holds the sampled analog input signal voltage during A/D conversion.

(3) Voltage comparator

The voltage comparator compares a voltage value that has been sampled and held with the voltage value of the series resistor string.

(4) Series resistor string

This series resistor string is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(5) ANI0 to ANI15 pins

These are analog input pins for the 16 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 analog input with the analog input channel specification register (ADSCM0L) can be used as input port pins.

Cautions: 1. Make sure that the voltages input to ANI0 to ANI15 do not exceed the rated values. In particular if a voltage higher than AV$_{REF0}$ 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. Analog input (ANI0 to ANI15) pins are alternate function pins that can also be used as input port (P70 to P715) pins. When A/D conversion is performed by selecting any one of ANI0 to ANI15, do not execute any input or output operation to port 7 during conversion. It may decrease the accuracy of conversion resolution.

(6) AV$_{REF0}$ pin

This is the pin used to input the reference voltage of the A/D converter. The signals input to the ANI0 to ANI15 pins are converted to digital signals based on the voltage applied between the AV$_{REF0}$ and AV$_{SS}$ pins.

(7) 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.

11.3 Control Registers

The A/D converter is controlled by the following registers.

- A/D conversion result register n (ADA0CRn, ADA0CRDD, ADA0CRSS)
- A/D converter mode registers 0, 1 (ADSCM0H, ADSCM0L, ADSCM1H)
- A/D converter extended mode control register 0 (ADVMS0)
- A/D converter external trigger control register (SELCNT1)

(1) A/D conversion result register n (ADA0CRn, ADA0CRDD, ADA0CRSS)

The ADA0CRn register is a 16-bit register that stores the A/D conversion result. ADA0CRn consist of 16 registers and the A/D conversion result is stored in the 10 lower bits of the register corresponding to analog input. The upper 6 bits are fixed to 0.

The ADA0CRn register is read with a 16-bit memory manipulation instruction. $\overline{\text{RESET}}$ input clears this register to 0000H.

Caution: A write operation to A/D converter mode register 0 (ADSCM0) and A/D converter mode register 1 (ADSCM1) may cause the contents of the ADA0CRn register to become undefined. After the conversion, read the conversion result before performing write to the ADSCMn registers. Correct conversion results may not be read if a sequence other than the above is used.

Figure 11-2: A/D Conversion Result Register n (ADA0CRn, ADA0CRDD, ADA0CRSS) Format (1/2)

(a) After $\overline{\text{RESET}}$: ADA0CRn = 0000H (n= 0 to 15), Read only registers

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
ADA0CR0	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF220H
ADA0CR1	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF222H
ADA0CR2	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF224H
ADA0CR3	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF226H
ADA0CR4	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF228H
ADA0CR5	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF22AH
ADA0CR6	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF22CH
ADA0CR7	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF22EH
ADA0CR8	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF230H
ADA0CR9	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF232H
ADA0CR10	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF234H
ADA0CR11	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF236H
ADA0CR12	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF238H
ADA0CR13	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF23AH
ADA0CR14	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF23CH
ADA0CR15	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF23EH

Figure 11-2: A/D Conversion Result Register n
(ADA0CRn, ADA0CRDD, ADA0CRSS) Format (2/2)

(b) After $\overline{\text{RESET}}$: ADA0CRDD, ADA0CRSS = 0000H, Read only registers

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
ADA0CRDD	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF240H
ADA0CRSS	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF242H

The relationship between the analog voltage input to an analog input pin (ANI0 to ANI15) and the value of the A/D conversion result register (ADA0CRn) is as follows (n = 0 to 15):

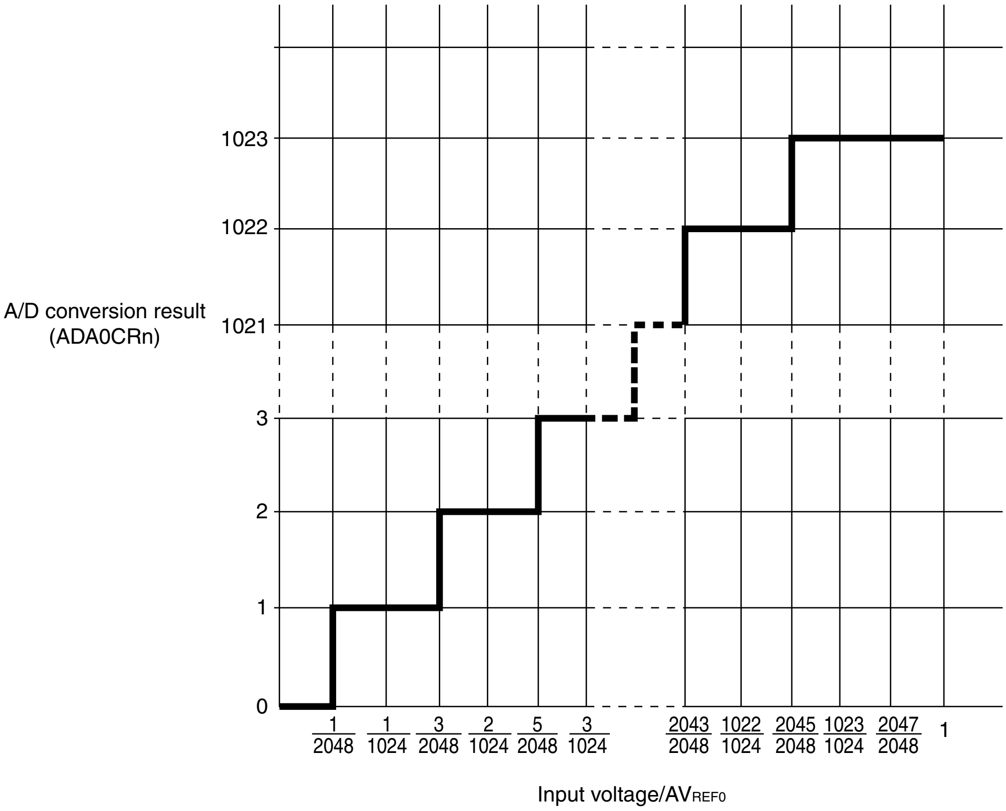
$$\text{ADA0CR} = \text{INT}\left(\frac{V_{IN}}{AV_{REF0}} \times 1{,}024 + 0.5\right)$$

or,

$$(\text{ADA0CR} - 0.5) \times \frac{AV_{REF0}}{1024} \leq V_{IN} < \text{ADA0CR} + 0.5) \times \frac{AV_{REF0}}{1024}$$

INT ( ): Function that returns integer of value in ( )
V_{IN}: Analog input voltage
AV_{REF0}: AV_{REF0} pin voltage
ADA0CR: Value of A/D conversion result register (ADA0CRn)

Figure 11-3: Relationship Between Analog Input Voltages and A/D Converter Results



Remark: n = 0 to 15

(2) A/D converter mode register 0H (ADSCM0H)

This is an 8-bit register used to specify the operating mode and to control conversion operation.
This register is accessed with an 8-bit or an 1-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 11-4: A/D Converter Mode Register 0H (ADSCM0H) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
ADSCM0H	CE	CS	0	MS	PLM	0	TRG1	TRG0	FFFFF201H	00H
R/W	R/W	R	R	R/W	R/W	R	R/W	R/W		

CE	A/D conversion enable bit
0	Stops conversion
1	Enables conversion

CS	A/D conversion status bit
0	A/D conversion is stopped
1	A/D conversion is executing

MS	A/D conversion mode selection bit
0	Scan mode
1	Select mode

PLM	A/D polling mode selection bit
0	A/D trigger mode
1	A/D polling mode (only with software trigger mode)

Trigger Mode	TRG1	TRG0	A/D Trigger mode
Software	0	0	A/D trigger mode (ADSCMOH.CE bit).
External	0	1	Select the external trigger using register SELCNT1 bit 0
	1	1	Select the external trigger using register SELCNT1 bit 1
Others than above			Setting prohibited

Caution: If ADSCM0H is written while CS = 1, the active conversion is aborted, and a new conversion operation is started.

(3) A/D converter mode register 0L (ADSCM0L)

This is an 8-bit register used to specify the analog input channel(s) for conversion.
This register is set with an 8-bit or an 1-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this bit to 00H.

Figure 11-5: A/D Converter Mode Register 0L (ADSCM0L) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
ADSCM0L	SANI3	SANI2	SANI1	SANI0	ANIS3	ANIS2	ANIS1	ANIS0	FFFFF200H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

SANI3	SANI2	SANI1	SANI0	Scan mode analog input start channel selector
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	0	1	1	ANI11
1	1	0	0	ANI12
1	1	0	1	ANI13
1	1	1	0	ANI14
1	1	1	1	Setting prohibited

Caution: If ADSCM0L is written while CS = 1, the active conversion result is not guaranteed and should be operated again. Do not write ADSCM0L while CS = 1.

Figure 11-5: A/D Converter Mode Register 0L (ADSCM0L) Format (2/2)

ANIS				Analog input channel selector			
				ADVMS0.DIAGEN=”0”		ADVMS0.DIAGEN=”1”	
3	2	1	0	Select mode (ADSCM0H.MS=1)	Scan mode conversion end channel (ADSCM0H.MS=0)	Select mode (ADSCM0H.MS=1)	Scan mode conversion end channel, including Diagnostic Function (ADSCM0H.MS=0)
0	0	0	0	ANI0	Setting prohibited	AV_{REF0}	Setting prohibited
0	0	0	1	ANI1	SANI → ANI1	AV_{SS}	SANI → ANI1 → AV_{REF0} → AV_{SS}
0	0	1	0	ANI2	SANI → ANI2	Setting prohibited	SANI → ANI2 → AV_{REF0} → AV_{SS}
0	0	1	1	ANI3	SANI → ANI3	Setting prohibited	SANI → ANI3 → AV_{REF0} → AV_{SS}
0	1	0	0	ANI4	SANI → ANI4	Setting prohibited	SANI → ANI4 → AV_{REF0} → AV_{SS}
0	1	0	1	ANI5	SANI → ANI5	Setting prohibited	SANI → ANI5 → AV_{REF0} → AV_{SS}
0	1	1	0	ANI6	SANI → ANI6	Setting prohibited	SANI → ANI6 → AV_{REF0} → AV_{SS}
0	1	1	1	ANI7	SANI → ANI7	Setting prohibited	SANI → ANI7 → AV_{REF0} → AV_{SS}
1	0	0	0	ANI8	SANI → ANI8	Setting prohibited	SANI → ANI8 → AV_{REF0} → AV_{SS}
1	0	0	1	ANI9	SANI → ANI9	Setting prohibited	SANI → ANI9 → AV_{REF0} → AV_{SS}
1	0	1	0	ANI10	SANI → ANI10	Setting prohibited	SANI → ANI10 → AV_{REF0} → AV_{SS}
1	0	1	1	ANI11	SANI → ANI11	Setting prohibited	SANI → ANI11 → AV_{REF0} → AV_{SS}
1	1	0	0	ANI12	SANI → ANI12	Setting prohibited	SANI → ANI12 → AV_{REF0} → AV_{SS}
1	1	0	1	ANI13	SANI → ANI13	Setting prohibited	SANI → ANI14 → AV_{REF0} → AV_{SS}
1	1	1	0	ANI14	SANI → ANI14	Setting prohibited	SANI → ANI14 → AV_{REF0} → AV_{SS}
1	1	1	1	ANI15	SANI → ANI15	Setting prohibited	SANI → ANI15 → AV_{REF0} → AV_{SS}

(4) AD converter mode register 1H (ADSCM1H)

This register is used to control the power supply to the AD converter and also set the conversion time.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 11-6: AD Converter Mode Register 1H (ADSCM1H) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
ADSCM1H	PDB	0	0	0	FR3	FR2	FR1	FR0	FFFFF203H	00H
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W		

PDB	Power down bit
0	A/D converter power OFF
1	A/D converter power ON

Cautions: 1. When bit PDB of A/D converter mode register 1H (ADSCM1H) is changed from 0 to 1, an A/D core stabilization time of 1 µs is required prior to the first conversion only.

2. When not using the A/D converter, stop its operation by setting the PDB bit = 0.

FR3	FR2	FR1	FR0	Conversion clock	Conversion Rate (µs)					
					f_{XX} = 24 MHz		f_{XX} = 32 MHz		f_{XX} = 40 MHz	
					Select Mode	Polling Mode and Scan Mode	Select Mode	Polling Mode and Scan Mode	Select Mode	Polling Mode and Scan Mode
0	0	0	0	f_{XX} / 4	15.1	14.7	11.3	11	9.1	8.8
0	0	0	1	f_{XX} / 4	9.8	9.3	7.3	7	5.9	5.6
0	0	1	0	f_{XX} / 4	7.1	6.7	5.3	5	4.3	4.0
0	0	1	1	f_{XX} / 4	5.8	5.3	4.3	4	3.5	3.2
0	1	0	0	f_{XX} / 2	7.5	7.3	5.7	5.5	illegal	
0	1	0	1	f_{XX} / 2	4.9	4.7	3.7	3.5		
0	1	1	0	f_{XX} / 2	3.5	3.3	2.7	2.5		
0	1	1	1	f_{XX} / 2	2.9	2.7	2.2	2.0		
1	0	0	0	f_{XX}	illegal					
1	0	0	1	f_{XX}						
1	0	1	0	f_{XX}						
1	0	1	1	f_{XX}						
1	1	0	0	(setting prohibited)						
1	1	0	1							
1	1	1	0							
1	1	1	1							

Caution: The maximum conversion clock is 16 MHz. Always select a conversion input clock and internal bus operating frequency that fits this specification.

(5) A/D discharge and diagnostic mode register (ADVMS0)

This register is used to select the extended mode operation of the AD converter.
$\overline{\text{RESET}}$ input clears this register to 01H.

Figure 11-7: AD Converter Extended Mode Register (ADVMS0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
ADVMS0	0	0	0	DIAGEN	0	0	0	VMSEN	FFFFF204H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

DIAGEN	Diagnostic mode selection bit
0	Conversion of ADA0CRDD and ADA0CRSS is disabled
1	Conversion of ADA0CRDD and ADA0CRSS is enabled

VMSEN	Discharge function mode selection bit
0	Standard mode
1	Discharge function mode enabled

Caution: When bit CE of ADSCM0H = “1”, it is prohibited to change the value of bit DIAGEN. Otherwise the operation and the result conversion can not be guaranteed.

(6) AD converter external trigger selection register (SELCNT1)

This register is used to select the external source for the conversion start trigger.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 11-8: AD Converter External Trigger Selection Register (SELCNT1) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
SELCNT1	0	0	0	0	0	0	ISEL11	ISEL10	FFFFF30AH	00H
R/W	R	R	R	R	R	R	R/W	R/W		

ISEL11	External trigger selection 1
0	ADTRG input
1	INTTP2CC1 (Timer P2 capture/compare channel 1 interrupt)

ISEL10	External trigger selection 0
0	ADTRG input
1	INTTP2CC0 (Timer P2 capture/compare channel 0 interrupt)

Caution: The edge detection function of ADTRG pin input is rising edge only.

11.4 Operation

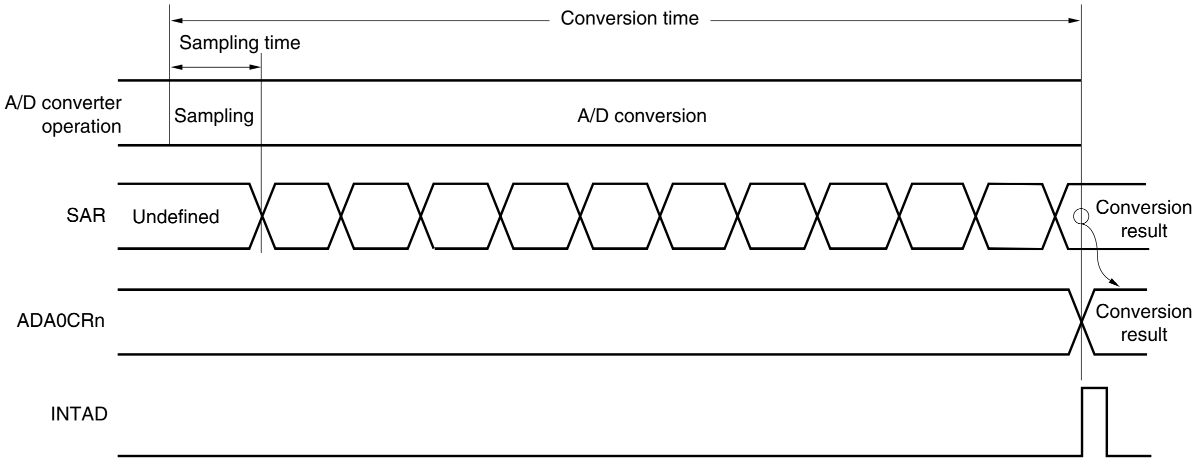
11.4.1 Basic operation

(1) Set the operation mode, trigger mode, and conversion time for executing A/D conversion by using the ADSCM0H, ADSCM0L, ADSCM1H, and ADVMS0 registers. When the CE bit of the ADSCM0H register is set, conversion is started in the software trigger mode and the A/D converter waits for a trigger.

(2) When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

(3) 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. At the same time, an A/D conversion end interrupt request signal (INTAD) is generated.

Figure 11-9: A/D Converter Basic Operation



11.4.2 Trigger mode

Several conversion operations can be specified for A/D converter by specifying operation modes and trigger modes. Those modes are set using bits PLM, TRG0 and TRG1 of A/D conversion mode register 0 (ADSCM0H).

There are two trigger modes, the software trigger mode and the external trigger mode.

The relationship between operation modes, software and external trigger modes is shown below.

Table 11-2: Software Trigger Mode (ADSCM0H Register Configuration)

Software Trigger Mode	PLM bit	MS bit	TRG0 bit	TRG1 bit	Operation Mode
A/D Trigger	0	1	0	0	Single select
	0	0			Single scan
Polling	1	1			Continuous select
	1	0			Continuous scan
Other than above	Setting prohibited				

Table 11-3: External Trigger Mode (ADSCM0H and SELCNT1 Registers Configuration)

External Trigger Mode	PLM bit	TRG1 bit	TRG0 bit	SELCTL11	SELTL10	Operation Mode
A/D Trigger	0	0	1	0	0	ADTRG trigger input pin
					1	INTTP2CC0 trigger input
		1	1	0	0	ADTRG trigger input pin
				1		INTTP2CC1 trigger input
Other than above	Setting prohibited					

Note: ADSCM0H.MS bit selects either the scan mode (MS= 0) or the select mode (MS=1) for each operation mode.

(1) Software trigger mode

The following trigger modes that serve as the start timing of A/D conversion processing are available: A/D trigger mode and polling mode.
These trigger modes are set using the ADSCM0H register.

(a) A/D trigger mode

In this mode, the A/D conversion is started by setting bit CE of A/D conversion mode register 0 (ADSCM0H) to 1.
To restart the A/D conversion after the INTAD interrupt (A/D conversion completed; CS bit =”0”), the CE bit has to be set again to start the next conversion.
If registers ADSCM0H, ADSCM0L, and ADSCM1H are written to during conversion, conversion stops and starts over from the beginning.

(b) Polling mode

In this mode, the A/D conversion is started by setting bit CE of A/D conversion mode register 0 (ADSCM0H) to 1.
Unless the CE bit is set to 0 following conversion, the next conversion operation is performed.
When conversion starts, the CS bit becomes 1 (operation in progress).
If registers ADSCM0H, ADSCM0L, and ADSCM1H are written to during conversion, conversion stops and starts over from the beginning.

(2) External trigger mode

In this mode, conversion is started by the input of an external trigger. External trigger rising edge detection of the ADTRG input pin, or by interrupt signal output from Timer P2 can be specified with bits TRG0 and TRG1 of the ADSCM0H register.
In this mode, setting bit CE of the ADSCM0H register to 1 causes the trigger standby state to be entered, and conversion starts upon input of an external trigger. Upon completion of conversion, the state changes back to the trigger standby state.
When conversion starts, the CS bit becomes 1 (operation in progress). In the trigger standby state, the CS bit is 0 (operation stopped).
Upon input of a valid trigger during conversion, the conversion operation stops and starts over from the beginning.
If registers ADSCM0H, ADSCM0L, and ADSCM1H are written to during conversion, conversion stops and starts over from the beginning.

11.4.3 Operation mode

Four operation modes are described below. The operation modes are set with bits MS and PLM of the ADSCM0H register (refer to **11.3 ”Control Registers” on page 371**).

(1) Select mode

In this mode, a single A/D conversion of one analog input specified with bits ANIS3 - ANIS00 of the ADSCM0L register is performed.

The conversion result for that analog input is saved to the ADA0CRn register. A/D conversion starts upon detection of a trigger, and the A/D conversion interrupt (INTAD) is output upon each conversion completion.

Figure 11-10: Example of Select Mode Operation (ANI2) (1/2)

(a) Timing Example

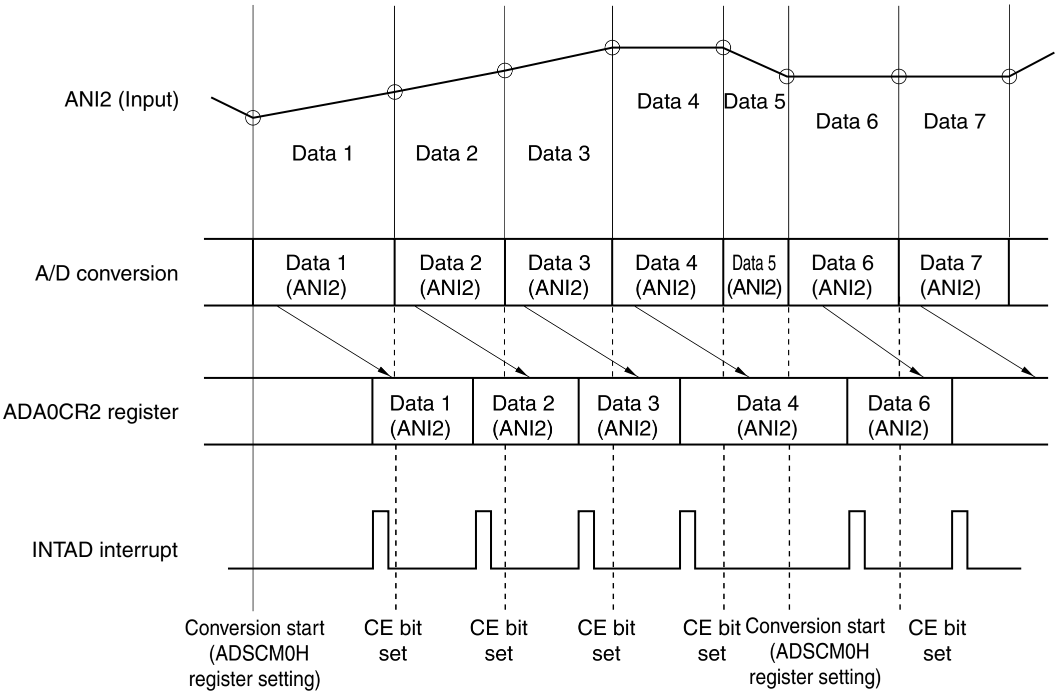
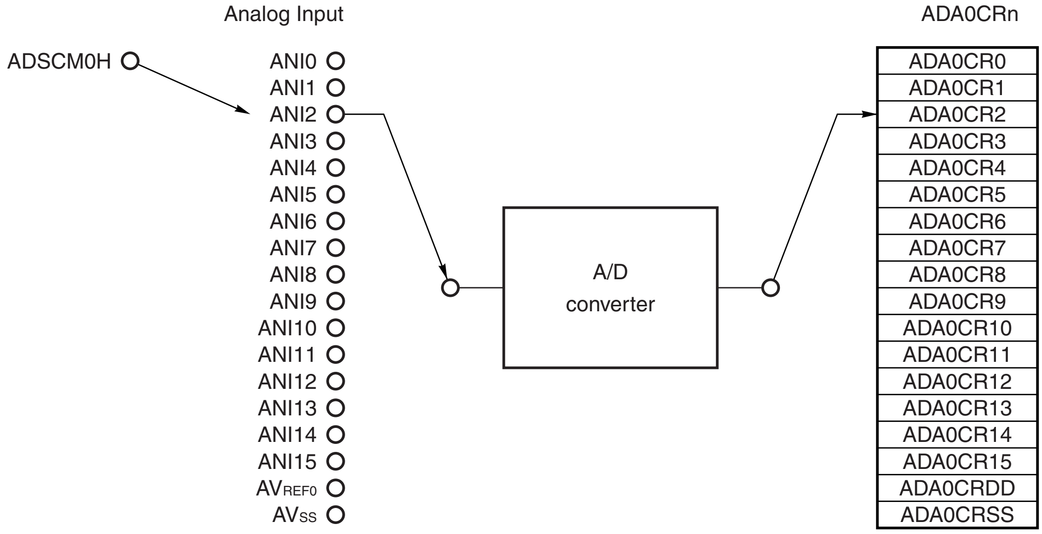


Figure 11-10: Example of Select Mode Operation (ANI2) (2/2)

(b) Block diagram Example



(a) CE bit of ADSCM0H = 1 (enabled)
(b) A/D conversion of ANI2
(c) Store conversion result in ADA0CR2
(d) Generate INTAD interrupt

(2) Single scan mode

In this mode, sequential A/D conversion of the selected analog input channels specified with the SANI3 - SANI0 and ANIS3 - ANIS0 bits of the ADSCM0L register is performed.

The conversion result for the selected analog inputs is saved to the ADA0CRn registers. A/D conversion starts upon detection of a trigger, the selected analog inputs are operated in sequence, and A/D conversion interrupt (INTAD) is output upon conversion completion of the final analog input channel.

Figure 11-11: Example of Single Scan Mode Operation (4-Channel Scan (ANI2 to ANI5)) (1/2)

(a) Timing Example

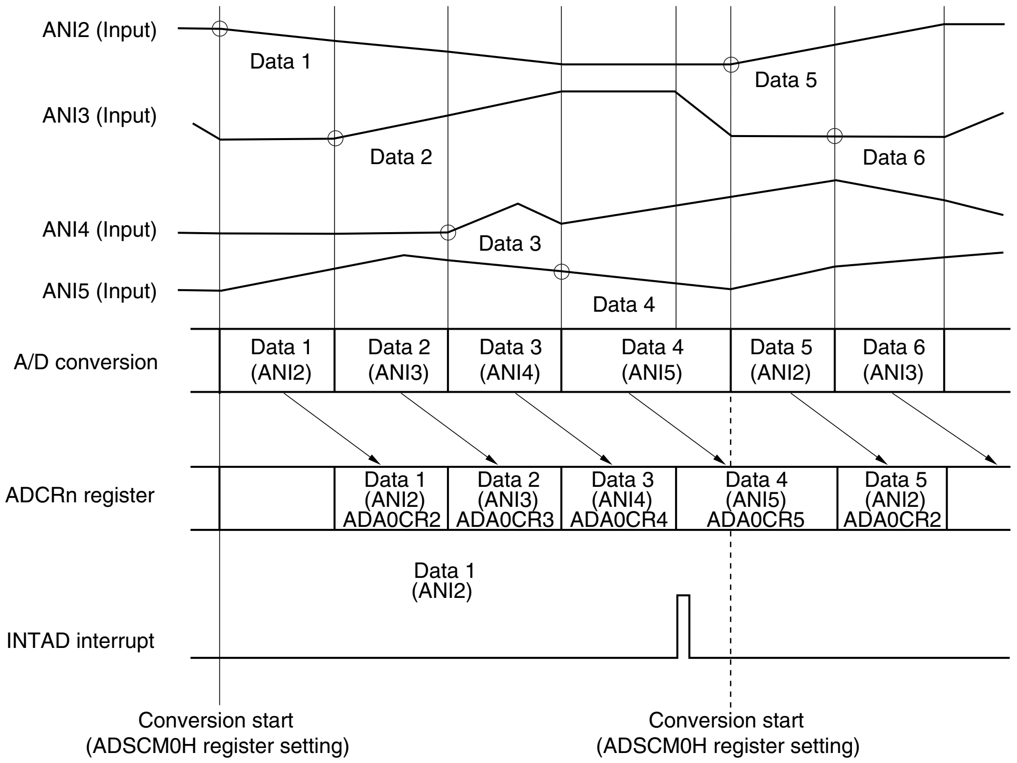
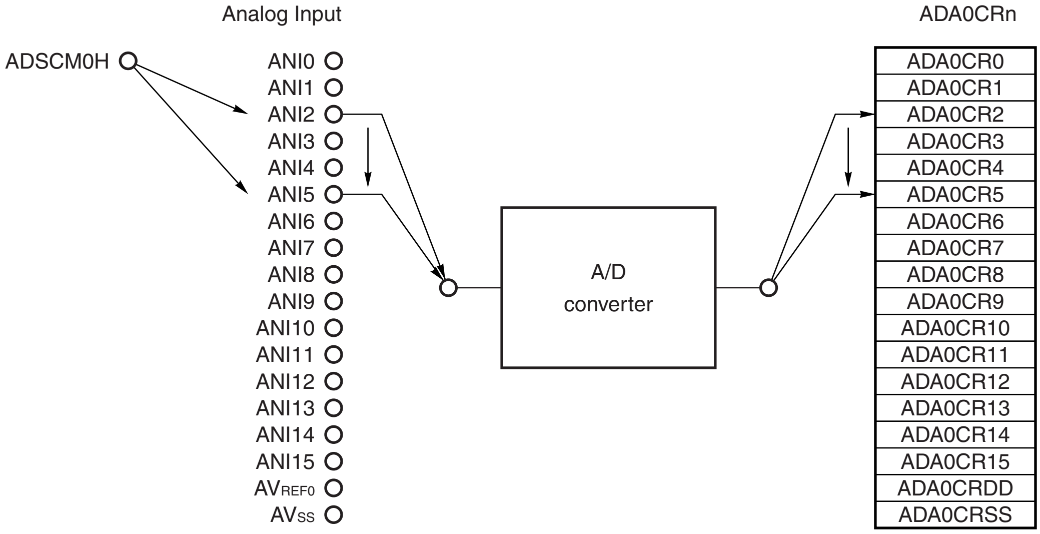


Figure 11-11: Example of Single Scan Mode Operation (4-Channel Scan (ANI2 to ANI5)) (2/2)

(b) Block diagram Example



(a) CE bit of ADSCM0H = 1 (enabled)
(b) A/D conversion of ANI2
(c) Store conversion result in ADA0CR2
(d) A/D conversion of ANI3
(e) Store conversion result in ADA0CR3
(f) A/D conversion of ANI4
(g) Store conversion result in ADA0CR4
(h) A/D conversion of ANI5
(i) Store conversion result in ADA0CR5
(j) Generate INTAD interrupt

(3) Continuous select mode (Polling mode)

In this mode, continuous A/D conversion of one analog input specified with the ADSCM0L register is performed. The conversion result for that analog input is saved to the ADA0CRn register. A/D conversion starts upon detection of a trigger, an A/D conversion interrupt (INTAD) is output upon each conversion completion, and A/D conversion is then started again unless CE bit of ADSCM0H register is set to 0.

Figure 11-12: Example of Continuous Select Mode Operation (ANI2) (1/2)

(a) Timing Example

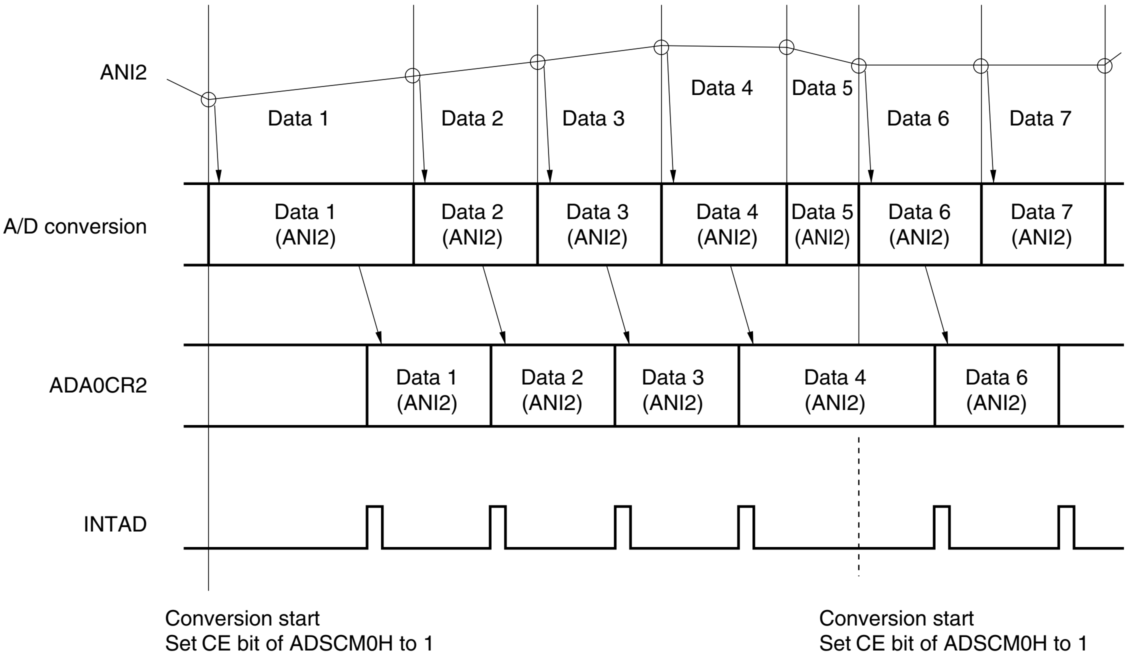
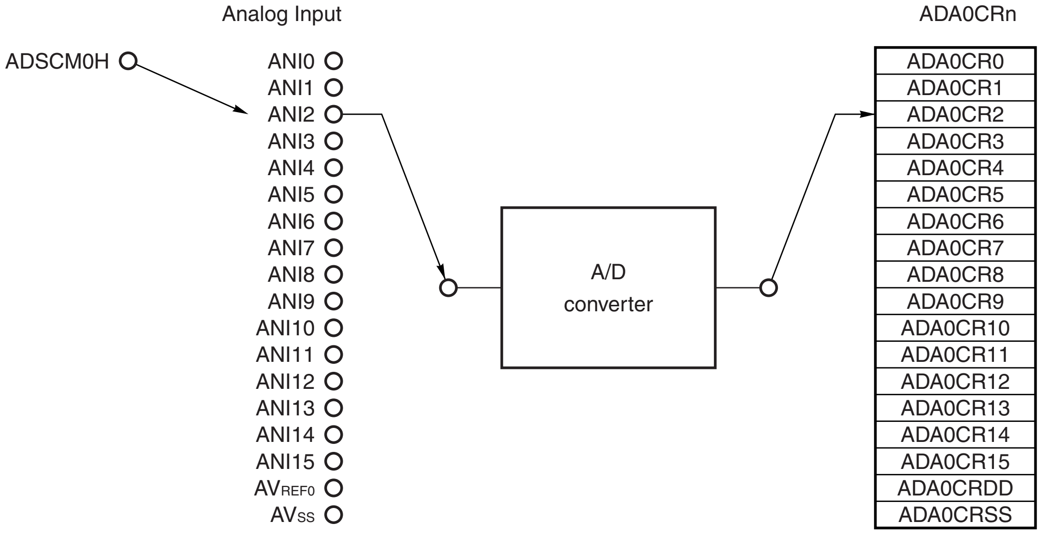


Figure 11-12: Example of Continuous Select Mode Operation (ANI2) (2/2)

(b) Block diagram Example



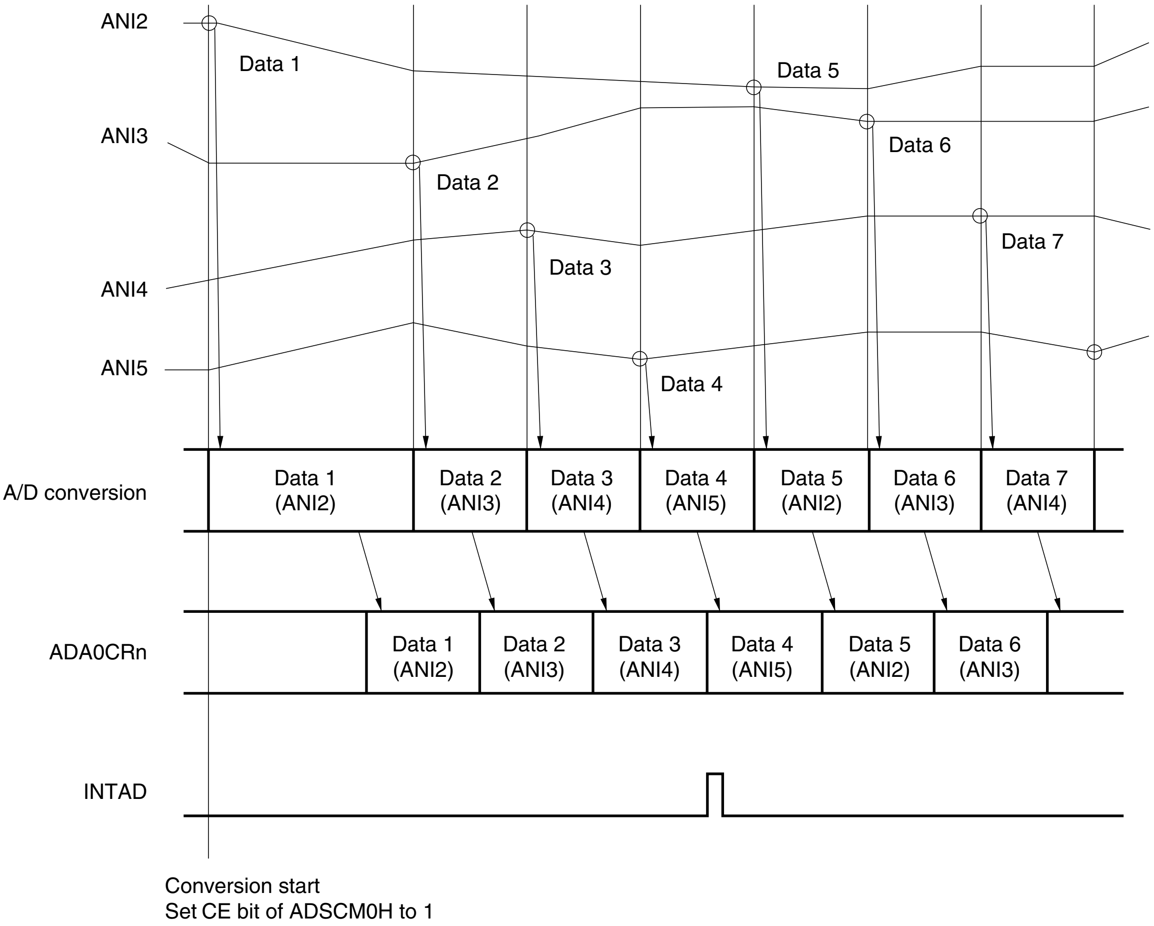
(a) CE bit of ADSCM0H = 1 (enabled)
(b) A/D conversion of ANI2
(c) Store conversion result in ADA0CR2
(d) Generate INTAD interrupt
(e) Return to (b)

(4) Continuous scan mode (Polling mode)

In this mode, A/D conversion is performed starting from ANI2 up to ANI5 specified by the ADSCM0L register. The A/D conversion result is saved to the ADA0CRn register corresponding to each analog input. A/D conversion starts from ANI0 upon trigger detection, an A/D conversion interrupt (INTAD) is generated when conversion of the specified analog input ends, and A/D conversion from ANI0 starts again.

Figure 11-13: Example of Continuous Scan Mode Operation (ANI2 to ANI5) (1/2)

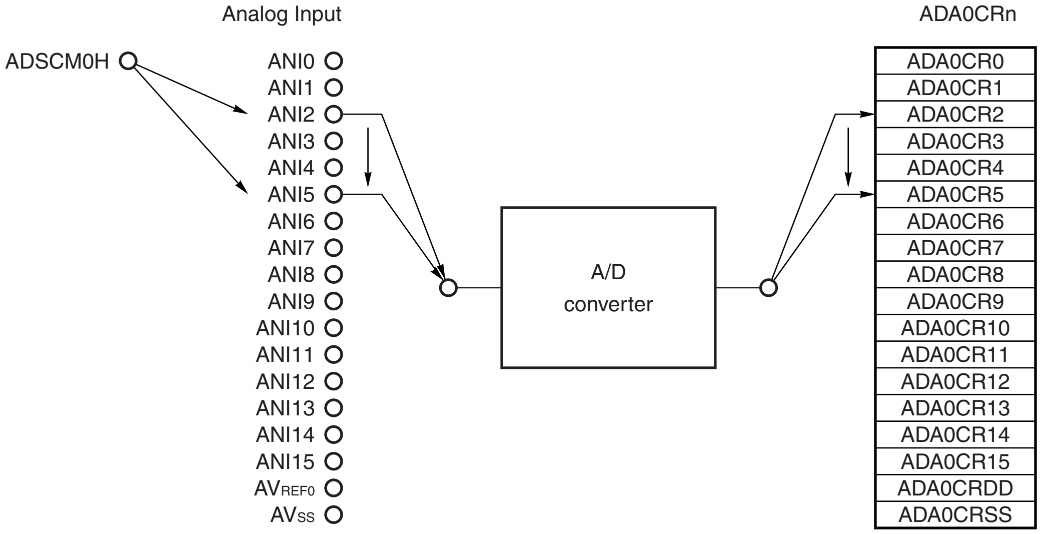
(a) Timing Example



Remark: n = 2 to 5

Figure 11-13: Example of Continuous Scan Mode Operation (ANI2 to ANI5) (2/2)

(b) Block diagram Example



(a) CE bit of ADSCM0H = 1 (enabled)
(b) A/D conversion of ANI2
(c) Store conversion result in ADA0CR2
(d) A/D conversion of ANI3
(e) Store conversion result in ADA0CR3
(f) A/D conversion of ANI4
(g) Store conversion result in ADA0CR4
(h) A/D conversion of ANI5
(i) Store conversion result in ADA0CR5
(j) Generate INTAD interrupt
(k) Return to (b)

11.5 Extended Functions

11.5.1 Automatic discharge operation

The automatic discharge operation internally connects the sample and hold circuit to GND in order to catch open analog inputs and other failure modes. This function is disabled by default and is controlled by the ADVMS0 register (refer to **11.3 "Control Registers" on page 371**).

11.5.2 Diagnostic mode

The diagnostic mode configures the A/D converter to perform a conversion operation on the AV_{REF0} and AV_{SS} terminals. Using this mode allows the user to implement a software algorithm in the application program to calibrate conversion results and thereby reduce the effect of certain error factors such as full scale error and zero offset error. This mode is disabled by default, and is controlled by the ADVMS0 register (refer to **11.3 "Control Registers" on page 371**).

(1) Operation in select mode and continuous select mode

In these modes, AV_{REF0} or AV_{SS} terminals can be selected using the ANISn bits of register ADSCM0L. The A/D conversion result is stored in the appropriate conversion result register and interrupt (INTAD) is output. Operation is performed in the same manner described in **11.4.3 "Operation mode" on page 383**.

(2) Operation in scan mode and continuous scan mode

In this mode, AV_{REF0} and AV_{SS} terminals are scanned immediately after A/D conversion of the Scan mode conversion end channel specified in register ADSCM0L. The A/D conversion result is stored in the appropriate conversion result register and interrupt (INTAD) is output. Operation is performed in the same manner described in **11.4.3 "Operation mode" on page 383**.

11.6 Precautions on Operation

11.6.1 Stopping A/D

If 0 is written in the CE bit of the ADSCM0H register during A/D conversion operation,
it stops A/D conversion operation and an A/D conversion result is not stored in the ADACRn register (n = 0 to 15).

11.6.2 Trigger input during A/D conversion operation

If a trigger is input during A/D conversion operation, the conversion operation stops and starts over from the beginning

11.6.3 External or timer trigger interval

Make the trigger interval (input time interval) in external or timer trigger mode longer than the conversion time specified by the FR3 to FR0 bits of the ADSCM1H register.

(1) When interval = 0

If multiple triggers are input simultaneously, the analog input whose ANIn pin number is smallest is converted. The other trigger signals input at the same time are ignored (n = 0 to 15).

(2) When 0 < interval < conversion time

If an external or timer trigger is input during A/D conversion operation, that trigger input is ignored.

(3) When interval = conversion time

If an external or timer trigger is input at the same time as A/D conversion termination (comparison termination signal and trigger contention), interrupt generation and ADA0CRn register storage of the value with which conversion terminated are performed correctly (n = 0 to 15).

11.6.4 Operation in standby modes

(1) HALT mode

A/D converter continues the operation. When recover from HALT mode, the ADSCM0H, ADSCM0L, ADSCM1H, ADVMS0, or SELCNT1 register and ADA0CRn, ADA0CRSS or ADA0CRDD register maintain their values (n = 0 to 15).
If released by $\overline{\text{RESET}}$ input, all registers are initialized.

(2) IDLE mode, software STOP mode

Since clock supply to A/D converter stops, A/D conversion operation is not performed.
If released by NMI or maskable interrupt input, the ADSCM0H, ADSCM0L, ADSCM1H, ADVMS0, or SELCNT1 register and ADA0CRn, ADA0CRSS or ADA0CRDD register maintain their values (n = 0 to 15).
However, if IDLE mode or software STOP mode is set during A/D conversion operation, A/D conversion operation stops. If released by a non-maskable interrupt request or an unmasked maskable interrupt request, conversion resumes but the conversion result written in the ADA0CRn register becomes undefined (n = 0 to 15).
If released by $\overline{\text{RESET}}$ input, all registers are initialized.

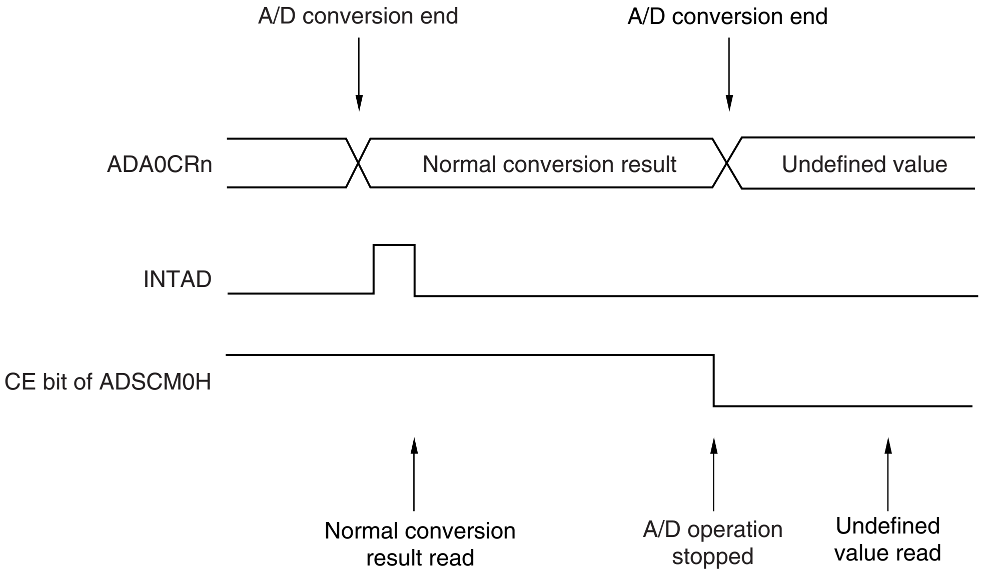
11.6.5 Compare match interrupt in timer trigger mode (External trigger mode)

A TMP2 timer P2 capture/compare registers 0, 1 (TP2CCR0, TP2CCR1) underflow interrupt (INTTP2CC1 or INTTP2CC0) is an A/D conversion start trigger that starts conversion operation. At this time, the TP2CCR0 or TP2CCR1 match interrupt (INTTP2CC1 or INTTP2CC0) also functions as a compare register match interrupt for the CPU.
In order not to generate these match interrupts for the CPU, disable interrupts using the mask bits (TP2CCMK1,TP2CCMK0) of the interrupt control registers (IMR1 or TP2CCIC1 and TP2CCIC0).

11.6.6 Timing that makes the A/D conversion result undefined

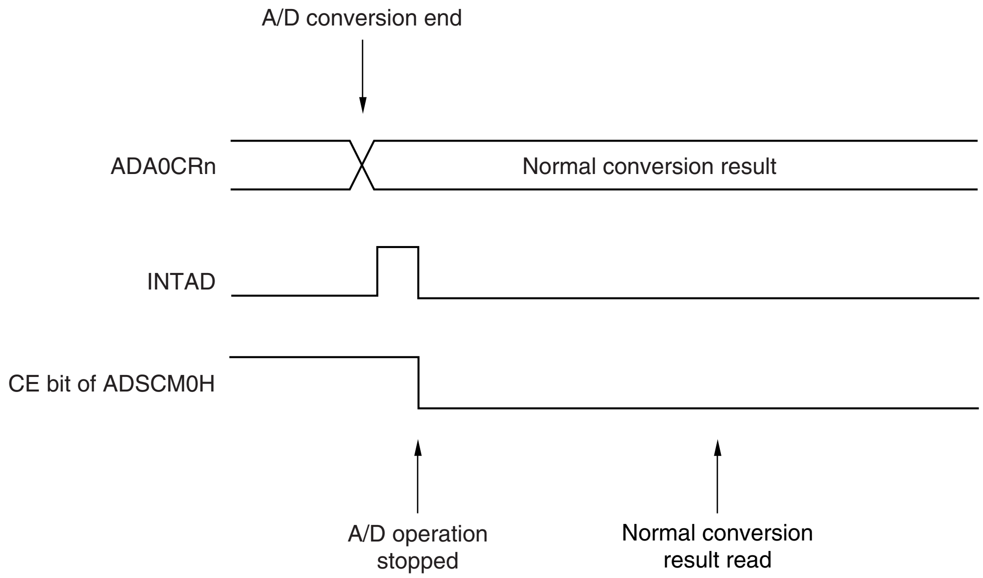
If the timing of the end of A/D conversion and the timing of the stop of operation of the A/D converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read the A/D conversion result while the A/D converter is in operation mode.
Furthermore, when reading an A/D conversion result after the A/D converter operation has been stopped, be sure to have it stop after the time of the next conversion result is complete.
The conversion result read timing is shown in Figures 11-14 and 11-15 below.

Figure 11-14: Conversion Result Read Timing (when Conversion Result is undefined)



Remark: n = 0 to 15

Figure 11-15: Conversion Result Read Timing (When Conversion Result Is Normal)



Remark: n = 0 to 15

11.7 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the CE bit of the ADSCM0H register and PDB bit of the ADSCM1H register to 0.

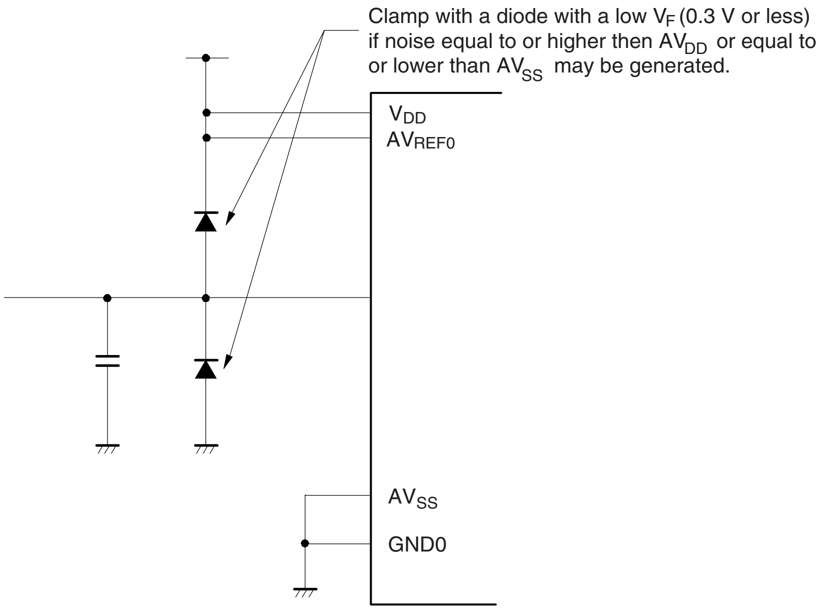
(2) Input range of ANI0 to ANI15 pins

Input the voltage within the specified range to the ANI0 to ANI15 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 ANI0 to ANI15 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 11-16 is recommended.

Figure 11-16: Processing of Analog Input Pin



Clamp with a diode with a low V_F (0.3 V or less) if noise equal to or higher then AV_{DD} or equal to or lower than AV_{SS} may be generated.
V_{DD}
AV_{REF0}
AV_{SS}
GND0

(4) Alternate I/O

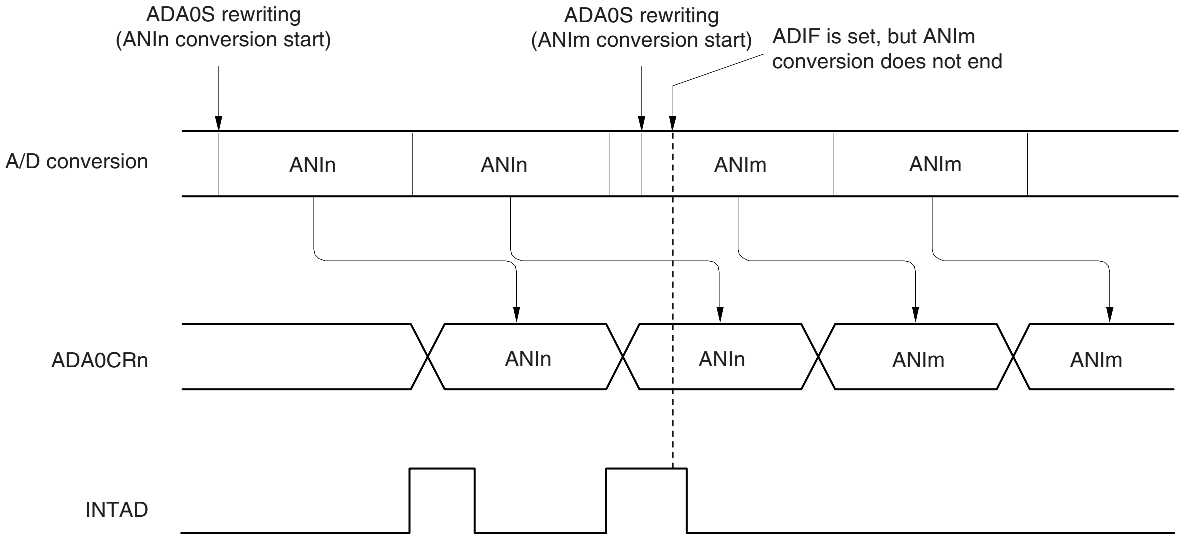
The analog input pins (ANI0 to ANI15) function alternately as port pins. When selecting one of the ANI0 to ANI15 pins to execute A/D conversion, do not execute an instruction to read an input port or write to an output port during conversion as the conversion resolution may drop.

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 ADSCM0L 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 ADSCM0L register is rewritten. If the ADIF flag is read immediately after the ADSCM0L 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 11-17: Generation Timing of A/D Conversion End Interrupt Request



Remark: n = 0 to 15
m = 0 to 15

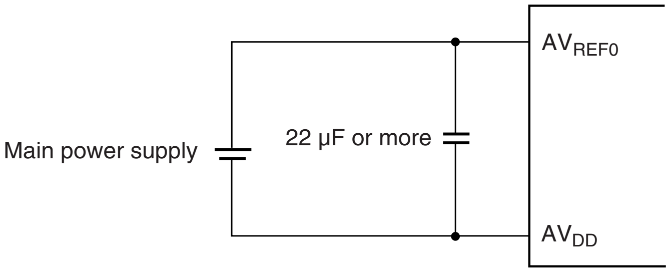
(6) AV$_{REF0}$ pin

(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, be sure to supply the same voltage as V$_{DD}$ to the AV$_{REF0}$ pin as shown in Figure 11-18.

(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 CE 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 11-18.

(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 11-18: AV$_{REF0}$ Pin Processing Example



(7) Reading ADA0CRn register

When ADSCM0H, ADSCM0L or ADSCM1H 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 ADSCM0H, ADSCM0L and ADSCM1H registers. The correct conversion result may not be read at a timing different from the above.

(8) About A/D conversion result

The illegal conversion result sometimes occur by noise, in the case that the analogue input pin and also reference voltage input pin receive the influence of noise. The software processing is necessary, to avoid that exerts bad influence to the system by this illegal conversion result. Next the example of software processing is shown.
Please use the mean value of A/D conversion result of the plural time as the result of A/D conversion.
In the case that does A/D conversion of the plural time continuously and the specific conversion result was obtained, please use the conversion result that is excluded this value.
Please do abnormal processing after abnormal occurrence is confirmed once again, without doing abnormal processing right away, in the case that A/D conversion result that is judged that abnormality occurred to the system was obtained.

11.8 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

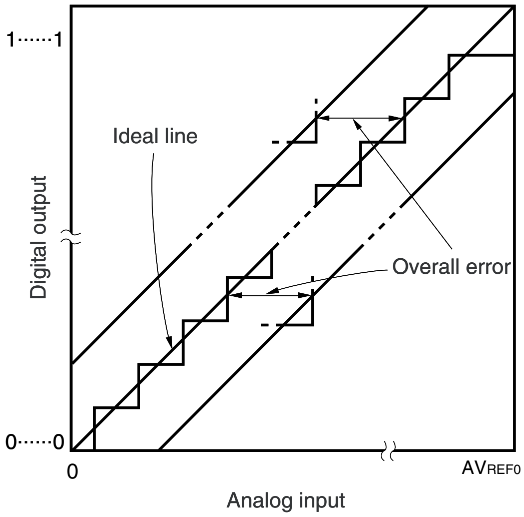
When the resolution is 10 bits, 1LSB is as follows:
1LSB = $1/2^{10}$ = 1/1024
= 0.098 %FSR

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 11-19: Overall Error



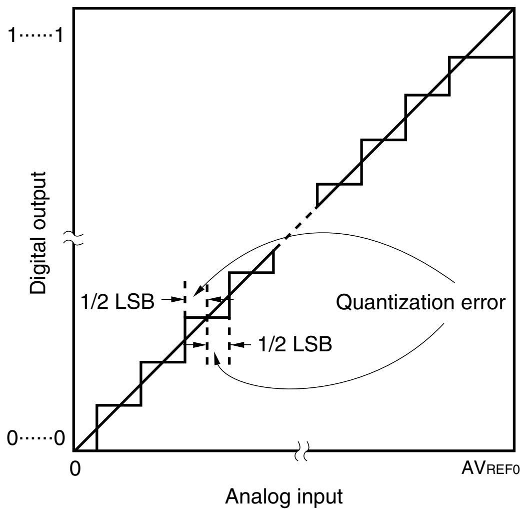
(3) Quantization error

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.

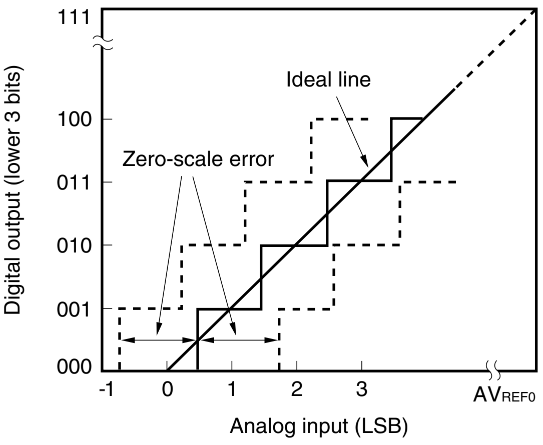
Figure 11-20: Quantization Error



(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).

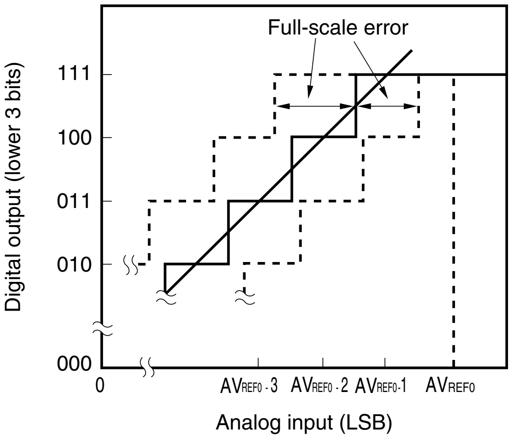
Figure 11-21: Zero-Scale Error



(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 0…111 (full scale 3/2 LSB).

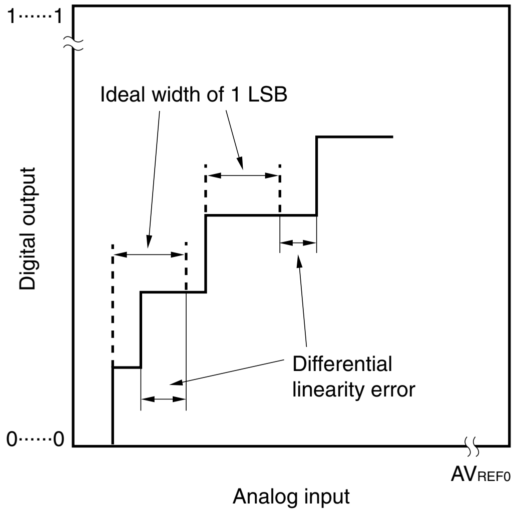
Figure 11-22: Full-Scale Error



(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.

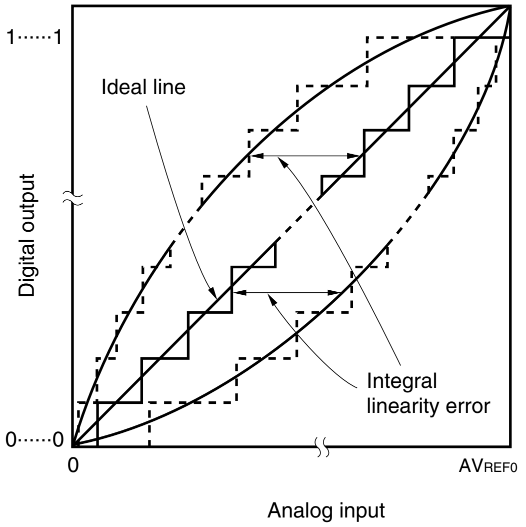
Figure 11-23: Differential Linearity Error



(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 11-24: Integral Linearity Error



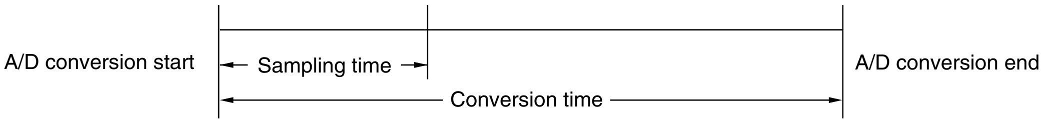
(8) Conversion time

This is the time required to obtain a digital output after an analog input voltage has been assigned. 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 11-25: Sampling Time



[MEMO]

Chapter 12 Asynchronous Serial Interface A (UARTA)

The V850E/RS1 includes two asynchronous serial interfaces A (UARTA).

12.1 Features

- Transfer rate: 300 bps to 312.5 kbps
- Full-duplex communication:
  - UARTA receive data register n (UAnRX)
  - UARTA transmit data register n (UAnTX)
- 2-pin configuration
  - TXDAn: Output pin of transmit data
  - RXDAn: Input pin of receive data
- Reception error detection function: parity error, framing error, overrun error
- Interrupt sources: 2 types
  - Reception complete interrupt (INTUAnR): An interrupt is generated in the reception enabled status by ORing three types of reception errors. It is also generated when receive data is transferred from the shift register to receive buffer register n after completion of serial transfer.
  - Transmission enable interrupt (INTUAnT): Generated when transmit data is transferred from the transmit buffer register to the shift register in the transmission enabled status.
- Character length of transmit/receive data: 7 or 8 bits
- Parity function: odd, even, 0, none
- Transmission stop bit: 1 or 2 bits
- Dedicated baud rate generator
- MSB/LSB first transfer selectable
- Transmit/receive data reversible

Remark: n = 0 to 1

12.2 Configuration

UARTA consists of the following hardware:

Table 12-1: Configuration of UARTA0 and UARTA1

Item	Configuration
Register	UARTAn reception shift register UARTAn reception data register (UAnRX) UARTAn transmit shift register UARTAn transmit data register (UAnTX)
Reception data input	2 (RXDA0, RXDA1)
Transmit data output	2 (TXDA0, TXDA1)
Baud rate clock input	2 ($\overline{\text{ASCKA0}}$, $\overline{\text{ASCKA1}}$)
Control register	UARTAn control register 0 to 2 (UAnCTL0 to UAnCTL2) UARTAn option control register 0 (UAnOPT0) UARTAn status register (UAnSTR)

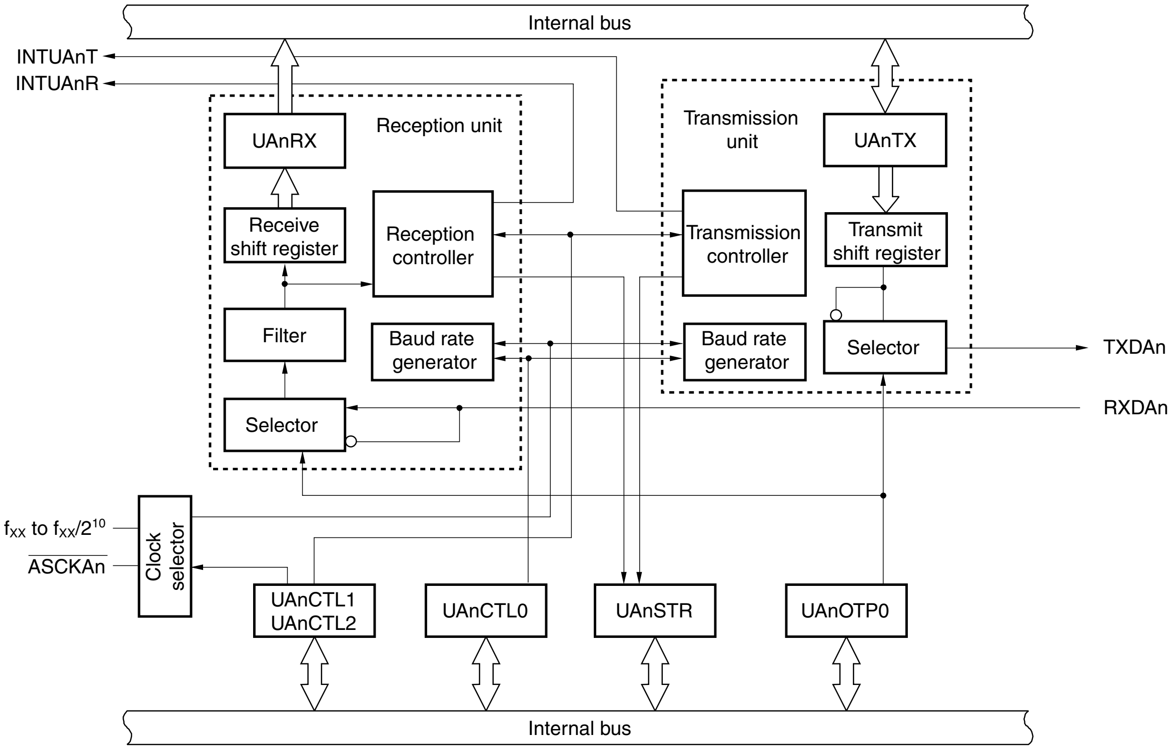
Remark: n = 0 to 1

The pins of asynchronous serial interface A (UARTA) function alternately as port pins. For how to select the alternate functions, refer to the description of registers in **Chapter 4 Port Functions**.

Table 12-2: List of Pins of Asynchronous Serial Interface A

Pin Name	Alternate-Function Pin	I/O	Function
RXDA0	P31/INTP7	Input	Serial receive data input (UARTA0)
RXDA1	P95		Serial receive data input (UARTA1)
TXDA0	P30	Output	Serial transmit data output (UARTA0)
TXDA1	P96		Serial transmit data output (UARTA1)
$\overline{\text{ASCKA0}}$	P32	Input	Baud rate clock input (UARTA0)
$\overline{\text{ASCKA1}}$	P94/CS303		Baud rate clock input (UARTA1)

Figure 12-1: Block Diagram of Asynchronous Serial Interface A



Remark: n = 0 to 1

12.2.1 Control registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that specifies the operation of the asynchronous serial interface.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the input clock of the asynchronous serial interface.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register that controls the baud rate of the asynchronous serial interface.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls serial transfer by the asynchronous serial interface.

(5) UARTAn status register (UAnSTR)

The UAnSTR register is a collection of flags that indicate the contents of the error when a reception error occurs. The corresponding reception error flag is set to 1 when a reception error occurs, and is reset to 0 when the UAnSTR register is read.

(6) UARTAn receive shift register

This shift register converts the serial data input to the RXDAn pin into parallel data. When data of 1 byte is received and then a stop bit is detected, the receive data is transferred to the UAnRX register.
This register cannot be directly manipulated.

(7) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that holds receive data. When seven characters are received, 0 is stored in the higher bit (in LSB-first reception).
While reception is enabled, receive data is transferred from the UARTAn receive shift register to the UAnRX register in synchronization with completion of shift-in processing of one frame.
When the data has been transferred to the UAnRX register, a reception complete interrupt request signal (INTUAnR) is generated.

(8) UARTAn transmit shift register

The transmit shift register converts the parallel data transferred from the UAnTX register into serial data.
When data of 1 byte is transferred from the UAnTX register, the data of the shift register is output from the TXDAn pin.
This register cannot be directly manipulated.

(9) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit buffer for transmit data. By writing transmit data to the UAnTX register, a transmission operation is started. When data can be written to the UAnTX register (when data of one frame is transferred from the UAnTX register to the UARTAn transmit shift register), a transmission enable interrupt request signal (INTUAnT) is generated.

12.3 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the serial transfer operation of UARTAn.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 10H.

Caution: Set the UAnPWR bit to 1 and UAnRXE bit to 1 while a high level is being input to the RXDAn pin (when the UAnRDL bit of the UAnOP0 register is 0). If the UAnPWR and UAnRXE bits are set to 1 while a low level is being input to the RXDAn pin, reception is started.

Figure 12-2: UARTAn Control Register 0 (UAnCTL0) Format (1/2)

Address: UA0CTL0: FFFFFA00H, UA1CTL0: FFFFFA10H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL		R/W	10H

Remark: n = 0 to 1

UAnPWR	Control of operation of UARTAn
0	Disable clock operation (asynchronously reset UARTAn).
1	Enable clock operation.
The UAnPWR bit controls the operating clock and asynchronously resets UARTAn. When this bit is cleared to 0, the output of the TXDAn pin is fixed to the high level for UAnTDL=0 or low level for UAnTDL=1.	

UAnTXE	Transmission operation enable
0	Stop transmission operation.
1	Enable transmission operation.
When the UAnTXE bit is cleared to 0, the output of the TXDAn pin is fixed to the high level for UAnTDL=0 or low level for UAnTDL=1. This bit is synchronized with the operating clock. When the transmission unit is initialized, therefore, set the UAnTXE bit from 0 to 1. The transmission operation will be enabled two clocks later. A value written to the UAnTXE bit is ignored when the UAnPWR bit = 0.	

UAnRXE	Reception operation enable
0	Stop reception operation.
1	Enable reception operation.
When the UAnRXE bit is cleared to 0, the reception operation is stopped. Consequently, even if specified data is transferred, the reception complete interrupt is not output, and the UAnRX register is not updated. The UAnRXE bit is synchronized with the operating clock. When the reception unit is initialized, therefore, set the UAnRXE bit from 0 to 1. The reception operation will be enabled two clocks later. A value written to the UAnRXE bit is ignored when the UAnPWR bit = 0.	

Figure 12-2: UARTAn Control Register 0 (UAnCTL0) Format (2/2)

UAnDIR	Selection of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first
This bit can be rewritten only when the UAnPWR bit = 0 or when UAnTXE bit = UAnRXE bit = 0.	

UAnPS1	UAnPS0	Selection of parity for transmission	Selection of parity for reception
0	0	No parity output	Reception without parity
0	1	Output 0 parity	Reception with 0 parity
1	0	Output odd parity	Identified as odd parity
1	1	Output even parity	Identified as even parity
• This bit can be rewritten only when the UAnPWR bit = 0 or when the UAnTXE bit = UAnRXE bit = 0. • If “Reception with 0 parity” is selected for reception, the parity is not identified. • Consequently, the UAnPE bit of the UAnSTR register is not set, and an error interrupt is not generated even if a parity error occurs.			

UAnCL	Specification of data character length of one frame of transmit/receive data.
0	7 bits
1	8 bits
This bit can be rewritten only when the UAnPWR bit = 0 or when the UAnTXE bit = UAnRXE bit = 0.	

UAnSL	Specification of stop bit length of transmit data.
0	1 bit
1	2 bits
This bit can be rewritten only when the UAnPWR bit = 0 or when the UAnTXE bit = UAnRXE bit = 0.	

Remark: For details of the parity, refer to **12.5.9 Types and operation of parity**.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the clock of UARTAn.
This register can be read or written in 8-bit units only.
Reset input clears this register to 00H.

Figure 12-3: UARTAn Control Register 1 (UAnCTL1) Format

Address: UA0CTL1: FFFFFA01H, UA1CTL1: FFFFFA11H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnCTL1	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0		R/W	00H

Remark: n = 0 to 1

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Selection of base clock (f_{XCLK})
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	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	0	0	1	$f_{XX}/512$
1	0	1	0	$f_{XX}/1024$
1	0	1	1	External clock ($\overline{\text{ASCKAn}}$ pin)
Other than above				Setting prohibited

Caution: This register can be rewritten only when the UAnPWR bit of the UAnCTL0 register = 0.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is used to select the baud rate (serial transfer rate) clock of UARTAn.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

Figure 12-4: UARTAn Control Register 2 (UAnCTL2) Format

Address: UA0CTL2: FFFFFA02H, UA1CTL2: FFFFFA12H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0		R/W	FFH

Remark: n = 0 to 1

UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0	Rated value (k)	Serial clock
0	0	0	0	0	0	–	–	–	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	0	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

Remark: f_{XCLK} is the frequency of the base clock selected by the UAnCTL1 register.

Cautions:

1. **This register can be rewritten only when the UAnPWR bit of the UAnCTL0 register = 0 or when the UAnTXE bit = UAnRXE bit = 0.**
2. **The baud rate is the serial clock divided by two.**

For details on the baud rate generator, please refer to section **12.6 ”Dedicated Baud Rate Generator” on page 425**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the serial transfer operation of UARTAn.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 14H.

Figure 12-5: UARTAn Option Control Register 0 (UAnOPT0)

Address: UA0OPT0: FFFFFA03H, UA1OPT0: FFFFFA13H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnOPT0	0	0	0	1	0	1	UAnTDL	UAnRDL		R/W	14H

Remarks: 1. n = 0 to 1

2. It is mandatory to write the fixed bits to the values given above to ensure proper operation.

UAnTDL	Transmit data level control bit
0	Normal output of transfer data
1	Inverted output of transfer data
• The value of the TXDAn bit can be inverted by the UAnTDL bit. • This bit can be set when the UAnPWR bit of the UAnCTL0 register = 0 or when the UAnTXE bit of the UAnCTL0 register = 0.	

UAnRDL	Receive data level control bit
0	Normal input of transfer data
1	Inverted input of transfer data
• The value of the RXDAn pin can be inverted by the UAnRDL bit. • This bit can be set when the UAnPWR bit of the UAnCTL0 register = 0 or when the UAnRXE bit of the UAnCTL0 register = 0.	

Remark: For details of the parity, refer to **12.5.9 Types and operation of parity**.

(5) UARTAn status register (UAnSTR)

The UAnSTR register is an 8-bit register that indicates the transfer status of UARTAn and the contents of a reception error.

This bit can be read or written in 8-bit or 1-bit units, but the UAnTSF bit can only be read. The UAnPE, UAnFE, and UAnOVE bits can be read or written, but they can only be cleared by writing 0 to them, and cannot be set by writing 1 (if 1 is written to these bits, they hold the current status). The following table shows the initialization conditions of these bits.

Register/Bit	Initialization Conditions
UAnSTR register	• Reset input • UAnPWR bit of UAnCTL0 register = 0
UAnTSF bit	• UAnTXE bit of UAnCTL0 register = 0
UAnPE, UAnFE, UAnOVE bits	• Writing of 0 • UAnRXE bit of UAnCTL0 register = 0

Figure 12-6: UARTAn Status Register (UAnSTR) Format (1/2)

Address: UA0STR: FFFFFA04H, UA1STR: FFFFFA14H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE		R/W	00H

Remark: n = 0 to 1

UAnTSF	Transfer status flag
0	• When UAnPWR bit of UAnCTL0 register = 0 or when UAnTXE bit of UAnCTL0 register = 0 • If next transfer data is not in UAnTX after completion of transfer
1	Writing to UAnTX register
The UAnTSF bit is always 1 when transmission is executed continuously. Before initializing the transmission unit, check that the UAnTSF bit = 0. If the transmission unit is initialized while the UAnTSF bit = 1, the transmit data cannot be guaranteed.	

UAnPE	Parity error flag
0	• When UAnPWR bit of UAnCTL0 register = 0 or when UAnRXE bit of UAnCTL0 register = 0 • When 0 is written to this bit
1	When the parity of the received data does not match the parity bit
• The operation of the UAnPE bit differs depending on how the UAnPS1 and UAnPS0 bits of the UAnCTL0 register are set. • Although the UAnPE bit can be read or written, it can only be cleared by writing 0, and cannot be set by writing 1. It holds the current status when 1 is written.	

Figure 12-6: UARTAn Status Register (UAnSTR) Format (2/2)

<table>
<tr><th>UAnFE</th><th>Framing error flag</th></tr>
<tr><td>0</td><td>• When UAnPWR bit of UAnCTL0 register = 0 or when UAnRXE bit of UAnCTL0 register = 0
• When 0 is written</td></tr>
<tr><td>1</td><td>When a stop bit is not detected on reception</td></tr>
<tr><td colspan="2">• Only the first bit of the receive data is checked as a stop bit, regardless of the value of the UAnSL bit of the UAnCTL0 register.
• Although the UAnFE bit can be read or written, it can only be cleared by writing 0, and cannot be set by writing 1. It holds the current status when 1 is written.</td></tr>
</table>

<table>
<tr><th>UAnOVE</th><th>Overrun error flag</th></tr>
<tr><td>0</td><td>• When UAnPWR bit of UAnCTL0 register = 0 or when UAnRXE bit of UAnCTL0 register = 0
• When 0 is written</td></tr>
<tr><td>1</td><td>When receive data is set to the UAnRX register and the next reception operation is completed before that data is read</td></tr>
<tr><td colspan="2">• If an overrun error occurs, the next receive data is not written to the receive buffer but discarded.
• Although the UAnOVE bit can be read or written, it can only be cleared by writing 0, and cannot be set by writing 1. It holds the current status when 1 is written.</td></tr>
</table>

(6) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that stores the parallel data converted by the receive shift register.
On completion of reception of 1 byte of data, the data stored in the receive shift register is transferred to the UAnRX register.
If the data length is specified to be 7 bits and when data is received with the LSB first, the receive data is transferred to bits 6 to 0 of the UAnRX register, and the MSB is always 0. If data is received with the MSB first, the receive data is transferred to bits 7 to 1 of the UAnRX register, and the LSB is always 0.
If an overrun error (UAnOVE) occurs, the receive data at that time is not transferred to the UAnRX register.
The UAnRX register is read-only, in 8-bit units.
Reset input and setting the UAnPWR bit of the UAnCTL0 register to 0 set this register to FFH.

Figure 12-7: UARTAn Receive Data Register (UAnRX) Format

Address: UA0RX: FFFFFA06H, UA1RX: FFFFFA16H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnRX										R	FFH

Remark: n = 0 to 1

(7) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit register that sets transmit data.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

Figure 12-8: UARTAn Transmit Data Register (UAnTX) Format

Address: UA0TX: FFFFFA07H, UA1TX: FFFFFA17H

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
UAnTX										R/W	FFH

Remark: n = 0 to 1

12.4 Interrupt Request Signals

UARTAn generates the following two types of interrupt request signals.

- Reception complete interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

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 12-3: Interrupts and Their Default Priority

Interrupt	Priority
Reception complete	High
Transmission enable	Low

(1) Reception complete interrupt request signal (INTUAnR)

When data is shifted in to the receive shift register with reception enabled, and transferred to the UAnRX register, the reception complete interrupt request signal is generated.
This interrupt request signal can also be generated if a reception error occurs, instead of a reception error interrupt.
When the reception complete interrupt request signal is acknowledged and the data is read, read the UAnSTR register to check that the result of reception is not an error.
Reception complete interrupt request signals are not generated while reception is disabled.

(2) Transmission enable interrupt request signal (INTUAnT)

The transmission enable interrupt request signal is generated when transmit data is transferred from the UAnTX register to the UARTAn transmit shift register with transmission enabled.

12.5 Operation

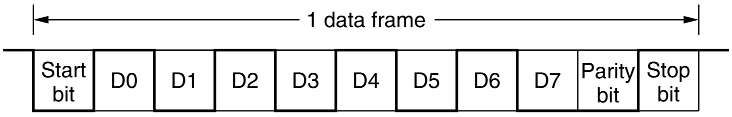
12.5.1 Data format

Full-duplex serial data is transmitted or received.
The transmit/receive data is in the format shown in Figure 12-9, consisting of a start bit, character bits, a parity bit, and 1 or 2 stop bits.
The character bit length in one data frame, parity, stop bit length, and whether data is transferred with the MSB or LSB first, are specified by the UAnCTL0 register.
The UAnTDL bit of the UAnOPT0 register is used to specify whether the signal output from the TXDAn pin is inverted or not.

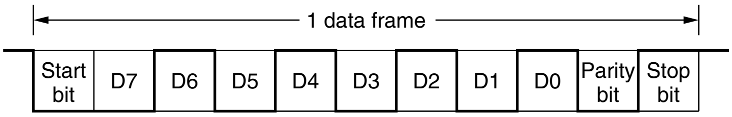
- Start bit … 1 bit
- Character bit … 7 or 8 bits
- Parity bit … Even parity, odd parity, 0 parity, or no parity
- Stop bit … 1 or 2 bits

Figure 12-9: Format of Transmit/Receive Data of UARTA (1/2)

(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inverted

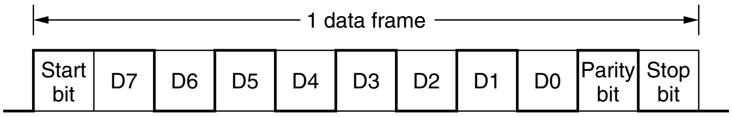
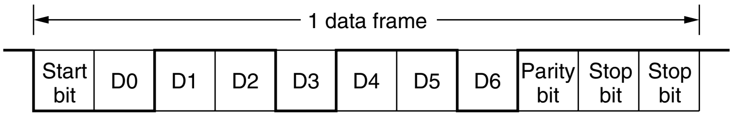
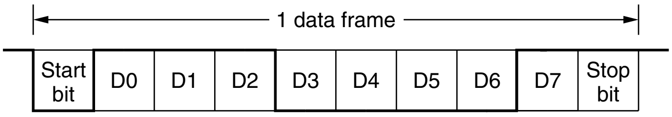


Figure 12-9: Format of Transmit/Receive Data of UARTA (2/2)

(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



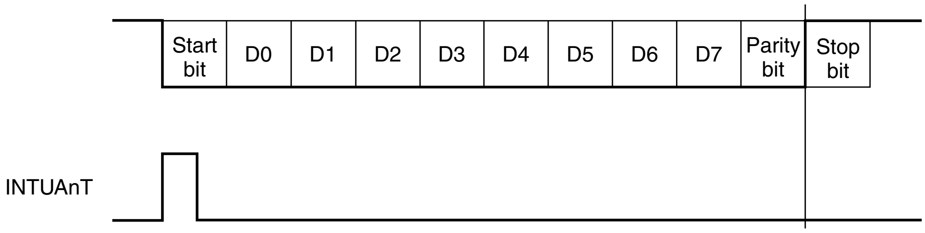
(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



12.5.2 UART transmission

When the UAnPWR bit of the UAnCTL0 register is set to 1, the TXDAn pin outputs a high level.
If the UAnTXE bit of the UAnCTL0 register is subsequently set to 1, transmission is enabled. Transmission is started by writing transmit data to the UAnTX register. A start bit, parity bit, and stop bit are automatically appended to the transmit data.
When transmission is started, the data in the UAnTX register is transferred to the UARTAn transmit shift register.
As soon as the data of the UAnTX register has been transferred to the UARTAn transmit shift register, a transmission enable interrupt request signal (INTUAnT) is generated. Then the UARTAn transmit shift register sequentially outputs the data to the TXDAn pin, starting from the LSB. When the INTUAnT signal is generated, writing the next transfer data to the UAnTX register is enabled.
By writing the data to be transmitted next to the UAnTX register during transfer, transmission can be continuously executed.

Figure 12-10: UART Transmission



12.5.3 Procedure of continuous transmission

With UARTA, the next transmit data can be written to the UAnTX register as soon as the UARTAn transmit shift register has started its shift operation. The timing at which data is transferred to the UARTAn transmit shift register can be identified by the transmission enable interrupt request signal (INTUAnT). The INTUAnT signal enables continuous transmission even while an interrupt is being serviced after transmission of 1 data frame, so that an efficient communication rate can be realized.
During continuous transmission, do not write the next transmit data to the UAnTX register before a transmit request interrupt signal (INTUAnT) is generated after transmit data is written to the UAnTX register and transferred to the UARTAn transmit shift register. If a value is written to the UAnTX register before a transmit request interrupt signal is generated, the previously set transmit data is overwritten by the latest transmit data.

Caution: Continuous transmission operating (UAnTSF bit is 1), can not change register. While continuous transmission is being executed, execute initialization after checking that the UAnTSF bit is 0. If initialization is executed while the UAnTSF bit is 1, the transmit data cannot be guaranteed.

Figure 12-11: Processing Flow of Continuous Transfer

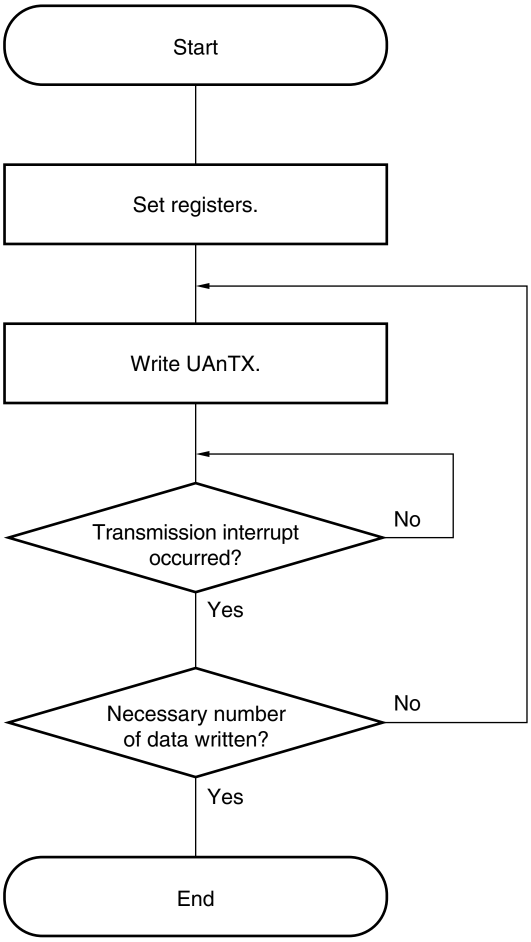
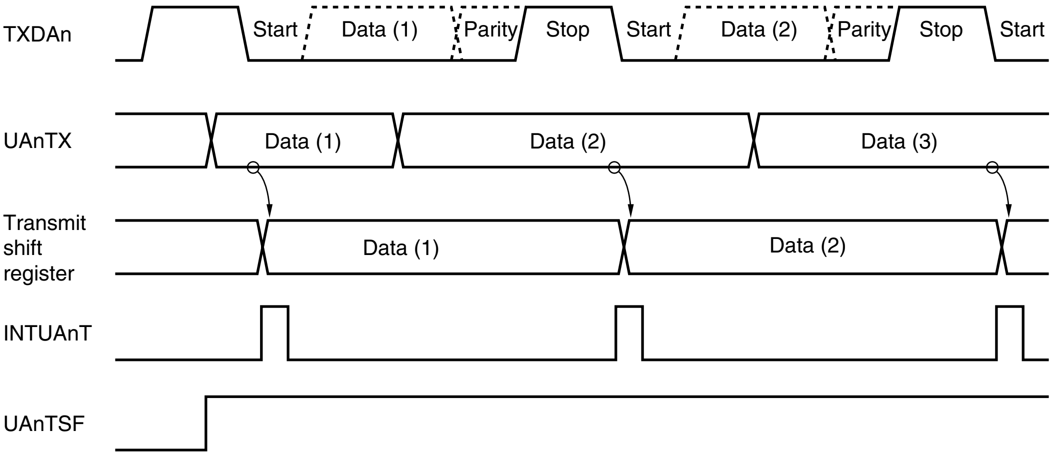
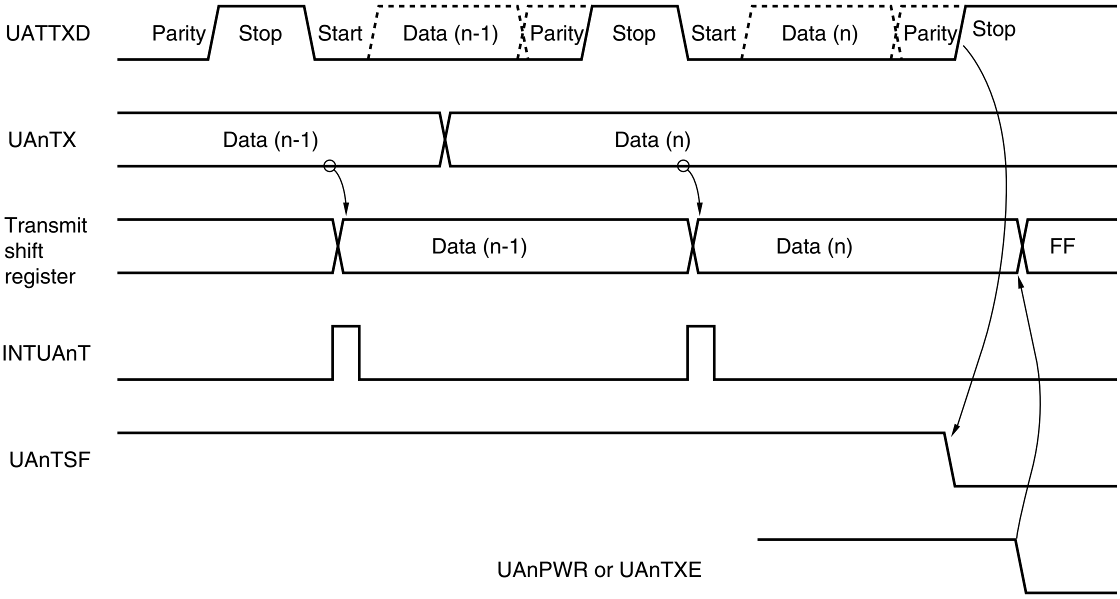

Start
Set registers.
Write UAnTX.
Transmission interrupt occurred?
No
Yes
Necessary number of data written?
No
Yes
End

Figure 12-12: Timing of Continuous Transmission Operation

(a) Start of transfer



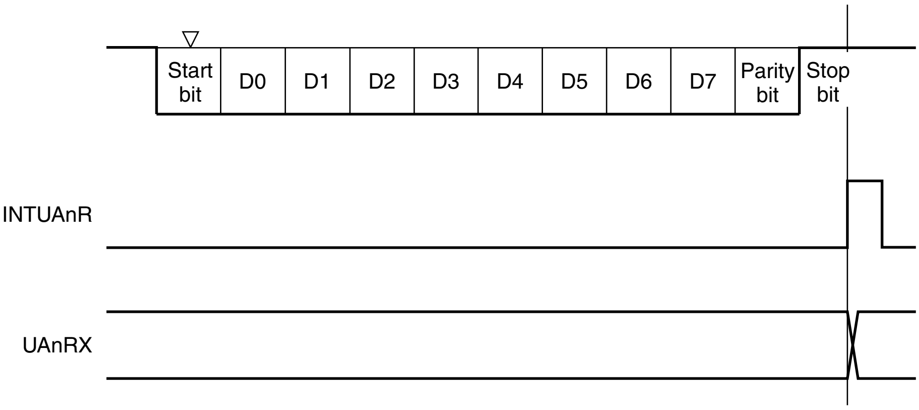
(b) End of transfer



12.5.4 UART reception

When the UAnPWR bit of the UAnCTL0 register is set to 1 and then the UAnRX bit of the UAnCTL0 register is set to 1, UARTA waits for reception. In the reception wait status, the RXDAn pin is monitored and the start bit is detected.
To recognize the start bit, a two-stage detection routine is used.
When the falling of the RXDAn pin is detected, an 8-bit counter starts counting. When the 8-bit counter has counted the set value of the UAnCTL2 register, the level of the RXDAn pin is monitored again (indicated by ▽ in Figure 12-13). If the RXDAn pin is low at this time, the start bit is recognized. When the start bit is recognized, reception is started, and serial data is sequentially stored in the UARTAn receive shift register at the selected baud rate.
When the stop bit is received, a reception complete interrupt request signal (INTUAnR) is generated and, at the same time, the data of the UARTAn receive shift register is written to the UAnRX register. If an overrun error occurs (indicated by the UAnOVE bit of the UAnSTR register), the receive data is not written to the UAnRX register.
Even if a parity error (indicated by the UAnPE bit of the UAnSTR register) or framing error (indicated by the UAnFE bit of the UAnSTR register) occurs in the middle of reception, reception continues to the reception position of the stop bit. The INTUAnR signal is generated when reception is completed.

Figure 12-13: UART Reception



Cautions: 1. Be sure to read the UAnRX register even when a reception error occurs. Unless the UAnRX register is read, an overrun error occurs when the next data is received, and the reception error status persists.

2. It is always assumed that the number of stop bits is 1 during reception. A second stop bit is ignored.

3. When reception is completed, read the UAnRX register after the reception complete interrupt (INTUAnR) has been generated and before clearing the UAnPWR or UAnRXE bit to 0. If UAnPWR or UAnRXE is cleared to 0 before the interrupt occurred, the read value of UAnRX is undefined.

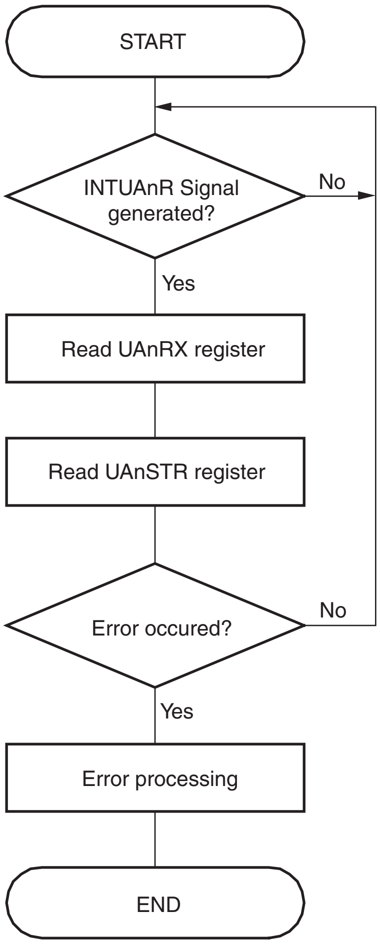
12.5.5 Reception errors

Reception errors are classified into three types: parity errors, framing errors, and overrun errors. As a result of receiving data, an error flag is set in the UAnSTR register, and a reception complete interrupt request signal (INTUAnR) is generated.
By reading the contents of the UAnSTR register in the reception error interrupt servicing, which error has occurred during reception can be checked.
The reception error flag is cleared by writing 0 to it.

Table 12-4: Reception Error Causes

Error Flag	Reception Error	Cause
UAnPE	Parity error	Received parity bit does not match setting.
UAnFE	Framing error	Stop bit is not detected.
UAnOVE	Overrun error	Next data reception is completed before data is read from receive buffer.

Figure 12-14: Receive Data Read Flow



When a reception errors occur, perform the following procedures depending upon the type of error:

- Parity error
  If false data is received due to problems such as noise in the reception line, discard the received data and retransmit.

- Framing error
  A baud rate error may have occurred between the reception side and transmission side or the start bit may have been erroneously detected. Since this is a fatal error for the communication format, check the operation stop in the transmission side, perform initialization processing each other, and then start the communication again.

- Overrun error
  Since the next reception is completed before reading receive data, 1 frame of data is discarded. If this data was needed, do a retransmission.

Caution: If a receive error interrupt occurs during continuous reception, read the contents of the UAnSTR register must be read before the next reception is completed, then perform error processing.

12.5.6 Types and operation of parity

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission side and reception side.
Even parity and odd parity can be used to detect a “1” bit error (odd number). With zero parity and no parity, no errors are detected.

(1) Even parity

(a) During transmission

The number of bits that are “1” in the transmit data, including the parity bit, is controlled to be even. The value of the parity bit is as follows.

- Number of bits that are “1” in transmit data is odd: 1
- Number of bits that are “1” in transmit data is even: 0

(b) During reception

The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(2) Odd parity

(a) During transmission

Opposite to even parity, the number of bits that are “1” in the transmit data, including the parity bit, is controlled to be odd. The value of the parity bit is as follows.

- Number of bits that are “1” in transmit data is odd: 0
- Number of bits that are “1” in transmit data is even: 1

(b) During reception

The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(3) 0 parity

The parity bit is cleared to 0 during transmission, regardless of the transmit data.
The parity bit is not checked during reception. Therefore, a parity error does not occur regardless of whether the parity bit is 0 or 1.

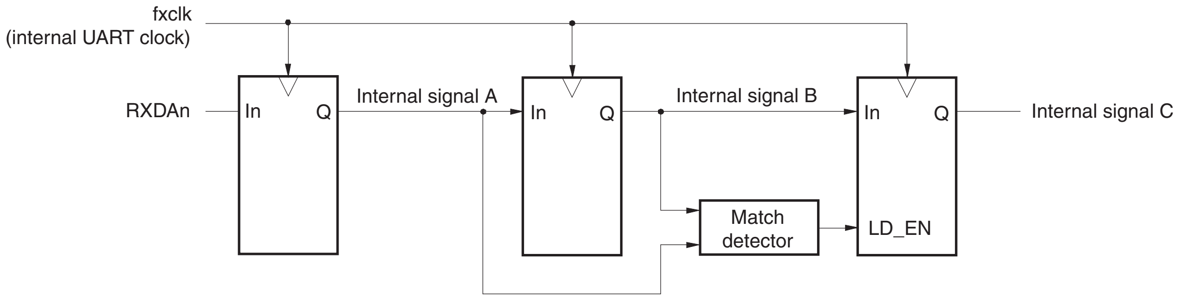
(4) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit. Because no parity bit is used, a parity error does not occur.

12.5.7 Noise filter of receive data

The RXDAn pin is sampled using the UART internal clock (f_{XCLK}).
If the sampled value is the same twice in a row, the output of the match detector changes, and the signal on the RXDAn pin is sampled as input data.
Because the circuit configuration of the noise filter is as shown in Figure 12-15, internal processing of a reception operation is delayed two clocks from the external signal status.

Figure 12-15: Noise Filter Circuit

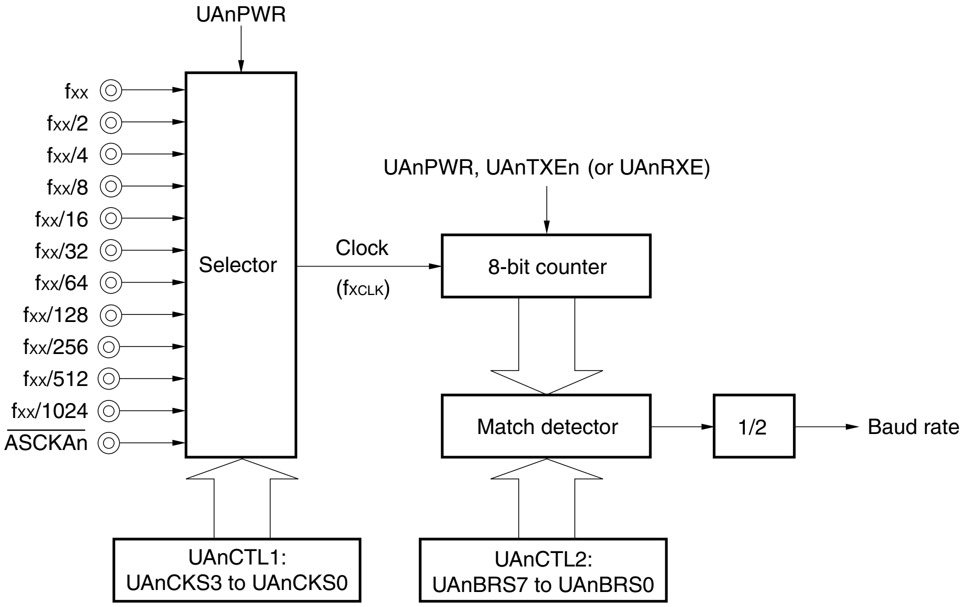


12.6 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector and 8-bit programmable counters, and generates a serial clock for transmission/reception by UARTAn. The output of the dedicated baud rate generator can be selected as the serial clock on a channel by channel basis.
8-bit counters are provided separately for transmission and reception.

(1) Configuration of baud rate generator

Figure 12-16: Configuration of Baud Rate Generator



Remarks: 1. n = 0 to 1

2. f_{XX}: Internal system clock

(a) Base clock

The clock selected by the UAnCKS3 to UAnCKS0 bits of the UAnCTL1 register is supplied to the 8-bit counter when the UAnPWR bit of the UAnCTL0 register is 1. This clock is called the base clock, and its frequency is called f_{XCLK}. When the UAnPWR bit is 0, the base clock is fixed to the low level.

(b) Generation of serial clock

A serial clock can be generated in accordance with the setting of the UAnCTL1 and UAnCTL2 registers (n = 0 to 1).
The base clock is selected by using the UAnCKS3 to UAnCKS0 bits of the UAnCTL1 register.
The division ratio of the 8-bit counter can be selected by using the UAnBRS7 to UAnBRS0 bits of the UAnCTL2 register.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is used to select the clock for UARTAn.
For details, refer to **12.3 (2) UARTAn control register 1 (UAnCTL1**).

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is used to select the baud rate (serial transfer rate) clock for UARTAn.
For details, refer to **12.3 (3) UARTAn control register 2 (UAnCTL2)**.

(4) Baud rate

The baud rate can be calculated by the following expression.

$$\text{Baud rate} = \frac{f_{XCLK}}{2 \times k} \text{ [bps]}$$

f_{XCLK} = Frequency of base clock selected by UAnCKS3 to UAnCKS0 bits of UAnCTL1 register
k = Value set by UAnBRS7 to UAnBRS0 bits of UAnCTL2 register (k = 4, 5, 6, ..., 255)

(5) Error of baud rate

The baud rate error is calculated by the following expression.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1\right) \times 100$$

Cautions: 1. Keep the baud rate error on the transmission side to within the permissible error on the reception side.

2. Keep the baud rate error on the reception side to within the range described in (7) Permissible baud rate range for reception.

Example:

Frequency of base clock = 32 MHz
Selection of $f_{XX}/2$ as base clock
Set value of UAnBRS7 to UAnBRS0 bits of UAnCTL2 register = 01101000B (k = 52)
Target baud rate = 153600 bps

Baud rate = (32000000 / 2) / (2 ¥ 52)
= 153846 [bps]

Error = (153846/153600 – 1) ¥ 100
= 0.160 [%]

(6) Example of baud rate setting

Table 12-5: Baud Rate Generator Set Data

Baud Rate (bps)	f_{XX} = 40 MHz			f_{XX} = 32 MHz			f_{XX} = 24 MHz		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	0AH	41H	0.16	0AH	34H	0.16	0AH	27H	0.16
600	09H	41H	0.16	0AH	1AH	0.16	09H	27H	0.16
1200	08H	41H	0.16	0AH	0DH	0.16	08H	27H	0.16
2,400	07H	41H	0.16	09H	0DH	0.16	07H	27H	0.16
4,800	06H	41H	0.16	08H	0DH	0.16	06H	27H	0.16
9,600	05H	41H	0.16	07H	0DH	0.16	05H	27H	0.16
19,200	04H	41H	0.16	06H	0DH	0.16	04H	27H	0.16
31,250	07H	05H	0.00	07H	04H	0.00	06H	06H	0.00
38,400	03H	41H	0.16	05H	0DH	0.16	03H	27H	0.16
57,600	02H	41H	-0.22	01H	8BH	-0.08	04H	0DH	0.16
76,800	02H	41H	0.16	04H	0DH	0.16	02H	27H	0.16
115,200	01H	57H	-0.22	00H	8BH	-0.08	03H	0DH	0.16
153,600	01H	41H	0.16	03H	0DH	0.16	01H	27H	0.16
230,400	00H	57H	-0.22	00H	45H	0.64	02H	0DH	0.16
312,500	04H	04H	0.00	00H	33H	0.39	01H	13H	1.05

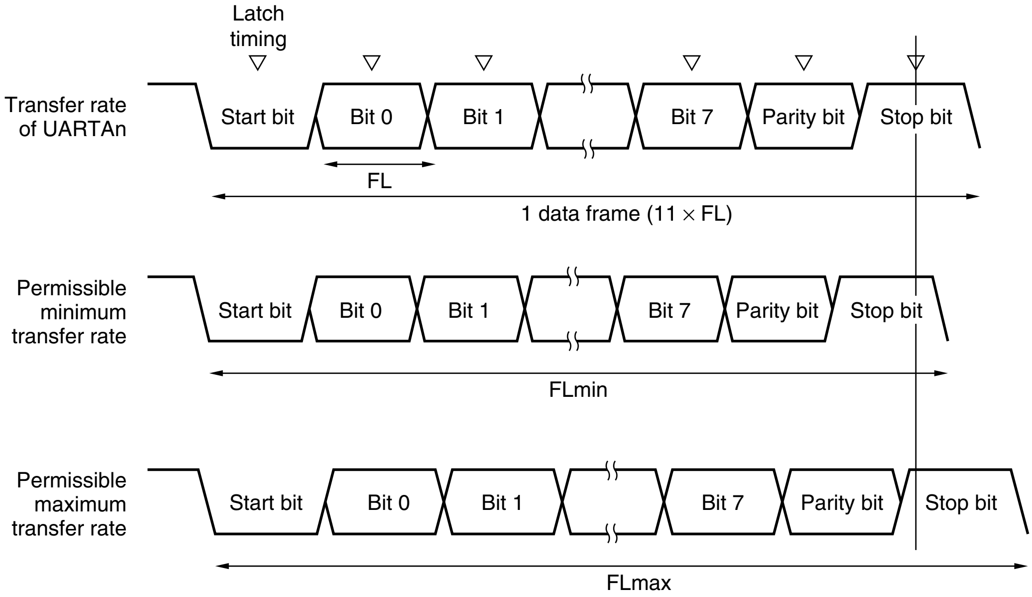
Remark: f_{XX}: Internal system clock
ERR: Baud rate error [%]

(7) Permissible baud rate range for reception

The permissible baud rate error during reception is shown below.

Caution: Be sure to set the baud rate error for reception to within the permissible error range, by using the expressions shown below.

Figure 12-17: Permissible Baud Rate Range for Reception



After the start bit is detected, the counter set by the UAnCTL2 register determines the latch timing of the receive data, as shown in Figure 12-17. If the last data (stop bit) is received at this latch timing, the data can be correctly received.
Assuming 11 bits of data are to be received, the theoretical baud rate is as follows.

FL = (Brate) – 1

Brate: Baud rate of UARTAn (n = 0 to 1)
k: Set value of UAnCTL2 (n = 0 to 1)
FL: 1-bit data length
Margin of latch timing: 2 clocks

Permissible minimum transfer rate:

$$\mathrm{Flmin} = 11 \times \mathrm{FL} - \frac{k - 2}{2k} \times \mathrm{FL} = \frac{21k + 2}{2k}\ \mathrm{FL}$$

Therefore, the maximum receivable baud rate on the transmission side is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k + 2}\ \text{Brate}$$

Similarly, the permissible maximum transfer rate can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k + 2}{2 \times k} \times \text{FL} = \frac{21k\text{- }2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k\text{- }2}{20k}\ \text{FL} \times 11$$

The minimum receivable baud rate on the transmission side is as follows.

$$\text{BRmin} = (\text{Flmax}/11)^{-1} = \frac{20k}{21k - 2}\ \text{Brate}$$

The permissible error of the baud rate from the transmission side of UARTAn can be calculated by the above expressions of the minimum/maximum baud rate. The result is as shown in Table 12-6.

Table 12-6: Permissible Maximum/Minimum Baud Rate Error

Division Ratio (k)	Permissible Maximum Baud Rate Error	Permissible Minimum Baud Rate Error
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%

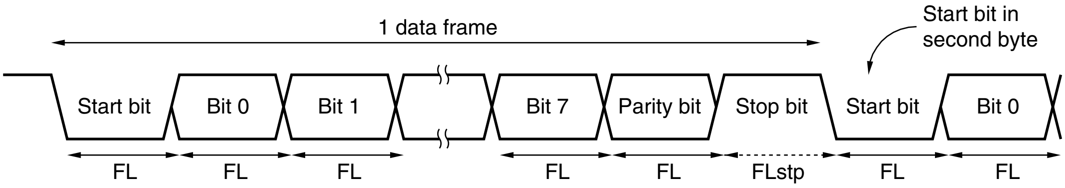
Remarks: 1. The reception accuracy is dependent upon the number of bits in 1 frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the accuracy.

2. k: Set value of UAnCTL2 (n = 0 to 1)

(8) Transfer rate for continuous transmission

The transfer rate from the stop bit to the start bit of the next data is extended two clocks when continuous transmission is executed. However, the timing on the reception side is initialized when the start bit is detected, and therefore, the transfer result is not affected.

Figure 12-18: Transfer Rate for Continuous Transmission



Where 1 bit data length is FL, stop bit length is FLstp, and base clock frequency is f_{XCLK}, the stop bit length can be calculated by the following expression.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate for continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/f_{XCLK}$$

12.7 Caution for Use in On-chip Debug Mode

When a break command is detected while UARTA received data in on-chip debug mode, an overrun error is generated.

Chapter 13 3-Wire Serial Interface (CSIB)

The V850E/RS1 includes two 3-wire serial interfaces (CSIB).

13.1 Features

- Master mode and slave mode selectable
- 3-wire serial interface for 8-bit to 16-bit transfer
- Interrupt request signals (INTCBnT and INTCBnR)
- Serial clock and data phase selectable
- Transfer data length selectable from 8 to 16 bits in 1-bit units
- Data transfer with MSB- or LSB-first selectable
- 3-wire

3-wire	SOBn:	Serial data output
	SIBn:	Serial data input
	$\overline{\text{SCKBn}}$:	Serial clock I/O

- Transmission mode, reception mode, and transmission/reception mode selectable

Remark: n = 0, 1

13.2 Configuration

CSIB consists of the following hardware:

Table 13-1: Configuration of CSIB0, CSIB1

Item	Configuration
Register	CSIBn reception data register (CBnRX) CSIBn transmit data register (CBnTX)
Reception data input	2 (SIB0, SIB1)
Transmit data output	2 (SOB0, SOB1)
Serial clock input/output	2 ($\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$)
Control register	CSIBn control register 0 to 2 (CBnCTL0 to CBnCTL2) CSIBn status register (CBnSTR)

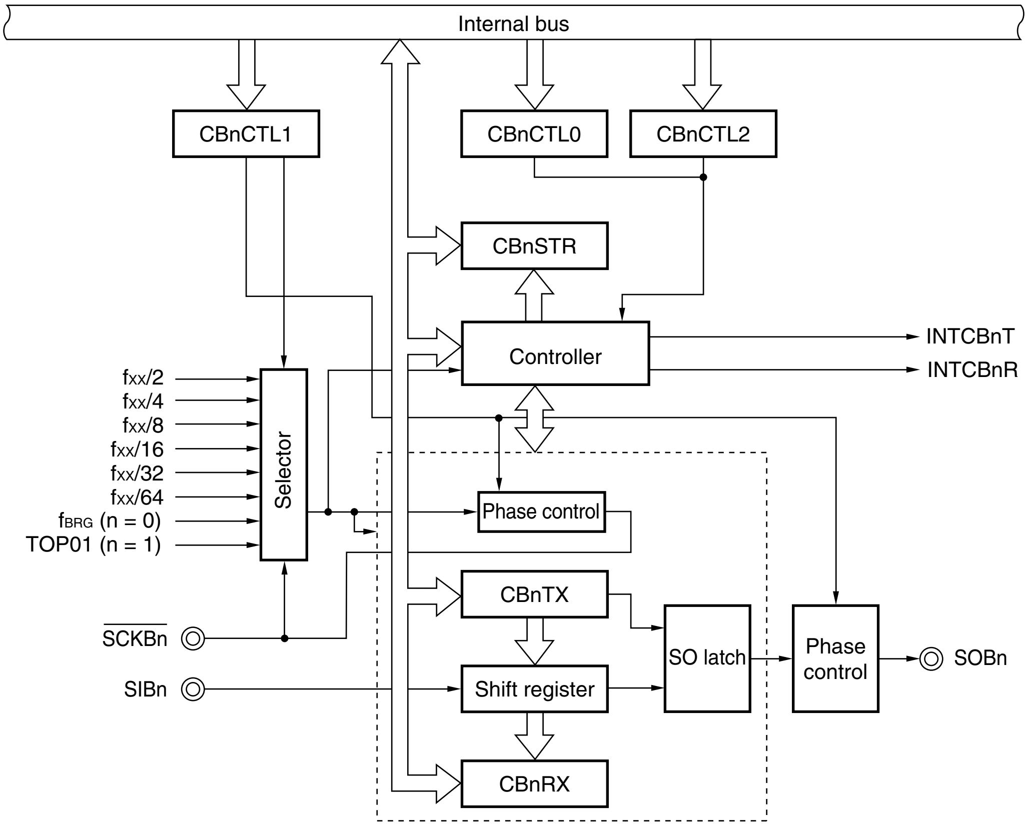
Remark: n = 0, 1

The pins of the 3-wire serial interface (CSIB) function alternately as port pins. For how to select the alternate function, refer to the descriptions of the registers in **Chapter 4 "Port Functions" on page 105**.

Table 13-2: List of 3-Wire Serial Interface Pins

Pin Name	Alternate-Function Pin	I/O	Function
SIB0	P40	Input	Serial receive data input (CSIB0)
SIB1	P97		Serial receive data input (CSIB1)
SOB0	P41	Output	Serial transmit data input (CSIB0)
SOB1	P98		Serial transmit data input (CSIB1)
$\overline{\text{SCKB0}}$	P42	I/O	Serial clock I/O (CSIB0)
$\overline{\text{SCKB1}}$	P99		Serial clock I/O (CSIB1)

Figure 13-1: Block Diagram of 3-Wire Serial Interface



Remark: n = 0, 1

13.3 Control Registers Overview

(1) CSIBn control register 0 (CBnCTL0)

The CBnCTL0 register is an 8-bit register that specifies the operation of the 3-wire serial interface.

(2) CSIBn control register 1 (CBnCTL1)

The CBnCTL1 register is an 8-bit register that selects the transmission/reception timing and input clock of the 3-wire serial interface.

(3) CSIBn control register 2 (CBnCTL2)

This is an 8-bit register that controls the number of serial transfer bits of CSIB.

(4) CSIBn status register (CBnSTR)

The CBnSTR register is a collection of flags that indicate the nature of a reception error that has occurred. Each error flag is set to 1 if the corresponding reception error occurs.

(5) CSIBn receive data register (CBnRX)

The CBnRX register is a 16-bit buffer register that holds receive data.
When data is transferred to the CBnRX register, a reception complete interrupt request signal (INTCBnR) is generated.

(6) CSIBn transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register that holds transmit data.
When data is transferred to the CBnTX register, a transmission enable interrupt request signal (INTCBnT) is generated.

13.4 Control Registers Description

(1) CSIBn control register 0 (CBnCTL0)

This register controls the serial transfer operation of CSIB.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 01H.

Figure 13-2: CSIBn Control Register 0 (CBnCTL0) Format (1/2)

Address: CB0CTL0: FFFFFD00H, CB1CTL0: FFFFFD10H

Symbol	7	6	5	4	3	2	1	0	R/W	After Reset
CBnCTL0	CBnPWR	CBnTXE$^{\text{Note}}$	CBnRXE$^{\text{Note}}$	CBnDIR$^{\text{Note}}$	0	0	CBnTMS$^{\text{Note}}$	CBnSCE	R/W	01H

Remark: n = 0, 1

CBnPWR	Specification of CSIB operation stop or enable
0	Stop clock operation (asynchronously reset CSIBn).
1	Enable clock operation.
The CBnPWR bit controls the operating clock of CSIB and resets the internal circuit.	

CBnTXE$^{\text{Note}}$	Specification of transmission operation stop or enable
0	Stop transmission.
1	Enable transmission.
When the CBnTXE bit is cleared to 0, the serial output pin SOBn is fixed to the low level and communication is stopped.	

CBnRXE$^{\text{Note}}$	Reception operation enable
0	Stop reception.
1	Enable reception.
When the CBnRXE bit is cleared to 0, because reception is stopped, the reception complete interrupt is not output and the receive data in the CBnRX register is not updated even if the specified data is transferred.	

CBnDIR$^{\text{Note}}$	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

Note: This register can be rewritten only when the CBnPWR bit = 0. However, the CBnPWR bit can also be set to 1 at the same time as rewriting these bits.

Figure 13-2: CSIBn Control Register 0 (CBnCTL0) Format (2/2)

CBnTMS[Note]	Specification of transfer mode
0	Single transfer mode
1	Continuous transfer mode
When the CBnTMS bit = 0, the single transfer mode is set in which continuous transmission/reception is not supported. Even when only transmission is executed, an interrupt is output on completion of reception transfer.	

CBnSCE	Specification of start transfer disable or enable
0	Disable transfer operation.
1	Enable transfer operation.
The CBnSCE bit controls starting of the transfer operation in the master mode. If only reception is enabled in the single transfer mode (CBnRXE bit = 1, CBnTXE bit = 0), the reception operation is started when the CBnRX register is read. To read the last receive data, clear the CBnSCE bit to 0, read the last receive data, and then disable starting the next reception operation. Similarly, if only reception is enabled in the continuous transfer mode (CBnTMS bit = 1), starting the reception operation can be disabled after completion of reception of the last receive data, by clearing the CBnSCE bit to 0 one clock before reception of the last receive data is completed. After the last data is read, reception is enabled by setting the CBnSCE bit to 1 again and reading the CBnRX register. In the slave reception mode, the CBnSCE bit also enables the internal operating clock, and therefore, set the CBnSCE bit to 1.	

Note: This register can be rewritten only when the CBnPWR bit = 0. However, the CBnPWR bit can also be set to 1 at the same time as rewriting these bits

(2) CSIBn control register 1 (CBnCTL1)

This is an 8-bit register that selects the transmission/reception timing and input clock of CSIBn.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Caution: The CBnCTL1 register can be rewritten only when the CBnPWR bit of the CBnCTL0 register is 0.

Figure 13-3: CSIBn Control Register 1 (CBnCTL1) Format (1/2)

Address: CB0CTL1: FFFFFD01H, CB1CTL1: FFFFFD11H

Symbol	7	6	5	4	3	2	1	0	R/W	After Reset
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0	R/W	00H

Remark: n = 0, 1

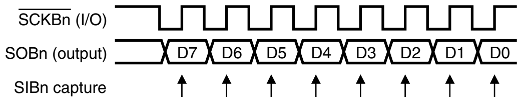
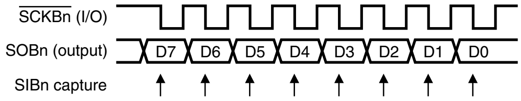
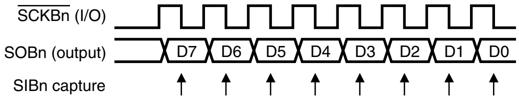
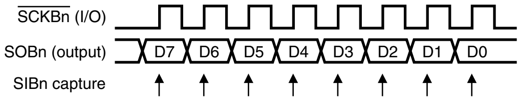
CBnCKP	CBnDAP	Specification of transmission/reception timing of data of $\overline{\text{SCKBn}}$
0	0	 $\overline{\text{SCKBn}}$ (I/O) SOBn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIBn capture
0	1	 $\overline{\text{SCKBn}}$ (I/O) SOBn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIBn capture
1	0	 $\overline{\text{SCKBn}}$ (I/O) SOBn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIBn capture
1	1	 $\overline{\text{SCKBn}}$ (I/O) SOBn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIBn capture

Figure 13-3: CSIBn Control Register 1 (CBnCTL1) Format (2/2)

CBnCKS2	CBnCKS1	CBnCKS0	Input clock		Mode
			n = 0	n = 1	
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**	TMP0 (TOP01)	Master mode
1	1	1	External clock ($\overline{\text{SCKBn}}$)		Slave mode

Note: f_{BRG}: Output clock frequency of prescaler 3
For details of the prescaler, refer to **13.9 ”Prescaler 3” on page 461**.

(3) CSIBn control register 2 (CBnCTL2)

This is an 8-bit register that controls the number of serial transfer bits of CSIB.
It can be read or written in 8-bit units.
Reset input clears this register to 00H.

Caution: The CBnCTL2 register can be rewritten when the CBnPWR bit of the CBnCTL0 register = 0 or when the CB0TXE and CB0RXE bits = 0.

Figure 13-4: CSIBn Control Register 2 (CBnCTL2) Format

Address: CB0CTL2: FFFFFD02H, CB1CTL2: FFFFFD12H

Symbol	7	6	5	4	3	2	1	0	R/W	After Reset
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0	R/W	00H

Remark: n = 0, 1

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Bit length of serial register
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	x	x	x	16 bits

Caution: If the number of transfer bits is not 8 or 16, prepare data, justifying it to the least significant bit of the CBnTX or CBnRX register.

(4) CSIBn status register (CBnSTR)

This is an 8-bit register that indicates the status of CSIB.
Although this register can be read or written in 8-bit or 1-bit units, the CBnSTF flag is read-only.
Reset input clears this register to 00H.
Clearing the CBnPWR bit of the CBnCTL0 register to 0 also initializes this register.

Figure 13-5: CSIBn Status Register (CBnSTR) Format

Address: CB0STR: FFFFFD03H, CB1STR: FFFFFD13H

Symbol	7	6	5	4	3	2	1	0	R/W	After Reset
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE	R/W	00H

Remark: n = 0, 1

CBnTSF	Transfer operation status flag
0	Idle status
1	Operating status
This bit is set when data is prepared in the CBnTX register for transmission. It is set when dummy data is read from the CBnRX register for reception. It is cleared when the edge of the last clock is completed.	

CBnOVE	Overrun error flag
0	No overrun
1	Overrun
• An overrun error occurs if the next reception is started without the CPU reading the value of the receive buffer during reception or on completion of a reception operation. • The CBnOVE flag indicates occurrence of this overrun error. • The OBnOVE flag is cleared when 0 is written to it. It is not set when 1 is written to it.	

(5) 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.
If reception is enabled, a reception operation is started when the CBnRX register is read.
If the transfer data length is 8 bits, the lower 8 bits of the CBnRX register are read-only in 8-bit units as the CBnRXL register.
Reset input clears this register to 0000H.
Clearing the CBnPWR bit of the CBnCTL0 register to 0 also initializes this register.

Figure 13-6: CSIBn Receive Data Register (CBnRX) Format

Address: CB0RX: FFFFFD04H, CB1RX: FFFFFD14H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After Reset
CBnRX																	R	0000H

Remark: n = 0, 1

(6) CSIBn transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register to which transfer data of CSIB is written.
This register can be read or written in 16-bit units.
If transmission is enabled, a transmission operation is started when the CBnTX register is written.
If the transfer data length is 8 bits, the lower 8 bits of the CBnTX register can be read or written in 8-bit units as the CBnTXL register.
Reset input clears this register to 0000H.

Figure 13-7: CSIBn Transmit Data Register (CBnTX) Format

Address: CB0TX: FFFFFD06H, CB1TX: FFFFFD16H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After Reset
CBnTX																	R/W	0000H

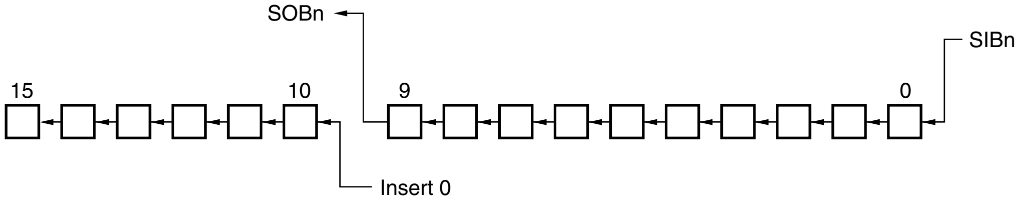
Remark: n = 0, 1

13.5 Transfer Data Length Change Function

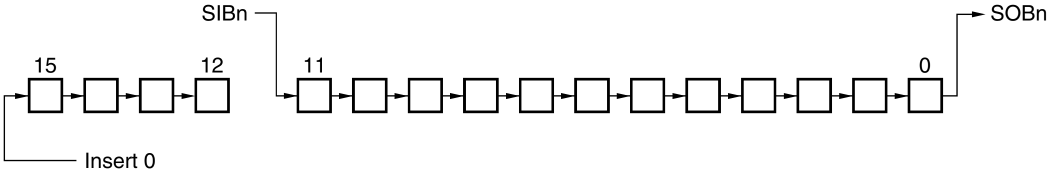
The transfer data length of CSIB can be changed from 8 to 16 bits in 1-bit units by using the CBnCL3 to CBnCL0 bits of the CBnCTL2 register.
If a transfer data length of other than 16 bits is specified, set data in the CBnTX or CBnRX register, justifying to the least significant bit, regardless of whether the first transfer bit is the MSB or LSB. Any data can be set to the higher bits that are not used, but the receive data is 0 after serial transfer.

Figure 13-8: Changing Transfer Data Length

(a) When transfer bit length = 10 bits, MSB first



(b) When transfer bit length = 12 bits, LSB first



13.6 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 13-3: 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 a reception error occurs, instead of a reception error interrupt.
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.
The reception complete interrupt request signal is not generated while reception is disabled.

(2) Transmission enable interrupt request signal (INTCBnT)

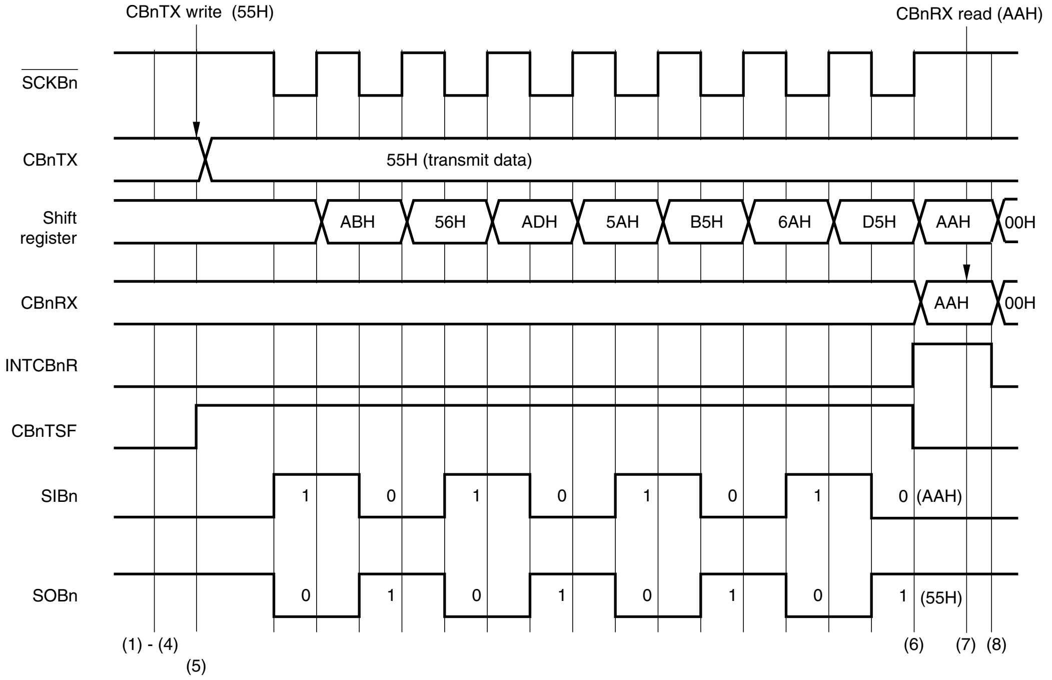
The transmission enable interrupt request signal is generated when transmit data is transferred from the CBnTX register while transmission enabled.

13.7 Operation

13.7.1 Single transfer mode (master mode, transmission/reception mode)

Figure 13-9 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 0, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-9: Single Transfer Timing (Master Mode, Transmission/Reception Mode)



(1) Clear the CBnPWR bit of the CBnCTL0 register.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable transmission/reception by setting the CBnTXE, CBnRXE and CBnCSE bits of the CBnCTL0 register to 1.
(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.
(5) Write transfer data to the CBnTX register (start transmission).
(6) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX (CBnRXL) register can be read.
(7) Read the CBnRX register before clearing the CBnPWR bit to 0.
(8) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0 (end of transmission/reception).

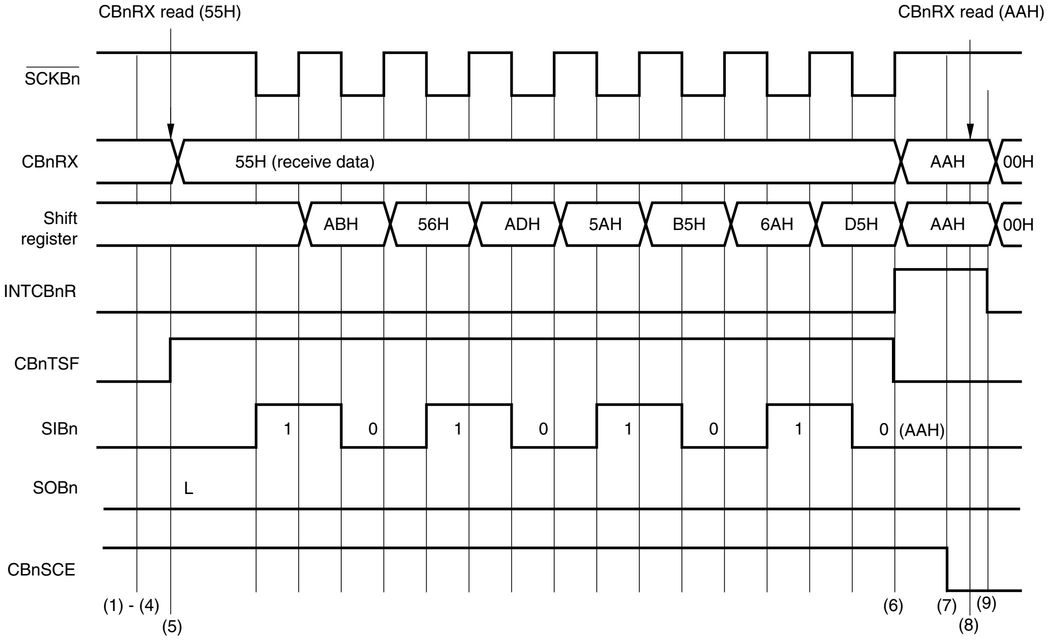
To transfer more data, repeat (5) to (7) before (8).

Remark: The processing in (3) and (4) can be set simultaneously.

13.7.2 Single transfer mode (master mode, reception mode)

Figure 13-10 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 0, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-10: Single Transfer Timing (Master Mode, Reception Mode)



(1) Clear the CBnPWR bit of the CBnCTL0 register.
(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.
(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable reception by setting the CBnRXE and CBnCSE bits of the CBnCTL0 register to 1.
(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.
(5) Read dummy data from the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX (CBnRXL) register can be read.
(7) Get the last receive data ready by clearing the CBnSCE bit of the CBnCTL0 register to 0.
(8) Read the CBnRX register before clearing the CBnPWR bit to 0.
(9) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0 (end of reception).

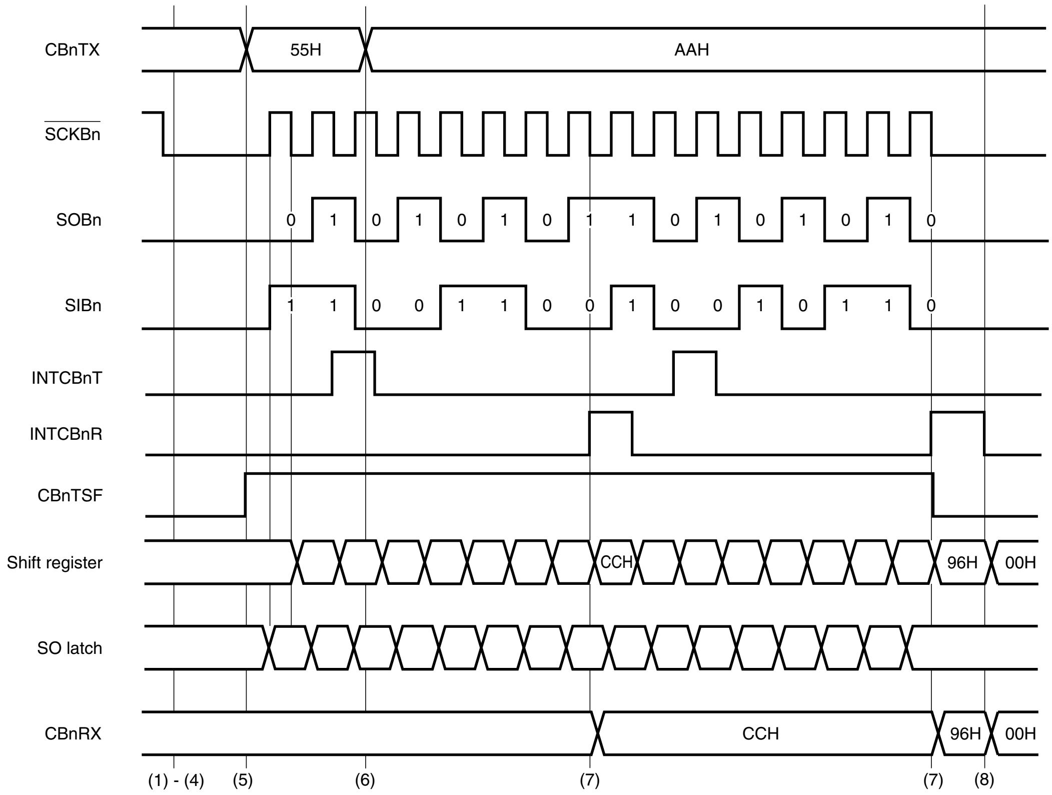
To transfer more data, repeat (5) and (6) before (7) (at this time, read receive data that is also used as a reception trigger, instead of dummy data).

Remark: The processing in (3) and (4) can be set simultaneously.

13.7.3 Continuous mode (master mode, transmission/reception mode)

Figure 13-11 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when continuous transfer mode (CBnTMS bit of CBnCTL0 register = 1), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 0, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-11: Continuous Transfer Timing (Master Mode, Transmission/Reception Mode)



(1) Clear the CBnPWR bit of the CBnCTL0 register.

(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.

(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable transmission/reception by setting the CBnTXE, CBnRXE and CBnCSE bits of the CBnCTL0 register to 1.

(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.

(5) Write transfer data to the CBnTX register (start transmission).

(6) Write the transfer data to the CBnTX register when the transmission enable interrupt request signal (INTCBnT) is generated.

(7) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX (CBnRXL) register can be read. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.

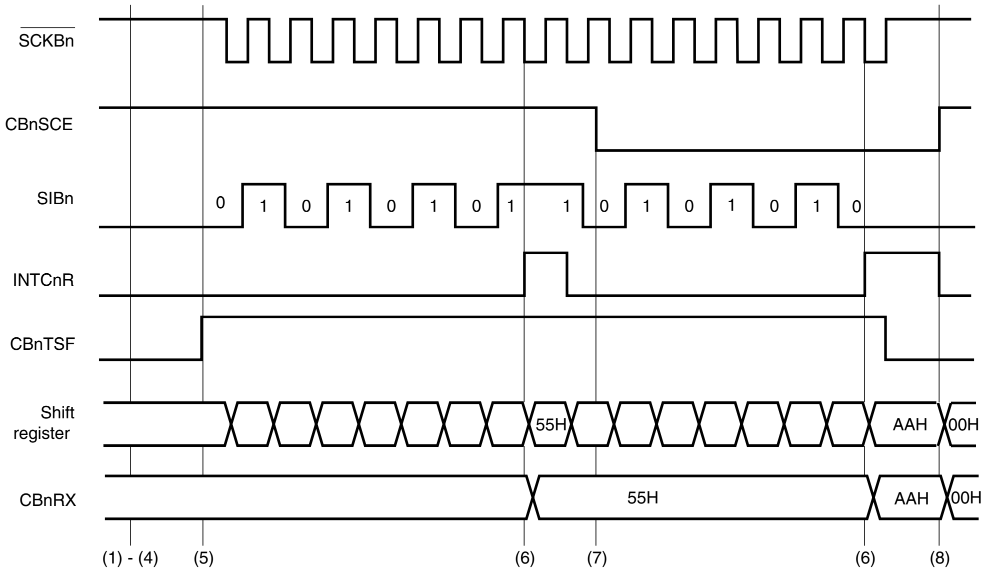
(8) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0.

To transfer more data, repeat (5) to (7) before (8).

13.7.4 Continuous mode (master mode, reception mode)

Figure 13-12 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when continuous transfer mode (CBnTMS bit of CBnCTL0 register = 1), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 1, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-12: Continuous Transfer Timing (Master Mode, Reception Mode)



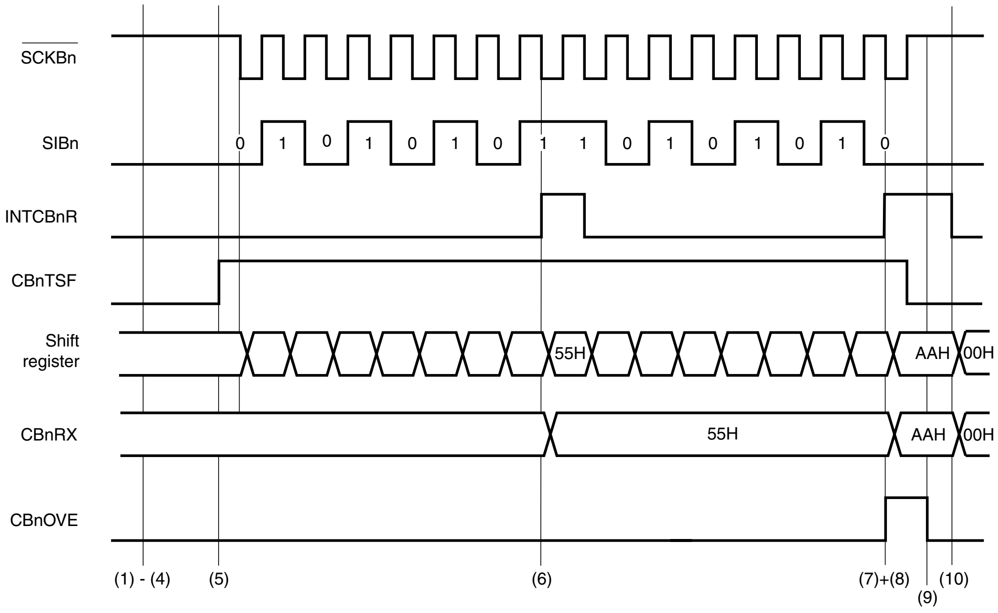
(1) Clear the CBnPWR bit of the CBnCTL0 register.
(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.
(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable reception by setting the CBnRXE bit of the CBnCTL0 register to 1.
(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.
(5) Read dummy data from the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX (CBnRXL) register can be read. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(7) Prepare the last receive data by clearing the CBnSCE bit of the CBnCTL0 register to 0.
(8) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0 (end of reception).

To transfer more data, repeat (5) and (6) before (7).

13.7.5 Continuous reception mode (error)

Figure 13-13 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when continuous transfer mode (CBnTMS bit of CBnCTL0 register = 1), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 1, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-13: Continuous Transfer Timing (Error)



(1) Clear the CBnPWR bit of the CBnCTL0 register.

(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.

(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable reception by setting the CBnRXE bit of the CBnCTL0 register to 1.

(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.

(5) Read dummy data from the CBnRX register (reception start trigger).

(6) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX (CBnRXL) register can be read.

(7) If the data cannot be read before the next transfer is completed, the CBnOVE flag of the CBnSTR register is set on completion of reception, and the reception complete interrupt request signal (INTCBnR) is generated. Read the CBnRX register before the next receive data arrives.

(8) Confirm that the CBnOVE bit = 1, in the INTCBnR interrupt processing, and perform overrun error processing.

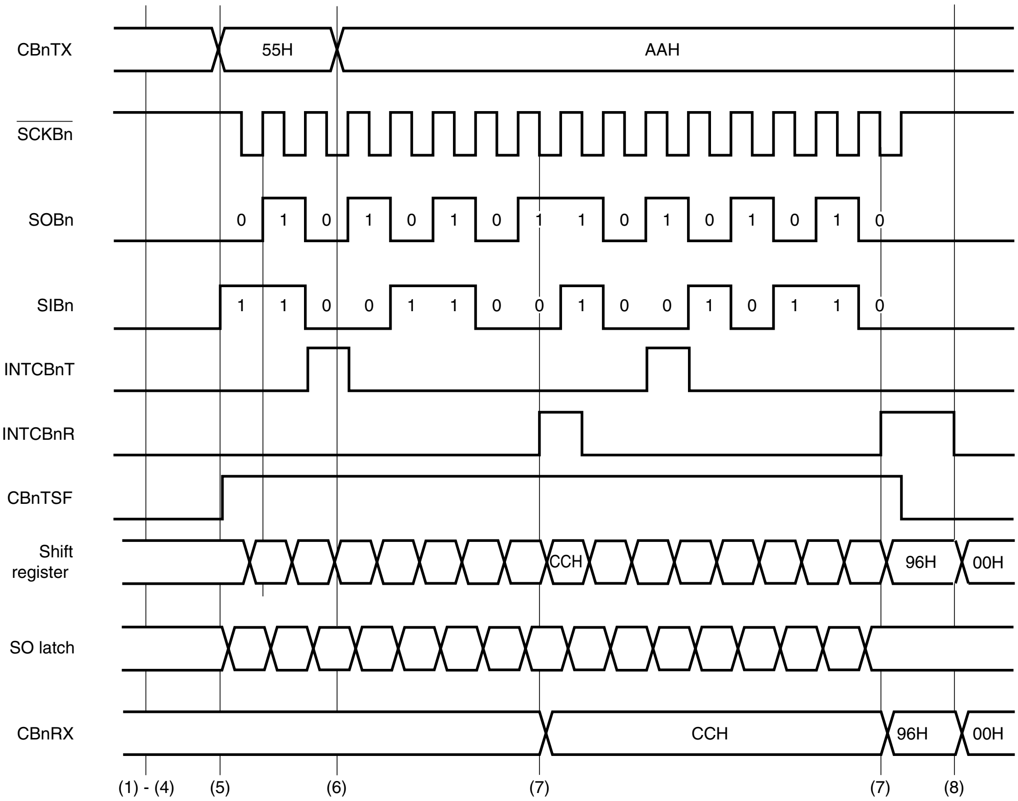
(9) Clear the CBnOVE bit to 0.

(10) Confirm that the CBnTSF bit of the CBnSTR register = 0. Clear the CBnPWR bit to 0 and stop clock supply to CSIB.

13.7.6 Continuous mode (slave mode, transmission/reception mode)

Figure 13-14 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when continuous transfer mode (CBnTMS bit of CBnCTL0 register = 1), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 1, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-14: Continuous Transfer Timing (Slave Mode, Transmission/Reception Mode)



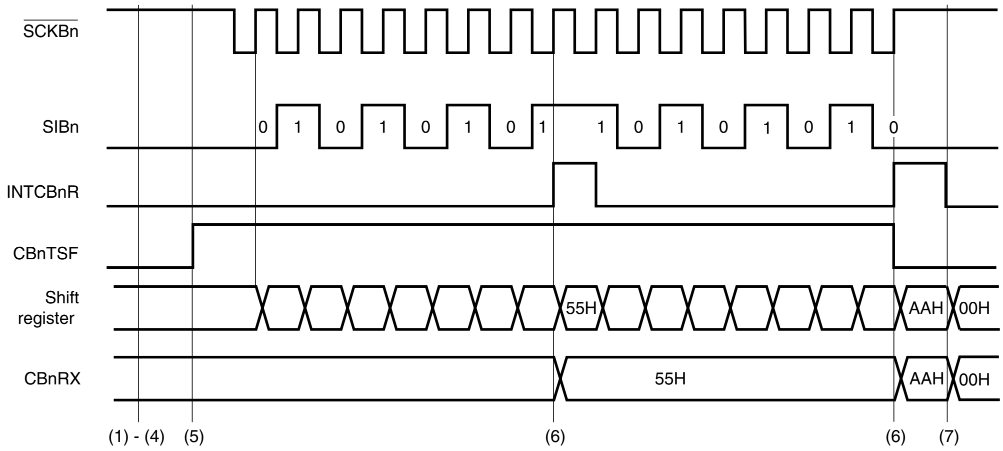
(1) Clear the CBnPWR bit of the CBnCTL0 register.
(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.
(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable transmission/reception by setting the CBnTXE, CBnRXE and CBnCSE bits of the CBnCTL0 register to 1.
(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.
(5) Write transfer data to the CBnTX.
(6) Write the transfer data to the CBnTX register when the transmission enable interrupt request signal (INTCBnT) is generated.
(7) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX register can be read. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(8) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0 (end of transmission/reception).

To transfer more data, repeat (5) to (7) before (8).

13.7.7 Continuous mode (slave mode, reception mode)

Figure 13-15 shows the transfer timing when data is transferred with the MSB first (CBnDIR bit of CBnCTL0 register = 0), when continuous transfer mode (CBnTMS bit of CBnCTL0 register = 1), when the CBnCKP bit of the CBnCTL1 register = 0, when the CBnDAP bit of the CBnCTL1 register = 0, and when the transfer data length is 8 bits (CBnCL3 to CBnCL0 bits of the CBnCTL2 register = 0, 0, 0, 0).

Figure 13-15: Continuous Transfer Timing (Slave Mode, Reception Mode)



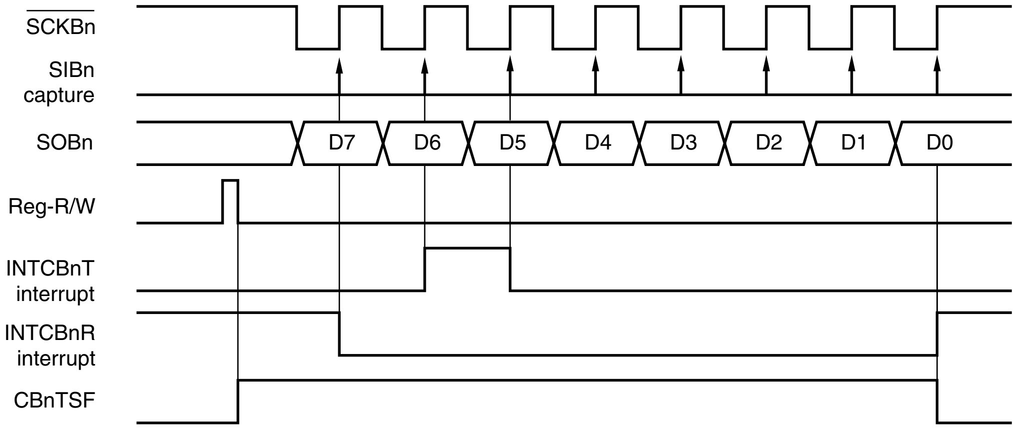
(1) Clear the CBnPWR bit of the CBnCTL0 register.
(2) Specify the transfer mode by setting the CBnCTL1 and CBnCTL2 registers.
(3) Specify the transfer mode by using the CBnDIR bit of the CBnCTL0 register and, at the same time, enable reception by setting the CBnRXE and CBnCSE bits of the CBnCTL0 register to 1.
(4) Enable CSIB operating clock supply by setting the CBnPWR bit of the CBnCTL0 register to 1.
(5) Read dummy data from the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is generated to inform the CPU that the CBnRX register can be read. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(7) Confirm that the CBnTSF bit of the CBnSTR register = 0, and stop clock supply to CSIB by clearing the CBnPWR bit to 0 (end of reception).

To transfer more data, repeat (5) and (6) before (7).

13.7.8 Clock timing

Figure 13-16: Clock Timing (1/2)

(a) CBnCKP = 0, CBnDAP = 0



(b) CBnCKP = 1, CBnDAP = 0

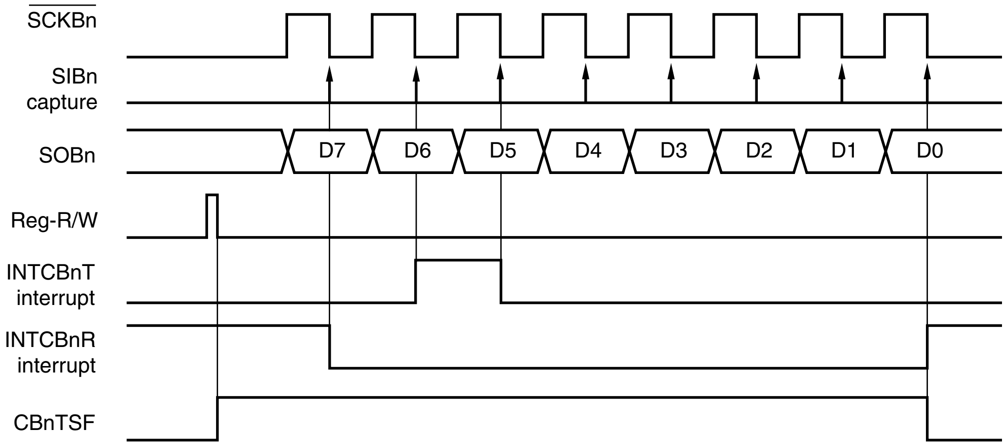
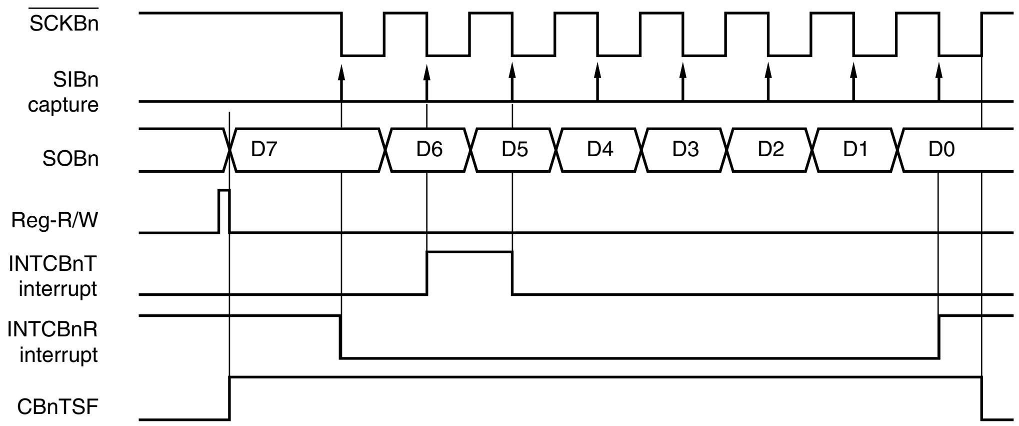
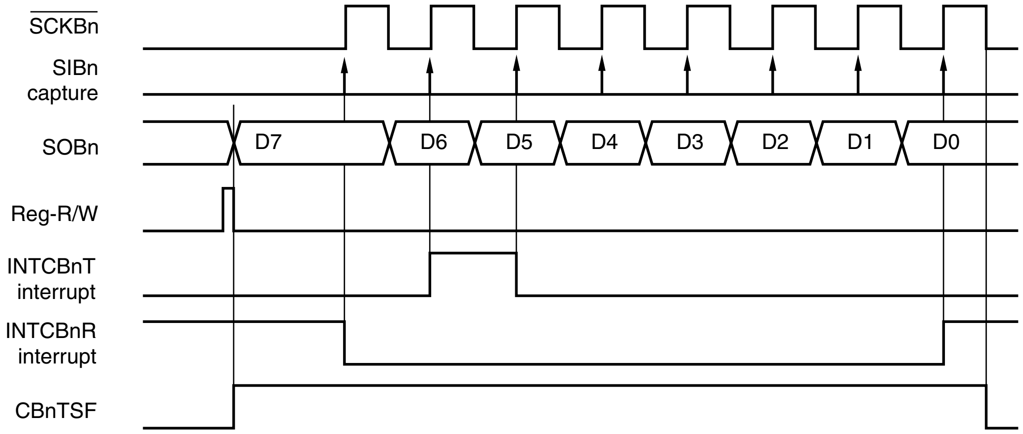


Figure 13-16: Clock Timing (2/2)

(c) CBnCKP = 0, CBnDAP = 1



(d) CBnCKP = 1, CBnDAP = 1



13.7.9 Output pin status with operation disabled

(1) $\overline{\text{SCKBn}}$ pin

The output status of the $\overline{\text{SCKBn}}$ pin is as follows when CSIBn operation is disabled (when the CBnPWR bit of the CBnCTL0 register = 0).

CBnCKP	$\overline{\text{SCKBn}}$ Pin Output
0	Fixed to high level
1	Fixed to low level

Remarks: 1. The output of the $\overline{\text{SCKBn}}$ pin changes if the CBnCKP bit of the CBnCTL1 register is rewritten.

2. n = 0, 1

(2) SOBn pin

The output status of the SOBn pin is as follows when CSIBn operation is disabled
(CBnPWR bit = 0).

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	x	x	Fixed to low level
1	0	x	Value of SOBn latch (low level)
1	1	0	Value of CBnTX (MSB)
1	1	1	Value of CBnTX (LSB)

Remarks: 1. The output of the SOBn pin changes if any of the CBnTXE, CBnDAP, and CBnDIR bits of the CBnCTL1 register is rewritten.

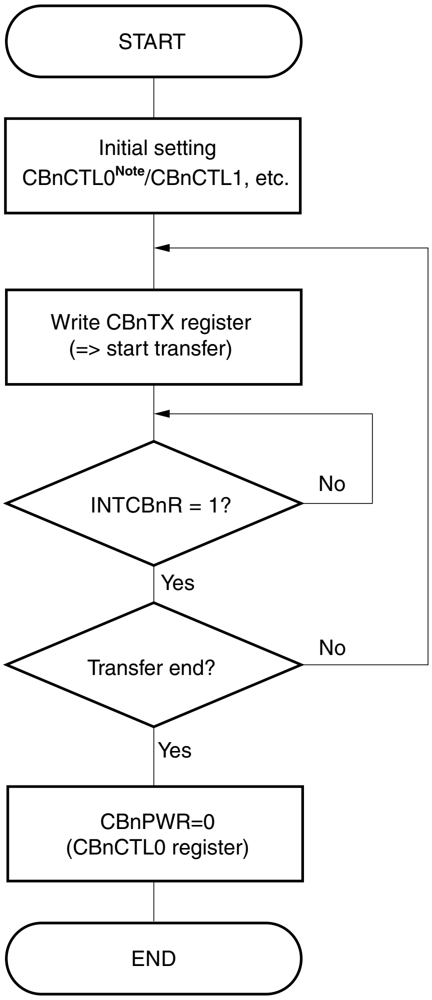
2. n = 0, 1

3. x: don’t care

13.8 Operation Flow

(1) Single transmission

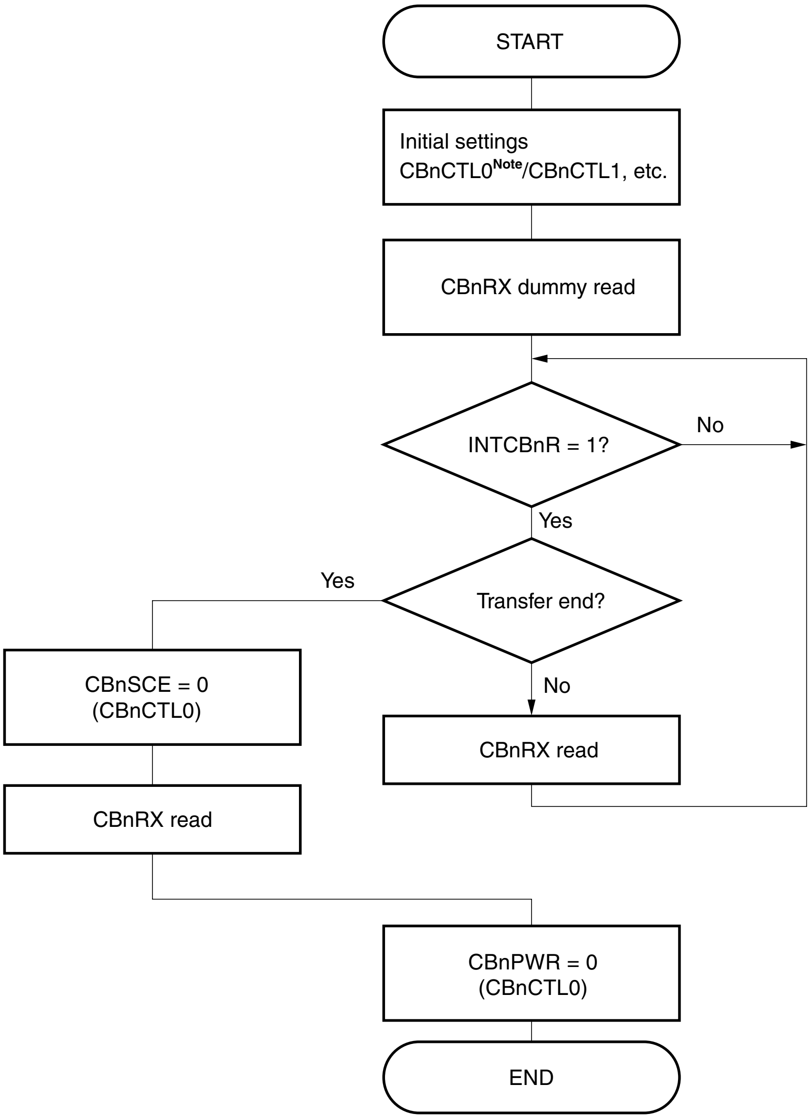
Figure 13-17: Single Transmission Flow



Note: Set the CBnSCE bit to 1 at the initial setting.

(2) Single reception (master)

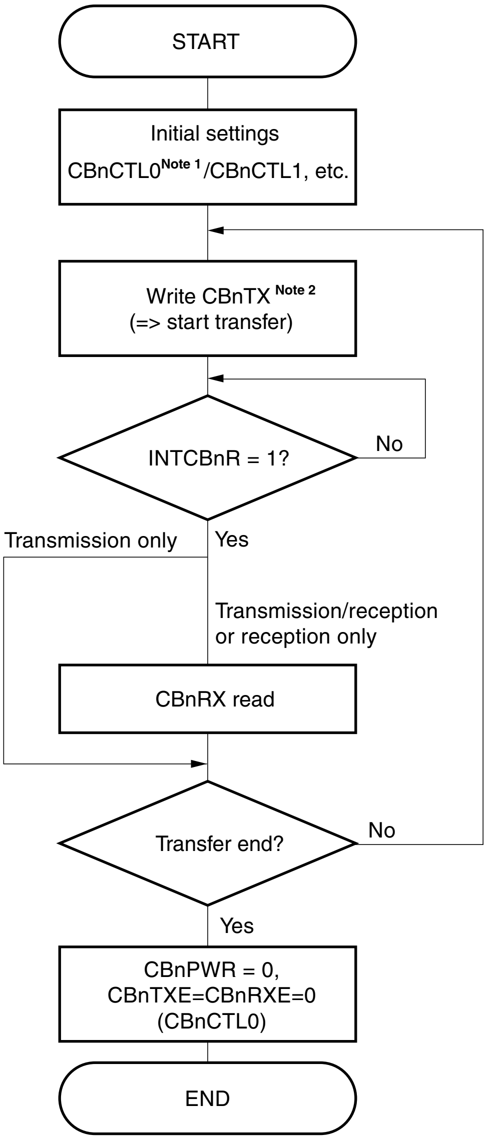
Figure 13-18: Single Reception Flow (Master)



Note: Set the CBnSCE bit to 1 at the initial setting.

(3) Single Transmission/reception

Figure 13-19: Single Transmission/Reception Flow (Master)



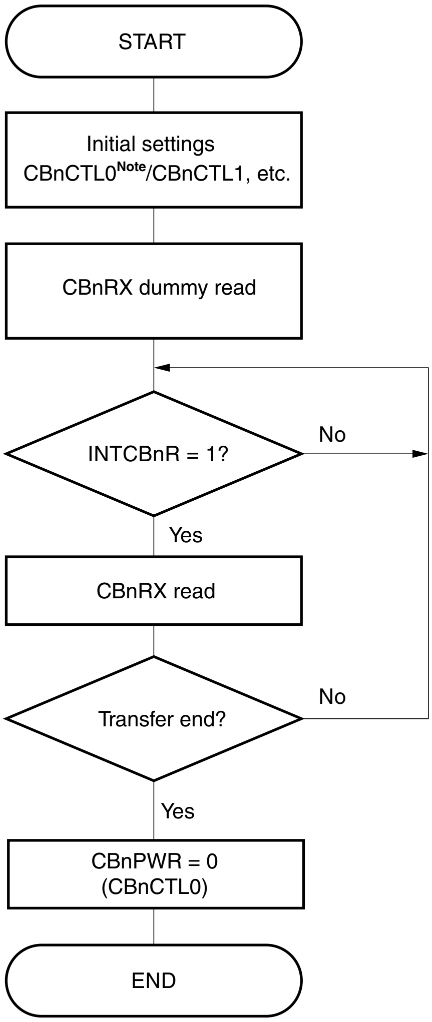
Notes: 1. Set the CBnSCE bit to 1 at the initial setting

2. If the transfer is reception only, write dummy data to the CBnTX register.

Caution: Even in single mode, the CBnOVE flag (CBnSTR register) is set to 1. If only transmission is used in the transmission/reception mode, there is no need to check this flag.

(4) Single reception (slave)

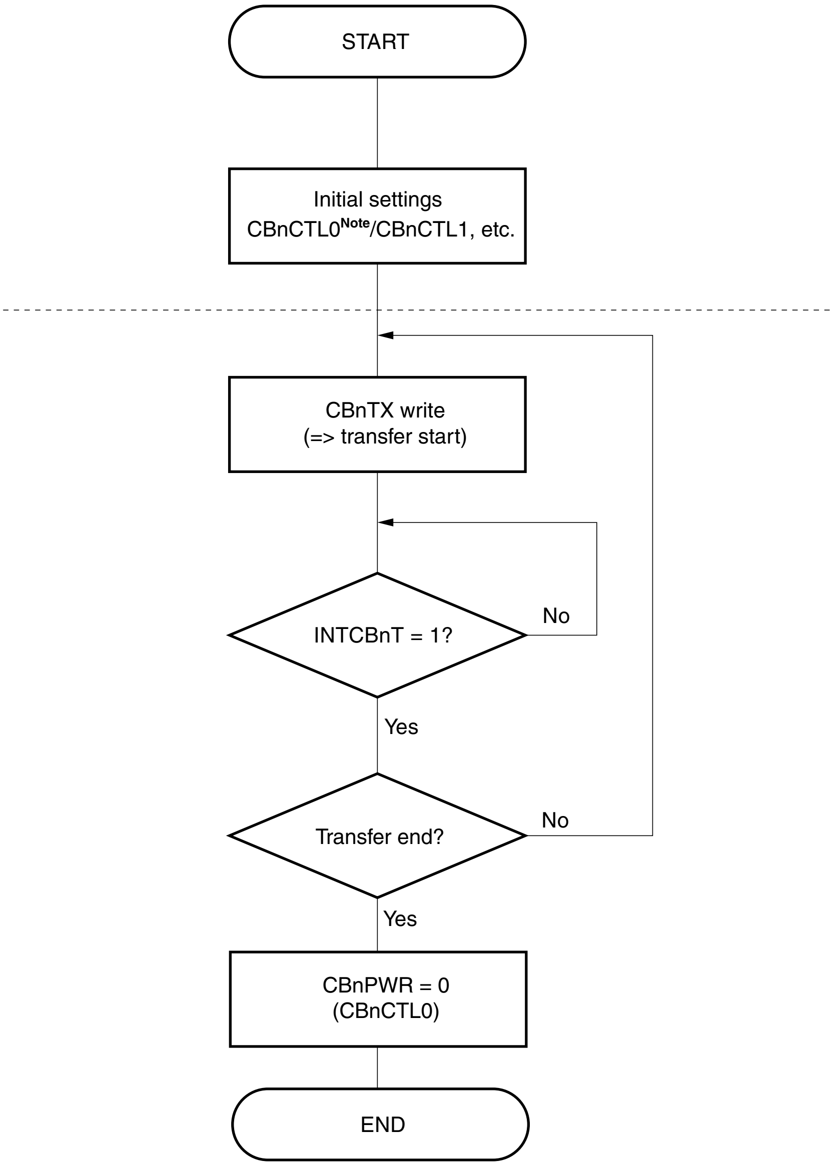
Figure 13-20: Single Reception Flow (Slave)



Note: Set the CBnSCE bit to 1 at the initial setting.

(5) Continuous transmission

Figure 13-21: Continuous Transmission Flow

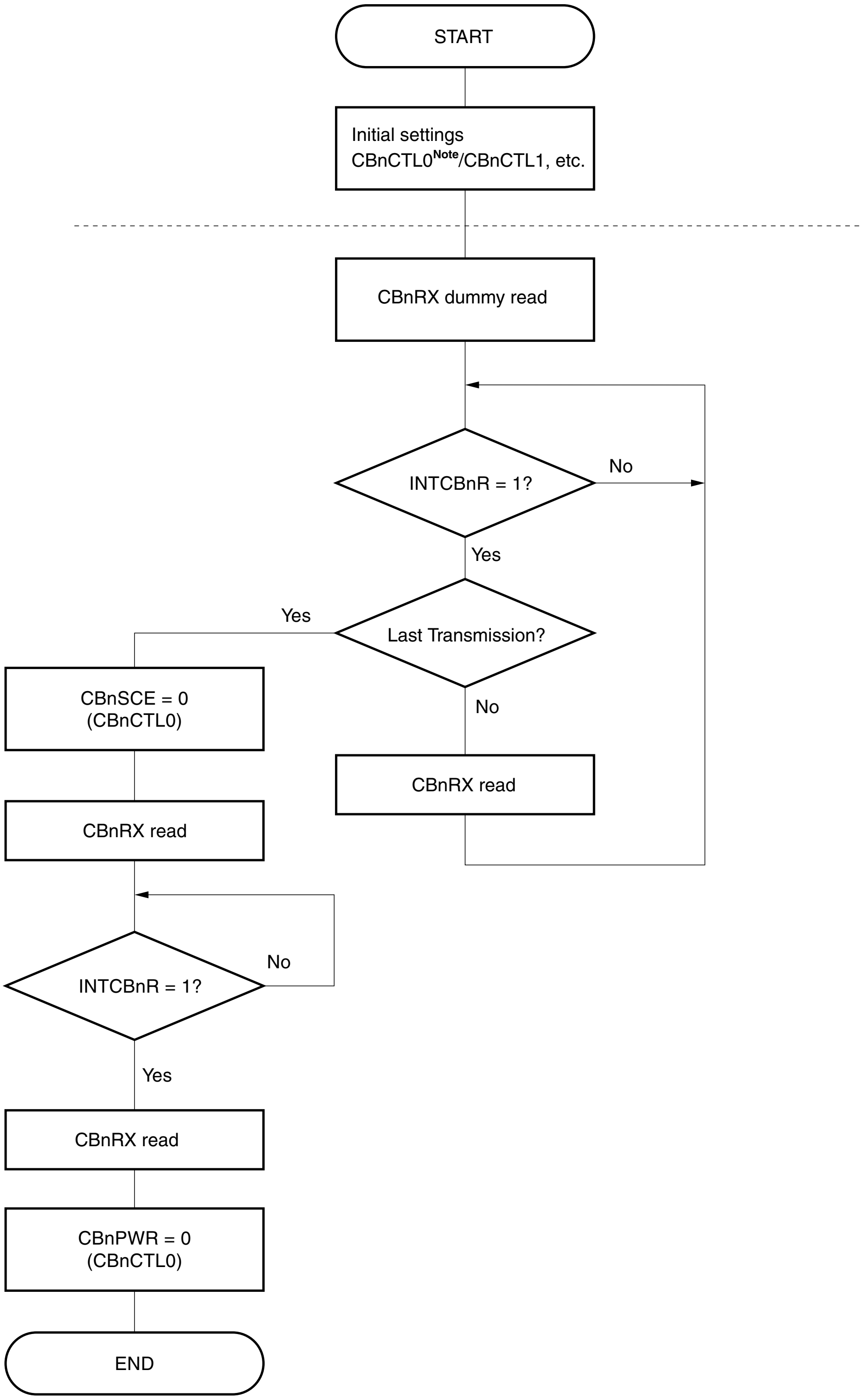


Note: Set the CBnSCE bit to 1 at the initial setting.

Remark: The flow shown below the broken lines is the flow of transmission. Execute this flow to start transmission a second and subsequent time.

(6) Continuous reception (master)

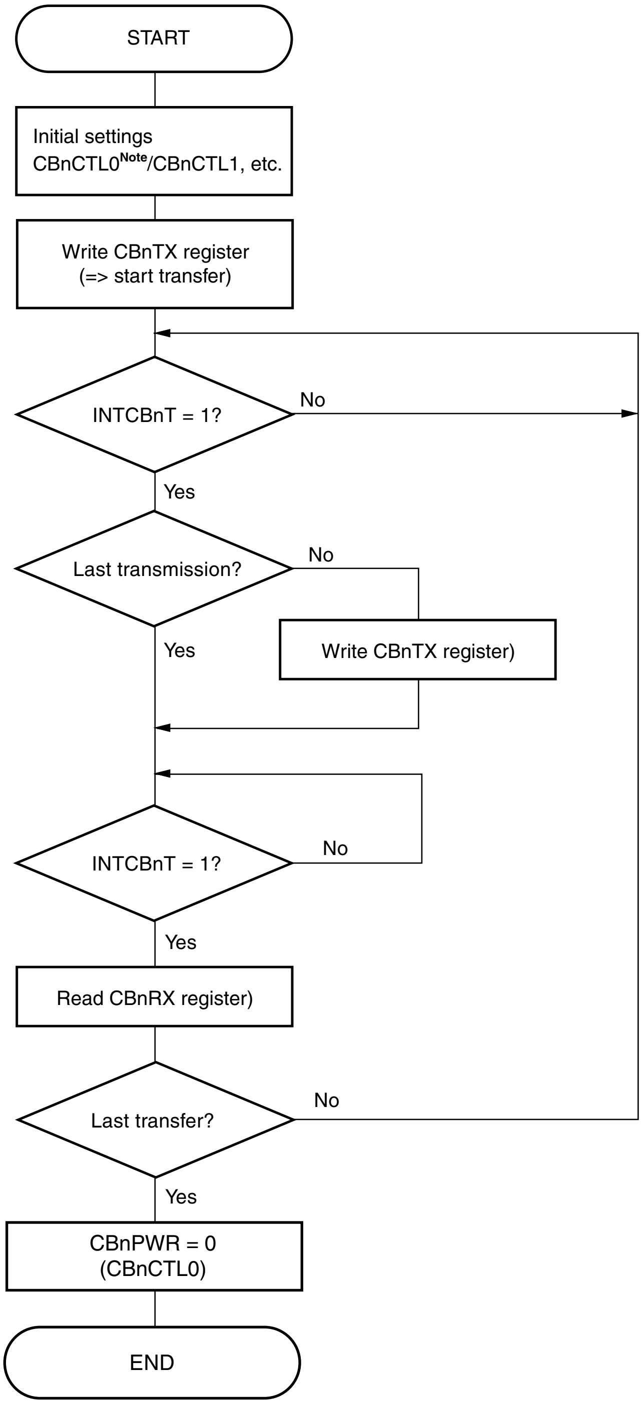
Figure 13-22: Continuous Reception Flow (Master)



Note: Set the CBnSCE bit to 1 at the initial setting.

(7) Continuous transmission/reception (master)

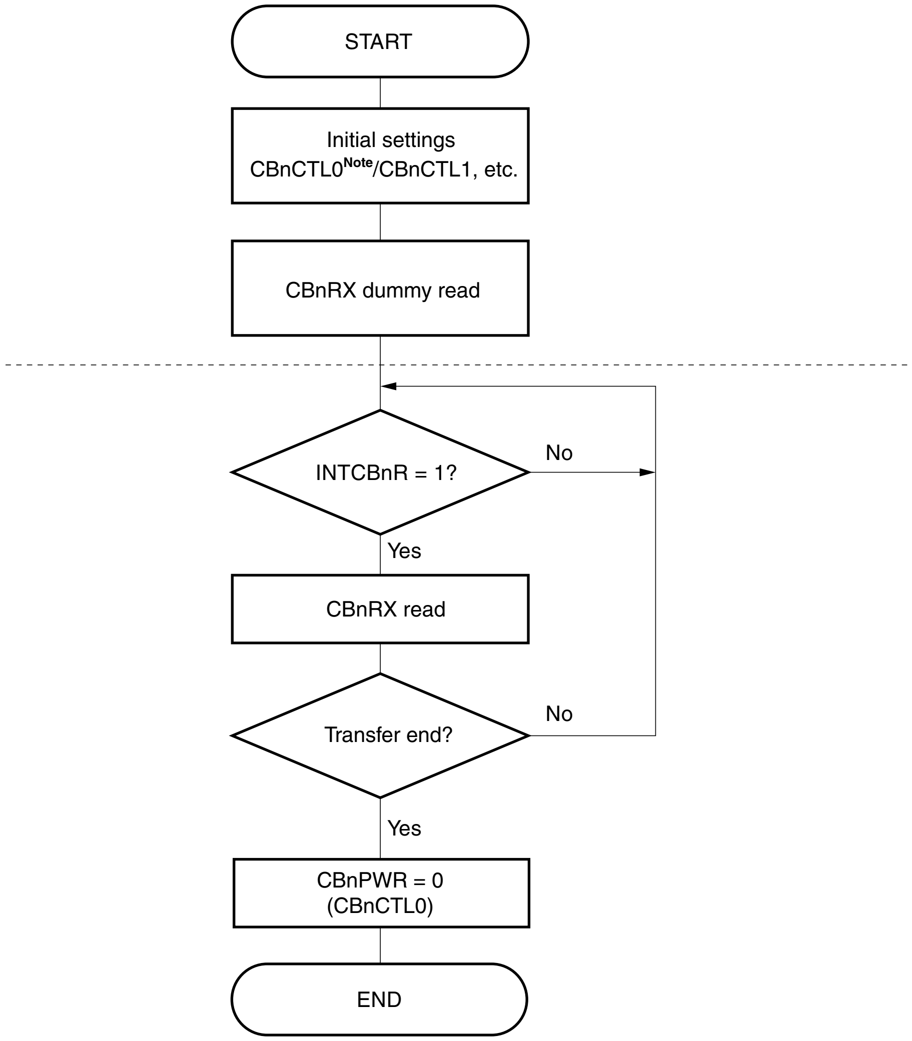
Figure 13-23: Continuous Transmission/Reception Flow (Master)



Note: Set the CBnSCE bit to 1 at the initial setting.

(8) Continuous reception (slave)

Figure 13-24: Continuous Reception Flow (Slave)



Note: Set the CBnSCE bit to 1 at the initial setting.

Remark: The flow shown below the broken lines is the flow of transmission. Execute this flow to start transmission a second and subsequent time.

13.9 Prescaler 3

Prescaler 3 has the following function.

- Generation of count clock for watch timer and CSIB0 (source clock: main oscillation clock)

13.9.1 Control registers of prescaler 3

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register is used to control generation of the count clock for the watch timer and CSIB0.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Cautions: 1. Do not change the values of the BGCS01 and BGCS00 bits while the watch timer is operating.

2. Set the PRSM0 register before setting the BGCE0 bit to 1.

Figure 13-25: Prescaler Mode Register 0 (PRSM0) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00	FFFFF8B0H	R/W	00H

BGCE0	Prescaler output
0	Disabled
1	Enabled

BGCS01	BGCS00	Selection of count clock (f_{BRG}) = 8 MHz	
			8 MHz
0	0	f_X	125 ns
0	1	$f_X/2$	250 ns
1	0	$f_X/4$	0.5 µs
1	1	$f_X/8$	1 µs

(2) Prescaler compare register 0 (PRSCM0)

This is an 8-bit compare register.
It can be read or written in 8-bit units.
Reset input clears this register to 00H.

Cautions: 1. Do not rewrite the PRSCM0 register while the watch timer is operating.

2. Set the PRSCM0 register before setting the BGCE0 bit of the PRSM0 register to 1.

Figure 13-26: Prescaler Compare Register 0 (PRSCM0) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After Reset
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00	FFFFF8B1H	R/W	00H

13.9.2 Generation of count clock

The clock input to the watch timer or CSIB0 (f_{BRG}) can be corrected to 32.768 kHz.
The relationship between the main clock (f_X), set value of count clock selection bits BGCS00 and BGCS01 (m), set value of the PRSCM0 register (N), and output clock (f_{BGR}) is as follows.

$$f_{BRG} = \frac{f_X}{2^m \times N \times 2}$$

Example:

Where f_X = 8.00 MHz, m = 0 (BGCS01 bit = BGCS00 bit = 0), and N = 7AH
f_{BGR} = 32.787 kHz

Remark: f_{BRG}: Count clock
N: Set value of PRSCM0 register (01H to FFH)
N = 256 if the set value of the PRSCM0 register is 00H.
m: Set value of BGCS01 and BGSC00 bits (00B to 11B)

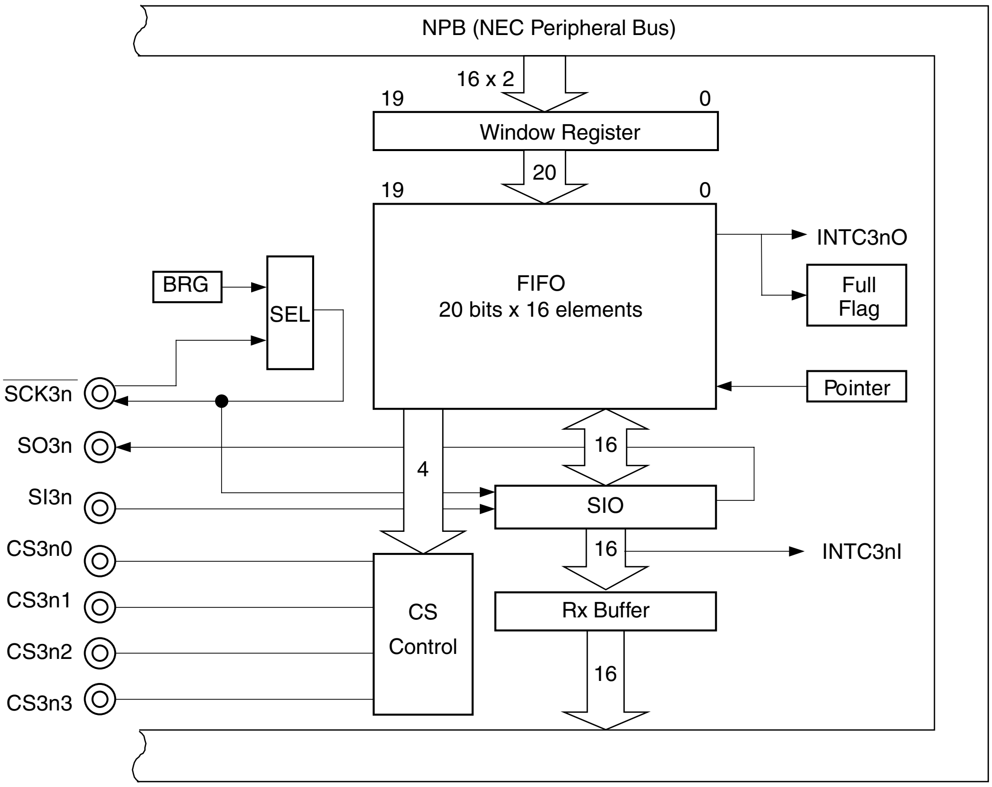
Chapter 14 Queued CSI (CSI30, CSI31)

14.1 Features

- 3-wire serial synchronous transfers
- The following 7 pins are provided to enable a 3-wire serial interface:
  (a) SO3 (serial data output)
  (b) SI3 (serial data input)
  (c) $\overline{\text{SCK3}}$ (serial clock I/O)
  (d) CS30 - CS33 (Chip Select)
- Master mode and slave mode selectable
- Serial clock and data phase selectable
- Transfer date length selectable from 8 to 16 bits in 1-bit units
- Data transfer with MSB- for LSB-first selectable
- Three selectable transfer modes:
  (a) transmit only mode
  (b) receive only mode
  (c) transmit/receive mode
- Transmit and receive FIFO (16 elements)
- Selection between single buffer transfer mode and FIFO buffer transfer mode
- Internal baud rate generator
- Programmable baudrate through BRG output (master) or slave clock
- DMA transfer of received data to memory available
- Maximum $\overline{\text{SCK3}}$ frequency: 10 MHz

14.1.1 Queued CSI Block Diagram

Figure 14-1: Queued CSI Block Diagram



14.1.2 Input/Output Pins

The table below shows the input/output pins of the CSI3.

Table 14-1: Input/Output Pins of the CSI3

Signal name	I/O	Active level	Disabled level	Function
$\overline{\text{SCK3n}}$	I/O	–	H	Serial clock signal
SI3n	I	–	–	Input serial data signal
SO3n	O	–	–	Output serial data signal
CS3n0	O	L**Note 2**	H**Note 2**	Serial peripheral chip select signal
CS3n1	O	L**Note 2**	H**Note 2**	Serial peripheral chip select signal
CS3n2	O	L**Note 2**	H**Note 2**	Serial peripheral chip select signal
CS3n2	O	L**Note 2**	H**Note 2**	Serial peripheral chip select signal

Notes: 1. n = 0, 1

2. The active level is programmable for each chip select.

14.2 Queued CSI Control Registers

(1) Register Map

The tables below show the Special Function Registers for the Queued CSI modules CSI30 and CSI31.

Table 14-2: CSI30

Register name	Function	Address	Reset Value	Access			R/W
				1-bit	8-bit	16-bit	
CSIM0	Queued CSI operation mode register	FFFFFD40H	00H	O	O	-	R/W
CSIC0	Queued CSI clock selection register	FFFFFD41H	07H	O	O	-	R/W
SIRB0	Receive data buffer	FFFFFD42H	0000H	-	-	O	R
SIRB0L	Receive data buffer L	FFFFFD42H	00H	-	O	-	R
SFCS0	Chip Select FIFO buffer	FFFFFD44H	FFFFH	-	-	O	R/W
SFCS0L	Chip Select FIFO buffer L	FFFFFD44H	FFH	-	O	-	R/W
SFDB0	Transmit data FIFO buffer	FFFFFD46H	0000H	-	-	O	R/W
SFDB0L	Transmit data FIFO buffer L	FFFFFD46H	00H	-	O	-	R/W
SFA0	FIFO status register	FFFFFD48H	20H	O	O	-	R/W
CSIL0	Data length select register	FFFFFD49H	00H	O	O	-	R/W
SFN0	Transfer number select register	FFFFFD4CH	00H	O	O	-	R/W

Table 14-3: CSI31

Register name	Function	Address	Reset Value	Access			R/W
				1-bit	8-bit	16-bit	
CSIM1	Queued CSI operation mode register	FFFFFD60H	00H	O	O	-	R/W
CSIC1	Queued CSI clock selection register	FFFFFD61H	07H	O	O	-	R/W
SIRB1	Receive data buffer	FFFFFD62H	0000H	-	-	O	R
SIRB1L	Receive data buffer L	FFFFFD62H	00H	-	O	-	R
SFCS1	Chip Select FIFO buffer	FFFFFD64H	FFFFH	-	-	O	R/W
SFCS1L	Chip Select FIFO buffer L	FFFFFD64H	FFH	-	O	-	R/W
SFDB1	Transmit data FIFO buffer	FFFFFD66H	0000H	-	-	O	R/W
SFDB1L	Transmit data FIFO buffer L	FFFFFD66H	00H	-	O	-	R/W
SFA1	Transmit FIFO status register	FFFFFD68H	20H	O	O	-	R/W
CSIL1	Data length select register	FFFFFD69H	00H	O	O	-	R/W
SFN1	Transmit data number select register	FFFFFD6CH	00H	O	O	-	R/W

(2) Queued CSI Operation Mode Registers (CSIM0, CSIM1)

The CSIM registers control the Queued CSI macro's operations.
These registers can be read or written in 1-bit and 8-bit units.
TRMD, DIR, CSIT, CSWE, CSMD bits can only be written when CTXE = 0 and CRXE = 0.
The registers are initialized to 00H at reset.

Figure 14-2: Queued CSI Operation Mode Registers (CSIM0, CSIM1) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
CSIMn (n = 0, 1)	POWER	CTXE	CRXE	TRMD	DIR	CSIT	CSWE	CSMD	FFFFFD40H, FFFFFD60H	R/W	00H

POWER	Queued CSI operation clock control
0	Stop macro operation clock (Reset internal control circuits)
1	Provide macro operation clock
Clearing POWER = "0" resets the internal circuits asynchronously, stops operation and sets the Queued CSI to standby state. Input clock is not provided to internal circuits. Set POWER = "1" to activate the Queued CSI.	

**Caution: When changing the POWER bit, do not change any other bit at the same time.
While POWER=”0”, the only registers that can be accessed are CSIM, SFDB, SFDBL, and SFA.
Set the POWER bit before writing any of the other bits of CSIMn.**

CTXE	Transmission enable/disable
0	Transmission disabled
1	Transmission enabled

CRXE	Receive enable/disable
0	Receive disabled
1	Receive enabled

TRMD	Transfer mode select
0	Single buffer transfer mode
1	FIFO buffer transfer mode

Caution: Write is permitted only when CTXE = 0 and CRXE = 0.

Figure 14-2: Queued CSI Operation Mode Registers (CSIM0, CSIM1) Format (2/2)

DIR	Serial data direction selection
0	Data is sent/received with MSB first
1	Data is sent/received with LSB first

Caution: Write is permitted only when CTXE = 0 and CRXE = 0.

See section **14.3.2 ”Serial Data Direction Select Function” on page 477** for further details on the DIR bit setting.

CSIT	Interrupt delay mode select (INTC3nI signal)
0	No delay
1	Half clock delay

Caution: Write is permitted only when CTXE = 0 and CRXE = 0.
This bit is only valid in master mode. In slave mode, no delay is generated.

CSWE	Transmission wait enable/disable select
0	Transmission wait disable. Not insert 1 clock ($\overline{\text{SCK3}}$) wait at transmission start.
1	Transmission wait enable. Insert 1 clock ($\overline{\text{SCK3}}$) wait at transmission start.

Caution: Write is permitted only when CTXE = 0 and CRXE = 0.
This bit is only valid in master mode. In slave mode, no wait is generated.

CSMD	Chip Select mode select
0	Chip select inactive level output disable. Do not force chip select inactive state after each transfer of a data element.
1	Chip select inactive level output enable. Hold all chip selects inactive for halt-length $\overline{\text{SCK3}}$ after each transfer of a data element.

Caution: Write is permitted only when CTXE = 0 and CRXE= 0.
This bit is only valid for CSWE=1.
This bit is only valid in master mode. In slave mode, CS signals are always held at inactive level.

In combination:

CSWE	CSMD	Transmission wait	Chip Select inactive level
0	0	None	Not output
0	1	None	Not output
1	0	One $\overline{\text{SCK3}}$ length clock wait	Not output
1	1	One $\overline{\text{SCK3}}$ length clock wait	Output inactive level of half-length $\overline{\text{SCK3}}$ during first half of transmission wait

See section **14.3.10 ”Additional Timing and Delay Selections” on page 486** for further details on the timing selections by CSIT, CSWE, CSMD bits.

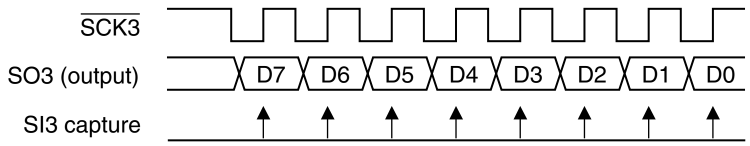
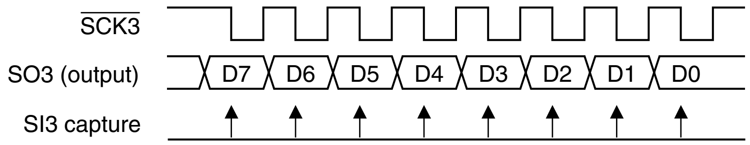
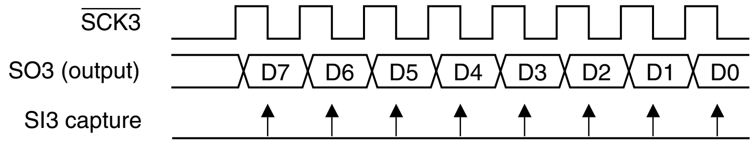
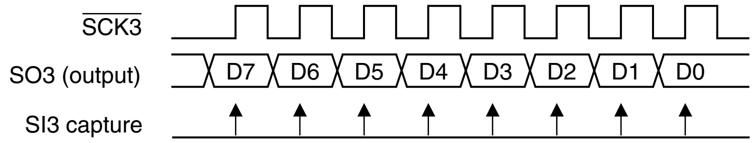
(3) Queued CSI Clock Selection Registers (CSIC0, CSIC1)

The CSIC register is an 8-bit register that is used to control the serial transfer operations. This register can be read or written in 1-bit and in 8-bit units.

Caution: This register can be written only while CSIM register's CTXE = 0 and CRXE = 0.

Figure 14-3: Queued CSI Clock Selection Registers (CSIC0, CSIC1) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
CSICn (n = 0, 1)	MDL2	MDL1	MDL0	CKP	DAP	CKS2	CKS1	CKS0	FFFFFD41H, FFFFFD61H	R/W	07H

CKP	DAP	Operation mode
0	0	 $\overline{\text{SCK3}}$ SO3 (output) SI3 capture D7 D6 D5 D4 D3 D2 D1 D0
0	1	 $\overline{\text{SCK3}}$ SO3 (output) SI3 capture D7 D6 D5 D4 D3 D2 D1 D0
1	0	 $\overline{\text{SCK3}}$ SO3 (output) SI3 capture D7 D6 D5 D4 D3 D2 D1 D0
1	1	 $\overline{\text{SCK3}}$ SO3 (output) SI3 capture D7 D6 D5 D4 D3 D2 D1 D0

Caution: Modification of these bits is permitted only when CTXE = 0 and CRXE = 0.

Remark: CKP: Clock phase selection bit
DAP: Data phase selection bit

Figure 14-3: Queued CSI Clock Selection Registers (CSIC0, CSIC1) Format (2/2)

CKS2	CKS1	CKS0	Prescaler output (PRSOUT)	Mode	K
0	0	0	f_{QCSI}	Master Mode	0
0	0	1	$f_{QCSI}/2$	Master Mode	1
0	1	0	$f_{QCSI}/4$	Master Mode	2
0	1	1	$f_{QCSI}/8$	Master Mode	3
1	0	0	$f_{QCSI}/16$	Master Mode	4
1	0	1	$f_{QCSI}/32$	Master Mode	5
1	1	0	$f_{QCSI}/64$	Master Mode	6
1	1	1	$\overline{\text{SCK3}}$ (input) Note	Slave Mode	–
These bits are used to select the input clock					

Note: $\overline{\text{SCK3}}$ pin is configured as input mode for serial transfer clock.
f_{QCSI} is the clock supply of the Queued-CSI macro.

Caution: Rewriting these bits is only permitted when CSIE = 0.

See sections **14.3.4 "Slave Mode" on page 479** and **14.3.5 "Master Mode" on page 479** for further explanation on slave mode and master mode.

MDL2	MDL1	MDL0	Operation clock	N	
0	0	0	BRG disable	–	BRG stop for power saving
0	0	1	PRSOUT/2	1	
0	1	0	PRSOUT/4	2	
0	1	1	PRSOUT/6	3	
1	0	0	PRSOUT/8	4	
1	0	1	PRSOUT/10	5	
1	1	0	PRSOUT/12	6	
1	1	1	PRSOUT/14	7	
These bits are used to modulate the selected clock					

Caution: Rewriting these bits is only permitted when CTXE = 0 and CRXE = 0.
MDL [2:0] = [0,0,1] is prohibited when CKS [2:0] = [0,0,0].

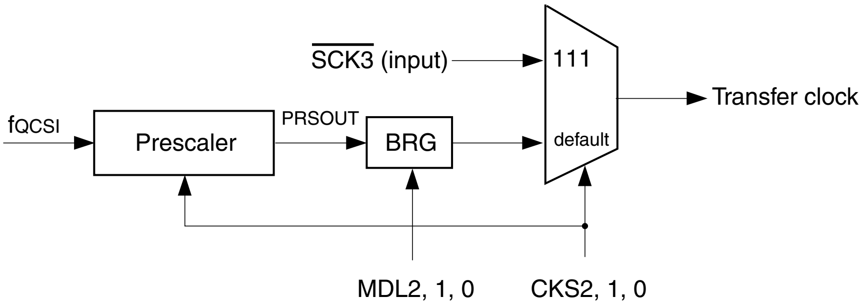
See section **14.3.6 "Transmission Clock Select Function" on page 480** for further explanation on the transfer clock selection.

The baudrate for the transmission is calculated with the following formula:

Transmission Baud Rate = f/(N*2(K+1))

where: f: f_{QCSI} frequency,
N = 1 - 7,
K = 0 - 6.

Figure 14-4: Queued CSI Baud Rate Block Diagram



(4) Receive Data Buffer Registers (SIRB0, SIRB1)

The SIRB register is a 16-bit register or separated as upper 8 bits (SIRBH) and lower 8 bits (SIRBL), that is used to store receive data. This register can be read in 8-bit or 16-bit units and is initialized to 0000H by reset.

Figure 14-5: Receive Data Buffer Registers (SIRB0, SIRB1) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
SIRBn (n=0, 1)																	FFFF FD42H, FFFF FD62H	R/W	0000H

SIRB15 - SIRB0	Data received from the SIO serial shift register.

Caution: The receive data buffer register is considered as emptied by the application software whenever the lower 8 bits of the register are read. It is therefore necessary to read SIRBH before SIRBL when 8-bit access is used.

(5) Chip Select Data Buffer Registers (SFCS0, SFCS1)

The SFCS register is a 16-bit register, or separated as upper 8 bits (SFCSH) and lower 8 bits (SFCSL), that stores Chip Select data. This register is read/write-enabled and is accessible in 8-bit or 16-bit units. Initial value is FFFFH by reset.

Following the FIFO write pointer, the value written to SFCS is stored in the FIFO data buffer as Chip Select bits. The value is stored to these bits when the transmit data is written to its register (SFDB or SFDBL).

SFCS write is prohibited when POWER = 1 and f_{QCSI} is stopped.

Figure 14-6: Chip Select Data Buffer Registers (SFCS0, SFCS1) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
SFCSn (n=0, 1)	-	-	-	-	-	-	-	-	-	-	-	-	SFCS 3	SFCS 2	SFCS 1	SFCS 0	FFFF FD44H, FFFF FD64H	R/W	FFFFH

SFCSm	Chip Select Output Selection (m=0 to 3)
0	Output an active level at the CS3nm pin during the data transfer
1	Output an inactive level at the CS3nm pin during the data transfer

Caution: The Chip Select register is stored in the FIFO buffer when the transmit data is written to its register. It is therefore necessary to write the SFCS register before the SFDB (SFDBL) register.

Remark: The active level for each chip select is defined in the CSILn register (CSA[3:0] bits)

(6) Transmission Data Buffer registers (SFDB0, SFDB1)

The SFDB register is a 16-bit buffer register, or separated as upper 8 bits (SFDBH) and lower 8 bits (SFDBL), that stores transmission data. The SFDB register is read/write-enabled and is accessible in 8-bit or 16-bit units. Initial value is 0000H by reset.

Incrementing and following the FIFO buffer write pointer, the value written to SFDB is stored to the FIFO buffer as transmission data.

SFDB write is prohibited when POWER = 1 and f_{QCSI} is stopped.

Figure 14-7: Transmission Data Buffer Registers (SFDB0, SFDB1) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	R/W	Initial value
SFDBn (n=0, 1)																	FFFF FD46H, FFFF FD66H	R/W	0000H

SFDB15 - SFDB0	Write data is stored to the FIFO data buffer as transmission data. A read operation reads out the last stored transmission data.

Caution: The transmit data buffer register is considered as written whenever the lower 8 bits of the register are written. It is therefore necessary to write SFDBH before SFDBL when 8-bit access is used.

(7) FIFO Buffer Status Registers (SFA0, SFA1)

The SFA register is an 8-bit register that shows the FIFO buffer status. The SFA register is read/write-enabled, accessible in 1-bit or 8-bit units. However, the bits SFFUL, SFEMP and CSOT are read only. Initial value is 20H by reset.

SFA read is prohibited when POWER = 1 and f_{QCSI} is stopped.

Figure 14-8: FIFO Buffer Status Registers (SFA0, SFA1) Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
SFAn (n = 0, 1)	FPCLR	SFFUL	SFEMP	CSOT	SFP3	SFP2	SFP1	SFP0	FFFFFD48H, FFFFFD68H	R/W	20H

FPCLR	FIFO buffer pointer clear command
0	No operation
1	Clear all FIFO pointers to "0"

Remark: Read value is always “0”.

Figure 14-8: FIFO Buffer Status Registers (SFA0, SFA1) Format (2/2)

SFFUL	FIFO buffer full status flag
0	FIFO buffer is not full
1	FIFO buffer is full

Remark: Read only

SFEMP	FIFO buffer empty status flag
0	FIFO buffer is not empty
1	FIFO buffer is empty

Remark: Read only

CSOT	Transmission status flag
0	Idle state
1	Transmission in on going or preparing

Remarks:
1. Read only bit.
2. This bit is cleared to “0” by POWER = 0 or (CTXE = 0 and CRXE = 0).
3. In Single Buffer Transfer Mode, this bit holds “1” from transmission start to FIFO empty.
4. In FIFO Buffer Transfer Mode, this bit holds “1” from transmission start until finish transferring all data to be sent.

SFP3 - SFP0	Transmission data count
nnnnH	In Single transfer mode, SFP3 - SFP0 indicates the number of remaining transfers in the FIFO. This value can be understood as: (Write FIFO pointer) - (SIO load pointer) SFP3 - SFP0 is read only in Single transfer mode.
	In FIFO transfer mode,SPF3 - SFP0 indicates the number of data transfers completed. In case of SFP3 - SFP0 = 0H: - SFEMP = 0, SFP3 - SFP0 = 0H: Number of receptions completed = 0 - SFEMP = 1, SFP3 - SFP0 = 0H: Number of receptions completed = 16

Remarks:
1. Read only bits.
2. SFP3 - SFP0 holds its value until RESET or FPCLR = 1.

Caution: **SFFUL, SFEMP, CSOT and SFP3 - SFP0 are continuously updated with the current status of the Queued CSI. This means that a value read might be outdated shortly after the read was executed.**
When writing accidentally a 17th data element in FIFO, an overflow interrupt (INTC3nO) will occur to indicate the error.

(8) Queued CSI Data Length Selection Registers (CSIL0, CSIL1)

The CSIL register is an 8-bit register that specifies the active voltage level of the Chip Select pins and the Queued CSI data length.
This register can be read or written in 1-bit and 8-bit units.
This register can be overwritten only while CTXE = 0 and CRXE = 0 (CSIM register). Write operation during CTXE = 1 or CRXE = 1 is prohibited.
Initial value is 00H by reset.

Figure 14-9: Queued CSI Data Length Selection Registers (CSIL0, CSIL1) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
CSILn (n = 0, 1)	CSLV3	CSLV2	CSLV1	CSLV0	CCL3	CCL2	CCL1	CCL0	FFFFFD49H, FFFFFD69H	R/W	00H

CSLV3 - CSLV0	Chip Select active level selection
0	Chip Select signal is active low
1	Chip Select signal is active high

CCL3	CCL2	CCL1	CCL0	Transfer data length
0	0	0	0	Data length is 16 bits
0	0	0	1	Setting prohibited
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	Data length is 8 bits
1	0	0	1	Data length is 9 bits
1	0	1	0	Data length is 10 bits
1	0	1	1	Data length is 11 bits
1	1	0	0	Data length is 12 bits
1	1	0	1	Data length is 13 bits
1	1	1	0	Data length is 14 bits
1	1	1	1	Data length is 15 bits

Caution: Rewriting this register is only permitted when CTXE = 0 and CRXE = 0.

See section **14.3.3 ”Data Length Select Function” on page 478** for further explanation on the data length selection.

(9) Queued CSI Transfer Number Selection Registers (SFN0, SFN1)

The SFN register is an 8-bit register that specifies the number of data elements to be transferred in FIFO buffer transfer mode. It can be read or written in 1-bit and 8-bit units.
Initial value is 00H by reset.

Figure 14-10: Queued CSI Transfer Number Selection Registers (SFN0, SFN1) Format

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
SFNn (n = 0, 1)	0[Note]	0[Note]	0[Note]	0[Note]	SFN3	SFN2	SFN1	SFN0	FFFFFD4CH, FFFFFD6CH	R/W	00H

Note: Unused bits must be written as 0.

SFN3	SFN2	SFN1	SFN0	Transfer data number
0	0	0	0	Transfer 16 data elements
0	0	0	1	Transfer 1 data element
0	0	1	0	Transfer 2 data elements
0	0	1	1	Transfer 3 data elements
0	1	0	0	Transfer 4 data elements
0	1	0	1	Transfer 5 data elements
0	1	1	0	Transfer 6 data elements
0	1	1	1	Transfer 7 data elements
1	0	0	0	Transfer 8 data elements
1	0	0	1	Transfer 9 data elements
1	0	1	0	Transfer 10 data elements
1	0	1	1	Transfer 11 data elements
1	1	0	0	Transfer 12 data elements
1	1	0	1	Transfer 13 data elements
1	1	1	0	Transfer 14 data elements
1	1	1	1	Transfer 15 data elements

14.3 Explanation of Queued CSI Functions

14.3.1 Transmit Buffer

Chip select data and transmission data can be stored to the transmit FIFO buffer continuously by writing to the SFCS register and SFDB register. The Writing FIFO pointer is automatically incremented when data is written to SFDB. The size of the transmit FIFO buffer is 20 bits × 16 entries.

In slave mode, chip select data does not need to be set.

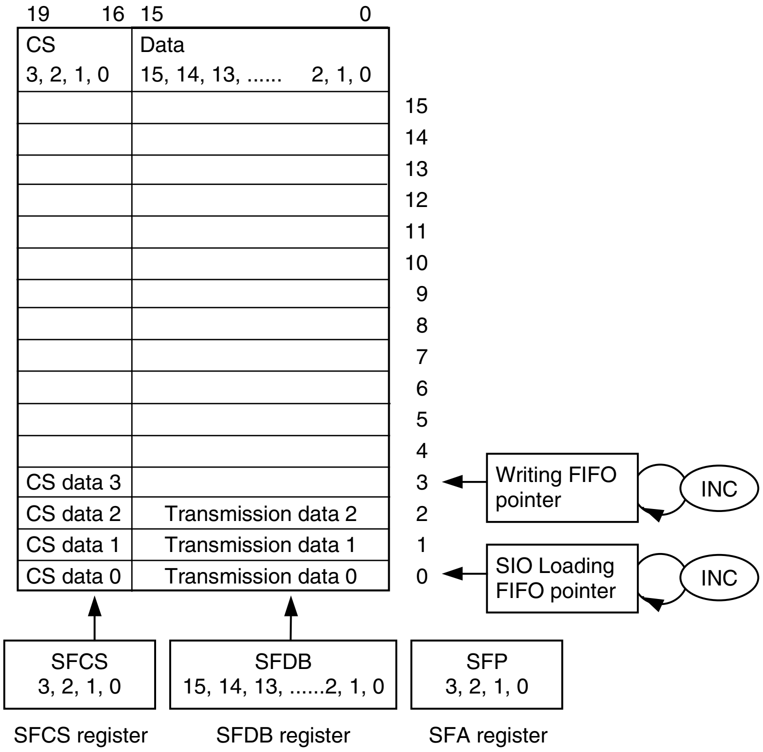
The transfer start condition (SFEMP = 0) is to write to the lower bits of SFDB register. If the transmission data length is 9 bits or more, data should be written by a 16-bit write to SFDB or by two 8-bit writes to first SFDBH, then SFDBL (in that order). When transmission data length is 8 bits, data should be set by one 8-bit write to SFDBL or by one 16-bit write to SFDB. For the 16-bit write the upper 8 bits are ignored in the 8-bit transmission.

The SFFUL bit in the SFA status register is set "1" after 16 writes have been made to the transmit buffer, assuming the Writing FIFO pointer was reset previously.

When a transmission write is attempted while the FIFO is full (SFFUL=1), the interrupt INTC3nO is generated to indicate an overflow. In that case, the transmission and chip select data are discarded and not stored.

When a transfer cycle is finished and the SIO Loading FIFO pointer is incremented, the FIFO buffer of the previous location is considered empty in case of single buffer transfer mode. Refer to section 14.3.8 on page 483 for more information on FIFO buffer transfer mode.

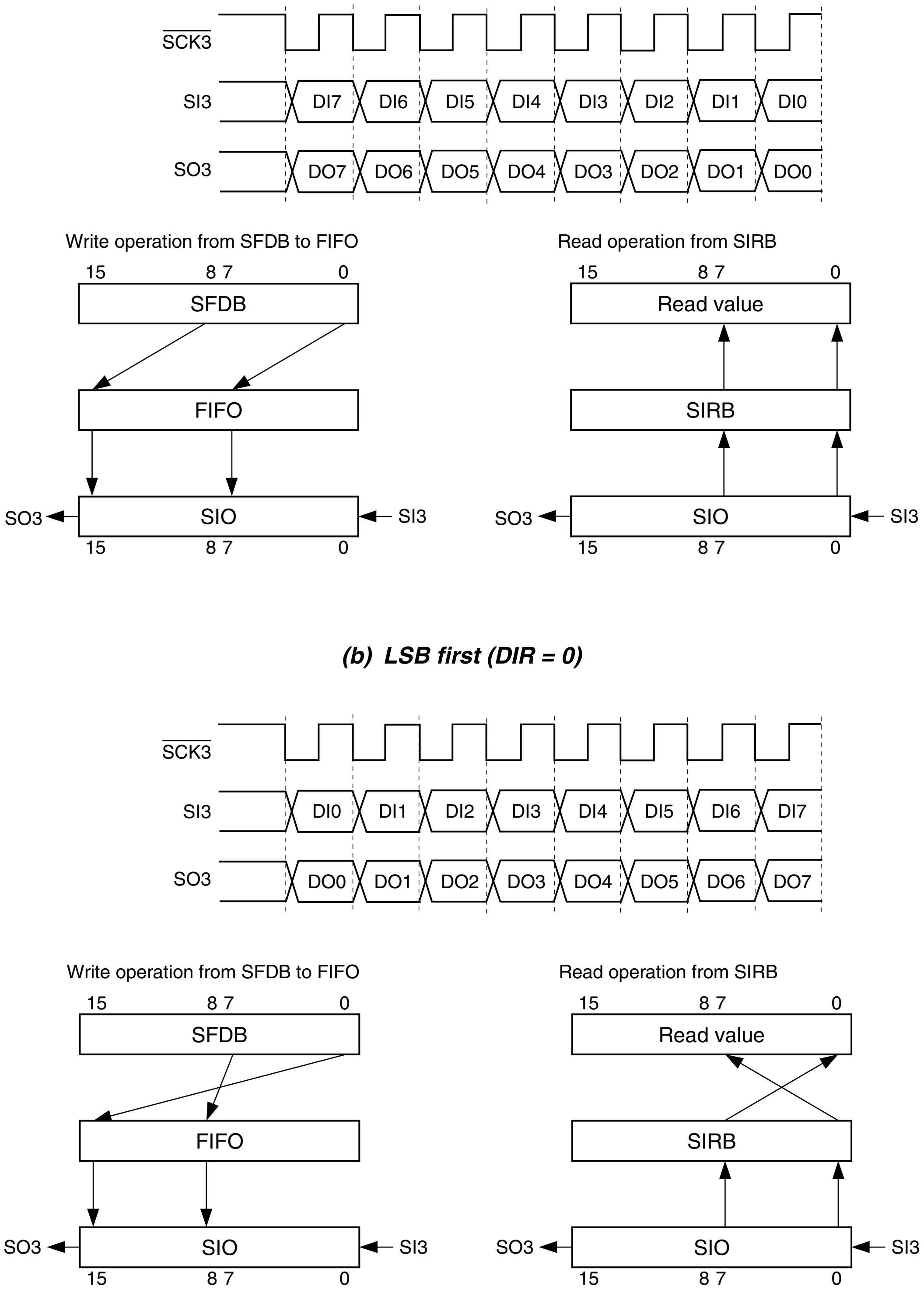
Figure 14-11: Transmit Buffer



14.3.2 Serial Data Direction Select Function

The serial data direction is selectable using the DIR bit in the CSIM register. The examples below show the communication for data length of 8 bit (CCL[3:0] = [1,0,0,0]):

Figure 14-12: Serial Data Direction Select Function



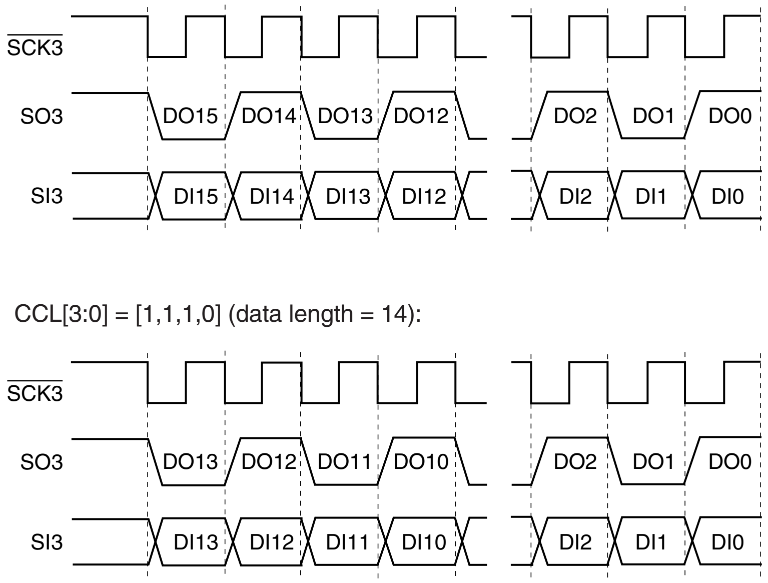
(a) MSB first (DIR = 0)

(b) LSB first (DIR = 0)

14.3.3 Data Length Select Function

Transmission data length is selectable from 8 bits to 16 bits using the CCL[3:0] bits in CSIL register. The examples below show the communication with MSB first (DIR = 1):

Figure 14-13: Data Length Select Function



CCL[3:0] = [0,0,0,0] (data length = 16):

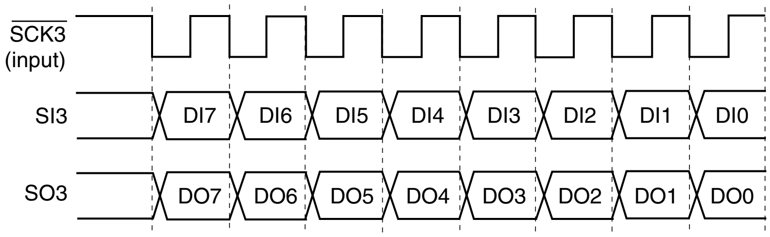
CCL[3:0] = [1,1,1,0] (data length = 14):

14.3.4 Slave Mode

When the CKS[2:0] bits in CSIC are set to [1,1,1], the Queued CSI operates in slave mode. In slave mode, the $\overline{\text{SCK3}}$ serial clock pin becomes input and another device is the CSI communication master. The baud rate generator “BRG” is recommended to be disabled by setting bits MDL[2:0] to [0,0,0] when using slave mode. Also, the chip select pin CS3n[3:0] outputs are not available in slave mode, as they are only available in master mode.

The example below shows the communication in slave mode for 8 data bits, CKP=0, DAP=0 and MSB first:

Figure 14-14: Slave Mode

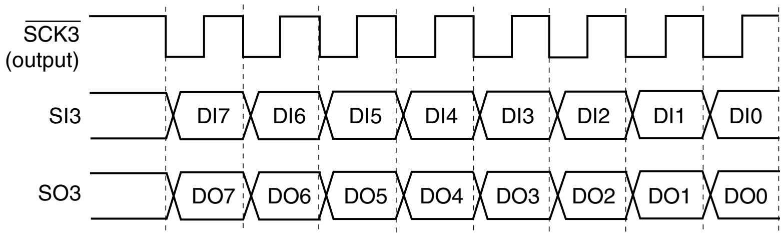


14.3.5 Master Mode

When the CKS[2:0] bits in CSIC are not set to [1,1,1], the Queued CSI operates in master mode. In master mode, the $\overline{\text{SCK3}}$ pin is configured as output and the serial communication clock is generated by the Queued CSI module. The $\overline{\text{SCK3}}$ pin’s default value is “1” when CKP = 1, and is default “0” when CKP = 0. The CS3n[3:0] pin outputs are available in master mode.

The example below shows the communication in master mode for 8 data bits, CKP=0, DAP=0 and MSB first:

Figure 14-15: Master Mode

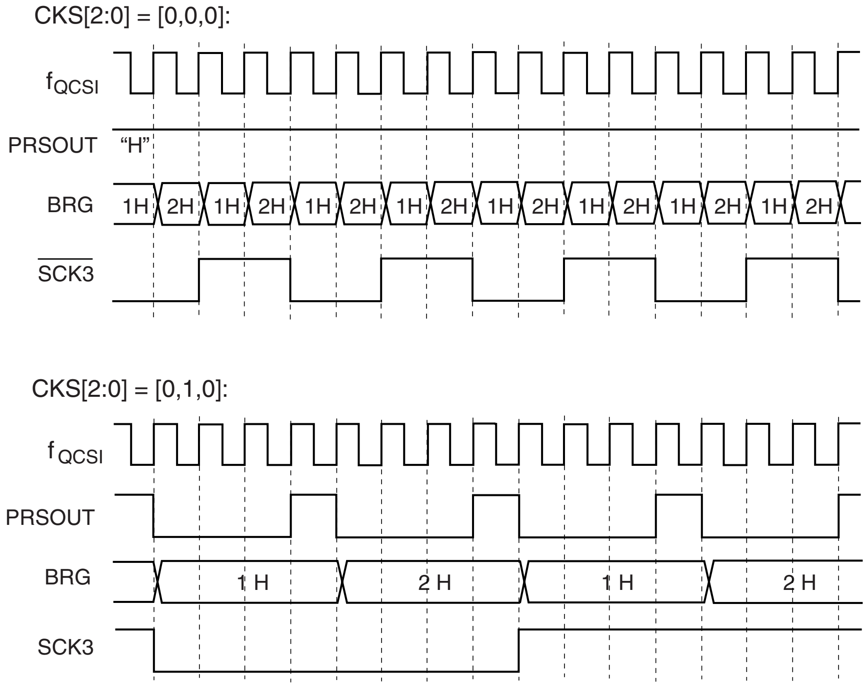


14.3.6 Transmission Clock Select Function

In Master Mode, the transfer baud rate is selectable using CKS[2:0] bits and MDL[2:0] bits in CSIC register. The baud rate generator “BRG” counts up at each rising edge of f_{QCSI}.

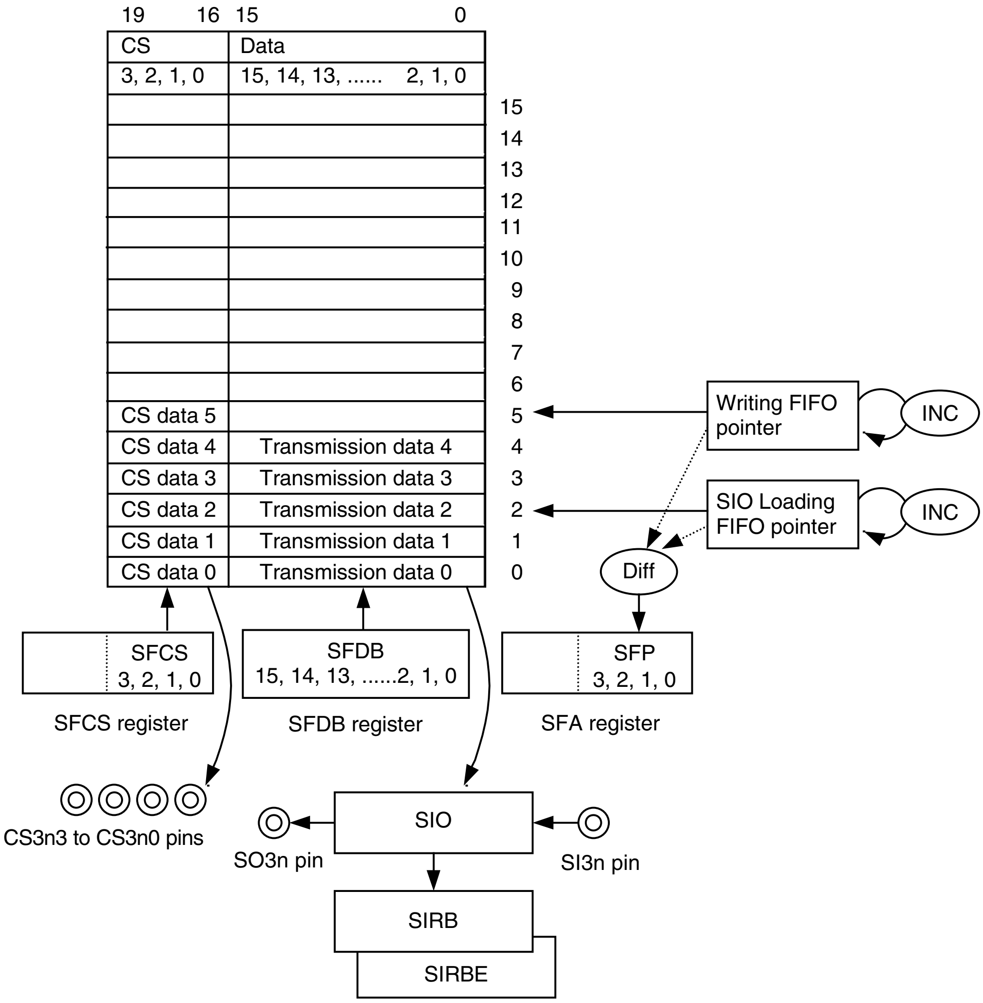
The example below illustrates the baud rate generation for MDL[2:0] = [0,1,0].

Figure 14-16: Transfer Clock Select Function



14.3.7 Description of the Single Buffer Transfer Mode

Figure 14-17: Single Buffer Transfer Mode Data Handling



Transfer start condition in single buffer transfer mode:
[CTXE = 1 or CRXE = 1] and
[Data exists in FIFO (SFEMP = 0)]

A transfer starts once the transmission data (pointed to by the SIO Loading FIFO pointer) is transferred from the FIFO buffer to the serial shift register SIO. At that time, the transfer status flag CSOT turns to “1”. The CS3n[3:0] pins output the chip select data from the FIFO buffer.

At the end of the transfer:

(A) If SIRB is empty, the received data is stored from SIO to SIRB, and transfer end interrupt signal INTC3nI is generated (in transmit only mode, INTC3nI will be generated only when the FIFO buffer becomes empty). Finally, the SIO L7oading FIFO pointer is incremented.

(B) If SIRB is not empty, the storing of receive data, INTC3nI generation and SIO Loading FIFO pointer incrementing wait for the SIRB to be emptied by a software read operation.

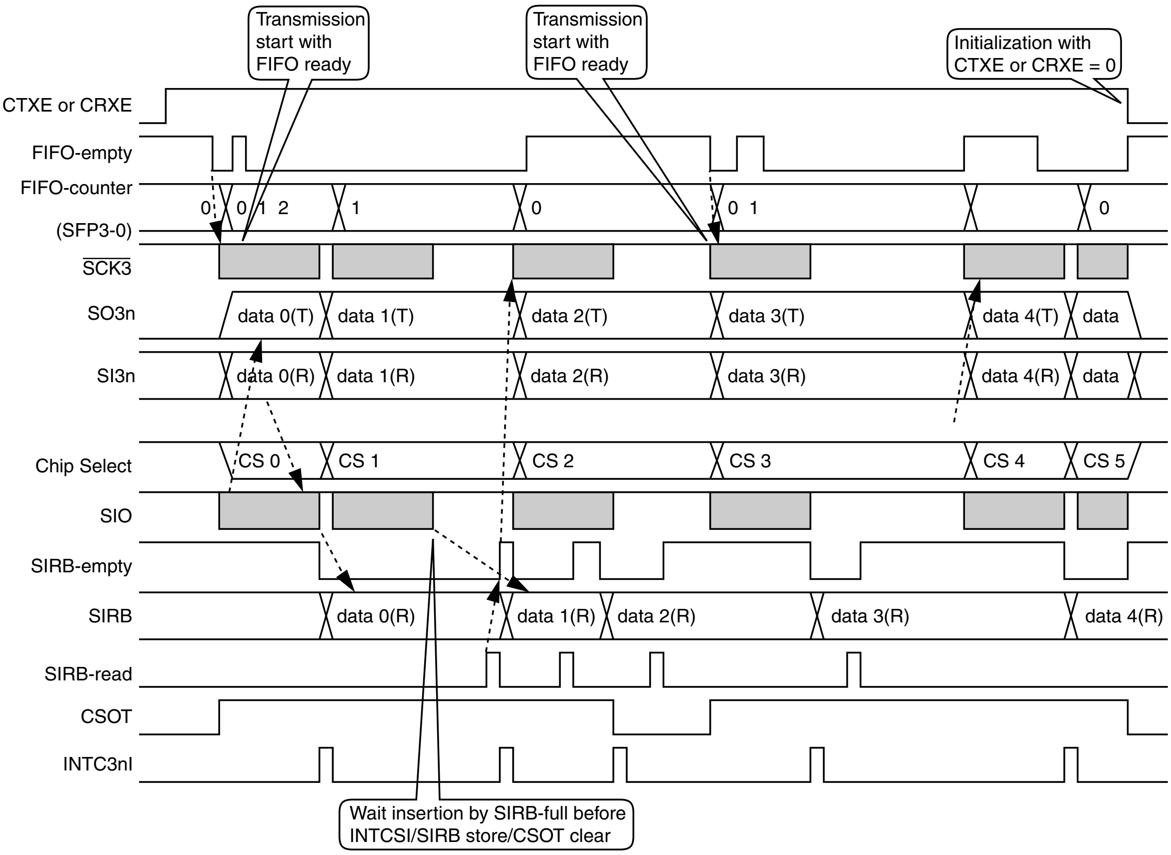
(C) In transmit only mode, if transmission data is available in the FIFO buffer, the next transfer will start immediately, regardless of the SIRB buffer condition.

When a transfer finishes and the FIFO buffer is empty (Writing FIFO pointer = SIO Loading FIFO pointer), CSOT is cleared “0”.

SFP[3:0] always show the current value of: (Writing FIFO pointer) - (SIO Loading FIFO pointer).

It is recommended to check that SFFUL = 0 just before data is written to the SFDB register. If SFFUL = 1 when a write to SFDB is attempted, the overflow interrupt INT3nO is generated and the written data is ignored.

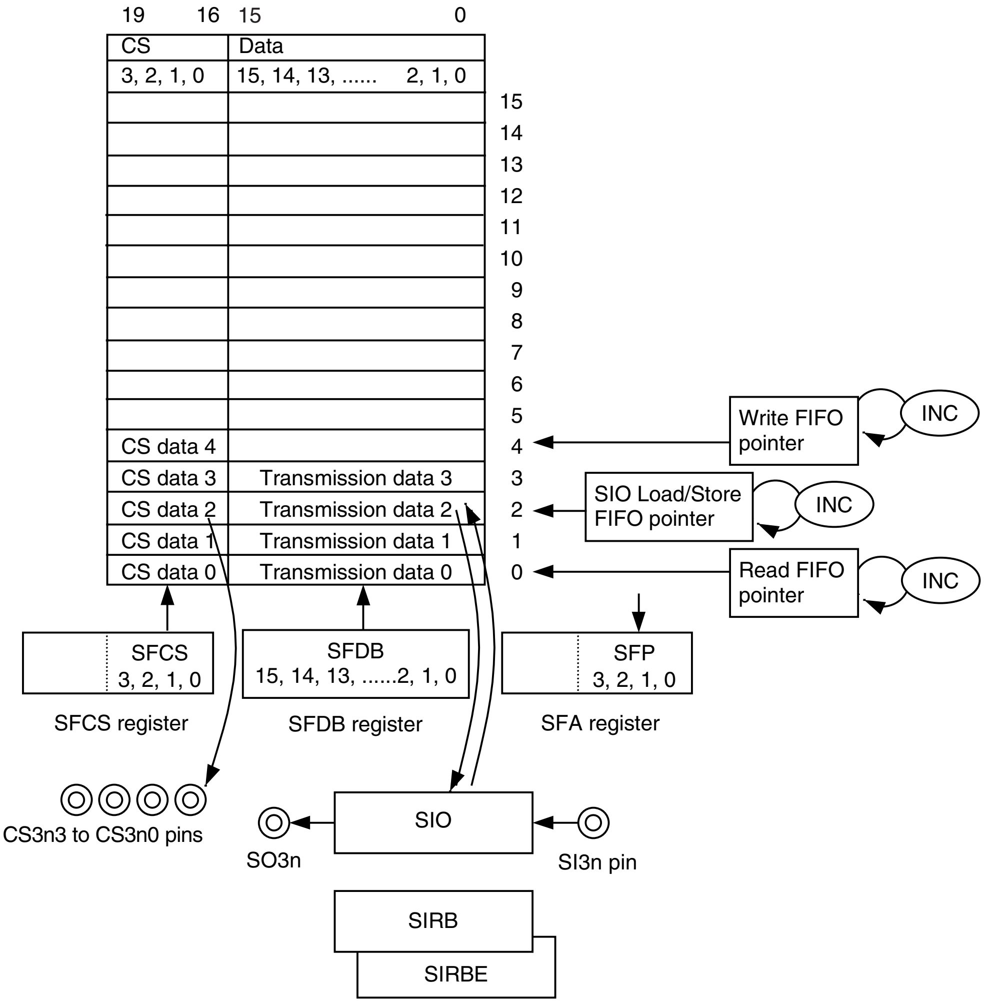
Figure 14-18: Single Buffer Transfer Mode (Master, Transmit/Receive) Timing



14.3.8 Description of the FIFO Buffer Transfer Mode

When the TRMD bit in the CSIM register is set “1”, the Queued CSI operates in FIFO buffer transfer mode.

Figure 14-19: FIFO Buffer Transfer Mode Data Handling



Transfer start condition in FIFO buffer transfer mode:
[CTXE = 1 or CRXE = 1] and
[Data exists in FIFO (SFEMP = 0)]

The transmission data number must be set in SFN[3:0]. Note that writing a value greater than 16 to the SFN register is prohibited, as the FIFO buffer design can hold up to 16 elements only.

The transfer starts by copying the first data element - pointed to by SIO Load/Store FIFO pointer - to the SIO shift register. At that time the transmission status flag CSOT is set to “1”, and the CS3n[3:0] pins output the CS value from the FIFO.

When the transfer of the data element is finished, the received data overwrites the location in the FIFO using the SIO Load/Store FIFO pointer, and the SIO Load/Store FIFO pointer is then incremented.

When the transmission/reception counter reaches the value set by SFN[3:0], then CSOT is cleared “0” and the transmission/reception end interrupt signal INTC3nI is generated.

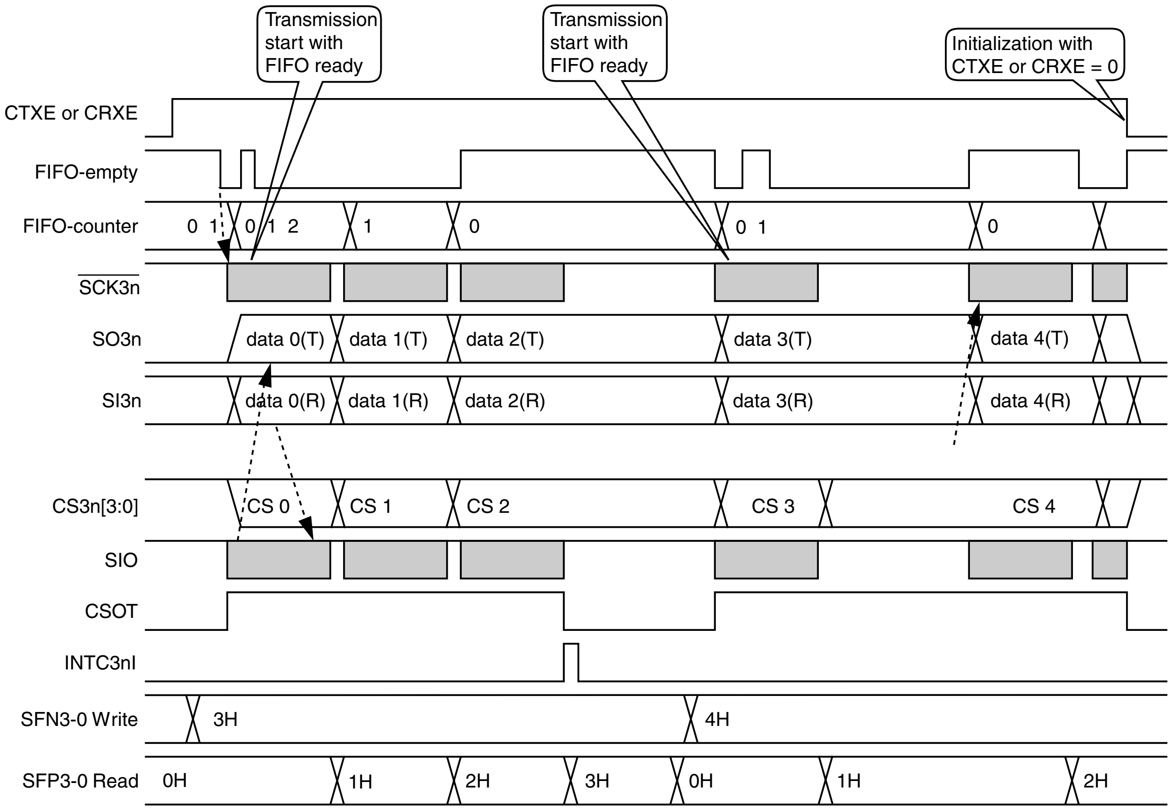
After the interrupt occurred, the received data can be read from SIRB. The Read FIFO pointer is automatically incremented by the SIRB read operation.

All FIFO pointers must be cleared by setting FPCLR = 1 before the next transmit/receive cycle can start.

SFP[3:0] represents the [SIO Load/Store FIFO pointer] and shows the number of transmission/receptions completed. In case of SFP[3:0]=0H, the numbers of transmissions/receptions depends on the setting of the SFEMP bit:

SFEMP=0: 0 transmissions/receptions completed
SFEMP=1: 16 transmissions/receptions completed

Figure 14-20: FIFO Buffer Transfer Mode (Master, Transmit/Receive) Timing



14.3.9 Description of the Operation Modes

(1) Transmit Only Mode

Setting the CSIM register's CTXE =1 and CRXE = 0 places the Queued CSI in transmit only mode. A transmission starts when transmit data is written in the SFDB register. The current condition of the SIRB buffer and SIO register no effect. The data in the SIRB and the SIO buffer is undefined after completion of the transmission.

(2) Receive Only Mode

Setting the CSIM register's CTXE = 0 and CRXE = 1 places the Queued CSI in receive only mode. A reception starts when dummy data is written in the SFDB register. It is mandatory, though, that the SIRB and SIO are empty. If a receive operation is terminated while the previous receive data remains unread in SIRB, the Queued CSI is placed on wait status until the previous data is completely read and SIRB becomes empty.

(3) Transmit/Receive Mode

Setting the CSIM register's CTXE = 1 and CRXE = 1 places the Queued CSI in transmit/receive mode. A transfer (meaning transmission and reception) starts when transmit data is written in the SFDB register. Note that an empty SIRB or SIO is mandatory. If a receive operation is terminated while the previous receive data remains unread in SIRB, the Queued CSI is placed on wait status until the previous data is completely read and SIRB becomes empty.

In FIFO buffer transfer mode with slave mode, only the first dummy data write operation is required. There is no need to write chip-select data, as these bits are ignored.

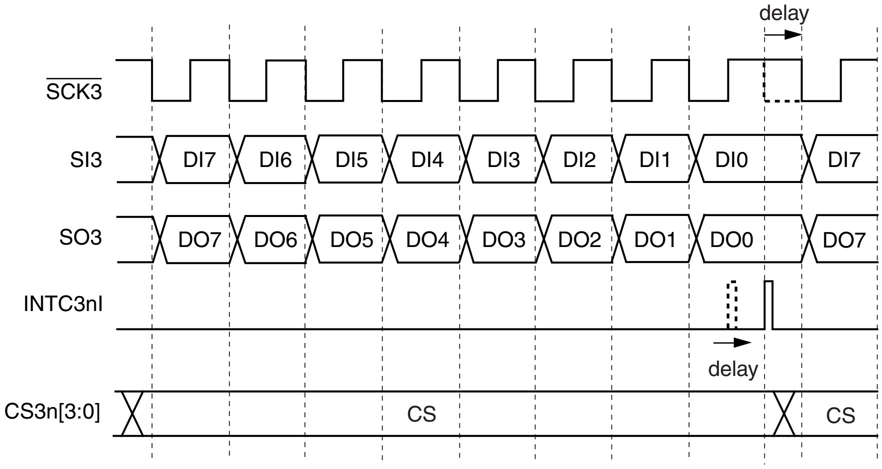
14.3.10 Additional Timing and Delay Selections

(1) Delay Selection of Receive Termination Interrupt Signal (INTC3nI)

In master mode, the CSIT bit of the CSIM register can be used to delay the generation of the receive termination interrupt signal (INTC3nI) by a half serial clock cycle ($\overline{\text{SCK3}}$). The CSIT bit takes effect only in the master mode and is ignored in slave mode.

Figure 14-21 below illustrates the CSIT function, assuming a setting of CSIT=1, CSWE=0, CKP=0, DAP=0 and CCL[3:0] = [1,0,0,0].

Figure 14-21: Delay Selection of Receive Termination Interrupt (INTC3nI)

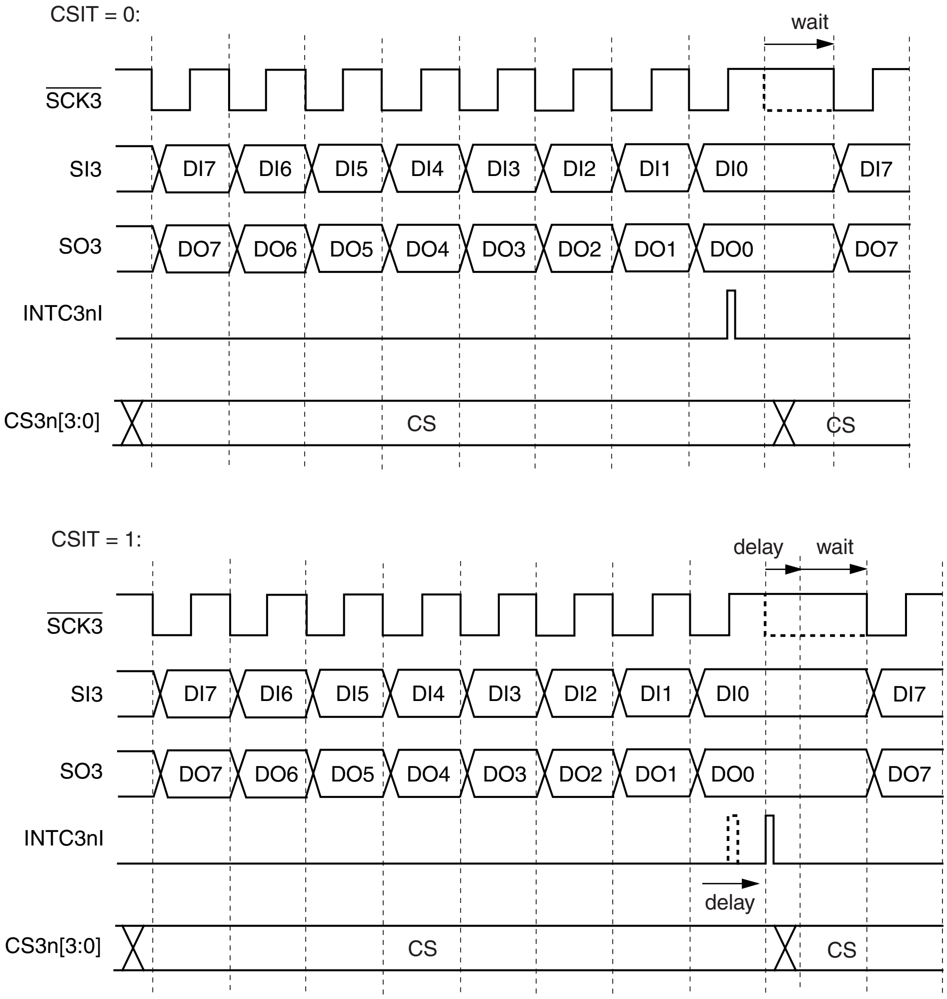


(2) Selection of Transmit Wait Enable/Disable

In master mode, the CSIM register's CSWE bit setting can be used to delay the start of transmission by one $\overline{\text{SCK3}}$ clock cycle. The CSWE bit takes effect only in the master mode and is ignored in slave mode.

Figure 14-22 below illustrates the CSWE function, assuming a setting of CSWE=1, CSMD=0, CKP=0, DAP=0 and CCL[3:0] = [1,0,0,0].

Figure 14-22: Selection of Transmit Wait Enable/Disable



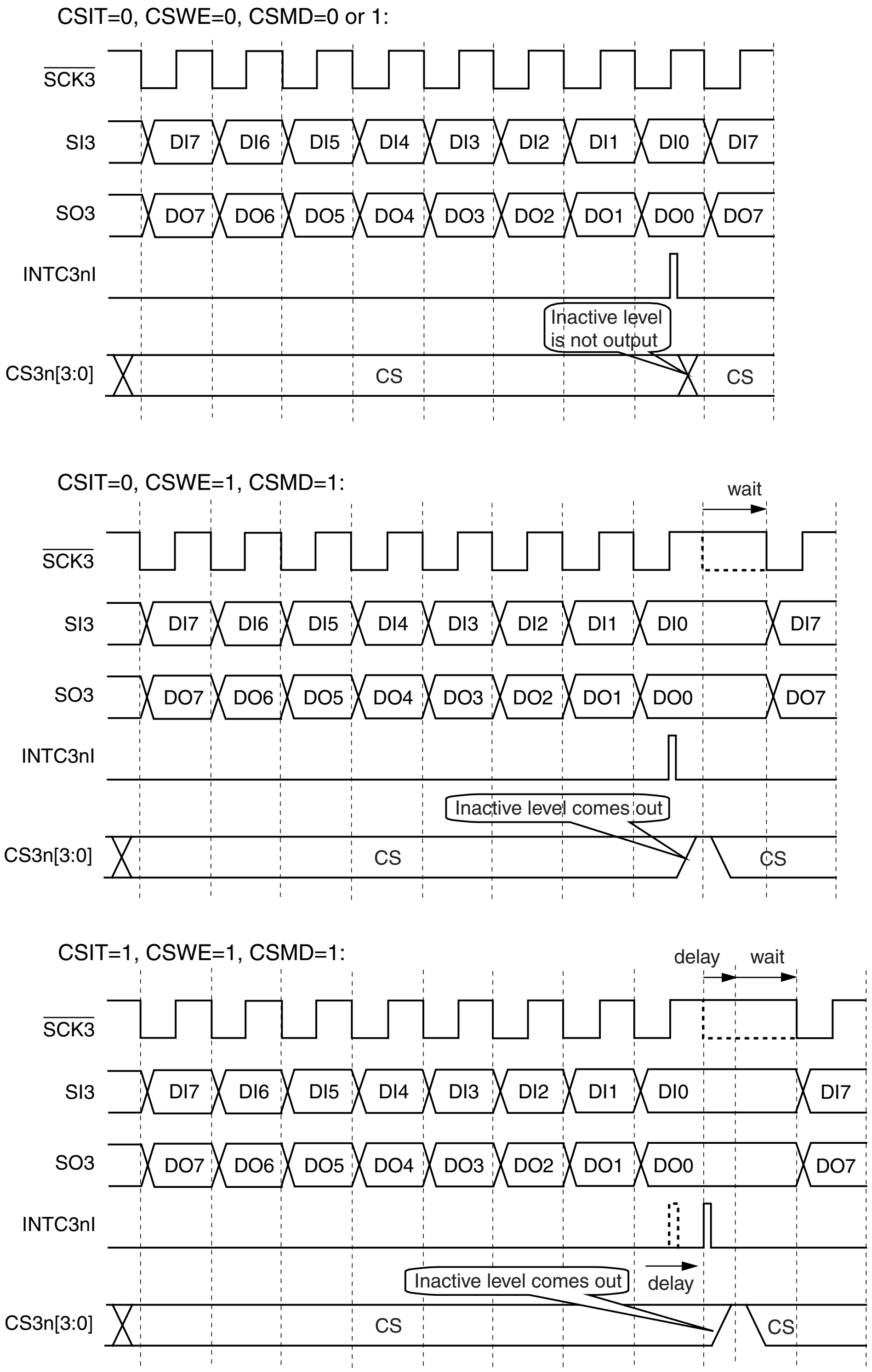
(3) Selection of Chip-Select Mode

In master mode with CSWE = 1, the CSMD bit setting can be used to output an inactive level at the chip select pins CS3n[3:0] during the delay between data transmissions. The CSMD bit takes effect only in the master mode and is ignored in slave mode.

The CSMD bit has no effect while CSWE is set to 0.

Figure 14-23 below illustrates the CSMD function, assuming a setting of CKP=0, DAP=0 and CCL[3:0] = [1,0,0,0] and the active levels of all chip selects set to “active low”.

Figure 14-23: Selection of Chip-Select Mode



CSIT=0, CSWE=0, CSMD=0 or 1:

CSIT=0, CSWE=1, CSMD=1:

CSIT=1, CSWE=1, CSMD=1:

14.3.11 Default Pin Levels

(1) $\overline{\text{SCK3}}$ Pin's Default Level

$\overline{\text{SCK3}}$ pin's default level with the CSIM register settings POWER = 0 or CTXE / CRXE = 0

CKP	CKS2, CKS1, CKS0	$\overline{\text{SCK3}}$ default level	
0	1, 1, 1 (slave mode)	1	← Initialization after reset
	Other than 1, 1, 1 (master mode)	1	
1	1, 1, 1 (slave mode)	1	
	Other than 1, 1, 1 (master mode)	0	

Remark: In slave mode, the $\overline{\text{SCK3}}$ pin is always set to “1.”

(2) SO3 Pin's Default Level

SO3 pin's default level with the CSIM register settings POWER = 0 or CTXE / CRXE = 0

SO3 pin's default level	
0	←Initialization after reset

(3) CS3n0 to CS3n3 Pins' Default Level

CS3n0 to CS3n3 pins' default level with the CSIM register settings POWER = 0 or CTXE / CRXE = 0

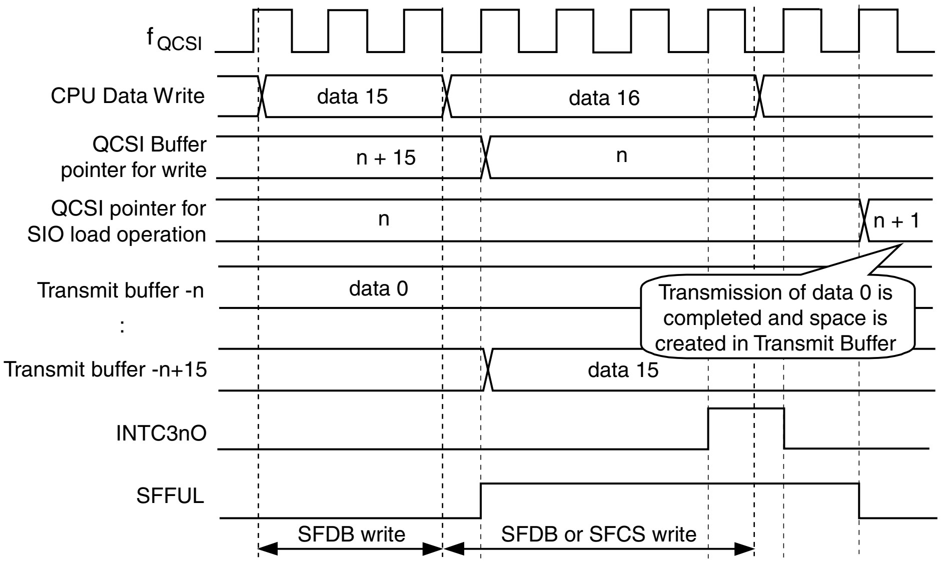
CS3n0 to CS3n3 pins' default level	
inactive	←Initialization after reset

14.3.12 Transmit Buffer Overflow Interrupt Signal (INTC3nO)

When the transmit FIFO buffer contains 16 elements, writing a 17th chip-select data (SFCS write) or transfer data (SFDB write) results in the generation of the overflow interrupt INTC3nO. For the 17th item, both chip-select and transfer data values are discarded.

The transmit FIFO buffer contains 16 elements if the FIFO pointer value for the write operation equals the FIFO pointer value for the SIO load operation plus 15. When the transfer is completed and the FIFO buffer pointer for the SIO load operation is incremented, space for one element is available again in transmit buffer.

Figure 14-24: Transmit Buffer Overflow Interrupt Signal (INTC3nO)

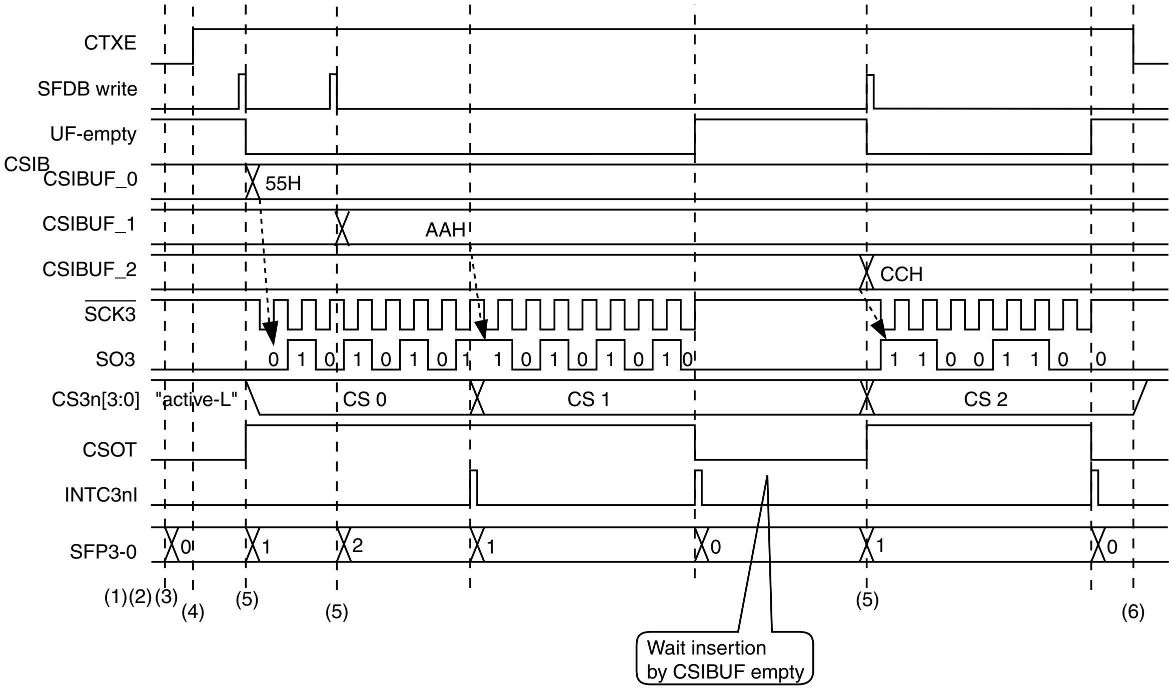


14.4 Operating Procedure

(1) Single Buffer Transfer Mode (Master Mode, Transmit Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT=0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-25: Single Buffer Transfer Mode (Master, Transmit Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE bit to 1 to enable transmission.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and transmission data in the SFCS and SFDB registers in this order.

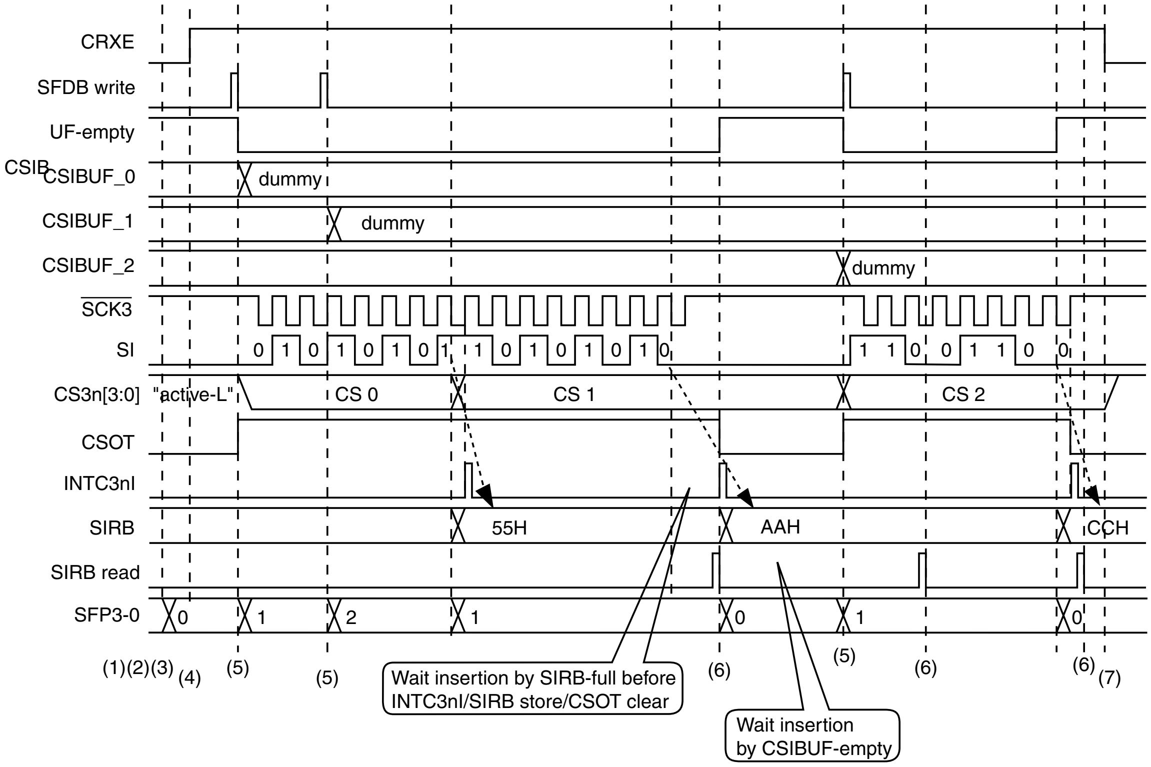
   Repeat step (5) until the last element to be transmitted is written in the SFCS/SFDB registers.

6. Set the CSIM register's CTXE bit to 0 to disable transmission (end of transmission).

(2) Single Buffer Transfer Mode (Master Mode, Receive Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to "active low":

Figure 14-26: Single Buffer Transfer Mode (Master, Receive Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write "1" in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CRXE bit to 1 to enable the receive operation.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and dummy transmission data in the SFCS and SFDB registers in this order (start-of-receive trigger).
6. Check for a reception to be completed (e.g. by monitoring the INTC3nI interrupt). If so, read the SIRB register.

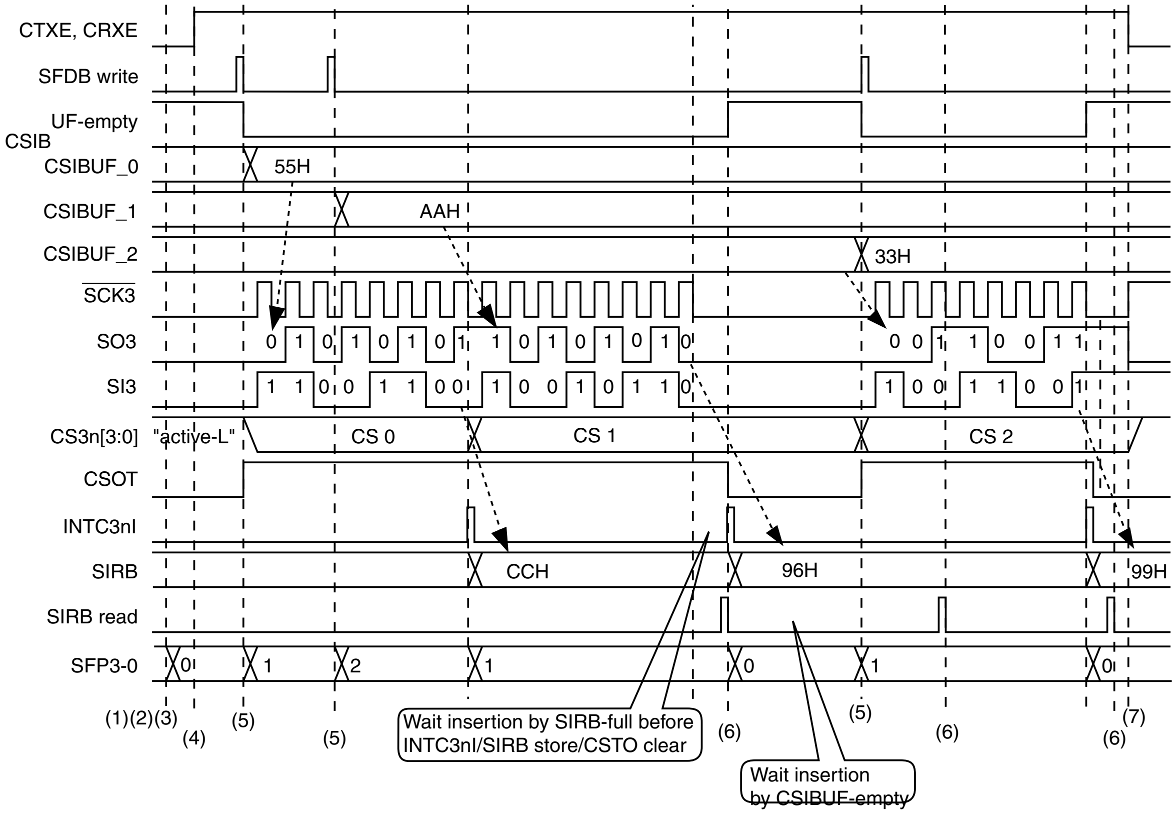
   Repeat steps (5) and (6) until the last element is received and read from the SIRB register.
7. Set the CSIM register's CRXE bit to 0 to disable the receive operation (end of receive operation).

Remark: The SO3 pin is invalid and maintains its signal level, as the output latch is disabled.

(3) Single Buffer Transfer Mode (Master Mode, Transmit/Receive Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 1, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-27: Single Buffer Transfer Mode (Master, Transmit/Receive) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE and CRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and transmission data in the SFCS and SFDB registers in this order.
6. Check for a transmission to be finished (e.g. by monitoring the INTC3nI interrupt). If so, read the SIRB register.

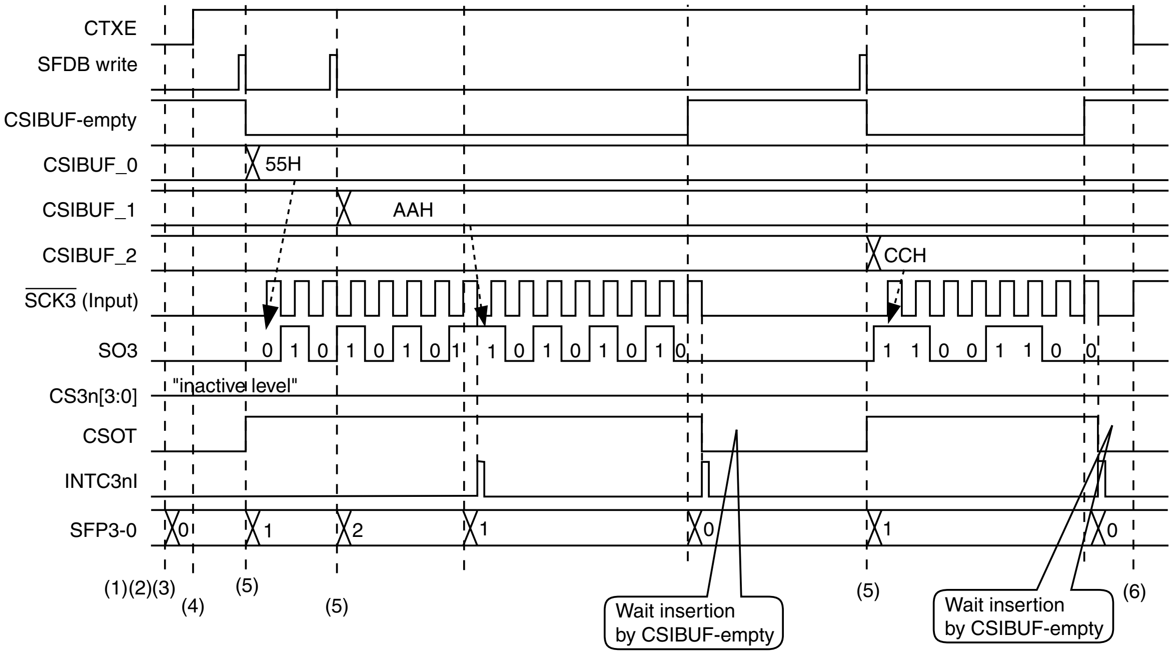
   Repeat steps (5) and (6) until the last element is send/received and read from the SIRB register.

7. Set the CSIM register's CTXE and CRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

(4) Single Buffer Transfer Mode (Slave Mode, Transmit Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 1, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-28: Single Buffer Transfer Mode (Slave, Transmit Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE bit to 1 to enable transmission.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write transmission data in the SFDB register. (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)

   Repeat step (5) until the last element to be transmitted is written in the SFDB register.

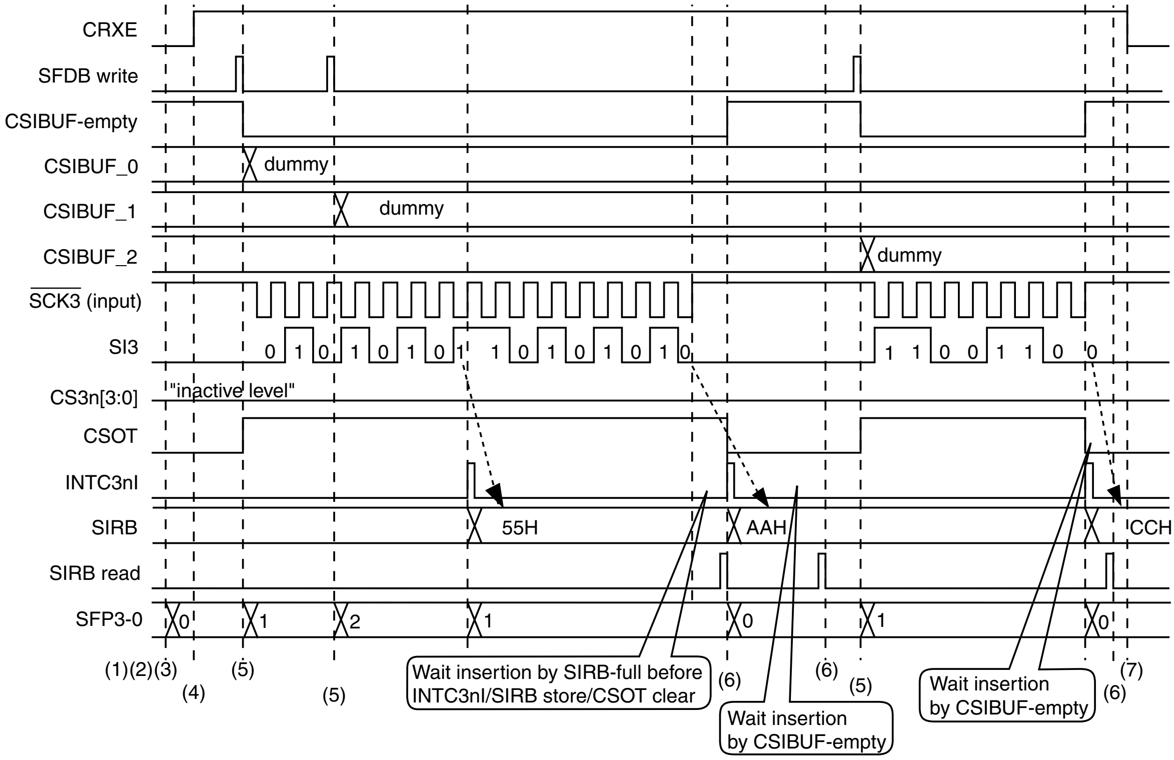
6. Set the CSIM register's CTXE bit to 0 to disable transmission (end of transmission).

To continue transmission, repeat step (5) before executing step (6).

(5) Single Buffer Transfer Mode (Slave Mode, Receive Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-29: Single Buffer Transfer Mode (Slave, Receive Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CRXE bit to 1 to enable the receive operation.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write dummy transmission data in the SFDB register (start-of-receive trigger). (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)
6. Check for a reception to be completed (e.g. by monitoring the INTC3nI interrupt). If so, read the SIRB register.

   Repeat steps (5) and (6) until the last element is received and read from the SIRB register.

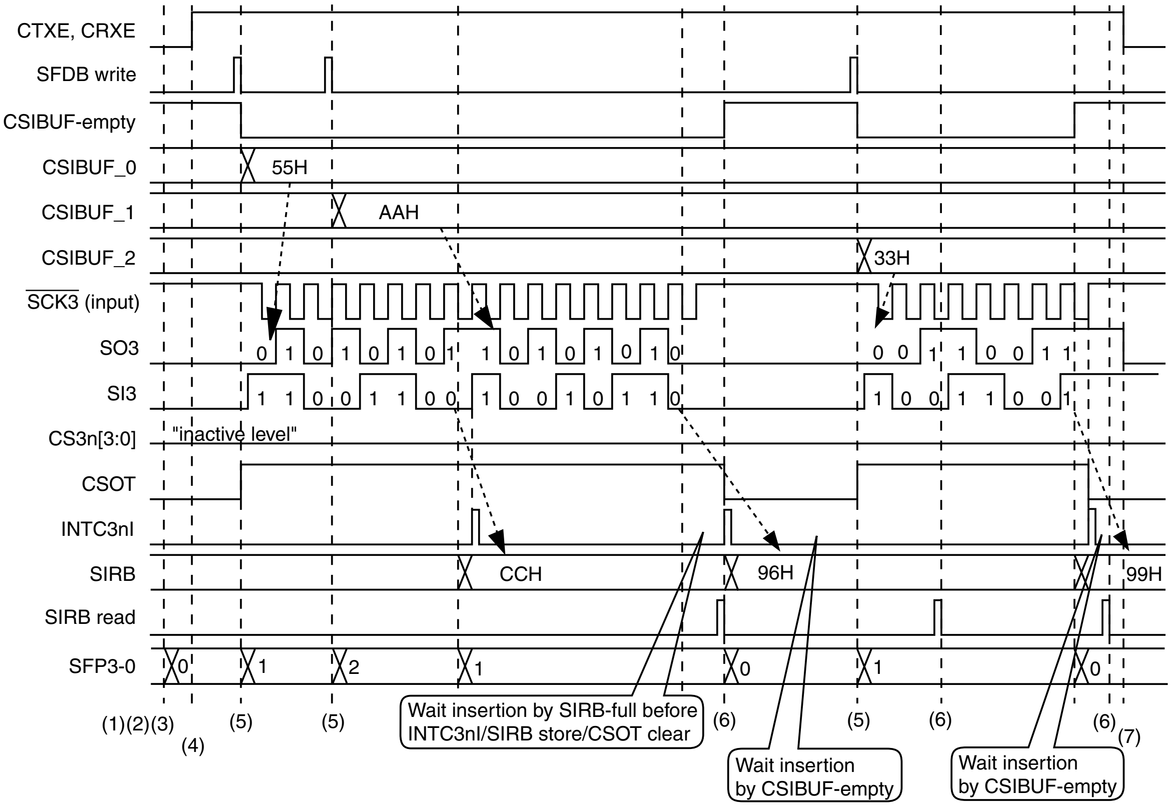
7. Set the CSIM register's CRXE bit to 0 to disable the receive operation (end of receive operation).

Remark: The SO3n pin is invalid and maintaining its signal level, as the output latch is disabled.

(6) Single Buffer Transfer Mode (Slave Mode, Transmit/Receive Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-30: Single Buffer Transfer Mode (Slave, Transmit/Receive) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transmit mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE and CRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the SFA register's SFFUL bit is set to 0, then write transmission data in the SFDB register. (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)
6. Check for a reception to be completed (e.g. by monitoring the INTC3nI interrupt). If so, read the SIRB register.

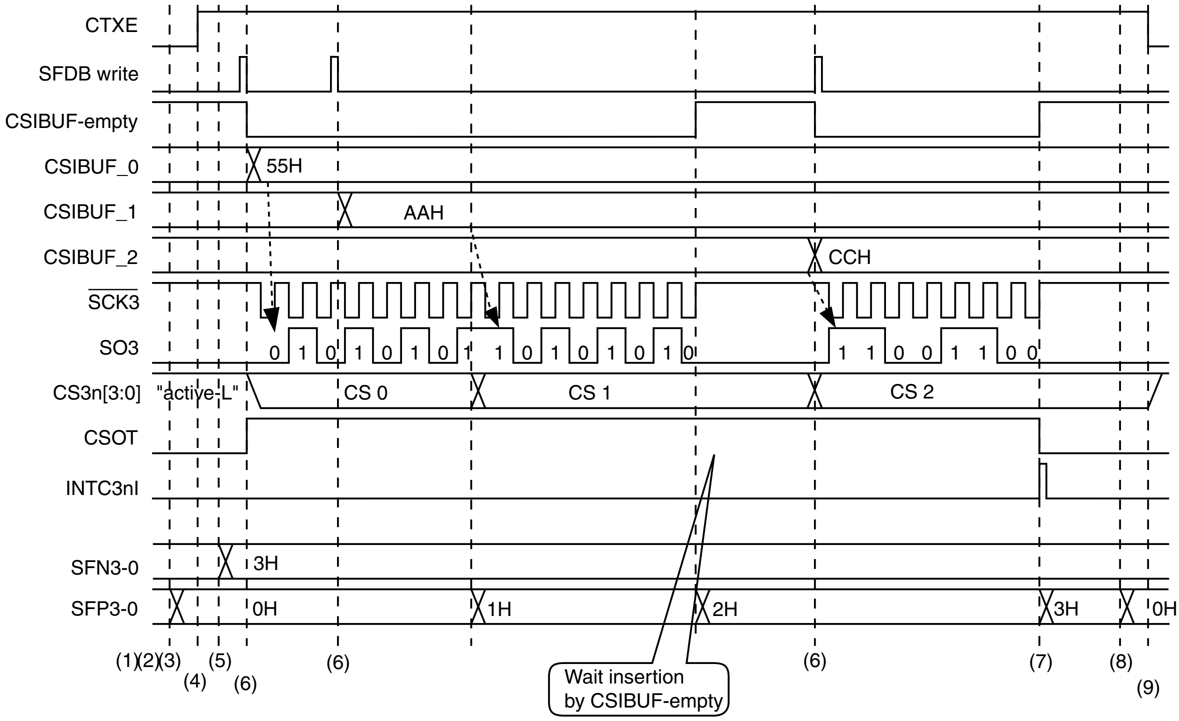
   Repeat steps (5) and (6) until the last element is send/received and read from the SIRB register.

7. Set the CSIM register's CTXE and CRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

(7) FIFO Buffer Transfer Mode (Master Mode, Transmit Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-31: FIFO Buffer Transfer Mode (Master, Transmit Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and transmission data in the SFCS and SFDB registers in this order.
7. Wait for the transmissions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Write “1” in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

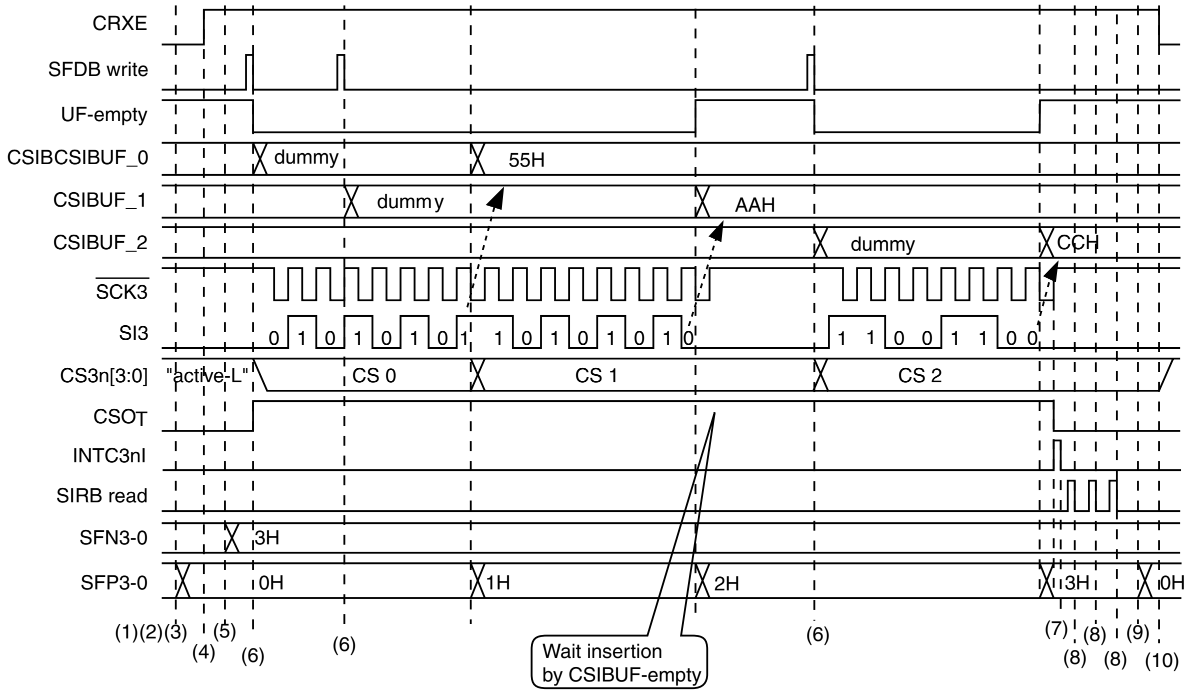
   To continue transmission, repeat steps (5) - (8).

9. Set the CSIM register's CTXE bit to 0 to disable transmission (end of transmission).

(8) FIFO Buffer Transfer Mode (Master Mode, Receive Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-32: FIFO Buffer Transfer Mode (Master, Receive Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and dummy transmission data in the SFCS and SFDB registers in this order (start-of-receive trigger).
7. Wait for the receptions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Read the received data by multiple read of the SIRB register (= sequential read from the FIFO).
9. Write “1” in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

   To continue reception, repeat steps (5) - (9).

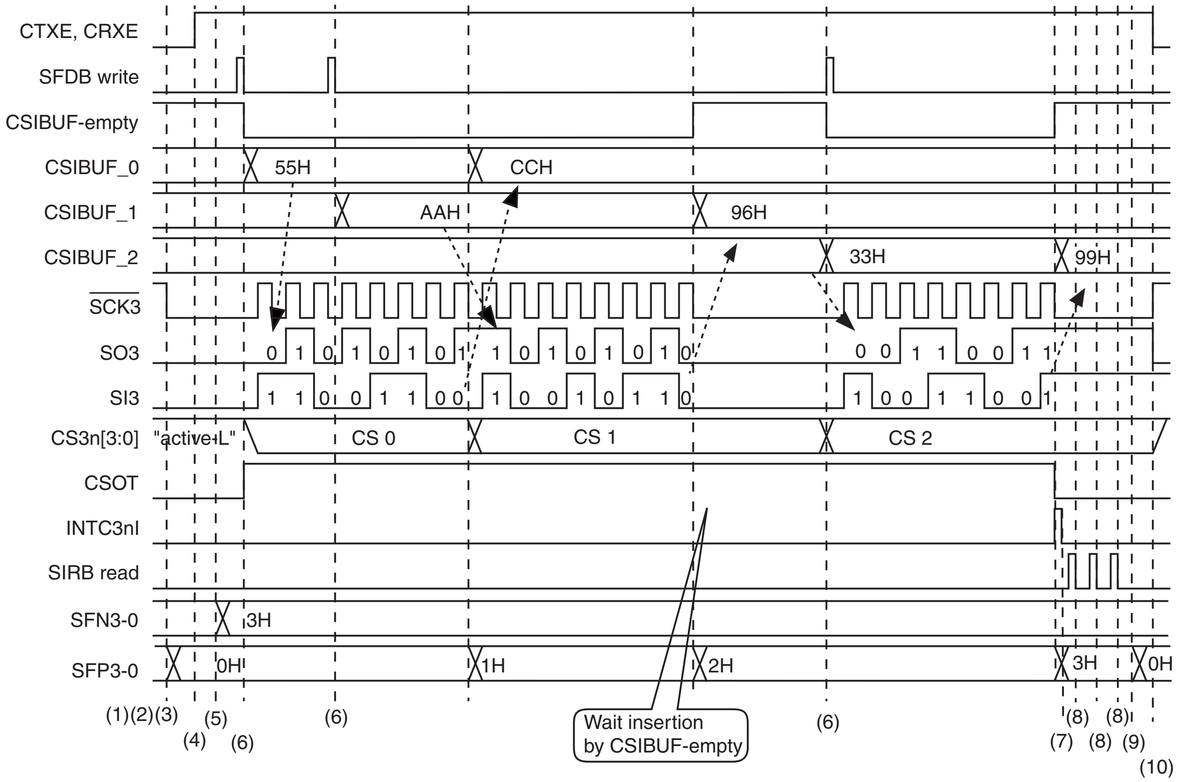
10. Set the CSIM register's CRXE bit to 0 to disable the receive operation (end of receive operation).

Remark: The SO3n pin is invalid and maintains its signal level, as the output latch is disabled.

(9) FIFO Buffer Transfer Mode (Master Mode, Transmit/Receive Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 1, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to "active low":

Figure 14-33: FIFO Buffer Transfer Mode (Master, Transmit/Receive) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write "1" in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE and CRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write chip-select data and transmission data in the SFCS and SFDB registers in this order.
7. Wait for the transmissions/receptions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Read the received data by multiple read of the SIRB register (= sequential read from the FIFO).
9. Write "1" in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

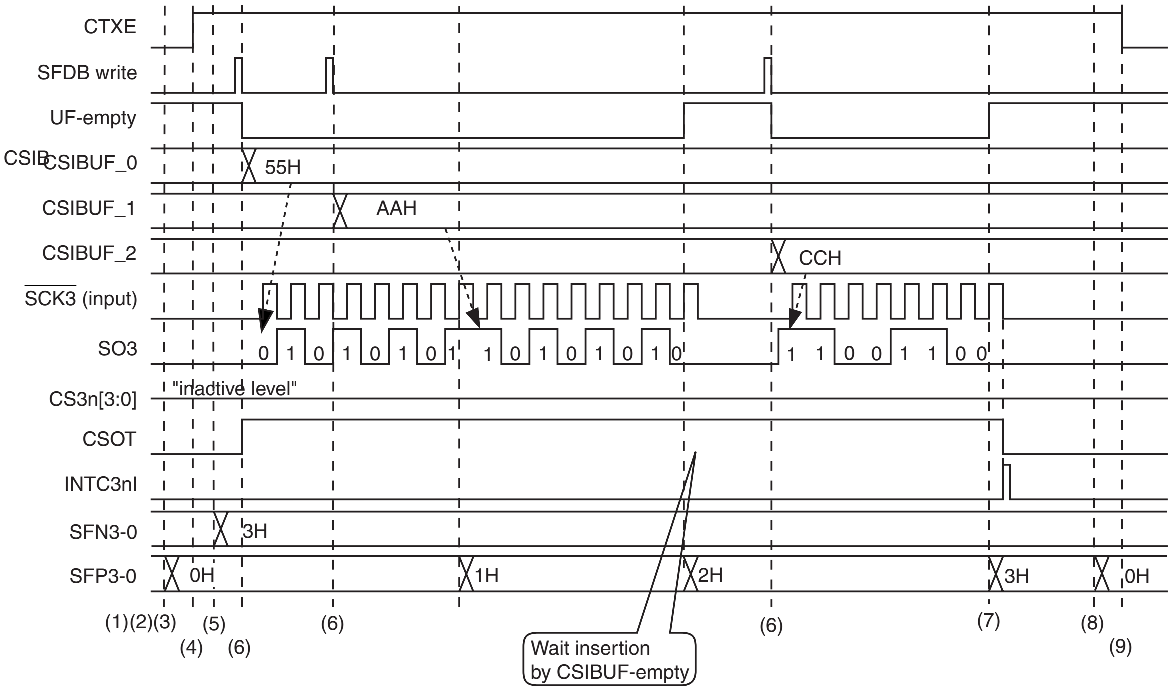
   To continue transmission/reception, repeat steps (5) - (9).

10. Set the CSIM register's CTXE and CRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

(10) FIFO Buffer Transfer Mode (Slave Mode, Transmit Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 1, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-34: FIFO Buffer Transfer Mode (Slave, Transmit Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write transmission data in the SFDB register. (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)
7. Wait for the transmissions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Write “1” in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

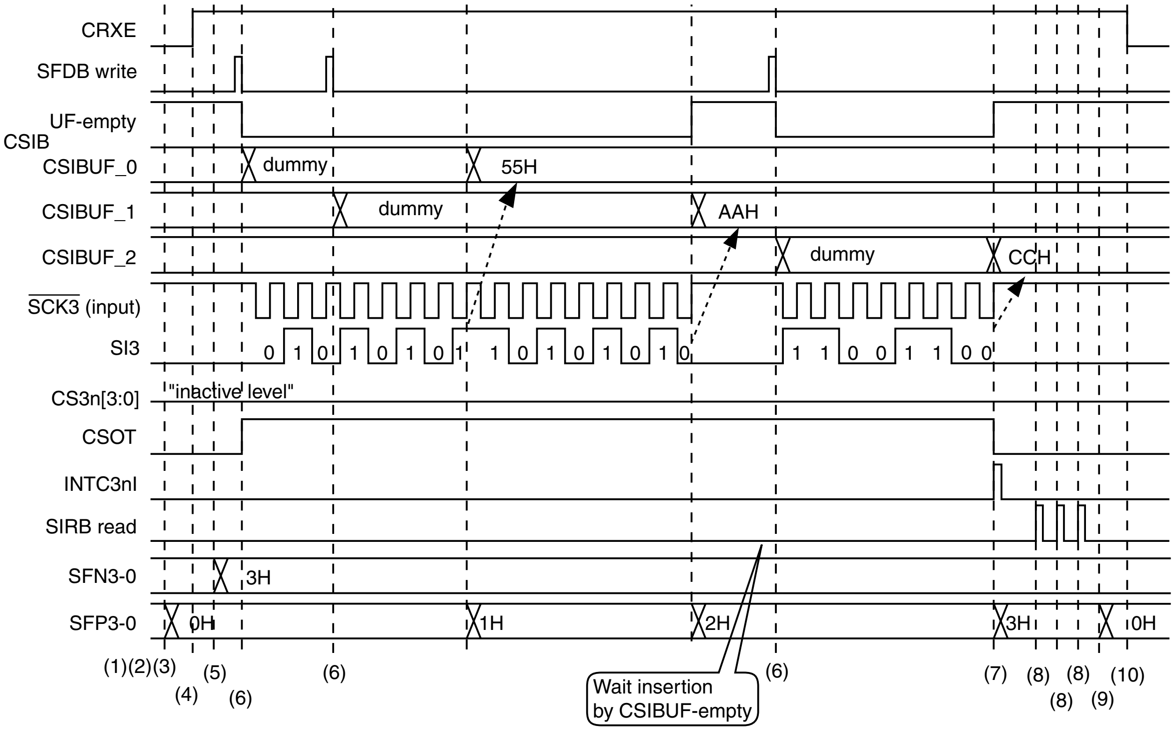
   To continue transmission, repeat steps (5) - (8).

9. Set the CSIM register's CTXE bit to 0 to disable transmission (end of transmission).

(11) FIFO Buffer Transfer Mode (Slave Mode, Receive Only Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 0, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-35: FIFO Buffer Transfer Mode (Slave, Receive Only) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write dummy transmission data in the SFDB register (start-of-receive trigger). (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)
7. Wait for the receptions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Read the received data by multiple read of the SIRB register (= sequential read from the FIFO).
9. Write “1” in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

   To continue reception, repeat steps (5) - (9).

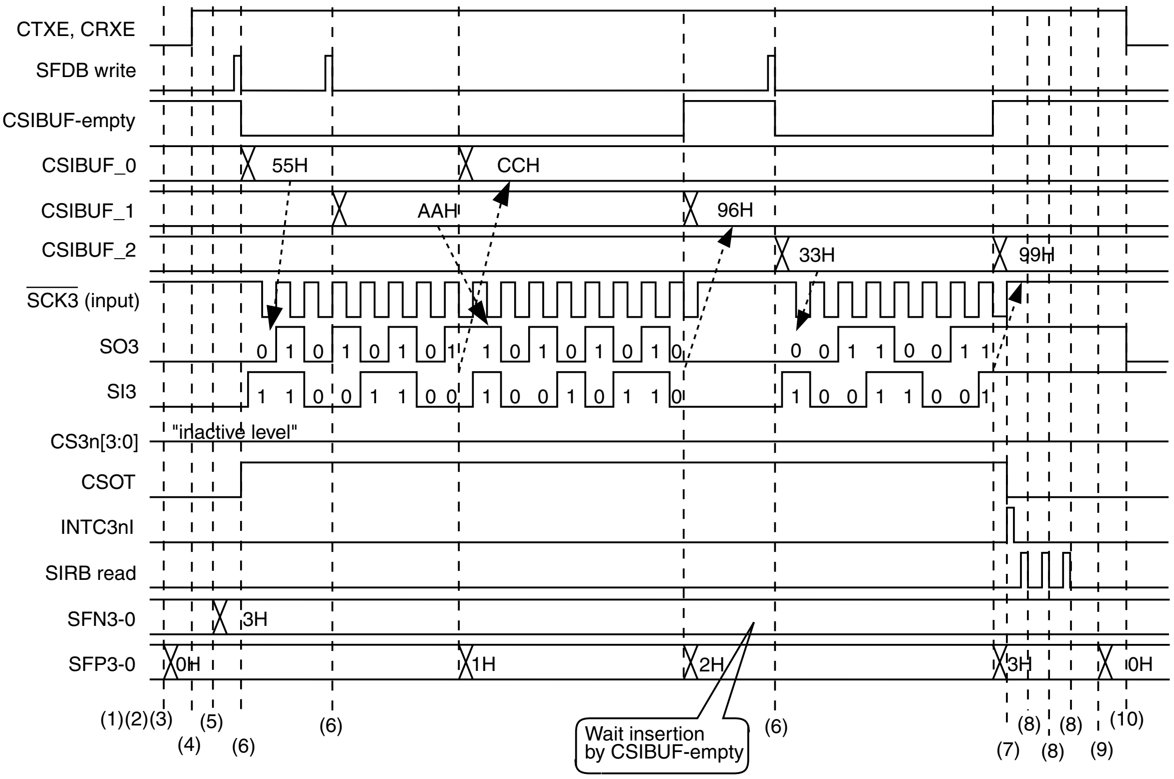
10. Set the CSIM register's CRXE bit to 0 to disable the receive operation (end of receive operation).

Remark: The SO3n pin is invalid and maintains its signal level, as the output latch is disabled.

(12) FIFO Buffer Transfer Mode (Slave Mode, Transmit/Receive Mode)

MSB first (DIR = 0), no INTC3nI delay (CSIT = 0), transmission wait disabled (CSWE = 0), CS inactive disabled (CSMD = 0), CKP = 0, DAP = 1, transmission data length of 8 bits (CCL[3:0] = [1,0,0,0]), active levels of all chip selects set to “active low”:

Figure 14-36: FIFO Buffer Transfer Mode (Slave, Transmit/Receive) Timing



1. Set the CSIM register's POWER bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CSIC and CSIL registers to specify the transfer mode.
3. Write “1” in the SFA register's FPCLR bit to clear all FIFO pointers.
4. Specify the transfer mode using the CSIM register's TRMD, DIR, and CSIT bits; at the same time, set the CTXE and CRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the SFN register's SFN[3:0] bits.
6. Make sure that the SFA register's SFFUL bit is set to 0, then write transmission data in the SFDB register. (In the slave mode, there is no need to set data in the SFCS register, as the chip select pins CS3n[3:0] are not used.)
7. Wait for the transmissions/receptions to be completed (e.g. by monitoring the INT3C3nI interrupt).
8. Read the received data by multiple read of the SIRB register (= sequential read from the FIFO).
9. Write “1” in the SFA register's FPCLR bit and clear all FIFO pointers for the next transmission.

   To continue transmission, repeat steps (5) - (9).

10. Set the CSIM register's CTXE and CRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

Chapter 15 DMA Functions (DMA Controller)

V850E/RS1 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfers. It supports transfers of data between internal or external memory and peripheral I/O, between internal and external memory, or between peripheral I/Os. DMA requests can be issued by the on-chip peripheral I/O (like serial interface, real-time pulse unit, or A/D converter), by interrupts from external input pins, or they can be triggered by software.

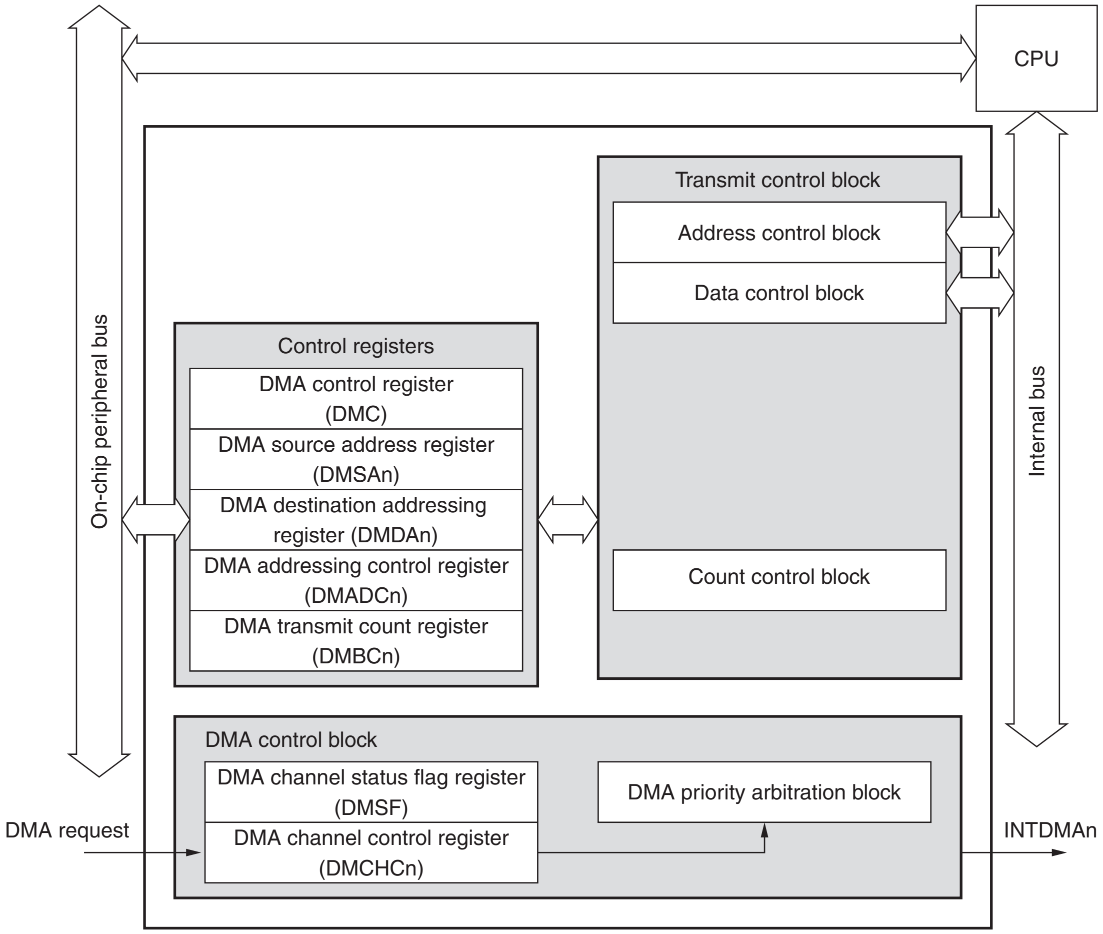
15.1 Features

- 6 independent DMA channels
- Transfer unit: 8/16/32 bits
- Maximum transfer count: 65,536 (2^{16})
- Transfer type: Two-cycle transfer
- Transfer mode:
  - Single transfer mode
  - Block transfer mode
  - Fixed channel transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O or interrupts from external input pin
  - Requests by software trigger
- Transfer objects
  - Internal RAM ↔ peripheral I/O
  - Peripheral I/O ↔ peripheral I/O
  - Internal RAM ↔ external memory[Note]
  - External memory ↔ external memory[Note]

Note: This is only for µPD70F3403 and µPD70F3403A

15.2 Configuration

Figure 15-1: DMA Block Diagram


CPU
On-chip peripheral bus
Internal bus
Transmit control block
Address control block
Data control block
Count control block
Control registers
DMA control register (DMC)
DMA source address register (DMSAn)
DMA destination addressing register (DMDAn)
DMA addressing control register (DMADCn)
DMA transmit count register (DMBCn)
DMA control block
DMA channel status flag register (DMSF)
DMA channel control register (DMCHCn)
DMA priority arbitration block
DMA request
INTDMAn

Remark: n = 0 to 5

15.3 Control Registers

(1) DMA control register (DMC)

The DMC register controls the operation and the clock supply of the DMA controller. It can be read or written in 8-bit or 1-bit units.
Initial value is 00H by reset.

Figure 15-2: DMA Control Register (DMC) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
DMC	STPDIS	IDMEN	STPSET	STPCLR	0	0	0	POWER	FFFFFE00H	00H
Reset value	0	0	0	0	0	0	0	0		
R/W	R	R/W	R/W	R/W	R	R	R	R/W		

STPDIS	DMA Transfer Interrupt Status Flag
0	All DMA transfers operational.
1	All DMA transfers are stopped. This bit is set by the IDMEN bit = 1 and input of NMI signal, or by setting the STPSET bit. It is cleared by applying the STPCLR bit.

IDMEN	DMA Transfer Interruption Permission by NMI
0	A NMI does not interrupt any on-going DMA transfer. When a NMI occurs while a DMA transfer is in progress, the processing of the NMI will be delayed until the DMA transfer is finished.
1	A NMI interrupts and stops all on-going DMA transfers. When a NMI occurs, all DMA transfers are interrupted and stopped. The STPDIS bit is set.

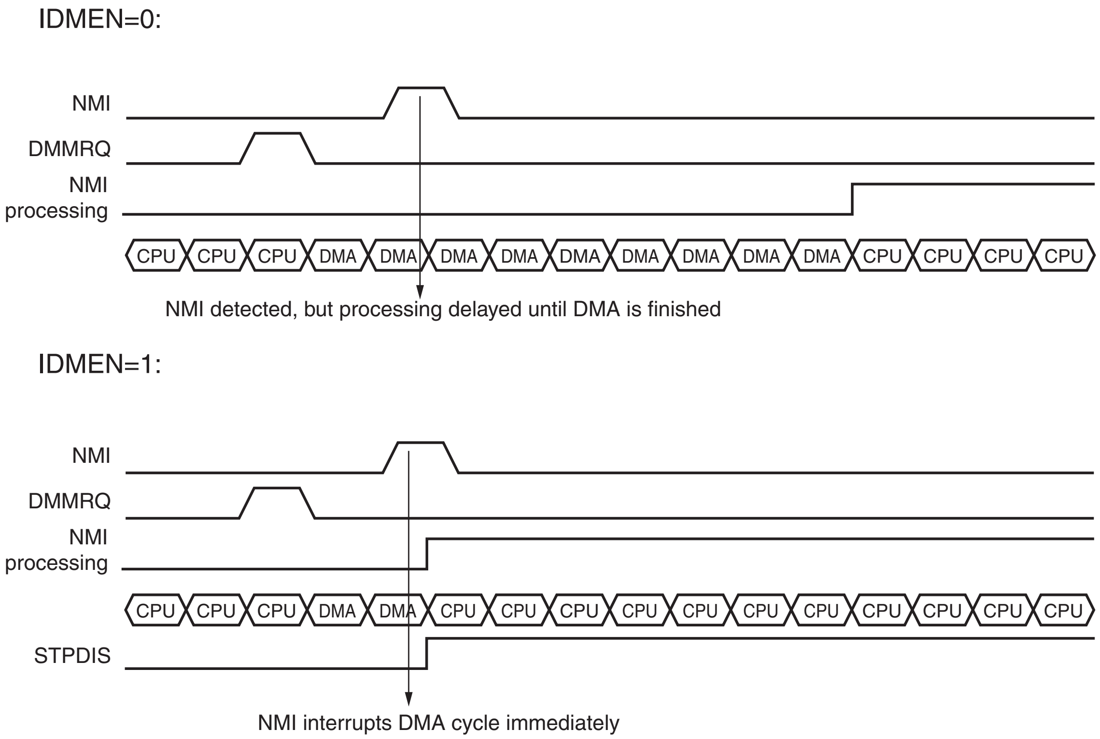
Caution: Write is permitted only when POWER = 0.

STPSET	Software DMA Transfer Interruption
0	No change
1	DMA transfer interrupted by software. When this bit set to 1, all DMA transfers are interrupted and stopped. The STPDIS bit is set.

Remark: This bit is a trigger bit only. When reading the bit, the read value is always 0.

Figure 15-3 below illustrates the start of a NMI processing for the different IDMEN bit settings. In the example a DMA transfer is set up as a block transfer with a transfer count of 8. While in the upper timing diagram the processing of the NMI is delayed until the 8th transfer is completed when IDMEN=0, the NMI interrupts and stops the DMA transfer immediately in the lower figure (IDMEN=1).

Figure 15-3: IDMEN Bit and NMI Handling


IDMEN=0:
NMI
DMMRQ
NMI processing
CPU CPU CPU DMA DMA DMA DMA DMA DMA DMA DMA DMA CPU CPU CPU CPU
NMI detected, but processing delayed until DMA is finished
IDMEN=1:
NMI
DMMRQ
NMI processing
CPU CPU CPU DMA DMA CPU CPU CPU CPU CPU CPU CPU CPU CPU CPU CPU
STPDIS
NMI interrupts DMA cycle immediately

STPCLR	Transfer Interruption Clear Trigger Bit
0	No change
1	Release all interrupted and stopped DMA transfers and resume operation. When this bit set to 1, SPTDIS bit is cleared to 0 and pending DMA transfers are resumed.

Cautions: 1. Do not set the STPSET and STPCLR bits at the same time. If both bits are set at the same time, STPCLR has priority over STPSET.

2. If STPCLR is set at the same time as the occurrence of a NMI that would set the STPDIS bit, the STPDIS bit will not be set.

Remark: This bit is a trigger bit only. When reading the bit, the read value is always 0.

POWER	DMA Controller Operation Enable
0	The clock supply to the DMA controller is stopped and all operation disabled. The DMA channel control registers (DMCHCn) and the internal circuits are reset.
1	The clock supply to the DMA controller is started and the DMA operation enabled. Set the POWER bit to 1 before initializing and using the DMA controller.

Cautions: 1. Do not set the IDMEN bit while POWER = 1.

2. Disable all DMA transfers (clear EN bit of DMCHCn registers) before stopping the DMA (POWER cleared to 0)

(2) DMA channel status flag register (DMSF)

The DMSF register shows the DMA transfer status of each DMA channel. It is a direct mirror of all ACF bits (DMCHCn register). It can be read in 16-bit units only.
The register is initialized by POWER = 0. Initial value is 0000H by reset.

Figure 15-4: DMA Channel Status Flag Register (DMSF) Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMSF	0	0	0	0	0	0	0	0	0	0	ACF (ch5)	ACF (ch4)	ACF (ch3)	ACF (ch2)	ACF (ch1)	ACF (ch0)	FFFFFE04H	0000H
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Refer to **(7)"DMA channel control register (DMCHCn)" on page 513** for the explanation of the ACF bits.

(3) DMA source address register (DMSAn)

The DMSAn register is used to set the DMA source address. When the TDIR bit (DMADCn register) is 0, it holds the address the data is transferred from (source), for TDIR=1 the register holds the address data is transferred to (destination). The register can be read or written in 32-bit or 16-bit units.

Caution: Write to DMSAn is permitted only when EN = 0 (DMCHCn register).

Figure 15-5: DMA Source Address Register (DMSAn) Format (1/2)

DMSA0H=FFFFFE0AH, DMSA1H=FFFFFE16H, DMSA2H=FFFFFE22H,
DMSA3H=FFFFFE2EH, DMSA4H=FFFFFE3AH, DMSA5H=FFFFFE46H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMSAnH	SC2	SC1	SC0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16		undef.
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

DMSA0L=FFFFFE08H, DMSA1L=FFFFFE14H, DMSA2L=FFFFFE20H,
DMSA3L=FFFFFE2CH, DMSA4L=FFFFFE38H, DMSA5L=FFFFFE44H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMSAnL	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0		undef.
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Figure 15-5: DMA Source Address Register (DMSAn) Format (2/2)

SC2	SC1	SC0	Selected chip and selection area
0	0	0	CS0 is selected.
0	0	1	CS1 is selected.
0	1	0	Setting prohibited
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Cautions: 1. When the DMA transfer object is in the external memory area, set the valid chip select signal for SC[2:0].

2. If the DMA transfer object is internal memory and peripheral I/O area, write “000” to SC[2:0].

SA25 - SA0	DMA transfer source address (address signals A25 - A0). If address increment is selected for the DMA transfer, the address will be updated after each transfer to the next transfer address.

The number of relevant bits of SA25 - SA0 is as follows:

Transfer object	Setting of SA25 - SA0
Internal RAM	Set the lower 16-bit address for SA15 - SA0
External Memory	Set the lower 26-bit address for SA25 - SA0
On-chip Peripheral I/O	Set the lower 14-bit address for SA13 - SA0

Caution: SA0 and DA0 have to be set to 0, when DMA transfer size is halfword.
SA0 and DA0 and SA1 and DA1 have to be set to 0, when DMA transfer size is word.

(4) DMA destination address register (DMDAn)

The DMDAn register is used to set the DMA destination address. When the TDIR bit (DMADCn register) is 0, it holds the address the data is transferred to (destination), for TDIR=1 the register holds the address data is transferred from (source). The register can be read or written in 32-bit or 16-bit units.

Caution: Write to DMDAn is permitted only when EN = 0 (DMCHCn register).

Figure 15-6: DMA Destination Address Register (DMDAn) Format (1/2)

DMDA0H=FFFFFE0EH, DMDA1H=FFFFFE1AH, DMDA2H=FFFFFE26H,
DMDA3H=FFFFFE32H, DMDA4H=FFFFFE3EH, DMDA5H=FFFFFE4AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMDAnH	DC2	DC1	DC0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16		undef.
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

DMDA0L=FFFFFE0CH, DMDA1L=FFFFFE18H, DMDA2L=FFFFFE24H,
DMDA3L=FFFFFE30H, DMDA4L=FFFFFE3CH, DMDA5L=FFFFFE48H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMDAnL	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0		undef.
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

DC2	DC1	DC0	Selected chip and selection area
0	0	0	CS0 is selected.
0	0	1	CS1 is selected.
0	1	0	Setting prohibited
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Cautions: 1. When the DMA transfer object is in the external memory area, set the valid chip select signal for DC[2:0].

2. If the DMA transfer object is internal memory and peripheral I/O area, write “000” to DC[2:0].

Figure 15-6: DMA Destination Address Register (DMDAn) Format (2/2)

DA25 - DA0	DMA transfer destination address (address signals A25 - A0). If address increment is selected for the DMA transfer, the address will be updated after each transfer to the next transfer address.

The number of relevant bits of DA25 - DA0 is as follows:

Transfer object	Setting of DA25 - DA0
Internal RAM	Set the lower 16-bit address for DA15 - DA0
External Memory	Set the lower 26-bit address for DA25 - DA0
On-chip Peripheral I/O	Set the lower 14-bit address for DA13 - DA0

Caution: SA0 and DA0 have to be set to 0, when DMA transfer size is halfword.
SA0 and DA0 and SA1 and DA1 have to be set to 0, when DMA transfer size is word.

(5) DMA transfer count register (DMBCn)

The DMBCn register sets the transfer count for DMA channel n. It holds the number of remaining transfers during a DMA transfer. The value is decremented with each transfer and hold a value of 0 at the end of a DMA transfer. The register can be read or written in 16-bit units.

Caution: Write to DMBCn is permitted only when EN = 0 (DMCHCn register).

Figure 15-7: DMA Transfer Count Register (DMBCn) Format

DMBC0=FFFFFE10H, DMBC1=FFFFFE1CH, DMBC2=FFFFFE28H,
DMBC3=FFFFFE34H, DMBC4=FFFFFE40H, DMBC5=FFFFFE4CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DMBCn	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0		undef.
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

BC15 - BC0	DMA Transfer Counter
0000H	1 transfer remaining [Note]
0001H	2 transfers remaining
...	...
FFFFH	65536 transfers remaining

Note: When the value of DMBCn register is read as 0, it can either mean that there is one remaining transfer pending, or that the transfer is completed. The EN bit can be used to determine if a further transfer is pending or if the transfer is completed.

(6) DMA addressing control register (DMADCn)

The DMADCn register control the address handling for each DMA channel. It can be read or written in 8-bit or 16-bit units.
Initial value is F000H by reset.

Caution: Write to DMADCn is permitted only when EN = 0 (DMCHCn register).

Figure 15-8: DMA Addressing Control Register (DMADCn) Format (1/2)

DMADC0H=FFFFFE13H, DMADC1H=FFFFFE1FH, DMADC2H=FFFFFE2BH,
DMADC3H=FFFFFE37H, DMADC4H=FFFFFE43H, DMADC5H=FFFFFE4FH

Symbol	7	6	5	4	3	2	1	0	Address	After reset
DMADCnH	TS1	TS0	TD1	TD0	SAD	0	DAD	0		F0H
Reset value	1	1	1	1	0	0	0	0		
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R		

DMADC0L=FFFFFE12H, DMADC1L=FFFFFE1EH, DMADC2L=FFFFFE2AH,
DMADC3L=FFFFFE36H, DMADC4L=FFFFFE42H, DMADC5L=FFFFFE4EH

Symbol	7	6	5	4	3	2	1	0	Address	After reset
DMADCnL	DS1	DS0	0	0	TM1	TM0	0	TDIR		00H
Reset value	0	0	0	0	0	0	0	0		
R/W	R/W	R/W	R	R	R/W	R/W	R	R/W		

TS1	TS0	Source Address Area Selector (SA[25:0] bits)
0	0	Source address is external I/O
0	1	Setting prohibited
1	0	Source address is Internal RAM
1	1	Source address is on-chip peripheral I/O

TD1	TD0	Destination Address Area Selector (DA[25:0] bits)
0	0	Destination address is external I/O
0	1	Setting prohibited
1	0	Destination address is internal RAM
1	1	Destination address is on-chip peripheral I/O

SAD	Source Address Count Mode (SA[25:0] bits)
0	Source address is incremented after each transfer.
1	Source address is fixed

Figure 15-8: DMA Addressing Control Register (DMADCn) Format (2/2)

DAD	Destination Address Count Mode (DA[25:0] bits)
0	Destination address is incremented after each transfer.
1	Destination address is fixed

DS1	DS0	Selected DMA Transfer Size
0	0	Byte (8-bit)
0	1	Half word (16-bit)
1	0	Word (32-bit)
1	1	Setting prohibited

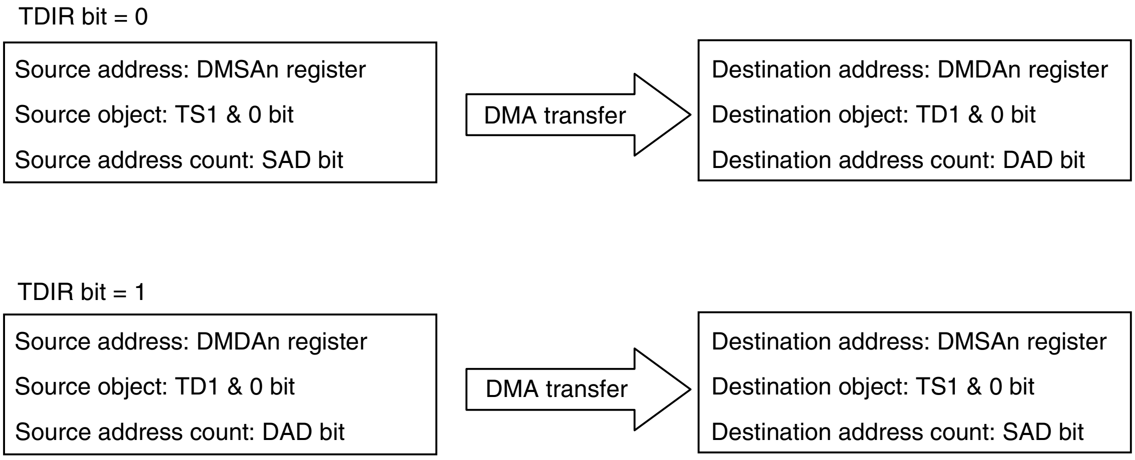
Remark: If address incrementation is selected for the source address and/or the destination address, the address is incremented based on the DMA transfer size. For 8-bit size, the address is incremented by 1, for 16-bit by 2, and for 32-bit by 4.

TM1	TM0	Selected DMA Transfer Mode
0	0	Single transfer mode (starting value)
0	1	Fixed Channel transfer mode
1	0	Setting prohibited
1	1	Block transfer mode

Please refer to **15.5 on page 522** and **Table 15-3 on page 524** for details on and a comparison of the transfer modes.

TDIR	DMA Transfer Direction
0	Transfer from source → destination
1	Transfer from destination → source

Figure 15-9: Effect of TDIR Flag on DMA Transfer


TDIR bit = 0
Source address: DMSAn register
Source object: TS1 & 0 bit
Source address count: SAD bit
DMA transfer
Destination address: DMDAn register
Destination object: TD1 & 0 bit
Destination address count: DAD bit
TDIR bit = 1
Source address: DMDAn register
Source object: TD1 & 0 bit
Source address count: DAD bit
DMA transfer
Destination address: DMSAn register
Destination object: TS1 & 0 bit
Destination address count: SAD bit

(7) DMA channel control register (DMCHCn)

The DMCHCn register controls the DMA transfer operation mode. It can be read or written in 8-bit or 1-bit units.
The RQST, ACF, FCLR, STG and EN bits are initialized by POWER=0. Initial value is 00H by reset.

Figure 15-10: DMA Channel Control Register (DMCHCn) Format (1/4)

DMCHC0=FFFFFDE0H, DMCHC1=FFFFFDE1H, DMCHC2=FFFFFDE2H,
DMCHC3=FFFFFDE3H, DMCHC4=FFFFFDE4H, DMCHC5=FFFFFDE5H

Symbol	7	6	5	4	3	2	1	0	Address	After reset
DMCHCn	0	0	TCS	RQST	ACF	FCLR	STG	EN		00H
R/W	R	R	R/W	R	R	R/W	R/W	R/W		

TCS	DMA Interrupt Timing Generation
0	INTDMAn is a direct mirror of the selected DMA request input signal of channel n
1	INTDMAn timing is based on DMA execution.

Cautions: 1. Write to TCS bit is permitted only when POWER bit = 0.

2. Before changing the TCS bit, mask the corresponding INTDMAn interrupt. After changing the TCS bit, clear any pending INTDMAn interrupt request flag.

The DMA controller has the option to mirror the selected input trigger (IFCn[5:0] bits of the DTFRn register) to the INTDMA signal. This mirroring is independent from the status of the DMA channel (disabled, enabled, DMA in progress or ended). The mirror function is used in devices where DMA interrupts are to be shared with interrupts also serving as interrupt request signals to the DMA controller.
For V850E/RS1, there is no sharing of interrupts needed, as each of the DMA channels has its own interrupt assigned on interrupt controller side. Therefore it is highly recommended NOT to use the TCS=0 setting.

Please refer to section **15.4.4 on page 521** for further explanation of the TCS bit and the resulting timing of the DMA interrupt.

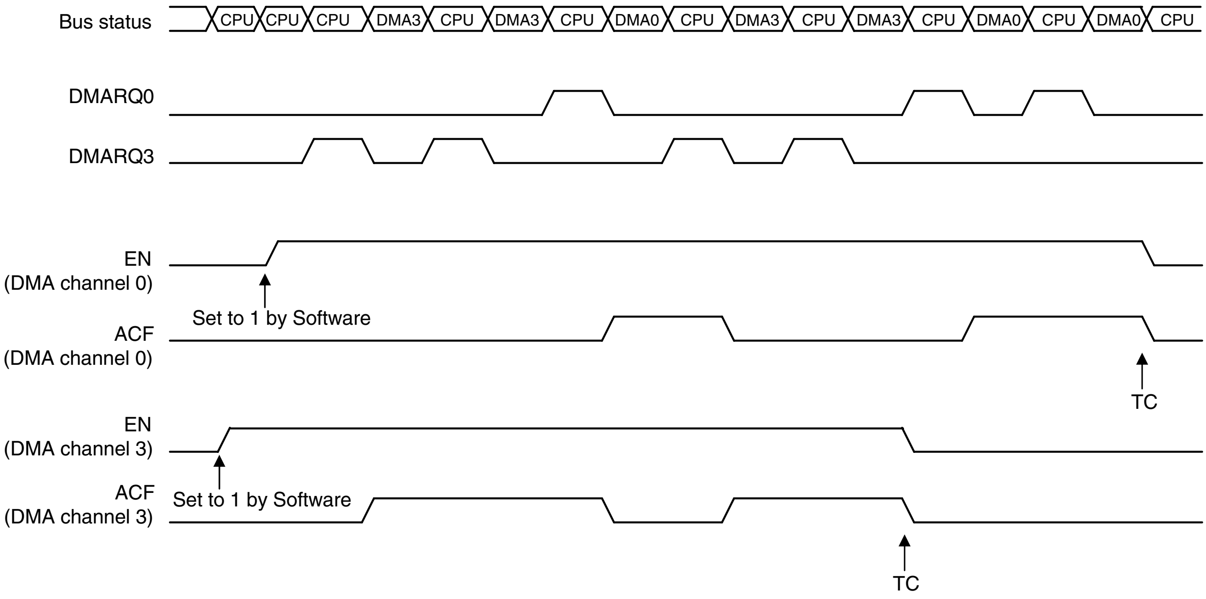
RQST	Transfer Request Status Flag
0	No DMA transfer request pending. This bit is cleared either by start of the DMA transfer or by writing 0 to the FCLR bit.
1	DMA transfer request pending. This bit is set by a DMA request signal input or when writing 1 to the STG bit.

Figure 15-10: DMA Channel Control Register (DMCHCn) Format (2/4)

ACF	DMA Acknowledge Flag
0	No DMA transfer in progress.
1	Indicate the DMA transfer status. When ACF is set (1), the state is different depending on the transfer mode. **Single or Fixed Channel transfer mode:** DMA transfer has been executed at least one time, but the total number of transfers has not yet been reached. If other channels of DMA transfer are occurring, the non-corresponding ACF bit will be cleared. **Block transfer mode:** At least one DMA transfer has been executed, but further transfers are pending as the total numbers of transfers not yet reached. In Block Transfer mode, the CPU can not access the ACF bit during the transfer. If a NMI interrupt occurs during a block transfer mode, then the ACF bit will be read by CPU as an "1".

(a) ACF bit status in single transfer mode

DMBC0 = 02H (3-times transfer)
DMBC3 = 03H (4-times transfer)

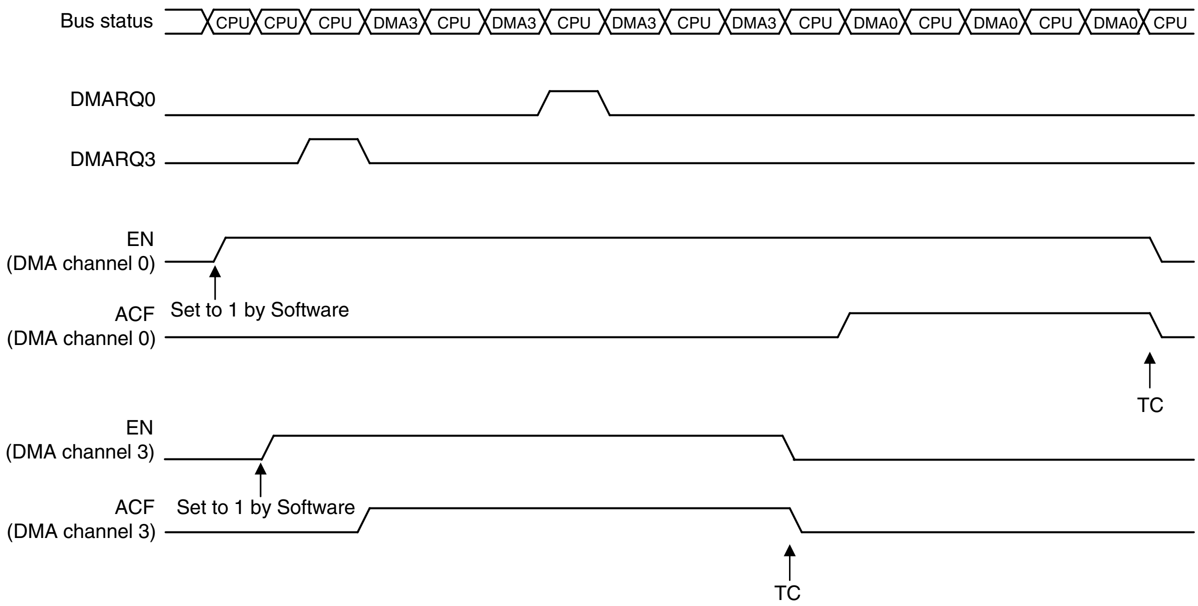


Remark: TC = Transfer Count achieved

Figure 15-10: DMA Channel Control Register (DMCHCn) Format (3/4)

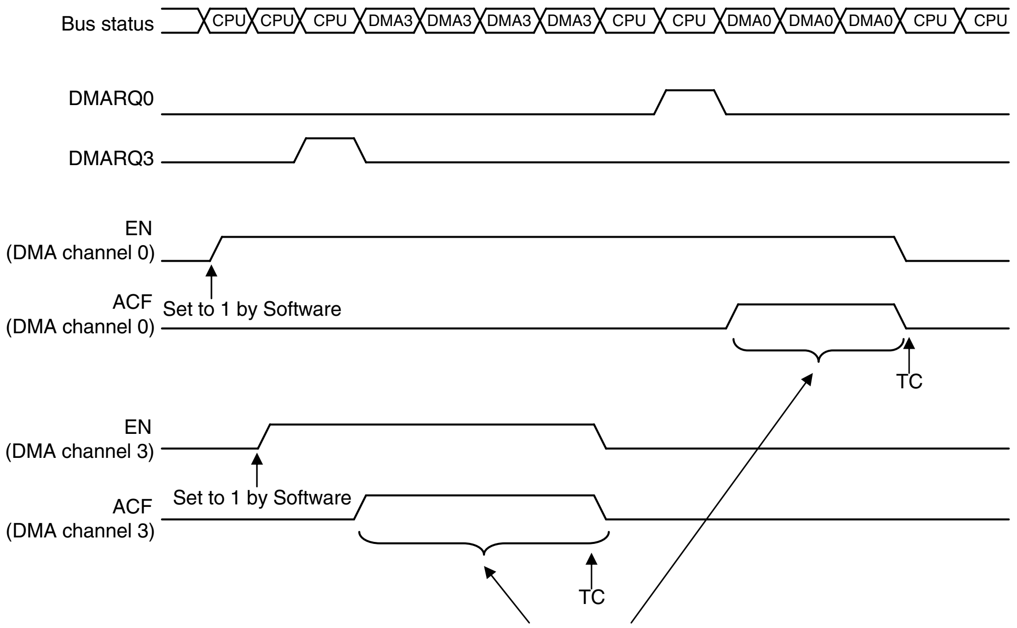
(b) ACF bit status in channel-fixed single transfer mode

DMBC0 = 02H (3-times transfer)
DMBC3 = 03H (4-times transfer)



(c) ACF bit status in block transfer mode

DMBC0 = 02H (3-times transfer)
DMBC3 = 03H (4-times transfer)



In this example, the internal bus is occupied by the DMA during these times.
Due to this behavior, the ACF bit during these times cannot be read by the CPU.
If a NMI interrupt occurs during a block transfer mode, then the ACF bit will be read by CPU as an“1”.

Figure 15-10: DMA Channel Control Register (DMCHCn) Format (4/4)

FCLR	DMA Request Clear Trigger
0	No change
1	Clear any pending DMA transfer request.

Caution: Write to FCLR = 1 is permitted only when EN = 0 (DMCHCn register).

Remarks: **1.** This bit is a trigger bit only. When reading the bit, the read value is always 0.

2. The RQST flag is cleared two clock cycles after setting FCLR = 1.

STG	DMA Software Trigger
0	No change
1	Generate a DMA transfer trigger.

Caution: Do not set FCLR and STG bits at the same time. If set at the same time, FCLR gets priority.

Remark: This bit is a trigger bit only. When reading the bit, the read value is always 0.

EN	DMA Transfer Enable
0	DMA transfer is disabled or the total number of DMA transfers is finished. If EN set to 0 while a “Single” or “Fixed Channel” transfer is active, the DMA transfer is stopped (the DMA source address, destination address, and transfer count registers are held.). When EN is set to 1 again, the DMA transfer is re-started.
1	DMA transfer is enabled and/or the total number of DMA transfers is not finished.

(8) DMA trigger factor register (DTFRn)

The DTFRn register selects the DMA transfer start trigger through interrupt. The interrupt source selected in this register serves as trigger to start the DMA transfer.

The register can be read or written in 8-bit or 1-bit units. However, only bit 7 (DFn) can be read/ written in 1-bit units.

Initial value is 00H by reset.

Figure 15-11: DMA Trigger Factor Register (DTFRn) Format

DTFR0=FFFFFF40H, DTFR1=FFFFFF42H, DTFR2=FFFFFF44H,
DTFR3=FFFFFF46H, DTFR4=FFFFFF48H, DTFR5=FFFFFF4AH

Symbol	7	6	5	4	3	2	1	0	Address	After reset
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0		00H
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W		

DF	DMA transfer request flag
0	No DMA transfer request
1	DMA transfer request

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. Be sure to follow the steps below when changing the DTFRn register settings.

- **When the values to be set to bits IFCn5 to IFCn0 are not set to bits IFCm5 to IFCm0 of another channel (n = 0 to 3, m = 0 to 3, n ≠ m)**
  <1> Stop the DMAn operation of the channel to be rewritten (DCHCn.Enn bit = 0).
  <2> Change the DTFRn register settings. (Be sure to set DFn bit = 0 and change the settings in the 8-bit manipulation.)
  <3> Confirm that DFn bit = 0. (Stop the interrupt generation source operation beforehand.)
  <4> Enable the DMAn operation (Enn bit = 1).
- **When the values to be set to bits IFCn5 to IFCn0 are set to bits IFCm5 to IFCm0 of another channel (n = 0 to 3, m = 0 to 3, n ≠ m)**
  <1> Stop the DMAn operation of the channel to be rewritten (DCHCn.Enn bit = 0).
  <2> Stop the DMAm operation of the channel where the same values are set to bits IFCm5 to IFCm0 as the values to be used to rewrite bits IFCn5 to IFCn0 (DCHCm.Emm bit = 0).
  <3> Change the DTFRn register settings. (Be sure to set DFn bit = 0 and change the settings in the 8-bit manipulation.)
  <4> Confirm that bits DFn and DFm = 0. (Stop the interrupt generation source operation beforehand.)
  <5> Enable the DMAn operation (bits Enn and Emm = 1).

Cautions: 3. 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).

4. 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.

Table 15-1: Interrupt Source for DMA Trigger Factor Register (DTFRn) (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
0	0	0	0	0	0	DMA request by interrupt disabled
0	0	0	0	0	1	INTLVI
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	INTTQ0OV
0	0	1	0	1	1	INTTQ0CC0
0	0	1	1	0	0	INTTQ0CC1
0	0	1	1	0	1	INTTQ0CC2
0	0	1	1	1	0	INTTQ0CC3
0	0	1	1	1	1	INTTP0OV
0	1	0	0	0	0	INTTP0CC0
0	1	0	0	0	1	INTTP0CC1
0	1	0	0	1	0	INTTP1OV
0	1	0	0	1	1	INTTP1CC0
0	1	0	1	0	0	INTTP1CC1
0	1	0	1	0	1	INTTP2OV
0	1	0	1	1	0	INTTP2CC0
0	1	0	1	1	1	INTTP2CC1
0	1	1	0	0	0	INTTP3OV
0	1	1	0	0	1	INTTP3CC0
0	1	1	0	1	0	INTTP3CC1
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	INTUA0R
1	0	0	0	0	1	INTUA0T

Table 15-1: Interrupt Source for DMA Trigger Factor Register (DTFRn) (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
1	0	0	0	1	0	INTUA1R
1	0	0	0	1	1	INTUA1T
1	0	0	1	0	0	INTAD
1	0	0	1	0	1	INTC0ERR
1	0	0	1	1	0	INTC0WUP
1	0	0	1	1	1	INTC0REC
1	0	1	0	0	0	INTC0TRX
1	0	1	0	0	1	INTC30I
1	0	1	0	1	0	INTC30O
1	0	1	0	1	1	INTC31I
1	0	1	1	0	0	INTC31O
1	0	1	1	0	1	Setting prohibited
1	0	1	1	1	0	Setting prohibited
1	0	1	1	1	1	INTTQ1OV
1	1	0	0	0	0	INTTQ1CC0
1	1	0	0	0	1	INTTQ1CC1
1	1	0	0	1	0	INTTQ1CC2
1	1	0	0	1	1	INTTQ1CC3
1	1	0	1	0	0	Setting prohibited
1	1	0	1	0	1	Setting prohibited
1	1	0	1	1	0	INTC1ERR
1	1	0	1	1	1	INTC1WUP
1	1	1	0	0	0	INTC1REC
1	1	1	0	0	1	INTC1TRX

Remark: n = 0 to 5

15.4 DMA Transfer Rules

15.4.1 Transfer targets

Table 15-2 shows the available source and destination selections for the DMA transfer:

Table 15-2: Transfer Targets

		Destination			
		On-chip Peripheral I/O	Internal RAM	Internal ROM	External memory
Source	On-Chip peripheral I/O	√	√	—	√
	Internal RAM	√	—	—	√
	Internal ROM	—	—	—	—
	External memory[Note]	√[Note]	√[Note]	—[Note]	√[Note]

Remark: √: transfer possible
—: transfer not possible

Note: This is only for µPD70F3403 and µPD70F3403A

15.4.2 DMA channel priority

The DMA channel priority are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3 > DMA channel 4 > DMA channel 5

Caution: If two or more channels are started with the same factor, a DMA channel with a lower priority may be acknowledged earlier than a DMA channel with a higher priority.

15.4.3 DMA transfer start factors

There are two types of DMA transfer start factors:

(1) Request from hardware

If the DMA is enabled (EN=1) an interrupt request is issued from the on-chip peripheral I/O selected in the DTFRn register, the DMA transfer starts.

(2) Request from software

If the DMA is enabled (EN=1), the software can start the DMA transfer by setting STG=1.

15.4.4 DMA transfer end

At the end of a DMA transfer, the EN bit is cleared to 0 by the hardware and a DMA interrupt (INTDMAn) is generated when TCS is set as 1.

The interrupt generation for TCS=0 and TCS=1 is as follows:

(1) TCS = 0:

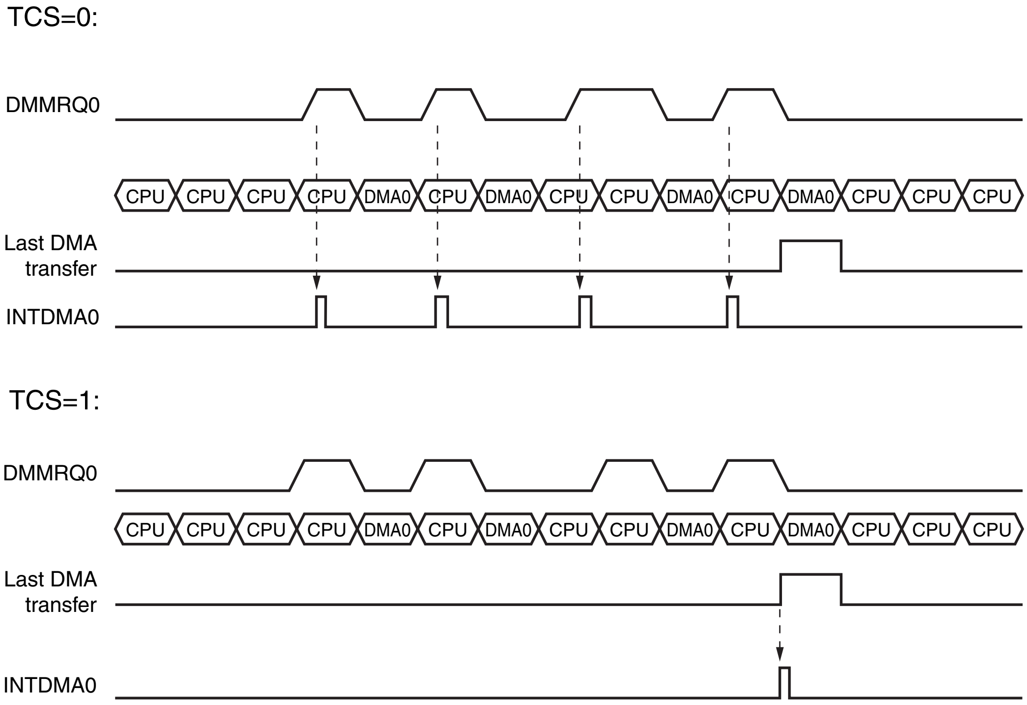
When TCS is set as 0, the DMA interrupt is a direct mirror the DMA trigger signal (e.g. the on-chip peripheral I/O interrupt).

(2) TCS = 1:

When TCS bit is set as 1, the DMA interrupt is generated when the last DMA transfer request is executed.

Figure 15-12 illustrates the timing of INTDMAn for the two TCS settings. As mentioned before, the mirror function is not required for V850E/RS1, therefore it is highly recommended NOT to use a setting of TCS=0.

Figure 15-12: TCS Bit and INTDMAn Generation


TCS=0:
DMMRQ0
CPU CPU CPU CPU DMA0 CPU DMA0 CPU CPU DMA0 CPU DMA0 CPU CPU CPU
Last DMA transfer
INTDMA0
TCS=1:
DMMRQ0
CPU CPU CPU CPU DMA0 CPU DMA0 CPU CPU DMA0 CPU DMA0 CPU CPU CPU
Last DMA transfer
INTDMA0

15.5 Transfer Modes

In the following examples it is assumed that TCS is set as “1” for all DMA channels.

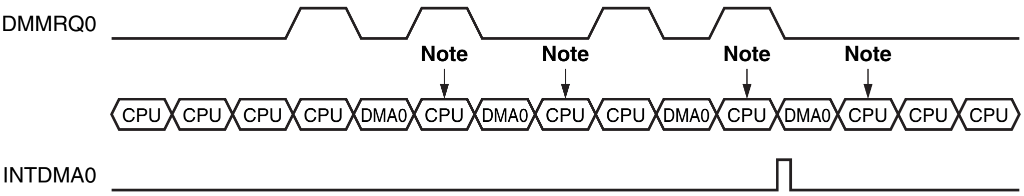
15.5.1 Single transfer mode

In Single transfer mode, the DMA releases the bus after each transfer. If there is a subsequent DMA transfer request, the transfer is performed again when the bus becomes available again. This operation continues until the transfer counter is cleared to 0 and the internal TC signal is generated.
When the DMA has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

A Single transfer mode example is shown in Figures 15-13 and 15-14.

Figure 15-13 shows the transfers of DMA0 with DMBC0 set to 3 at the beginning of the transfers:

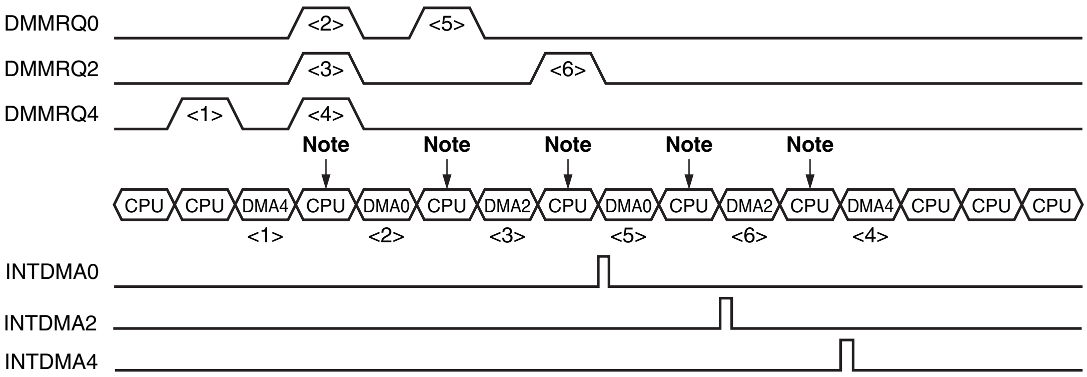
Figure 15-13: Single Transfer Mode Example (1 Channel)



Note: The bus is always released.

Figure 15-14 shows the Single Transfer mode with DMA requests on multiple DMA channels. The number of transfers is set to 2 for all channels (DMBCn=1) at the beginning.

Figure 15-14: Single Transfer Mode Example (3 Channels)



Note: The bus is always released.

After each bus release, the DMA controller checks which pending DMA request has the highest priority, delaying the execution of the lower-prioritized DMA transfer accordingly.

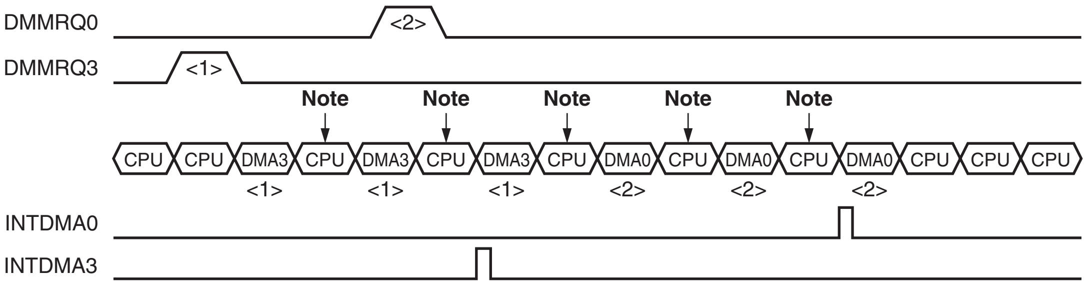
15.5.2 Fixed channel transfer mode

In “Fixed Channel” transfer mode, the DMA releases the bus after each transfer, but continues to repeat the DMA transfer until the transfer counter is cleared to 0 without requiring a new DMA transfer request. When the DMA has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request does not take precedence.

A “Fixed Channel” transfer mode example is shown in Figure 15-15 and 15-16.

In Figure 15-15, DMA channel 0 and 3 are set to “Fixed Channel” transfer mode, with DMBC0 and DMBC3 set to 2 at the beginning (transfer counter 3).

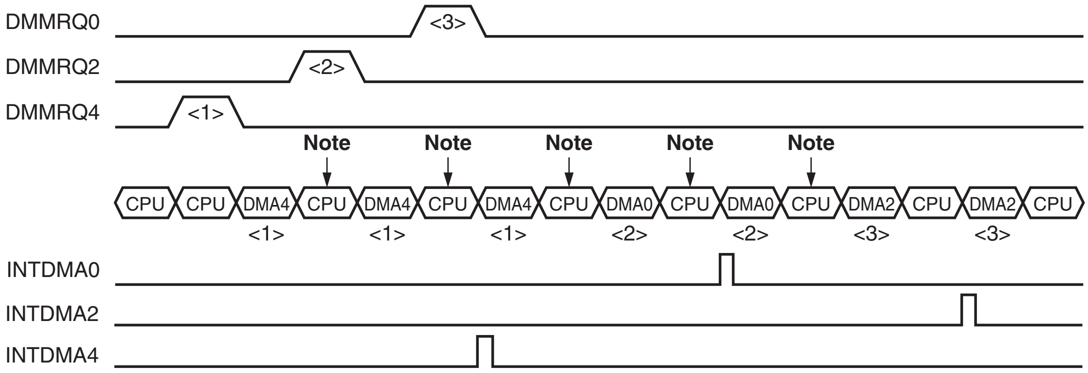
Figure 15-15: Fixed Channel Transfer Example (2 Channels)



Note: The bus is always released.

Figure 15-16 shows also “Fixed Channel” transfer, but with 3 DMA transfers active. DMBC4 is set to 2, while DMBC2 and DMBC0 are set as 1 (2 transfers):

Figure 15-16: Fixed Channel Transfer Mode Example (3 Channels)



Note: The bus is always released.

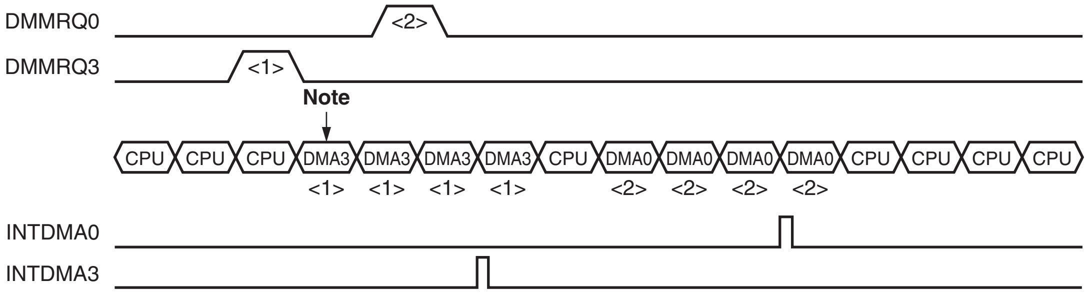
The DMA controller executes and completes the DMA first transfer (channel 4) before it checks which of the remaining pending DMA transfer requests has the highest priority. It executes and completes that DMA transfer (channel 0) before the DMA transfer with the lowest priority is handled.

15.5.3 Block transfer mode

In Block transfer mode, once transfer starts, DMA continues the transfer operation without releasing the bus until the set number of transfers is done and the TC signal is set. No other DMA request is acknowledged during that Block transfer.
After the Block transfer ends, the DMAC releases the bus before another DMA transfer can be acknowledged.

Figure 15-17 is an example of a Block transfer when two DMA forwarding demands of DMMRQ0 and DMMRQ3 are generated. The transfer counter for both channels is set to 4 transfers.

Figure 15-17: Block Transfer Example (2 Channels)


DMMRQ0
<2>
DMMRQ3
<1>
Note
CPU CPU CPU DMA3 DMA3 DMA3 DMA3 CPU DMA0 DMA0 DMA0 DMA0 CPU CPU CPU CPU
<1> <1> <1> <1> <2> <2> <2> <2>
INTDMA0
INTDMA3

Note: The bus is not released

15.5.4 Summary on the transfer modes

Table 15-3: Comparison of DMA Transfer Modes

Item	Single transfer mode	Fixed Channel transfer mode	Block transfer mode
Number of elements transferred by each DMA trigger	1	DMBCn elements	DMBCn elements
Number of triggers required to transfer DMBCn elements	DMBCn triggers	1	1
Bus released after the transfer of a single element	yes	yes	no
A higher prioritized DMA request can preempt an ongoing DMA transfer of lower priority	yes	no	no

Chapter 16 FCAN Controller

16.1 Overview

This product features an on-chip 2-channel CAN (Controller Area Network) controller that complies with the CAN protocol as standardized in ISO 11898. The number of channels varies depending on the product as shown below.

16.1.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (CAN clock input ≥ 8 MHz)
- 32 message buffers × 2 channels
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

16.1.2 Overview of Functions

Table 16-1 presents an overview of the CAN controller functions.

Table 16-1: Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input Š 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	• 32 message buffers ¥ 2 channels • Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	• Unique ID can be set to each message buffer. • Mask setting of four patterns is possible for each channel. • A receive completion interrupt is generated each time a message is received and stored in a message buffer. • Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). • Receive history list function
Message transmission	• Unique ID can be set to each message buffer. • Transmit completion interrupt for each message buffer • Message buffer numbers 0 to 7 specified as transmit message buffers can be used for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). • Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	• The time stamp function can be set for a receive message when a 16-bit timer is used in combination. SOF or EOF in a CAN message frame can be detected by using a trigger that selects a time stamp capture.
Diagnostic function	• Readable error counters • “Valid protocol operation flag” for verification of bus connections • Receive-only mode • Single-shot mode • CAN protocol error type decoding • Self-test mode
Forced release from bus-off state	• Default mode can be set while bus is off, so that bus can be forcibly released from bus-off state.
Power save mode	• CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

16.1.3 Configuration

The CAN controller is composed of the following four blocks.

(1) NPB interface

This functional block provides an NPB (NEC Peripheral I/O Bus) interface and means of transmitting and
receiving signals between the CAN module and the host CPU.

(2) MAC (Memory Access Controller)

This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.

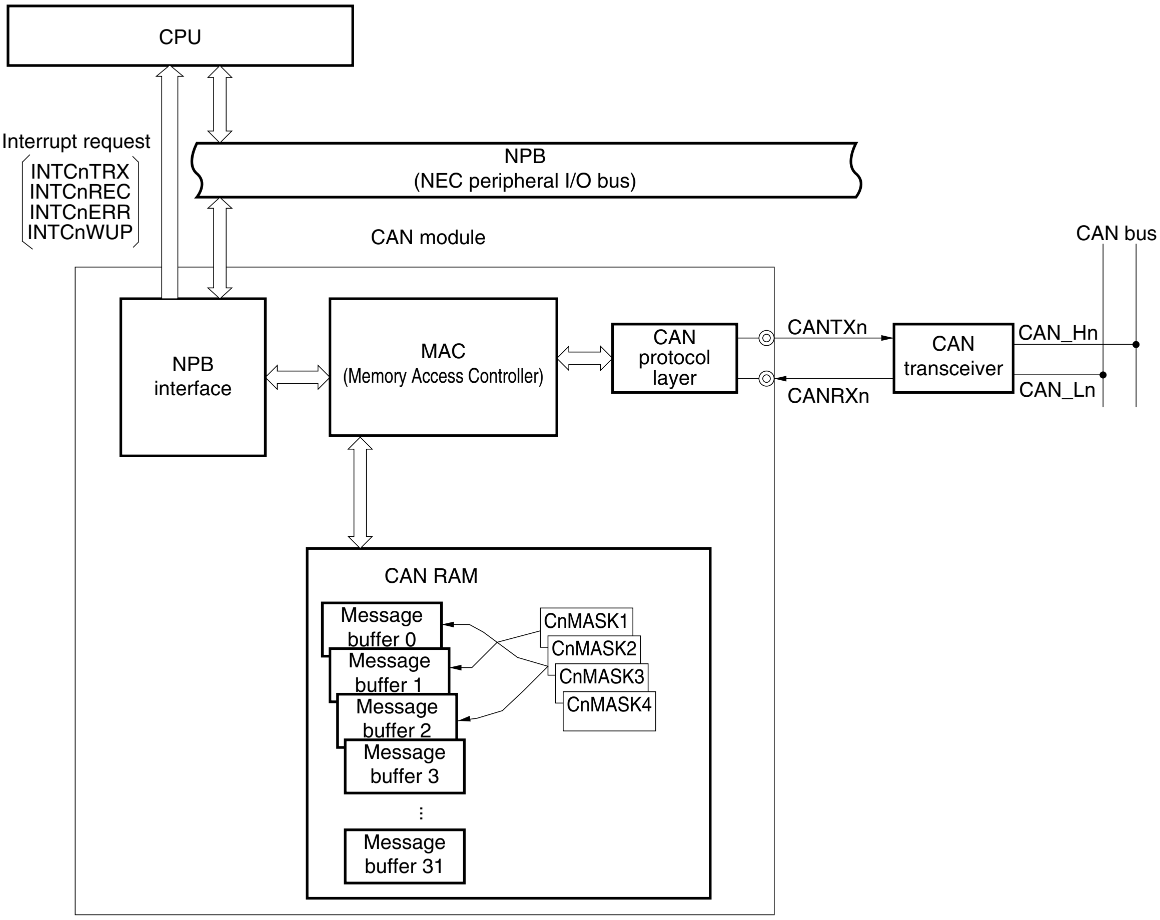
(3) CAN protocol layer

This functional block is involved in the operation of the CAN protocol and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 16-1: Block Diagram of CAN Module



Remark: n = 0, 1

16.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.
The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

Figure 16-2: Composition of Layers

Higher	Data link layer[Note]	• Logical link control (LLC)	• Acceptance filtering • Overload report • Recovery management
		• Medium access control (MAC)	• Data capsuled/not capsuled • Frame coding (stuffing/no stuffing) • Medium access management • Error detection • Error report • Acknowledgement • Seriated/not seriated
Lower	Physical layer		Prescription of signal level and bit description

Note: CAN controller specification

16.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2,048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers, which increases the number of messages that can be handled to 2,048 ¥ 218 messages.
- An extended format frame is set when “recessive level” (CMOS level of “1”) is set for both the SRR and IDE bits in the arbitration field.

16.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 16-2: Frame Types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

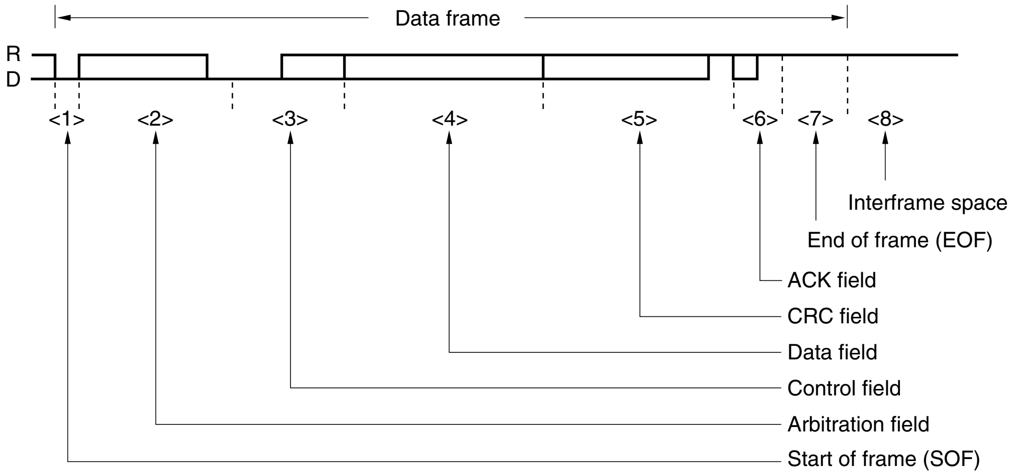
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

16.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 16-3: Data Frame

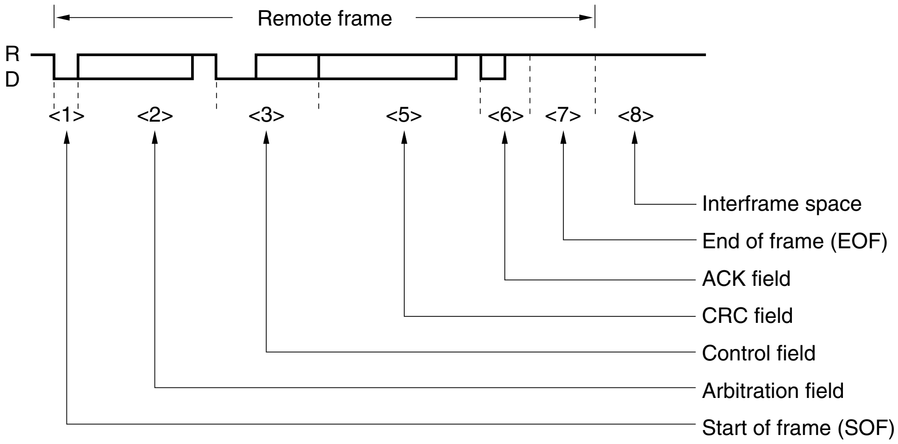


Remark: D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 16-4: Remote Frame



Remarks: 1. The data field is not transferred even if the control field's data length code is not "0000B".

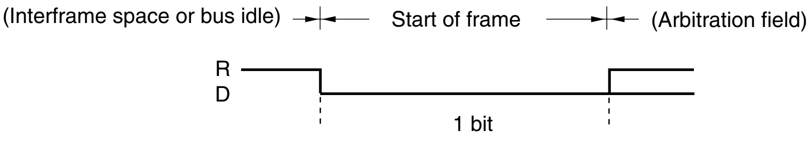
2. D: Dominant = 0
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 16-5: Start of Frame (SOF)



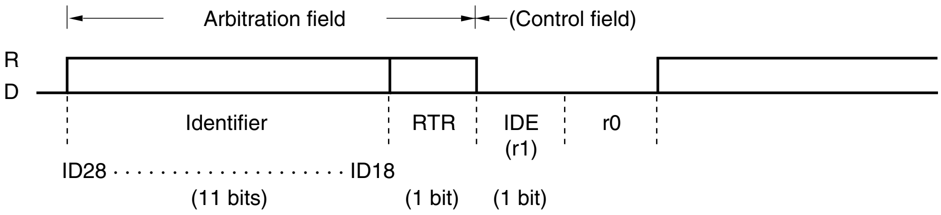
Remark: D: Dominant = 0
R: Recessive = 1

- If a dominant level is detected in the bus idle state, a hardware synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If a dominant level is sampled at the sample point following such a hardware synchronization, the bit is assigned to be a SOF. If a recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a noise only. In this case an error frame is not generated.

<2> Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 16-6: Arbitration Field (in Standard Format Mode)

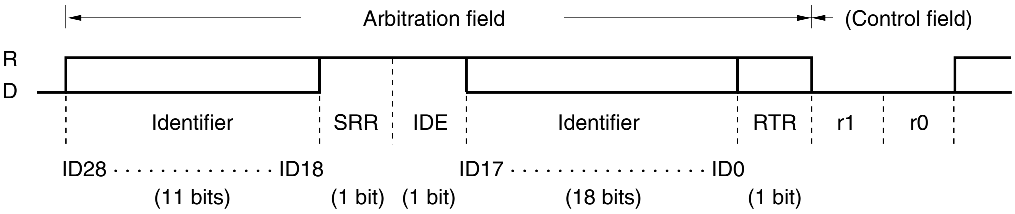


Cautions: 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark: D: Dominant = 0
R: Recessive = 1

Figure 16-7: Arbitration Field (in Extended Format Mode)



Cautions: 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark: D: Dominant = 0
R: Recessive = 1

Table 16-3: RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

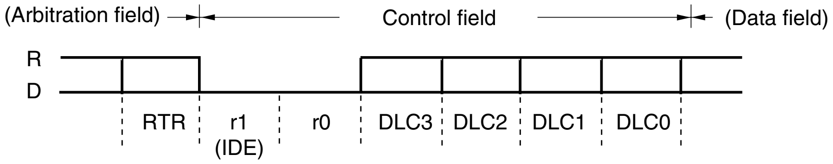
Table 16-4: Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	Number of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “N” as the number of data bytes in the data field (N = 0 to 8).

Figure 16-8: Control Field



Remark: D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 16-5: Data Length Setting

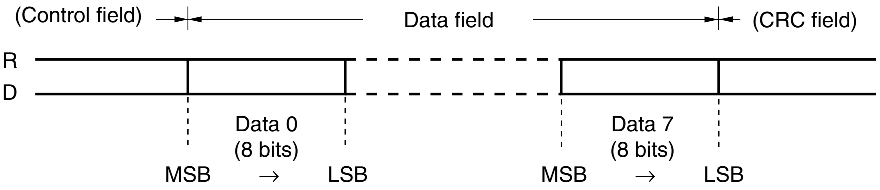
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution: In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

Figure 16-9: Data Field

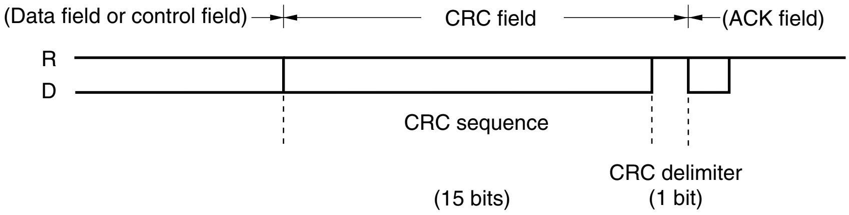


Remark: D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 16-10: CRC Field



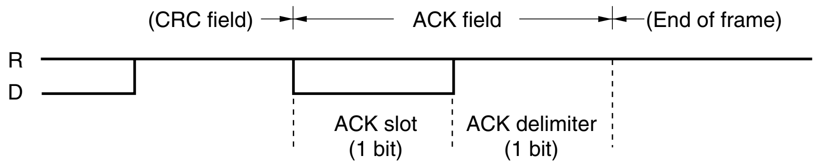
Remark: D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.
  $P(X) = X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$
- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

<6> ACK field

The ACK field is used to acknowledge normal reception.

Figure 16-11: ACK Field



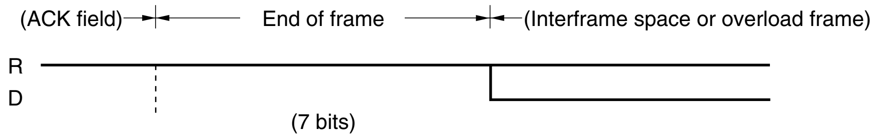
Remark: D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 16-12: End of Frame (EOF)



Remark: D: Dominant = 0
R: Recessive = 1

<8> Interframe space

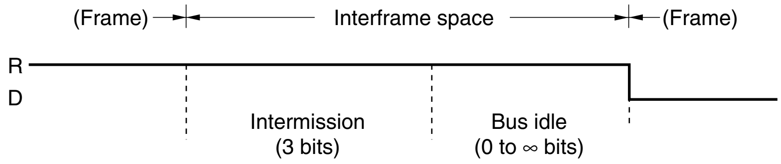
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

(a) Error active node

The interframe space consists of a 3-bit intermission field and a bus idle field.

Figure 16-13: Interframe Space (Error Active Node)



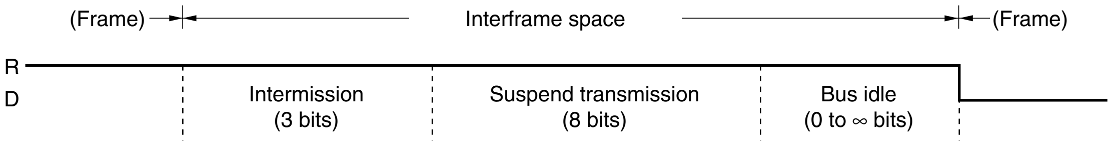
Remarks: **1.** Bus idle: State in which the bus is not used by any node.

2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

Figure 16-14: Interframe Space (Error Passive Node)



Remarks: 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.

2. D: Dominant = 0
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

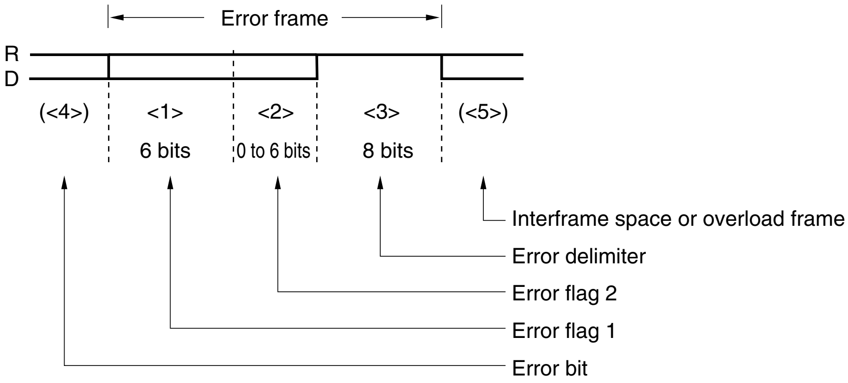
Table 16-6: Operation in Error Status

Error Status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

16.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 16-15: Error Frame



Remark: D: Dominant = 0
R: Recessive = 1

Table 16-7: Definition of Error Frame Fields

No.	Name	Bit Count	Definition
<1>	Error flag 1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag 2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/ overload frame	–	An interframe space or overload frame starts from here.

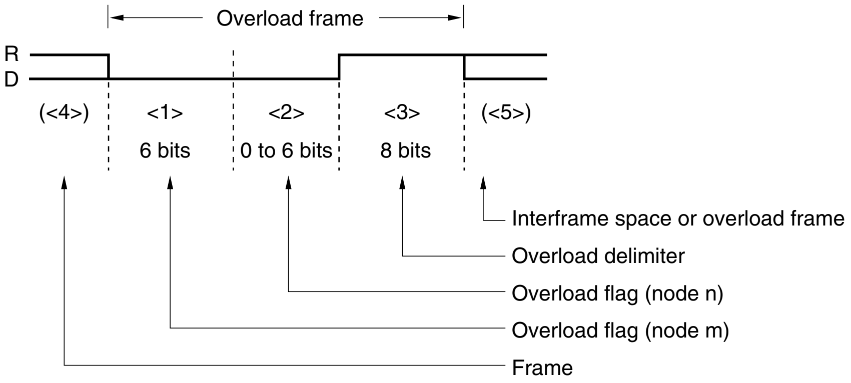
16.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

Note: In this CAN controller, all reception frames can be loaded without outputting an overload frame because of the high-speed internal processing.

Figure 16-16: Overload Frame



Remark: D: Dominant = 0
R: Recessive = 1
Node n ¼ node m

Table 16-8: Definition of Overload Frame Fields

No	Name	Bit Count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/ overload frame	–	An interframe space or overload frame starts from here.

16.3 Functions

16.3.1 Determining bus priority

(1) When a node starts transmission:

During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

The node that consecutively outputs the dominant level for the longest from the first bit of the arbitration field has the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
The transmitting node compares its output arbitration field and the data level on the bus.

Table 16-9: Determining Bus Priority

Level match	Continuous transmission
Level mismatch	Continuous transmission

(3) Priority of data frame and remote frame

When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Caution: If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frame takes priority.

16.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1 bit of inverted-level data if the same level continues for 5 bits, in order to prevent a burst error.

Table 16-10: Bit Stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

16.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

16.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

16.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN controller in waiting mode to achieve low power consumption.
The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

16.3.6 Error control function

(1) Error types

Table 16-11: Error Types

Type	Description of Error		Detection State	
	Detection Method	Detection Condition	Transmission/ Reception	Field/Frame
Bit error	Comparison of the output level and level on the bus (except stuff bit)	Mismatch of levels	Transmitting/ receiving node	Bit that is outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check of the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 16-12: Output Timing of Error Frame

Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CEC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame. (However, it does not re-transmit the frame in the single-shot mode.)

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification.

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) of the CAN error counter register as shown in Table 16-13.
The present error state is indicated by the CAN module information register (CnINFO).
When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the CnINFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the CnINFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the CnINFO register is set to 1.
- If only one node is active on the bus at start-up (i.e., when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Remark: n = 0, 1

Table 16-13: Types of Error States

Type	Operation	Value of Error Counter	Indication of CnINFO Register	Operation Specific to Error State
Error active	Transmission	0 to 95	TECS1, TECS0 = 00	• Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0 to 95	RECS1, RECS0 = 00	
	Transmission	96 to 127	TECS1, TECS0 = 01	
	Reception	96 to 127	RECS1, RECS0 = 01	
Error passive	Transmission	128 to 255	TECS1, TECS0 = 11	• Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. • Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission).
	Reception	128 or more	RECS1, RECS0 = 11	
Bus-off	Transmission	256 or more (not indicated)[Note]	BOFF = 1, TECS1, TECS0 = 11	• Communication is not possible. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. • If the initialization mode is set and then 11 recessive-level bits are generated 128 times in a row in an operation mode other than the initialization mode, the error counter is reset to 0 and the error active state can be restored.

Note: The value of the transmit error counter (TEC) does not carry any meaning if BOFF has been set. If an error that increments the value of the transmission error counter by 8 while the counter value is in a range of 248 to 255 occurs, the counter is not incremented and the bus-off state is assumed.

Remark: n = 0, 1

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated during the first bit of the error delimiter.

Table 16-14: Error Counter

State	Transmission Error Counter (TEC7 to TEC0)	Reception Error Counter (REC6 to REC0)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (REPS bit = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (REPS bit = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1>ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2>A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (transmitting)	+8 (receiving, REPS bit = 0)
When the transmitting node has completed transmission without error (±0 if error counter = 0)	–1	No change
When the receiving node has completed reception without error	No change	–1 (1 ≤ REC6 to REC0 ≤ 127, REPS bit = 0) ±0 (REC6 to REC0 = 0, REPS bit = 0) Any value of 119 to 127 is set (REPS bit = 1)

(c) Occurrence of bit error in intermission

An overload frame is generated.

Caution: If an error occurs, it is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

(5) Recovery from bus-off state

When the CAN module is in the bus-off state, the transmission pins (CTXDn) cut off from the CAN bus always
output the recessive level.
The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

<1> Request to enter the CAN initialization mode

<2> Request to enter a CAN operation mode

(a) Recovery operation through normal recovery sequence
(b) Forced recovery operation that skips recovery sequence

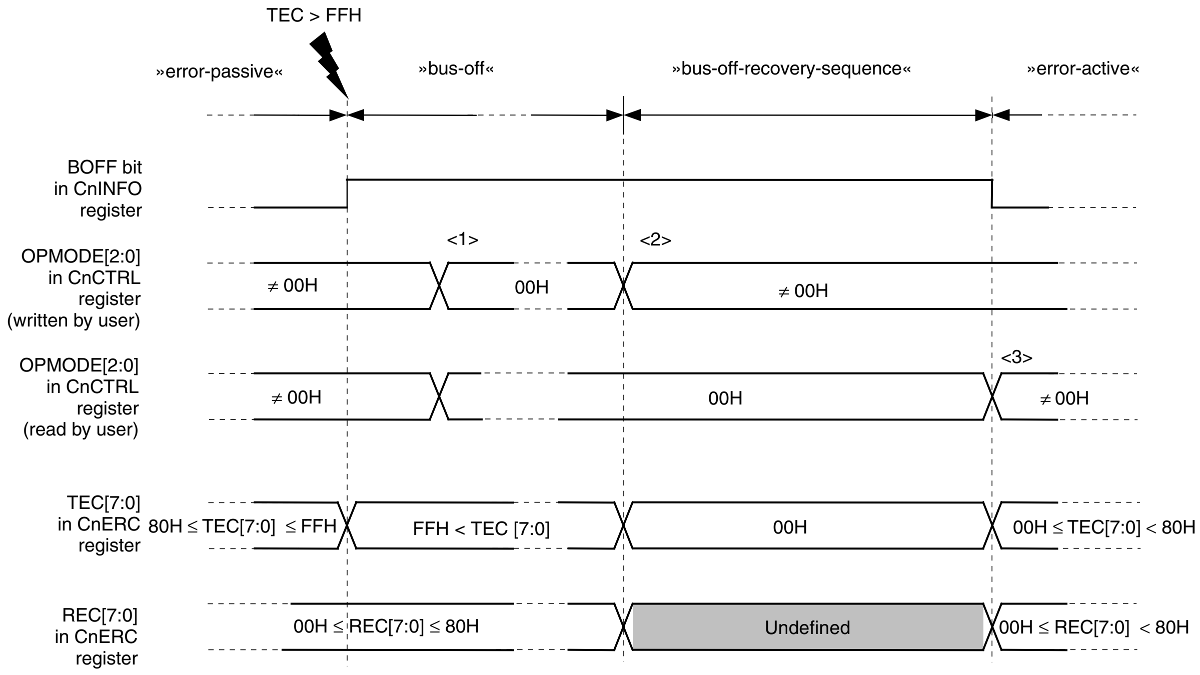
(a) Recovery from bus-off state through normal recovery sequence

The CAN module first issues a request to enter the initialization mode (refer to timing <1> in Figure 16-17). This request will be immediately acknowledged, and the OPMODE bits of the CnCTRL register are cleared to 000B. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.
Next, the module requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in Figure 16-17). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times or more. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (refer to timing <3> in Figure 16-17), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Whether the CAN module has entered the operation mode can be confirmed by reading the OPMODE bits of the CnCTRL register.
During the bus-off period and bus-off recovery sequence, the BOFF bit of the CnINFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

Caution: In the bus-off recovery sequence, the REC0 to REC6 bits counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To be released from the bus-off state, the module must enter the initialization mode once. If the module is in the CAN sleep mode or CAN stop mode, however, it cannot directly enter the initialization mode. In this case, the bus off recovery sequence is started at the same time as the CAN sleep mode is released even without shifting to the initialization mode. In addition to clearing the CnCTRL.PSMODE1 and CnCTRL.PSMODE0 bits by software, the bus off recovery sequence is also started due to wake-up by dominant edge detection on the CAN bus.

Remark: n = 0, 1

Figure 16-17: Recovery from Bus-off State Through Normal Recovery Sequence


TEC > FFH
»error-passive«
»bus-off«
»bus-off-recovery-sequence«
»error-active«
BOFF bit in CnINFO register
OPMODE[2:0] in CnCTRL register (written by user)
<1>
<2>
≠ 00H
00H
≠ 00H
OPMODE[2:0] in CnCTRL register (read by user)
<3>
≠ 00H
00H
≠ 00H
TEC[7:0] in CnERC register
80H ≤ TEC[7:0] ≤ FFH
FFH < TEC [7:0]
00H
00H ≤ TEC[7:0] < 80H
REC[7:0] in CnERC register
00H ≤ REC[7:0] ≤ 80H
Undefined
00H ≤ REC[7:0] < 80H

Remark: n = 0, 1

(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.
First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, refer to **16.3.6 (5) (a)"Recovery from bus-off state through normal recovery sequence" on page 544**.
Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the CnCTRL register must be set to 1.
As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in **Figure 16-74, "Setting CAN Sleep Mode/Stop Mode," on page 677**.

Caution: This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

Remark: n = 0, 1

(c) Initializing CAN module error counter register (CnERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (CnERC) and CAN module information register (CnINFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the CnCTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Cautions: 1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the CnERC and CnINFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

Remark: n = 0, 1

16.3.7 Baud rate control function

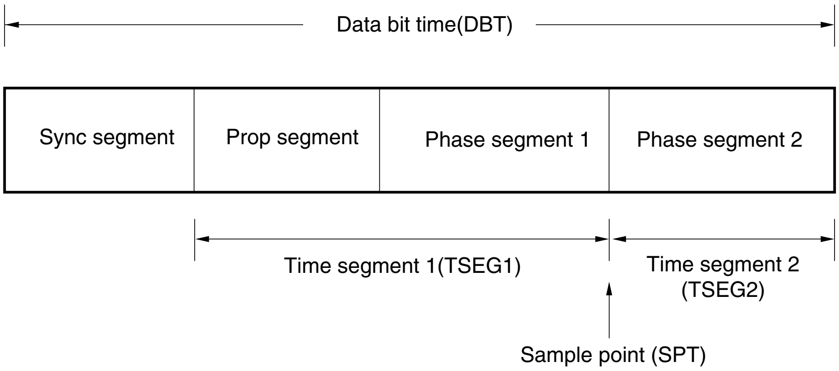
(1) Prescaler

The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer base clock (f_{TQ}) that is the CAN module system clock (f_{CANMOD}) divided by 1 to 256 (refer to **16.6 (12)"CAN module bit rate prescaler register (CnBRP)" on page 606**).

(2) Data bit time (8 to 25 time quanta)

One data bit time is defined as Figure 16-18. The CAN controller sets time segment 1, time segment 2, and resynchronization Jump Width (SJW) as the data bit time, as shown in Figure 16-18. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

Figure 16-18: Segment Setting

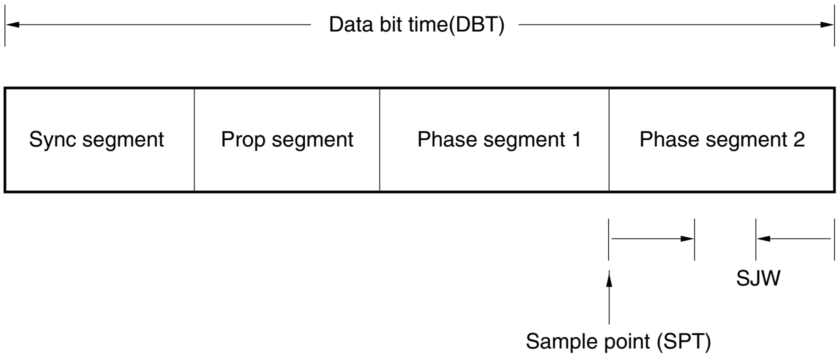


Segment name	Settable range	Notes on setting to conform to CAN specification
Time segment 1 (TSEG1)	2TQ to 16TQ	—
Time segment 2 (TSEG2)	1TQ to 8TQ	IPT[Note] of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
resynchronization Jump Width (SJW)	1TQ to 4TQ	Time segment1-1 TQ length to 4TQ, whichever is smaller

Note: IPT: Information Processing Time

Reference: The CAN protocol specification defines the segments constituting the data bit time as shown in Figure 16-19.

Figure 16-19: Reference: Configuration of Data Bit Time Defined by CAN Specification



Segment name	Segment length	Description
Sync segment (Synchronization segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hardware synchronization is established.
Prop segment (Propagation segment)	Programmable to 1 to 8 or more great	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 × (Delay of CAN bus) + (Delay of input buffer)
Phase segment 1 (Phase buffer segment 1)	Programmable to 1 to 8	
Phase segment 2 (Phase buffer segment 2)	Phase segment 1 or IPT-[Note], whichever greater	This segment compensates for an error in the data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
SJW (resynchronization Jump Width)	Programmable from 1 TQ to segment 1TQ to 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during resynchronization.

Note: IPT: Information Processing Time

(3) Synchronizing data bit

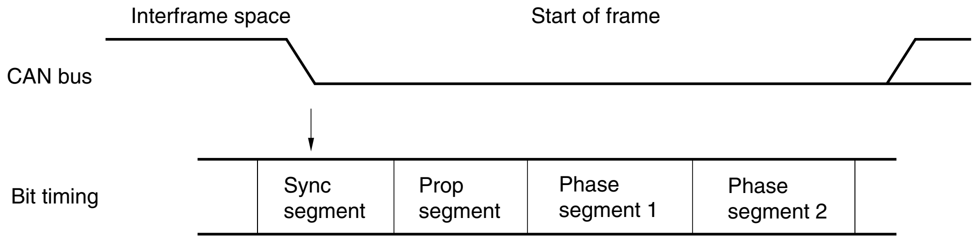
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(a) Hardware synchronization

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.

Figure 16-20: Adjusting Synchronization of Data Bit

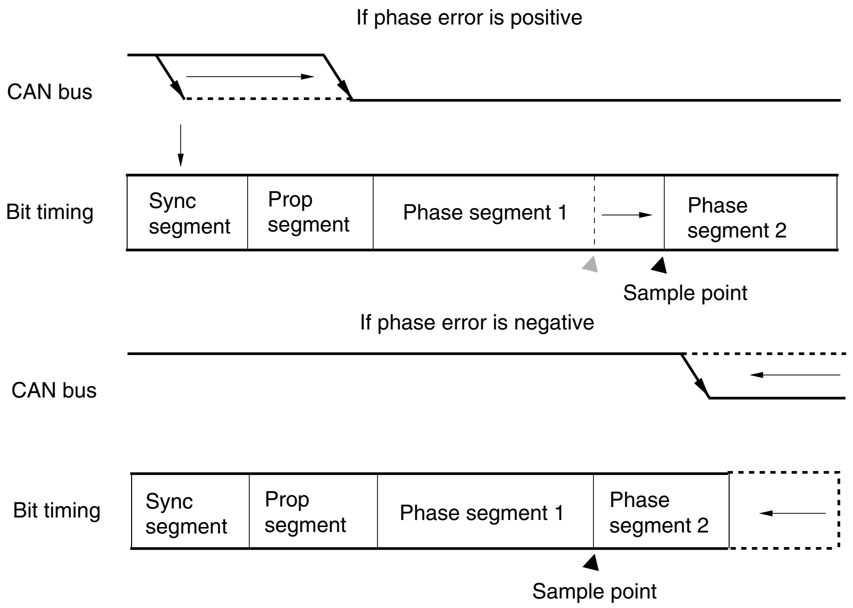


(b) Resynchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.
  <Sign of phase error>
  0: If the edge is within the sync segment
  Positive: If the edge is before the sample point (phase error)
  Negative: If the edge is after the sample point (phase error)
  If phase error is positive: Phase segment 1 is longer by specified SJW.
  If phase error is negative: Phase segment 2 is shorter by specified SJW.
- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in the baud rate between the transmitting node and receiving node.

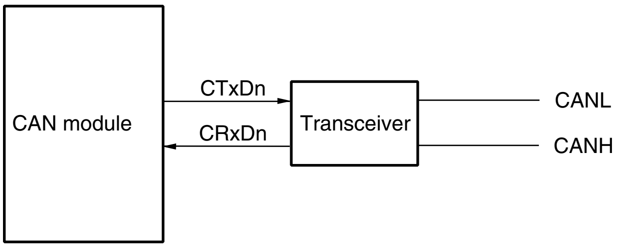
Figure 16-21: Resynchronization



16.4 Connection with Target System

The CAN module has to be connected to the CAN bus using an external transceiver.

Figure 16-22: Connection to CAN Bus



Remark: n = 0, 1

16.5 Internal Registers of CAN controller

16.5.1 CAN controller configuration

Table 16-15: List of CAN Controller Registers

Item	Register Name
CAN global registers	CAN global control register (CnGMCTRL)
	CAN global clock selection register (CnGMCS)
	CAN global automatic block transmission control register (CnGMABT)
	CAN global automatic block transmission delay setting register (CnGMABTD)
CAN module registers	CAN module mask 1 register (CnMASK1L, CnMASK1H)
	CAN module mask 2 register (CnMASK2L, CnMASK2H)
	CAN module mask3 register (CnMASK3L, CnMASK3H)
	CAN module mask 4 registers (CnMASK4L, CnMASK4H)
	CAN module control register (CnCTRL)
	CAN module last error information register (CnLEC)
	CAN module information register (CnINFO)
	CAN module error counter register (CnERC)
	CAN module interrupt enable register (CnIE)
	CAN module interrupt status register (CnINTS)
	CAN module bit rate prescaler register (CnBRP)
	CAN module bit rate register (CnBTR)
	CAN module last in-pointer register (CnLIPT)
	CAN module receive history list register (CnRGPT)
	CAN module last out-pointer register (CnLOPT)
	CAN module transmit history list register (CnTGPT)
	CAN module time stamp register (CnTS)
Message buffer registers	CAN message data byte 01 register m (CnMDATA01m)
	CAN message data byte 0 register m (CnMDATA0m)
	CAN message data byte 1 register m (CnMDATA1m)
	CAN message data byte 23 register m (CnMDATA23m)
	CAN message data byte 2 register m (CnMDATA2m)
	CAN message data byte 3 register m (CnMDATA3m)
	CAN message data byte 45 register m (CnMDATA45m)
	CAN message data byte 4 register m (CnMDATA4m)
	CAN message data byte 5 register m (CnMDATA5m)
	CAN message data byte 67 register m (CnMDATA67m)
	CAN message data byte 6 register m (CnMDATA6m)
	CAN message data byte 7 register m (CnMDATA7m)
	CAN message data length register m (CnMDLCm)
	CAN message configuration register m (CnMCONFm)
	CAN message ID register m (CnMIDLm, CnMIDHm)
	CAN message control register m (CnMCTRLm)

Remark: n = 0, 1
m = 0 to 31

16.5.2 Register access type

The peripheral I/O register for the CAN controller is assigned to 03FEC000H to 03FEEFFFH. For details, refer to **3.5.1 "Programmable peripheral I/O control register (BPC)" on page 92**.

Table 16-16: Register Access Type (1/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC000H	CAN0 global control register	C0GMCTRL	R/W			√	0000H
03FEC002H	CAN0 global clock select register	C0GMCS	R/W		√		0FH
03FEC006H	CAN0 global automatic block transmission control register	C0GMABT	R/W			√	0000H
03FEC008H	CAN0 global automatic block transmission delay setting register	C0GMABTD	R/W		√		00H
03FEC040H	CAN0 module mask 1 register	C0MASK1L	R/W		–	√	Undefined
03FEC042H		C0MASK1H					
03FEC044H	CAN0 module mask 2 register	C0MASK2L	R/W	–	–	√	Undefined
03FEC046H		C0MASK2H					
03FEC048H	CAN0 module mask 3 register	C0MASK3L	R/W	–	–	√	Undefined
03FEC04AH		C0MASK3H					
03FEC04CH	CAN0 module mask 4 register	C0MASK4L	R/W	–	–	√	Undefined
03FEC04EH		C0MASK4H					
03FEC050H	CAN0 module control register	C0CTRL	R/W			√	0000H
03FEC052H	CAN0 module last error information register	C0LEC	R/W		√		00H
03FEC053H	CAN0 module information register	C0INFO	R		√		00H
03FEC054H	CAN0 module error counter register	C0ERC	R			√	0000H
03FEC056H	CAN0 module interrupt enable register	C0IE	R/W			√	0000H
03FEC058H	CAN0 module interrupt status register	C0INTS	R/W			√	0000H
03FEC05AH	CAN0 module bit rate prescaler register	C0BRP	R/W		√		FFH
03FEC05CH	CAN0 module bit rate register	C0BTR	R/W			√	370FH
03FEC05EH	CAN0 module last in-pointer register	C0LIPT	R		√	–	Undefined
03FEC060H	CAN0 module receive history list register	C0RGPT	R/W			√	xx02H
03FEC062H	CAN0 module last out-pointer register	C0LOPT	R		√		Undefined
03FEC064H	CAN0 module transmit history list register	C0TGPT	R/W			√	xx02H
03FEC066H	CAN0 module time stamp register	C0TS	R/W			√	0000H
03FEC100H	CAN0 message data byte 01 register 00	C0MDATA0100	R/W			√	Undefined
03FEC100H	CAN0 message data byte 0 register 00	C0MDATA000	R/W		√		Undefined
03FEC101H	CAN0 message data byte 1 register 00	C0MDATA100	R/W		√		Undefined
03FEC102H	CAN0 message data byte 23 register 00	C0MDATA2300	R/W			√	Undefined
03FEC102H	CAN0 message data byte 2 register 00	C0MDATA200	R/W		√		Undefined
03FEC103H	CAN0 message data byte 3 register 00	C0MDATA300	R/W		√		Undefined

Table 16-16: Register Access Type (2/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC104H	CAN0 message data byte 45 register 00	C0MDATA4500	R/W			√	Undefined
03FEC104H	CAN0 message data byte 4 register 00	C0MDATA400	R/W		√		Undefined
03FEC105H	CAN0 message data byte 5 register 00	C0MDATA500	R/W		√		Undefined
03FEC106H	CAN0 message data byte 67 register 00	C0MDATA6700	R/W			√	undefined
03FEC106H	CAN0 message data byte 6 register 00	C0MDATA600	R/W		√		undefined
03FEC107H	CAN0 message data byte 7 register 00	C0MDATA700	R/W		√		undefined
03FEC108H	CAN0 message data length code register 00	C0MDLC00	R/W		√		0000xxxxB
03FEC109H	CAN0 message configuration register 00	C0MCONF00	R/W		√		undefined
03FEC10AH	CAN0 message ID register 00	C0MIDL00	R/W			√	undefined
03FEC10CH		C0MIDH00	R/W			√	undefined
03FEC10EH	CAN0 message control register 00	C0MCTRL00	R/W			√	00x00000 000xx000B
03FEC120H	CAN0 message data byte 01 register 01	C0MDATA0101	R/W			√	undefined
03FEC120H	CAN0 message data byte 0 register 01	C0MDATA001	R/W		√		undefined
03FEC121H	CAN0 message data byte 1 register 01	C0MDATA101	R/W		√		undefined
03FEC122H	CAN0 message data byte 23 register 01	C0MDATA2301	R/W			√	undefined
03FEC122H	CAN0 message data byte 2 register 01	C0MDATA201	R/W		√		undefined
03FEC123H	CAN0 message data byte 3 register 01	C0MDATA301	R/W		√		undefined
03FEC124H	CAN0 message data byte 45 register 01	C0MDATA4501	R/W			√	undefined
03FEC124H	CAN0 message data byte 4 register 01	C0MDATA401	R/W		√		undefined
03FEC125H	CAN0 message data byte 5 register 01	C0MDATA501	R/W		√		undefined
03FEC126H	CAN0 message data byte 67 register 01	C0MDATA6701	R/W			√	undefined
03FEC126H	CAN0 message data byte 6 register 01	C0MDATA601	R/W		√		undefined
03FEC127H	CAN0 message data byte 7 register 01	C0MDATA701	R/W		√		undefined
03FEC128H	CAN0 message data length code register 01	C0MDLC01	R/W		√		0000xxxxB
03FEC129H	CAN0 message configuration register 01	C0MCONF01	R/W		√		undefined
03FEC12AH	CAN0 message ID register 01	C0MIDL01	R/W			√	undefined
03FEC12CH		C0MIDH01	R/W			√	undefined
03FEC12EH	CAN0 message control register 01	C0MCTRL01	R/W			√	00x00000 000xx000B
03FEC140H	CAN0 message data byte 01 register 02	C0MDATA0102	R/W			√	undefined
03FEC140H	CAN0 message data byte 0 register 02	C0MDATA002	R/W		√		undefined
03FEC141H	CAN0 message data byte 1 register 02	C0MDATA102	R/W		√		undefined
03FEC142H	CAN0 message data byte 23 register 02	C0MDATA2302	R/W			√	undefined
03FEC142H	CAN0 message data byte 2 register 02	C0MDATA202	R/W		√		undefined
03FEC143H	CAN0 message data byte 3 register 02	C0MDATA302	R/W		√		undefined
03FEC144H	CAN0 message data byte 45 register 02	C0MDATA4502	R/W			√	undefined
03FEC144H	CAN0 message data byte 4 register 02	C0MDATA402	R/W		√		undefined
03FEC145H	CAN0 message data byte 5 register 02	C0MDATA502	R/W		√		undefined

Table 16-16: Register Access Type (3/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC146H	CAN0 message data byte 67 register 02	C0MDATA6702	R/W			√	undefined
03FEC146H	CAN0 message data byte 6 register 02	C0MDATA602	R/W		√		undefined
03FEC147H	CAN0 message data byte 7 register 02	C0MDATA702	R/W		√		undefined
03FEC148H	CAN0 message data length code register 02	C0MDLC02	R/W		√		0000xxxxB
03FEC149H	CAN0 message configuration register 02	C0MCONF02	R/W		√		undefined
03FEC14AH	CAN0 message ID register 02	C0MIDL02	R/W			√	undefined
03FEC14CH		C0MIDH02	R/W			√	undefined
03FEC14EH	CAN0 message control register 02	C0MCTRL02	R/W			√	00x00000 000xx000B
03FEC160H	CAN0 message data byte 01 register 03	C0MDATA0103	R/W			√	undefined
03FEC160H	CAN0 message data byte 0 register 03	C0MDATA003	R/W		√		undefined
03FEC161H	CAN0 message data byte 1 register 03	C0MDATA103	R/W		√		undefined
03FEC162H	CAN0 message data byte 23 register 03	C0MDATA2303	R/W			√	undefined
03FEC162H	CAN0 message data byte 2 register 03	C0MDATA203	R/W		√		undefined
03FEC163H	CAN0 message data byte 3 register 03	C0MDATA303	R/W		√		undefined
03FEC164H	CAN0 message data byte 45 register 03	C0MDATA4503	R/W			√	undefined
03FEC164H	CAN0 message data byte 4 register 03	C0MDATA403	R/W		√		undefined
03FEC165H	CAN0 message data byte 5 register 03	C0MDATA503	R/W		√		undefined
03FEC166H	CAN0 message data byte 67 register 03	C0MDATA6703	R/W			√	undefined
03FEC166H	CAN0 message data byte 6 register 03	C0MDATA603	R/W		√		undefined
03FEC167H	CAN0 message data byte 7 register 03	C0MDATA703	R/W		√		undefined
03FEC168H	CAN0 message data length code register 03	C0MDLC03	R/W		√		0000xxxxB
03FEC169H	CAN0 message configuration register 03	C0MCONF03	R/W		√		undefined
03FEC16AH	CAN0 message ID register 03	C0MIDL03	R/W			√	undefined
03FEC16CH		C0MIDH03	R/W			√	undefined
03FEC16EH	CAN0 message control register 03	C0MCTRL03	R/W			√	00x00000 000xx000B
03FEC180H	CAN0 message data byte 01 register 04	C0MDATA0104	R/W			√	undefined
03FEC180H	CAN0 message data byte 0 register 04	C0MDATA004	R/W		√		undefined
03FEC181H	CAN0 message data byte 1 register 04	C0MDATA104	R/W		√		undefined
03FEC182H	CAN0 message data byte 23 register 04	C0MDATA2304	R/W			√	undefined
03FEC182H	CAN0 message data byte 2 register 04	C0MDATA204	R/W		√		undefined
03FEC183H	CAN0 message data byte 3 register 04	C0MDATA304	R/W		√		undefined
03FEC184H	CAN0 message data byte 45 register 04	C0MDATA4504	R/W			√	undefined
03FEC184H	CAN0 message data byte 4 register 04	C0MDATA404	R/W		√		undefined
03FEC185H	CAN0 message data byte 5 register 04	C0MDATA504	R/W		√		undefined
03FEC186H	CAN0 message data byte 67 register 04	C0MDATA6704	R/W			√	undefined
03FEC186H	CAN0 message data byte 6 register 04	C0MDATA604	R/W		√		undefined
03FEC187H	CAN0 message data byte 7 register 04	C0MDATA704	R/W		√		undefined
03FEC188H	CAN0 message data length code register 04	C0MDLC04	R/W		√		0000xxxxB

Table 16-16: Register Access Type (4/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC189H	CAN0 message configuration register 04	C0MCONF04	R/W		√		undefined
03FEC18AH	CAN0 message ID register 04	C0MIDL04	R/W			√	undefined
03FEC18CH		C0MIDH04	R/W			√	undefined
03FEC18EH	CAN0 message control register 04	C0MCTRL04	R/W			√	00x00000 000xx000B
03FEC1A0H	CAN0 message data byte 01 register 05	C0MDATA0105	R/W			√	undefined
03FEC1A0H	CAN0 message data byte 0 register 05	C0MDATA005	R/W		√		undefined
03FEC1A1H	CAN0 message data byte 1 register 05	C0MDATA105	R/W		√		undefined
03FEC1A2H	CAN0 message data byte 23 register 05	C0MDATA2305	R/W			√	undefined
03FEC1A2H	CAN0 message data byte 2 register 05	C0MDATA205	R/W		√		undefined
03FEC1A3H	CAN0 message data byte 3 register 05	C0MDATA305	R/W		√		undefined
03FEC1A4H	CAN0 message data byte 45 register 05	C0MDATA4505	R/W			√	undefined
03FEC1A4H	CAN0 message data byte 4 register 05	C0MDATA405	R/W		√		undefined
03FEC1A5H	CAN0 message data byte 5 register 05	C0MDATA505	R/W		√		undefined
03FEC1A6H	CAN0 message data byte 67 register 05	C0MDATA6705	R/W			√	undefined
03FEC1A6H	CAN0 message data byte 6 register 05	C0MDATA605	R/W		√		undefined
03FEC1A7H	CAN0 message data byte 7 register 05	C0MDATA705	R/W		√		undefined
03FEC1A8H	CAN0 message data length code register 05	C0MDLC05	R/W		√		0000xxxxB
03FEC1A9H	CAN0 message configuration register 05	C0MCONF05	R/W		√		undefined
03FEC1AAH	CAN0 message ID register 05	C0MIDL05	R/W			√	undefined
03FEC1ACH		C0MIDH05	R/W			√	undefined
03FEC1AEH	CAN0 message control register 05	C0MCTRL05	R/W			√	00x00000 000xx000B
03FEC1C0H	CAN0 message data byte 01 register 06	C0MDATA0106	R/W			√	undefined
03FEC1C0H	CAN0 message data byte 0 register 06	C0MDATA006	R/W		√		undefined
03FEC1C1H	CAN0 message data byte 1 register 06	C0MDATA106	R/W		√		undefined
03FEC1C2H	CAN0 message data byte 23 register 06	C0MDATA2306	R/W			√	undefined
03FEC1C2H	CAN0 message data byte 2 register 06	C0MDATA206	R/W		√		undefined
03FEC1C3H	CAN0 message data byte 3 register 06	C0MDATA306	R/W		√		undefined
03FEC1C4H	CAN0 message data byte 45 register 06	C0MDATA4506	R/W			√	undefined
03FEC1C4H	CAN0 message data byte 4 register 06	C0MDATA406	R/W		√		undefined
03FEC1C5H	CAN0 message data byte 5 register 06	C0MDATA506	R/W		√		undefined
03FEC1C6H	CAN0 message data byte 67 register 06	C0MDATA6706	R/W			√	undefined
03FEC1C6H	CAN0 message data byte 6 register 06	C0MDATA606	R/W		√		undefined
03FEC1C7H	CAN0 message data byte 7 register 06	C0MDATA706	R/W		√		undefined
03FEC1C8H	CAN0 message data length code register 06	C0MDLC06	R/W		√		0000xxxxB
03FEC1C9H	CAN0 message configuration register 06	C0MCONF06	R/W		√		undefined
03FEC1CAH	CAN0 message ID register 06	C0MIDL06	R/W			√	undefined
03FEC1CCH		C0MIDH06	R/W			√	undefined
03FEC1CEH	CAN0 message control register 06	C0MCTRL06	R/W			√	00x00000 000xx000B

Table 16-16: Register Access Type (5/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC1E0H	CAN0 message data byte 01 register 07	C0MDATA0107	R/W			√	undefined
03FEC1E0H	CAN0 message data byte 0 register 07	C0MDATA007	R/W		√		undefined
03FEC1E1H	CAN0 message data byte 1 register 07	C0MDATA107	R/W		√		undefined
03FEC1E2H	CAN0 message data byte 23 register 07	C0MDATA2307	R/W			√	undefined
03FEC1E2H	CAN0 message data byte 2 register 07	C0MDATA207	R/W		√		undefined
03FEC1E3H	CAN0 message data byte 3 register 07	C0MDATA307	R/W		√		undefined
03FEC1E4H	CAN0 message data byte 45 register 07	C0MDATA4507	R/W			√	undefined
03FEC1E4H	CAN0 message data byte 4 register 07	C0MDATA407	R/W		√		undefined
03FEC1E5H	CAN0 message data byte 5 register 07	C0MDATA507	R/W		√		undefined
03FEC1E6H	CAN0 message data byte 67 register 07	C0MDATA6707	R/W			√	undefined
03FEC1E6H	CAN0 message data byte 6 register 07	C0MDATA607	R/W		√		undefined
03FEC1E7H	CAN0 message data byte 7 register 07	C0MDATA707	R/W		√		undefined
03FEC1E8H	CAN0 message data length code register 07	C0MDLC07	R/W		√		0000xxxxB
03FEC1E9H	CAN0 message configuration register 07	C0MCONF07	R/W		√		undefined
03FEC1EAH	CAN0 message ID register 07	C0MIDL07	R/W			√	undefined
03FEC1ECH		C0MIDH07	R/W			√	undefined
03FEC1EEH	CAN0 message control register 07	C0MCTRL07	R/W			√	00x00000 000xx000B
03FEC200H	CAN0 message data byte 01 register 08	C0MDATA0108	R/W			√	undefined
03FEC200H	CAN0 message data byte 0 register 08	C0MDATA008	R/W		√		undefined
03FEC201H	CAN0 message data byte 1 register 08	C0MDATA108	R/W		√		undefined
03FEC202H	CAN0 message data byte 23 register 08	C0MDATA2308	R/W			√	undefined
03FEC202H	CAN0 message data byte 2 register 08	C0MDATA208	R/W		√		undefined
03FEC203H	CAN0 message data byte 3 register 08	C0MDATA308	R/W		√		undefined
03FEC204H	CAN0 message data byte 45 register 08	C0MDATA4508	R/W			√	undefined
03FEC204H	CAN0 message data byte 4 register 08	C0MDATA408	R/W		√		undefined
03FEC205H	CAN0 message data byte 5 register 08	C0MDATA508	R/W		√		undefined
03FEC206H	CAN0 message data byte 67 register 08	C0MDATA6708	R/W			√	undefined
03FEC206H	CAN0 message data byte 6 register 08	C0MDATA608	R/W		√		undefined
03FEC207H	CAN0 message data byte 7 register 08	C0MDATA708	R/W		√		undefined
03FEC208H	CAN0 message data length code register 08	C0MDLC08	R/W		√		0000xxxxB
03FEC209H	CAN0 message configuration register 08	C0MCONF08	R/W		√		undefined
03FEC20AH	CAN0 message ID register 08	C0MIDL08	R/W			√	undefined
03FEC20CH		C0MIDH08	R/W			√	undefined
03FEC20EH	CAN0 message control register 08	C0MCTRL08	R/W			√	00x00000 000xx000B
03FEC220H	CAN0 message data byte 01 register 09	C0MDATA0109	R/W			√	undefined
03FEC220H	CAN0 message data byte 0 register 09	C0MDATA009	R/W		√		undefined
03FEC221H	CAN0 message data byte 1 register 09	C0MDATA109	R/W		√		undefined

Table 16-16: Register Access Type (6/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC222H	CAN0 message data byte 23 register 09	C0MDATA2309	R/W			√	undefined
03FEC222H	CAN0 message data byte 2 register 09	C0MDATA209	R/W		√		undefined
03FEC223H	CAN0 message data byte 3 register 09	C0MDATA309	R/W		√		undefined
03FEC224H	CAN0 message data byte 45 register 09	C0MDATA4509	R/W			√	undefined
03FEC224H	CAN0 message data byte 4 register 09	C0MDATA409	R/W		√		undefined
03FEC225H	CAN0 message data byte 5 register 09	C0MDATA509	R/W		√		undefined
03FEC226H	CAN0 message data byte 67 register 09	C0MDATA6709	R/W			√	undefined
03FEC226H	CAN0 message data byte 6 register 09	C0MDATA609	R/W		√		undefined
03FEC227H	CAN0 message data byte 7 register 09	C0MDATA709	R/W		√		undefined
03FEC228H	CAN0 message data length code register 09	C0MDLC09	R/W		√		0000xxxxB
03FEC229H	CAN0 message configuration register 09	C0MCONF09	R/W		√		undefined
03FEC22AH	CAN0 message ID register 09	C0MIDL09	R/W			√	undefined
03FEC22CH		C0MIDH09	R/W			√	undefined
03FEC22EH	CAN0 message control register 09	C0MCTRL09	R/W			√	00x00000 000xx000B
03FEC240H	CAN0 message data byte 01 register 10	C0MDATA0110	R/W			√	undefined
03FEC240H	CAN0 message data byte 0 register 10	C0MDATA010	R/W		√		undefined
03FEC241H	CAN0 message data byte 1 register 10	C0MDATA110	R/W		√		undefined
03FEC242H	CAN0 message data byte 23 register 10	C0MDATA2310	R/W			√	undefined
03FEC242H	CAN0 message data byte 2 register 10	C0MDATA210	R/W		√		undefined
03FEC243H	CAN0 message data byte 3 register 10	C0MDATA310	R/W		√		undefined
03FEC244H	CAN0 message data byte 45 register 10	C0MDATA4510	R/W			√	undefined
03FEC244H	CAN0 message data byte 4 register 10	C0MDATA410	R/W		√		undefined
03FEC245H	CAN0 message data byte 5 register 10	C0MDATA510	R/W		√		undefined
03FEC246H	CAN0 message data byte 67 register 10	C0MDATA6710	R/W			√	undefined
03FEC246H	CAN0 message data byte 6 register 10	C0MDATA610	R/W		√		undefined
03FEC247H	CAN0 message data byte 7 register 10	C0MDATA710	R/W		√		undefined
03FEC248H	CAN0 message data length code register 10	C0MDLC10	R/W		√		0000xxxxB
03FEC249H	CAN0 message configuration register 10	C0MCONF10	R/W		√		undefined
03FEC24AH	CAN0 message ID register 10	C0MIDL10	R/W			√	undefined
03FEC24CH		C0MIDH10	R/W			√	undefined
03FEC24EH	CAN0 message control register 10	C0MCTRL10	R/W			√	00x00000 000xx000B
03FEC260H	CAN0 message data byte 01 register 11	C0MDATA0111	R/W			√	undefined
03FEC260H	CAN0 message data byte 0 register 11	C0MDATA011	R/W		√		undefined
03FEC261H	CAN0 message data byte 1 register 11	C0MDATA111	R/W		√		undefined
03FEC262H	CAN0 message data byte 23 register 11	C0MDATA2311	R/W			√	undefined
03FEC262H	CAN0 message data byte 2 register 11	C0MDATA211	R/W		√		undefined
03FEC263H	CAN0 message data byte 3 register 11	C0MDATA311	R/W		√		undefined

Table 16-16: Register Access Type (7/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC264H	CAN0 message data byte 45 register 11	C0MDATA4511	R/W			√	undefined
03FEC264H	CAN0 message data byte 4 register 11	C0MDATA411	R/W		√		undefined
03FEC265H	CAN0 message data byte 5 register 11	C0MDATA511	R/W		√		undefined
03FEC266H	CAN0 message data byte 67 register 11	C0MDATA6711	R/W			√	undefined
03FEC266H	CAN0 message data byte 6 register 11	C0MDATA611	R/W		√		undefined
03FEC267H	CAN0 message data byte 7 register 11	C0MDATA711	R/W		√		undefined
03FEC268H	CAN0 message data length code register 11	C0MDLC11	R/W		√		0000xxxxB
03FEC269H	CAN0 message configuration register 11	C0MCONF11	R/W		√		undefined
03FEC26AH	CAN0 message ID register 11	C0MIDL11	R/W			√	undefined
03FEC26CH		C0MIDH11	R/W			√	undefined
03FEC26EH	CAN0 message control register 11	C0MCTRL11	R/W			√	00x00000 000xx000B
03FEC280H	CAN0 message data byte 01 register 12	C0MDATA0112	R/W			√	undefined
03FEC280H	CAN0 message data byte 0 register 12	C0MDATA012	R/W		√		undefined
03FEC281H	CAN0 message data byte 1 register 12	C0MDATA112	R/W		√		undefined
03FEC282H	CAN0 message data byte 23 register 12	C0MDATA2312	R/W			√	undefined
03FEC282H	CAN0 message data byte 2 register 12	C0MDATA212	R/W		√		undefined
03FEC283H	CAN0 message data byte 3 register 12	C0MDATA312	R/W		√		undefined
03FEC284H	CAN0 message data byte 45 register 12	C0MDATA4512	R/W			√	undefined
03FEC284H	CAN0 message data byte 4 register 12	C0MDATA412	R/W		√		undefined
03FEC285H	CAN0 message data byte 5 register 12	C0MDATA512	R/W		√		undefined
03FEC286H	CAN0 message data byte 67 register 12	C0MDATA6712	R/W			√	undefined
03FEC286H	CAN0 message data byte 6 register 12	C0MDATA612	R/W		√		undefined
03FEC287H	CAN0 message data byte 7 register 12	C0MDATA712	R/W		√		undefined
03FEC288H	CAN0 message data length code register 12	C0MDLC12	R/W		√		0000xxxxB
03FEC289H	CAN0 message configuration register 12	C0MCONF12	R/W		√		undefined
03FEC28AH	CAN0 message ID register 12	C0MIDL12	R/W			√	undefined
03FEC28CH		C0MIDH12	R/W			√	undefined
03FEC28EH	CAN0 message control register 12	C0MCTRL12	R/W			√	00x00000 000xx000B
03FEC2A0H	CAN0 message data byte 01 register 13	C0MDATA0113	R/W			√	undefined
03FEC2A0H	CAN0 message data byte 0 register 13	C0MDATA013	R/W		√		undefined
03FEC2A1H	CAN0 message data byte 1 register 13	C0MDATA113	R/W		√		undefined
03FEC2A2H	CAN0 message data byte 23 register 13	C0MDATA2313	R/W			√	undefined
03FEC2A2H	CAN0 message data byte 2 register 13	C0MDATA213	R/W		√		undefined
03FEC2A3H	CAN0 message data byte 3 register 13	C0MDATA313	R/W		√		undefined
03FEC2A4H	CAN0 message data byte 45 register 13	C0MDATA4513	R/W			√	undefined
03FEC2A4H	CAN0 message data byte 4 register 13	C0MDATA413	R/W		√		undefined
03FEC2A5H	CAN0 message data byte 5 register 13	C0MDATA513	R/W		√		undefined

Table 16-16: Register Access Type (8/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC2A6H	CAN0 message data byte 67 register 13	C0MDATA6713	R/W			√	undefined
03FEC2A6H	CAN0 message data byte 6 register 13	C0MDATA613	R/W		√		undefined
03FEC2A7H	CAN0 message data byte 7 register 13	C0MDATA713	R/W		√		undefined
03FEC2A8H	CAN0 message data length code register 13	C0MDLC13	R/W		√		0000xxxxB
03FEC2A9H	CAN0 message configuration register 13	C0MCONF13	R/W		√		undefined
03FEC2AAH	CAN0 message ID register 13	C0MIDL13	R/W			√	undefined
03FEC2ACH		C0MIDH13	R/W			√	undefined
03FEC2AEH	CAN0 message control register 13	C0MCTRL13	R/W			√	00x00000 000xx000B
03FEC2C0H	CAN0 message data byte 01 register 14	C0MDATA0114	R/W			√	undefined
03FEC2C0H	CAN0 message data byte 0 register 14	C0MDATA014	R/W		√		undefined
03FEC2C1H	CAN0 message data byte 1 register 14	C0MDATA114	R/W		√		undefined
03FEC2C2H	CAN0 message data byte 23 register 14	C0MDATA2314	R/W			√	undefined
03FEC2C2H	CAN0 message data byte 2 register 14	C0MDATA214	R/W		√		undefined
03FEC2C3H	CAN0 message data byte 3 register 14	C0MDATA314	R/W		√		undefined
03FEC2C4H	CAN0 message data byte 45 register 14	C0MDATA4514	R/W			√	undefined
03FEC2C4H	CAN0 message data byte 4 register 14	C0MDATA414	R/W		√		undefined
03FEC2C5H	CAN0 message data byte 5 register 14	C0MDATA514	R/W		√		undefined
03FEC2C6H	CAN0 message data byte 67 register 14	C0MDATA6714	R/W			√	undefined
03FEC2C6H	CAN0 message data byte 6 register 14	C0MDATA614	R/W		√		undefined
03FEC2C7H	CAN0 message data byte 7 register 14	C0MDATA714	R/W		√		undefined
03FEC2C8H	CAN0 message data length code register 14	C0MDLC14	R/W		√		0000xxxxB
03FEC2C9H	CAN0 message configuration register 14	C0MCONF14	R/W		√		undefined
03FEC2CAH	CAN0 message ID register 14	C0MIDL14	R/W			√	undefined
03FEC2CCH		C0MIDH14	R/W			√	undefined
03FEC2CEH	CAN0 message control register 14	C0MCTRL14	R/W			√	00x00000 000xx000B
03FEC2E0H	CAN0 message data byte 01 register 15	C0MDATA0115	R/W			√	undefined
03FEC2E0H	CAN0 message data byte 0 register 15	C0MDATA015	R/W		√		undefined
03FEC2E1H	CAN0 message data byte 1 register 15	C0MDATA115	R/W		√		undefined
03FEC2E2H	CAN0 message data byte 23 register 15	C0MDATA2315	R/W			√	undefined
03FEC2E2H	CAN0 message data byte 2 register 15	C0MDATA215	R/W		√		undefined
03FEC2E3H	CAN0 message data byte 3 register 15	C0MDATA315	R/W		√		undefined
03FEC2E4H	CAN0 message data byte 45 register 15	C0MDATA4515	R/W			√	undefined
03FEC2E4H	CAN0 message data byte 4 register 15	C0MDATA415	R/W		√		undefined
03FEC2E5H	CAN0 message data byte 5 register 15	C0MDATA515	R/W		√		undefined
03FEC2E6H	CAN0 message data byte 67 register 15	C0MDATA6715	R/W			√	undefined
03FEC2E6H	CAN0 message data byte 6 register 15	C0MDATA615	R/W		√		undefined
03FEC2E7H	CAN0 message data byte 7 register 15	C0MDATA715	R/W		√		undefined
03FEC2E8H	CAN0 message data length code register 15	C0MDLC15	R/W		√		0000xxxx

Table 16-16: Register Access Type (9/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC2E9H	CAN0 message configuration register 15	C0MCONF15	R/W		√		undefined
03FEC2EAH	CAN0 message ID register 15	C0MIDL15	R/W			√	undefined
03FEC2ECH		C0MIDH15	R/W			√	undefined
03FEC2EEH	CAN0 message control register 15	C0MCTRL15	R/W			√	00x00000 000xx000B
03FEC300H	CAN0 message data byte 01 register 16	C0MDATA0116	R/W			√	undefined
03FEC300H	CAN0 message data byte 0 register 16	C0MDATA016	R/W		√		undefined
03FEC301H	CAN0 message data byte 1 register 16	C0MDATA116	R/W		√		undefined
03FEC302H	CAN0 message data byte 23 register 16	C0MDATA2316	R/W			√	undefined
03FEC302H	CAN0 message data byte 2 register 16	C0MDATA216	R/W		√		undefined
03FEC303H	CAN0 message data byte 3 register 16	C0MDATA316	R/W		√		undefined
03FEC304H	CAN0 message data byte 45 register 16	C0MDATA4516	R/W			√	undefined
03FEC304H	CAN0 message data byte 4 register 16	C0MDATA416	R/W		√		undefined
03FEC305H	CAN0 message data byte 5 register 16	C0MDATA516	R/W		√		undefined
03FEC306H	CAN0 message data byte 67 register 16	C0MDATA6716	R/W			√	undefined
03FEC306H	CAN0 message data byte 6 register 16	C0MDATA616	R/W		√		undefined
03FEC307H	CAN0 message data byte 7 register 16	C0MDATA716	R/W		√		undefined
03FEC308H	CAN0 message data length code register 16	C0MDLC16	R/W		√		0000xxxxB
03FEC309H	CAN0 message configuration register 16	C0MCONF16	R/W		√		undefined
03FEC30AH	CAN0 message ID register 16	C0MIDL16	R/W			√	undefined
03FEC30CH		C0MIDH16	R/W			√	undefined
03FEC30EH	CAN0 message control register 16	C0MCTRL16	R/W			√	00x00000 000xx000B
03FEC320H	CAN0 message data byte 01 register 17	C0MDATA0117	R/W			√	undefined
03FEC320H	CAN0 message data byte 0 register 17	C0MDATA017	R/W		√		undefined
03FEC321H	CAN0 message data byte 1 register 17	C0MDATA117	R/W		√		undefined
03FEC322H	CAN0 message data byte 23 register 17	C0MDATA2317	R/W			√	undefined
03FEC322H	CAN0 message data byte 2 register 17	C0MDATA217	R/W		√		undefined
03FEC323H	CAN0 message data byte 3 register 17	C0MDATA317	R/W		√		undefined
03FEC324H	CAN0 message data byte 45 register 17	C0MDATA4517	R/W			√	undefined
03FEC324H	CAN0 message data byte 4 register 17	C0MDATA417	R/W		√		undefined
03FEC325H	CAN0 message data byte 5 register 17	C0MDATA517	R/W		√		undefined
03FEC326H	CAN0 message data byte 67 register 17	C0MDATA6717	R/W			√	undefined
03FEC326H	CAN0 message data byte 6 register 17	C0MDATA617	R/W		√		undefined
03FEC327H	CAN0 message data byte 7 register 17	C0MDATA717	R/W		√		undefined
03FEC328H	CAN0 message data length code register 17	C0MDLC17	R/W		√		0000xxxxB
03FEC329H	CAN0 message configuration register 17	C0MCONF17	R/W		√		undefined
03FEC32AH	CAN0 message ID register 17	C0MIDL17	R/W			√	undefined
03FEC32CH		C0MIDH17	R/W			√	undefined
03FEC32EH	CAN0 message control register 17	C0MCTRL17	R/W			√	00x00000 000xx000B

Table 16-16: Register Access Type (10/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC340H	CAN0 message data byte 01 register 18	C0MDATA0118	R/W			√	undefined
03FEC340H	CAN0 message data byte 0 register 18	C0MDATA018	R/W		√		undefined
03FEC341H	CAN0 message data byte 1 register 18	C0MDATA118	R/W		√		undefined
03FEC342H	CAN0 message data byte 23 register 18	C0MDATA2318	R/W			√	undefined
03FEC342H	CAN0 message data byte 2 register 18	C0MDATA218	R/W		√		undefined
03FEC343H	CAN0 message data byte 3 register 18	C0MDATA318	R/W		√		undefined
03FEC344H	CAN0 message data byte 45 register 18	C0MDATA4518	R/W			√	undefined
03FEC344H	CAN0 message data byte 4 register 18	C0MDATA418	R/W		√		undefined
03FEC345H	CAN0 message data byte 5 register 18	C0MDATA518	R/W		√		undefined
03FEC346H	CAN0 message data byte 67 register 18	C0MDATA6718	R/W			√	undefined
03FEC346H	CAN0 message data byte 6 register 18	C0MDATA618	R/W		√		undefined
03FEC347H	CAN0 message data byte 7 register 18	C0MDATA718	R/W		√		undefined
03FEC348H	CAN0 message data length code register 18	C0MDLC18	R/W		√		0000xxxxB
03FEC349H	CAN0 message configuration register 18	C0MCONF18	R/W		√		undefined
03FEC34AH	CAN0 message ID register 18	C0MIDL18	R/W			√	undefined
03FEC34CH		C0MIDH18	R/W			√	undefined
03FEC34EH	CAN0 message control register 18	C0MCTRL18	R/W			√	00x00000 000xx000B
03FEC360H	CAN0 message data byte 01 register 19	C0MDATA0119	R/W			√	undefined
03FEC360H	CAN0 message data byte 0 register 19	C0MDATA019	R/W		√		undefined
03FEC361H	CAN0 message data byte 1 register 19	C0MDATA119	R/W		√		undefined
03FEC362H	CAN0 message data byte 23 register 19	C0MDATA2319	R/W			√	undefined
03FEC362H	CAN0 message data byte 2 register 19	C0MDATA219	R/W		√		undefined
03FEC363H	CAN0 message data byte 3 register 19	C0MDATA319	R/W		√		undefined
03FEC364H	CAN0 message data byte 45 register 19	C0MDATA4519	R/W			√	undefined
03FEC364H	CAN0 message data byte 4 register 19	C0MDATA419	R/W		√		undefined
03FEC365H	CAN0 message data byte 5 register 19	C0MDATA519	R/W		√		undefined
03FEC366H	CAN0 message data byte 67 register 19	C0MDATA6719	R/W			√	undefined
03FEC366H	CAN0 message data byte 6 register 19	C0MDATA619	R/W		√		undefined
03FEC367H	CAN0 message data byte 7 register 19	C0MDATA719	R/W		√		undefined
03FEC368H	CAN0 message data length code register 19	C0MDLC19	R/W		√		0000xxxxB
03FEC369H	CAN0 message configuration register 19	C0MCONF19	R/W		√		undefined
03FEC36AH	CAN0 message ID register 19	C0MIDL19	R/W			√	undefined
03FEC36CH		C0MIDH19	R/W			√	undefined
03FEC36EH	CAN0 message control register 19	C0MCTRL19	R/W			√	00x00000 000xx000B
03FEC380H	CAN0 message data byte 01 register 20	C0MDATA0120	R/W			√	undefined
03FEC380H	CAN0 message data byte 0 register 20	C0MDATA020	R/W		√		undefined
03FEC381H	CAN0 message data byte 1 register 20	C0MDATA120	R/W		√		undefined

Table 16-16: Register Access Type (11/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC382H	CAN0 message data byte 23 register 20	C0MDATA2320	R/W			√	undefined
03FEC382H	CAN0 message data byte 2 register 20	C0MDATA220	R/W		√		undefined
03FEC383H	CAN0 message data byte 3 register 20	C0MDATA320	R/W		√		undefined
03FEC384H	CAN0 message data byte 45 register 20	C0MDATA4520	R/W			√	undefined
03FEC384H	CAN0 message data byte 4 register 20	C0MDATA420	R/W		√		undefined
03FEC385H	CAN0 message data byte 5 register 20	C0MDATA520	R/W		√		undefined
03FEC386H	CAN0 message data byte 67 register 20	C0MDATA6720	R/W			√	undefined
03FEC386H	CAN0 message data byte 6 register 20	C0MDATA620	R/W		√		undefined
03FEC387H	CAN0 message data byte 7 register 20	C0MDATA720	R/W		√		undefined
03FEC388H	CAN0 message data length code register 20	C0MDLC20	R/W		√		0000xxxxB
03FEC389H	CAN0 message configuration register 20	C0MCONF20	R/W		√		undefined
03FEC38AH	CAN0 message ID register 20	C0MIDL20	R/W			√	undefined
03FEC38CH		C0MIDH20	R/W			√	undefined
03FEC38EH	CAN0 message control register 20	C0MCTRL20	R/W			√	00x00000 000xx000B
03FEC3A0H	CAN0 message data byte 01 register 21	C0MDATA0121	R/W			√	undefined
03FEC3A0H	CAN0 message data byte 0 register 21	C0MDATA021	R/W		√		undefined
03FEC3A1H	CAN0 message data byte 1 register 21	C0MDATA121	R/W		√		undefined
03FEC3A2H	CAN0 message data byte 23 register 21	C0MDATA2321	R/W			√	undefined
03FEC3A2H	CAN0 message data byte 2 register 21	C0MDATA221	R/W		√		undefined
03FEC3A3H	CAN0 message data byte 3 register 21	C0MDATA321	R/W		√		undefined
03FEC3A4H	CAN0 message data byte 45 register 21	C0MDATA4521	R/W			√	undefined
03FEC3A4H	CAN0 message data byte 4 register 21	C0MDATA421	R/W		√		undefined
03FEC3A5H	CAN0 message data byte 5 register 21	C0MDATA521	R/W		√		undefined
03FEC3A6H	CAN0 message data byte 67 register 21	C0MDATA6721	R/W			√	undefined
03FEC3A6H	CAN0 message data byte 6 register 21	C0MDATA621	R/W		√		undefined
03FEC3A7H	CAN0 message data byte 7 register 21	C0MDATA721	R/W		√		undefined
03FEC3A8H	CAN0 message data length code register 21	C0MDLC21	R/W		√		0000xxxxB
03FEC3A9H	CAN0 message configuration register 21	C0MCONF21	R/W		√		undefined
03FEC3AAH	CAN0 message ID register 21	C0MIDL21	R/W			√	undefined
03FEC3ACH		C0MIDH21	R/W			√	undefined
03FEC3AEH	CAN0 message control register 21	C0MCTRL21	R/W			√	00x00000 000xx000B
03FEC3C0H	CAN0 message data byte 01 register 22	C0MDATA0122	R/W			√	undefined
03FEC3C0H	CAN0 message data byte 0 register 22	C0MDATA022	R/W		√		undefined
03FEC3C1H	CAN0 message data byte 1 register 22	C0MDATA122	R/W		√		undefined
03FEC3C2H	CAN0 message data byte 23 register 22	C0MDATA2322	R/W			√	undefined
03FEC3C2H	CAN0 message data byte 2 register 22	C0MDATA222	R/W		√		undefined
03FEC3C3H	CAN0 message data byte 3 register 22	C0MDATA322	R/W		√		undefined

Table 16-16: Register Access Type (12/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC3C4H	CAN0 message data byte 45 register 22	C0MDATA4522	R/W			√	undefined
03FEC3C4H	CAN0 message data byte 4 register 22	C0MDATA422	R/W		√		undefined
03FEC3C5H	CAN0 message data byte 5 register 22	C0MDATA522	R/W		√		undefined
03FEC3C6H	CAN0 message data byte 67 register 22	C0MDATA6722	R/W			√	undefined
03FEC3C6H	CAN0 message data byte 6 register 22	C0MDATA622	R/W		√		undefined
03FEC3C7H	CAN0 message data byte 7 register 22	C0MDATA722	R/W		√		undefined
03FEC3C8H	CAN0 message data length code register 22	C0MDLC22	R/W		√		0000xxxxB
03FEC3C9H	CAN0 message configuration register 22	C0MCONF22	R/W		√		undefined
03FEC3CAH	CAN0 message ID register 22	C0MIDL22	R/W			√	undefined
03FEC3CCH		C0MIDH22	R/W			√	undefined
03FEC3CEH	CAN0 message control register 22	C0MCTRL22	R/W			√	00x00000 000xx000B
03FEC3E0H	CAN0 message data byte 01 register 23	C0MDATA0123	R/W			√	undefined
03FEC3E0H	CAN0 message data byte 0 register 23	C0MDATA023	R/W		√		undefined
03FEC3E1H	CAN0 message data byte 1 register 23	C0MDATA123	R/W		√		undefined
03FEC3E2H	CAN0 message data byte 23 register 23	C0MDATA2323	R/W			√	undefined
03FEC3E2H	CAN0 message data byte 2 register 23	C0MDATA223	R/W		√		undefined
03FEC3E3H	CAN0 message data byte 3 register 23	C0MDATA323	R/W		√		undefined
03FEC3E4H	CAN0 message data byte 45 register 23	C0MDATA4523	R/W			√	undefined
03FEC3E4H	CAN0 message data byte 4 register 23	C0MDATA423	R/W		√		undefined
03FEC3E5H	CAN0 message data byte 5 register 23	C0MDATA523	R/W		√		undefined
03FEC3E6H	CAN0 message data byte 67 register 23	C0MDATA6723	R/W			√	undefined
03FEC3E6H	CAN0 message data byte 6 register 23	C0MDATA623	R/W		√		undefined
03FEC3E7H	CAN0 message data byte 7 register 23	C0MDATA723	R/W		√		undefined
03FEC3E8H	CAN0 message data length code register 23	C0MDLC23	R/W		√		0000xxxxB
03FEC3E9H	CAN0 message configuration register 23	C0MCONF23	R/W		√		undefined
03FEC3EAH	CAN0 message ID register 23	C0MIDL23	R/W			√	undefined
03FEC3ECH		C0MIDH23	R/W			√	undefined
03FEC3EEH	CAN0 message control register 23	C0MCTRL23	R/W			√	00x00000 000xx000B
03FEC400H	CAN0 message data byte 01 register 24	C0MDATA0124	R/W			√	undefined
03FEC400H	CAN0 message data byte 0 register 24	C0MDATA024	R/W		√		undefined
03FEC401H	CAN0 message data byte 1 register 24	C0MDATA124	R/W		√		undefined
03FEC402H	CAN0 message data byte 23 register 24	C0MDATA2324	R/W			√	undefined
03FEC402H	CAN0 message data byte 2 register 24	C0MDATA224	R/W		√		undefined
03FEC403H	CAN0 message data byte 3 register 24	C0MDATA324	R/W		√		undefined
03FEC404H	CAN0 message data byte 45 register 24	C0MDATA4524	R/W			√	undefined
03FEC404H	CAN0 message data byte 4 register 24	C0MDATA424	R/W		√		undefined
03FEC405H	CAN0 message data byte 5 register 24	C0MDATA524	R/W		√		undefined

Table 16-16: Register Access Type (13/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC406H	CAN0 message data byte 67 register 24	C0MDATA6724	R/W			√	undefined
03FEC406H	CAN0 message data byte 6 register 24	C0MDATA624	R/W		√		undefined
03FEC407H	CAN0 message data byte 7 register 24	C0MDATA724	R/W		√		undefined
03FEC408H	CAN0 message data length code register 24	C0MDLC24	R/W		√		0000xxxxB
03FEC409H	CAN0 message configuration register 24	C0MCONF24	R/W		√		undefined
03FEC40AH	CAN0 message ID register 24	C0MIDL24	R/W			√	undefined
03FEC40CH		C0MIDH24	R/W			√	undefined
03FEC40EH	CAN0 message control register 24	C0MCTRL24	R/W			√	00x00000 000xx000B
03FEC420H	CAN0 message data byte 01 register 25	C0MDATA0125	R/W			√	undefined
03FEC420H	CAN0 message data byte 0 register 25	C0MDATA025	R/W		√		undefined
03FEC421H	CAN0 message data byte 1 register 25	C0MDATA125	R/W		√		undefined
03FEC422H	CAN0 message data byte 23 register 25	C0MDATA2325	R/W			√	undefined
03FEC422H	CAN0 message data byte 2 register 25	C0MDATA225	R/W		√		undefined
03FEC423H	CAN0 message data byte 3 register 25	C0MDATA325	R/W		√		undefined
03FEC424H	CAN0 message data byte 45 register 25	C0MDATA4525	R/W			√	undefined
03FEC424H	CAN0 message data byte 4 register 25	C0MDATA425	R/W		√		undefined
03FEC425H	CAN0 message data byte 5 register 25	C0MDATA525	R/W		√		undefined
03FEC426H	CAN0 message data byte 67 register 25	C0MDATA6725	R/W			√	undefined
03FEC426H	CAN0 message data byte 6 register 25	C0MDATA625	R/W		√		undefined
03FEC427H	CAN0 message data byte 7 register 25	C0MDATA725	R/W		√		undefined
03FEC428H	CAN0 message data length code register 25	C0MDLC25	R/W		√		0000xxxxB
03FEC429H	CAN0 message configuration register 25	C0MCONF25	R/W		√		undefined
03FEC42AH	CAN0 message ID register 25	C0MIDL25	R/W			√	undefined
03FEC42CH		C0MIDH25	R/W			√	undefined
03FEC42EH	CAN0 message control register 25	C0MCTRL25	R/W			√	00x00000 000xx000B
03FEC440H	CAN0 message data byte 01 register 26	C0MDATA0126	R/W			√	undefined
03FEC440H	CAN0 message data byte 0 register 26	C0MDATA026	R/W		√		undefined
03FEC441H	CAN0 message data byte 1 register 26	C0MDATA126	R/W		√		undefined
03FEC442H	CAN0 message data byte 23 register 26	C0MDATA2326	R/W			√	undefined
03FEC442H	CAN0 message data byte 2 register 26	C0MDATA226	R/W		√		undefined
03FEC443H	CAN0 message data byte 3 register 26	C0MDATA326	R/W		√		undefined
03FEC444H	CAN0 message data byte 45 register 26	C0MDATA4526	R/W			√	undefined
03FEC444H	CAN0 message data byte 4 register 26	C0MDATA426	R/W		√		undefined
03FEC445H	CAN0 message data byte 5 register 26	C0MDATA526	R/W		√		undefined
03FEC446H	CAN0 message data byte 67 register 26	C0MDATA6726	R/W			√	undefined
03FEC446H	CAN0 message data byte 6 register 26	C0MDATA626	R/W		√		undefined
03FEC447H	CAN0 message data byte 7 register 26	C0MDATA726	R/W		√		undefined

Table 16-16: Register Access Type (14/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC448H	CAN0 message data length code register 26	C0MDLC26	R/W		√		0000xxxxB
03FEC449H	CAN0 message configuration register 26	C0MCONF26	R/W		√		undefined
03FEC44AH	CAN0 message ID register 26	C0MIDL26	R/W			√	undefined
03FEC44CH		C0MIDH26	R/W			√	undefined
03FEC44EH	CAN0 message control register 26	C0MCTRL26	R/W			√	00x00000 000xx000B
03FEC460H	CAN0 message data byte 01 register 27	C0MDATA0127	R/W			√	undefined
03FEC460H	CAN0 message data byte 0 register 27	C0MDATA027	R/W		√		undefined
03FEC461H	CAN0 message data byte 1 register 27	C0MDATA127	R/W		√		undefined
03FEC462H	CAN0 message data byte 23 register 27	C0MDATA2327	R/W			√	undefined
03FEC462H	CAN0 message data byte 2 register 27	C0MDATA227	R/W		√		undefined
03FEC463H	CAN0 message data byte 3 register 27	C0MDATA327	R/W		√		undefined
03FEC464H	CAN0 message data byte 45 register 27	C0MDATA4527	R/W			√	undefined
03FEC464H	CAN0 message data byte 4 register 27	C0MDATA427	R/W		√		undefined
03FEC465H	CAN0 message data byte 5 register 27	C0MDATA527	R/W		√		undefined
03FEC466H	CAN0 message data byte 67 register 27	C0MDATA6727	R/W			√	undefined
03FEC466H	CAN0 message data byte 6 register 27	C0MDATA627	R/W		√		undefined
03FEC467H	CAN0 message data byte 7 register 27	C0MDATA727	R/W		√		undefined
03FEC468H	CAN0 message data length code register 27	C0MDLC27	R/W		√		0000xxxxB
03FEC469H	CAN0 message configuration register 27	C0MCONF27	R/W		√		undefined
03FEC46AH	CAN0 message ID register 27	C0MIDL27	R/W			√	undefined
03FEC46CH		C0MIDH27	R/W			√	undefined
03FEC46EH	CAN0 message control register 27	C0MCTRL27	R/W			√	00x00000 000xx000B
03FEC480H	CAN0 message data byte 01 register 28	C0MDATA0128	R/W			√	undefined
03FEC480H	CAN0 message data byte 0 register 28	C0MDATA028	R/W		√		undefined
03FEC481H	CAN0 message data byte 1 register 28	C0MDATA128	R/W		√		undefined
03FEC482H	CAN0 message data byte 23 register 28	C0MDATA2328	R/W			√	undefined
03FEC482H	CAN0 message data byte 2 register 28	C0MDATA228	R/W		√		undefined
03FEC483H	CAN0 message data byte 3 register 28	C0MDATA328	R/W		√		undefined
03FEC484H	CAN0 message data byte 45 register 28	C0MDATA4528	R/W			√	undefined
03FEC484H	CAN0 message data byte 4 register 28	C0MDATA428	R/W		√		undefined
03FEC485H	CAN0 message data byte 5 register 28	C0MDATA528	R/W		√		undefined
03FEC486H	CAN0 message data byte 67 register 28	C0MDATA6728	R/W			√	undefined
03FEC486H	CAN0 message data byte 6 register 28	C0MDATA628	R/W		√		undefined
03FEC487H	CAN0 message data byte 7 register 28	C0MDATA728	R/W		√		undefined
03FEC488H	CAN0 message data length code register 28	C0MDLC28	R/W		√		0000xxxxB
03FEC489H	CAN0 message configuration register 28	C0MCONF28	R/W		√		undefined
03FEC48AH	CAN0 message ID register 28	C0MIDL28	R/W			√	undefined
03FEC48CH		C0MIDH28	R/W			√	undefined

Table 16-16: Register Access Type (15/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC48EH	CAN0 message control register 28	C0MCTRL28	R/W			√	00x00000 000xx000B
03FEC4A0H	CAN0 message data byte 01 register 29	C0MDATA0129	R/W			√	undefined
03FEC4A0H	CAN0 message data byte 0 register 29	C0MDATA029	R/W		√		undefined
03FEC4A1H	CAN0 message data byte 1 register 29	C0MDATA129	R/W		√		undefined
03FEC4A2H	CAN0 message data byte 23 register 29	C0MDATA2329	R/W			√	undefined
03FEC4A2H	CAN0 message data byte 2 register 29	C0MDATA229	R/W		√		undefined
03FEC4A3H	CAN0 message data byte 3 register 29	C0MDATA329	R/W		√		undefined
03FEC4A4H	CAN0 message data byte 45 register 29	C0MDATA4529	R/W			√	undefined
03FEC4A4H	CAN0 message data byte 4 register 29	C0MDATA429	R/W		√		undefined
03FEC4A5H	CAN0 message data byte 5 register 29	C0MDATA529	R/W		√		undefined
03FEC4A6H	CAN0 message data byte 67 register 29	C0MDATA6729	R/W			√	undefined
03FEC4A6H	CAN0 message data byte 6 register 29	C0MDATA629	R/W		√		undefined
03FEC4A7H	CAN0 message data byte 7 register 29	C0MDATA729	R/W		√		undefined
03FEC4A8H	CAN0 message data length code register 29	C0MDLC29	R/W		√		0000xxxxB
03FEC4A9H	CAN0 message configuration register 29	C0MCONF29	R/W		√		undefined
03FEC4AAH	CAN0 message ID register 29	C0MIDL29	R/W			√	undefined
03FEC4ACH		C0MIDH29	R/W			√	undefined
03FEC4AEH	CAN0 message control register 29	C0MCTRL29	R/W			√	00x00000 000xx000B
03FEC4C0H	CAN0 message data byte 01 register 30	C0MDATA0130	R/W			√	undefined
03FEC4C0H	CAN0 message data byte 0 register 30	C0MDATA030	R/W		√		undefined
03FEC4C1H	CAN0 message data byte 1 register 30	C0MDATA130	R/W		√		undefined
03FEC4C2H	CAN0 message data byte 23 register 30	C0MDATA2330	R/W			√	undefined
03FEC4C2H	CAN0 message data byte 2 register 30	C0MDATA230	R/W		√		undefined
03FEC4C3H	CAN0 message data byte 3 register 30	C0MDATA330	R/W		√		undefined
03FEC4C4H	CAN0 message data byte 45 register 30	C0MDATA4530	R/W			√	undefined
03FEC4C4H	CAN0 message data byte 4 register 30	C0MDATA430	R/W		√		undefined
03FEC4C5H	CAN0 message data byte 5 register 30	C0MDATA530	R/W		√		undefined
03FEC4C6H	CAN0 message data byte 67 register 30	C0MDATA6730	R/W			√	undefined
03FEC4C6H	CAN0 message data byte 6 register 30	C0MDATA630	R/W		√		undefined
03FEC4C7H	CAN0 message data byte 7 register 30	C0MDATA730	R/W		√		undefined
03FEC4C8H	CAN0 message data length code register 30	C0MDLC30	R/W		√		0000xxxxB
03FEC4C9H	CAN0 message configuration register 30	C0MCONF30	R/W		√		undefined
03FEC4CAH	CAN0 message ID register 30	C0MIDL30	R/W			√	undefined
03FEC4CCH		C0MIDH30	R/W			√	undefined
03FEC4CEH	CAN0 message control register 30	C0MCTRL30	R/W			√	00x00000 000xx000B
03FEC4E0H	CAN0 message data byte 01 register 31	C0MDATA0131	R/W			√	undefined
03FEC4E0H	CAN0 message data byte 0 register 31	C0MDATA031	R/W		√		undefined
03FEC4E1H	CAN0 message data byte 1 register 31	C0MDATA131	R/W		√		undefined

Table 16-16: Register Access Type (16/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC4E2H	CAN0 message data byte 23 register 31	C0MDATA2331	R/W			√	undefined
03FEC4E2H	CAN0 message data byte 2 register 31	C0MDATA231	R/W		√		undefined
03FEC4E3H	CAN0 message data byte 3 register 31	C0MDATA331	R/W		√		undefined
03FEC4E4H	CAN0 message data byte 45 register 31	C0MDATA4531	R/W			√	undefined
03FEC4E4H	CAN0 message data byte 4 register 31	C0MDATA431	R/W		√		undefined
03FEC4E5H	CAN0 message data byte 5 register 31	C0MDATA531	R/W		√		undefined
03FEC4E6H	CAN0 message data byte 67 register 31	C0MDATA6731	R/W			√	undefined
03FEC4E6H	CAN0 message data byte 6 register 31	C0MDATA631	R/W		√		undefined
03FEC4E7H	CAN0 message data byte 7 register 31	C0MDATA731	R/W		√		undefined
03FEC4E8H	CAN0 message data length code register 31	C0MDLC31	R/W		√		0000xxxx
03FEC4E9H	CAN0 message configuration register 31	C0MCONF31	R/W		√		undefined
03FEC4EAH	CAN0 message ID register 31	C0MIDL31	R/W			√	undefined
03FEC4ECH		C0MIDH31	R/W			√	undefined
03FEC4EEH	CAN0 message control register 31	C0MCTRL31	R/W			√	00x00000 000xx000B
03FEC600H	CAN1 global control register	C1GMCTRL	R/W	–	–	√	0000H
03FEC602H	CAN1 global clock select register	C1GMCS	R/W	–	√	–	0FH
03FEC606H	CAN1 global automatic block transmission control register	C1GMABT	R/W	–	–	√	0000H
03FEC608H	CAN1 global automatic block transmission delay setting register	C1GMABTD	R/W	–	√	–	00H
03FEC640H	CAN1 module mask 1 register	C1MASK1L	R/W	–	–	√	undefined
03FEC642H		C1MASK1H					
03FEC644H	CAN1 module mask 2 register	C1MASK2L	R/W	–	–	√	undefined
03FEC646H		C1MASK2H					
03FEC648H	CAN1 module mask 3 register	C1MASK3L	R/W	–	–	√	undefined
03FEC64AH		C1MASK3H					
03FEC64CH	CAN1 module mask 4 register	C1MASK4L	R/W	–	–	√	undefined
03FEC64EH		C1MASK4H					
03FEC650H	CAN1 module control register	C1CTRL	R/W	–	–	√	0000H
03FEC652H	CAN1 module last error information register	C1LEC	R/W	–	√	–	00H
03FEC653H	CAN1 module information register	C1INFO	R	–	√	–	00H
03FEC654H	CAN1 module error counter register	C1ERC	R	–	–	√	0000H
03FEC656H	CAN1 module interrupt enable register	C1IE	R/W	–	–	√	0000H
03FEC658H	CAN1 module interrupt status register	C1INTS	R/W	–	–	√	0000H
03FEC65AH	CAN1 module bit rate prescaler register	C1BRP	R/W	–	√	–	FFH
03FEC65CH	CAN1 module bit rate register	C1BTR	R/W	–	–	√	370FH
03FEC65EH	CAN1 module last in-pointer register	C1LIPT	R	–	√	–	undefined
03FEC660H	CAN1 module receive history list register	C1RGPT	R/W	–	–	√	xx02H
03FEC662H	CAN1 module last out-pointer register	C1LOPT	R	–	√	–	undefined
03FEC664H	CAN1 module transmit history list register	C1TGPT	R/W	–	–	√	xx02H

Table 16-16: Register Access Type (17/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC666H	CAN1 module time stamp register	C1TS	R/W	–	–	√	0000H
03FEC700H	CAN1 message data byte 01 register 00	C1MDATA0100	R/W			√	undefined
03FEC700H	CAN1 message data byte 0 register 00	C1MDATA000	R/W		√		undefined
03FEC701H	CAN1 message data byte 1 register 00	C1MDATA100	R/W		√		undefined
03FEC702H	CAN1 message data byte 23 register 00	C1MDATA2300	R/W			√	undefined
03FEC702H	CAN1 message data byte 2 register 00	C1MDATA200	R/W		√		undefined
03FEC703H	CAN1 message data byte 3 register 00	C1MDATA300	R/W		√		undefined
03FEC704H	CAN1 message data byte 45 register 00	C1MDATA4500	R/W			√	undefined
03FEC704H	CAN1 message data byte 4 register 00	C1MDATA400	R/W		√		undefined
03FEC705H	CAN1 message data byte 5 register 00	C1MDATA500	R/W		√		undefined
03FEC706H	CAN1 message data byte 67 register 00	C1MDATA6700	R/W			√	undefined
03FEC706H	CAN1 message data byte 6 register 00	C1MDATA600	R/W		√		undefined
03FEC707H	CAN1 message data byte 7 register 00	C1MDATA700	R/W		√		undefined
03FEC708H	CAN1 message data length code register 00	C1MDLC00	R/W		√		0000xxxxB
03FEC709H	CAN1 message configuration register 00	C1MCONF00	R/W		√		undefined
03FEC70AH	CAN1 message ID register 00	C1MIDL00	R/W			√	undefined
03FEC70CH		C1MIDH00	R/W			√	undefined
03FEC70EH	CAN1 message control register 00	C1MCTRL00	R/W			√	00x00000 000xx000B
03FEC720H	CAN1 message data byte 01 register 01	C1MDATA0101	R/W			√	undefined
03FEC720H	CAN1 message data byte 0 register 01	C1MDATA001	R/W		√		undefined
03FEC721H	CAN1 message data byte 1 register 01	C1MDATA101	R/W		√		undefined
03FEC722H	CAN1 message data byte 23 register 01	C1MDATA2301	R/W			√	undefined
03FEC722H	CAN1 message data byte 2 register 01	C1MDATA201	R/W		√		undefined
03FEC723H	CAN1 message data byte 3 register 01	C1MDATA301	R/W		√		undefined
03FEC724H	CAN1 message data byte 45 register 01	C1MDATA4501	R/W			√	undefined
03FEC724H	CAN1 message data byte 4 register 01	C1MDATA401	R/W		√		undefined
03FEC725H	CAN1 message data byte 5 register 01	C1MDATA501	R/W		√		undefined
03FEC726H	CAN1 message data byte 67 register 01	C1MDATA6701	R/W			√	undefined
03FEC726H	CAN1 message data byte 6 register 01	C1MDATA601	R/W		√		undefined
03FEC727H	CAN1 message data byte 7 register 01	C1MDATA701	R/W		√		undefined
03FEC728H	CAN1 message data length code register 01	C1MDLC01	R/W		√		0000xxxxB
03FEC729H	CAN1 message configuration register 01	C1MCONF01	R/W		√		undefined
03FEC72AH	CAN1 message ID register 01	C1MIDL01	R/W			√	undefined
03FEC72CH		C1MIDH01	R/W			√	undefined
03FEC72EH	CAN1 message control register 01	C1MCTRL01	R/W			√	00x00000 000xx000B
03FEC740H	CAN1 message data byte 01 register 02	C1MDATA0102	R/W			√	undefined
03FEC740H	CAN1 message data byte 0 register 02	C1MDATA002	R/W		√		undefined
03FEC741H	CAN1 message data byte 1 register 02	C1MDATA102	R/W		√		undefined

Table 16-16: Register Access Type (18/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC742H	CAN1 message data byte 23 register 02	C1MDATA2302	R/W			√	undefined
03FEC742H	CAN1 message data byte 2 register 02	C1MDATA202	R/W		√		undefined
03FEC743H	CAN1 message data byte 3 register 02	C1MDATA302	R/W		√		undefined
03FEC744H	CAN1 message data byte 45 register 02	C1MDATA4502	R/W			√	undefined
03FEC744H	CAN1 message data byte 4 register 02	C1MDATA402	R/W		√		undefined
03FEC745H	CAN1 message data byte 5 register 02	C1MDATA502	R/W		√		undefined
03FEC746H	CAN1 message data byte 67 register 02	C1MDATA6702	R/W			√	undefined
03FEC746H	CAN1 message data byte 6 register 02	C1MDATA602	R/W		√		undefined
03FEC747H	CAN1 message data byte 7 register 02	C1MDATA702	R/W		√		undefined
03FEC748H	CAN1 message data length code register 02	C1MDLC02	R/W		√		0000xxxxB
03FEC749H	CAN1 message configuration register 02	C1MCONF02	R/W		√		undefined
03FEC74AH	CAN1 message ID register 02	C1MIDL02	R/W			√	undefined
03FEC74CH		C1MIDH02	R/W			√	undefined
03FEC74EH	CAN1 message control register 02	C1MCTRL02	R/W			√	00x00000 000xx000B
03FEC760H	CAN1 message data byte 01 register 03	C1MDATA0103	R/W			√	undefined
03FEC760H	CAN1 message data byte 0 register 03	C1MDATA003	R/W		√		undefined
03FEC761H	CAN1 message data byte 1 register 03	C1MDATA103	R/W		√		undefined
03FEC762H	CAN1 message data byte 23 register 03	C1MDATA2303	R/W			√	undefined
03FEC762H	CAN1 message data byte 2 register 03	C1MDATA203	R/W		√		undefined
03FEC763H	CAN1 message data byte 3 register 03	C1MDATA303	R/W		√		undefined
03FEC764H	CAN1 message data byte 45 register 03	C1MDATA4503	R/W			√	undefined
03FEC764H	CAN1 message data byte 4 register 03	C1MDATA403	R/W		√		undefined
03FEC765H	CAN1 message data byte 5 register 03	C1MDATA503	R/W		√		undefined
03FEC766H	CAN1 message data byte 67 register 03	C1MDATA6703	R/W			√	undefined
03FEC766H	CAN1 message data byte 6 register 03	C1MDATA603	R/W		√		undefined
03FEC767H	CAN1 message data byte 7 register 03	C1MDATA703	R/W		√		undefined
03FEC768H	CAN1 message data length code register 03	C1MDLC03	R/W		√		0000xxxxB
03FEC769H	CAN1 message configuration register 03	C1MCONF03	R/W		√		undefined
03FEC76AH	CAN1 message ID register 03	C1MIDL03	R/W			√	undefined
03FEC76CH		C1MIDH03	R/W			√	undefined
03FEC76EH	CAN1 message control register 03	C1MCTRL03	R/W			√	00x00000 000xx000B
03FEC780H	CAN1 message data byte 01 register 04	C1MDATA0104	R/W			√	undefined
03FEC780H	CAN1 message data byte 0 register 04	C1MDATA004	R/W		√		undefined
03FEC781H	CAN1 message data byte 1 register 04	C1MDATA104	R/W		√		undefined
03FEC782H	CAN1 message data byte 23 register 04	C1MDATA2304	R/W			√	undefined
03FEC782H	CAN1 message data byte 2 register 04	C1MDATA204	R/W		√		undefined
03FEC783H	CAN1 message data byte 3 register 04	C1MDATA304	R/W		√		undefined

Table 16-16: Register Access Type (19/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC784H	CAN1 message data byte 45 register 04	C1MDATA4504	R/W			√	undefined
03FEC784H	CAN1 message data byte 4 register 04	C1MDATA404	R/W		√		undefined
03FEC785H	CAN1 message data byte 5 register 04	C1MDATA504	R/W		√		undefined
03FEC786H	CAN1 message data byte 67 register 04	C1MDATA6704	R/W			√	undefined
03FEC786H	CAN1 message data byte 6 register 04	C1MDATA604	R/W		√		undefined
03FEC787H	CAN1 message data byte 7 register 04	C1MDATA704	R/W		√		undefined
03FEC788H	CAN1 message data length code register 04	C1MDLC04	R/W		√		0000xxxxB
03FEC789H	CAN1 message configuration register 04	C1MCONF04	R/W		√		undefined
03FEC78AH	CAN1 message ID register 04	C1MIDL04	R/W			√	undefined
03FEC78CH		C1MIDH04	R/W			√	undefined
03FEC78EH	CAN1 message control register 04	C1MCTRL04	R/W			√	00x00000 000xx000B
03FEC7A0H	CAN1 message data byte 01 register 05	C1MDATA0105	R/W			√	undefined
03FEC7A0H	CAN1 message data byte 0 register 05	C1MDATA005	R/W		√		undefined
03FEC7A1H	CAN1 message data byte 1 register 05	C1MDATA105	R/W		√		undefined
03FEC7A2H	CAN1 message data byte 23 register 05	C1MDATA2305	R/W			√	undefined
03FEC7A2H	CAN1 message data byte 2 register 05	C1MDATA205	R/W		√		undefined
03FEC7A3H	CAN1 message data byte 3 register 05	C1MDATA305	R/W		√		undefined
03FEC7A4H	CAN1 message data byte 45 register 05	C1MDATA4505	R/W			√	undefined
03FEC7A4H	CAN1 message data byte 4 register 05	C1MDATA405	R/W		√		undefined
03FEC7A5H	CAN1 message data byte 5 register 05	C1MDATA505	R/W		√		undefined
03FEC7A6H	CAN1 message data byte 67 register 05	C1MDATA6705	R/W			√	undefined
03FEC7A6H	CAN1 message data byte 6 register 05	C1MDATA605	R/W		√		undefined
03FEC7A7H	CAN1 message data byte 7 register 05	C1MDATA705	R/W		√		undefined
03FEC7A8H	CAN1 message data length code register 05	C1MDLC05	R/W		√		0000xxxxB
03FEC7A9H	CAN1 message configuration register 05	C1MCONF05	R/W		√		undefined
03FEC7AAH	CAN1 message ID register 05	C1MIDL05	R/W			√	undefined
03FEC7ACH		C1MIDH05	R/W			√	undefined
03FEC7AEH	CAN1 message control register 05	C1MCTRL05	R/W			√	00x00000 000xx000B
03FEC7C0H	CAN1 message data byte 01 register 06	C1MDATA0106	R/W			√	undefined
03FEC7C0H	CAN1 message data byte 0 register 06	C1MDATA006	R/W		√		undefined
03FEC7C1H	CAN1 message data byte 1 register 06	C1MDATA106	R/W		√		undefined
03FEC7C2H	CAN1 message data byte 23 register 06	C1MDATA2306	R/W			√	undefined
03FEC7C2H	CAN1 message data byte 2 register 06	C1MDATA206	R/W		√		undefined
03FEC7C3H	CAN1 message data byte 3 register 06	C1MDATA306	R/W		√		undefined
03FEC7C4H	CAN1 message data byte 45 register 06	C1MDATA4506	R/W			√	undefined
03FEC7C4H	CAN1 message data byte 4 register 06	C1MDATA406	R/W		√		undefined
03FEC7C5H	CAN1 message data byte 5 register 06	C1MDATA506	R/W		√		undefined

Table 16-16: Register Access Type (20/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC7C6H	CAN1 message data byte 67 register 06	C1MDATA6706	R/W			√	undefined
03FEC7C6H	CAN1 message data byte 6 register 06	C1MDATA606	R/W		√		undefined
03FEC7C7H	CAN1 message data byte 7 register 06	C1MDATA706	R/W		√		undefined
03FEC7C8H	CAN1 message data length code register 06	C1MDLC06	R/W		√		0000xxxxB
03FEC7C9H	CAN1 message configuration register 06	C1MCONF06	R/W		√		undefined
03FEC7CAH	CAN1 message ID register 06	C1MIDL06	R/W			√	undefined
03FEC7CCH		C1MIDH06	R/W			√	undefined
03FEC7CEH	CAN1 message control register 06	C1MCTRL06	R/W			√	00x00000 000xx000B
03FEC7E0H	CAN1 message data byte 01 register 07	C1MDATA0107	R/W			√	undefined
03FEC7E0H	CAN1 message data byte 0 register 07	C1MDATA007	R/W		√		undefined
03FEC7E1H	CAN1 message data byte 1 register 07	C1MDATA107	R/W		√		undefined
03FEC7E2H	CAN1 message data byte 23 register 07	C1MDATA2307	R/W			√	undefined
03FEC7E2H	CAN1 message data byte 2 register 07	C1MDATA207	R/W		√		undefined
03FEC7E3H	CAN1 message data byte 3 register 07	C1MDATA307	R/W		√		undefined
03FEC7E4H	CAN1 message data byte 45 register 07	C1MDATA4507	R/W			√	undefined
03FEC7E4H	CAN1 message data byte 4 register 07	C1MDATA407	R/W		√		undefined
03FEC7E5H	CAN1 message data byte 5 register 07	C1MDATA507	R/W		√		undefined
03FEC7E6H	CAN1 message data byte 67 register 07	C1MDATA6707	R/W			√	undefined
03FEC7E6H	CAN1 message data byte 6 register 07	C1MDATA607	R/W		√		undefined
03FEC7E7H	CAN1 message data byte 7 register 07	C1MDATA707	R/W		√		undefined
03FEC7E8H	CAN1 message data length code register 07	C1MDLC07	R/W		√		0000xxxxB
03FEC7E9H	CAN1 message configuration register 07	C1MCONF07	R/W		√		undefined
03FEC7EAH	CAN1 message ID register 07	C1MIDL07	R/W			√	undefined
03FEC7ECH		C1MIDH07	R/W			√	undefined
03FEC7EEH	CAN1 message control register 07	C1MCTRL07	R/W			√	00x00000 000xx000B
03FEC800H	CAN1 message data byte 01 register 08	C1MDATA0108	R/W			√	undefined
03FEC800H	CAN1 message data byte 0 register 08	C1MDATA008	R/W		√		undefined
03FEC801H	CAN1 message data byte 1 register 08	C1MDATA108	R/W		√		undefined
03FEC802H	CAN1 message data byte 23 register 08	C1MDATA2308	R/W			√	undefined
03FEC802H	CAN1 message data byte 2 register 08	C1MDATA208	R/W		√		undefined
03FEC803H	CAN1 message data byte 3 register 08	C1MDATA308	R/W		√		undefined
03FEC804H	CAN1 message data byte 45 register 08	C1MDATA4508	R/W			√	undefined
03FEC804H	CAN1 message data byte 4 register 08	C1MDATA408	R/W		√		undefined
03FEC805H	CAN1 message data byte 5 register 08	C1MDATA508	R/W		√		undefined
03FEC806H	CAN1 message data byte 67 register 08	C1MDATA6708	R/W			√	undefined
03FEC806H	CAN1 message data byte 6 register 08	C1MDATA608	R/W		√		undefined
03FEC807H	CAN1 message data byte 7 register 08	C1MDATA708	R/W		√		undefined
03FEC808H	CAN1 message data length code register 08	C1MDLC08	R/W		√		0000xxxxB

Table 16-16: Register Access Type (21/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC809H	CAN1 message configuration register 08	C1MCONF08	R/W		√		undefined
03FEC80AH	CAN1 message ID register 08	C1MIDL08	R/W			√	undefined
03FEC80CH		C1MIDH08	R/W			√	undefined
03FEC80EH	CAN1 message control register 08	C1MCTRL08	R/W			√	00x00000 000xx000B
03FEC820H	CAN1 message data byte 01 register 09	C1MDATA0109	R/W			√	undefined
03FEC820H	CAN1 message data byte 0 register 09	C1MDATA009	R/W		√		undefined
03FEC821H	CAN1 message data byte 1 register 09	C1MDATA109	R/W		√		undefined
03FEC822H	CAN1 message data byte 23 register 09	C1MDATA2309	R/W			√	undefined
03FEC822H	CAN1 message data byte 2 register 09	C1MDATA209	R/W		√		undefined
03FEC823H	CAN1 message data byte 3 register 09	C1MDATA309	R/W		√		undefined
03FEC824H	CAN1 message data byte 45 register 09	C1MDATA4509	R/W			√	undefined
03FEC824H	CAN1 message data byte 4 register 09	C1MDATA409	R/W		√		undefined
03FEC825H	CAN1 message data byte 5 register 09	C1MDATA509	R/W		√		undefined
03FEC826H	CAN1 message data byte 67 register 09	C1MDATA6709	R/W			√	undefined
03FEC826H	CAN1 message data byte 6 register 09	C1MDATA609	R/W		√		undefined
03FEC827H	CAN1 message data byte 7 register 09	C1MDATA709	R/W		√		undefined
03FEC828H	CAN1 message data length code register 09	C1MDLC09	R/W		√		0000xxxxB
03FEC829H	CAN1 message configuration register 09	C1MCONF09	R/W		√		undefined
03FEC82AH	CAN1 message ID register 09	C1MIDL09	R/W			√	undefined
03FEC82CH		C1MIDH09	R/W			√	undefined
03FEC82EH	CAN1 message control register 09	C1MCTRL09	R/W			√	00x00000 000xx000B
03FEC840H	CAN1 message data byte 01 register 10	C1MDATA0110	R/W			√	undefined
03FEC840H	CAN1 message data byte 0 register 10	C1MDATA010	R/W		√		undefined
03FEC841H	CAN1 message data byte 1 register 10	C1MDATA110	R/W		√		undefined
03FEC842H	CAN1 message data byte 23 register 10	C1MDATA2310	R/W			√	undefined
03FEC842H	CAN1 message data byte 2 register 10	C1MDATA210	R/W		√		undefined
03FEC843H	CAN1 message data byte 3 register 10	C1MDATA310	R/W		√		undefined
03FEC844H	CAN1 message data byte 45 register 10	C1MDATA4510	R/W			√	undefined
03FEC844H	CAN1 message data byte 4 register 10	C1MDATA410	R/W		√		undefined
03FEC845H	CAN1 message data byte 5 register 10	C1MDATA510	R/W		√		undefined
03FEC846H	CAN1 message data byte 67 register 10	C1MDATA6710	R/W			√	undefined
03FEC846H	CAN1 message data byte 6 register 10	C1MDATA610	R/W		√		undefined
03FEC847H	CAN1 message data byte 7 register 10	C1MDATA710	R/W		√		undefined
03FEC848H	CAN1 message data length code register 10	C1MDLC10	R/W		√		0000xxxxB
03FEC849H	CAN1 message configuration register 10	C1MCONF10	R/W		√		undefined
03FEC84AH	CAN1 message ID register 10	C1MIDL10	R/W			√	undefined
03FEC84CH		C1MIDH10	R/W			√	undefined
03FEC84EH	CAN1 message control register 10	C1MCTRL10	R/W			√	00x00000 000xx000B

Table 16-16: Register Access Type (22/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC860H	CAN1 message data byte 01 register 11	C1MDATA0111	R/W			√	undefined
03FEC860H	CAN1 message data byte 0 register 11	C1MDATA011	R/W		√		undefined
03FEC861H	CAN1 message data byte 1 register 11	C1MDATA111	R/W		√		undefined
03FEC862H	CAN1 message data byte 23 register 11	C1MDATA2311	R/W			√	undefined
03FEC862H	CAN1 message data byte 2 register 11	C1MDATA211	R/W		√		undefined
03FEC863H	CAN1 message data byte 3 register 11	C1MDATA311	R/W		√		undefined
03FEC864H	CAN1 message data byte 45 register 11	C1MDATA4511	R/W			√	undefined
03FEC864H	CAN1 message data byte 4 register 11	C1MDATA411	R/W		√		undefined
03FEC865H	CAN1 message data byte 5 register 11	C1MDATA511	R/W		√		undefined
03FEC866H	CAN1 message data byte 67 register 11	C1MDATA6711	R/W			√	undefined
03FEC866H	CAN1 message data byte 6 register 11	C1MDATA611	R/W		√		undefined
03FEC867H	CAN1 message data byte 7 register 11	C1MDATA711	R/W		√		undefined
03FEC868H	CAN1 message data length code register 11	C1MDLC11	R/W		√		0000xxxxB
03FEC869H	CAN1 message configuration register 11	C1MCONF11	R/W		√		undefined
03FEC86AH	CAN1 message ID register 11	C1MIDL11	R/W			√	undefined
03FEC86CH		C1MIDH11	R/W			√	undefined
03FEC86EH	CAN1 message control register 11	C1MCTRL11	R/W			√	00x00000 000xx000B
03FEC880H	CAN1 message data byte 01 register 12	C1MDATA0112	R/W			√	undefined
03FEC880H	CAN1 message data byte 0 register 12	C1MDATA012	R/W		√		undefined
03FEC881H	CAN1 message data byte 1 register 12	C1MDATA112	R/W		√		undefined
03FEC882H	CAN1 message data byte 23 register 12	C1MDATA2312	R/W			√	undefined
03FEC882H	CAN1 message data byte 2 register 12	C1MDATA212	R/W		√		undefined
03FEC883H	CAN1 message data byte 3 register 12	C1MDATA312	R/W		√		undefined
03FEC884H	CAN1 message data byte 45 register 12	C1MDATA4512	R/W			√	undefined
03FEC884H	CAN1 message data byte 4 register 12	C1MDATA412	R/W		√		undefined
03FEC885H	CAN1 message data byte 5 register 12	C1MDATA512	R/W		√		undefined
03FEC886H	CAN1 message data byte 67 register 12	C1MDATA6712	R/W			√	undefined
03FEC886H	CAN1 message data byte 6 register 12	C1MDATA612	R/W		√		undefined
03FEC887H	CAN1 message data byte 7 register 12	C1MDATA712	R/W		√		undefined
03FEC888H	CAN1 message data length code register 12	C1MDLC12	R/W		√		0000xxxxB
03FEC889H	CAN1 message configuration register 12	C1MCONF12	R/W		√		undefined
03FEC88AH	CAN1 message ID register 12	C1MIDL12	R/W			√	undefined
03FEC88CH		C1MIDH12	R/W			√	undefined
03FEC88EH	CAN1 message control register 12	C1MCTRL12	R/W			√	00x00000 000xx000B
03FEC8A0H	CAN1 message data byte 01 register 13	C1MDATA0113	R/W			√	undefined
03FEC8A0H	CAN1 message data byte 0 register 13	C1MDATA013	R/W		√		undefined
03FEC8A1H	CAN1 message data byte 1 register 13	C1MDATA113	R/W		√		undefined

Table 16-16: Register Access Type (23/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC8A2H	CAN1 message data byte 23 register 13	C1MDATA2313	R/W			√	undefined
03FEC8A2H	CAN1 message data byte 2 register 13	C1MDATA213	R/W		√		undefined
03FEC8A3H	CAN1 message data byte 3 register 13	C1MDATA313	R/W		√		undefined
03FEC8A4H	CAN1 message data byte 45 register 13	C1MDATA4513	R/W			√	undefined
03FEC8A4H	CAN1 message data byte 4 register 13	C1MDATA413	R/W		√		undefined
03FEC8A5H	CAN1 message data byte 5 register 13	C1MDATA513	R/W		√		undefined
03FEC8A6H	CAN1 message data byte 67 register 13	C1MDATA6713	R/W			√	undefined
03FEC8A6H	CAN1 message data byte 6 register 13	C1MDATA613	R/W		√		undefined
03FEC8A7H	CAN1 message data byte 7 register 13	C1MDATA713	R/W		√		undefined
03FEC8A8H	CAN1 message data length code register 13	C1MDLC13	R/W		√		0000xxxxB
03FEC8A9H	CAN1 message configuration register 13	C1MCONF13	R/W		√		undefined
03FEC8AAH	CAN1 message ID register 13	C1MIDL13	R/W			√	undefined
03FEC8ACH		C1MIDH13	R/W			√	undefined
03FEC8AEH	CAN1 message control register 13	C1MCTRL13	R/W			√	00x00000 000xx000B
03FEC8C0H	CAN1 message data byte 01 register 14	C1MDATA0114	R/W			√	undefined
03FEC8C0H	CAN1 message data byte 0 register 14	C1MDATA014	R/W		√		undefined
03FEC8C1H	CAN1 message data byte 1 register 14	C1MDATA114	R/W		√		undefined
03FEC8C2H	CAN1 message data byte 23 register 14	C1MDATA2314	R/W			√	undefined
03FEC8C2H	CAN1 message data byte 2 register 14	C1MDATA214	R/W		√		undefined
03FEC8C3H	CAN1 message data byte 3 register 14	C1MDATA314	R/W		√		undefined
03FEC8C4H	CAN1 message data byte 45 register 14	C1MDATA4514	R/W			√	undefined
03FEC8C4H	CAN1 message data byte 4 register 14	C1MDATA414	R/W		√		undefined
03FEC8C5H	CAN1 message data byte 5 register 14	C1MDATA514	R/W		√		undefined
03FEC8C6H	CAN1 message data byte 67 register 14	C1MDATA6714	R/W			√	undefined
03FEC8C6H	CAN1 message data byte 6 register 14	C1MDATA614	R/W		√		undefined
03FEC8C7H	CAN1 message data byte 7 register 14	C1MDATA714	R/W		√		undefined
03FEC8C8H	CAN1 message data length code register 14	C1MDLC14	R/W		√		0000xxxxB
03FEC8C9H	CAN1 message configuration register 14	C1MCONF14	R/W		√		undefined
03FEC8CAH	CAN1 message ID register 14	C1MIDL14	R/W			√	undefined
03FEC8CCH		C1MIDH14	R/W			√	undefined
03FEC8CEH	CAN1 message control register 14	C1MCTRL14	R/W			√	00x00000 000xx000B
03FEC8E0H	CAN1 message data byte 01 register 15	C1MDATA0115	R/W			√	undefined
03FEC8E0H	CAN1 message data byte 0 register 15	C1MDATA015	R/W		√		undefined
03FEC8E1H	CAN1 message data byte 1 register 15	C1MDATA115	R/W		√		undefined
03FEC8E2H	CAN1 message data byte 23 register 15	C1MDATA2315	R/W			√	undefined
03FEC8E2H	CAN1 message data byte 2 register 15	C1MDATA215	R/W		√		undefined
03FEC8E3H	CAN1 message data byte 3 register 15	C1MDATA315	R/W		√		undefined

Table 16-16: Register Access Type (24/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC8E4H	CAN1 message data byte 45 register 15	C1MDATA4515	R/W			√	undefined
03FEC8E4H	CAN1 message data byte 4 register 15	C1MDATA415	R/W		√		undefined
03FEC8E5H	CAN1 message data byte 5 register 15	C1MDATA515	R/W		√		undefined
03FEC8E6H	CAN1 message data byte 67 register 15	C1MDATA6715	R/W			√	undefined
03FEC8E6H	CAN1 message data byte 6 register 15	C1MDATA615	R/W		√		undefined
03FEC8E7H	CAN1 message data byte 7 register 15	C1MDATA715	R/W		√		undefined
03FEC8E8H	CAN1 message data length code register 15	C1MDLC15	R/W		√		0000xxxxB
03FEC8E9H	CAN1 message configuration register 15	C1MCONF15	R/W		√		undefined
03FEC8EAH	CAN1 message ID register 15	C1MIDL15	R/W			√	undefined
03FEC8ECH		C1MIDH15	R/W			√	undefined
03FEC8EEH	CAN1 message control register 15	C1MCTRL15	R/W			√	00x00000 000xx000B
03FEC900H	CAN1 message data byte 01 register 16	C1MDATA0116	R/W			√	undefined
03FEC900H	CAN1 message data byte 0 register 16	C1MDATA016	R/W		√		undefined
03FEC901H	CAN1 message data byte 1 register 16	C1MDATA116	R/W		√		undefined
03FEC902H	CAN1 message data byte 23 register 16	C1MDATA2316	R/W			√	undefined
03FEC902H	CAN1 message data byte 2 register 16	C1MDATA216	R/W		√		undefined
03FEC903H	CAN1 message data byte 3 register 16	C1MDATA316	R/W		√		undefined
03FEC904H	CAN1 message data byte 45 register 16	C1MDATA4516	R/W			√	undefined
03FEC904H	CAN1 message data byte 4 register 16	C1MDATA416	R/W		√		undefined
03FEC905H	CAN1 message data byte 5 register 16	C1MDATA516	R/W		√		undefined
03FEC906H	CAN1 message data byte 67 register 16	C1MDATA6716	R/W			√	undefined
03FEC906H	CAN1 message data byte 6 register 16	C1MDATA616	R/W		√		undefined
03FEC907H	CAN1 message data byte 7 register 16	C1MDATA716	R/W		√		undefined
03FEC908H	CAN1 message data length code register 16	C1MDLC16	R/W		√		0000xxxxB
03FEC909H	CAN1 message configuration register 16	C1MCONF16	R/W		√		undefined
03FEC90AH	CAN1 message ID register 16	C1MIDL16	R/W			√	undefined
03FEC90CH		C1MIDH16	R/W			√	undefined
03FEC90EH	CAN1 message control register 16	C1MCTRL16	R/W			√	00x00000 000xx000B
03FEC920H	CAN1 message data byte 01 register 17	C1MDATA0117	R/W			√	undefined
03FEC920H	CAN1 message data byte 0 register 17	C1MDATA017	R/W		√		undefined
03FEC921H	CAN1 message data byte 1 register 17	C1MDATA117	R/W		√		undefined
03FEC922H	CAN1 message data byte 23 register 17	C1MDATA2317	R/W			√	undefined
03FEC922H	CAN1 message data byte 2 register 17	C1MDATA217	R/W		√		undefined
03FEC923H	CAN1 message data byte 3 register 17	C1MDATA317	R/W		√		undefined
03FEC924H	CAN1 message data byte 45 register 17	C1MDATA4517	R/W			√	undefined
03FEC924H	CAN1 message data byte 4 register 17	C1MDATA417	R/W		√		undefined
03FEC925H	CAN1 message data byte 5 register 17	C1MDATA517	R/W		√		undefined

Table 16-16: Register Access Type (25/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC926H	CAN1 message data byte 67 register 17	C1MDATA6717	R/W			√	undefined
03FEC926H	CAN1 message data byte 6 register 17	C1MDATA617	R/W		√		undefined
03FEC927H	CAN1 message data byte 7 register 17	C1MDATA717	R/W		√		undefined
03FEC928H	CAN1 message data length code register 17	C1MDLC17	R/W		√		0000xxxxB
03FEC929H	CAN1 message configuration register 17	C1MCONF17	R/W		√		undefined
03FEC92AH	CAN1 message ID register 17	C1MIDL17	R/W			√	undefined
03FEC92CH		C1MIDH17	R/W			√	undefined
03FEC92EH	CAN1 message control register 17	C1MCTRL17	R/W			√	00x00000 000xx000B
03FEC940H	CAN1 message data byte 01 register 18	C1MDATA0118	R/W			√	undefined
03FEC940H	CAN1 message data byte 0 register 18	C1MDATA018	R/W		√		undefined
03FEC941H	CAN1 message data byte 1 register 18	C1MDATA118	R/W		√		undefined
03FEC942H	CAN1 message data byte 23 register 18	C1MDATA2318	R/W			√	undefined
03FEC942H	CAN1 message data byte 2 register 18	C1MDATA218	R/W		√		undefined
03FEC943H	CAN1 message data byte 3 register 18	C1MDATA318	R/W		√		undefined
03FEC944H	CAN1 message data byte 45 register 18	C1MDATA4518	R/W			√	undefined
03FEC944H	CAN1 message data byte 4 register 18	C1MDATA418	R/W		√		undefined
03FEC945H	CAN1 message data byte 5 register 18	C1MDATA518	R/W		√		undefined
03FEC946H	CAN1 message data byte 67 register 18	C1MDATA6718	R/W			√	undefined
03FEC946H	CAN1 message data byte 6 register 18	C1MDATA618	R/W		√		undefined
03FEC947H	CAN1 message data byte 7 register 18	C1MDATA718	R/W		√		undefined
03FEC948H	CAN1 message data length code register 18	C1MDLC18	R/W		√		0000xxxxB
03FEC949H	CAN1 message configuration register 18	C1MCONF18	R/W		√		undefined
03FEC94AH	CAN1 message ID register 18	C1MIDL18	R/W			√	undefined
03FEC94CH		C1MIDH18	R/W			√	undefined
03FEC94EH	CAN1 message control register 18	C1MCTRL18	R/W			√	00x00000 000xx000B
03FEC960H	CAN1 message data byte 01 register 19	C1MDATA0119	R/W			√	undefined
03FEC960H	CAN1 message data byte 0 register 19	C1MDATA019	R/W		√		undefined
03FEC961H	CAN1 message data byte 1 register 19	C1MDATA119	R/W		√		undefined
03FEC962H	CAN1 message data byte 23 register 19	C1MDATA2319	R/W			√	undefined
03FEC962H	CAN1 message data byte 2 register 19	C1MDATA219	R/W		√		undefined
03FEC963H	CAN1 message data byte 3 register 19	C1MDATA319	R/W		√		undefined
03FEC964H	CAN1 message data byte 45 register 19	C1MDATA4519	R/W			√	undefined
03FEC964H	CAN1 message data byte 4 register 19	C1MDATA419	R/W		√		undefined
03FEC965H	CAN1 message data byte 5 register 19	C1MDATA519	R/W		√		undefined
03FEC966H	CAN1 message data byte 67 register 19	C1MDATA6719	R/W			√	undefined
03FEC966H	CAN1 message data byte 6 register 19	C1MDATA619	R/W		√		undefined
03FEC967H	CAN1 message data byte 7 register 19	C1MDATA719	R/W		√		undefined
03FEC968H	CAN1 message data length code register 19	C1MDLC19	R/W		√		0000xxxxB

Table 16-16: Register Access Type (26/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC969H	CAN1 message configuration register 19	C1MCONF19	R/W		√		undefined
03FEC96AH	CAN1 message ID register 19	C1MIDL19	R/W			√	undefined
03FEC96CH		C1MIDH19	R/W			√	undefined
03FEC96EH	CAN1 message control register 19	C1MCTRL19	R/W			√	00x00000 000xx000B
03FEC980H	CAN1 message data byte 01 register 20	C1MDATA0120	R/W			√	undefined
03FEC980H	CAN1 message data byte 0 register 20	C1MDATA020	R/W		√		undefined
03FEC981H	CAN1 message data byte 1 register 20	C1MDATA120	R/W		√		undefined
03FEC982H	CAN1 message data byte 23 register 20	C1MDATA2320	R/W			√	undefined
03FEC982H	CAN1 message data byte 2 register 20	C1MDATA220	R/W		√		undefined
03FEC983H	CAN1 message data byte 3 register 20	C1MDATA320	R/W		√		undefined
03FEC984H	CAN1 message data byte 45 register 20	C1MDATA4520	R/W			√	undefined
03FEC984H	CAN1 message data byte 4 register 20	C1MDATA420	R/W		√		undefined
03FEC985H	CAN1 message data byte 5 register 20	C1MDATA520	R/W		√		undefined
03FEC986H	CAN1 message data byte 67 register 20	C1MDATA6720	R/W			√	undefined
03FEC986H	CAN1 message data byte 6 register 20	C1MDATA620	R/W		√		undefined
03FEC987H	CAN1 message data byte 7 register 20	C1MDATA720	R/W		√		undefined
03FEC988H	CAN1 message data length code register 20	C1MDLC20	R/W		√		0000xxxxB
03FEC989H	CAN1 message configuration register 20	C1MCONF20	R/W		√		undefined
03FEC98AH	CAN1 message ID register 20	C1MIDL20	R/W			√	undefined
03FEC98CH		C1MIDH20	R/W			√	undefined
03FEC98EH	CAN1 message control register 20	C1MCTRL20	R/W			√	00x00000 000xx000B
03FEC9A0H	CAN1 message data byte 01 register 21	C1MDATA0121	R/W			√	undefined
03FEC9A0H	CAN1 message data byte 0 register 21	C1MDATA021	R/W		√		undefined
03FEC9A1H	CAN1 message data byte 1 register 21	C1MDATA121	R/W		√		undefined
03FEC9A2H	CAN1 message data byte 23 register 21	C1MDATA2321	R/W			√	undefined
03FEC9A2H	CAN1 message data byte 2 register 21	C1MDATA221	R/W		√		undefined
03FEC9A3H	CAN1 message data byte 3 register 21	C1MDATA321	R/W		√		undefined
03FEC9A4H	CAN1 message data byte 45 register 21	C1MDATA4521	R/W			√	undefined
03FEC9A4H	CAN1 message data byte 4 register 21	C1MDATA421	R/W		√		undefined
03FEC9A5H	CAN1 message data byte 5 register 21	C1MDATA521	R/W		√		undefined
03FEC9A6H	CAN1 message data byte 67 register 21	C1MDATA6721	R/W			√	undefined
03FEC9A6H	CAN1 message data byte 6 register 21	C1MDATA621	R/W		√		undefined
03FEC9A7H	CAN1 message data byte 7 register 21	C1MDATA721	R/W		√		undefined
03FEC9A8H	CAN1 message data length code register 21	C1MDLC21	R/W		√		0000xxxxB
03FEC9A9H	CAN1 message configuration register 21	C1MCONF21	R/W		√		undefined
03FEC9AAH	CAN1 message ID register 21	C1MIDL21	R/W			√	undefined
03FEC9ACH		C1MIDH21	R/W			√	undefined
03FEC9AEH	CAN1 message control register 21	C1MCTRL21	R/W			√	00x00000 000xx000B

Table 16-16: Register Access Type (27/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FEC9C0H	CAN1 message data byte 01 register 22	C1MDATA0122	R/W			√	undefined
03FEC9C0H	CAN1 message data byte 0 register 22	C1MDATA022	R/W		√		undefined
03FEC9C1H	CAN1 message data byte 1 register 22	C1MDATA122	R/W		√		undefined
03FEC9C2H	CAN1 message data byte 23 register 22	C1MDATA2322	R/W			√	undefined
03FEC9C2H	CAN1 message data byte 2 register 22	C1MDATA222	R/W		√		undefined
03FEC9C3H	CAN1 message data byte 3 register 22	C1MDATA322	R/W		√		undefined
03FEC9C4H	CAN1 message data byte 45 register 22	C1MDATA4522	R/W			√	undefined
03FEC9C4H	CAN1 message data byte 4 register 22	C1MDATA422	R/W		√		undefined
03FEC9C5H	CAN1 message data byte 5 register 22	C1MDATA522	R/W		√		undefined
03FEC9C6H	CAN1 message data byte 67 register 22	C1MDATA6722	R/W			√	undefined
03FEC9C6H	CAN1 message data byte 6 register 22	C1MDATA622	R/W		√		undefined
03FEC9C7H	CAN1 message data byte 7 register 22	C1MDATA722	R/W		√		undefined
03FEC9C8H	CAN1 message data length code register 22	C1MDLC22	R/W		√		0000xxxxB
03FEC9C9H	CAN1 message configuration register 22	C1MCONF22	R/W		√		undefined
03FEC9CAH	CAN1 message ID register 22	C1MIDL22	R/W			√	undefined
03FEC9CCH		C1MIDH22	R/W			√	undefined
03FEC9CEH	CAN1 message control register 22	C1MCTRL22	R/W			√	00x00000 000xx000B
03FEC9E0H	CAN1 message data byte 01 register 23	C1MDATA0123	R/W			√	undefined
03FEC9E0H	CAN1 message data byte 0 register 23	C1MDATA023	R/W		√		undefined
03FEC9E1H	CAN1 message data byte 1 register 23	C1MDATA123	R/W		√		undefined
03FEC9E2H	CAN1 message data byte 23 register 23	C1MDATA2323	R/W			√	undefined
03FEC9E2H	CAN1 message data byte 2 register 23	C1MDATA223	R/W		√		undefined
03FEC9E3H	CAN1 message data byte 3 register 23	C1MDATA323	R/W		√		undefined
03FEC9E4H	CAN1 message data byte 45 register 23	C1MDATA4523	R/W			√	undefined
03FEC9E4H	CAN1 message data byte 4 register 23	C1MDATA423	R/W		√		undefined
03FEC9E5H	CAN1 message data byte 5 register 23	C1MDATA523	R/W		√		undefined
03FEC9E6H	CAN1 message data byte 67 register 23	C1MDATA6723	R/W			√	undefined
03FEC9E6H	CAN1 message data byte 6 register 23	C1MDATA623	R/W		√		undefined
03FEC9E7H	CAN1 message data byte 7 register 23	C1MDATA723	R/W		√		undefined
03FEC9E8H	CAN1 message data length code register 23	C1MDLC23	R/W		√		0000xxxxB
03FEC9E9H	CAN1 message configuration register 23	C1MCONF23	R/W		√		undefined
03FEC9EAH	CAN1 message ID register 23	C1MIDL23	R/W			√	undefined
03FEC9ECH		C1MIDH23	R/W			√	undefined
03FEC9EEH	CAN1 message control register 23	C1MCTRL23	R/W			√	00x00000 000xx000B
03FECA00H	CAN1 message data byte 01 register 24	C1MDATA0124	R/W			√	undefined
03FECA00H	CAN1 message data byte 0 register 24	C1MDATA024	R/W		√		undefined
03FECA01H	CAN1 message data byte 1 register 24	C1MDATA124	R/W		√		undefined

Table 16-16: Register Access Type (28/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FECA02H	CAN1 message data byte 23 register 24	C1MDATA2324	R/W			√	undefined
03FECA02H	CAN1 message data byte 2 register 24	C1MDATA224	R/W		√		undefined
03FECA03H	CAN1 message data byte 3 register 24	C1MDATA324	R/W		√		undefined
03FECA04H	CAN1 message data byte 45 register 24	C1MDATA4524	R/W			√	undefined
03FECA04H	CAN1 message data byte 4 register 24	C1MDATA424	R/W		√		undefined
03FECA05H	CAN1 message data byte 5 register 24	C1MDATA524	R/W		√		undefined
03FECA06H	CAN1 message data byte 67 register 24	C1MDATA6724	R/W			√	undefined
03FECA06H	CAN1 message data byte 6 register 24	C1MDATA624	R/W		√		undefined
03FECA07H	CAN1 message data byte 7 register 24	C1MDATA724	R/W		√		undefined
03FECA08H	CAN1 message data length code register 24	C1MDLC24	R/W		√		0000xxxxB
03FECA09H	CAN1 message configuration register 24	C1MCONF24	R/W		√		undefined
03FECA0AH	CAN1 message ID register 24	C1MIDL24	R/W			√	undefined
03FECA0CH		C1MIDH24	R/W			√	undefined
03FECA0EH	CAN1 message control register 24	C1MCTRL24	R/W			√	00x00000 000xx000B
03FECA20H	CAN1 message data byte 01 register 25	C1MDATA0125	R/W			√	undefined
03FECA20H	CAN1 message data byte 0 register 25	C1MDATA025	R/W		√		undefined
03FECA21H	CAN1 message data byte 1 register 25	C1MDATA125	R/W		√		undefined
03FECA22H	CAN1 message data byte 23 register 25	C1MDATA2325	R/W			√	undefined
03FECA22H	CAN1 message data byte 2 register 25	C1MDATA225	R/W		√		undefined
03FECA23H	CAN1 message data byte 3 register 25	C1MDATA325	R/W		√		undefined
03FECA24H	CAN1 message data byte 45 register 25	C1MDATA4525	R/W			√	undefined
03FECA24H	CAN1 message data byte 4 register 25	C1MDATA425	R/W		√		undefined
03FECA25H	CAN1 message data byte 5 register 25	C1MDATA525	R/W		√		undefined
03FECA26H	CAN1 message data byte 67 register 25	C1MDATA6725	R/W			√	undefined
03FECA26H	CAN1 message data byte 6 register 25	C1MDATA625	R/W		√		undefined
03FECA27H	CAN1 message data byte 7 register 25	C1MDATA725	R/W		√		undefined
03FECA28H	CAN1 message data length code register 25	C1MDLC25	R/W		√		0000xxxxB
03FECA29H	CAN1 message configuration register 25	C1MCONF25	R/W		√		undefined
03FECA2AH	CAN1 message ID register 25	C1MIDL25	R/W			√	undefined
03FECA2CH		C1MIDH25	R/W			√	undefined
03FECA2EH	CAN1 message control register 25	C1MCTRL25	R/W			√	00x00000 000xx000B
03FECA40H	CAN1 message data byte 01 register 26	C1MDATA0126	R/W			√	undefined
03FECA40H	CAN1 message data byte 0 register 26	C1MDATA026	R/W		√		undefined
03FECA41H	CAN1 message data byte 1 register 26	C1MDATA126	R/W		√		undefined
03FECA42H	CAN1 message data byte 23 register 26	C1MDATA2326	R/W			√	undefined
03FECA42H	CAN1 message data byte 2 register 26	C1MDATA226	R/W		√		undefined
03FECA43H	CAN1 message data byte 3 register 26	C1MDATA326	R/W		√		undefined

Table 16-16: Register Access Type (29/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FECA44H	CAN1 message data byte 45 register 26	C1MDATA4526	R/W			√	undefined
03FECA44H	CAN1 message data byte 4 register 26	C1MDATA426	R/W		√		undefined
03FECA45H	CAN1 message data byte 5 register 26	C1MDATA526	R/W		√		undefined
03FECA46H	CAN1 message data byte 67 register 26	C1MDATA6726	R/W			√	undefined
03FECA46H	CAN1 message data byte 6 register 26	C1MDATA626	R/W		√		undefined
03FECA47H	CAN1 message data byte 7 register 26	C1MDATA726	R/W		√		undefined
03FECA48H	CAN1 message data length code register 26	C1MDLC26	R/W		√		0000xxxxB
03FECA49H	CAN1 message configuration register 26	C1MCONF26	R/W		√		undefined
03FECA4AH	CAN1 message ID register 26	C1MIDL26	R/W			√	undefined
03FECA4CH		C1MIDH26	R/W			√	undefined
03FECA4EH	CAN1 message control register 26	C1MCTRL26	R/W			√	00x00000 000xx000B
03FECA60H	CAN1 message data byte 01 register 27	C1MDATA0127	R/W			√	undefined
03FECA60H	CAN1 message data byte 0 register 27	C1MDATA027	R/W		√		undefined
03FECA61H	CAN1 message data byte 1 register 27	C1MDATA127	R/W		√		undefined
03FECA62H	CAN1 message data byte 23 register 27	C1MDATA2327	R/W			√	undefined
03FECA62H	CAN1 message data byte 2 register 27	C1MDATA227	R/W		√		undefined
03FECA63H	CAN1 message data byte 3 register 27	C1MDATA327	R/W		√		undefined
03FECA64H	CAN1 message data byte 45 register 27	C1MDATA4527	R/W			√	undefined
03FECA64H	CAN1 message data byte 4 register 27	C1MDATA427	R/W		√		undefined
03FECA65H	CAN1 message data byte 5 register 27	C1MDATA527	R/W		√		undefined
03FECA66H	CAN1 message data byte 67 register 27	C1MDATA6727	R/W			√	undefined
03FECA66H	CAN1 message data byte 6 register 27	C1MDATA627	R/W		√		undefined
03FECA67H	CAN1 message data byte 7 register 27	C1MDATA727	R/W		√		undefined
03FECA68H	CAN1 message data length code register 27	C1MDLC27	R/W		√		0000xxxxB
03FECA69H	CAN1 message configuration register 27	C1MCONF27	R/W		√		undefined
03FECA6AH	CAN1 message ID register 27	C1MIDL27	R/W			√	undefined
03FECA6CH		C1MIDH27	R/W			√	undefined
03FECA6EH	CAN1 message control register 27	C1MCTRL27	R/W			√	00x00000 000xx000B
03FECA80H	CAN1 message data byte 01 register 28	C1MDATA0128	R/W			√	undefined
03FECA80H	CAN1 message data byte 0 register 28	C1MDATA028	R/W		√		undefined
03FECA81H	CAN1 message data byte 1 register 28	C1MDATA128	R/W		√		undefined
03FECA82H	CAN1 message data byte 23 register 28	C1MDATA2328	R/W			√	undefined
03FECA82H	CAN1 message data byte 2 register 28	C1MDATA228	R/W		√		undefined
03FECA83H	CAN1 message data byte 3 register 28	C1MDATA328	R/W		√		undefined
03FECA84H	CAN1 message data byte 45 register 28	C1MDATA4528	R/W			√	undefined
03FECA84H	CAN1 message data byte 4 register 28	C1MDATA428	R/W		√		undefined
03FECA85H	CAN1 message data byte 5 register 28	C1MDATA528	R/W		√		undefined

Table 16-16: Register Access Type (30/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FECA86H	CAN1 message data byte 67 register 28	C1MDATA6728	R/W			√	undefined
03FECA86H	CAN1 message data byte 6 register 28	C1MDATA628	R/W		√		undefined
03FECA87H	CAN1 message data byte 7 register 28	C1MDATA728	R/W		√		undefined
03FECA88H	CAN1 message data length code register 28	C1MDLC28	R/W		√		0000xxxxB
03FECA89H	CAN1 message configuration register 28	C1MCONF28	R/W		√		undefined
03FECA8AH	CAN1 message ID register 28	C1MIDL28	R/W			√	undefined
03FECA8CH		C1MIDH28	R/W			√	undefined
03FECA8EH	CAN1 message control register 28	C1MCTRL28	R/W			√	00x00000 000xx000B
03FECAA0H	CAN1 message data byte 01 register 29	C1MDATA0129	R/W			√	undefined
03FECAA0H	CAN1 message data byte 0 register 29	C1MDATA029	R/W		√		undefined
03FECAA1H	CAN1 message data byte 1 register 29	C1MDATA129	R/W		√		undefined
03FECAA2H	CAN1 message data byte 23 register 29	C1MDATA2329	R/W			√	undefined
03FECAA2H	CAN1 message data byte 2 register 29	C1MDATA229	R/W		√		undefined
03FECAA3H	CAN1 message data byte 3 register 29	C1MDATA329	R/W		√		undefined
03FECAA4H	CAN1 message data byte 45 register 29	C1MDATA4529	R/W			√	undefined
03FECAA4H	CAN1 message data byte 4 register 29	C1MDATA429	R/W		√		undefined
03FECAA5H	CAN1 message data byte 5 register 29	C1MDATA529	R/W		√		undefined
03FECAA6H	CAN1 message data byte 67 register 29	C1MDATA6729	R/W			√	undefined
03FECAA6H	CAN1 message data byte 6 register 29	C1MDATA629	R/W		√		undefined
03FECAA7H	CAN1 message data byte 7 register 29	C1MDATA729	R/W		√		undefined
03FECAA8H	CAN1 message data length code register 29	C1MDLC29	R/W		√		0000xxxxB
03FECAA9H	CAN1 message configuration register 29	C1MCONF29	R/W		√		undefined
03FECAAAH	CAN1 message ID register 29	C1MIDL29	R/W			√	undefined
03FECAACH		C1MIDH29	R/W			√	undefined
03FECAAEH	CAN1 message control register 29	C1MCTRL29	R/W			√	00x00000 000xx000B
03FECAC0H	CAN1 message data byte 01 register 30	C1MDATA0130	R/W			√	undefined
03FECAC0H	CAN1 message data byte 0 register 30	C1MDATA030	R/W		√		undefined
03FECAC1H	CAN1 message data byte 1 register 30	C1MDATA130	R/W		√		undefined
03FECAC2H	CAN1 message data byte 23 register 30	C1MDATA2330	R/W			√	undefined
03FECAC2H	CAN1 message data byte 2 register 30	C1MDATA230	R/W		√		undefined
03FECAC3H	CAN1 message data byte 3 register 30	C1MDATA330	R/W		√		undefined
03FECAC4H	CAN1 message data byte 45 register 30	C1MDATA4530	R/W			√	undefined
03FECAC4H	CAN1 message data byte 4 register 30	C1MDATA430	R/W		√		undefined
03FECAC5H	CAN1 message data byte 5 register 30	C1MDATA530	R/W		√		undefined
03FECAC6H	CAN1 message data byte 67 register 30	C1MDATA6730	R/W			√	undefined
03FECAC6H	CAN1 message data byte 6 register 30	C1MDATA630	R/W		√		undefined
03FECAC7H	CAN1 message data byte 7 register 30	C1MDATA730	R/W		√		undefined
03FECAC8H	CAN1 message data length code register 30	C1MDLC30	R/W		√		0000xxxxB

Table 16-16: Register Access Type (31/31)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1-bit	8-bit	16-bit	
03FECAC9H	CAN1 message configuration register 30	C1MCONF30	R/W		√		undefined
03FECACAH	CAN1 message ID register 30	C1MIDL30	R/W			√	undefined
03FECACCH		C1MIDH30	R/W			√	undefined
03FECACEH	CAN1 message control register 30	C1MCTRL30	R/W			√	00x00000 000xx000B
03FECAE0H	CAN1 message data byte 01 register 31	C1MDATA0131	R/W			√	undefined
03FECAE0H	CAN1 message data byte 0 register 31	C1MDATA031	R/W		√		undefined
03FECAE1H	CAN1 message data byte 1 register 31	C1MDATA131	R/W		√		undefined
03FECAE2H	CAN1 message data byte 23 register 31	C1MDATA2331	R/W			√	undefined
03FECAE2H	CAN1 message data byte 2 register 31	C1MDATA231	R/W		√		undefined
03FECAE3H	CAN1 message data byte 3 register 31	C1MDATA331	R/W		√		undefined
03FECAE4H	CAN1 message data byte 45 register 31	C1MDATA4531	R/W			√	undefined
03FECAE4H	CAN1 message data byte 4 register 31	C1MDATA431	R/W		√		undefined
03FECAE5H	CAN1 message data byte 5 register 31	C1MDATA531	R/W		√		undefined
03FECAE6H	CAN1 message data byte 67 register 31	C1MDATA6731	R/W			√	undefined
03FECAE6H	CAN1 message data byte 6 register 31	C1MDATA631	R/W		√		undefined
03FECAE7H	CAN1 message data byte 7 register 31	C1MDATA731	R/W		√		undefined
03FECAE8H	CAN1 message data length code register 31	C1MDLC31	R/W		√		0000xxxx
03FECAE9H	CAN1 message configuration register 31	C1MCONF31	R/W		√		undefined
03FECAEAH	CAN1 message ID register 31	C1MIDL31	R/W			√	undefined
03FECAECH		C1MIDH31	R/W			√	undefined
03FECAEEH	CAN1 message control register 31	C1MCTRL31	R/W			√	00x00000 000xx000B

Table 16-17: CAN Module Register Bit Configuration (1/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
03FExx40H	CnMASK1L	CM1ID[7:0]							
03FExx41H		CM1ID[15:8]							
03FExx42H	CnMASK1H	CM1ID[23:16]							
03FExx43H		0	0	0	CM1ID[28:24]				
03FExx44H	CnMASK2L	CM2ID[7:0]							
03FExx45H		CM2ID[15:8]							
03FExx46H	CnMASK2H	CM2ID[23:16]							
03FExx47H		0	0	0	CM2ID[28:24]				
03FExx48H	CnMASK3L	CM3ID[7:0]							
03FExx49H		CM3ID[15:8]							
03FExx4AH	CnMASK3H	CM3ID[23:16]							
03FExx4BH		0	0	0	CM3ID[28:24]				
03FExx4CH	CnMASK4L	CM4ID[7:0]							
03FExx4DH		CM4ID[15:8]							
03FExx4EH	CnMASK4H	CM14D[23:16]							
03FExx4FH		0	0	0	CM4ID[28:24]				
03FExx50H	CnCTRL (W)	0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0
03FExx51H		Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
03FExx50H	CnCTRL (R)	CCERC	AL	VALID	PS MODE1	PS MODE0	OP MODE2	OP MODE1	OP MODE0
03FExx51H		0	0	0	0	0	0	RSTAT	TSTAT
03FExx52H	CnLEC (W)	0	0	0	0	0	0	0	0
03FExx52H	CnLEC (R)	0	0	0	0	0	LEC2	LEC1	LEC0
03FExx53H	CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
03FExx54H	CnERC	TEC[7:0]							
03FExx55H		REPS	REC[6:0]						
03FExx56H	CnIE (W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
03FExx57H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
03FExx56H	CnIE (R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
03FExx57H		0	0	0	0	0	0	0	0
03FExx58H	CnINTS (W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
03FExx59H		0	0	0	0	0	0	0	0
03FExx58H	CnINTS (R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
03FExx59H		0	0	0	0	0	0	0	0
03FExx5AH	CnBRP	TQPRS[7:0]							

Table 16-17: CAN Module Register Bit Configuration (2/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
03FExx5CH	CnBTR	0	0	0	0	TSEG1[3:0]			
03FExx5DH		0	0	SJW[1:0]		0	TSEG2[2:0]		
03FExx5EH	CnLIPT	LIPT[7:0]							
03FExx60H	CnRGPT (W)	0	0	0	0	0	0	0	Clear ROVF
03FExx61H		0	0	0	0	0	0	0	0
03FExx60H	CnRGPT (R)	0	0	0	0	0	0	RHPM	ROVF
03FExx61H		RGPT[7:0]							
03FExx62H	CnLOPT	LOPT[7:0]							
03FExx64H	CnTGPT (W)	0	0	0	0	0	0	0	Clear TOVF
03FExx65H		0	0	0	0	0	0	0	0
03FExx64H	CnTGPT (R)	0	0	0	0	0	0	THPM	TOVF
03FExx65H		TGPT[7:0]							
03FExx66H	CnTS (W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
03FExx67H		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
03FExx66H	CnTS (R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
03FExx67H		0	0	0	0	0	0	0	0
03FExx68H to 03FExxFFH	-	Access prohibited (reserved for future use)							

Remark: n = 0, 1

16.5.3 Register bit configuration

Table 16-18: CAN Global Register Bit Configuration

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
03FExx00H	CnGMCTRL (W)	0	0	0	0	0	0	0	Clear GOM
03FExx01H		0	0	0	0	0	0	Set EFSD	Set GOM
03FExx00H	CnGMCTRL (R)	0	0	0	0	0	0	EFSD	GOM
03FExx01H		MBON	0	0	0	0	0	0	0
03FExx02H	CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
03FExx06H	CnGMABT (W)	0	0	0	0	0	0	0	Clear ABTTRG
03FExx07H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
03FExx06H	CnGMABT (R)	0	0	0	0	0	0	ABTCLR	ABTTRG
03FExx07H		0	0	0	0	0	0	0	0
03FExx08H	CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

Remark: n = 0, 1

Table 16-19: Message Buffer Register Bit Configuration

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
03FExxx0H	CnMDATA01m	Message data (byte 0)							
03FExxx1H		Message data (byte 1)							
03FExxx0H	CnMDATA0m	Message data (byte 0)							
03FExxx1H	CnMDATA1m	Message data (byte 1)							
03FExxx2H	CnMDATA23m	Message data (byte 2)							
03FExxx3H		Message data (byte 3)							
03FExxx2H	CnMDATA2m	Message data (byte 2)							
03FExxx3H	CnMDATA3m	Message data (byte 3)							
03FExxx4H	CnMDATA45m	Message data (byte 4)							
03FExxx5H		Message data (byte 5)							
03FExxx4H	CnMDATA4m	Message data (byte 4)							
03FExxx5H	CnMDATA5m	Message data (byte 5)							
03FExxx6H	CnMDATA67m	Message data (byte 6)							
03FExxx7H		Message data (byte 7)							
03FExxx6H	CnMDATA6m	Message data (byte 6)							
03FExxx7H	CnMDATA7m	Message data (byte 7)							
03FExxx8H	CnMDLCm	0				MDLC3	MDLC2	MDLC1	MDLC0
03FExxx9H	CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
03FExxxAH	CnMIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
03FExxxBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
03FExxxCH	CnMIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
03FExxxDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
03FExxxEH	CnMCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
03FExxxFH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
03FExxxEH	CnMCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
03FExxxFH		0	0	MUC	0	0	0	0	0
03FExxx0 to 03FExxxFH	-	Access prohibited (reserved for future use)							

Remark: n = 0, 1
m = 0 to 31

16.6 Control Registers

(1) CAN global control register (CnGMCTRL)

The CnGMCTRL register is used to control the operation of the CAN module.

Figure 16-23: CAN Global Control Register (CnGMCTRL) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnGMCTRL	MBON	0	0	0	0	0	0	0	see Table 16-4	R/W	0000H
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	EFSD	GOM			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnGMCTRL	0	0	0	0	0	0	Set EFSD	Set GOM			
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	0	Clear GOM			

(a) Read

MBON	Bit enabling access to message buffer register, transmit/receive history registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

Cautions: 1. While the MBON bit is cleared (to 0), software access to the message buffers (CnMDATA0m, CnMDATA1m, CnMDATA01m, CnMDATA2m, CnMDATA3m, CnMDATA23m, CnMDATA4m, CnMDATA5m, CnMDATA45m, CnMDATA6m, CnMDATA7m, CnMDATA67m, CnMDLCm, CnMCONFm, CnMIDLm, CnMIDHm, and CnMCTRLm), or registers related to transmit history or receive history (CnLOPT, CnTGPT, CnLIPT, and CnRGPT) is disabled.

2. This bit is read-only. Even if 1 is written to MBON while it is 0, the value of MBON does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Remark: When the CAN sleep mode/CAN stop mode is entered, or when the GOM bit is cleared to 0, the MBON bit is cleared to 0.
When the CAN sleep mode/CAN stop mode is released, or when the GOM bit is set to 1, the MBON bit is set to 1.

Figure 16-23: CAN Global Control Register (CnGMCTRL) Format (2/2)

EFSD	Bit enabling forced shut down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

Caution: To request forced shut down, the GOM bit must be cleared to 0 immediately after the EFSD bit has been set to 1. If access to another register (including reading the CnGMCTRL register) is executed without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shut down request is invalid.

GOM	Global operation mode bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

Caution: The GOM bit is cleared to 0 only in the initialization mode or immediately after the EFSD bit is set to 1.

(b) Write

Set EFSD	EFSD bit setting
0	No change in EFSD bit.
1	EFSD bit set to 1.

Set GOM	Clear GOM	GOM bit setting
0	1	GOM bit cleared to 0.
1	0	GOM bit set to 1.
Other than above		No change in GOM bit.

(2) CAN global clock selection register (CnGMCS)

The CnGMCS register is used to select the CAN module system clock.

Figure 16-24: CAN Global Clock Selection Register (CnGMCS) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0	see Table 16-4	R/W	0FH

CCP3	CCP2	CCP1	CCP1	CAN module system clock (f_{CANMOD})
0	0	0	0	$f_{CAN}/1$
0	0	0	1	$f_{CAN}/2$
0	0	1	0	$f_{CAN}/3$
0	0	1	1	$f_{CAN}/4$
0	1	0	0	$f_{CAN}/5$
0	1	0	1	$f_{CAN}/6$
0	1	1	0	$f_{CAN}/7$
0	1	1	1	$f_{CAN}/8$
1	0	0	0	$f_{CAN}/9$
1	0	0	1	$f_{CAN}/10$
1	0	1	0	$f_{CAN}/11$
1	0	1	1	$f_{CAN}/12$
1	1	0	0	$f_{CAN}/13$
1	1	0	1	$f_{CAN}/14$
1	1	1	0	$f_{CAN}/15$
1	1	1	1	$f_{CAN}/16$ (Default value)

Remark: f_{CAN} = Clock supplied to CAN = f_{XX}

(3) CAN global automatic block transmission control register (CnGMABT)

The CnGMABT register is used to control the automatic block transmission (ABT) operation.

Figure 16-25: CAN Global Automatic Block Transmission Control Register (CnGMABT) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnGMABT	0	0	0	0	0	0	0	0	see Table 16-4	R/W	0000H
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	ABTCLR	ABTTRG			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnGMABT	0	0	0	0	0	0	Set ABTCLR	Set ABTTRG			
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	0	Clear ABTTRG			

Caution: Before changing the normal operation mode with ABT to the initialization mode, be sure to set the CnGMABT register to the default value (0000H). After setting, confirm that the CnGMABT register is initialized to 0000H.

(a) Read

ABTCLR	Automatic block transmission engine clear status bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Remarks: **1.** Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0.
The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.

2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic block transmission status bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Cautions: 1. Do not set the ABTTRG bit to 1 in the initialization mode. If the ABTTRG bit is set to 1 in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.

2. Do not set the ABTTRG bit to 1 while the CnCTRL.TSTATbit is set to 1. Confirm that the TSTAT bit = 0 before setting the ABTTRG bit to 1.

Figure 16-25: CAN Global Automatic Block Transmission Control Register (CnGMABT) Format (2/2)

(b) Write

Set ABTCLR	Automatic block transmission engine clear request bit
0	The automatic block transmission engine is in idle state or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic block transmission start bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

(4) CAN global automatic block transmission delay register (CnGMABTD)

The CnGMABTD register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

Figure 16-26: CAN Global Automatic Block Transmission Delay Register (CnGMABTD) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0	see Table 16-16	R/W	00H

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission (Unit: Data bit time (DBT))
0	0	0	0	0 DBT (default value)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

Cautions: 1. Do not change the contents of the CnGMABTD register while the ABTTRG bit is set to 1.

2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to 31) is made.

(5) CAN module mask control register (CnMASKaL, CnMASKaH) (a = 1, 2, 3, or 4)

The CnMASKaL and CnMASKaH registers are used to extend the number of receivable messages by masking part of the identifier (ID) of a message and invalidating the ID of the masked part.

Figure 16-27: CAN Module Mask Control Register (CnMASKaL, CnMASKaH) (a = 1, 2, 3, or 4) Format (1/2)

- **CANn module mask 1 register (CnMASK1L, CnMASK1H)**

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0			
	15	14	13	12	11	10	9	8			
CnMASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24			
	7	6	5	4	3	2	1	0			
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16			

- **CANn module mask 2 register (CnMASK2L, CnMASK2H)**

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0			
	15	14	13	12	11	10	9	8			
CnMASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24			
	7	6	5	4	3	2	1	0			
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16			

- **CANn module mask 3 register (CnMASK3L, CnMASK3H)**

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0			
	15	14	13	12	11	10	9	8			
CnMASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24			
	7	6	5	4	3	2	1	0			
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16			

Figure 16-27: CAN Module Mask Control Register (CnMASKaL, CnMASKaH) (a = 1, 2, 3, or 4) Format (2/2)

- **CANn module mask 4 register (CnMASK4L, CnMASK4H)**

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0			
	15	14	13	12	11	10	9	8			
CnMASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24			
	7	6	5	4	3	2	1	0			
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16			

CMID28 to CMID0	Mask pattern setting of ID bit
0	The ID bits of the message buffer set by the CMID28 to CMID0 bits are compared with the ID bits of the received message frame.
1	The ID bits of the message buffer set by the CMID28 to CMID0 bits are not compared with the ID bits of the received message frame (they are masked).

Remark: Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(6) CAN module control register (CnCTRL)

The CnCTRL register is used to control the operation mode of the CAN module.

Figure 16-28: CAN Module Control Register (CnCTRL) Format (1/4)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnCTRL	0	0	0	0	0	0	RSTAT	TSTAT		R/W	0000H
	7	6	5	4	3	2	1	0	see Table 16-16		
	CCERC	AL	VALID	PSMODE 1	PSMODE 0	OPMODE 2	OPMODE 1	OPMODE 0			

(b)Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnCTRL	Set CCERC	Set AL	0	Set PSMODE 1	Set PSMODE 0	Set OPMODE 2	Set OPMODE 1	Set OPMODE 0			
	7	6	5	4	3	2	1	0			
	0	Clear AL	Clear VALID	Clear PSMODE 1	Clear PSMODE 0	Clear OPMODE 2	Clear OPMODE 1	Clear OPMODE 0			

(a) Read

RSTAT	Reception status bit
0	Reception is stopped.
1	Reception is in progress.

Remarks: 1. The RSTAT bit is set to 1 under the following conditions (timing):
- The SOF bit of a receive frame is detected
- On occurrence of arbitration loss during a transmit frame

2. The RSTAT bit is cleared to 0 under the following conditions (timing):
- When a recessive level is detected at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

Figure 16-28: CAN Module Control Register (CnCTRL) Format (2/4)

TSTAT	Transmission status bit
0	Transmission is stopped.
1	Transmission is in progress.

Remarks: 1. The TSTAT bit is set to 1 under the following conditions (timing):
- The SOF bit of a transmit frame is detected
- The first bit of an error flag is detected during a transmit frame

2. The TSTAT bit is cleared to 0 under the following conditions (timing):
- During transition to bus-off state
- On occurrence of arbitration loss in transmit frame
- On detection of recessive level at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

CCERC	Error counter clear bit
0	The CnERC and CnINFO registers are not cleared in the initialization mode.
1	The CnERC and CnINFO registers are cleared in the initialization mode.

Remarks: 1. The CCERC bit is used to clear the CnERC and CnINFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.

2. When the CnERC and CnINFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.

3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.

4. If the CCERC bit is set to 1 immediately after the INIT mode is entered in the self test mode, the receive data may be corrupted.

AL	Bit to set operation in case of arbitration loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Remark: The AL bit is valid only in the single-shot mode.

Figure 16-28: CAN Module Control Register (CnCTRL) Format (3/4)

VALID	Valid receive message frame detection bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

Remarks: 1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).

2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.

3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the reception mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state.

4. The VALID bit is read-only in the CAN sleep mode or CAN stop mode.

5. To clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

PSMODE1	PSMODE0	Power save mode
0	0	No power save mode is selected.
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

Caution: Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.

OPMODE2	OPMODE1	OPMODE0	Operation mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode).
0	0	1	Normal operation mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited

Remark: The OPMODE[2:0] bits are read-only in the CAN sleep mode or CAN stop mode.

(b) Write

Set CCERC	Setting of CCERC bit
1	CCERC bit is set to 1.
Other than above	CCERC bit is not changed.

Figure 16-28: CAN Module Control Register (CnCTRL) Format (4/4)

(b) Write

Set AL	Clear AL	Setting of AL bit
0	1	AL bit is cleared to 0.
1	0	AL bit is set to 1.
Other than above		AL bit is not changed.

Clear VALID	Setting of VALID bit
0	VALID bit is not changed.
1	VALID bit is cleared to 0.

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 bit
0	1	PSMODE0 bit is cleared to 0.
1	0	PSMODE bit is set to 1.
Other than above		PSMODE0 bit is not changed.

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 bit
0	1	PSMODE1 bit is cleared to 0.
1	0	PSMODE1 bit is set to 1.
Other than above		PSMODE1 bit is not changed.

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 bit
0	1	OPMODE0 bit is cleared to 0.
1	0	OPMODE0 bit is set to 1.
Other than above		OPMODE0 bit is not changed.

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 bit
0	1	OPMODE1 bit is cleared to 0.
1	0	OPMODE1 bit is set to 1.
Other than above		OPMODE1 bit is not changed.

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 bit
0	1	OPMODE2 bit is cleared to 0.
1	0	OPMODE2 bit is set to 1.
Other than above		OPMODE2 bit is not changed.

(7) CAN module last error information register (CnLEC)

The CnLEC register provides the error information of the CAN protocol.

Figure 16-29: CAN Module Last Error Information Register (CnLEC) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnLEC	0	0	0	0	0	LEC2	LEC1	LEC0	see Table 16-16	R/W	00H

Remarks: 1. The contents of the CnLEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.

2. If an attempt is made to write a value other than 00H to the CnLEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN protocol error information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error. (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error. (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	undefined

(8) CAN module information register (CnINFO)

The CnINFO register indicates the status of the CAN module.

Figure 16-30: CAN Module Information Register (CnINFO) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0	see Table 16-16	R/W	00H

BOFF	Bus-off status bit
0	Not bus-off state (transmit error counter £ 255). (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255). (The value of the transmit counter is 256 or more.)

TECS1	TECS0	Transmission error counter status bit
0	0	The value of the transmission error counter is less than that of the warning level (< 96).
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127).
1	0	undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off state (≥ 128).

RECS1	RECS0	Reception error counter status bit
0	0	The value of the reception error counter is less than that of the warning level (< 96).
0	1	The value of the reception error counter is in the range of the warning level (96 to 127).
1	0	undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128).

(9) CAN module error counter register (CnERC)

The CnERC register indicates the count value of the transmission/reception error counter.

Figure 16-31: CAN Module Error Counter Register (CnERC) Format

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0	see Table 16-16	R	00H
	7	6	5	4	3	2	1	0			
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0			

REPS	Reception error passive status bit
0	Reception error counter is not error passive (< 128)
1	Reception error counter is error passive range (≥ 128)

REC6 to REC0	Reception error counter bit
0 to 127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

Remark: REC7 to REC0 of the reception error counter are invalid in the reception error passive state (RECS[1:0] = 11B).

TEC7 to TEC0	Transmission error counter bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Remark: TEC7 to TEC0 of the transmission error counter are invalid in the bus-off state (BOFF = 1).

(10) CAN module interrupt enable register (CnIE)

The CnIE register is used to enable or disable the interrupts of the CAN module.

Figure 16-32: CAN Module Interrupt Enable Register (CnIE) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnIE	0	0	0	0	0	0	0	0	see Table 16-16	R/W	00H
	7	6	5	4	3	2	1	0			
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnIE	0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0			
	7	6	5	4	3	2	1	0			
	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0			

(a) Read

CIE5 to CIE0	CAN module interrupt enable bit
0	Output of the interrupt corresponding to interrupt status register CINTSx is disabled.
1	Output of the interrupt corresponding to interrupt status register CINTSx is enabled.

(b) Write

Set CIE5	Clear CIE5	Setting of CIE5 bit
0	1	CIE5 bit is cleared to 0.
1	0	CIE5 bit is set to 1.
Other than above		CIE5 bit is not changed.

Set CIE4	Clear CIE4	Setting of CIE4 bit
0	1	CIE4 bit is cleared to 0.
1	0	CIE4 bit is set to 1.
Other than above		CIE4 bit is not changed.

Figure 16-32: CAN Module Interrupt Enable Register (CnIE) Format (2/2)

(b) Write

Set CIE3	Clear CIE3	Setting of CIE3 bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than above		CIE3 bit is not changed.

Set CIE2	Clear CIE2	Setting of CIE2 bit
0	1	CIE2 bit is cleared to 0.
1	0	CIE2 bit is set to 1.
Other than above		CIE2 bit is not changed.

Set CIE1	Clear CIE1	Setting of CIE1 bit
0	1	CIE1 bit is cleared to 0.
1	0	CIE1 bit is set to 1.
Other than above		CIE1 bit is not changed.

Set CIE0	Clear CIE0	Setting of CIE0 bit
0	1	CIE0 bit is cleared to 0.
1	0	CIE0 bit is set to 1.
Other than above		CIE0 bit is not changed.

(11) CAN module interrupt status register (CnINTS)

The CnINTS register indicates the interrupt status of the CAN module.

Figure 16-33: CAN Module Interrupt Status Register (CnINTS) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnINTS	0	0	0	0	0	0	0	0	see Table 16-16	R/W	00H
	7	6	5	4	3	2	1	0			
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnINTS	0	0	0	0	0	0	0	0			
	7	6	5	4	3	2	1	0			
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0			

(a) Read

CINTS5 to CINTS0	CAN interrupt status bit
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt status bit	Related interrupt source event
CINTS5	Wake-up interrupt from CAN sleep mode[Note]
CINTS4	Arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN error status interrupt
CINTS1	Interrupt on completion of reception of valid message frame to message buffer m
CINTS0	Interrupt on normal completion of transmission of message frame from message buffer m

Note: The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.

Figure 16-33: CAN Module Interrupt Status Register (CnINTS) Format (2/2)

(b) Write

Clear CINTS5 to CINTS0	Setting of CINTS5 to CINTS0 bits
0	CINTS5 to CINTS0 bits are not changed.
1	CINTS5 to CINTS0 bits are cleared to 0.

(12) CAN module bit rate prescaler register (CnBRP)

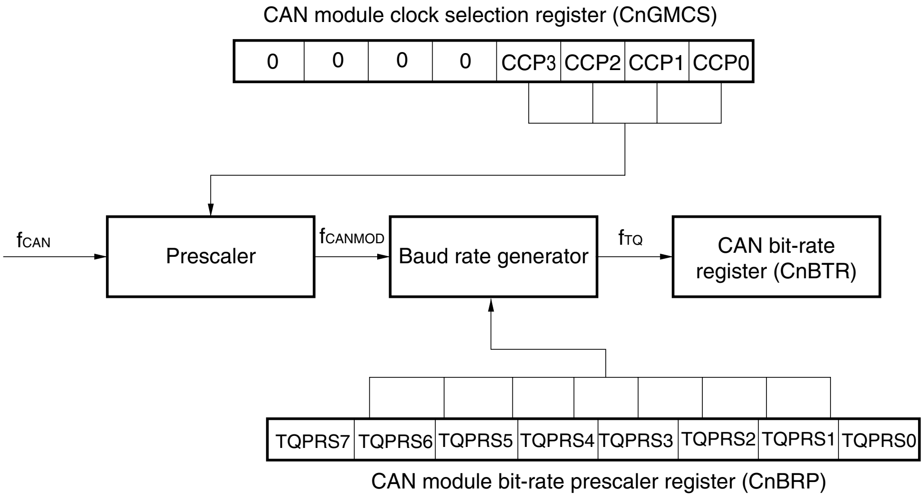
The CnBRP register is used to select the CAN protocol layer base clock (f_{TQ}). The communication baud rate is set to the CnBTR register.

Figure 16-34: CAN Module Bit Rate Prescaler Register (CnBRP) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnBRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0	see Table 16-16	R/W	FFH

TQPRS7 to TQPRS0	CAN protocol layer base system clock (f_{TQ})
0	$f_{CANMOD}/1$
1	$f_{CANMOD}/2$
n	$f_{CANMOD}/(n+1)$
.....	.....
255	$f_{CANMOD}/256$ (default value)

Figure 16-35: CAN Module Clock


CAN module clock selection register (CnGMCS)
0
0
0
0
CCP3
CCP2
CCP1
CCP0
fCAN
Prescaler
fCANMOD
Baud rate generator
fTQ
CAN bit-rate register (CnBTR)
TQPRS7
TQPRS6
TQPRS5
TQPRS4
TQPRS3
TQPRS2
TQPRS1
TQPRS0
CAN module bit-rate prescaler register (CnBRP)

Remark: f_{CAN}: Clock supplied to CAN = f_{XX}
f_{CANMOD}: CAN module system clock
f_{TQ}: CAN protocol layer basic system clock

Caution: The CnBRP register can be write-accessed only in the initialization mode.

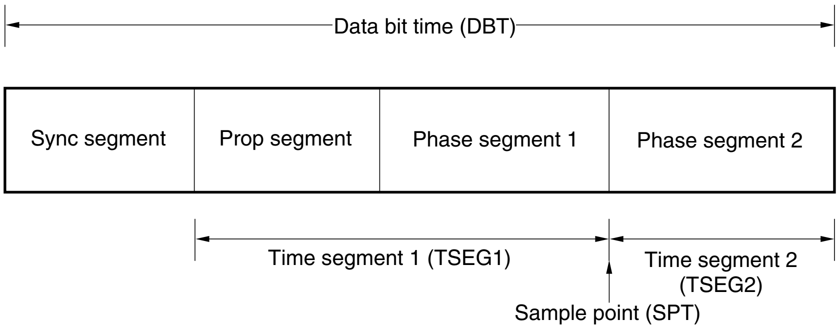
(13) CAN module bit rate register (CnBTR)

The CnBTR register is used to control the data bit time of the communication baud rate.

Figure 16-36: CAN Module Bit Rate Register (CnBTR) Format (1/2)

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnBTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20	see Table 16-16	R/W	370FH
	7	6	5	4	3	2	1	0			
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10			

Data Bit Time



SJW1	SJW0	Length of synchronization jump width
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default value)

Figure 16-36: CAN Module Bit Rate Register (CnBTR) Format (2/2)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time segment 1
0	0	0	0	Setting prohibited
0	0	0	1	2TQ^{Note}
0	0	1	0	3TQ^{Note}
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default value)

Note: This setting must not be made when the CnBRP register = 00H.

Remark: TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)

(14) CAN module last in-pointer register (CnLIPT)

The CnLIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Figure 16-37: CAN Module Last In-Pointer Register (CnLIPT) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnLIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0	see Table 16-16	R	undefined

LIPT7 to LIPT0	Last in-pointer register (CnLIPT)
0.....31	When the CnLIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored.

Remark: The read value of the CnLIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPM bit of the CnRGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLIPT register is undefined.

(15) CAN module receive history list register (CnRGPT)

The CnRGPT register is used to read the receive history list.

Figure 16-38: CAN Module Receive History List Register (CnRGPT) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnRGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0	see Table 16-16	R/W	xx02H
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	RHPM	ROVF			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnRGPT	0	0	0	0	0	0	0	0			
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	0	Clear ROVF			

(a) Read

RGPT7 to RGPT0	Receive history list read pointer
0.....31	When the CnRGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

RHPM^Note 1	Receive history list pointer match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that have not been read.

ROVF	Receive history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least 23 entries have been stored since the host processor serviced the RHL last time (i.e. read CnRGPT). The first 22 entries are sequentially stored whereas the last entry might have been overwritten by newly received messages a number of times because all buffer numbers are stored at position LIPT-1 when the ROVF bit is set to 1. As a consequence receptions cannot be completely recovered in the order that they were received.

Note: The read value of RGPT0 to 7 is invalid when RHPM = 1.

Figure 16-38: CAN Module Receive History List Register (CnRGPT) Format (2/2)

(b) Write

Clear ROVF	Setting of ROVF bit
0	ROVF bit is not changed.
1	ROVF bit is cleared to 0.

(16) CAN module last out-pointer register (CnLOPT)

The CnLOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

Figure 16-39: CAN Module Last Out-Pointer Register (CnLOPT) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnLOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0	see Table 16-16	R	undefined

LOPT7 to LOPT0	Last out-pointer of transmit history list (LOPT)
0.....31	When the CnLOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

Remark: The value read from the CnLOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLOPT register is undefined.

(17) CAN module transmit history list register (CnTGPT)

The CnTGPT register is used to read the transmit history list.

Figure 16-40: CAN Module Transmit History List Register (CnTGPT) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnTGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0	see Table 16-16	R/W	xx02H
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	THPM	TOVF			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnTGPT	0	0	0	0	0	0	0	0			
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	0	0	Clear TOVF			

TGPT7 to TGPT0	Transmit history list read pointer
0.....31	When the CnTGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

THPM[Note 1]	Transmit history pointer match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer numbers that have not been read.

TOVF	Transmit history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least 7 entries have been stored since the host processor serviced the THL last time (i.e. read CnTGPT). The first 6 entries are sequentially stored whereas the last entry might have been overwritten by newly transmitted messages a number of times because all buffer numbers are stored at position LOPT-1 when TOVF bit is set to 1. As a consequence receptions cannot be completely recovered in the order that they were received

Note: The read value of TGPT0 to TGPT7 is invalid when THPM = 1.3.

Remark: Transmission from message buffers 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

Figure 16-40: CAN Module Transmit History List Register (CnTGPT) Format (2/2)

(b) Write

Clear TOVF	Setting of TOVF bit
0	TOVF bit is not changed.
1	TOVF bit is cleared to 0.

(18) CAN module time stamp register (CnTS)

The CnTS register is used to control the time stamp function.

Figure 16-41: CAN Module Time Stamp Register (CnTS) Format (1/2)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnTS	0	0	0	0	0	0	0	0	see Table 16-16	R/W	0000H
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	TSLOCK	TSSEL	TSEN			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnTS	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN			
	7	6	5	4	3	2	1	0			
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN			

Remark: The time stamp function must not be used when the CAN module is in the normal operation mode with ABT.

(a) Read

TSLOCK	Time stamp lock function enable bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled.The TSOUT signal toggled each time the selected time stamp capture event occurred. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0[Note].

Note: The TSEN bit is automatically cleared to 0.

Figure 16-41: CAN Module Time Stamp Register (CnTS) Format (2/2)

TSSEL	Time stamp capture event selection bit
0	The time capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT operation setting bit
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

(b) Write

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.

(19) CAN message data byte register (CnMDATAxm) (x = 0 to 7)

The CnMDATAxm register is used to store the data of a transmit/receive message.

Figure 16-42: CAN Message Data Byte Register (CnMDATAxm) (x = 0 to 7) Format (1/2)

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMDATA01m	MDATA01 15	MDATA01 14	MDATA01 13	MDATA01 12	MDATA01 11	MDATA01 10	MDATA01 9	MDATA01 8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	MDATA01 7	MDATA01 6	MDATA01 5	MDATA01 4	MDATA01 3	MDATA01 2	MDATA01 1	MDATA01 0			

	7	6	5	4	3	2	1	0
CnMDATA0m	MDATA0 7	MDATA0 6	MDATA0 5	MDATA0 4	MDATA0 3	MDATA0 2	MDATA0 1	MDATA0 0

	7	6	5	4	3	2	1	0
CnMDATA1m	MDATA1 7	MDATA1 6	MDATA1 5	MDATA1 4	MDATA1 3	MDATA1 2	MDATA1 1	MDATA1 0

	15	14	13	12	11	10	9	8
CnMDATA23m	MDATA23 15	MDATA23 14	MDATA23 13	MDATA23 12	MDATA23 11	MDATA23 10	MDATA23 9	MDATA23 8
	7	6	5	4	3	2	1	0
	MDATA23 7	MDATA23 6	MDATA23 5	MDATA23 4	MDATA23 3	MDATA23 2	MDATA23 1	MDATA23 0

	7	6	5	4	3	2	1	0
CnMDATA2m	MDATA2 7	MDATA2 6	MDATA2 5	MDATA2 4	MDATA2 3	MDATA2 2	MDATA2 1	MDATA2 0

	7	6	5	4	3	2	1	0
CnMDATA3m	MDATA3 7	MDATA3 6	MDATA3 5	MDATA3 4	MDATA3 3	MDATA3 2	MDATA3 1	MDATA3 0

Figure 16-42: CAN Message Data Byte Register (CnMDATAxm) (x = 0 to 7) Format (2/2)

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMDATA45m	MDATA45 15	MDATA45 14	MDATA45 13	MDATA45 12	MDATA45 11	MDATA45 10	MDATA45 9	MDATA45 8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	MDATA45 7	MDATA45 6	MDATA45 5	MDATA45 4	MDATA45 3	MDATA45 2	MDATA45 1	MDATA45 0			

	7	6	5	4	3	2	1	0
CnMDATA4m	MDATA4 7	MDATA4 6	MDATA4 5	MDATA4 4	MDATA4 3	MDATA4 2	MDATA4 1	MDATA4 0

	7	6	5	4	3	2	1	0
CnMDATA5m	MDATA5 7	MDATA5 6	MDATA5 5	MDATA5 4	MDATA5 3	MDATA5 2	MDATA5 1	MDATA5 0

	15	14	13	12	11	10	9	8
CnMDATA67m	MDATA67 15	MDATA67 14	MDATA67 13	MDATA67 12	MDATA67 11	MDATA67 10	MDATA67 9	MDATA67 8
	7	6	5	4	3	2	1	0
	MDATA67 7	MDATA67 6	MDATA67 5	MDATA67 4	MDATA67 3	MDATA67 2	MDATA67 1	MDATA67 0

	7	6	5	4	3	2	1	0
CnMDATA6m	MDATA6 7	MDATA6 6	MDATA6 5	MDATA6 4	MDATA6 3	MDATA6 2	MDATA6 1	MDATA6 0

	7	6	5	4	3	2	1	0
CnMDATA7m	MDATA7 7	MDATA7 6	MDATA7 5	MDATA7 4	MDATA7 3	MDATA7 2	MDATA7 1	MDATA7 0

(20) CAN message data length register m (CnMDLCm)

The CnMDLCm register is used to set the number of bytes of the data field of a message buffer.

Figure 16-43: CAN Message Data Length Register m (CnMDLCm) Format

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnMDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0	see Table 16-16	R/W	0000xxxxB

MDLC3	MDLC2	MDLC1	MDLC0	Data length of transmit/receive message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)[Note]
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Note: The data and DLC value actually transmitted to CAN bus are as follows.

Type of transmit frame	Length of transmit data	DLC transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if DLC ≥ 8)	MDLC[3:0]
Remote frame	0 bytes	

Cautions: 1. Be sure to set bits 7 to 4 to 0000B.

2. Receive data is stored in as many CnMDATAx as the number of bytes (however, the upper limit is 8) corresponding to DLC. CnMDATAx in which no data is stored is undefined.

(21) CAN message configuration register (CnMCONFm)

The CnMCONFm register is used to specify the type of the message buffer and to set a mask.

Figure 16-44: CAN Message Configuration Register (CnMCONFm) Format (1/2)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0	see Table 16-16	R/W	undefined

OWS	Overwrite control bit
0	The message buffer[Note] that has already received a data frame is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame is overwritten by a newly received data frame.

Note: The “message buffer that has already received a data frame” is a receive message buffer whose DN bit has been set to 1.

Remark: A remote frame is received and stored, regardless of the setting of OWS and DN. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

RTR	Remote frame request bit[Note]
0	Transmit a data frame.
1	Transmit a remote frame.

Note: The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer. Even if a valid remote frame has been received, RTR of the transmit message buffer that has received the frame remains cleared to 0. Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

MT2	MT1	MT0	Message buffer type setting bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited

Figure 16-44: CAN Message Configuration Register (CnMCONFm) Format (2/2)

MA0	Message buffer assignment bit
0	Message buffer not used.
1	Message buffer used.

Caution: Be sure to write 0 to bits 2 and 1.

(22) CAN message ID register m (CnMIDLm, CnMIDHm)

The CnMIDLm and CnMIDHm registers are used to set an identifier (ID).

Figure 16-45: CAN message ID register m (CnMIDLm, CnMIDHm) Format

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	see Table 16-16	R/W	undefined
	7	6	5	4	3	2	1	0			
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0			

	15	14	13	12	11	10	9	8
CnMIDHm	IDE	0	0	ID28	ID27	ID26	ID25	ID24
	7	6	5	4	3	2	1	0
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format mode specification bit
0	Standard format mode (ID28 to ID18: 11 bits)[Note]
1	Extended format mode (ID28 to ID0: 29 bits)

Note: The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

Caution: Be sure to write 0 to bits 14 and 13 of the CnMIDHm register.

(23) CAN message control register m (CnMCTRLm)

The CnMCTRLm register is used to control the operation of the message buffer.

Figure 16-46: CAN Message Control Register m (CnMCTRLm) Format (1/3)

(a) Read

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMCTRLm	0	0	MUC	0	0	0	0	0	see Table 16-16	R/W	00x000000 00000000B
	7	6	5	4	3	2	1	0			
	0	0	0	MOW	IE	DN	TRQ	RDY			

(b) Write

	15	14	13	12	11	10	9	8	Address	R/W	After reset
CnMCTRLm	0	0	0	0	Set IE	0	Set TRQ	Set RDY			
	7	6	5	4	3	2	1	0			
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY			

(a) Read

MUC[Note]	Bit indicating that message buffer data is being updated
0	The CAN module is not updating the message buffer (reception and storage).
1	The CAN module is updating the message buffer (reception and storage).

Note: The MUC bit is undefined until the first reception and storage is performed.

MOW	Message buffer overwrite status bit
0	The message buffer is not overwritten by a newly received data frame.
1	The message buffer is overwritten by a newly received data frame.

Remark: MOW is not set to 1 even if a remote frame is received and stored in the transmit message buffer with DN = 1.

IE	Message buffer interrupt request enable bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

Figure 16-46: CAN Message Control Register m (CnMCTRLm) Format (2/3)

DN	Message buffer data update bit
0	A data frame or remote frame is not stored in the message buffer.
1	A data frame or remote frame is stored in the message buffer.

TRQ	Message buffer transmission request bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Caution: Do not set the TRQ bit and RDY bit to 1 at the same time. Be sure to set the RDY bit to 1 before setting the TRQ bit to 1.

RDY	Message buffer ready bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer.

Caution: Do not clear the RDY bit (0) during message transmission. Follow transmission abort procedures in order to clear the RDY bit for redefinition.

(b) Write

Clear MOW	Setting of MOW bit
0	MOW bit is not changed.
1	MOW bit is cleared to 0.

Set IE	Clear IE	Setting of IE bit
0	1	IE bit is cleared to 0.
1	0	IE bit is set to 1.
Other than above		IE bit is not changed.

Clear DN	Setting of DN bit
1	DN bit is cleared to 0.
0	DN bit is not changed.

Caution: Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.

Figure 16-46: CAN Message Control Register m (CnMCTRLm) Format (3/3)

Set TRQ	Clear TRQ	Setting of TRQ bit
0	1	TRQ bit is cleared to 0.
1	0	TRQ bit is set to 1.
Other than above		TRQ bit is not changed.

Set RDY	Clear RDY	Setting of RDY bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

16.7 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CAN global control register (CnGMCTRL)
- CAN global automatic block transmission control register (CnGMABT)
- CAN module control register (CnCTRL)
- CAN module interrupt enable register (CnIE)
- CAN module interrupt status register (CnINTS)
- CAN module receive history list register (CnRGPT)
- CAN module transmit history list register (CnTGPT)
- CAN module time stamp register (CnTS)
- CAN message control register (CnMCTRLm)

Remark: n = 0, 1
m = 0 to 31

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in Figure 16-47 below to set or clear the lower 8 bits in these registers.
Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the bit status after set/clear operation is specified in Figure 16-48). Figure 16-47 shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 16-47: Example of Bit Setting/Clearing Operations

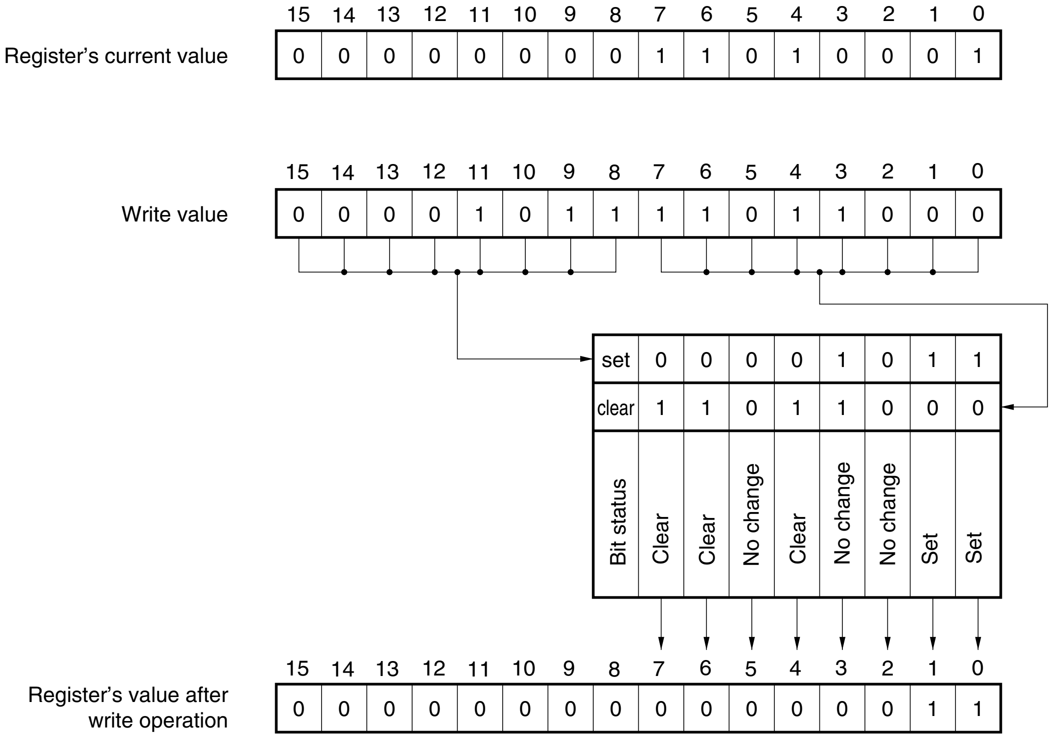


Figure 16-48: Bit Status After Bit Setting/Clearing Operations

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

Set n	Clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark: n = 0 to 7

16.8 CAN Controller Initialization

16.8.1 Initialization of CAN module

Before CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the CnGMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.
The CAN module is enabled by setting the GOM bit of the CnGMCTRL register.
For the procedure of initializing the CAN module, refer to **16.16 ”Operation of CAN Controller” on page 660.**

16.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.
- Clear the MA0 bit of the CnMCONFm register to 0.

Remark: n = 0, 1
m = 0 to 31

16.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module to an operation mode.

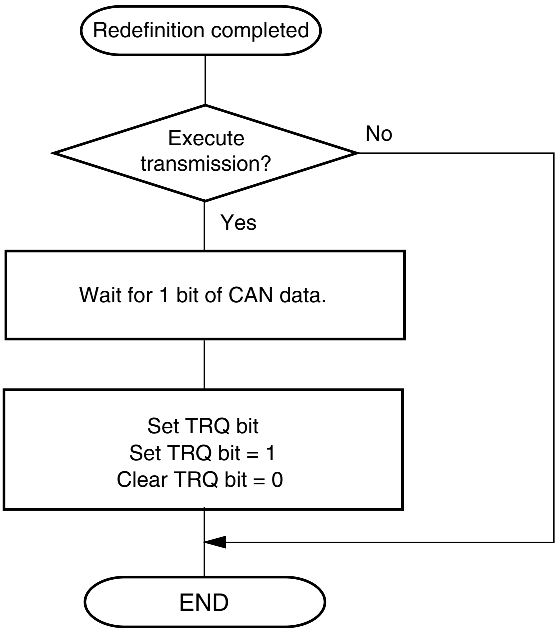
(2) To redefine message buffer during reception

Perform redefinition as shown in **Figure 16-60, “Message Buffer Redefinition,” on page 663** and **Figure 16-61, “Transmitting Message Buffer Redefinition,” on page 664**.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (refer to **16.10.4 (1)”Transmission abort in normal operation mode” on page 641** and **16.10.4 (2)”Transmission abort in normal operation mode with automatic block transmission (ABT)” on page 641**). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

Figure 16-49: Setting Transmission Request (TRQ) to Transmit Message Buffer After Redefinition



Cautions: 1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in the Figure 16-60, "Message Buffer Redefinition," on page 663 is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

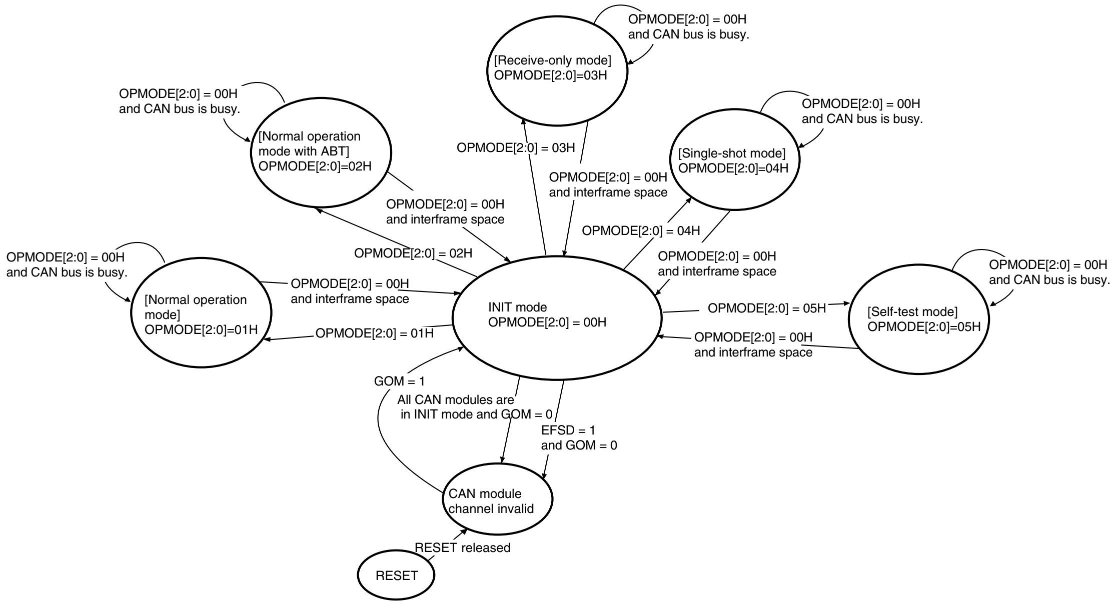
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 16-49 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

16.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

Figure 16-50: Transition to Operation Modes


[Receive-only mode]
OPMODE[2:0]=03H
OPMODE[2:0] = 00H and CAN bus is busy.
[Normal operation mode with ABT]
OPMODE[2:0]=02H
OPMODE[2:0] = 00H and CAN bus is busy.
[Single-shot mode]
OPMODE[2:0]=04H
OPMODE[2:0] = 00H and CAN bus is busy.
OPMODE[2:0] = 03H
OPMODE[2:0] = 00H and interframe space
OPMODE[2:0] = 00H and interframe space
OPMODE[2:0] = 04H
OPMODE[2:0] = 02H
OPMODE[2:0] = 00H and interframe space
[Normal operation mode]
OPMODE[2:0]=01H
OPMODE[2:0] = 00H and CAN bus is busy.
OPMODE[2:0] = 00H and interframe space
OPMODE[2:0] = 01H
INIT mode
OPMODE[2:0] = 00H
OPMODE[2:0] = 05H
OPMODE[2:0] = 00H and interframe space
[Self-test mode]
OPMODE[2:0]=05H
OPMODE[2:0] = 00H and CAN bus is busy.
GOM = 1
All CAN modules are in INIT mode and GOM = 0
EFSD = 1 and GOM = 0
CAN module channel invalid
RESET released
RESET

The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE[2:0] in the CnCTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from an operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the values of OPMODE[2:0] are changed to 00H). After issuing a request to change the mode to the initialization mode, read the OPMODE[2:0] bits until their values become 000B to confirm that the module has entered the initialization mode (refer to **Figure 16-58, "Re-initialization," on page 661**).

Remark: n = 0, 1
m = 0 to 31

16.8.5 Resetting error counter CnERC of CAN module

If it is necessary to reset the CAN module error counter CnERC and the CAN module information register CnINFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the CnCTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter CnERC and the CAN module information register CnINFO are cleared to their default values.

Remark: n = 0, 1
m = 0 to 31

16.9 Message Reception

16.9.1 Message reception

In all the operation modes, when a message is received, a message buffer that is to store the message is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
(MA0 bit of CnMCONFm register set to 1B.)
- Set as a receive message buffer
(MT[2:0] bits of CnMCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
(RDY bit of CnMCTRLm register set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1 that has not received a message, even if a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set to store a message in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to receive and store a message (i.e., when the DN bit = 1 indicating that a message has already been received, but rewriting is disabled because the OWS bit = 0). In this case, the message is not actually received and stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Priority	Storing Condition If Same ID Is Set	
1 (high)	Unmasked message buffer	DN = 0
		DN = 1 and OWS = 1
2	Message buffer linked to mask 1	DN = 0
		DN = 1 and OWS = 1
3	Message buffer linked to mask 2	DN = 0
		DN = 1 and OWS = 1
4	Message buffer linked to mask 3	DN = 0
		DN = 1 and OWS = 1
5 (low)	Message buffer linked to mask 4	DN = 0
		DN = 1 and OWS = 1

Remark: n = 0, 1
m = 0-31

16.9.2 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding CnLIPT register and the receive history list get pointer (RGPT) with the corresponding CnRGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CnLIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the CnLIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CnRGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the CnRGPT register, the RGPT pointer is automatically incremented.

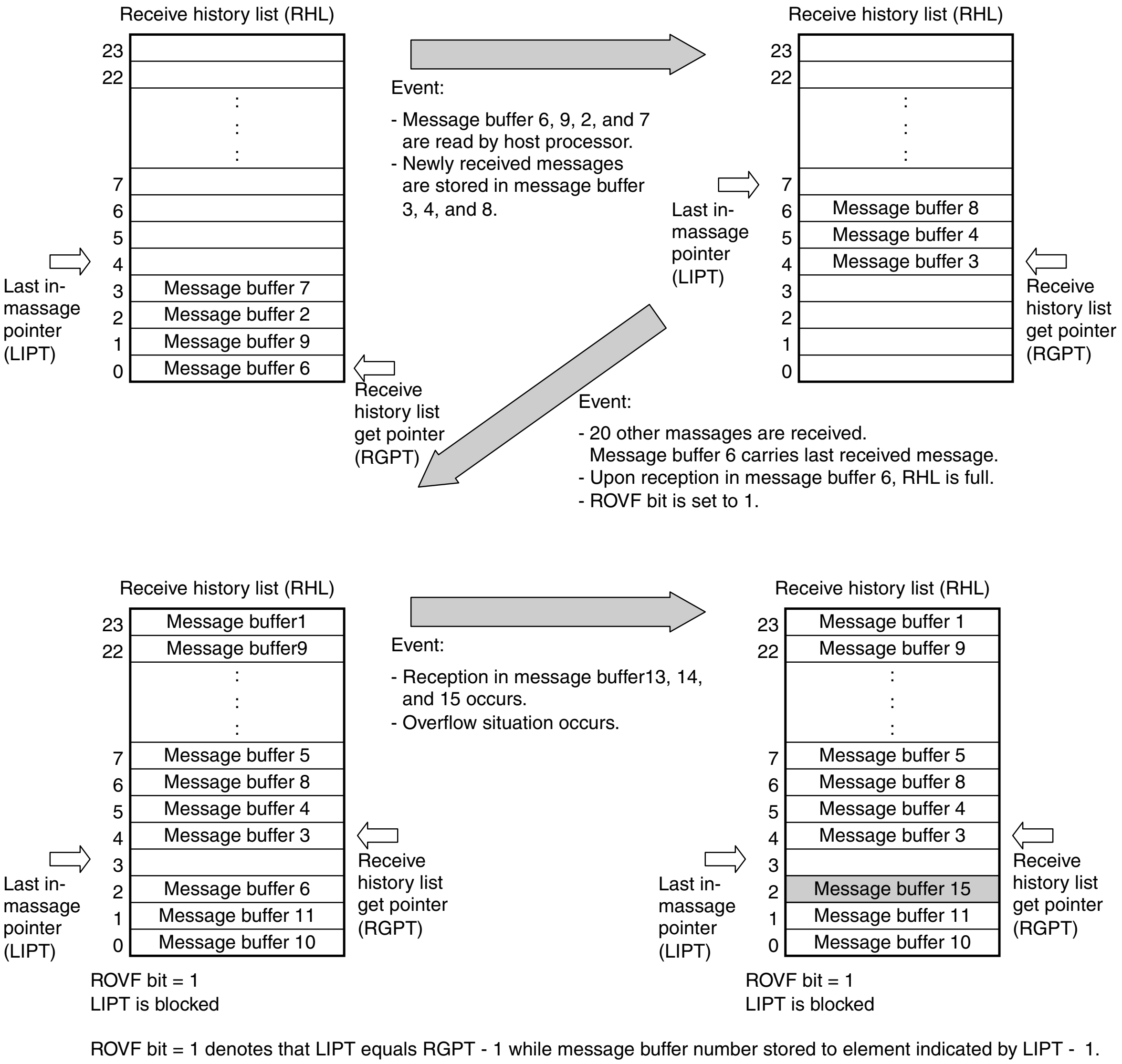
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the CnRGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the CnRGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the new message. After the ROVF bit has been set (1), therefore, the recorded message buffer numbers in the RHL do not completely reflect the chronological order.

However the messages themselves are not lost and can be located by a CPU search in the message buffer memory with the help of the DN bit. As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without the RHL being read by the host processor, a complete sequence of receptions can not be recovered.

Remark: n = 0, 1
m = 0 to 31

Figure 16-51: Receive History List


Receive history list (RHL)
23
22
7
6
5
4
3 Message buffer 7
2 Message buffer 2
1 Message buffer 9
0 Message buffer 6
Last in-massage pointer (LIPT)
Receive history list get pointer (RGPT)
Event:
- Message buffer 6, 9, 2, and 7 are read by host processor.
- Newly received messages are stored in message buffer 3, 4, and 8.
Receive history list (RHL)
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
Last in-massage pointer (LIPT)
Receive history list get pointer (RGPT)
Event:
- 20 other massages are received. Message buffer 6 carries last received message.
- Upon reception in message buffer 6, RHL is full.
- ROVF bit is set to 1.
Receive history list (RHL)
23 Message buffer1
22 Message buffer9
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3
2 Message buffer 6
1 Message buffer 11
0 Message buffer 10
Last in-massage pointer (LIPT)
Receive history list get pointer (RGPT)
ROVF bit = 1
LIPT is blocked
Event:
- Reception in message buffer13, 14, and 15 occurs.
- Overflow situation occurs.
Receive history list (RHL)
23 Message buffer 1
22 Message buffer 9
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3
2 Message buffer 15
1 Message buffer 11
0 Message buffer 10
Last in-massage pointer (LIPT)
Receive history list get pointer (RGPT)
ROVF bit = 1
LIPT is blocked
ROVF bit = 1 denotes that LIPT equals RGPT - 1 while message buffer number stored to element indicated by LIPT - 1.

16.9.3 Mask function

It can be defined whether masking of the identifier that is set to a message buffer is linked with another message buffer.

By using the mask function, the identifier of a message received from the CAN bus can be compared with the identifier set to a message buffer in advance. Regardless of whether the masked ID is set to 0 or 1, the received message can be stored in the defined message buffer.

While the mask function is in effect, an identifier bit that is defined to be 1 by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as 0 by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are 0 and bits ID24 and ID22 are 1, are to be stored in message buffer 14. The procedure for this example is shown below.

<1> Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

Remark: x = don’t care

<2> Identifier to be configured in message buffer 14 (example)
(Using CANn message ID registers L14 and H14 (CnMIDL14 and CnMIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

ID with ID27 to ID25 cleared to 0 and ID24 and ID22 set to 1 is registered (initialized) to message buffer 14.

Remark: Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT[2:0] of CnMCONF14 register are set to 010B).

<3> Mask setting for CAN module 1 (mask 1) (Example)
(Using CAN1 address mask 1 registers L and H (C1MASKL1 and C1MASKH1))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

Remark: 1: Not compared (masked)
0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to 0, and the CMID28, CMID23, and CMID21 to CMID0 bits are set to 1.

16.9.4 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type.

Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches an ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

When the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the CnMCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions:

1. **MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.**
2. **MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.**
3. **MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.**
4. **With MBRB, “matching ID” means “matching ID after mask”. Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.**

Remark: n = 0, 1
m = 0 to 31

16.9.5 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits in CnMCONFm register set to 000B)
- Ready for reception
  (RDY bit of CnMCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of CnMCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of CnMCTRLm register is cleared to 1.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The DLC[3:0] bit string in the CnMDLCm register stores the received DLC value.
- CnMDATA0m to CnMDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the CnMCTRLm register is set to 1.
- The CINTS1 bit of the CnINTS register is set to 1 (if the IE bit in the CnMCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The receive completion interrupt (INTRECn) is output (if the IE bit in the CnMCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the CnIE register is set to 1).
- The message buffer number is recorded in the receive history list.

Caution: When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the CnMCONFm register of the message buffer and the DN bit of the CnMCTRLm register are not affected.
If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

Remark: n = 0, 1
m = 0 to 31

16.10 Message Transmission

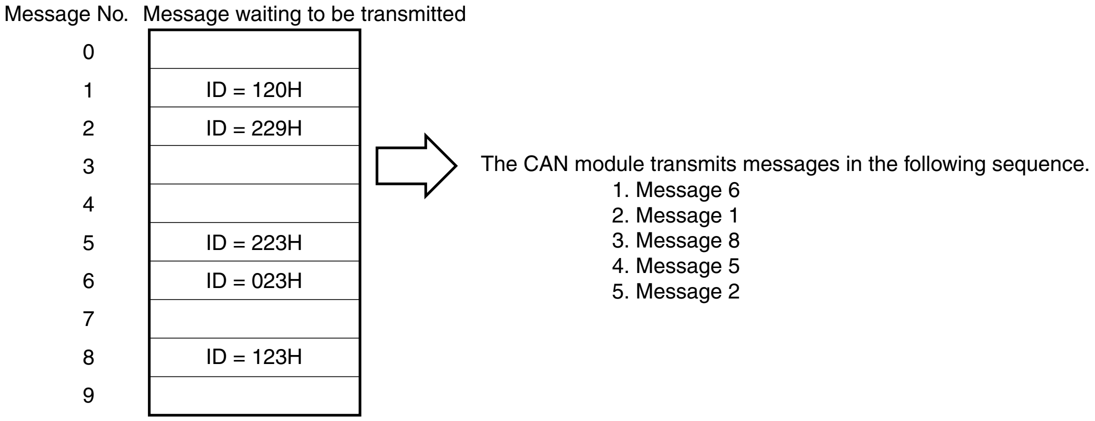
16.10.1 Message transmission

In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits of CnMCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of CnMCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.
Transmission priority is controlled by the identifier (ID).

Figure 16-52: Message Processing Example



Message No.	Message waiting to be transmitted
0	
1	ID = 120H
2	ID = 229H
3	
4	
5	ID = 223H
6	ID = 023H
7	
8	ID = 123H
9	

The CAN module transmits messages in the following sequence.
1. Message 6
2. Message 1
3. Message 8
4. Message 5
5. Message 2

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.
If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than a message frame with a 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If one or more transmission-pending extended ID message frame has equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Remarks: 1. If the automatic block transmission request bit CnGMABT.ABTTRGbit is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.If the ABT mode was triggered by the ABTTRG bit (1), one TRQ bit is set to 1 in the ABT area (buffers 0 to 7). In addition to this TRQ bit, the application can request transmissions (set TRQ bit to 1) for other TX-message buffers that do not belong to the ABT area. In that case an internal arbitration process (TX-search) evaluates all of the TX-message buffers with the TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted first.Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the CnINTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTRRX1 is output (if the CIE0 bit of the CnIE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

2. n = 0, 1
m = 0 to 31

16.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer in which each data frame or remote frame was received and stored. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding CnLOPT register, and the transmit history list get pointer (TGPT) with the corresponding CnTGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CnLOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the CnLOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the CnTGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the CnTGPT register, the TGPT pointer is automatically incremented.

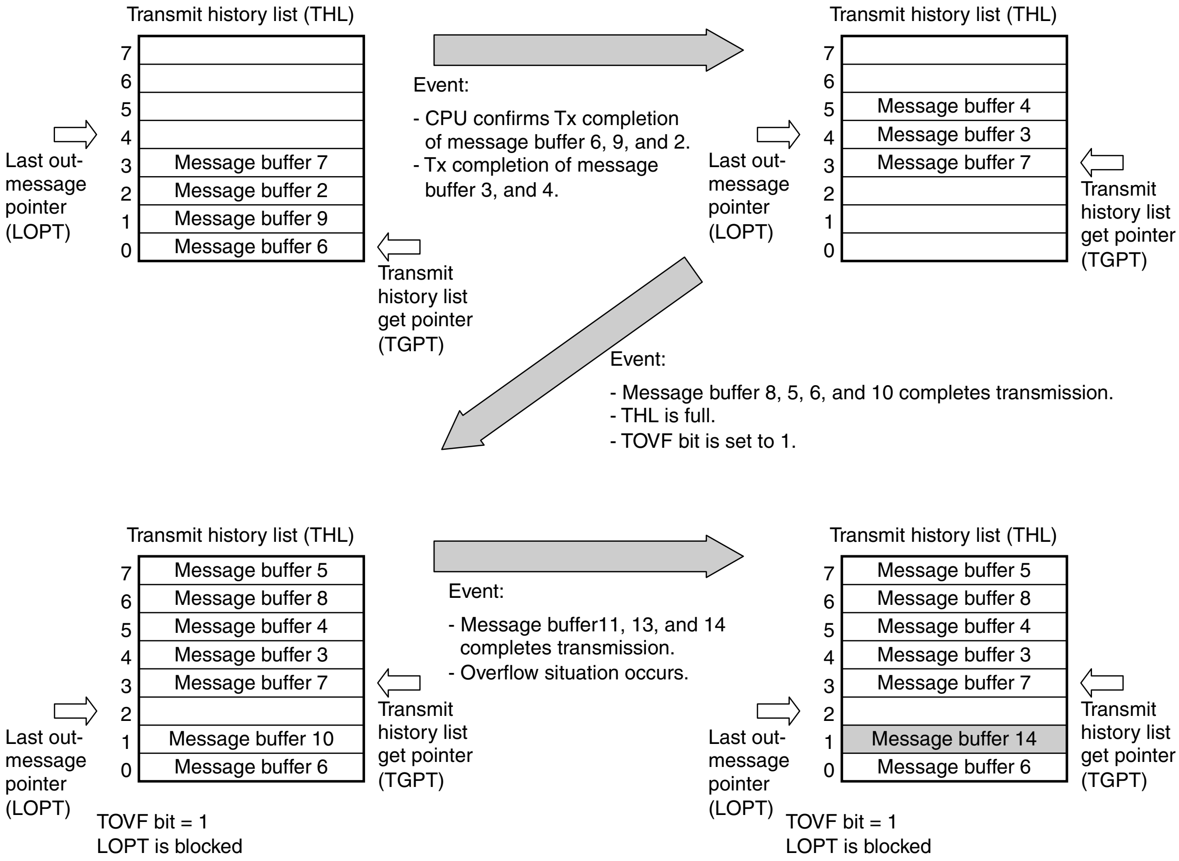
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the CnTGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (receive history list overflow) of the CnTGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the number of the message buffer that received and stored the new message. After the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order.

However the transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.

Remark: n = 0, 1
m = 0-31

Figure 16-53: Transmit History List


Transmit history list (THL)
7
6
5
4
3 Message buffer 7
2 Message buffer 2
1 Message buffer 9
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
Event:
- CPU confirms Tx completion of message buffer 6, 9, and 2.
- Tx completion of message buffer 3, and 4.
Transmit history list (THL)
7
6
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1
0
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
Event:
- Message buffer 8, 5, 6, and 10 completes transmission.
- THL is full.
- TOVF bit is set to 1.
Transmit history list (THL)
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1 Message buffer 10
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
TOVF bit = 1
LOPT is blocked
Event:
- Message buffer11, 13, and 14 completes transmission.
- Overflow situation occurs.
Transmit history list (THL)
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1 Message buffer 14
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
TOVF bit = 1
LOPT is blocked

TOVF bit = 1 denotes that LOPT equals TGPT - 1 while message buffer number stored to element indicated by LOPT - 1.

16.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting OPMODE[2:0] of the CnCTRL register to 010B, “normal operation mode with automatic block transmission function” (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting the MT[2:0] bits to 000B. Be sure to set the same ID for the message buffers for ATB even when that ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the CnMIDLm and CnMIDHm registers. Set the CnMDLCm and CnMDATA0m to CnMDATA7m registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 is finished, TRQ of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the CnGMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the CnBRP and CnBTR registers.

During ABT, the priority of the transmission ID is not searched. The data of message buffers 0 to 7 is sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and ABTTRG is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the CnMCTRLm register of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function (message buffer 8 to 31) is assigned to a transmit message buffer, the priority of the message to be transmitted is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions:

1. **To resume the normal operation mode with ABT from the message buffer 0, set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.**
2. **If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.**
3. **Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.**
4. **Do not set TRQ of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.**
5. **The CnGMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffer 8 to 31).**
6. **If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (CnGMABTD = 00H), messages other than ABT messages are transmitted. At this time, transmission does not depend on the priority of the ABT message.**
7. **Do not clear the RDY bit to 0 when ABTTRG = 1.**
8. **If a message is received from another node in the normal operation mode with ABT, the message may be transmitted after the time of one frame has elapsed (when CnGMABTD register = 00H).**

16.10.4 Transmission abort process

(1) Transmission abort in normal operation mode

The user can clear the TRQ bit of the CnMCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 16-66, "Transmission via Software Polling," on page 669**).

(2) Transmission abort in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the CnGMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.
If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.
In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in **Figure 16-67, "Transmission Abort Processing (except Normal Operation Mode with ABT)," on page 670**).

If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is increased in increments of 1 and indicates the next message buffer in the ABT area (for details, refer to the process in Figure **16-67**).

When the normal operation mode with ABT is resumed after ABT has been aborted and ABTTRG is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of TRQ of ABT Message Buffer	Abort After Successful Transmission	Abort After Erroneous Transmission
Set (1)	Next message buffer in the ABT area^Note	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area^Note	Next message buffer in the ABT area^Note

Note: The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if ABTTRG is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if ABTTRG is set to 1, and ABT ends immediately.

16.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the CnMCONFm register. Setting (1) the RTR bit sets remote frame transmission.

16.11 Power Saving Modes

16.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.
In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01B to the PSMODE[1:0] bits of the CnCTRL register.
This transition request is only acknowledged only under the following conditions.

(a) The CAN module is already in one of the following operation modes

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode
- CAN stop mode in all the above operation modes

(b) The CAN bus state is bus idle (the 4th bit in the interframe space is recessive)[Note]

Note: If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending.
Also the transition from CAN stop mode to CAN sleep mode is independent of the CAN bus state.

(c) No transmission request is pending

If any one of the conditions mentioned above is not met, the CAN module will operate as follows.

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request has to be held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE[1:0] bits remain 00B. When the module has entered the CAN sleep mode, PSMODE[1:0] are set to 01B.
- If a request for transition to the initialization mode and a request for transition to the CAN sleep mode are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.
- If a CAN sleep mode request is pending waiting for the CAN bus state to become bus idle while the CAN module is in one of the operation modes, and if a request for transition to the initialization mode is made, the pending CAN sleep mode request becomes disabled, and only the initialization mode request is enabled (in this case, the CAN sleep mode request continues to be held pending).
- If the CAN sleep mode transition request is made while a initialization mode transition request is held pending waiting for completion of communication in one of the operation modes, the CAN sleep mode transition request is ignored and only the initialization mode transition request remains valid (in this case, the CAN sleep mode request continues to be held pending).

Even when the initialization mode and sleep mode are not requested simultaneously (i.ethe first request was not granted when a second request was made), the request for initialization has priority over the CAN sleep mode request. The CAN sleep mode request is cancelled when the initialization mode is requested. When a pending request for the initialization mode is present, a subsequent request for the CAN sleep mode request is cancelled right at the point in time when it was submitted.

(2) Status in CAN sleep mode

- The CAN module is in one of the following states after it enters the CAN sleep mode.
- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRXDn) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to PSMODE[1:0] of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CnLIPT, CnRGPT, CnLOPT, and CnTGPT.
- The CAN message buffer registers cannot be written or read.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing CAN sleep mode

The CAN sleep mode is released by the following events.

- When the CPU writes 00B to the PSMODE[1:0] bits of the CnCTRL register
- A falling edge at the CAN reception pin (CRXDn) (i.e. the CAN bus level shifts from recessive to dominant)

Caution: Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock to the CAN while the CAN was in sleep mode, later on the CAN sleep mode will not be released and PSMODE[1:0] bits will continue to be 01B unless the clock for the CAN is provided again. In addition to this, the receive message will not be received afterwards.

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE[1:0] bits of the CnCTRL register are reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the CnINTS register is set to 1, regardless of the CIE bit of the CnIE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus.
When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CPU has to be released from sleep mode by software first before entering the initialization mode.

Remark: n = 0, 1
m = 0 to 31

16.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.
The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bits of the CnCTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11B to the PSMODE[1:0] bits of the CnCTRL register.
A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

Caution: To set the CAN module to the CAN stop mode, the module must be in the CAN sleep mode. To confirm that the module is in the sleep mode, check that PSMODE[1:0] = 01B, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRXD) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged.

(2) Status in CAN stop mode

- The CAN module is in one of the following states after it enters the CAN stop mode.
- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to PSMODE[1:0] of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CnLIPT, CnRGPT, CnLOPT, and CnTGPT.
- The CAN message buffer registers cannot be written or read.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01B to the PSMODE1 and PSMODE0 bits. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently the CAN sleep mode before entering into initialization mode. It is impossible to enter another operation mode directly from the CAN stop mode without entering the CAN sleep mode, the request will be ignored.

Remark: n = 0, 1
m = 0 to 31

16.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example of using the power saving modes.

First, put the CAN module in the CAN sleep mode (PSMODE = 01B). Next, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the CnCTRL register is set to 1, a wake-up interrupt (INTWUP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returns to the normal operation mode. The CPU, in response to INTWUP, can release its own power saving mode and return to the normal operation mode. To further reduce the power consumption of the CPU, the internal clocks, including that of the CAN module, may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module is put in the CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wake-up interrupt (INTWUP) even if it is not supplied with the clock. The other functions, however, do not operate because clock supply to the CAN module is stopped, and the module remains in the CAN sleep mode. The CPU, in response to INTWUP, releases its power saving mode, resumes supply of the internal clocks, including the clock to the CAN module, after the oscillation stabilization time has elapsed, and starts instruction execution. The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode
(PSMODE = 00B).

Remark: n = 0, 1
m = 0 to 31

16.12 Interrupt Function

The CAN module provides 6 different interrupt sources.
The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 16-20: List of CAN Module Interrupt Sources

No	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0	CnINTS	CIE0**Note**	CnIE	INTCnTRX	Message frame successfully transmitted from message buffer m
2	CINTS1	CnINTS	CIE1**Note**	CnIE	INTCnREC	Valid message frame reception in message buffer m
3	CINTS2	CnINTS	CIE2	CnIE	INTCnERR	CAN module error state interrupt (Supplement 1)
4	CINTS3	CnINTS	CIE3	CnIE		CAN module protocol error interrupt (Supplement 2)
5	CINTS4	CnINTS	CIE4	CnIE		CAN module arbitration loss interrupt
6	CINTS5	CnINTS	CIE5	CnIE	INTCnWUP	CAN module wake-up interrupt from CAN sleep mode (Supplement 3)

Note: The IE bit (message buffer interrupt enable bit) in the CnMCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

Supplements:

1. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.
2. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
3. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

Remark: n = 0, 1
m = 0 to 31

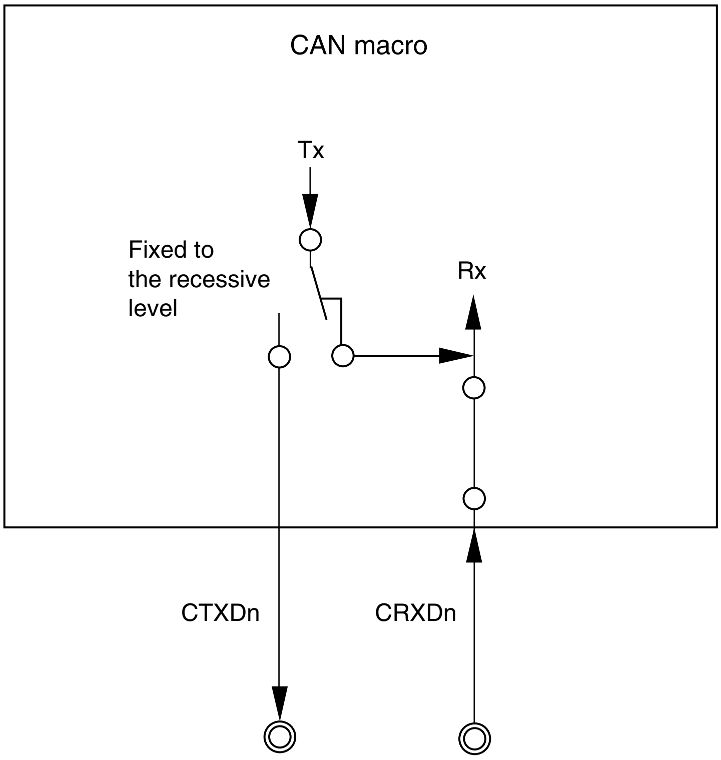
16.13 Diagnosis Functions and Special Operational Modes

The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of special CAN communication methods.

16.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.
For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until "valid reception" is detected, so that the baud rates in the module match ("valid reception" means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the CnCTRL register (1).

Figure 16-54: CAN Module Terminal Connection in Receive-Only Mode



In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.
In the receive-only mode, the CAN transmission pin (CTXDn) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter TEC is never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.
Furthermore, ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution: If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). When the message frame is transmitted for the 17th time, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

16.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behaviour of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.
The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the CnCTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events.

- Successful transmission of the message frame
- Arbitration loss while sending the message frame (AL bit = 0)
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the CnINTS register, and the type of the error can be identified by reading the LEC[2:0] bits of the CnLEC register.
Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the CnINTS register is set to 1. If the CIE0 bit of the CnIE register is set to 1 at this time, an interrupt request signal is output.
The single-shot mode can be used when emulating time-triggered communication methods (e.g., TTCAN level 1).

Caution: The AL bit is only valid in single-shot mode. It does not affect the operation of re-transmission upon arbitration loss in other operation modes.

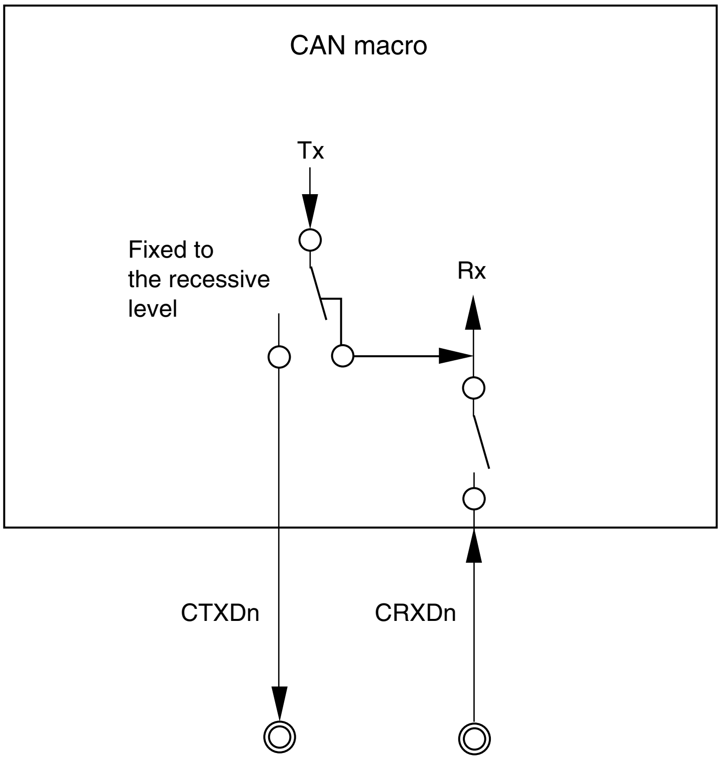
16.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTXDn) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRXDn) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes. To keep the module in the CAN sleep mode, use the CAN reception pin (CRXDn) as a port pin.

Figure 16-55: CAN Module Terminal Connection in Self-Test Mode



16.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may even have different frequencies).
In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

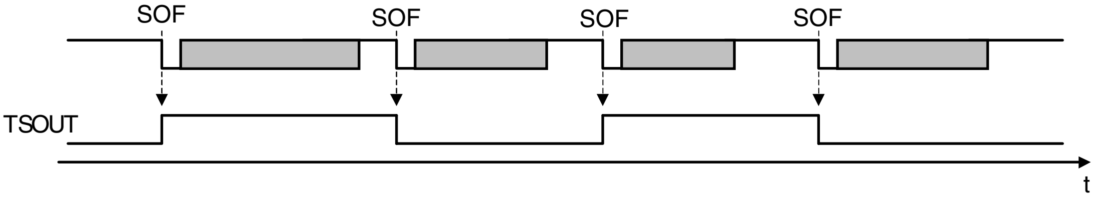
16.14.1 Time stamp function

The CAN controller supports the capturing of timer values triggered by successful reception of a data frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. TSOUT can be selected from the following two event sources and is specified by the TSSEL bit of the CnTS register.

• SOF event (start of frame)	(TSSEL = 0)
• EOF event (last bit of end of frame)	(TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the CnTS register to 1.

Figure 16-56: Timing Diagram of Capture Signal TSOUT



TSOUT toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT, the capture timer unit must detect the capture signal at both the rising edge and falling edge.
This time stamp function is controlled by the TSLOCK bit of the CnTS register. When TSLOCK is cleared to 0, TSOUT toggles upon occurrence of the selected event. If TSLOCK is set to 1, TSOUT toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 when a data frame is received and stored in message buffer 0. This suppresses the subsequent toggle occurrence by TSOUT, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution: The time stamp function using TSLOCK stops toggle of TSOUT by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of TSOUT cannot be stopped by reception of a remote frame. Toggle of TSOUT does not stop when a data frame is received in a message buffer other than message buffer 0. For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of TSOUT by TSLOCK cannot be used.

Remark: n = 0, 1
m = 0 to 31

16.15 Baud Rate Settings

16.15.1 Bit rate setting conditions

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN controller, as follows.

(a) 5TQ ≤ SPT (sampling point) ≤ 17 TQ
SPT = TSEG1 + 1
(b) 8 TQ ≤ DBT (data bit time) ≤ 25 TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT
(c) 1 TQ ≤ SJW (synchronization jump width) ≤ 4TQ
SJW ≤ DBT – SPT
(d) 4 ≤ TSEG1 ≤ 16 [3 ≤ Setting value of TSEG1[3:0] ≤ 15]
(e) 1 ≤ TSEG2 ≤ 8 [0 ≤ Setting value of TSEG2[2:0] ≤ 7]

Remark: TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)
TSEG1[3:0] (Bits 3 to 0 of CANn bit rate register (CnBTR))
TSEG2[2:0] (Bits 10 to 8 of CANn bit rate register (CnBTR))

Table 16-21 shows the combinations of bit rates that satisfy the above conditions.

Table 16-21: Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point Unit (%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8

Table 16-21: Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point Unit (%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9

Table 16-21: Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point Unit (%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7**Note**	1	2	2	2	0011	001	71.4
7**Note**	1	4	1	1	0100	000	85.7
6**Note**	1	1	2	2	0010	001	66.7
6**Note**	1	3	1	1	0011	000	83.3
5**Note**	1	2	1	1	0010	000	80.0
4**Note**	1	1	1	1	0001	000	75.0

Remark: n = 0, 1

Note: Setting with a DBT value of 7 or less is valid only when the value of the CnBRP register is other than 00H.

Caution: The values in Table 16-21 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

16.15.2 Representative examples of baud rate settings

Tables 16-22 and 16-23 show representative examples of baud rate settings.

Table 16-22: Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling Point Unit: (%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3

Table 16-22: Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling Point Unit: (%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Caution: The values in Table 16-22 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Remark: n = 0, 1

Table 16-23: Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling Point Unit: (%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0

Table 16-23: Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling Point Unit: (%)
			Length of DBT	SYNC SEG-MENT	PROP SEG-MENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

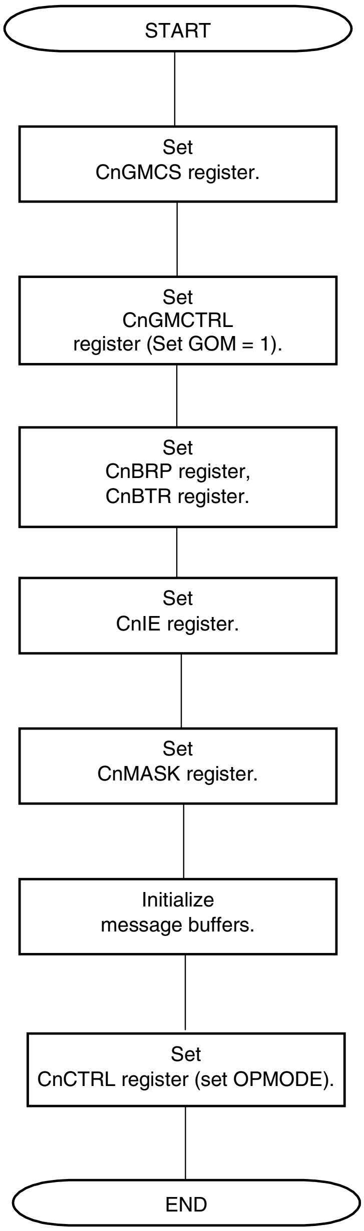
Caution: The values in Table 16-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Remark: n = 0, 1

16.16 Operation of CAN Controller

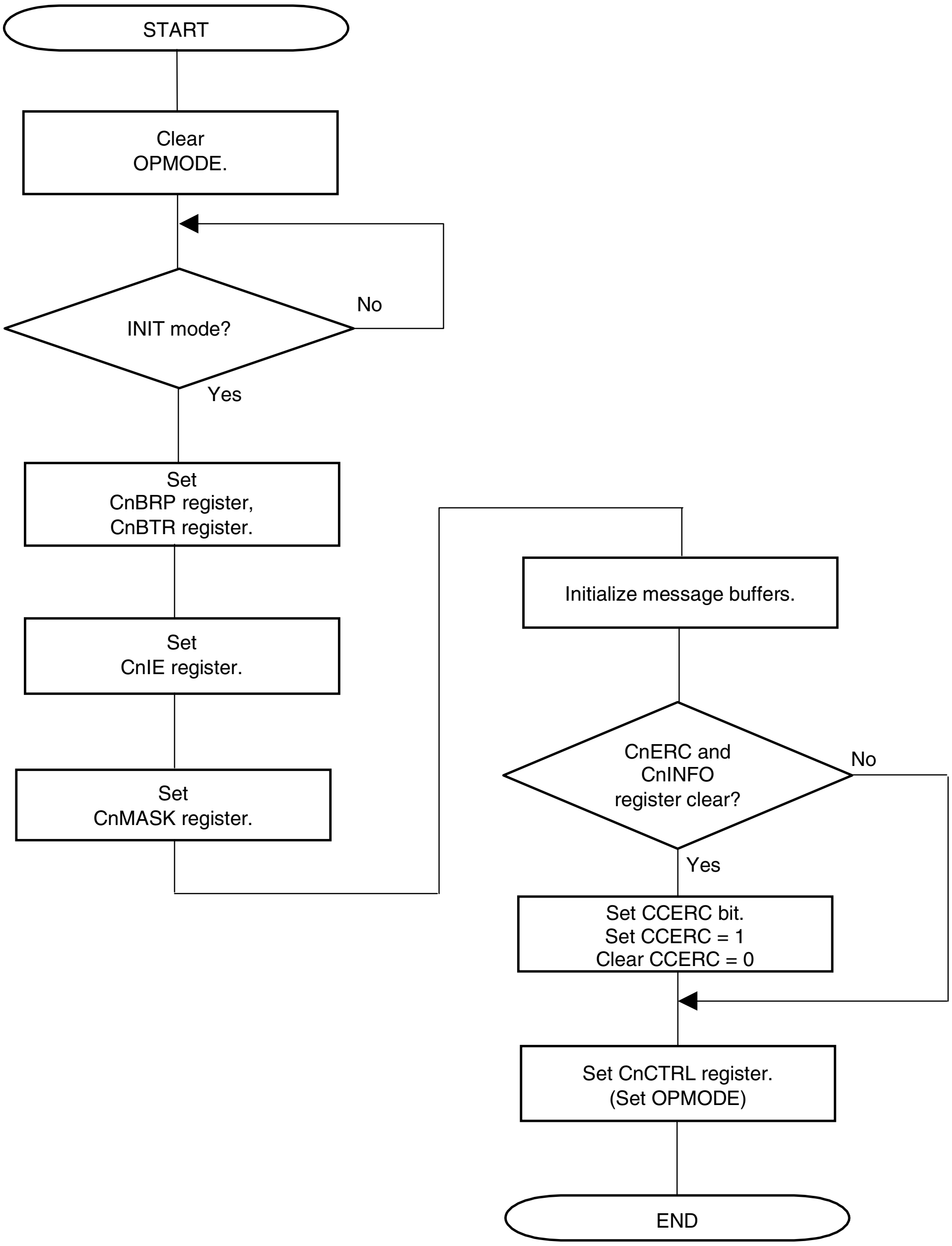
Remark: n = 0, 1
m = 0 to 31

Figure 16-57: Initialization



Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

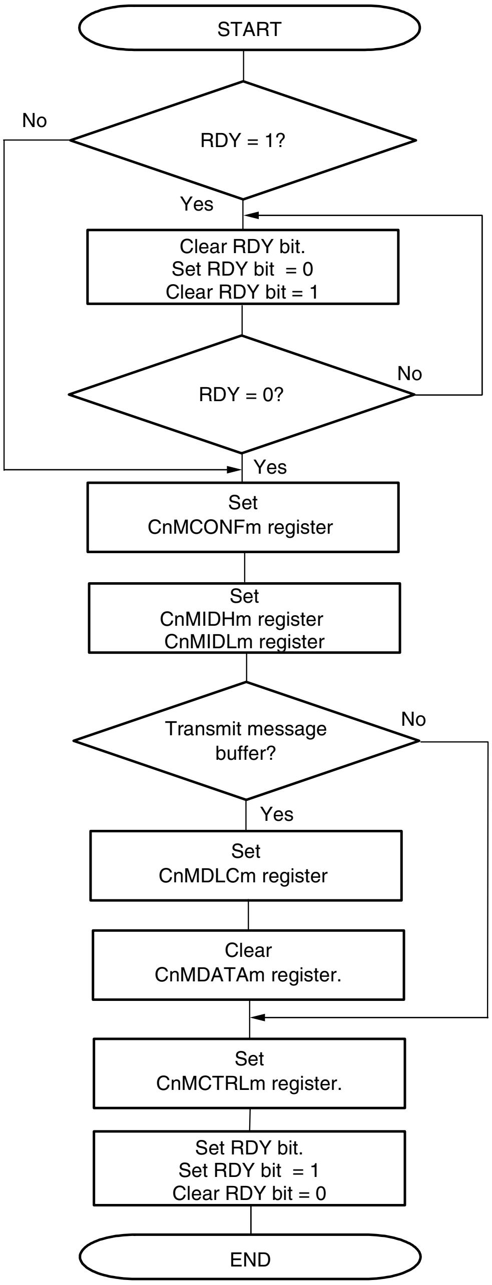
Figure 16-58: Re-initialization



Caution: After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the CnCTRL and CnGMCTRL registers (e.g., set a message buffer).

Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 16-59: Message Buffer Initialization



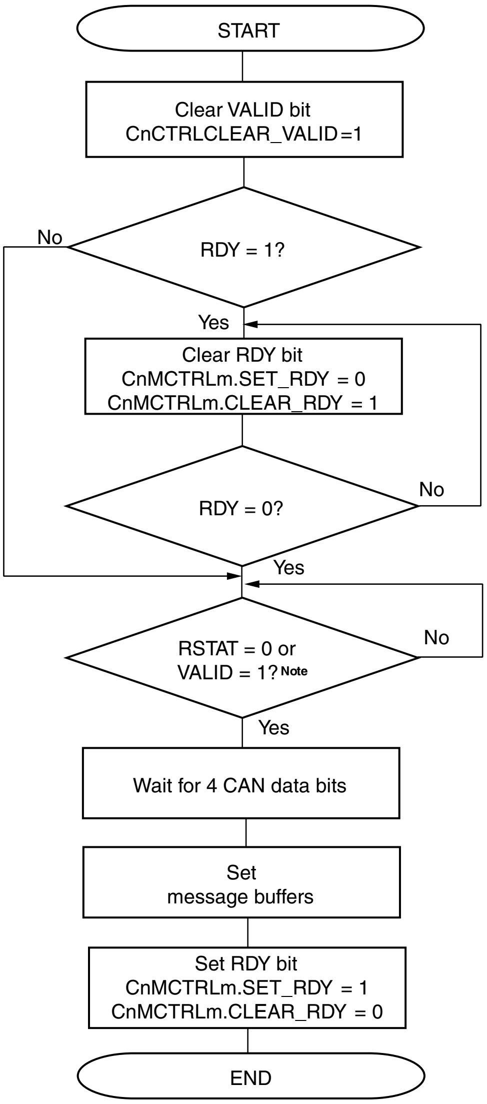
Cautions: 1. Before a message buffer is initialized, the RDY bit must be cleared.

2. Make the following settings for message buffers not used by the application.

- **Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.**
- **Clear the MA0 bit of the CnMCONFm register to 0.**

Figure 16-60 shows the processing for a receive message buffer (MT[2:0] bits of CnMCONFm register = 001B to 101B).

Figure 16-60: Message Buffer Redefinition



Note: If redefinition is performed during a message reception, confirm that a message is being received because the RDY bit must be set after a message is completely received.

Transmitting message buffer redefinition in the following according to cases, perform processing transmitting message buffer redefinition flow.

Figure 16-61: Transmitting Message Buffer Redefinition

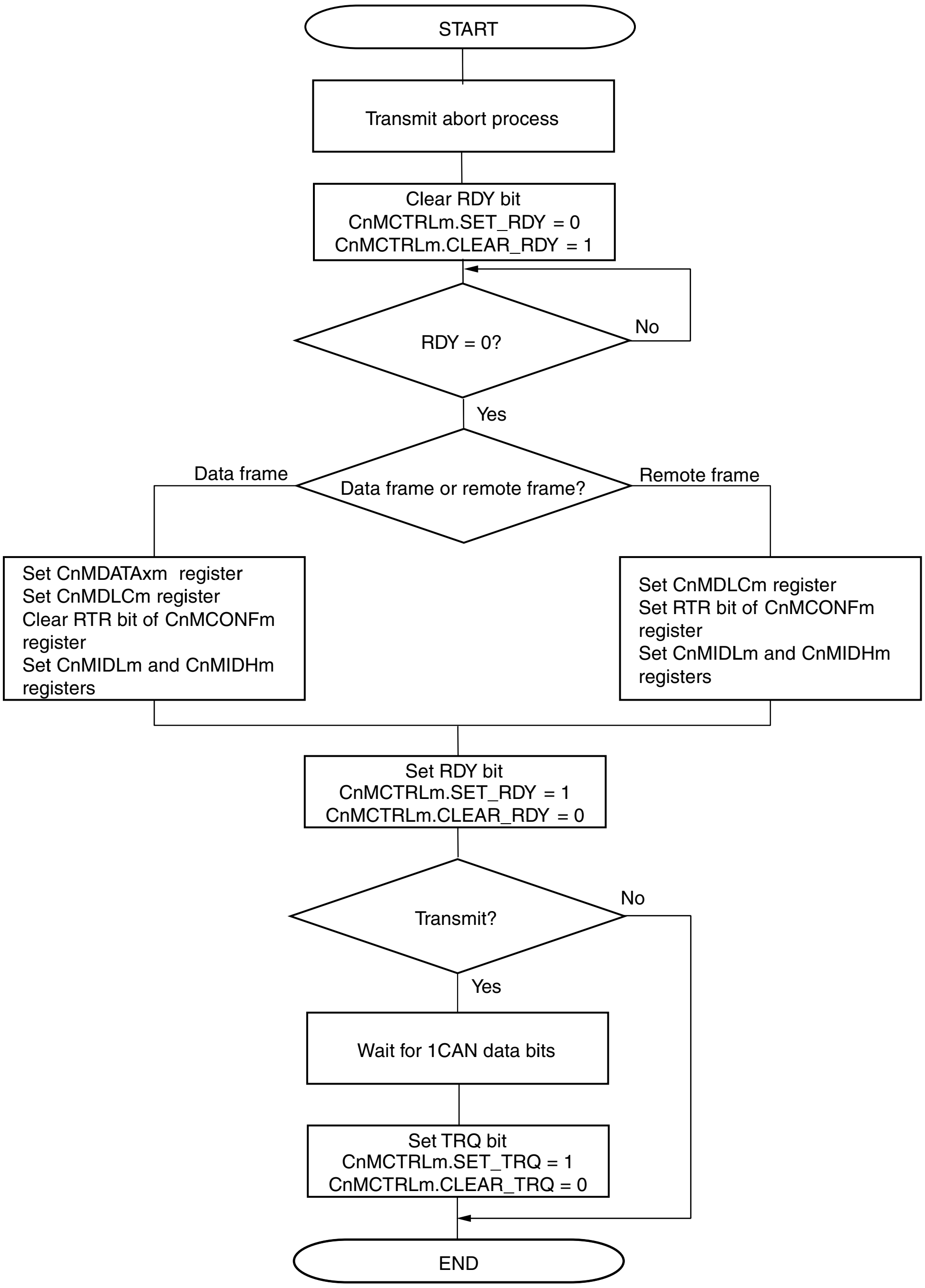
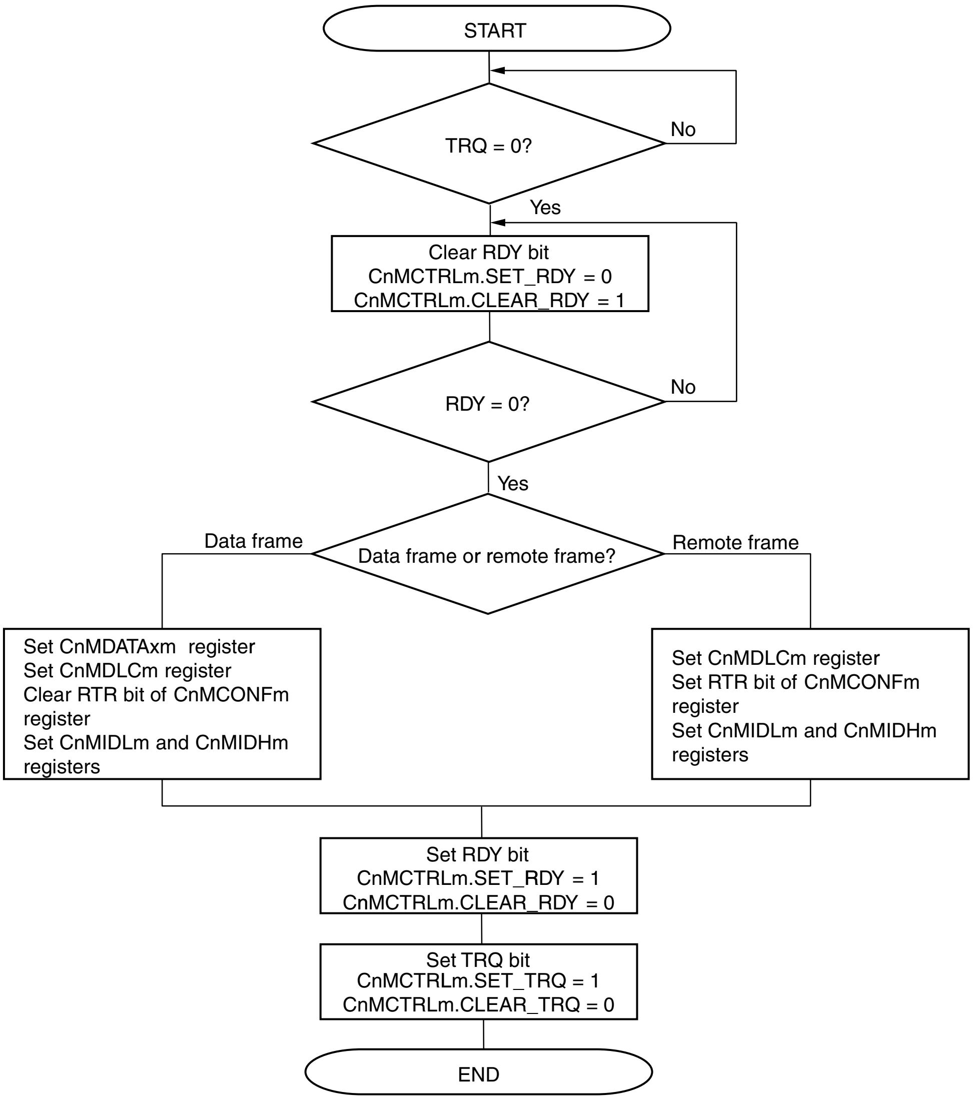


Figure 16-62 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000B).

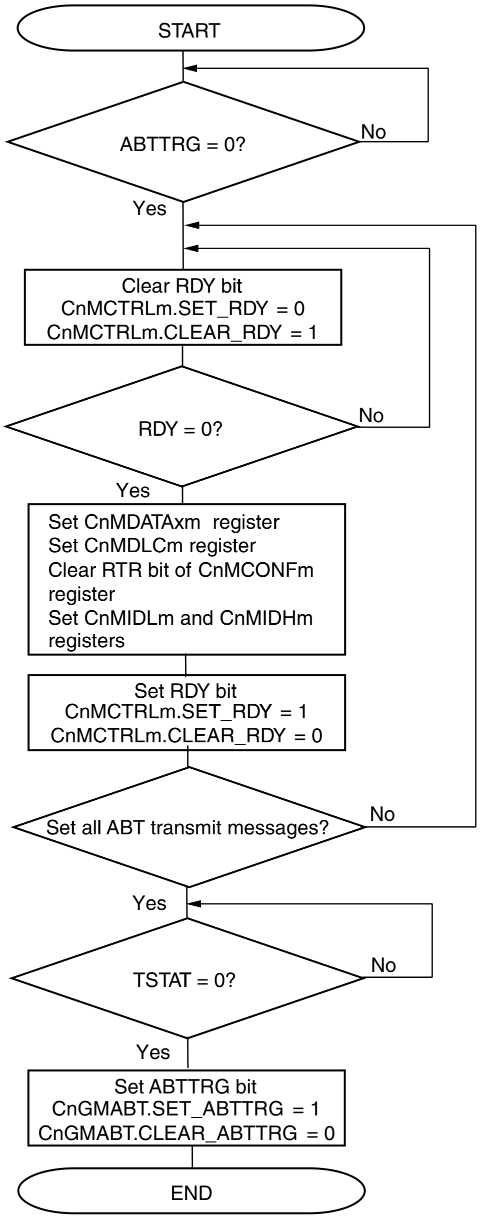
Figure 16-62: Message Transmit Processing (Normal Operation Mode)



Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 16-63 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000B).

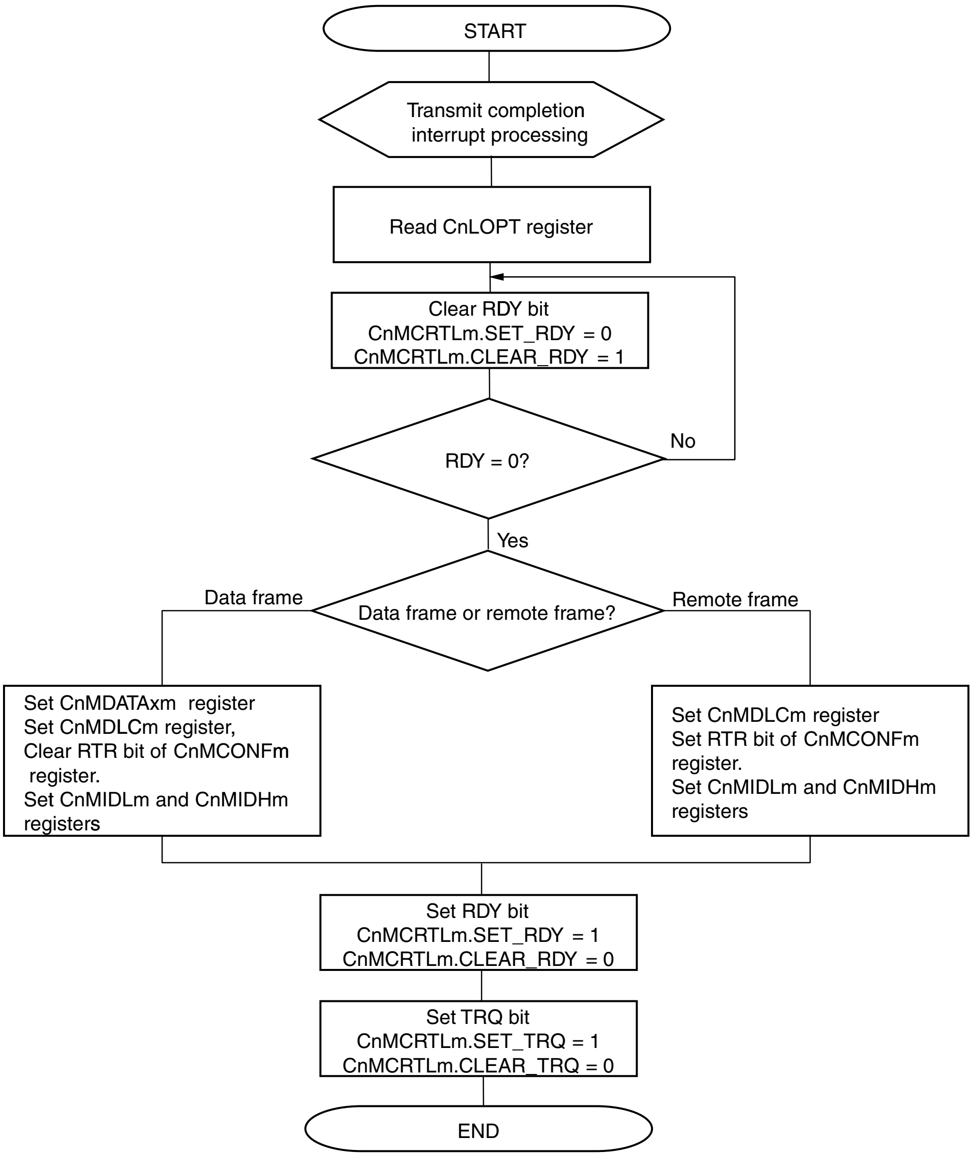
Figure 16-63: Message Transmit Processing (Normal Operation Mode with ABT)



Remark: This processing (normal operation mode with ABT) can only be applied to message buffers 0 to 7.
For message buffers other than the ABT message buffers, refer to **Figure 16-61**.

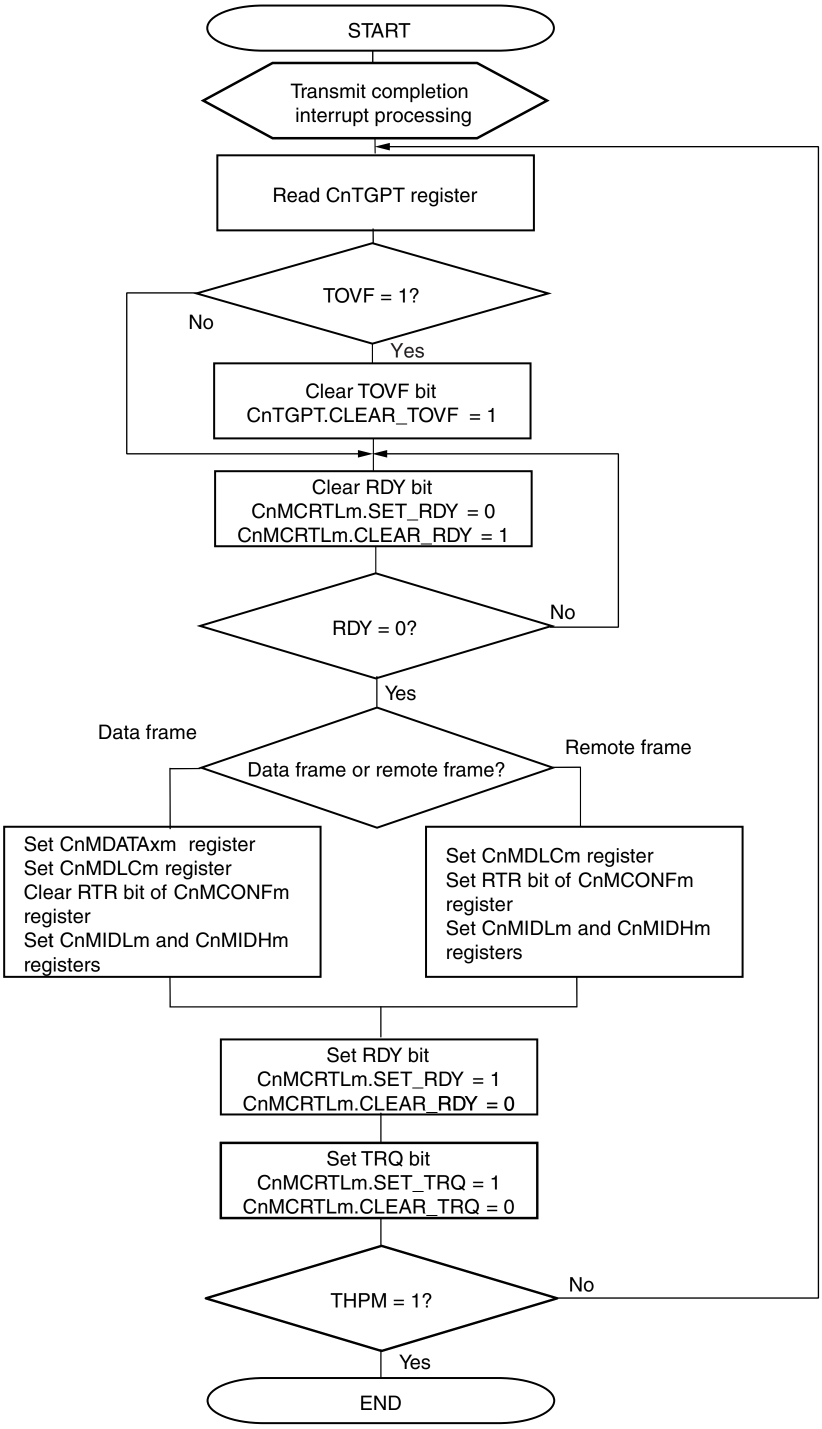
Caution: Set (1) ABTTRG bit after TSTAT bit is clear (0) check TSTAT bit and set ABTTRG bit, must be processing successively.

Figure 16-64: Transmission via Interrupt (Using CnLOPT register)



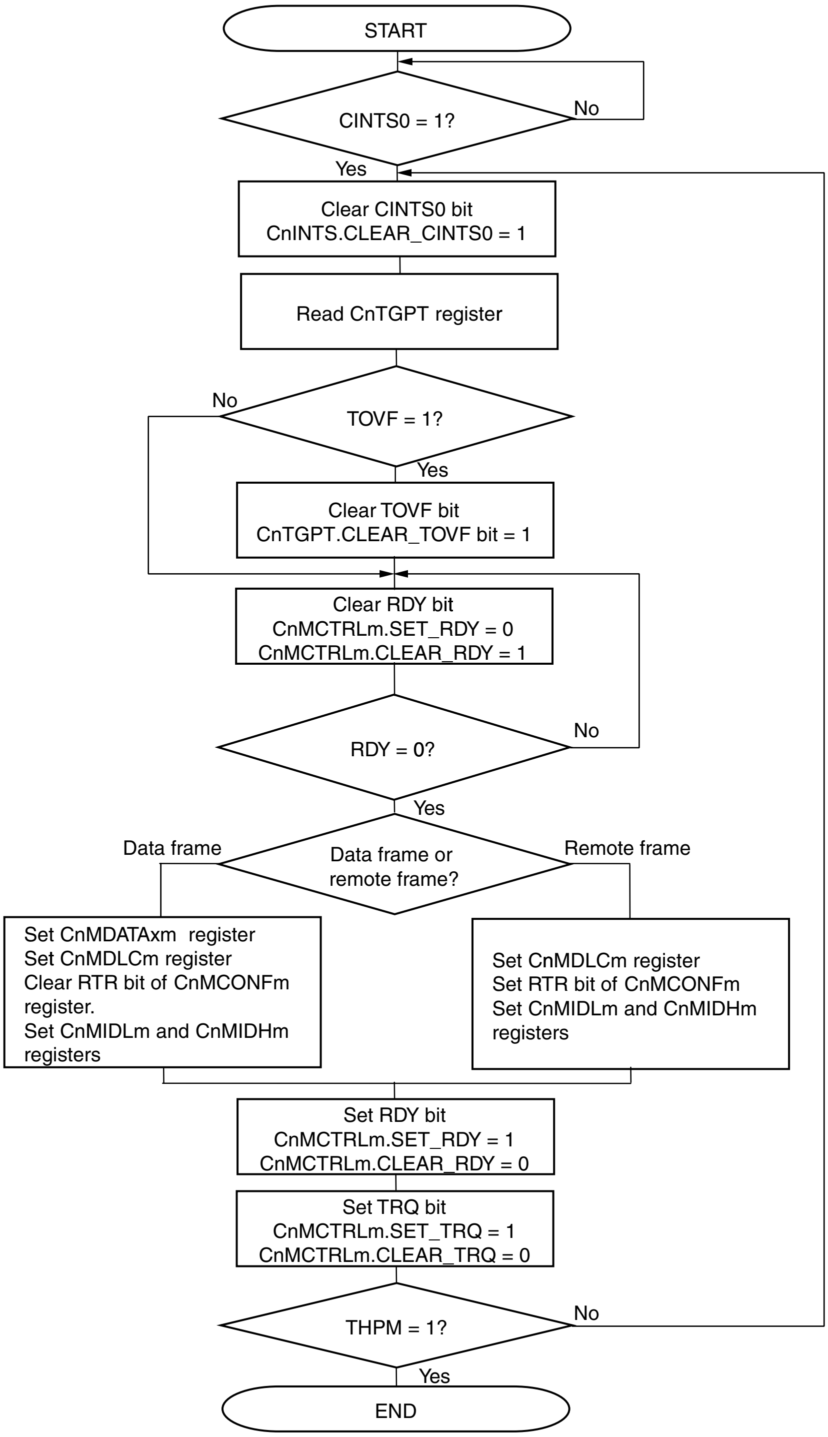
Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 16-65: Transmission via Interrupt (Using CnTGPT Register)



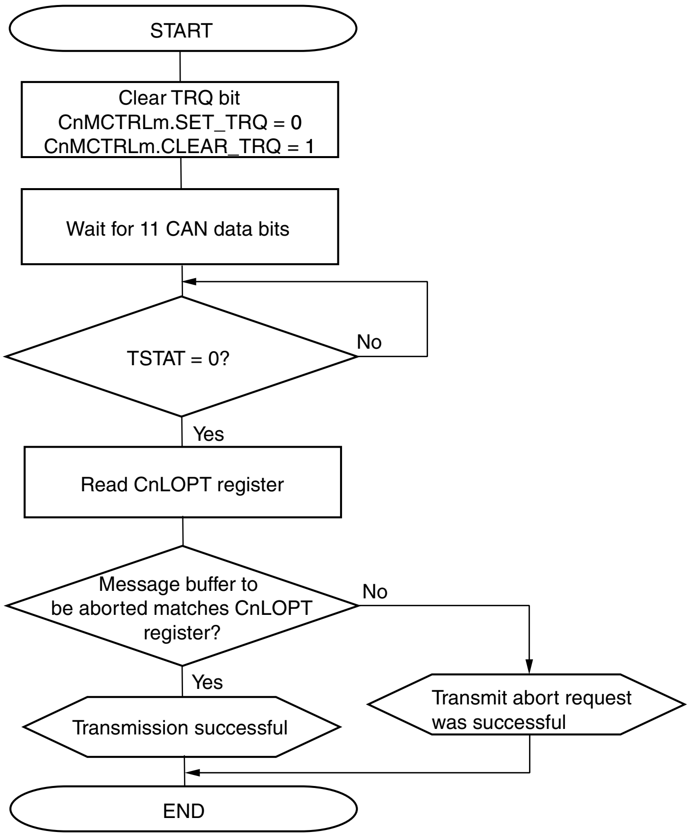
**Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.**

Figure 16-66: Transmission via Software Polling



Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 16-67: Transmission Abort Processing (except Normal Operation Mode with ABT)

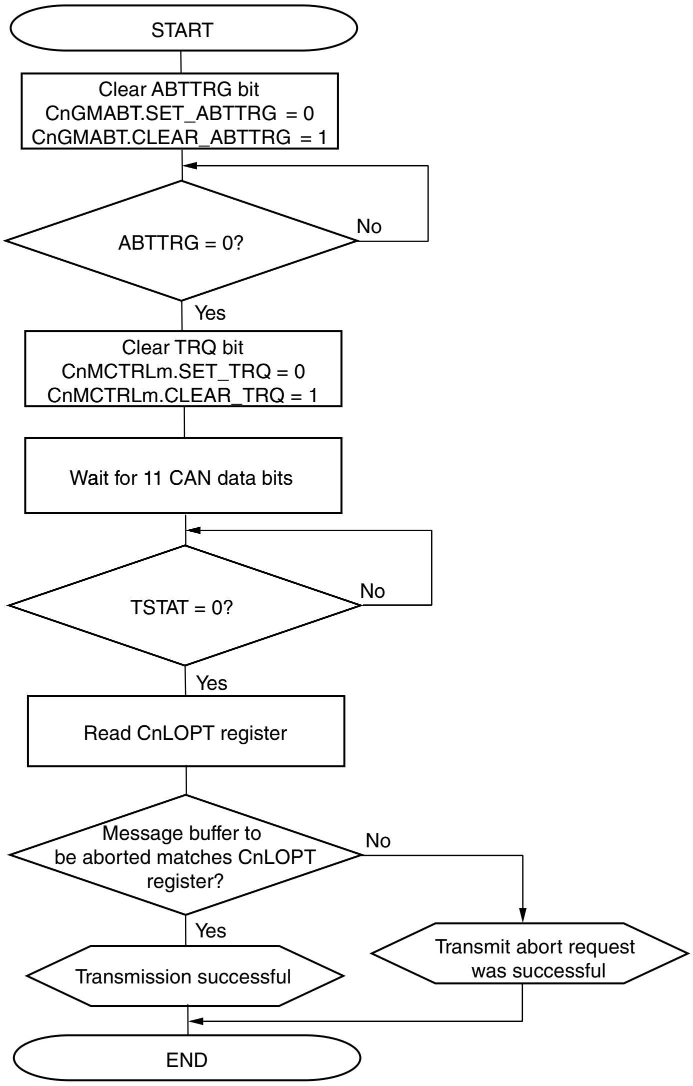


Cautions:

1. **Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.**
2. **Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.**
3. **The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.**
4. **Do not execute a new transmission request that includes other message buffers while transmission request abort processing is in progress.**

In the normal operation with ABT, to abort transmit except transmission with ABT, using this processing flow.

Figure 16-68: Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)



Cautions: 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

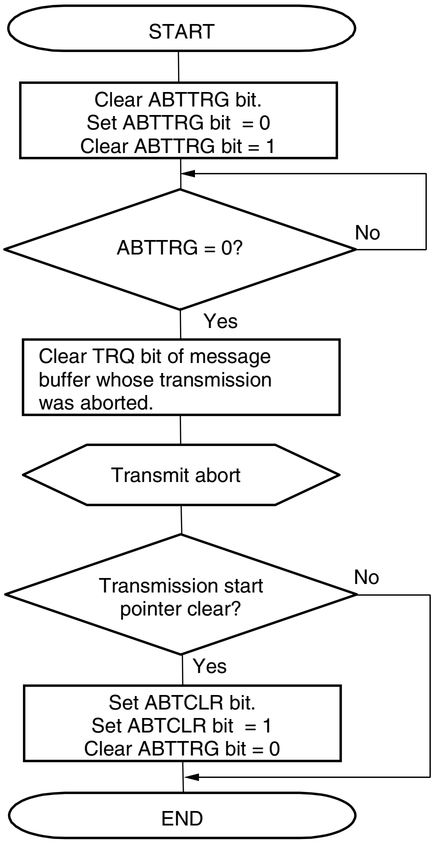
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute a new transmission request that includes other message buffers while transmission request abort processing is in progress.

Figure 16-69 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 16-69: Transmission Abort Processing (Normal Operation Mode with ABT)

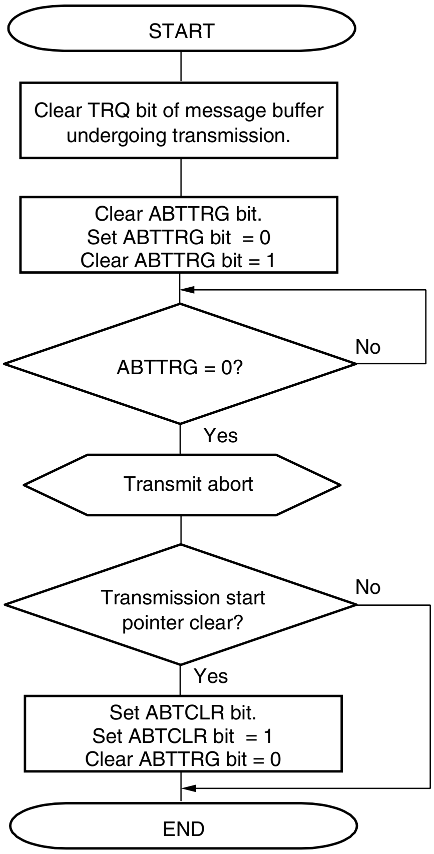


Cautions: 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG is cleared following the procedure shown in Figure 16-69 or Figure 16-70. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 16-67.

Figure 16-70 shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

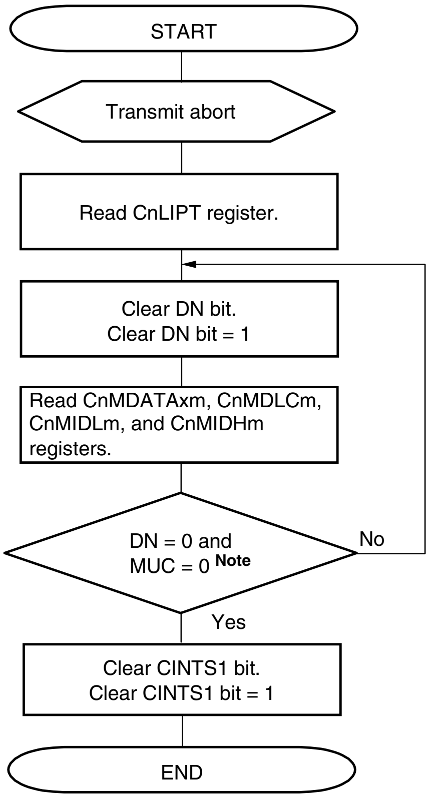
Figure 16-70: Transmission Request Abort Processing (Normal Operation Mode with ABT)



Cautions: 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

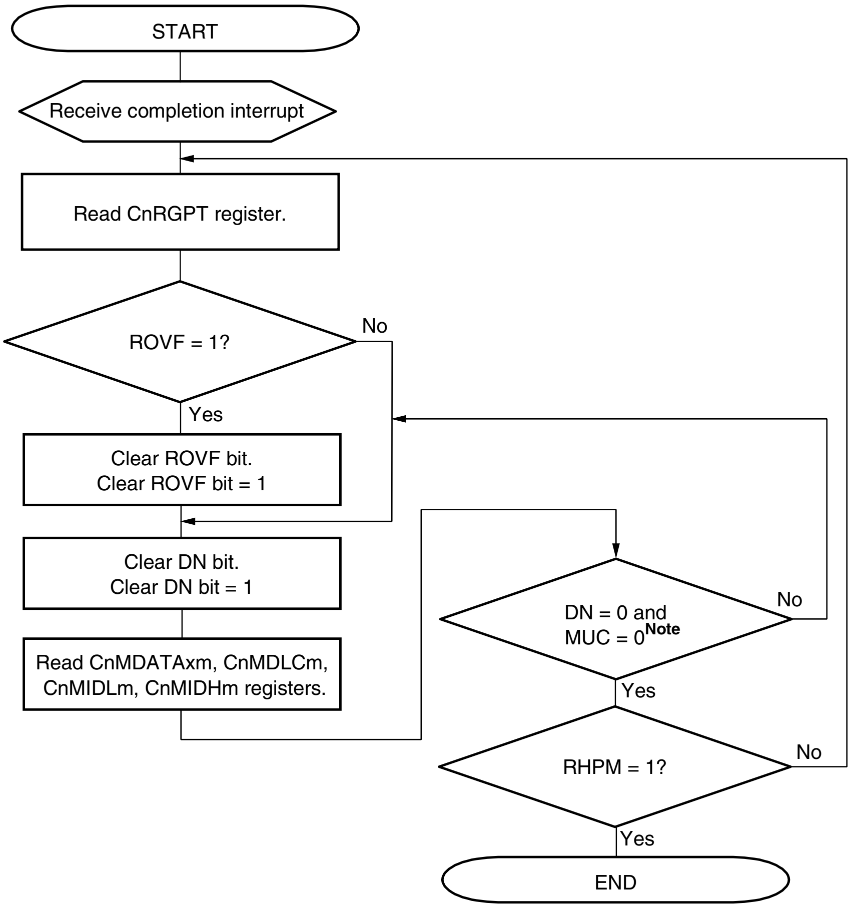
2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared following the procedure shown in Figure 16-69 or Figure 16-70. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 16-67.

Figure 16-71: Reception via Interrupt (Using CnLIPT Register)



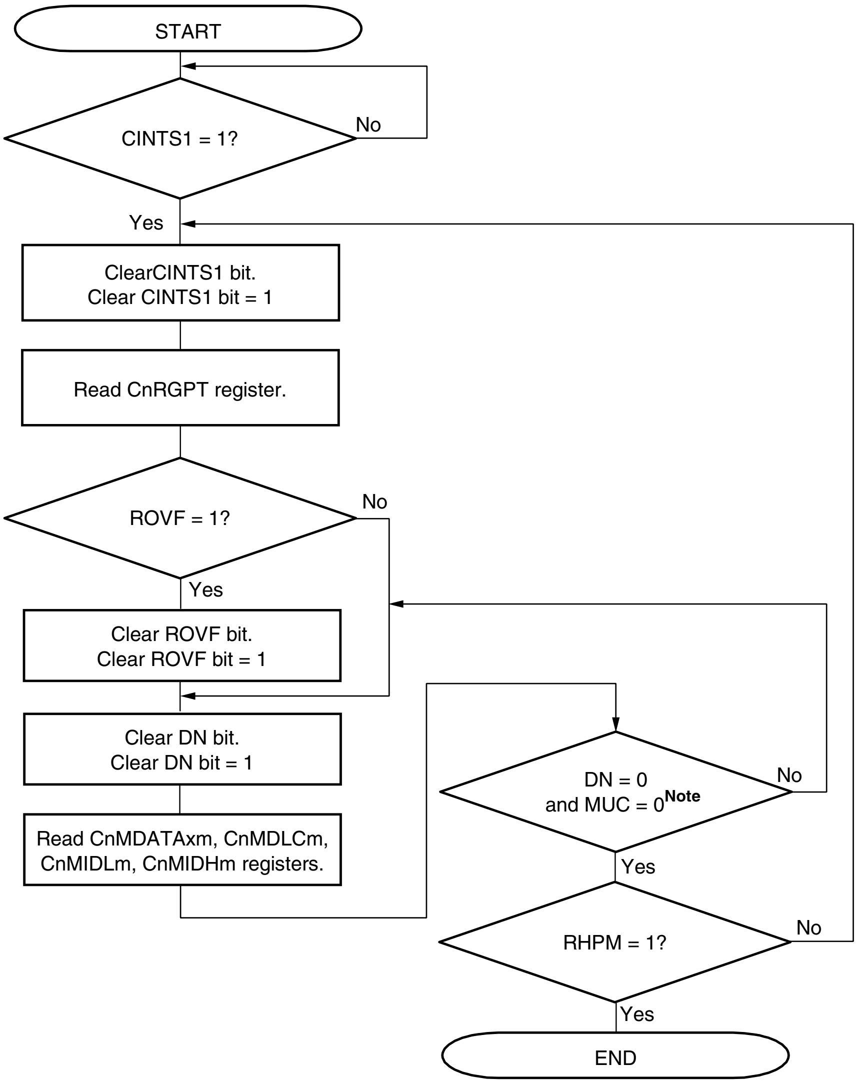
Note: Check the MUC and DN bits using one read access.

Figure 16-72: Reception via Interrupt (Using CnRGPT Register)



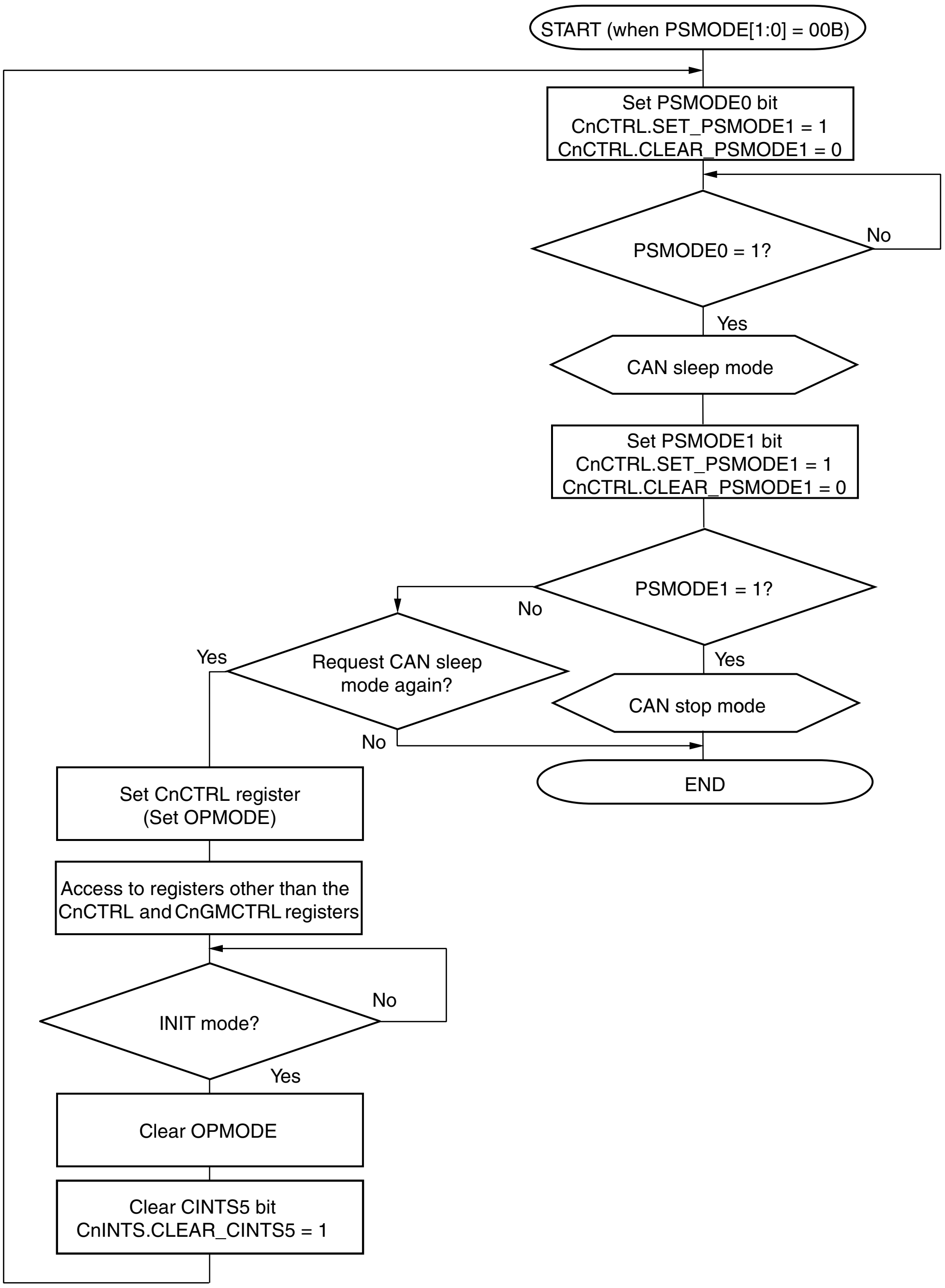
Note: Check the MUC and DN bits using one read access.

Figure 16-73: Reception via Software Polling



Note: Check the MUC and DN bits using one read access.

Figure 16-74: Setting CAN Sleep Mode/Stop Mode



Cautions: 1. To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 16-67 and 16-68.

2. If the host CPU wants to enter a power save mode as well, the interrupt processing needs to be disabled before the CPU validates that sleep mode has beenentered. If the interrupt processing can not be disabled, the host CPU will never wake-up by CAN bus activity when the CAN sleep mode is released between validation of the sleep state and execution of the i.e. CPU HALT instruction.

Figure 16-75: Clear CAN Sleep/Stop Mode

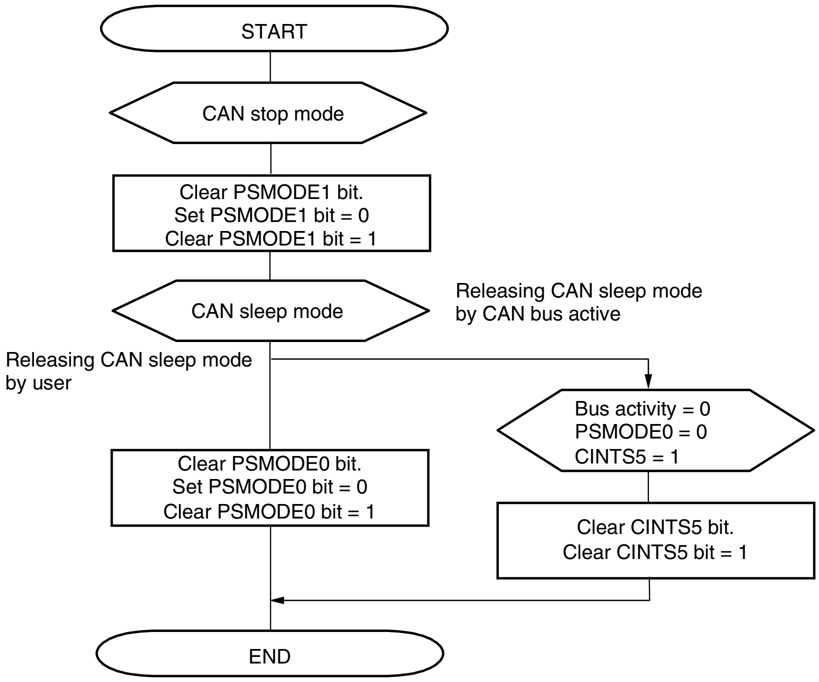
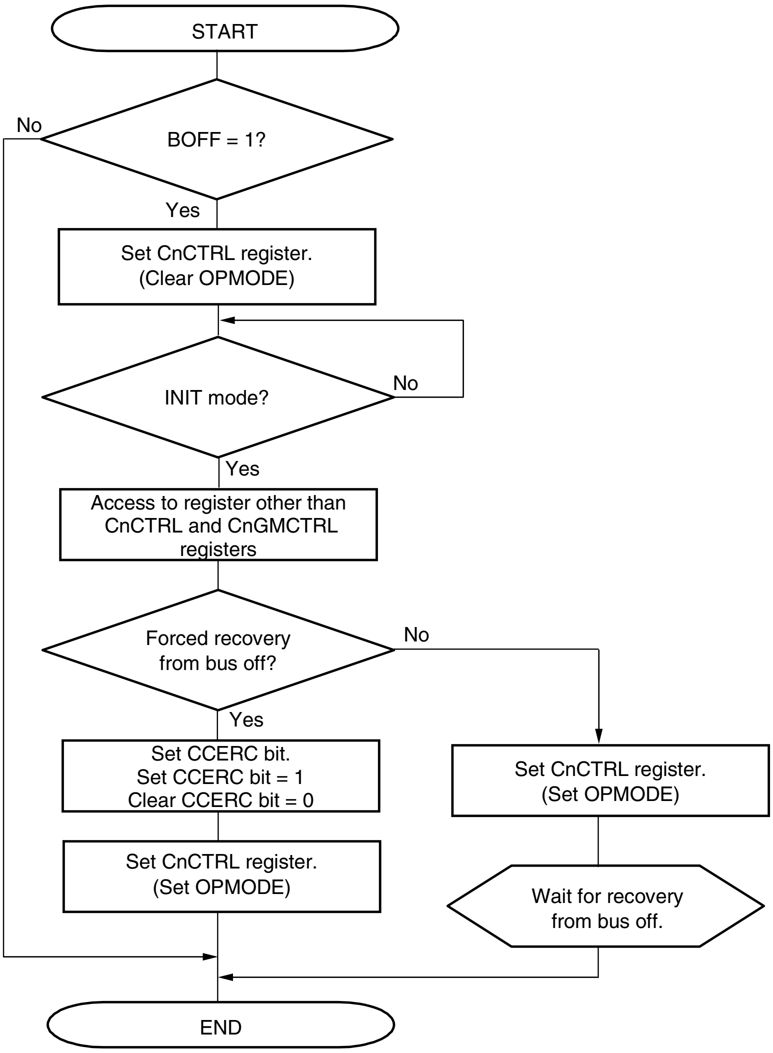


Figure 16-76: Bus-Off Recovery



Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode.

Figure 16-77: Normal Shutdown Process

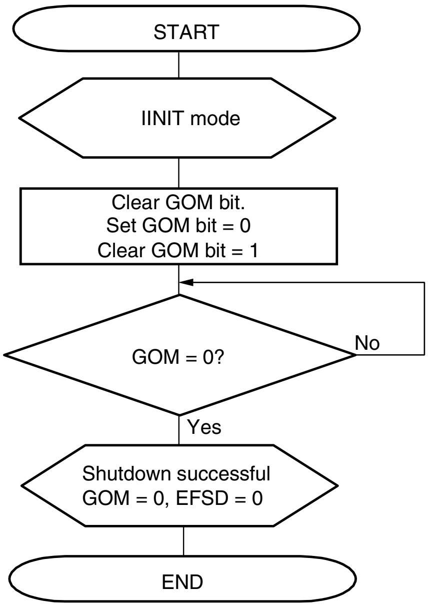
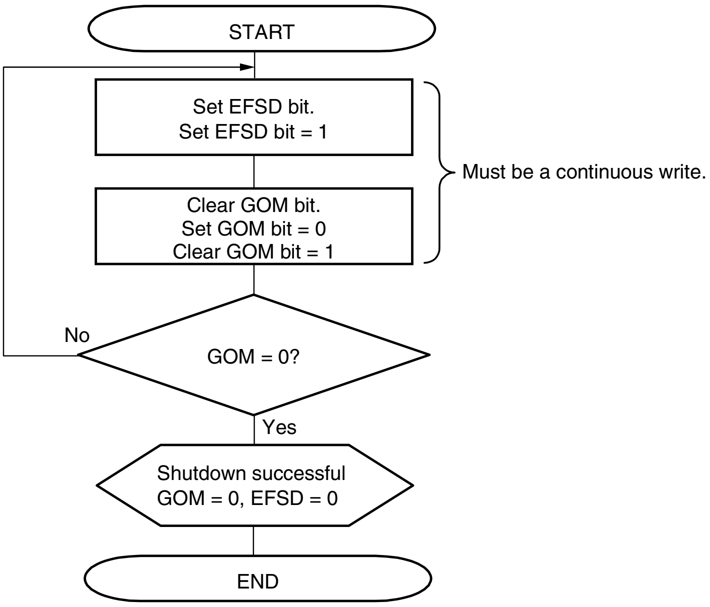
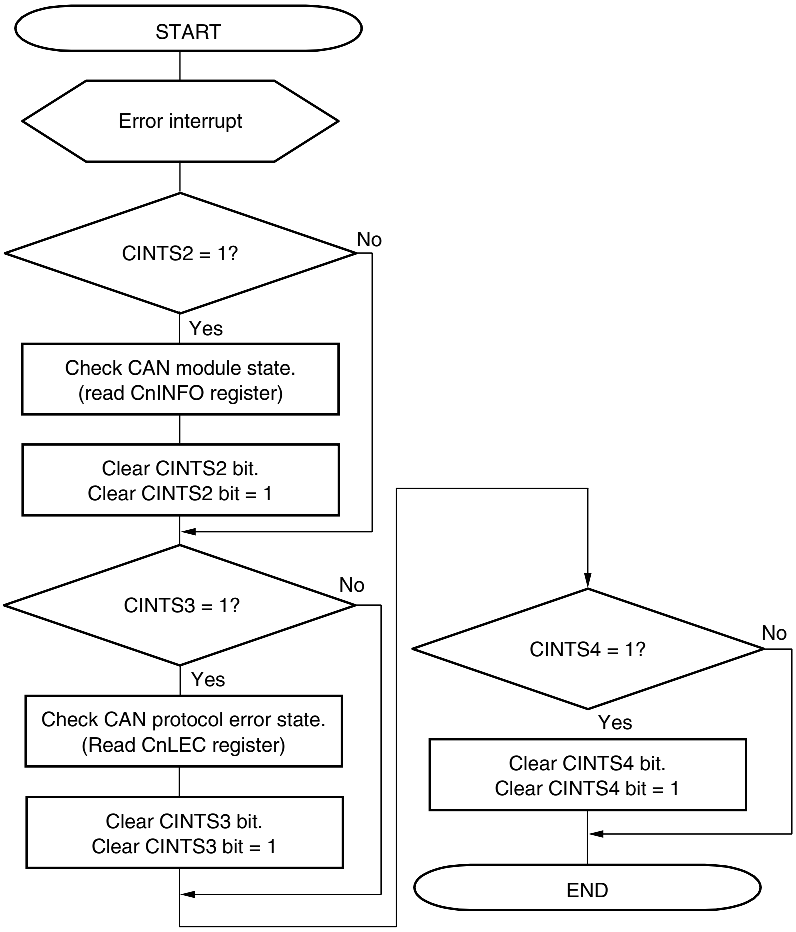


Figure 16-78: Forced Shutdown Process



Caution: Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

Figure 16-79: Error Handling



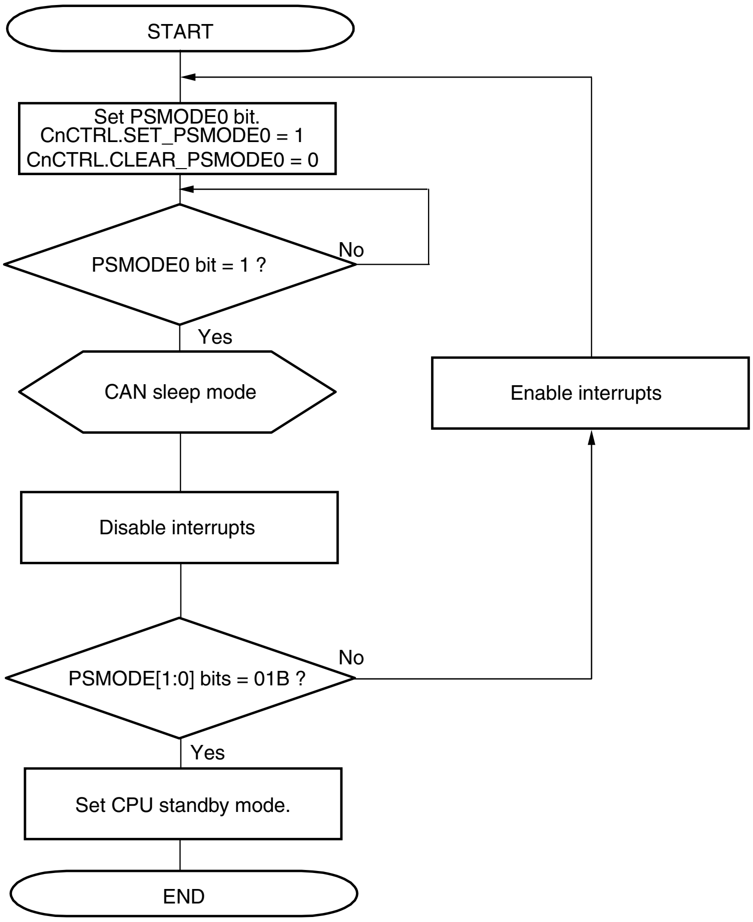
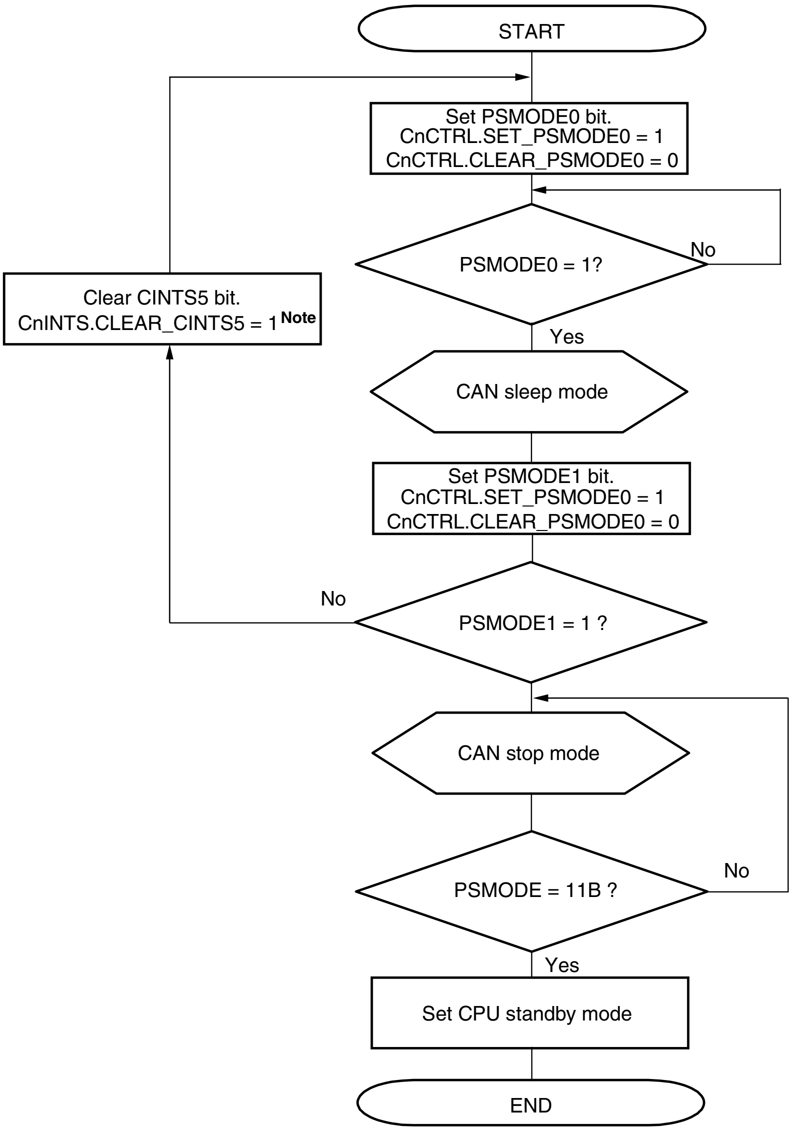


Figure 16-80: Setting CPU Standby (from CAN Sleep Mode)

Figure 16-81: Setting CPU Standby (from CAN Stop Mode)



Note: During wake-up interrupts

Caution: The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] of the CnCTRL register and not by a change in the CAN bus state.

[MEMO]

Chapter 17 Interrupt/Exception Processing Function

The V850E/RS1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 69 interrupt requests.
An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.
The V850E/RS1 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

17.1 Features

- Interrupts

Non-maskable interrupts:	2 sources
Maskable interrupts:	External: 8, Internal: 59 sources

8 levels of programmable priorities (maskable interrupts)
Multiple interrupt control according to priority
Masks can be specified for each maskable interrupt request.
Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

- Exceptions

Software exceptions:	32 sources
Exception trap:	2 sources (illegal opcode exception)

Interrupt/exception sources are listed in Table 17-1.

Table 17-1: Interrupt/Exception Source List (1/3)

Type	Classifica-tion	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	-	RESET	RESET pin input or Internal RESET	RESET	0000H	00000000H	undef.	-
Non-maskable	Interrupt	-	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	nextPC	-
Non-maskable	Interrupt	-	NMI2	NMI pin valid edge input	PORT	0030H	00000030H	nextPC	-
Software exception	Exception	-	TRAP0n	TRAP instruction	-	004nH	00000040H	nextPC	-
Software exception	Exception	-	TRAP1n	TRAP instruction	-	005nH	00000050H	nextPC	-
Excep-tion trap	Exception	-	ILGOP/ DBG0	Illegal open code/ DBTRAP instruction	-	0060H	00000060H	nextPC	-
Maskable	Interrupt	0	INTLVI	Low voltage Indicator interrupt	POCLVI	0080H	00000080H	nextPC	LVIIC
Maskable	Interrupt	1	INTP0	INTP0 pin valid edge input	PORT	0090H	00000090H	nextPC	PIC0
Maskable	Interrupt	2	INTP1	INTP1 pin valid edge input	PORT	00A0H	000000A0H	nextPC	PIC1
Maskable	Interrupt	3	INTP2	INTP2 pin valid edge input	PORT	00B0H	000000B0H	nextPC	PIC2
Maskable	Interrupt	4	INTP3	INTP3 pin valid edge input	PORT	00C0H	000000C0H	nextPC	PIC3
Maskable	Interrupt	5	INTP4	INTP4 pin valid edge input	PORT	00D0H	000000D0H	nextPC	PIC4

Table 17-1: Interrupt/Exception Source List (2/3)

Type	Classifica-tion	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	6	INTP5	INTP5 pin valid edge input	PORT	00E0H	000000E0H	nextPC	PIC5
Maskable	Interrupt	7	INTP6	INTP6 pin valid edge input	PORT	00F0H	000000F0H	nextPC	PIC6
Maskable	Interrupt	8	INTP7	INTP7 pin valid edge input	PORT	0100H	00000100H	nextPC	PIC7
Maskable	Interrupt	9	INTTQ0OV	TMQ0 overflow	TMQ0	0110H	00000110H	nextPC	TQ0OVIC
Maskable	Interrupt	10	INTTQ0CC0	TMQ0 capture0 trigger input/ CMP0 match	TMQ0	0120H	00000120H	nextPC	TQ0CCIC0
Maskable	Interrupt	11	INTTQ0CC1	TMQ0 capture1 trigger input/ RELD1 match	TMQ0	0130H	00000130H	nextPC	TQ0CCIC1
Maskable	Interrupt	12	INTTQ0CC2	TMQ0 capture2 trigger input/ RELD2 match	TMQ0	0140H	00000140H	nextPC	TQ0CCIC2
Maskable	Interrupt	13	INTTQ0CC3	TMQ0 capture3 trigger input/ RELD3 match	TMQ0	0150H	00000150H	nextPC	TQ0CCIC3
Maskable	Interrupt	14	INTTP0OV	TMP0 overflow	TMP0	0160H	00000160H	nextPC	TP0OVIC
Maskable	Interrupt	15	INTTP0CC0	TMP0 capture0 trigger input/ RELD0 match	TMP0	0170H	00000170H	nextPC	TP0CCIC0
Maskable	Interrupt	16	INTTP0CC1	TMP0 capture1 trigger input/ RELD1 match	TMP0	0180H	00000180H	nextPC	TP0CCIC1
Maskable	Interrupt	17	INTTP1OV	TMP1 overflow	TMP1	0190H	00000190H	nextPC	TP1OVIC
Maskable	Interrupt	18	INTTP1CC0	TMP1 capture0 trigger input/ RELD0 match	TMP1	01A0H	000001A0H	nextPC	TP1CCIC0
Maskable	Interrupt	19	INTTP1CC1	TMP1 capture1 trigger input/ RELD1 match	TMP1	01B0H	000001B0H	nextPC	TP1CCIC1
Maskable	Interrupt	20	INTTP2OV	TMP2 overflow	TMP2	01C0H	000001C0H	nextPC	TP2OVIC
Maskable	Interrupt	21	INTTP2CC0	TMP2 capture0 trigger input/ RELD0 match	TMP2	01D0H	000001D0H	nextPC	TP2CCIC0
Maskable	Interrupt	22	INTTP2CC1	TMP2 capture1 trigger input/ RELD1 match	TMP2	01E0H	000001E0H	nextPC	TP2CCIC1
Maskable	Interrupt	23	INTTP3OV	TMP3 overflow	TMP3	01F0H	000001F0H	nextPC	TP3OVIC
Maskable	Interrupt	24	INTTP3CC0	TMP3 capture0 trigger input/ RELD0 match	TMP3	0200H	00000200H	nextPC	TP3CCIC0
Maskable	Interrupt	25	INTTP3CC1	TMP3 capture1 trigger input/ RELD1 match	TMP3	0210H	00000210H	nextPC	TP3CCIC1
Maskable	Interrupt	26	INTTM0EQ0	TMM0 compare match	TMM0	0220H	00000220H	nextPC	TM0EQIC0
Maskable	Interrupt	27	INTCB0R	CSIB0 reception completion/ error occurrence	CSIB0	0230H	00000230H	nextPC	CB0RIC
Maskable	Interrupt	28	INTCB0T	CSIB0 continuous transfer write enable	CSIB0	0240H	00000240H	nextPC	CB0TIC
Maskable	Interrupt	29	INTCB1R	CSIB1 transfer completion/ error occurrence	CSIB1	0250H	00000250H	nextPC	CB1RIC
Maskable	Interrupt	30	INTCB1T	CSIB1 continuous transfer write enable	CSIB1	0260H	00000260H	nextPC	CB1TIC
Maskable	Interrupt	31	INTUA0R	UARTA0 reception completion/ error occurrence	UARTA0	0270H	00000270H	nextPC	UA0RIC
Maskable	Interrupt	32	INTUA0T	UARTA0 transfer completion	UARTA0	0280H	00000280H	nextPC	UA0TIC
Maskable	Interrupt	33	INTUA1R	UARTA1 reception completion/ error occurrence	UARTA1	0290H	00000290H	nextPC	UA1RIC

Table 17-1: Interrupt/Exception Source List (3/3)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	34	INTUA1T	UARTA1 transfer completion	UARTA1	02A0H	000002A0H	nextPC	UA1TIC
Maskable	Interrupt	35	INTAD	AD conversion completion	AD	02B0H	000002B0H	nextPC	ADIC
Maskable	Interrupt	36	INTC0ERR	AFCAN0 error occurrence	AFCAN0	02C0H	000002C0H	nextPC	C0ERRIC
Maskable	Interrupt	37	INTC0WUP	AFCAN0 wake up request	AFCAN0	02D0H	000002D0H	nextPC	C0WUPIC
Maskable	Interrupt	38	INTC0REC	AFCAN0 reception completion	AFCAN0	02E0H	000002E0H	nextPC	C0RECIC
Maskable	Interrupt	39	INTC0TRX	AFCAN0 transfer completion	AFCAN0	02F0H	000002F0H	nextPC	C0TRXIC
Maskable	Interrupt	40	INTC30I	CSI30 interrupt	CSI30	0300H	00000300H	nextPC	C30IC
Maskable	Interrupt	41	INTC30O	CSI30 overflow	CSI30	0310H	00000310H	nextPC	C30OC
Maskable	Interrupt	42	INTC31I	CSI31 interrupt	CSI31	0320H	00000320H	nextPC	C31IC
Maskable	Interrupt	43	INTC31O	CSI31 overflow	CSI31	0330H	00000330H	nextPC	C31OC
Maskable	Interrupt	44	INTTQ1OV	TMQ1 overflow	TMQ1	0360H	00000360H	nextPC	TQ1OVIC
Maskable	Interrupt	45	INTTQ1CC0	TMQ1 capture0 trigger input	TMQ1	0370H	00000370H	nextPC	TQ1CCIC0
Maskable	Interrupt	46	INTTQ1CC1	TMQ1 capture1 trigger input	TMQ1	0380H	00000380H	nextPC	TQ1CCIC1
Maskable	Interrupt	47	INTTQ1CC2	TMQ1 capture2 trigger input	TMQ1	0390H	00000390H	nextPC	TQ1CCIC2
Maskable	Interrupt	48	INTTQ1CC3	TMQ1 capture3 trigger input	TMQ1	03A0H	000003A0H	nextPC	TQ1CCIC3
Maskable	Interrupt	49	INTC1ERR	AFCAN1 error occurrence	AFCAN1	03D0H	000003D0H	nextPC	C1ERRIC
Maskable	Interrupt	50	INTC1WUP	AFCAN1 wake up request	AFCAN1	03E0H	000003E0H	nextPC	C1WUPIC
Maskable	Interrupt	51	INTC1REC	AFCAN1 reception completion	AFCAN1	03F0H	000003F0H	nextPC	C1RECIC
Maskable	Interrupt	52	INTC1TRX	AFCAN1 transfer completion	AFCAN1	0400H	00000400H	nextPC	C1TRXIC
Maskable	Interrupt	53	INTDMA0	DMA0 transfer completion	DMA	0410H	00000410H	nextPC	DMAIC0
Maskable	Interrupt	54	INTDMA1	DMA1 transfer completion	DMA	0420H	00000420H	nextPC	DMAIC1
Maskable	Interrupt	55	INTDMA2	DMA2 transfer completion	DMA	0430H	00000430H	nextPC	DMAIC2
Maskable	Interrupt	56	INTDMA3	DMA3 transfer completion	DMA	0440H	00000440H	nextPC	DMAIC3
Maskable	Interrupt	57	INTDMA4	DMA4 transfer completion	DMA	0450H	00000450H	nextPC	DMAIC4
Maskable	Interrupt	58	INTDMA5	DMA5 transfer completion	DMA	0460H	00000460H	nextPC	DMAIC5

Remarks: 1. Default Priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC or FEPC when interrupt processing 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).

17.2 Non-Maskable Interrupts

A non-maskable interrupt request 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 interrupts.
This product has the following two non-maskable interrupts.

- NMI pin input (NMI)
- Non-maskable interrupt request generated by overflow of watchdog timer (INTWDT2)

The valid edge of the NMI pin can be selected from three types: “rising edge”, “falling edge”, and “both edges”.
The non-maskable interrupt generated by overflow of the watchdog timer (INTWDT2) functions when the WDM21 and WDM20 bits of the watchdog timer mode register 2 (WDTM2) are set to “01”.
If two or more non-maskable interrupts occur at the same time, the interrupt with the higher priority is serviced, as follows (the interrupt with the lower priority is ignored).

NMI > INTWDT2

If a new NMI or INTWDT2 request is issued while a NMI is being serviced, it is serviced as follows.

(1) If new NMI request is issued while NMI is being serviced

The new NMI request is (strike through: held pending) serviced, regardless of the value of the NP bit of the program status word (PSW) in the CPU. The pending NMI interrupt is acknowledged after the NMI currently under execution has been serviced. The new NMI interrupts the current NMI routine.
A system reset has to be executed in this typical case of nested NMI interrupt.

(2) If INTWDT2 request is issued while NMI is being serviced

The INTWDT2 request is held pending. if the NP bit of the PSW remains set (1) while the NMI whatever the NP bit is while the NMI, which has an higher priority, is being serviced. The pending INTWDT2 request is acknowledged after the NMI currently under execution has been serviced.

Caution: If a non-maskable interrupt request is generated, the values of the PC and PSW are saved to the NMI status save registers (FEPC and FEPSW). Therefore, reset the system after the interrupt has been serviced.

Figure 17-1: Non-Maskable Interrupt Request Acknowledgement Operation (1/2)

(a) NMI and INTWDT2 requests generated at the same time

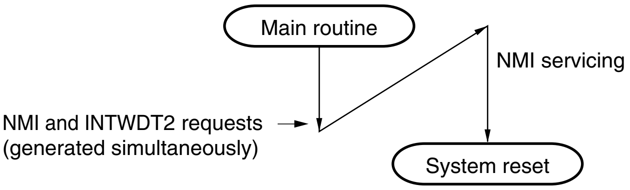
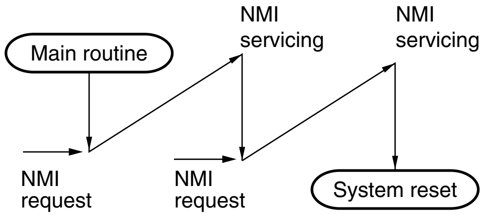
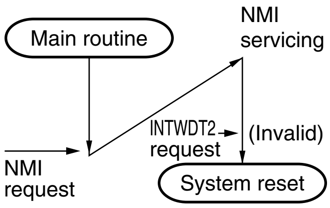
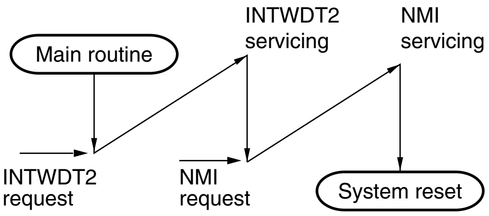
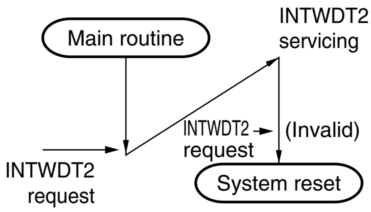


Figure 17-1: Non-Maskable Interrupt Request Acknowledgement Operation (2/2)

(b) Non-maskable interrupt request generated during non-maskable interrupt servicing

Non-maskable interrupt being servicing	Non-maskable interrupt request signal generated during non-maskable interrupt servicing	
	NMI	INTWDT2
NMI	• NMI request generated during NMI servicing  NMI servicing NMI servicing Main routine NMI request NMI request System reset	• INTWDT2 request generated during NMI servicing (NP = 1 retained before INTWDT2 request)  NMI servicing Main routine INTWDT2 request (Invalid) NMI request System reset
INTWDT2	• NMI request generated during INTWDT2 servicing  INTWDT2 servicing NMI servicing Main routine INTWDT2 request NMI request System reset	• INTWDT2 request generated during INTWDT2 servicing (NP = 1 retained before INTWDT2 request)  INTWDT2 servicing Main routine INTWDT2 request (Invalid) INTWDT2 request System reset

Cautions: 1. NMI signal must not input during NMI servicing. The CPU always accepts the NMI pin input, even if NP bit is set.

2. In NMI routine, Software must not operate the PSW: NP_bit. If the NP bit is cleared by software, the CPU is able to accept the INTWDT2 interrupt and NMI pin input.

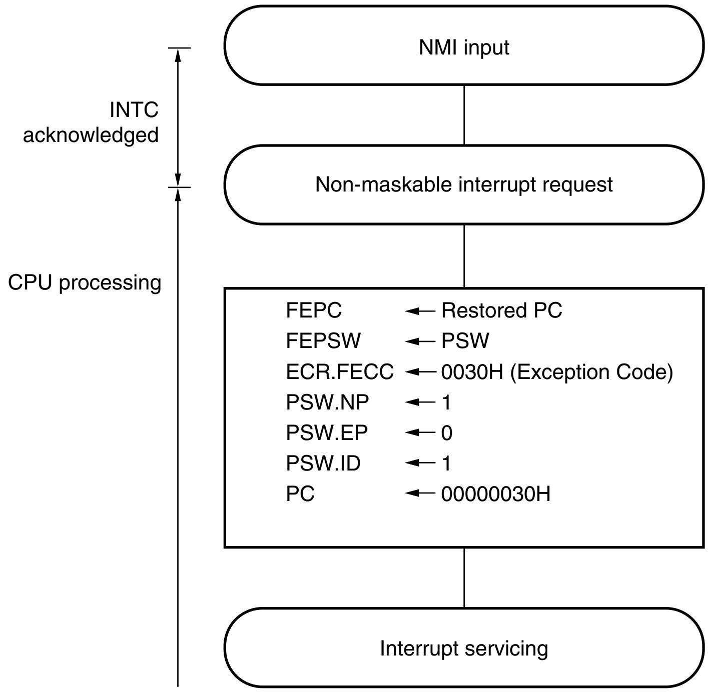
17.2.1 Operation

If a non-maskable interrupt is generated by NMI input, 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 to the higher halfword (FECC) of ECR.
<4> Sets the NP and ID bits of the PSW and clears the EP bit.
<5> Sets the handler address (00000030H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing configuration of a non-maskable interrupt is shown in Figure 17-2.

Figure 17-2: Servicing Configuration of Non-Maskable Interrupt



17.2.2 Restore

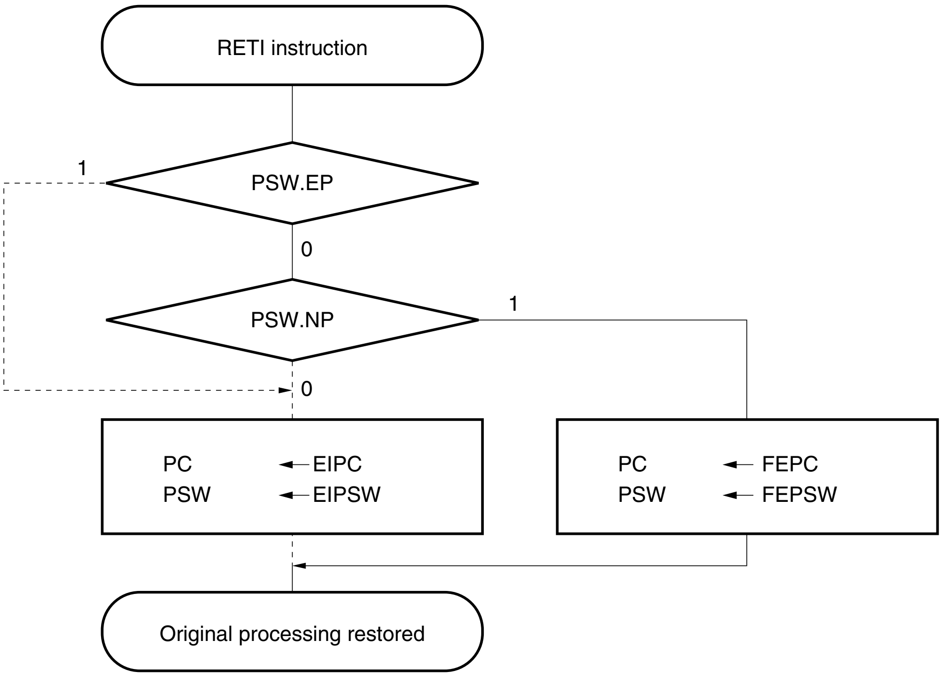
(1) From NMI

Execution is restored from the NMI 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 EP bit of the PSW is 0 and the NP bit of the PSW is 1.
<2> Transfers control back to the address of the restored PC and PSW.

Figure 17-3 illustrates how the RETI instruction is processed.

Figure 17-3: RETI Instruction Processing



Caution: When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid line shows the CPU processing flow.

(2) From INTWDT2

Execution cannot be returned from INTWDT by the RETI instruction. Execute a system reset after the interrupt has been serviced.

17.2.3 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 has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

Figure 17-4: NP Flag Format

After reset: 00000020H

<table>
<tr><td></td><td>31</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>PSW</td><td colspan="2">0</td><td>NP</td><td>EP</td><td>ID</td><td>SAT</td><td>CY</td><td>OV</td><td>S</td><td>Z</td></tr>
</table>

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt currently being serviced

17.2.4 Eliminating noise on NMI pin

The NMI pin has a noise eliminator that eliminates noise using analog delay. Unless the level input to the NMI pin is held for a specific time, therefore, it cannot be detected as an edge i.e., the edge is detected after specific time.
The NMI pin is used to release the software STOP mode. Because the internal system clock is stopped in the software STOP mode, digital noise elimination (over-sampling) by the system clock is not implemented.

17.2.5 Function to detect edge of NMI pin

The valid edge of the NMI pin can be selected from three types: “rising edge”, “falling edge”, and “both edges”.
Specify the valid edge of the non-maskable interrupt (NMI) by using the NMI mode register (NMIM). This register can be read or written in only 8-bit units, and can be written only once after each $\overline{\text{RESET}}$ condition.

Figure 17-5: NMI Mode Register (NMIM) Format

After reset: 03H R/W Address: FFFFFC08H

	7	6	5	4	3	2	1	0
NMIM	PNMI	0	0	0	0	0	ESN1	ESN0

ESN1	ESN0	NMI Edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Both edges
1	1	Both edges

PNMI	NMI Pin level Monitor
0	NMI pin 0 level
1	NMI pin 1 level

17.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850E/RS1 has 59 maskable interrupt sources.

If two or more maskable interrupt requests 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 has been acknowledged, the acknowledgement of other maskable interrupt requests 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 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 registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

17.3.1 Operation

If a maskable interrupt occurs by INT input, 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 half-word of ECR (EICC).

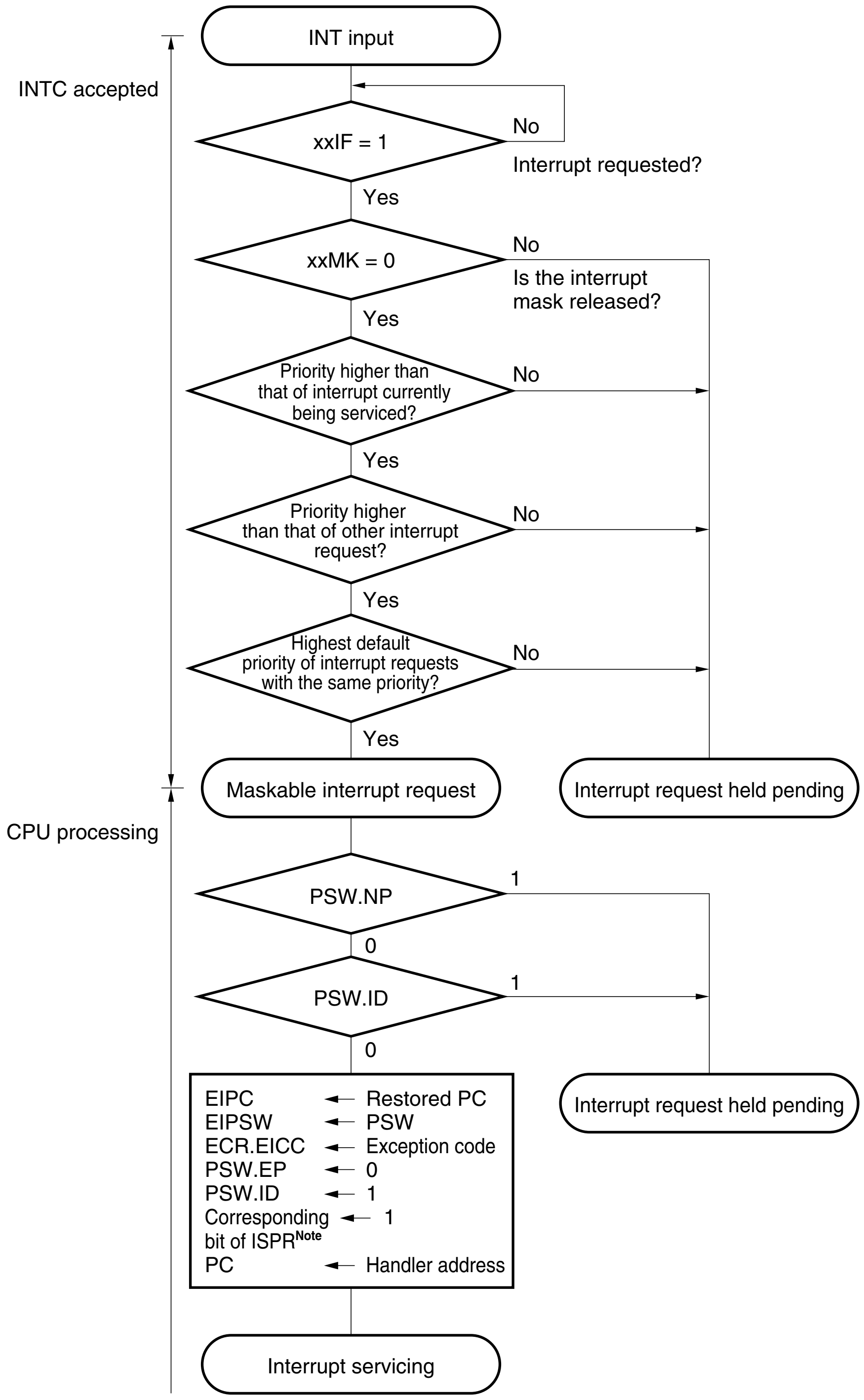
<4> Sets the ID bit of the PSW and clears the EP bit.

<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request masked by INTC and the maskable interrupt request generated while another interrupt is being serviced (while PSW.NP = 1 or PSW.ID = 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 if either the maskable interrupt is unmasked or PSW.NP and PSW.ID are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 17-6: Maskable Interrupt Servicing


INT input
INTC accepted
xxIF = 1
No
Interrupt requested?
Yes
xxMK = 0
No
Is the interrupt mask released?
Yes
Priority higher than that of interrupt currently being serviced?
No
Yes
Priority higher than that of other interrupt request?
No
Yes
Highest default priority of interrupt requests with the same priority?
No
Yes
Maskable interrupt request
Interrupt request held pending
CPU processing
PSW.NP
1
0
PSW.ID
1
0
EIPC ← Restored PC
EIPSW ← PSW
ECR.EICC ← Exception code
PSW.EP ← 0
PSW.ID ← 1
Corresponding bit of ISPRNote ← 1
PC ← Handler address
Interrupt request held pending
Interrupt servicing

Note: For the ISPR register, see **13.3.6 In-service priority register (ISPR)**.

The INT input masked by the interrupt controllers and the INT input that occurs while another interrupt is being serviced (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt servicing.

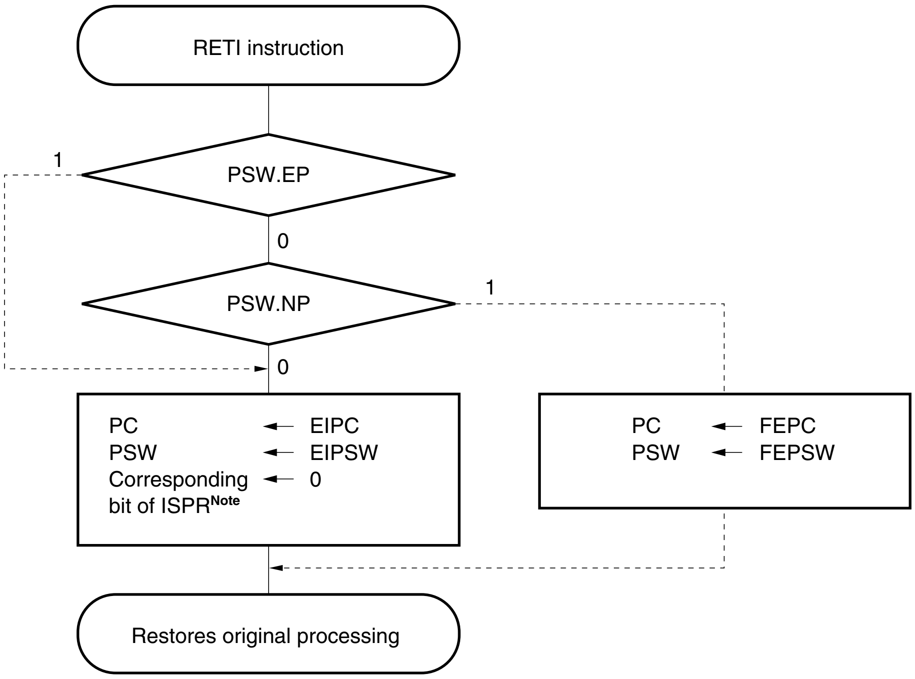
17.3.2 Restore

Recovery from maskable interrupt servicing is carried out by the RETI instruction.
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.
<2> Transfers control to the address of the restored PC and PSW.

Figure 17-7 illustrates the processing of the RETI instruction.

Figure 17-7: RETI Instruction Processing



Note: For the ISPR register, see **13.3.6 In-service priority register (ISPR)**.

Caution: When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid line shows the CPU processing flow.

17.3.3 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, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 17-1, "Interrupt/Exception Source List," on page 685**.

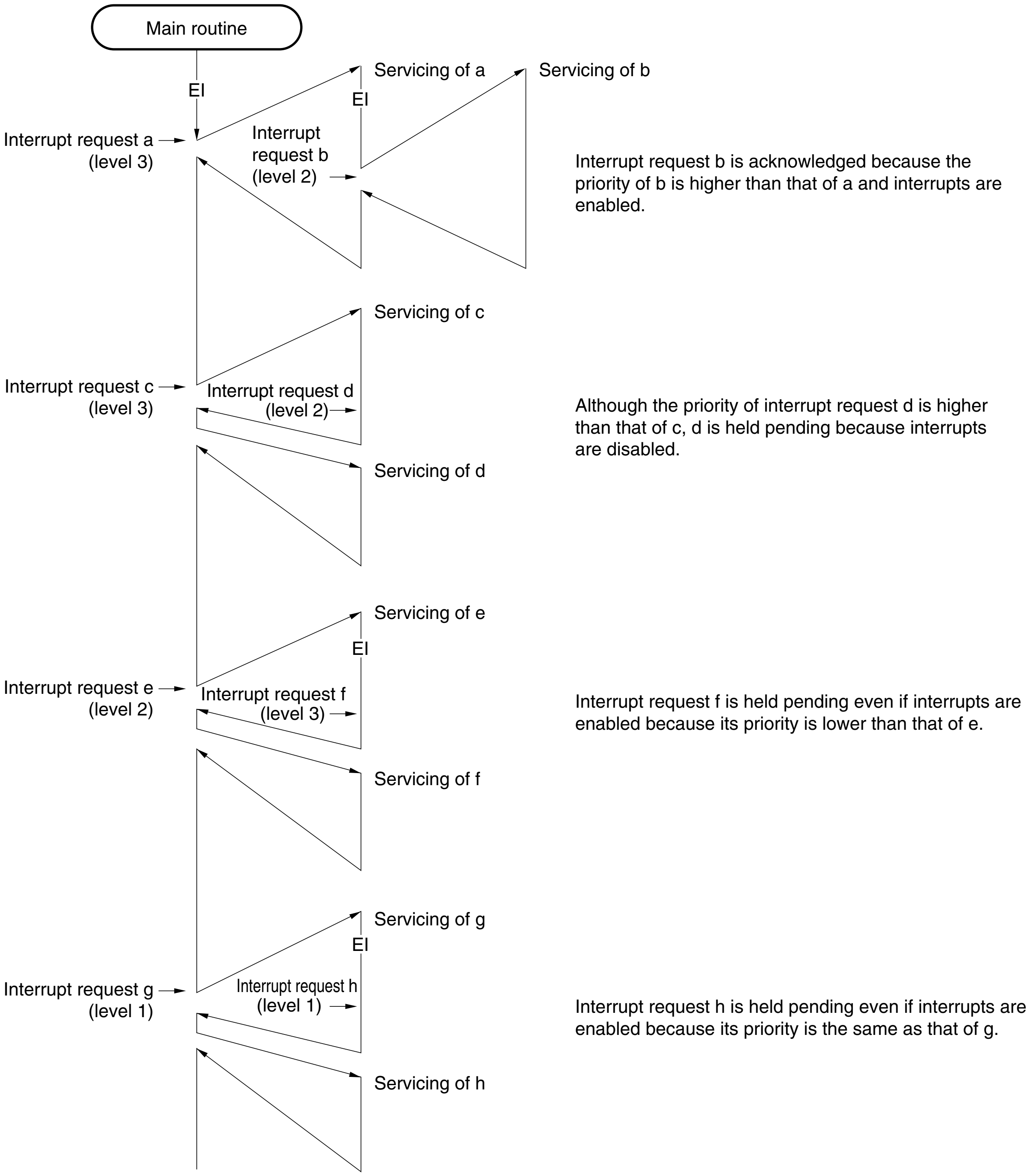
The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW 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.

Remarks: **1.** xx: Identification name of each peripheral unit (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 702**)

2. n: Peripheral unit number (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 700**).

Figure 17-8: Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2)

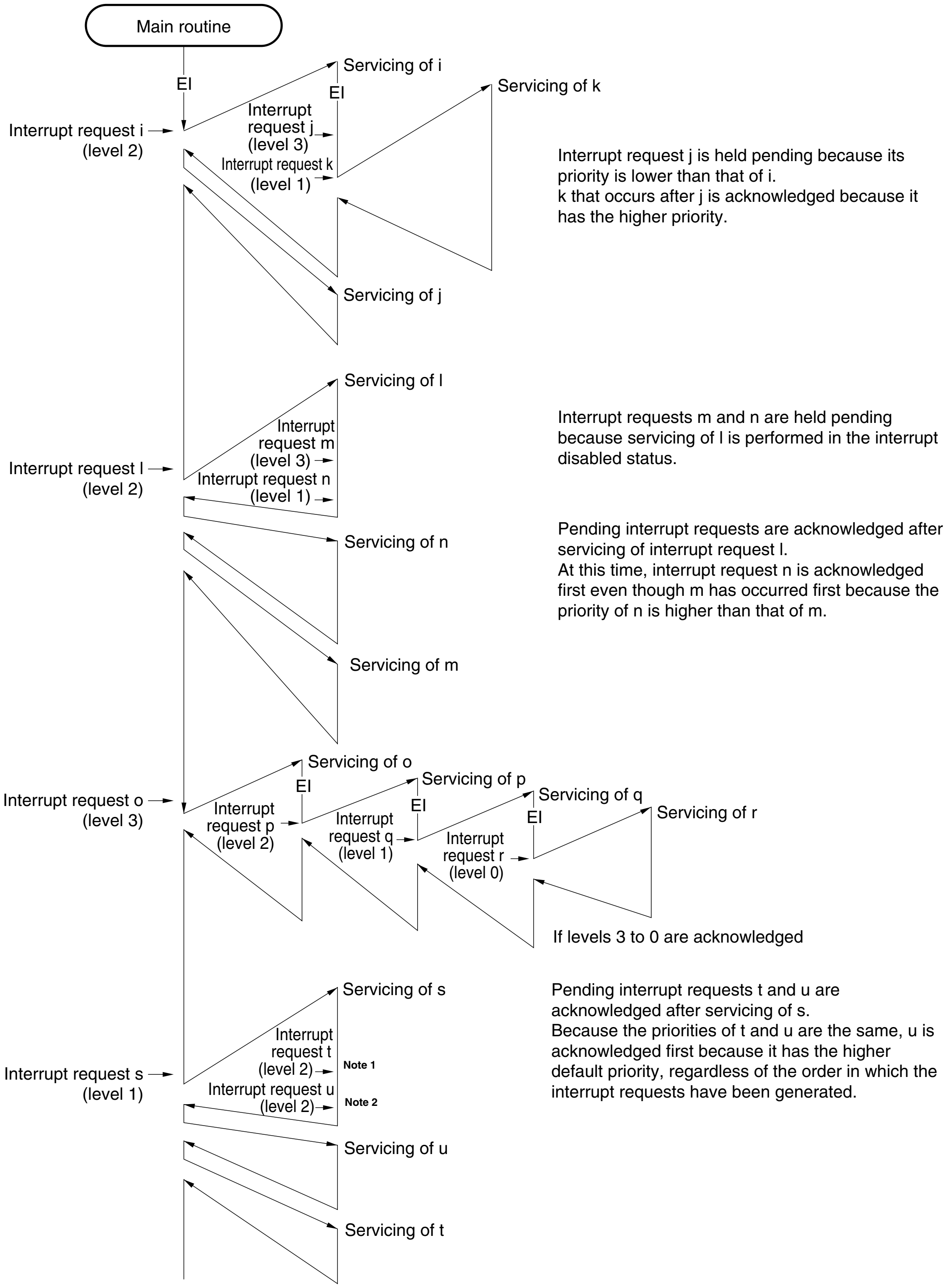

Main routine
EI
Interrupt request a (level 3)
Servicing of a
EI
Interrupt request b (level 2)
Servicing of b
Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.
Servicing of c
Interrupt request c (level 3)
Interrupt request d (level 2)
Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.
Servicing of d
Servicing of e
EI
Interrupt request e (level 2)
Interrupt request f (level 3)
Interrupt request f is held pending even if interrupts are enabled because its priority is lower than that of e.
Servicing of f
Servicing of g
EI
Interrupt request g (level 1)
Interrupt request h (level 1)
Interrupt request h is held pending even if interrupts are enabled because its priority is the same as that of g.
Servicing of h

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 requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 17-8: Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (2/2)


Main routine
EI
Interrupt request i (level 2)
Servicing of i
EI
Interrupt request j (level 3)
Interrupt request k (level 1)
Servicing of k
Servicing of j
Servicing of l
Interrupt request l (level 2)
Interrupt request m (level 3)
Interrupt request n (level 1)
Servicing of n
Servicing of m
Interrupt request o (level 3)
Servicing of o
EI
Interrupt request p (level 2)
Servicing of p
EI
Interrupt request q (level 1)
Servicing of q
EI
Interrupt request r (level 0)
Servicing of r
Interrupt request s (level 1)
Servicing of s
Interrupt request t (level 2)
Note 1
Interrupt request u (level 2)
Note 2
Servicing of u
Servicing of t

Interrupt request j is held pending because its priority is lower than that of i.
k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.
At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

If levels 3 to 0 are acknowledged

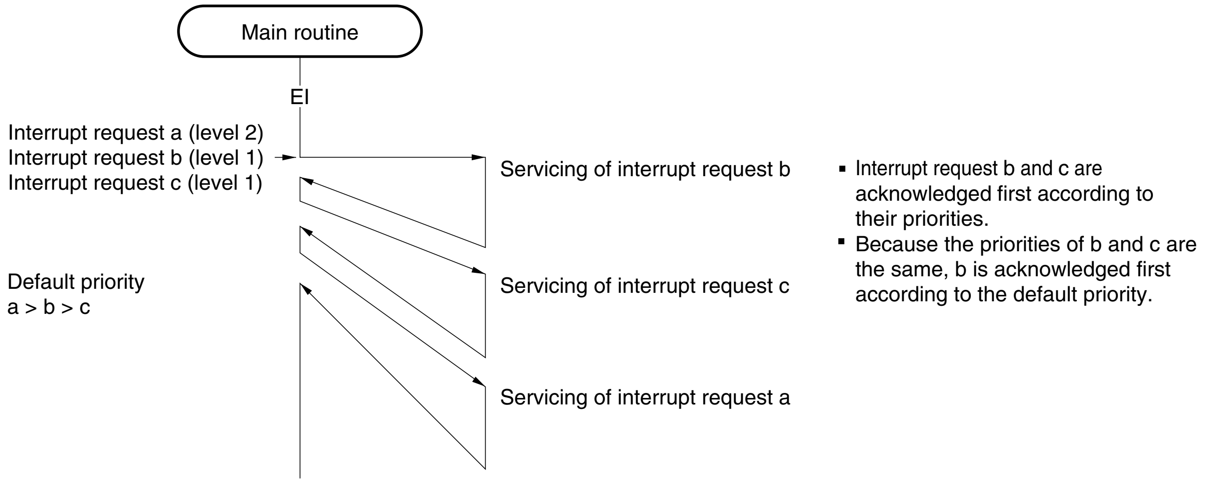
Pending interrupt requests t and u are acknowledged after servicing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.

Notes: 1. Lower default priority

2. Higher default priority

Caution: To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 17-9: Example of Servicing Interrupt Requests Simultaneously Generated


Main routine
EI
Interrupt request a (level 2)
Interrupt request b (level 1)
Interrupt request c (level 1)
Default priority
a > b > c
Servicing of interrupt request b
Servicing of interrupt request c
Servicing of interrupt request a
▪ Interrupt request b and c are acknowledged first according to their priorities.
▪ Because the priorities of b and c are the same, b is acknowledged first according to the default priority.

Caution: To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks: 1. **a** to **c** in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

17.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each interrupt request (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.

Figure 17-10: Interrupt Control Register (xxICn) Format

After reset: 47H R/W Address: FFFFF110H to FFFFF18EH

	<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 is acknowledged.

Remark: xx: Identification name of each peripheral unit (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 702**)
n: Peripheral unit number (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 700**)

Caution: Disable interrupts (DI) or mask the interrupt to read the xxICn bit.
If the xxIFn bit is read while interrupts are enable (EI) or while the interrupt is unmasked, the correct value may not be read when acknowledging an interrupt are reading the bit conflict.

The addresses and bits of the interrupt control registers are as follows.

Table 17-2: Interrupt Control Register (xxICn) (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVIIC	LVIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TM0EQIC0	TM0EQIF	TM0EQMK	0	0	0	TM0EQPR2	TM0EQPR1	TM0EQPR0
FFFFF146H	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF148H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF14AH	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF14CH	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF14EH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF150H	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF152H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF154H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF156H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF158H	C0ERRIC	C0ERRIF	C0ERRMK	0	0	0	C0ERRPR2	C0ERRPR1	C0ERRPR0
FFFFF15AH	C0WUPIC	C0WUPIF	C0WUPMK	0	0	0	C0WUPPR2	C0WUPPR1	C0WUPPR0
FFFFF15CH	C0RECIC	C0RECIF	C0RECMK	0	0	0	C0RECPR2	C0RECPR1	C0RECPR0

Table 17-2: Interrupt Control Register (xxICn) (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF15EH	C0TRXIC	C0TRXIF	C0TRXMK	0	0	0	C0TRXPR2	C0TRXPR1	C0TRXPR0
FFFFF160H	C30IC	C30IF	C30MK	0	0	0	C30PR2	C30PR1	C30PR0
FFFFF162H	C30OC	C30OIF	C30OMK	0	0	0	C30OPR2	C30OPR1	C30OPR0
FFFFF164H	C31IC	C31IF	C31MK	0	0	0	C31PR2	C31PR1	C31PR0
FFFFF166H	C31OC	C31OIF	C31OMK	0	0	0	C31OPR2	C31OPR1	C31OPR0
FFFFF16CH	TQ1OVIC	TQ1OVIF	TQ1OVMK	0	0	0	TQ1OVPR2	TQ1OVPR1	TQ1OVPR0
FFFFF16EH	TQ1CCIC0	TQ1CCIF0	TQ1CCMK0	0	0	0	TQ1CCPR02	TQ1CCPR01	TQ1CCPR00
FFFFF170H	TQ1CCIC1	TQ1CCIF1	TQ1CCMK1	0	0	0	TQ1CCPR12	TQ1CCPR11	TQ1CCPR10
FFFFF172H	TQ1CCIC2	TQ1CCIF2	TQ1CCMK2	0	0	0	TQ1CCPR22	TQ1CCPR21	TQ1CCPR20
FFFFF174H	TQ1CCIC3	TQ1CCIF3	TQ1CCMK3	0	0	0	TQ1CCPR32	TQ1CCPR31	TQ1CCPR30
FFFFF17AH	C1ERRIC	C1ERRIF	C1ERRMK	0	0	0	C1ERRPR2	C1ERRPR1	C1ERRPR0
FFFFF17CH	C1WUPIC	C1WUPIF	C1WUPMK	0	0	0	C1WUPPR2	C1WUPPR1	C1WUPPR0
FFFFF17EH	C1RECIC	C1RECIF	C1RECMK	0	0	0	C1RECPR2	C1RECPR1	C1RECPR0
FFFFF180H	C1TRXIC	C1TRXIF	C1TRXMK	0	0	0	C1TRXPR2	C1TRXPR1	C1TRXPR0
FFFFF182H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF184H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF186H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF188H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF18AH	DMAIC4	DMAIF4	DMAMK4	0	0	0	DMAPR42	DMAPR41	DMAPR40
FFFFF18CH	DMAIC5	DMAIF5	DMAMK5	0	0	0	DMAPR52	DMAPR51	DMAPR50

17.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)

These registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR3 registers is equivalent to the xxMKn bit of the xxICn register.
The IMRm register can be read or written in 16-bit units (m = 0 to 3).
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 3).

Caution: The device file defines the xxMKn bit of the xxICn register 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).

Figure 17-11: Interrupt Mask Registers 0 to 3 (IMR0 to IMR3) Format

After reset: FFFFH R/W Address: FFFF106H

IMR3	15	14	13	12	11	10	9	8
	1[Note]	DMAMK5	DMAMK4	DMAMK3	DMAMK2	DMAMK1	DMAMK0	C1TRXMK
	7	6	5	4	3	2	1	0
	C1RECMK	C1WUPMK	C1ERRMK	1[Note]	1[Note]	TQ1CCMK3	TQ1CCMK2	TQ1CCMK1

After reset: FFFFH R/W Address: FFFF104H

IMR2	15	14	13	12	11	10	9	8
	TQ1CCMK0	TQ1OVMK	1[Note]	1[Note]	C31OMK	C31MK	C30OMK	C30MK
	7	6	5	4	3	2	1	0
	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	ADMK	UA1TMK	UA1RMK	UA0TMK

After reset: FFFFH R/W Address: FFFF102H

IMR1	15	14	13	12	11	10	9	8
	UA0RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK	TM0EQMK	TP3CCMK1	TP3CCMK0
	7	6	5	4	3	2	1	0
	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1

After reset: FFFFH R/W Address: FFFF100H

IMR0	15	14	13	12	11	10	9	8
	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
	7	6	5	4	3	2	1	0
	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Remark: xx:Identification name of each peripheral unit (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 702**)
n:Peripheral unit number (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 700**)

Note: Be sure to set bit to 1. If these bits are cleared to 0, the operation cannot be guaranteed.

17.3.6 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set to 1 and remains set while the interrupt is serviced.
When the RETI instruction is executed, the bit corresponding to the interrupt request 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.

Figure 17-12: In-Service Priority Register (ISPR) Format

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)

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 disable (DI).**

17.3.7 Maskable interrupt status flag

This flag controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests. An interrupt disable flag (ID) is incorporated, which is assigned to the PSW.

Figure 17-13: Maskable Interrupt Status Flag Format

After reset: 00000021H

	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 reset 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 requests and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.
The interrupt request generated during the acknowledgement disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is reset to 0.

17.3.8 Watchdog timer mode register 2 (WDTM2)

This register is a special register and can be written only in a specific sequence. To generate a maskable interrupt (INTWDT2), clear the WDM20 bit of this register to 0.
This register can be read or written in 8-bit or 1-bit units (for details, refer to **Chapter 10 ”Functions of Watchdog Timer 2” on page 359**).

Figure 17-14: Watchdog Timer Mode Register 2 (WDTM2) Format

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)
1	-	Reset mode (generation of WDTRES2)

Cautions:

1. **For details of bits WDCS20 to WDCS24, refer to Table 17-3, “Watchdog Timer 2 Clock Selection,” on page 708.**
2. **Although Watchdog timer 2 can be stopped just by stopping the operation of Ring-OSC, set the WDTM2 register to 1FH to securely stop the timer (to avoid selection of the main clock due to an erroneous write operation).**
3. **If the WDTM2 register is rewritten twice after reset, an overflow signal is forcibly generated. But, the overflow signal does not occur, even if the WDTM2 register is written twice after the watch dog timer is suspended.**
4. **To stop the operation of Watchdog timer 2, set the RSTP bit of the RCM register to 1 (to stop Ring-OSC) and the WDTM2 register to 1FH.**

Table 17-3: Watchdog Timer 2 Clock Selection

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected clock	100 kHz (MIN.)	200 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2621.4 ms	1310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5242.9 ms	2621.47 ms	1310.7 ms
						f_{XX} = 24 MHz	f_{XX} = 32 MHz	f_{XX} = 40 MHz
0	1	0	0	0	$2^{18}/f_{XX}$	10.9 ms	8.19 ms	6.6 ms
0	1	0	0	1	$2^{19}/f_{XX}$	21.8 ms	16.4 ms	13.1 ms
0	1	0	1	0	$2^{20}/f_{XX}$	43.7 ms	32.8 ms	26.2 ms
0	1	0	1	1	$2^{21}/f_{XX}$	87.4 ms	65.5 ms	52.2 ms
0	1	1	0	0	$2^{22}/f_{XX}$	174.8 ms	131.1 ms	104.9 ms
0	1	1	0	1	$2^{23}/f_{XX}$	349.5 ms	262.1 ms	209.7 ms
0	1	1	1	0	$2^{24}/f_{XX}$	699.1 ms	524.3 ms	419.4 ms
0	1	1	1	1	$2^{25}/f_{XX}$	1398.1 ms	1048.6 ms	838.9 ms

17.3.9 Eliminating noise on INTP0 to INTP7 pins

The INTP0 to INTP7 pins have a noise eliminator that eliminates noise using analog delay. Unless the level input to each pin is held for a specific time, therefore, it cannot be detected as a signal edge i.e., the edge is detected after specific time.

17.3.10 Function to detect edge of INTP0 to INTP7 pins

The valid edge of the INTP0 to INTP7 pins can be selected from the following four.

- Rising edge
- Falling edge
- Both edges
- No edge detection

(1) External interrupt falling edge specification register 0 (INTF0)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF0n and INTR0n to 0, and then set the port mode.

Figure 17-15: External Interrupt Falling Edge Specification Register 0 (INTF0) Format

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	0	0	0	0	0

Remark: For how to specify a valid edge, refer to Table 17-4.

(2) External interrupt rising edge specification register 0 (INTR0)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF0n and INTR0n to 0, and then set the port mode.

Figure 17-16: External Interrupt Rising Edge Specification Register 0 (INTR0) Format

After reset: 00H R/W Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	0	0	0	0	0

Remark: For how to specify a valid edge, refer to Table 17-4.

Table 17-4: Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 5, 6)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Caution: Be sure to set INTF0n and INTR0n to 00 when these terminals are not used as external interrupt function.

Remark: n = 5: Control of INTP2
n = 6: Control of INTP3

(3) External interrupt falling edge specification register 1 (INTF1)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF1n and INTR1n to 0, and then set the port mode.

Figure 17-17: External Interrupt Falling Edge Specification Register 1 (INTF1) Format

After reset: 00H R/W Address: FFFFFC02H

	7	6	5	4	3	2	1	0
INTF1	0	0	0	0	0	0	0	INTF10

Remark: For how to specify a valid edge, refer to Table 17-5.

(4) External interrupt rising edge specification register 1 (INTR1)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF1n and INTR1n to 0, and then set the port mode.

Figure 17-18: External Interrupt Rising Edge Specification Register 1 (INTR1) Format

After reset: 00H R/W Address: FFFFFC22H

	7	6	5	4	3	2	1	0
INTR1	0	0	0	0	0	0	0	INTR10

Remark: For how to specify a valid edge, refer to Table 17-5.

Table 17-5: Valid Edge Specification

INTF1n	INTR1n	Valid Edge Specification (n = 0)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Caution: Be sure to set INTF1 and INTR1 to 00 when these terminals are not used as external interrupt function.

Remark: n = 0: Control of INTP0

(5) External interrupt falling edge specification register 3 (INTF3)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF3n and INTR3n to 0, and then set the port mode.

Figure 17-19: External Interrupt Falling Edge Specification Register 3 (INTF3) Format

After reset: 00H R/W Address: FFFFFC06H

	7	6	5	4	3	2	1	0
INTF3	INTF37	0	0	0	0	0	INTF31	0

Remark: For how to specify a valid edge, refer to Table 17-6.

(6) External interrupt rising edge specification register 3 (INTR3)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF3n and INTR3n to 0, and then set the port mode.

Figure 17-20: External Interrupt Rising Edge Specification Register 3 (INTR3) Format

After reset: 00H R/W Address: FFFFFC26H

	7	6	5	4	3	2	1	0
INTR3	INTR37	0	0	0	0	0	INTR31	0

Remark: For how to specify a valid edge, refer to Table 17-6.

Table 17-6: Valid Edge Specification

INTF3n	INTR3n	Valid Edge Specification (n = 1, 7)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Caution: Be sure to set INTF3 and INTR3 to 00 when these terminals are not used as external interrupt function.

Remark: n = 1: Control of INTP7
n = 7: Control of INTP1

(7) External interrupt falling edge specification register 9H (INTF9H)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Figure 17-21: External Interrupt Falling Edge Specification Register 9H (INTF9H) Format

After reset: 00H R/W Address: FFFFFC13H

	7	6	5	4	3	2	1	0
INTF9H	INTF915	INTF914	INTF93	0	0	0	0	0

Remark: For how to specify a valid edge, refer to Table 17-7.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF6n and INTR6n to 0, and then set the port mode.

(8) External interrupt rising edge specification register 9H (INTR9H)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pins. This register can be read or written in 8-bit or 1-bit units.

Figure 17-22: External Interrupt Rising Edge Specification Register 9H (INTR9H) Format

After reset: 00H R/W Address: FFFFFC33H

	7	6	5	4	3	2	1	0
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0

Remark: For how to specify a valid edge, refer to Table 17-7.

Caution: When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF6n and INTR6n to 0, and then set the port mode.

Table 17-7: Valid Edge Specification

INTF9Hn	INTR9Hn	Valid Edge Specification (n = 13 to 15)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Caution: Be sure to set INTF9H and INTR9H to 00 when these terminals are not used as external interrupt function.

Remark: n = 13: Control of INTP4
n = 14: Control of INTP5
n = 15: Control of INTP6

(9) Noise elimination control register

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are performed with the NFC register.

When digital noise elimination is selected, the sampling clock for digital sampling can be selected from among f_{XX}/32, f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1024. Sampling is performed 3 times.

Even when digital noise elimination is selected, using f_{XT} as the sampling clock makes it possible to use the INTP3 interrupt request to release the IDLE and STOP modes.

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

Caution: After the sampling clock has been changed, it takes 3 sampling clocks to initialize the digital noise eliminator. Therefore, if an INTP3 valid edge is input within these 3 sampling clocks after the sampling clock has been changed, an interrupt request may occur. Therefore, be careful about the following points when using the DMA function.

- **When using the interrupt function, after the 3 sampling clocks have elapsed, allow the interrupt after the interrupt request flag (bit 7 of PIC3) has been cleared.**
- **When using the DMA function (started with INTP3), enable DMA after 3 sampling clocks have elapsed.**

Figure 17-23: Noise Elimination Control Register Format

After reset: 00H R/W Address: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

NFEN	Settings of INTP3 pin noise elimination
0	Analog noise elimination (60 ns (TYP.))
1	Digital noise elimination

NFC2	NFC1	NFC0	Digital sampling clock
0	0	0	f_{XX}/64
0	0	1	f_{XX} /128
0	1	0	f_{XX} /256
0	1	1	f_{XX} /512
1	0	0	f_{XX} /1,024
1	0	1	f_{XX}/32
Other than above			Setting prohibited

Remarks: 1. Since sampling is performed 3 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 is generated if noise synchronized with the sampling clock is input.

17.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

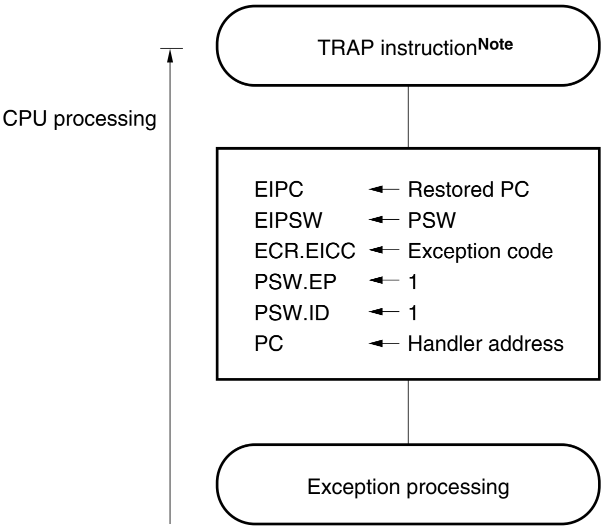
17.4.1 Operation

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 EP and ID bits of the PSW.
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 17-24 illustrates the processing of a software exception.

Figure 17-24: Software Exception Processing



Note: TRAP instruction format: TRAP vector (the vector is a value from 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

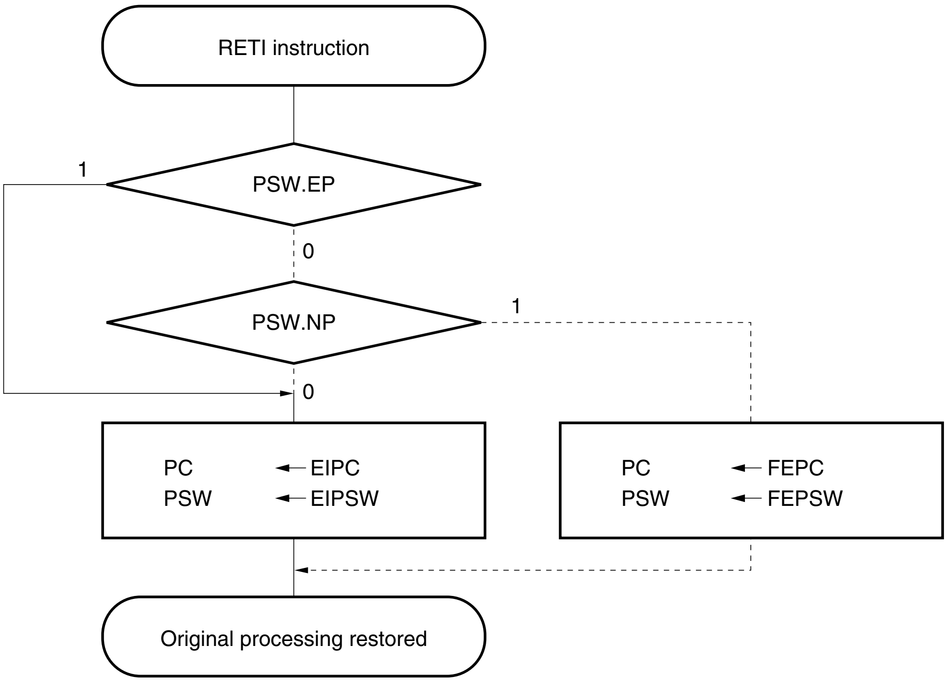
17.4.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 1.
<2> Transfers control to the address of the restored PC and PSW.

Figure 17-25 illustrates the processing of the RETI instruction.

Figure 17-25: RETI Instruction Processing



Caution: **When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.**

Remark: The solid line shows the CPU processing flow.

17.4.3 Exception status flag (EP)

The EP flag is bit 6 of the PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

Figure 17-26: Exception Status Flag (EP) Format

After reset: 00000021H

	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.

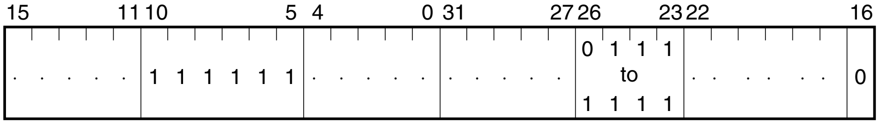
17.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. Therefore, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

17.5.1 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.

Figure 17-27: Illegal Opcode Definition



15 11	10 5	4 0	31 27	26 23	22	16
· · · · ·	1 1 1 1 1 1	· · · · ·	· · · · ·	0 1 1 1 to 1 1 1 1	· · · · · ·	0

Remark: ×: 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.

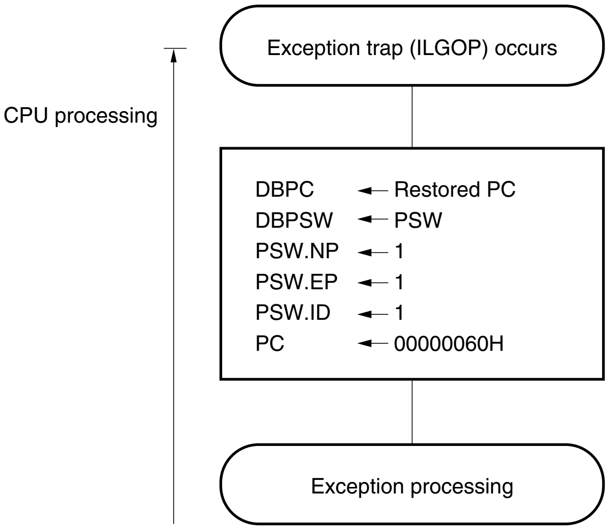
<2> Saves the current PSW to DBPSW.

<3> Sets the NP, EP, and ID bits of the PSW.

<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 17-28 illustrates the processing of the exception trap.

Figure 17-28: Exception Trap Processing



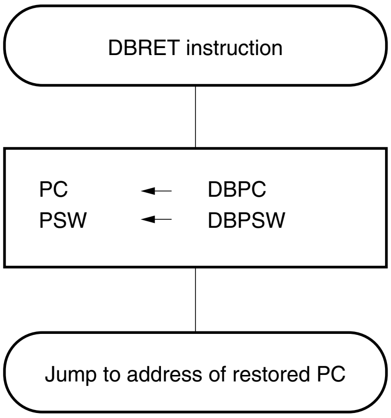
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 17-29 illustrates the restore processing from an exception trap.

Figure 17-29: Restore Processing from Exception Trap



17.5.2 Debug trap

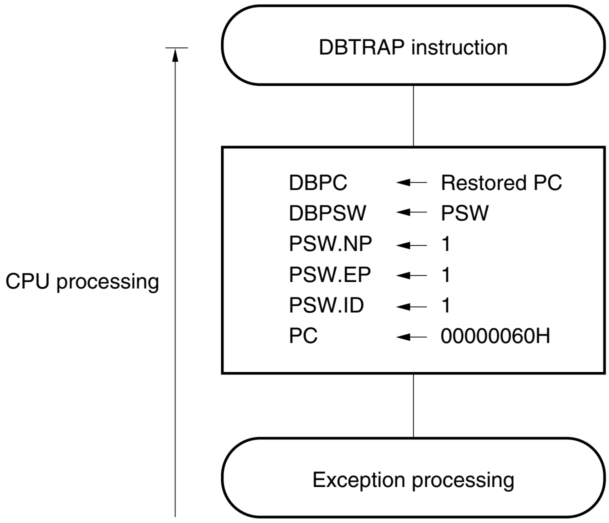
A debug trap is an exception that is generated when the DBTRAP instruction is executed and is always acknowledged.
Upon occurrence of a debug trap, the CPU performs the following processing.

(1) Operation

<1> Saves restored PC to DBPC.
<2> Saves current PSW to DBPSW.
<3> Sets the NP, EP, and ID bits of PSW.
<4> Sets handler address (00000060H) for debug trap to PC and transfers control.

Figure 17-30 shows the debug trap processing format.

Figure 17-30: Debug Trap Processing Format



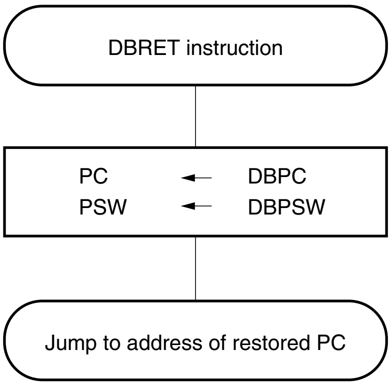
(2) Restoration

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.

Figure 17-31 shows the processing format for restoration from a debug trap.

Figure 17-31: Processing Format of Restoration from Debug Trap

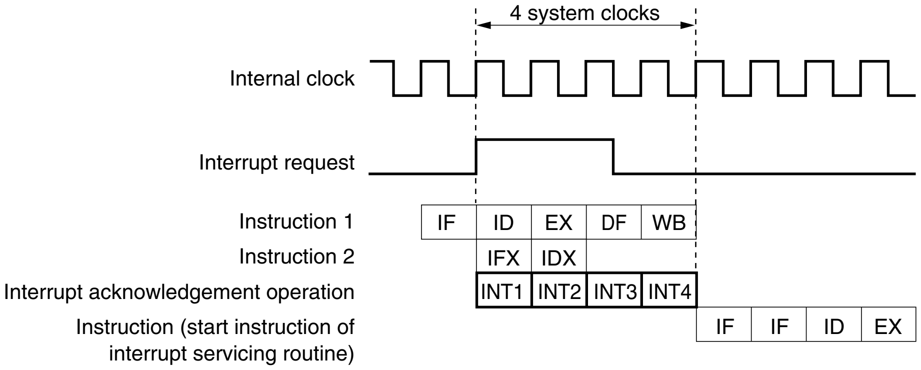


17.6 Interrupt Acknowledge Time of CPU

Except the following cases, the interrupt acknowledge time of the CPU is 4 clocks minimum. To input interrupt requests successively, input the next interrupt at least 4 clocks after the preceding interrupt.

- In software/hardware STOP mode
- When the external bus is accessed
- When interrupt request non-sampling instructions are successively executed (Refer to **17.7 ”Periods in Which Interrupts Are Not Acknowledged by CPU” on page 722**).
- When the interrupt control register is accessed

Figure 17-32: Pipeline Operation at Interrupt Request Acknowledgement (Outline)



Remark: INT1 to INT4: Interrupt acknowledgement 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 IDLE/software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to peripheral I/O register • Access to interrupt control register • Access to programmable peripheral I/O register
Maximum	7	7 + Analog delay time	

17.7 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
- Store instruction and SET1, NOT1, CLR1 instruction for special area
  (xFFF100H-xFFF1FFH, xFFF900H-xFFF9FFH)

Remark: xx: Identification name of each peripheral unit (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 702**)
n: Peripheral unit number (Refer to **Table 17-2, "Interrupt Control Register (xxICn)," on page 700**).

Chapter 18 Standby Function

18.1 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 18-1.

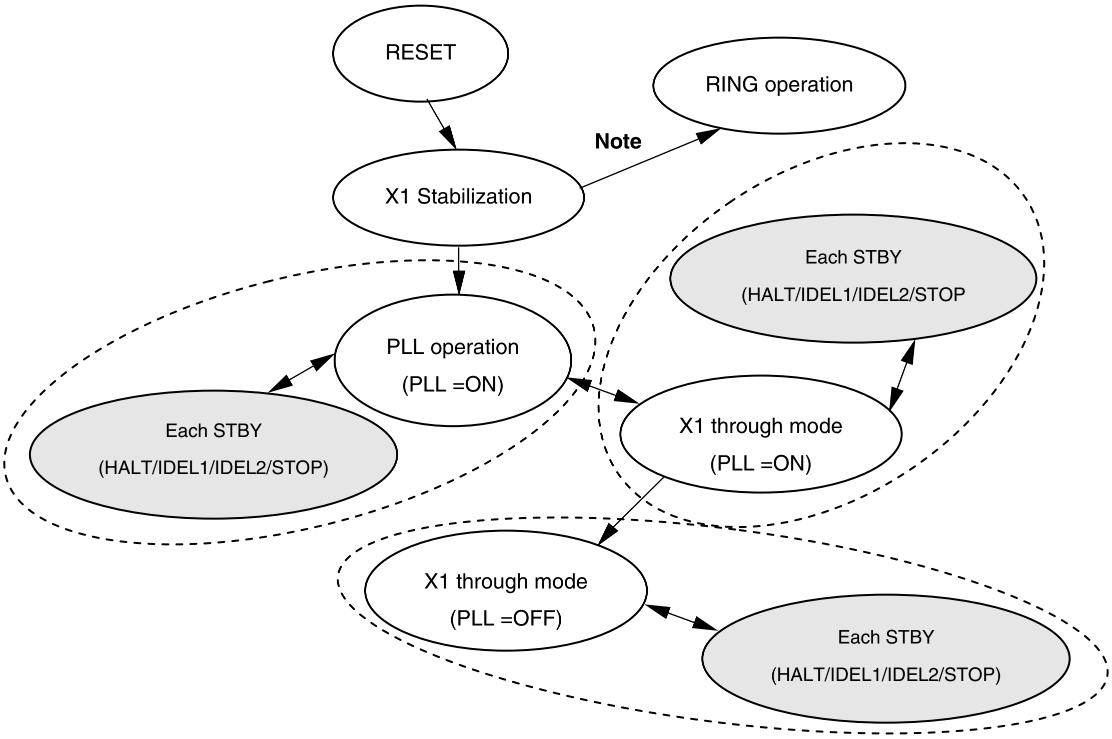
Table 18-1: Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU.
IDLE1 mode	Mode to stop all the internal operations of the chip except the oscillator, PLL[Note], Flash memory.
IDLE2 mode	Mode to stop all the internal operations of the chip except the oscillator, Flash memory.
STOP mode	Mode to stop all the internal operations of the chip.

Note: The PLL holds the previous operating status.

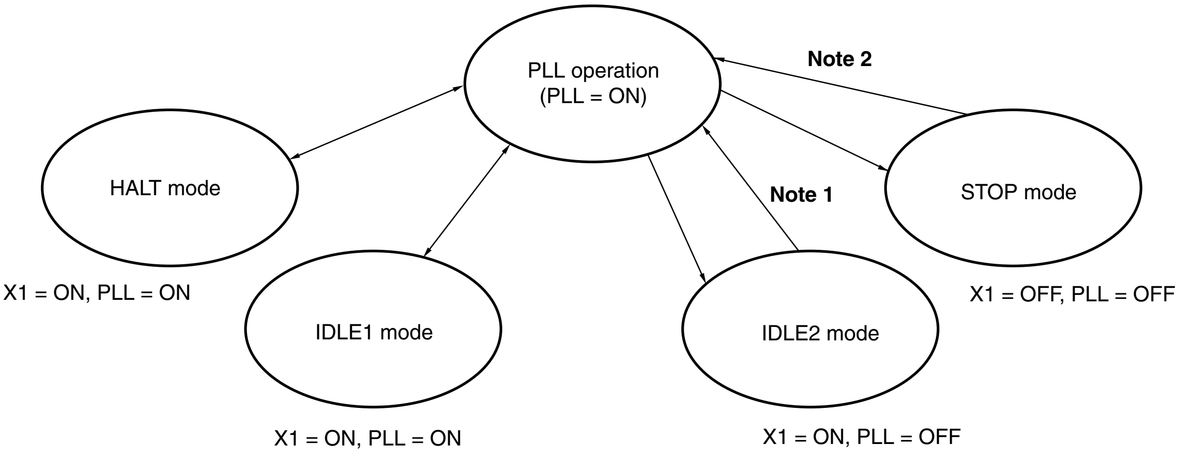
18.2 Status Transition

Figure 18-1: Status Transition



Note: RING operation is executed when WDT2RES is generated during the oscillation stabilization time.

Figure 18-2: Standby Transition from PLL Operation (PLL = ON)

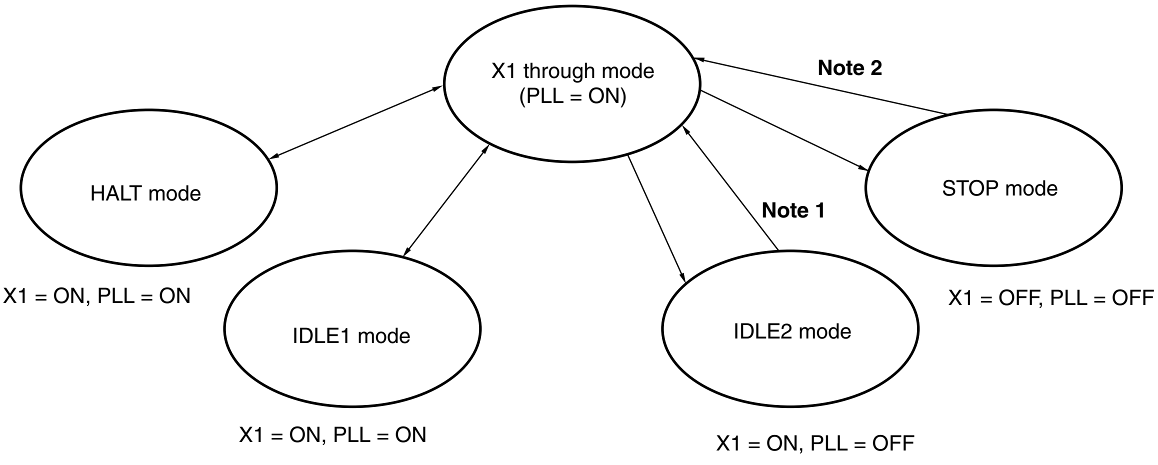


Notes: 1. After OSTS time is expired CPU returns to PLL operation mode. (Flash setup time, PLL lock-up time)

2. After OSTS time is expired CPU returns to PLL operation mode. If WDT2RES occurs when counting the oscillation stabilization time, CPU clock changed to ring oscillator.

Remark: For details explanation of OSTS, refer to 6.3 (7)”Oscillation stabilization time select register (OSTS)” on page 241 and 10.3 (1)”Oscillation stabilization time select register (OSTS)” on page 360.

Figure 18-3: Standby Transition from X1 Through Mode (PLL = ON)

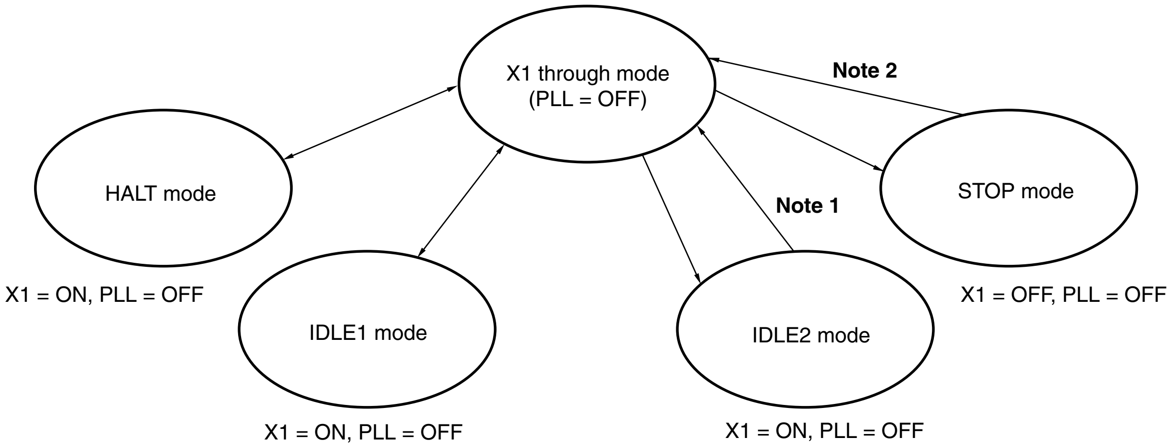


Notes: 1. After OSTS time is expired CPU returns to X1 through mode operation. (Flash setup time)

2. After OSTS time is expired CPU returns to X1 through mode operation. If WDT2RES occurs when counting the oscillation stabilization time, CPU clock changed to ring oscillator.

Remark: For details explanation of OSTS, refer to 6.3 (7)”Oscillation stabilization time select register (OSTS)” on page 241 and 10.3 (1)”Oscillation stabilization time select register (OSTS)” on page 360.

Figure 18-4: Standby Transition from X1 Through Mode (PLL = OFF)



Notes: 1. After OSTS time is expired CPU returns to X1 through mode operation.

2. After OSTS time is expired CPU returns to through mode operation. If WDT2RES occurs when counting the oscillation stabilization time, CPU clock changed to ring oscillator.

Remark: For details explanation of OSTS, refer to 6.3 (7)”Oscillation stabilization time select register (OSTS)” on page 241 and 10.3 (1)”Oscillation stabilization time select register (OSTS)” on page 360.

18.3 HALT Mode

18.3.1 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 18-2 shows the operation status in the HALT mode.

Cautions: 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an unmasked interrupt request signal is being held pending, the status shifts to HALT mode, but the HALT mode is then released immediately by the pending interrupt request.

18.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request (NMI pin input, INTWDT2 occurrence), unmasked external interrupt request (INTP0 to INTP7 pin input), unmasked internal interrupt request from the peripheral functions operable in the software HALT mode, or reset signals (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)).

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request or unmasked maskable interrupt request

The HALT mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the HALT mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the HALT mode is released and that interrupt request is acknowledged.

Table 18-2: Operation After Releasing HALT Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-3: Operation Status in HALT Mode

Item \ Setting of HALT Mode		Operation Status
Main clock oscillator (f_X)		Oscillation enabled
Ring clock generator (f_R)		Oscillation enabled
PLL		Operable
Flash charge pump		Continues operation
CPU		Stops operation
DMA		Operable
Interrupt controller		Operable
Timer P (TMP0 to TMP3)		Operable
Timer Q (TMQ0, TMQ1)		Operable
Timer M (TMM0)		Operable
Watchdog timer 2 (WDT2)		Operable
Serial interface	CSIB0, CSIB1	Operable
	UARTA0, UARTA1	Operable
	CSI30, CSI31	Operable
AFCAN0, AFCAN1		Operable
A/D converter		Operable
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before HALT mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.

18.4 IDLE1 Mode

18.4.1 Setting and operation status

The IDLE1 mode is set by clearing the PSM1, 0 bit of the power save mode register (PSMR) to 00 and setting the STP bit of the power save control register (PSC) to 1 in the normal operation mode.
In the IDLE1 mode, the clock oscillator, PLL and Flash continues operation but clock supply to the CPU and other on-chip peripheral functions stops.
As a result, program execution stops and the contents of the internal RAM before the IDLE1 mode was set are retained. Because the IDLE1 stops operation of the on-chip peripheral functions, it reduces the current consumption to a level lower than the HALT mode.
Table 18-4 shows the operation status in the IDLE1 mode.
The IDLE1 mode can reduce the current consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The main clock oscillator, PLL and Flash 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.

Caution: Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE1 mode.

18.4.2 Releasing IDLE1 mode

The IDLE1 mode is released by a non-maskable interrupt request (NMI pin input, INTWDT2 occurrence), unmasked external interrupt request (INTP0 to INTP7 pin input), unmasked internal interrupt request from the peripheral functions operable in the software IDLE1 mode, or reset signals.
After the IDLE1 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE1 mode by non-maskable interrupt request or unmasked maskable interrupt request

The IDLE1 mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the IDLE1 mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Caution: An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI1M, and PSC.INTM bits to 1 becomes invalid and IDLE1 mode is not released.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the IDLE1 mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the IDLE1 mode is released and that interrupt request is acknowledged.

Table 18-4: Operation After Releasing IDLE1 Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing IDLE1 mode by RESET pin input

The same operation as the normal reset operation is performed.

Table 18-5: Operation Status in IDLE1 Mode

Setting of IDLE1 Mode Item		Operation Status
Main clock oscillator (f_X)		Oscillation enabled
Ring clock generator (f_R)		Oscillation enabled
PLL		Operable
Flash charge pump		Continues operation
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation (Standby mode release enabled)
Timer P (TMP0 to TMP3)		Stops operation
Timer Q (TMQ0, TMQ1)		Stops operation
Timer M (TMM0)		Operable when f_R/8 is selected as the count clock
Watchdog timer 2 (WDT2)		Operable when f_R is selected as the count clock
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$ input clock is selected as operation clock
	UARTA0, UARTA1	Stops operation (Operable when ASCKA0 input clock is selected as operation clock)
	CSI30, CSI31	Stops operation
AFCAN0, AFCAN1		Stops operation
A/D converter		Stops operation
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before IDLE1 mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE1 mode was set.

18.5 IDLE2 Mode

18.5.1 Setting and operation status

The IDLE2 mode is set by clearing the PSM1, 0 bit of the power save mode register (PSMR) to 10 and setting the STP bit of the power save control register (PSC) to 1 in the normal operation mode.
In the IDLE2 mode, the clock oscillator continues operation but clock supply to the CPU, PLL 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. Because the IDLE2 stops operation of the on-chip peripheral functions, it reduces the current consumption to a level lower than the IDLE1 mode.
Table 18-6 shows the operation status in the IDLE2 mode.
The IDLE2 mode can reduce the current consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. PLL stop, so the normal operation mode needs PLL setup time when the HALT mode is released.

Caution: Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE2 mode.

18.5.2 Releasing IDLE2 mode

The IDLE2 mode is released by a non-maskable interrupt request (NMI pin input, INTWDT2 occurrence), unmasked external interrupt request (INTP0 to INTP7 pin input), unmasked internal interrupt request from the peripheral functions operable in the software IDLE2 mode, or reset signals.
After the IDLE2 mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

(1) Releasing IDLE2 mode by non-maskable interrupt request or unmasked maskable interrupt request

The IDLE2 mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the IDLE2 mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Caution: An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI1M, and PSC.INTM bits to 1 becomes invalid and IDLE2 mode is not released.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the IDLE2 mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the IDLE2 mode is released and that interrupt request is acknowledged.

Table 18-6: Operation After Releasing IDLE2 Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address after securing the prescribed setup time.	
Maskable interrupt request	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 $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-7: Operation Status in IDLE2 Mode

Setting of IDLE2 Mode Item		Operation Status
Main clock oscillator (f_X)		Oscillation enabled
Ring clock generator (f_R)		Oscillation enabled
PLL		Stops operation
Flash charge pump		Operable
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation (Standby mode release enabled)
Timer P (TMP0 to TMP3)		Stops operation
Timer Q (TMQ0, TMQ1)		Stops operation
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock
Watchdog timer 2 (WDT2)		Operable when f_R is selected as the count clock
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$ input clock is selected as operation clock
	UARTA0, UARTA1	Stops operation (Operable when ASCKA0 input clock is selected as operation clock)
	CSI30, CSI31	Stops operation
AFCAN0, AFCAN1		Stops operation
A/D converter		Stops operation
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before IDLE2 mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE2 mode was set.

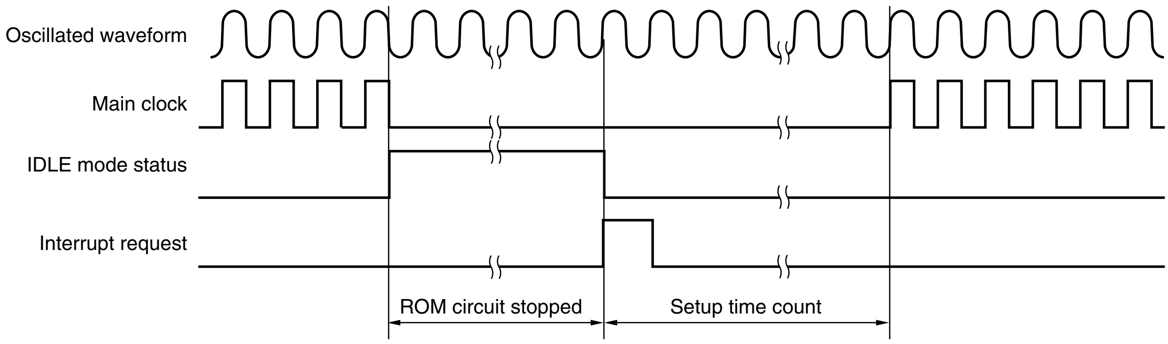
18.5.3 Securing setup time when releasing IDLE2 mode

Secure the setup time for the ROM (flash memory) after releasing the IDLE2 mode because the operation of the blocks other than the main clock oscillator stops after IDLE2 mode is set.

(1) Releasing IDLE2 mode by non-maskable interrupt request or unmasked maskable interrupt request

Secure the specified setup time by setting the OSTS register.
When the releasing source is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.

Figure 18-5: IDLE Mode Timing



(2) Release by reset input ($\overline{\text{RESET}}$ pin input, WDT2RES occurrence)

This operation is the same as that of a normal reset.
The oscillation stabilization time is the initial value of the OSTS register, $2^{13}/f_X$.

18.6 STOP Mode

18.6.1 Setting and operation status

The STOP mode is set when the PSM1, 0 bits of the PSMR register are set to 01 and the STP bit of the PSC register is set to 1 in the normal operation mode.
In the STOP mode, the main clock oscillator stops. Clock supply to the CPU and the on-chip peripheral functions is stopped.
As a result, program execution is stopped, and the contents of the internal RAM before the STOP mode was set are retained. The on-chip peripheral functions that operate with the clock oscillated by an external clock continue operating.
Table 18-8 shows the operation status in the STOP mode.
Because the STOP stops operation of the main clock oscillator, it reduces the current consumption to a level lower than the IDLE2 mode. If the external clock is not used, the power consumption can be minimized with only leakage current flowing.

Caution: Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

18.6.2 Releasing STOP mode

The STOP mode is released by a non-maskable interrupt request (NMI pin input, INTWDT2 occurrence), unmasked external interrupt request (INTP0 to INTP7 pin input), unmasked internal interrupt request from the peripheral functions operable in the STOP mode, or reset signals.
After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

Caution: An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI1M, and PSC.INTM bits to 1 becomes invalid and STOP mode is not released.

(1) Releasing STOP mode by non-maskable interrupt request or unmasked maskable interrupt request

The STOP mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the STOP mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the STOP mode is released and that interrupt request is acknowledged.

Table 18-8: Operation After Releasing STOP Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address after securing the prescribed setup time.	
Maskable interrupt request	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 STOP mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-9: Operation Status in STOP Mode

Item \ Setting of STOP Mode		Operation Status
Main clock oscillator (f_X)		Stops operation
Ring clock generator (f_R)		Oscillation enabled
PLL		Stops operation
Flash charge pump		Stops operation
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation (Standby mode release enabled)
Timer P (TMP0 to TMP3)		Stops operation
Timer Q (TMQ0, TMQ1)		Stops operation
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock
Watchdog timer 2 (WDT2)		Operable when f_R is selected as the count clock
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$ input clock is selected as operation clock
	UARTA0, UARTA1	Stops operation (Operable when ASCKA0 input clock is selected as operation clock)
	CSI30, CSI31	Stops operation
AFCAN0, AFCAN1		Stops operation
A/D converter		Stops operation
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before STOP mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the STOP mode was set.

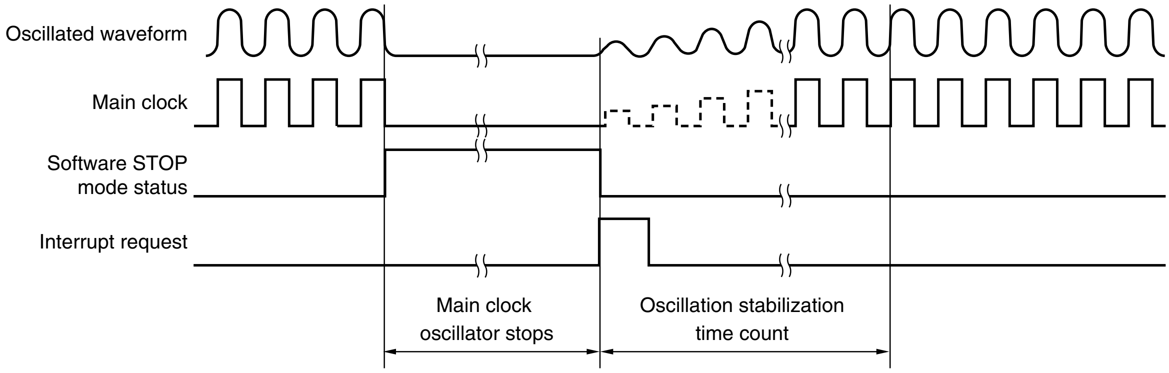
18.7 Securing Oscillation Stabilization Time

When the STOP mode is released, only the oscillation stabilization time set by the OSTS register elapses. If the STOP mode has been released by RESET pin input, however, the reset value of the OSTS register, $2^{16}/f_X$ elapses.

The timer for counting the oscillation stabilization time is shared with watchdog timer, so the oscillation stabilization time equal to the overflow time of the watchdog timer elapses.

Figure 18-6 shows the operation performed when the STOP mode is released by an interrupt request.

Figure 18-6: Oscillation Stabilization Time


Oscillated waveform
Main clock
Software STOP mode status
Interrupt request
Main clock oscillator stops
Oscillation stabilization time count

Caution: For details of the OSTS register, refer to 6.3 (7)"Oscillation stabilization time select register (OSTS)" on page 241.

18.8 Control Registers

(1) Power save control register (PSC)

This is an 8-bit register that controls the standby function. The STP bit of this register is used to specify the STOP mode. The PSC register is a special register (refer to **3.2.3 ”Special registers” on page 70**). Data can be written to this register only in a specific sequence so that its contents are not rewritten by mistake due to a program hang-up.
This register can be read or written in 8-bit or 1-bit units.

Figure 18-7: Power Save Control Register (PSC) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PSC	NMI2M	NMI1M	0	INTM	0	0	STP	0	FFFFF1FEH	00H
R/W	R/W	R/W	R	R/W	R	R	R/W	R		

NMI2M	Standby mode release control by NMI pin input
0	Enables standby mode release through NMI pin input
1	Disables standby mode release through NMI pin input

NMI1M	Standby mode release control upon occurrence of INTWDT2
0	Enables standby mode release through INTWDT2 signal
1	Disables standby mode release through INTWDT2 signal

INTM	Standby mode release control through maskable interrupt request
0	Enables standby mode release through maskable interrupt request
1	Disables standby mode release through maskable interrupt request

STP	Standby mode setting
0	Normal mode
1	Standby mode

Cautions:

1. **When set to IDLE1 mode, IDLE2 mode and STOP mode, set PSM1 and PSM0 bit to 1 on PSMR register then set STP bit to 1.**
2. **Settings of the NMI2M, NMI1M, and INTM bits are invalid when HALT mode is released.**
3. **Be sure to clear bit 5 to 0.**

(2) Power save mode register (PSMR)

This 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.

Figure 18-8: Power Save Mode Register (PSMR) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PSMR	0	0	0	0	0	0	PSM1	PSM0	FFFFF820H	00H
R/W	R	R	R	R	R	R	R/W	R/W		

PSM1	PSM0	Specifies operation in software standby mode (this bit becomes valid when bit 1 (STP) of the PSC register is set to 1)
0	0	IDLE1 mode
0	1	STOP mode
1	0	IDLE2 mode
1	1	Setting prohibited

Cautions: 1. Be sure to clear bits 2 to 7 of the PSMR register to 0.

2. The PSM0 and PSM1 bits are valid only when the STP bit of the PSC register is 1.

Remark: IDLE1: In this mode, all operations except the oscillator operation, flash memory, and PLL are stopped. After the IDLE1 mode is released, the normal mode need not wait the lapse of the oscillation stabilization time.

IDLE2: <Case of PLL not use>
In this mode, all operations except the oscillator operation are stopped.
After the IDLE2 mode is released, the normal mode is returned to following the lapse of the setup time (flash memory) specified by the OSTS register.
<Case of PLL use>
Refer to **CHAPTER 6.6.2 How to Use**.

STOP: In this mode, all operations are stopped.
After the STOP mode is released, the normal mode is returned to following the lapse of the oscillation stabilization time specified by the OSTS register.

Chapter 19 RESET Function

19.1 Overview

The following reset functions are available.

(1) Five 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 (see **Chapter 24 ”Low-Voltage Detector” on page 781**)
- System reset via the detecting clock monitor (CLM) oscillation stop (see **Chapter 25 ”Clock Monitor” on page 789**)
- System reset by power-on clear circuit (POC) (see **Chapter 23 ”Power-On-Clear Circuit” on page 779**).

(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 Ring-OSC.

Caution: When the CPU is being operated via the Ring-OSC, access to the register in which a wait state is generated is prohibited. For the register in which a wait state is generated, refer to 3.4.7 (3)”Accessing specific on-chip peripheral I/O registers” on page 89.

19.2 Registers to Check Reset Source

(1) Reset source flag register (RESF)

The RESF register indicates from which source a reset signal has been generated.

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

$\overline{\text{RESET}}$ input/POC reset clears this register to 00H. The default value differs if reset is effected from a source other than $\overline{\text{RESET}}$.

Figure 19-1: Reset Source Flag Register (RESF) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF	FFFFF888H	00H
R/W	R	R	R	R/W	R	R	R	R/W		

WDT2RF	Generation of reset signal from WDT2
0	Internal reset request has not occurred, or RESF is cleared.
1	Internal reset request has occurred.

CLMRF	Generation of reset signal from CLM
0	Internal reset request has not occurred, or RESF is cleared.
1	Internal reset request has occurred.

LVIRF	Generation of reset signal from LVI
0	Internal reset request has not occurred, or RESF is cleared.
1	Internal reset request has occurred.

Note: The value of this register is cleared to 00H when an external reset is executed via the $\overline{\text{RESET}}$ pin or when the Power on Clear occurred. When a reset is executed by the WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM), the reset flag of the corresponding register (WDT2RF bit, CLMRF bit, and LVIRF bit) is set to 1 (the other sources are held).

Caution: Only 0 can be written to each bit. If writing 0 and flag setting (occurrence of reset) conflict, flag setting takes precedence.

19.3 Operation

19.3.1 Reset operation by $\overline{\text{RESET}}$ 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 and the oscillation stabilization timer begins counting (reset value of OSTS register: $2^{13}/f_X$). When the OSTS timer elapses, then the CPU starts program execution.

Table 19-1: Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Continues oscillation	
Ring-OSC generator	Stops oscillation	Starts oscillation
Peripheral clock (f_{XX} to f_{XX}/1,024)	Stops operation	Starts operation after oscillation stabilization time (initialized to $f_{XX} \times 4$)
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Stops operation	Starts operation after oscillation stabilization time (initialized to $f_{XX} \times 4$)
CPU	Initialized	Program execution starts after oscillation stabilization time
Watchdog timer 2	Stops operation	Starts operation
Internal RAM	undefined after a reset while power is on or if a data access to RAM (by the CPU) and a reset input conflict (data corrupted). Otherwise, retains value immediately before reset input[Note].	
I/O lines (ports/alternate-function pins)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Note: Because the V850E/RS1 supports a boot swap function, the firmware uses part of the internal RAM after the internal system reset is released. Therefore, the contents of some areas (RAM size is 3FC800H-3FC895H into 10K byte and 3FB000H-3FB095H into 16K byte and 2K byte down from the uppermost address 3FEFFFH- 3FE7FFH) of the RAM are not retained even when power-on reset is executed.

Caution: The on-chip debug mode (flash memory products only) may be set depending on the pin status after reset has been released. For details, see Chapter 4 ”Port Functions” on page 105 and Chapter 22 ”On-Chip Debug Function” on page 767.

Figure 19-2: Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

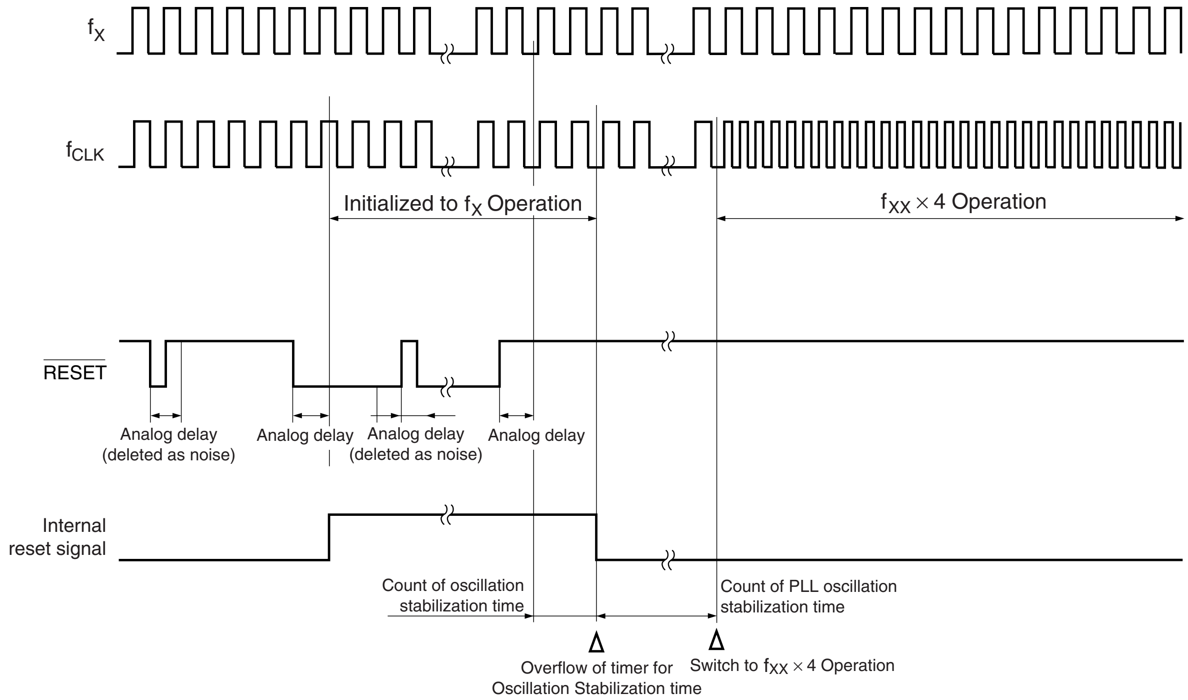
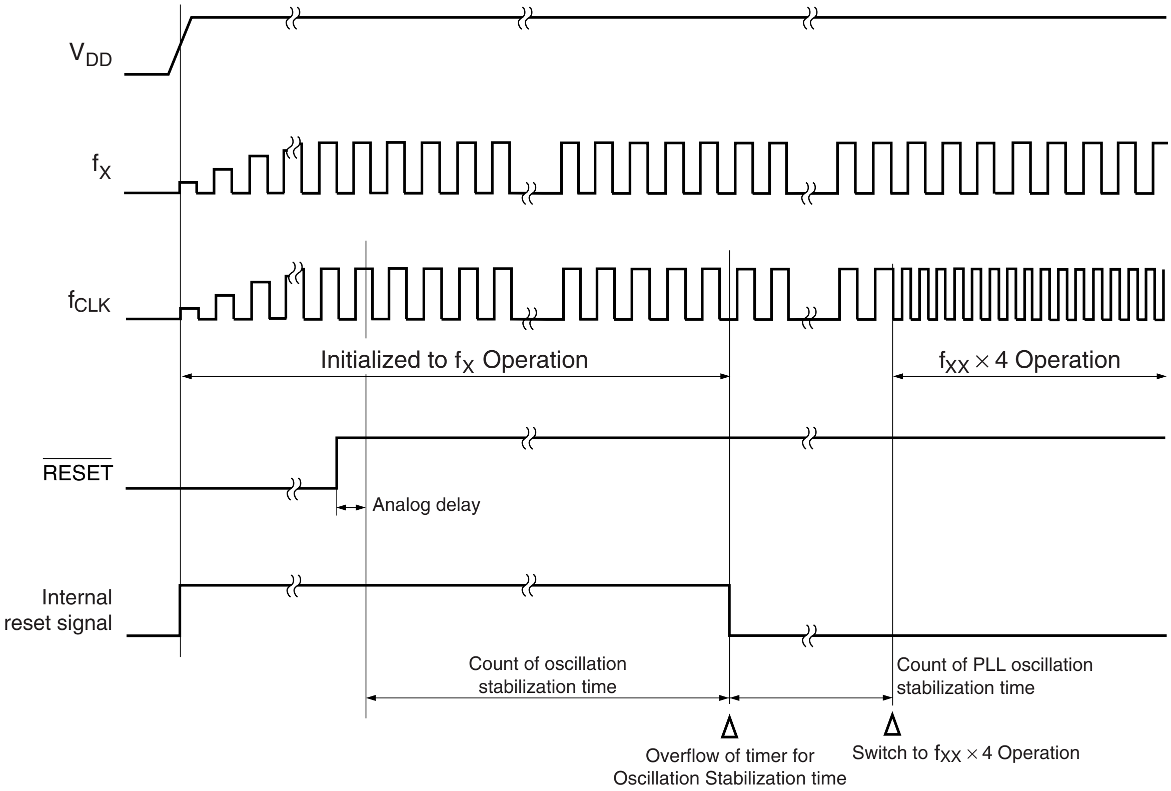


Figure 19-3: Timing of Power-on Reset Operation



19.3.2 Reset operation by WDT2RES signal

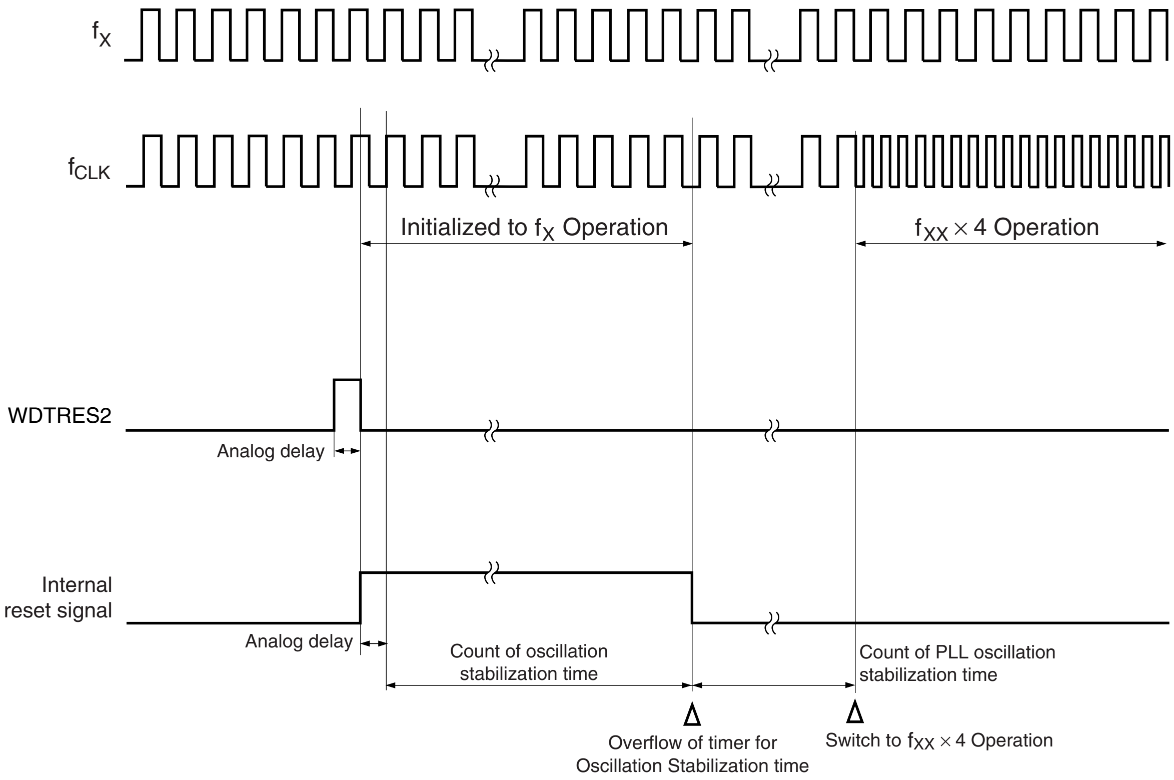
When watchdog timer 2 is set to the reset operation mode due to overflow, upon watchdog timer 2 overflow (WDT2RES signal generation), a system reset is executed and the hardware is initialized to the initial status.
Following watchdog timer 2 overflow, the reset status is entered and lasts the period of time predetermined by the analog delay, then the reset status is automatically released. Following reset release, the CPU starts program execution after securing the oscillation stabilization time (initial value of OSTS register: $2^{13}/f_X$) of the main clock oscillator.
The main clock oscillator is not stopped during the reset period.

Table 19-2: Hardware Status During WDT2RES Signal Generation

Item	During Reset	After Reset
Main clock oscillator (f_X)	Continues oscillation	
Ring-OSC generator	Continues oscillation	
Peripheral clock (f_{XX} to f_{XX}/1,024)	Operation stops	Operation starts after securing of oscillation stabilization time (initialized to f_{XX} x 4)
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing of oscillation stabilization time (initialized to f_{XX} x 4)
CPU	Initialized	Program execution starts after securing of oscillation stabilization time
Watchdog timer 2	Operation stops	Operation starts
Internal RAM	undefined if power-on reset or writing data to RAM (by CPU) 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 registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Note: Because the V850E/RS1 supports a boot swap function, the firmware uses part of the internal RAM after the internal system reset is released. Therefore, the contents of some areas (RAM size is 3FC800H-3FC895H into 10K byte and 3FB000H-3FB095H into 16K byte and 2K byte down from the uppermost address 3FEFFFH- 3FE7FFH) of the RAM are not retained even when power-on reset is executed.

Figure 19-4: Timing of Reset Operation by WDT2RES Signal Generation


f_X
f_{CLK}
Initialized to f_X Operation
$f_{XX} \times 4$ Operation
WDTRES2
Analog delay
Internal reset signal
Analog delay
Count of oscillation stabilization time
Count of PLL oscillation stabilization time
Overflow of timer for Oscillation Stabilization time
Switch to $f_{XX} \times 4$ Operation

19.3.3 Reset operation by Low Voltage Detector and Power On Clear

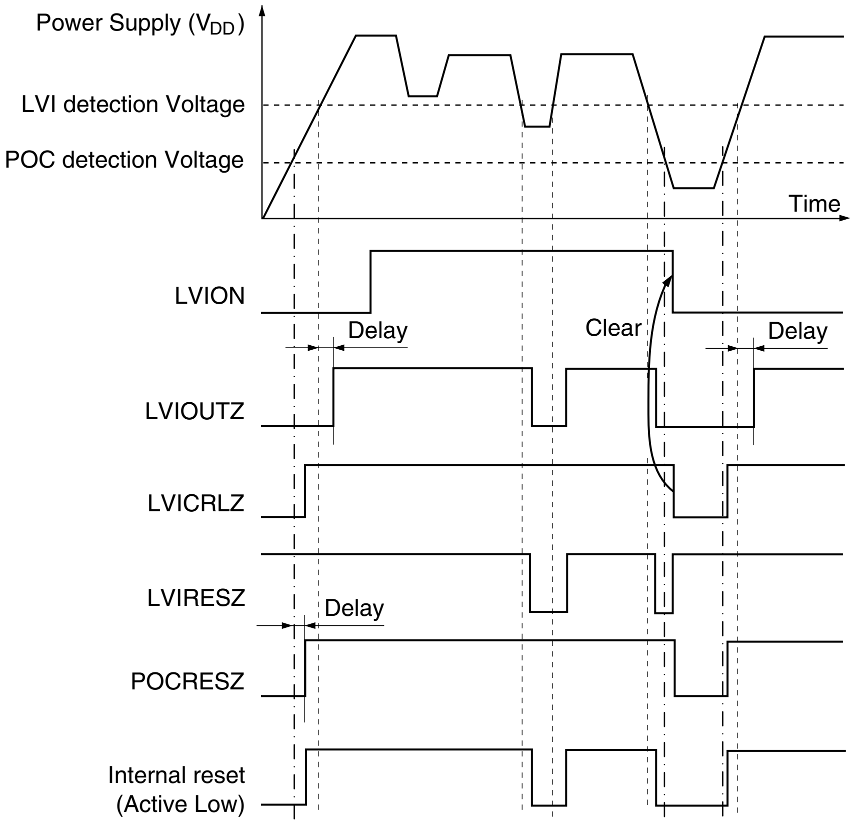
If the V_{DD1} power supply falls below a voltage level configured in the low-voltage detector and LVI operation is enabled in reset mode (when the LVIMD bit of the LVIM register is set to 1), a system reset is executed, and the hardware is initialized.
The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the LVI detection voltage level. Following reset release, the CPU starts boot processing execution after securing the oscillation stabilization time (initial value of OSTS register: $2^{13}/f_X$) of the main clock oscillator.
The main clock oscillator is not stopped during the reset period.

Table 19-3: Hardware Status During Reset Operation by Low-Voltage Detector

Item	During Reset	After Reset
Main clock oscillator (f_X)	Continues oscillation	
Ring-OSC generator	Continues oscillation	
Peripheral clock (f_{XX} to $f_{XX}/1,024$)	Operation stops	Operation starts after securing of oscillation stabilization time (initialized to $f_{XX} \times 4$)
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing of oscillation stabilization time (initialized to $f_{XX} \times 4$)
CPU	Initialized	Program execution starts after securing of oscillation stabilization time
Watchdog timer 2	Operation stops	Operation starts
Internal RAM	undefined if power-on reset or writing data to RAM (by CPU) 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 registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Note: Because the V850E/RS1 supports a boot swap function, the firmware uses part of the internal RAM after the internal system reset is released. Therefore, the contents of some areas (RAM size is 3FC800H-3FC895H into 10K byte and 3FB000H-3FB095H into 16K byte and 2K byte down from the uppermost address 3FEFFFH- 3FE7FFH) of the RAM are not retained even when power-on reset is executed.

Figure 19-5: Timing of Reset Operation by Low-Voltage Detector



Notes: 1. The time period between a dashed line and the rising or falling edge of a control signal represents the minimum analog delay in the circuit.

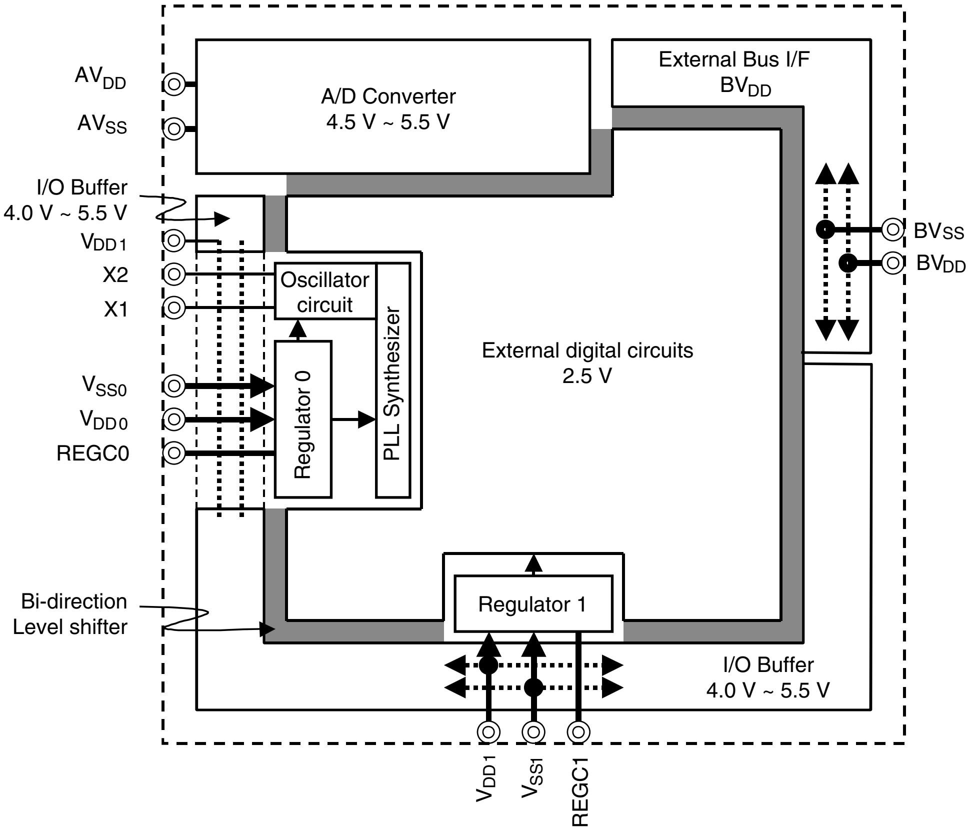
2. The LVION and LVI reset signal are set to the inactive state by the POC reset signal active level.

Chapter 20 Regulator

20.1 Outline

The V850E/RS1 has an on-chip regulator to lower the 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 I/O buffer).
The regulator output voltage is set to 2.5 V (± 0.2 V).

Figure 20-1: Regulator Block Diagram

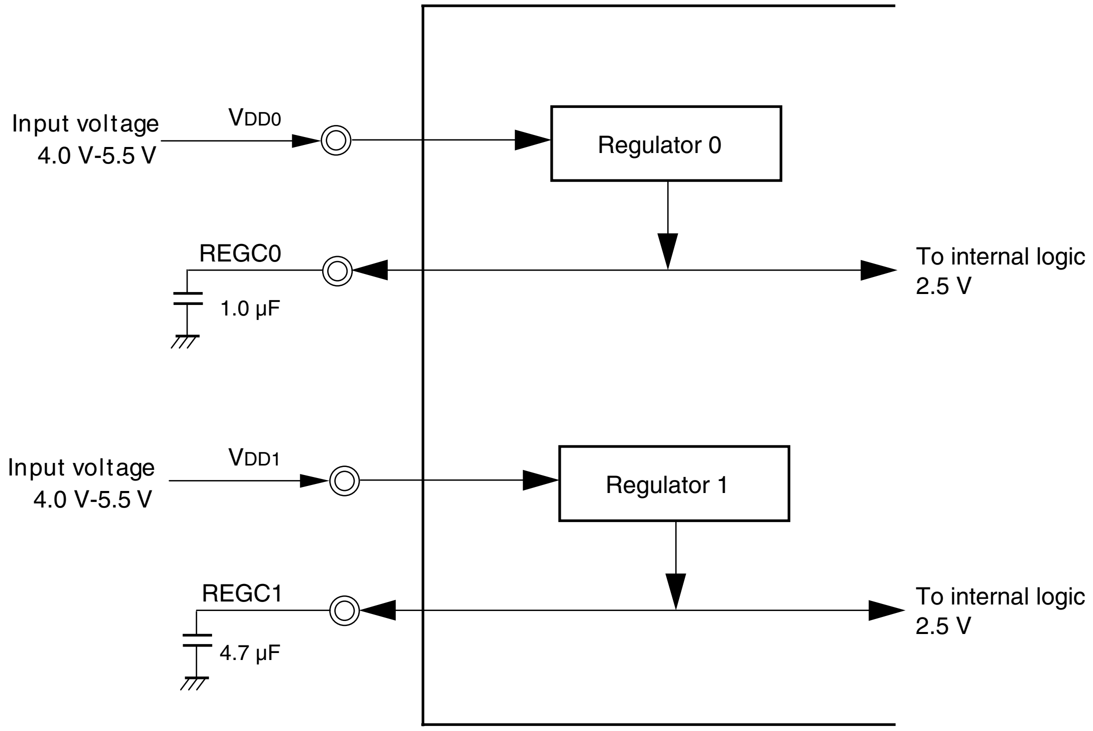


20.2 Operation

The regulator operates in all modes (Normal mode, HALT mode, IDLE1 mode, IDLE2 mode, STOP mode, and during RESET).
Be sure to connect a capacitor (REGC0: 4.7 µF, REGC1: 1.0 µF) to the REGC0 and REGC1 pin to stabilize the regulator output.

A diagram of the regulator pin connections is shown below.

Figure 20-2: REGC Pin Connection (REGC = Capacity)



Chapter 21 Flash Memory

The following products are the flash memory versions of the V850E/RS1.

Caution: There are differences in the amount of noise tolerance and noise radiation between flash memory versions and mask ROM versions. When considering changing from a flash memory version to a mask ROM version during the process from experimental manufacturing to mass production, make sure to sufficiently evaluate commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.

- µPD70F3403 and µPD70F3403A
  On-chip 256 KB flash memory
- µPD70F3402
  On-chip 128 KB flash memory

In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock, the same as in the mask ROM version.
The flash memory can be written mounted on the target board (on-board write), by connecting a dedicated flash programmer to the target system.
Flash memory is commonly used in the following development environments and applications.

- For altering software after solder-mounting the V850E/RS1 on the target system
- For differentiating software in small-scale production of various models.
- For data adjustment when starting mass production

21.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- Chip erase/block unit erase
- Communication via serial interface with the dedicated flash programmer
- Erase/write voltage: Can be erased/written with a single power supply (FLMD0 = V_{DD}, FLMD1 = V_{SS}).
- On-board programming
- Flash memory programming by self-writing

21.2 Erasure Unit

The units in which the 256 or 128 KB flash memory can be erased are as follows.

(1) Chip erase

The areas of xx000000H to xx01FFFFH and xx000000H to xx03FFFFH can be erased at the same time.

(2) Block erase

Erasure can be performed in block units[Note].

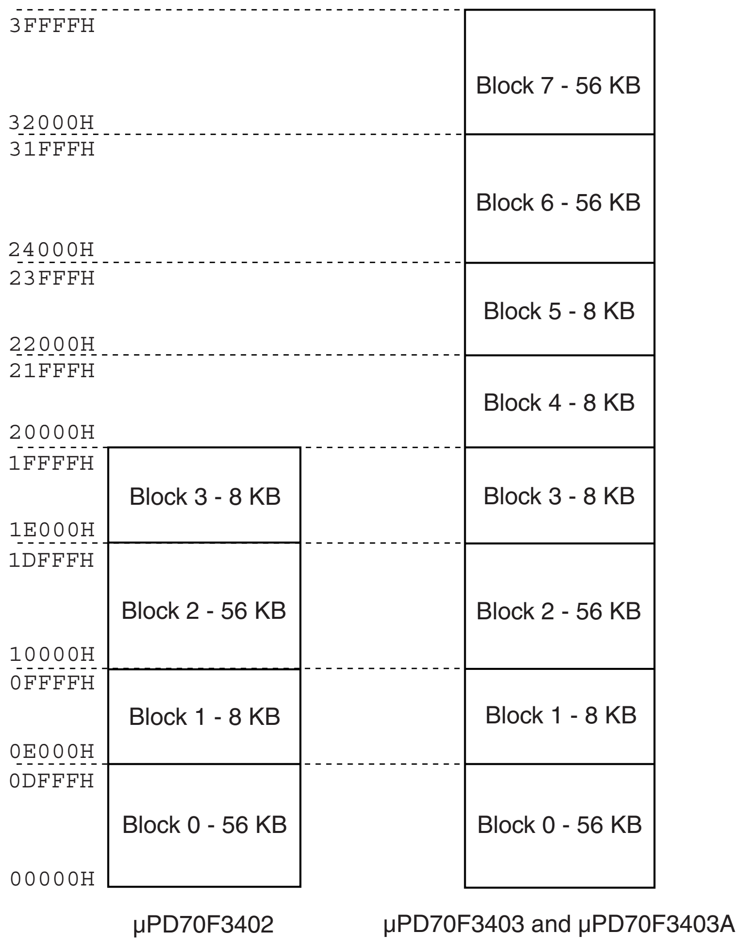
Block 0:	56 KB
Block 1:	8 KB
Block 2:	56 KB
Block 3:	8 KB
Block 4:	8 KB
Block 5:	8 KB
Block 6:	56 KB
Block 7:	56 KB

Note: 8 blocks, blocks 0 to 7, for the 256 KB version (μPD70F3403 and μPD70F3403A).

21.2.1 Address assignment in the memory map

Each block of the flash device is assigned to the address space of the internal ROM.

Figure 21-1: Address Assignment of Flash Blocks for V850E/RS1



21.3 Writing with Flash Programmer

A dedicated flash programmer can be used for on-board or off-board writing of the flash memory.

(1) On-board programming

The contents of the flash memory can be rewritten with the V850E/RS1 mounted on the target system.
Mount a connector that connects the dedicated flash programmer on the target system.

(2) Off-board programming

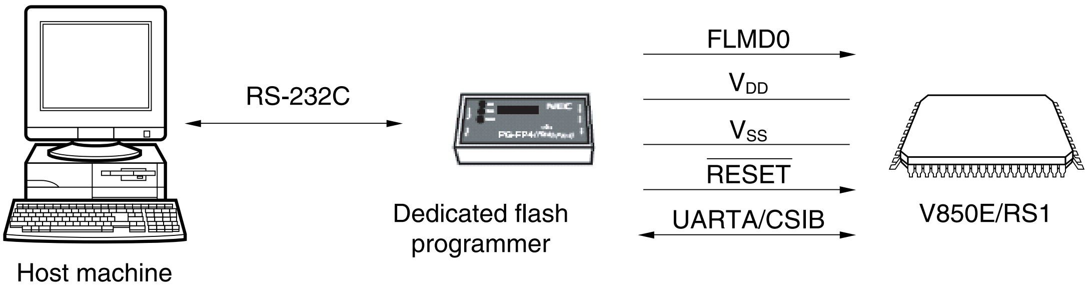
The flash memory of the V850E/RS1 can be written before the device is mounted on the target system, by using a dedicated program adapter (FA series).

Remark: FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

21.4 Programming Environment

The following shows the environment required for writing programs to the flash memory of V850E/RS1.

Figure 21-2: Environment Required for Writing Programs to Flash Memory



A host machine is required for controlling the dedicated flash programmer.
UARTA0 or CSIB0 is used for the interface between the dedicated flash programmer and the V850E/RS1 to manipulate the flash programmer by writing or erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

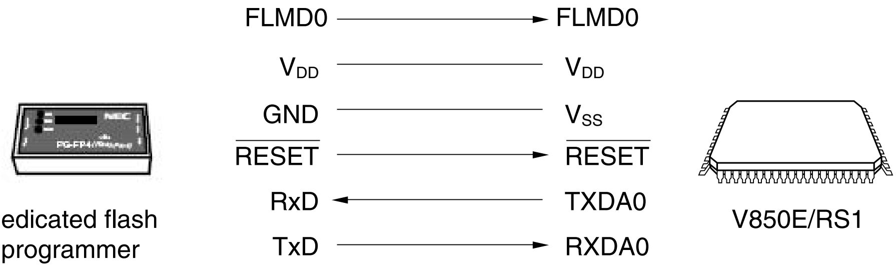
21.5 Communication Mode

Serial communication is performed between the dedicated flash programmer and the V850E/RS1 by using UARTA0 or CSIB0 of the V850E/RS1.

(1) UARTA0

Transfer rate: 9,600 to 153,600 bps

Figure 21-3: Communication with Dedicated Flash Programmer (UARTA0)


FLMD0
FLMD0
V_{DD}
V_{DD}
GND
V_{SS}
$\overline{\text{RESET}}$
$\overline{\text{RESET}}$
RxD
TXDA0
TxD
RXDA0
Dedicated flash programmer
V850E/RS1

Cautions: 1. Process the pins not shown in accordance with processing of unused pins (see 2.4 ”Pin I/O Circuit Types, I/O Buffer Power Supply and Handling of Unused Pins” on page 57). To connect a resistor, a resistor of 1 k to 10 Ω is recommended.

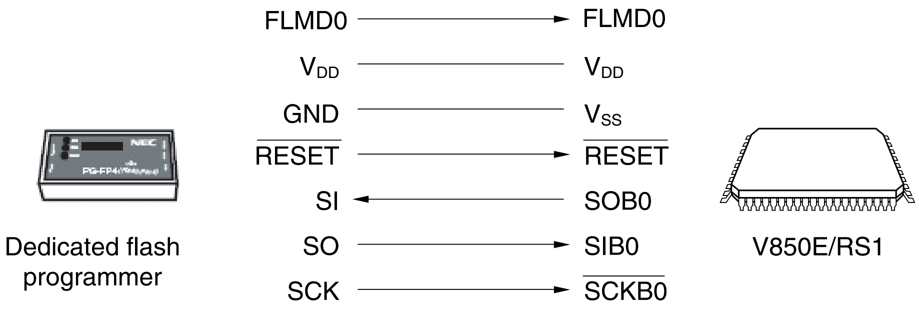
2. Please do not input high level in $\overline{\text{DRST}}$ pin.

3. Please pull down FMLD1 without fail.

(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 21-4: Communication with Dedicated Flash Programmer (CSIB0)


FLMD0
FLMD0
V_{DD}
V_{DD}
GND
V_{SS}
$\overline{\text{RESET}}$
$\overline{\text{RESET}}$
SI
SOB0
SO
SIB0
SCK
$\overline{\text{SCKB0}}$
Dedicated flash programmer
V850E/RS1

Cautions: 1. Process the pins not shown in accordance with processing of unused pins (see 2.4 ”Pin I/O Circuit Types, I/O Buffer Power Supply and Handling of Unused Pins” on page 57). To connect a resistor, a resistor of 1 k to 10 Ω is recommended.

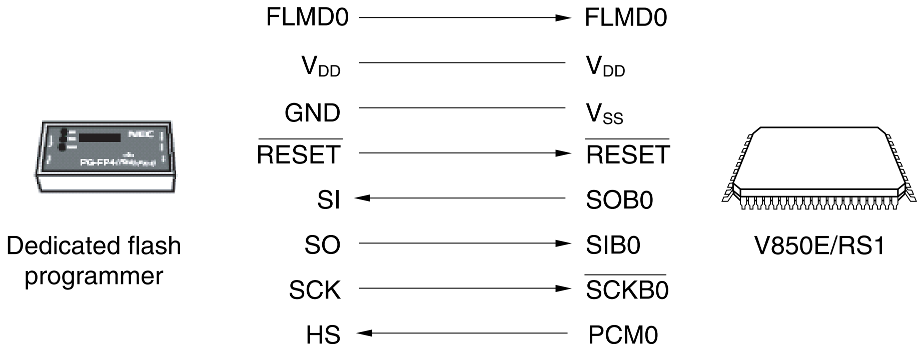
2. Please do not input high level in $\overline{\text{DRST}}$ pin.

3. Please pull down FMLD1 without fail.

(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 21-5: Communication with Dedicated Flash Programmer (CSIB0 + HS)


FLMD0
FLMD0
V_{DD}
V_{DD}
GND
V_{SS}
$\overline{\text{RESET}}$
$\overline{\text{RESET}}$
SI
SOB0
SO
SIB0
SCK
$\overline{\text{SCKB0}}$
HS
PCM0
Dedicated flash programmer
V850E/RS1

Cautions: 1. Process the pins not shown in accordance with processing of unused pins (see 2.4 ”Pin I/O Circuit Types, I/O Buffer Power Supply and Handling of Unused Pins” on page 57). To connect a resistor, a resistor of 1 k to 10 Ω is recommended.

2. Please do not input high level in $\overline{\text{DRST}}$ pin.

3. Please pull down FMLD1 without fail.

The dedicated flash programmer outputs the transfer clock, and the V850E/RS1 operates as a slave. When the PG-FP4 is used as the dedicated flash programmer, it generates the following signals to the V850E/RS1. For details, refer to the **PG-FP4 manual (U15260E)**.

Table 21-1: Signal Generation of Dedicated Flash Programmer (PG-FP4)

PG-FP4			V850E/RS1	Connection Handling		
Signal Name	I/O	Pin Function	Pin Name	CSIB0	UARTA0	CSIB0 + HS
FLMD0	Output	Writing enable/disable	FLMD0	◎	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	◎	◎	◎
GND	-	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output to V850E/RS1	X1	×	×	×
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI/RXD	Input	Receive signal	SOB0/TXDA0	◎	◎	◎
SO/TXD	Output	Transmit signal	SIB0/RXDA0	◎	◎	◎
$\overline{\text{SCK}}$	Output	Transfer clock	$\overline{\text{SCKB0}}$	◎	×	◎
HS	Input	Handshake signal of CSI0 + HS communication	PCM0	×	×	◎

Remark: ◎: Always connected
×: Does not need to be connected

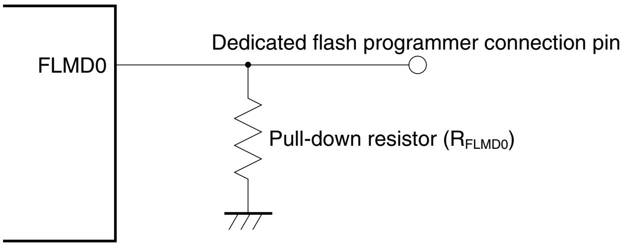
21.6 Pin Connection

A connector must be mounted on the target system to connect the dedicated flash programmer for on-board writing.
In addition, a function to switch between the normal operation mode and flash memory programming mode must be provided on the board.
When the flash memory programming mode is set, all the pins not used for flash memory programming are in the same status as that immediately after reset. Therefore, all the ports go into an output high-impedance state, and the pins must be processed correctly if the external device does not recognize the output high-impedance state.

21.6.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An example of connection of the FLMD0 pin is shown below.

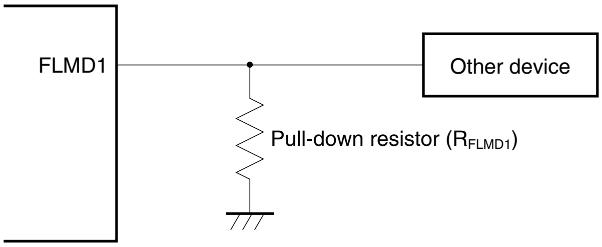
Figure 21-6: FLMD0 Pin Connection Example


FLMD0
Dedicated flash programmer connection pin
Pull-down resistor (R_{FLMD0})

21.6.2 FLMD1 pin

If 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. If VDD is supplied to the FLMD0 pin, 0 V must be input to the FLMD1 pin to set the flash memory programming mode. An example of the connection of the FLMD1 pin is shown below.

Figure 21-7: 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 21-2: Relationship Between FLMD0 and FLMD1 Pins and Operation Mode

FLMD0	FLMD1	Operation Mode
0	×	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

21.6.3 Serial interface pin

The pins used by each serial interface are shown in the table below.

Table 21-3: Pins Used by each Serial Interface

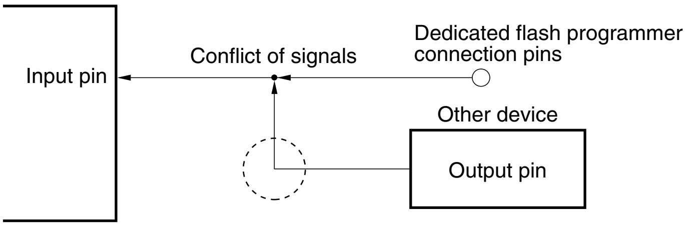
Serial Interface	Pins Used
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
UARTA0	TXDA0, RXDA0

When connecting a dedicated flash programmer to a serial interface pin that is connected to another device on board, exercise care so that signal conflict and malfunction of the other device do not occur.

(1) 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 in an output high-impedance state.

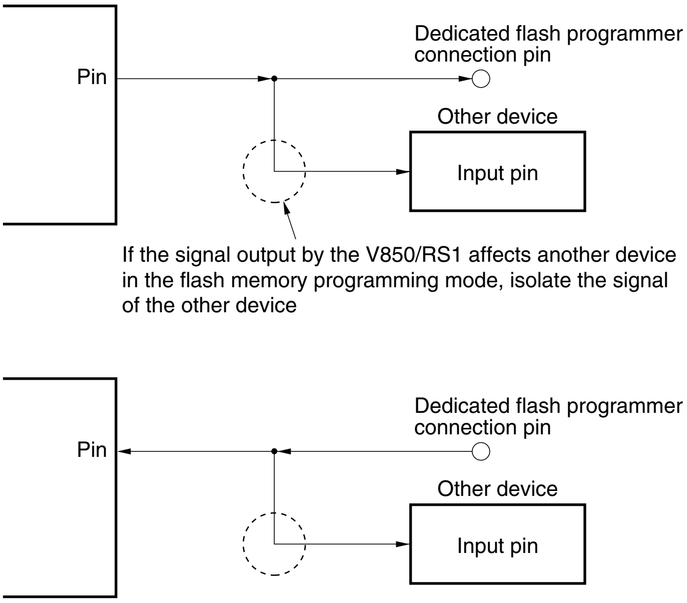
Figure 21-8: Signal Conflict (Input Pin of Serial Interface)



(2) Abnormal operation of other device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) connected to another device (input), a signal is output to the other device, causing a malfunction. To avoid this malfunction, isolate the connection with the other device, or set so that the other device ignores an input signal from V850E/RS1.

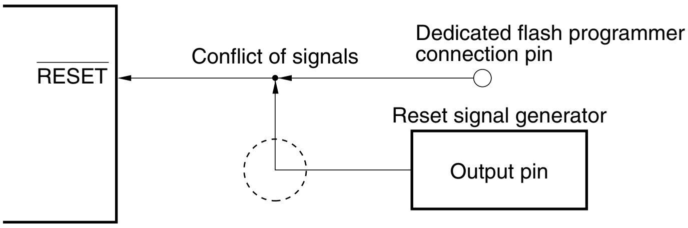
Figure 21-9: Abnormal Operation of Other Device


Pin
Dedicated flash programmer connection pin
Other device
Input pin
If the signal output by the V850/RS1 affects another device in the flash memory programming mode, isolate the signal of the other device
Pin
Dedicated flash programmer connection pin
Other device
Input pin

21.6.4 $\overline{\text{RESET}}$ pin

When the reset signal of the dedicated flash programmer is connected to the RESET pin connected to a reset signal generator on board, a signal conflict occurs. To avoid this signal conflict, isolate the connection with the reset signal generator.
If a reset signal is input from the user system in flash memory programming mode, the programming operation is not performed correctly. Do not input a reset signal other than that from the dedicated flash programmer.

Figure 21-10: Signal Conflict ($\overline{\text{RESET}}$ Pin)


$\overline{\text{RESET}}$
Conflict of signals
Dedicated flash programmer connection pin
Reset signal generator
Output pin

21.6.5 Port pins (including NMI)

All the port pins, including the pin connected to the dedicated flash programmer, go into an output high-impedance state in the flash memory programming mode. If there is a problem such as that the external device connected to a port prohibits the output high-impedance state, connect the port to VDD or VSS via a resistor.

21.6.6 Other signal pins

Connect X1, and X2 pins in the same status as in the normal operation mode.

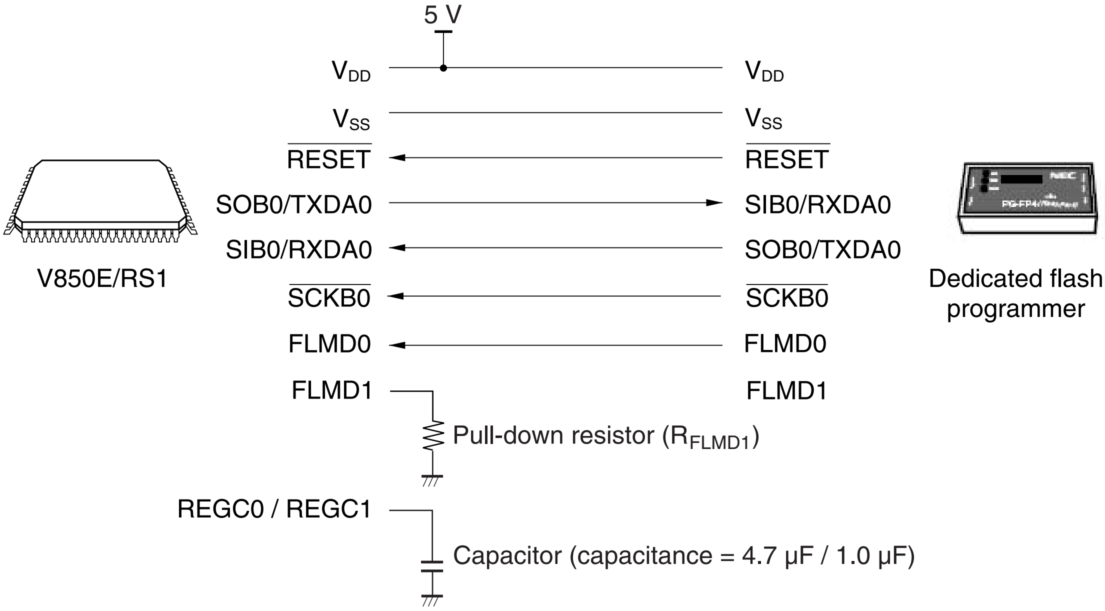
21.6.7 Power supply

Supply the same power to the supply power pins (V_{DD0}, V_{DD1}, V_{SS0}, $V_{SS1,}$ BV_{DD}, BV_{SS}, AV_{SS}, AV_{REF0}) as in the normal operation mode.

21.7 Recommended Circuit Example of the Flash Write Mode

Figure 21-11 shows the recommended circuit example of the flash write mode.

Figure 21-11: Recommended Circuit Example

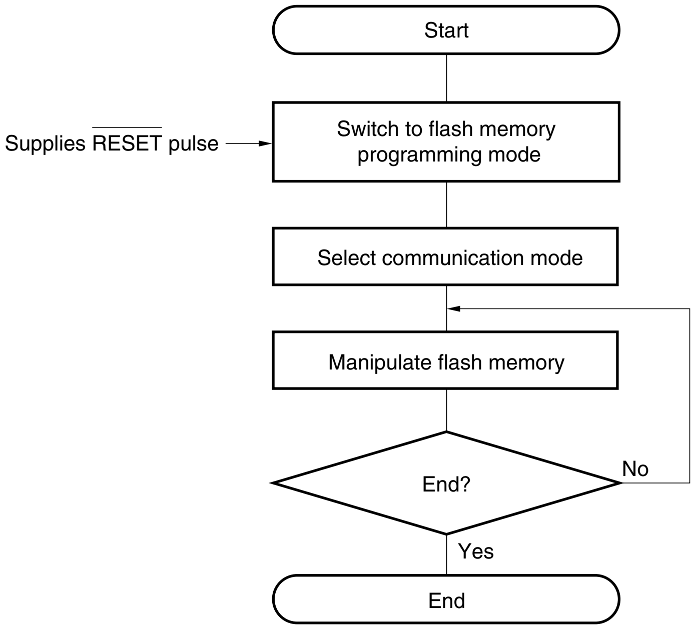


21.8 Programming Method

21.8.1 Flash memory control

The procedure to manipulate the flash memory is illustrated below.

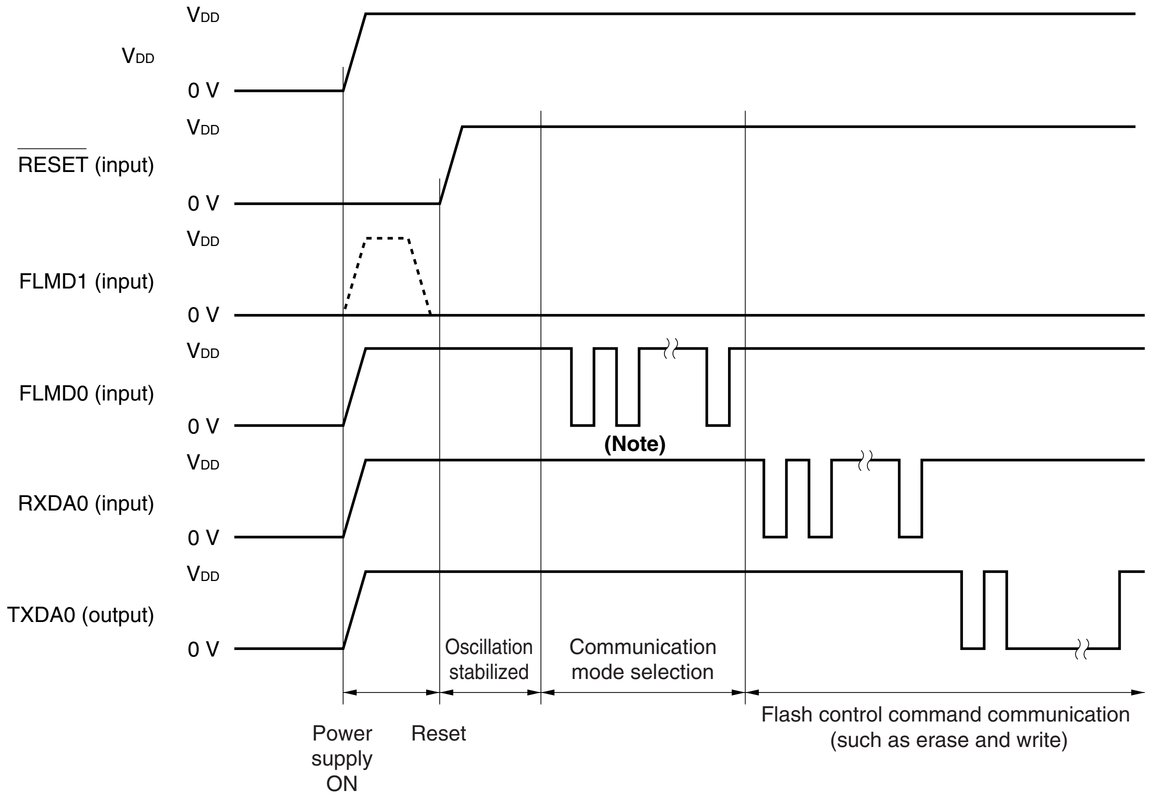
Figure 21-12: Procedure for Manipulating Flash Memory


Start
Supplies $\overline{\text{RESET}}$ pulse
Switch to flash memory programming mode
Select communication mode
Manipulate flash memory
End?
No
Yes
End

21.8.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash programmer, set the V850E/RS1 in the flash memory programming mode.
To set the mode, set the FLMD0 and FLMD1 pins, and release reset.
Change the mode by using a jumper when performing on-board writing.

Figure 21-13: Flash Memory Programming Mode



Note: The number of clocks to be inserted differs depending on the communication mode. For details, refer to Table 21-4, “List of Communication Modes,” on page 764.

21.8.3 Selection of communication mode

In the V850E/RS1, the communication mode is selected by inputting pulses (12 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.
The following shows the relationship between the number of pulses and the communication mode.

Table 21-4: List of Communication Modes

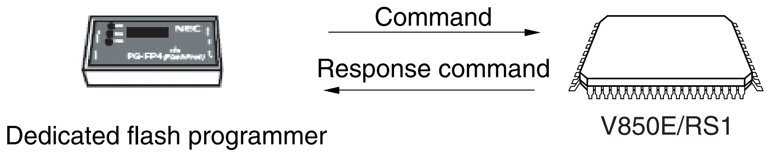
FLMD0 Pulse	Communication Mode	Remarks
8	CSIB0	V850E/RS1 operates as slave, MSB first
11	CSIB0 + HS	V850E/RS1 operates as slave, MSB first
0	UARTA0	Communication rate: 9,600 bps (after reset), LSB first
Others	RFU	Setting prohibited

Caution: When UARTA is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after reception of the FLMD0 pulse.

21.8.4 Communication command

The V850E/RS1 communicates with the dedicated flash programmer via commands. The commands sent by the dedicated flash programmer to the V850ES/FG2 are called commands, and the response signals sent by the V850E/RS1 to the flash programmer are called response commands.

Figure 21-14: Communication Command



The following table lists the flash memory control commands of the V850E/RS1. All these commands are issued by the programmer, and the V850E/RS1 performs the corresponding processing.

Table 21-5: Flash Memory Control Command

Category	Command Name	Support			Function
		CSIB	CSIB + HS	UARTA	
Blank check	Block blank check command	√	√	√	Checks the erase state of the entire memory.
Erase	Chip erase command	√	√	√	Erases all memory contents.
	Block erase command	√	√	√	Erases memory contents of specified block.
Write	Write command	√	√	√	Writes data by specifying write address and number of bytes to be written, and executes verify check.
Verify	Verify command	√	√	√	Compares input data with all memory contents.
System setting and control	Reset command	√	√	√	Escapes from each state.
	Oscillating frequency setting command	√	√	√	Sets oscillation frequency.
	Baud rate setting command	–	–	√	Sets baud rate when UARTA is used.
	Silicon signature command	√	√	√	Reads silicon signature information.
	Version acquisition command	√	√	√	Reads version information of device.
	Status command	√	√	–	Acquires operation status.
	Security setting command	√	√	√	Sets security of chip erasure, block erasure, and writing.

The V850E/RS1 returns a response command in response to the command issued by the flash programmer. The response commands sent by the V850E/RS1 are listed below.

Table 21-6: Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data.
NAK (not acknowledge)	Acknowledges illegal command/data.

[MEMO]

Chapter 22 On-Chip Debug Function

The V850E/RS1 Series includes an on-chip debug unit. By connecting an N-Wire emulator, on-chip debugging can be executed with the V850E/RS1 microcontroller alone.

Caution: The following debug functions are supported by the V850E/RS1 series, but whether they are usable or not differs depending on the debugger. For details of the debugging function, refer to the user's manual of the debugger to be used.

22.1 Functional Outline

22.1.1 Type of on-chip debug unit

The on-chip debug unit of the V850E/RS1 series is RCU1 (Run Control Unit 1).

22.1.2 Debug functions

The on-chip debug function supports the following operations implemented in hardware or with support from the integrated debugger. For details on these functions, please refer to the user's manual of the ID850-NW debugger.

(1) Debug interface

Communication with the host machine is established by using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO signals via an N-Wire emulator. The communication specifications of N-Wire are used for the interface.

(2) On-chip debug

On-chip debugging can be executed by preparing wiring and a connector for on-chip debugging on the target system. An N-Wire emulator is used as the connector that connects the emulator.
Clear the OCDM0 bit of the OCDM register (special register) to 0 when you use on-chip debug mode. Please refer to Table 4-8, "Port Type," on page 170 for details.

(3) Forced reset function

The V850E/RS1 can be forcibly reset.

(4) Break reset function

The CPU can be started in the debug mode immediately after reset of the CPU is released.

(5) Forced break function

Execution of the user program can be forcibly aborted (however, the illegal operation code exception handler (first address: 00000060H) cannot be used).

(6) Hardware break function

Two breakpoints for instruction and access can be used. The instruction breakpoint can abort program execution at any address. The access breakpoint can abort program execution by data access to any address.

(7) Software break function

Up to four software breakpoints can be set in the internal ROM area. The number of software breakpoints that can be set in the RAM area differs depending on the debugger to be used.

(8) Debug monitor function

A memory space for debugging that is different from the user memory space is used during debugging (background monitor mode). The user program can be executed starting from any address.
While execution of the user program is aborted, the user resources (such as memory and I/O) can be read and written, and the user program can be downloaded.

(9) Mask function

Each signal can be masked.
The correspondence with the mask functions of the debugger (ID850NWC) for the N-Wire emulator (IEV850E1-CD-NW) of NEC Electronics is shown below.

- NMI0 mask function: –
- NMI1 mask function: WDT2 interrupt
- NMI2 mask function: NMI pin
- RESET mask function: $\overline{\text{RESET}}$ pin, WDT2 reset, POC reset, LVI reset, clock monitor reset

(10) Timer function

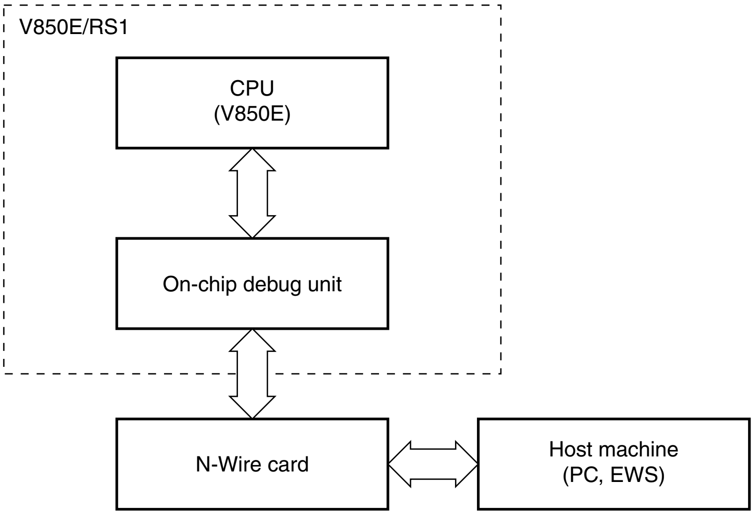
The execution time of the user program can be measured.

(11) Peripheral macro operation/stop selection function during break

Depending on the debugger to be used, whether the peripheral macro operates or is stopped during a break can be selected.

- Functions that are always stopped during break
  - Clock monitor
  - Watchdog timer 2
- Functions that can operate or be stopped during break (however, each function cannot be selected individually)
  - A/D converter
  - Timer M
  - Timer P
  - Timer Q
  - Watch timer
- Peripheral functions that continue operating during break (functions that cannot be stopped)
  - Peripheral functions other than above

Figure 22-1: Block Diagram of On-chip Debug Function



22.2 Security Function

The V850E/RS1 series have a security function that limits starting of the N-Wire emulator by comparing an ID code written in advance to the internal ROM area with an ID code that is input when the debugger is started, when the N-Wire emulator is connected. This function prevents a third party from reading the internal ROM area.

(1) ID code

Be sure to write an ID code when writing a program to the internal ROM.

The area of the ID code is 10 bytes wide and in the range of addresses 00000070H to 00000079H.

The ID code when the memory is erased is shown below.

Table 22-1: ID Code

Address	ID Code
00000079H	FFH
00000078H	FFH
00000077H	FFH
00000076H	FFH
00000075H	FFH
00000074H	FFH
00000073H	FFH
00000072H	FFH
00000071H	FFH
00000070H	FFH

(2) Security bit

Bit 7 of address 00000079H enables or disables use of the N-Wire emulator.

- Bit 7 of address 00000079H
  0: Disable
  1: Enable

Cautions: 1. If the value of address 00000079H is 00H to 7FH, the N-Wire emulator cannot be connected.

2. If the value of address 00000079H is 80H to FFH, the N-Wire emulator can be connected if the 10-byte ID code to be input when the N-Wire emulator is connected matches.

22.3 Control Register

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

This register is used to select the normal operation mode or on-chip debug mode. This is a special register (refer to 3.2.3 ”Special registers” on page 70). It can be written only in a specific sequence so that its contents cannot be rewritten by mistake in the case of a program loop.
If the OCDM0 bit is 1 and if the $\overline{\text{DRST}}$ pin is high, the on-chip debug mode is selected.
The default value of the OCDM0 bit after the pin is reset is 1. It is therefore necessary to clear the OCDM0 bit to 0 when the on-chip debug function is not used, and until then, the $\overline{\text{DRST}}$ pin must be kept low (see Figure 22-4). The $\overline{\text{DRST}}$ pin is internally pulled down while the OCDM0 bit is 1, and therefore, it may be left open.
After POC reset, the default value of the OCDM1 bit is 0, and the normal operation mode is selected.
Therefore, it is necessary to set the OCDM0 bit to 1 by resetting the pin to use the on-chip debug mode.
If POC reset occurs during on-chip debugging, communication with the emulator is disrupted. Therefore, POC reset cannot be emulated (see Figure 22-5).

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

Figure 22-2: On-Chip Debug Mode Register (OCDM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
OCDM	0	0	0	0	0	0	0	OCDM0	FFFFF9FCH	01H[Note 1]
R/W	R	R/W	R/W	R/W	R	R	R	R/W		

OCDM0	Specification of alternate-function pin of on-chip debug function[Note 2]
0	Used as port/peripheral function pin
1	Used as on-chip debug pin

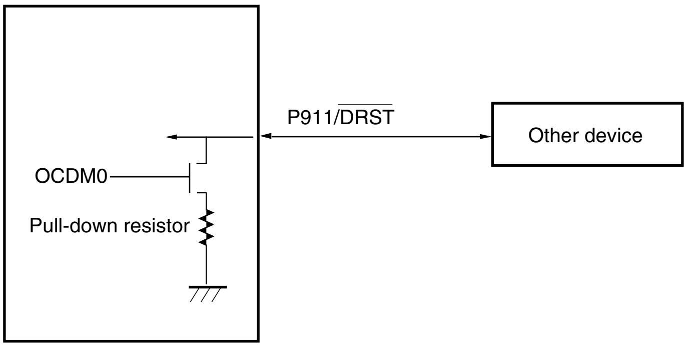
Notes: 1. On input to RESET pin (external reset): OCDM0 = 1
On reset by power-on clear: OCDM0 = 0
On occurrence of internal source reset (other than power-on clear): The OCDM register holds the value before occurrence of reset.

2. P97/SIB1/{DDI}
P98/SOB1/{DCK
P99/SCKB1/{DMS}
P910/CS301/{DDO}
P911/{DRST}

(2) Configuration of Port 9

While the OCDM0 bit is set, the corresponding pins of Port 9 are configured to the on-chip debug mode and digital input/output cannot be performed. Also, the on-chip pull down resistor connected to P911/ $\overline{\text{DRST}}$ is likewise controlled by the PD9 control register, but this function may not be disabled while the OCDM0 bit is set. Clearing the OCDM0 bit to 0 has the effect of enabling the digital I/O function on the corresponding pins of Port 9.

Figure 22-3: P911/ $\overline{\text{DRST}}$ Built-in Pull Down Resistor



22.4 Operation of On-Chip Debug Function

Figure 22-4: Timing Chart of Selecting Normal Operation Mode

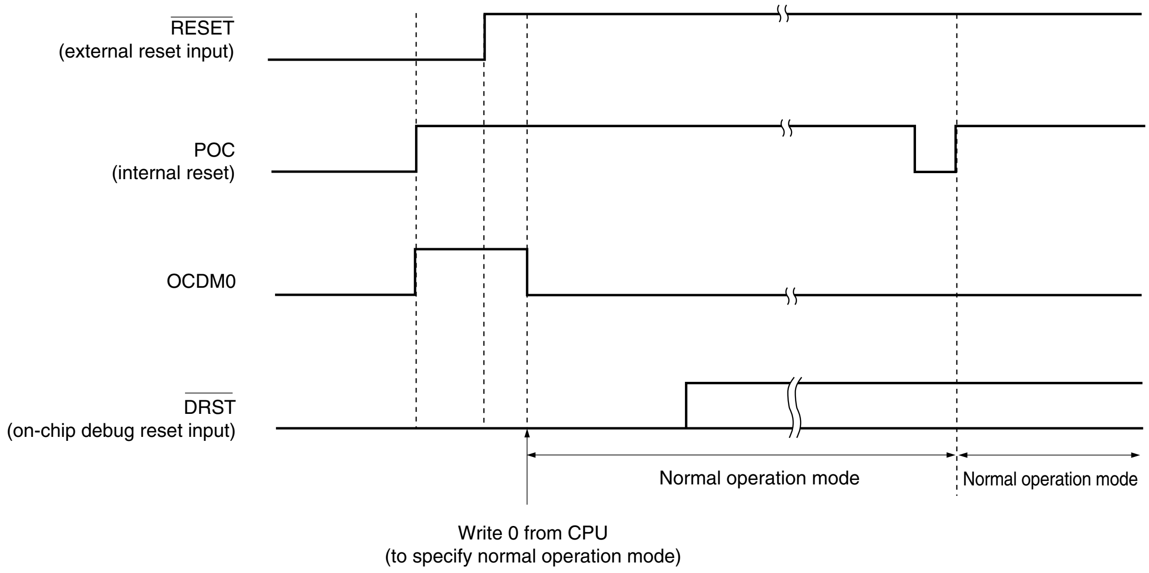
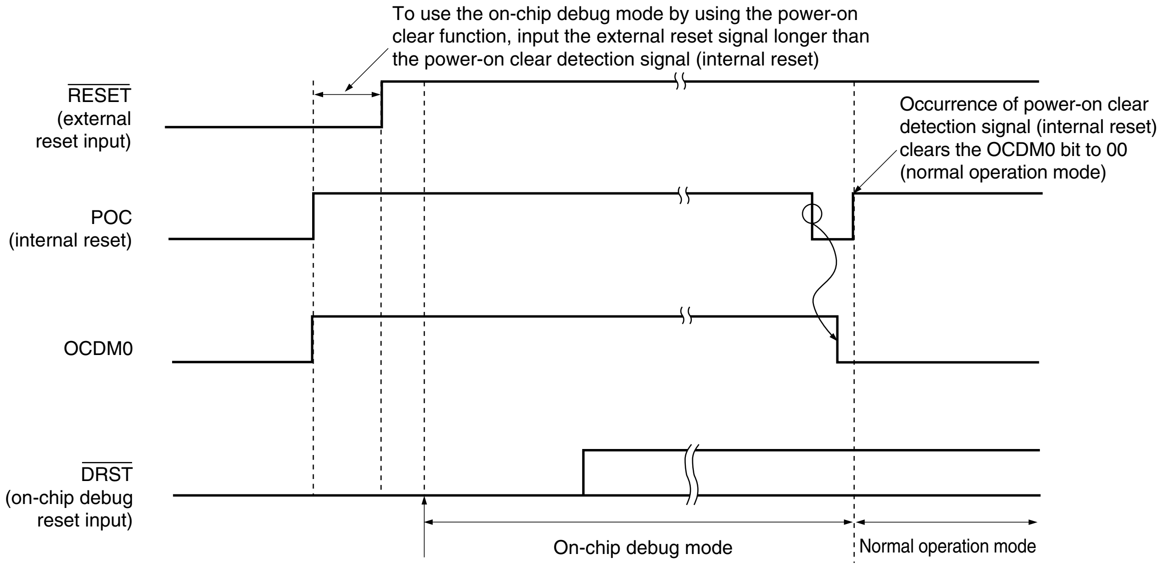


Figure 22-5: Timing Chart of Selecting On-Chip Debug Mode


To use the on-chip debug mode by using the power-on clear function, input the external reset signal longer than the power-on clear detection signal (internal reset)
RESET
(external
reset input)
Occurrence of power-on clear detection signal (internal reset) clears the OCDM0 bit to 00 (normal operation mode)
POC
(internal reset)
OCDM0
DRST
(on-chip debug
reset input)
On-chip debug mode
Normal operation mode

Caution: To use the on-chip debug function in a product with a power-on clear function, input a low level to the RESET input pin for 2,000 ms or longer after power application.

22.5 Connection to N-Wire Emulator

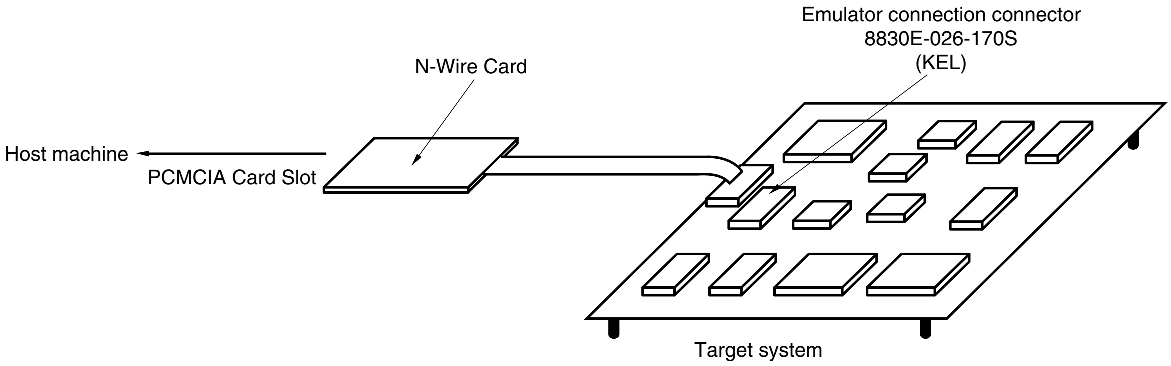
To connect the N-Wire emulator, a connector for emulator connection and a connection circuit must be mounted on the target system.
Select the KEL connector, MICTOR connector (product name: 2-767004-2, Tyco Electronics AMP K.K.), or a 20-pin general-purpose connector with a 2.54 mm pitch as the emulator connection connector. Connectors other than the KEL connector may not be supported by some emulators. Refer to the user's manual of the emulator to be used.

22.5.1 KEL connector

- Product name

  - 8830E-026-170S (KEL): Straight type
  - 8830E-026-170L (KEL): Right-angle type

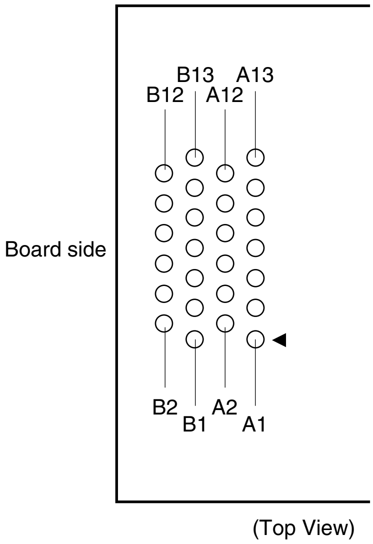
Figure 22-6: Connection to N-Wire Emulator (NEC Electronics IE-V850E1-CD-NW: N-Wire Card)



(1) Pin configuration

Figure 22-7 shows the pin configuration of the connector for emulator connection (target system side), and Table 22-2 shows the pin functions.

Figure 22-7: Pin Configuration of Connector for Emulator Connection (Target System Side)



Caution: Evaluate the dimensions of the connector when actually mounting the connector on the target board.

(2) Pin functions

The following table shows the pin functions of the connector for emulator connection (target system side). “I/O” indicates the direction viewed from the device.

Table 22-2: Pin Functions of Connector for Emulator Connection (Target System Side)

Pin No.	Pin Name	I/O	Pin Function
A1	(Reserved 1)	–	(Connect to GND)
A2	(Reserved 2)	–	(Connect to GND)
A3	(Reserved 3)	–	(Connect to GND)
A4	(Reserved 4)	–	(Connect to GND)
A5	(Reserved 5)	–	(Connect to GND)
A6	(Reserved 6)	–	(Connect to GND)
A7	DDI	Input	Data input for N-Wire interface
A8	DCK	Input	Clock input for N-Wire interface
A9	DMS	Input	Transfer mode select input for N-Wire interface
A10	DD0	Output	Data output for N-Wire interface
A11	DRST	Input	On-chip debug unit reset input
A12	RESET	Input	Reset input. (In a system that uses only POC reset and not pin reset, some emulators input an external reset signal as shown in Figure 18-5 to set the OCDM0 bit to 1.)
A13	FLMD0	Input	Control signal for flash download (flash memory versions only)
B1	GND	–	–
B2	GND	–	–
B3	GND	–	–
B4	GND	–	–
B5	GND	–	–
B6	GND	–	–
B7	GND	–	–
B8	GND	–	–
B9	GND	–	–
B10	GND	–	–
B11	(Reserved 8)	–	(Connect to GND)
B12	(Reserved 9)	–	(Connect to GND)
B13	V_{DD}	–	5 V input (for monitoring power supply to target)

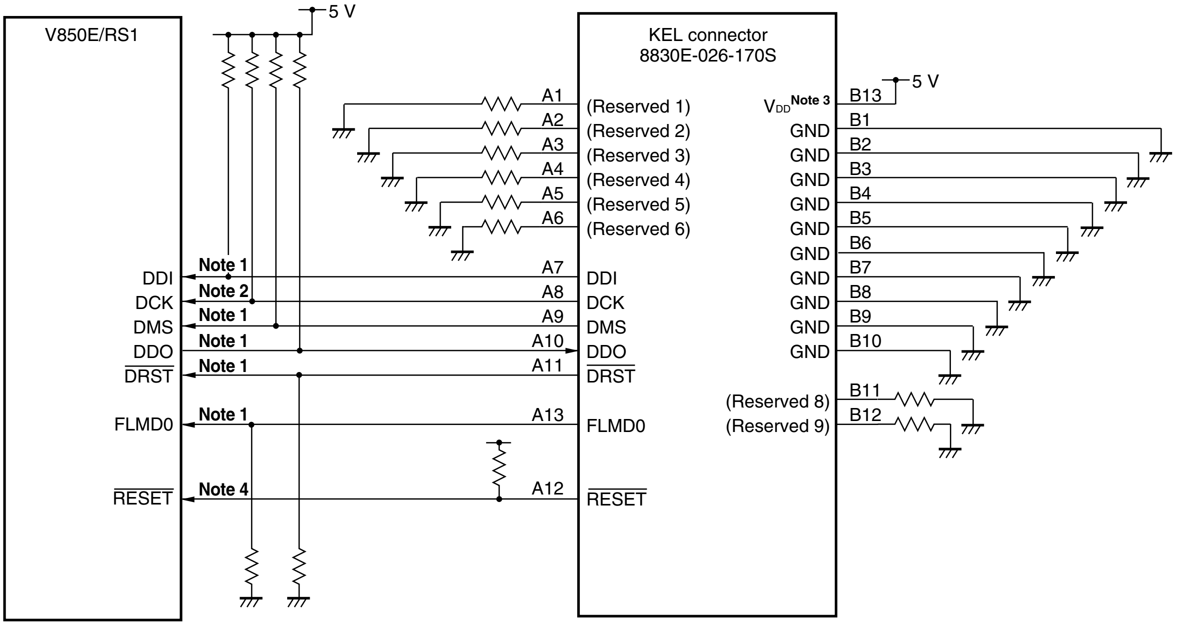
Cautions: 1. The connection of the pins not supported by the V850E/RS1 series is dependent upon the emulator to be used.

2. The pattern of the target board must satisfy the following conditions.
- **The pattern length must be 100 mm or less.**
- **The clock signal must be shielded by GND.**

(3) Example of recommended circuit

An example of the recommended circuit of the connector for emulator connection (target system side) is shown below.

Figure 22-8: Example of Recommended Emulator Connection Circuit



Notes: 1. The pattern length must be 100 mm or less.

2. Shield the DCK signal by enclosing it with GND.

3. This pin is used to detect power to the target board. Connect the voltage of the N-Wire interface to this pin.

4. In a system that uses only POC reset and not pin reset, some emulators input an external reset signal as shown in Figure 22-8 to set the OCDM0 bit to 1.

Caution: The N-Wire emulator may not support a 5 V interface and may require a level shifter. Refer to the user's manual of the emulator to be used.

22.6 Restrictions and Cautions on On-Chip Debug Function

- Do not mount a device that was used for debugging on a mass-produced product (this is because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed).
- If a reset signal (reset input from the target system or reset by an internal reset source) is input during RUN (program execution), the break function may malfunction.
- Even if reset is masked by using a mask function, the I/O buffer (port pin, etc.) is reset when a pin reset signal is input.
- With a debugger that can set software breakpoints in the internal flash memory, the breakpoints temporarily become invalid when pin reset or internal reset is effected. The breakpoints become valid again if a break such as a hardware break or forced break is executed. Until then, no software break occurs.
- The RESET signal input is masked during a break.
- The POC reset operation cannot be emulated.
- Pin reset must be input to execute on-chip debugging, because the OCDM0 bit of the OCDM register must be set to 1.
- For details, refer to 22.3 (1)”On-chip debug mode register (OCDM)” on page 771.

Caution: Caution for useful

When break command is based, and application software accesses for UARTA/CSIB/AFCAN peripheral I/O register, to restart without reset, CSIB and AFCAN that may be not correct operation.

Chapter 23 Power-On-Clear Circuit

23.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
- Compares supply voltage (V_{DD}) and detected voltage (V_{POC}), and generates internal reset signal when $V_{DD} < V_{POC}$.

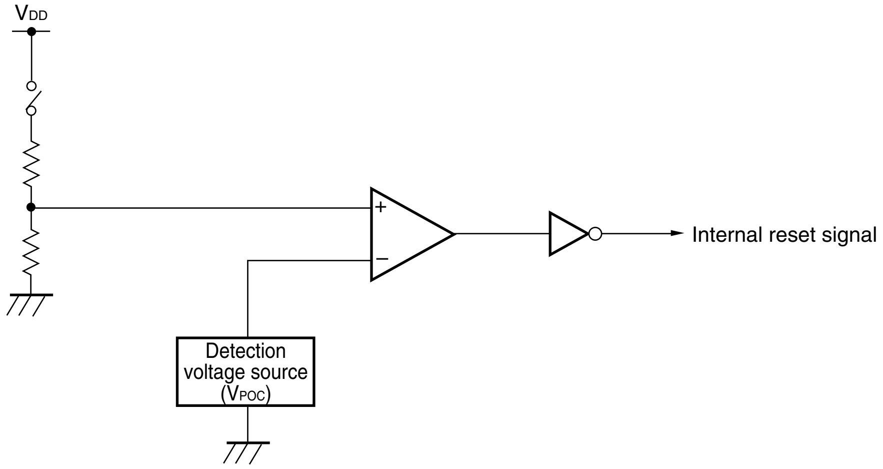
Caution: **If an internal reset signal is generated by the POC circuit, the reset source flag register (RESF) is cleared to 00H.**

Remark: This product has several hardware functions that generate an internal reset signal. When an internal reset signal is generated by the watchdog timer (WDT2RES), the low-voltage-detection (LVI) circuit, or clock monitor (CLM), a flag located in the reset control flag register (RESF) identifies the reset source. If an internal reset signal is generated by WDT2RES, LVI, or the clock monitor.RESF is not cleared to 00H but the corresponding flag is set to 1. For details of the RESF, refer to **Chapter 19 ”RESET Function” on page 735**.

23.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 23-1.

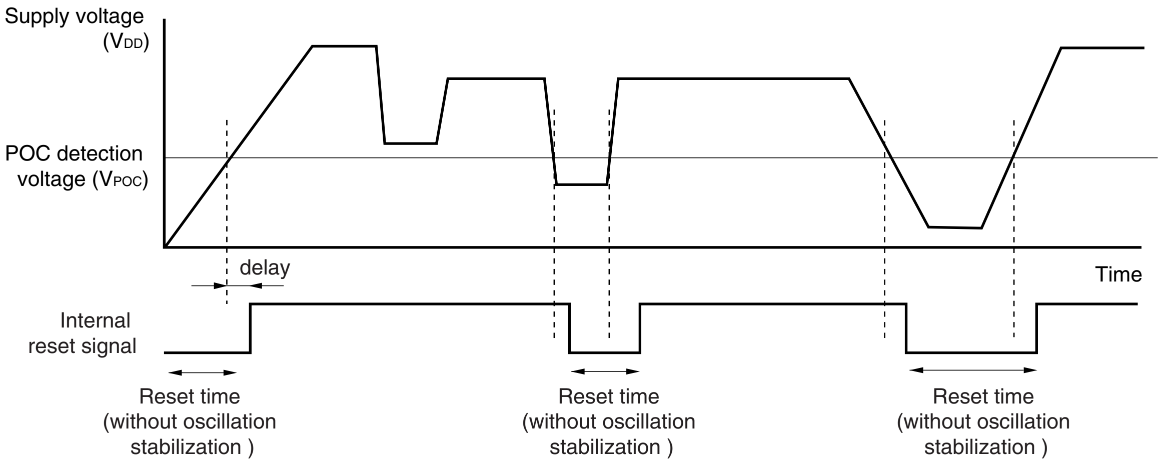
Figure 23-1: Block Diagram of Power-on-Clear Circuit



23.3 Operation of Power-on-Clear Circuit

The power-on-clear circuit compares the supply voltage (V_{DD}) and detected voltage (V_{POC}), and generates an internal reset signal when $V_{DD} < V_{POC}$.

Figure 23-2: Timing of Internal Reset Signal Generation by Power-on-Clear Circuit


Supply voltage (VDD)
POC detection voltage (VPOC)
delay
Time
Internal reset signal
Reset time (without oscillation stabilization)
Reset time (without oscillation stabilization)
Reset time (without oscillation stabilization)

Chapter 24 Low-Voltage Detector

24.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has following functions:

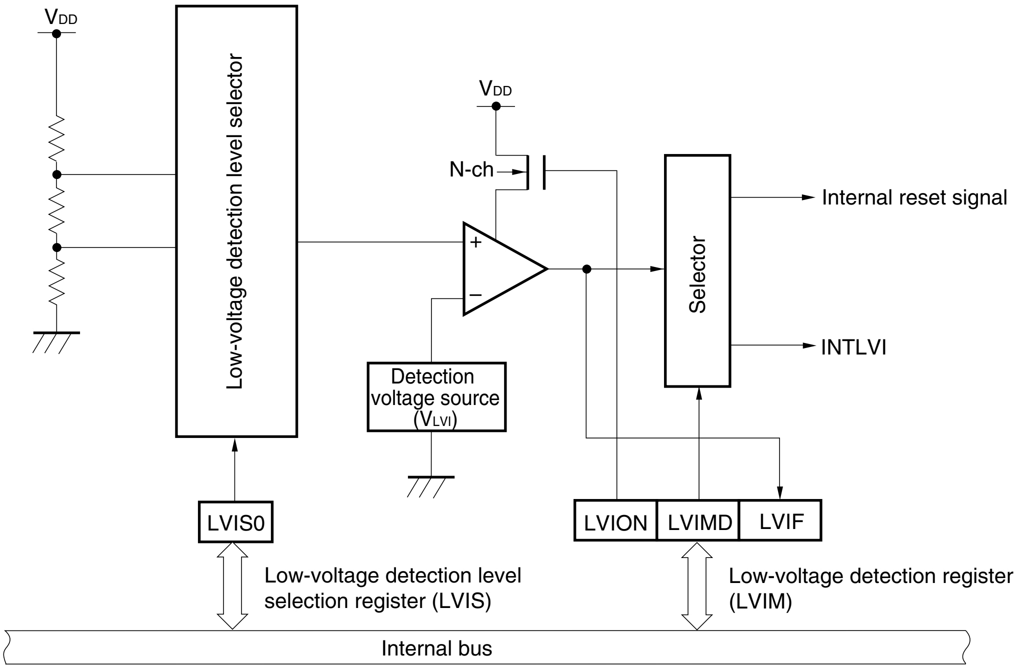
- Compares the supply voltage (V_{DD}) and detected voltage (V_{LVI}), and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- Interrupt or reset signal can be selected by software.
- Can operate in STOP mode.
- Operation can be stopped by software.

If the low-voltage detector is used to generate a reset signal, bit 0 (LVIRF) of the reset source flag register (RESF) is set to 1 when the reset signal is generated., refer to **Chapter 19 ”RESET Function” on page 739**.

24.2 Configuration of Low-Voltage Detector

Figure 23-1 shows the block diagram of the low-voltage detector.

Figure 24-1: Block Diagram of Low-Voltage Detector


VDD
Low-voltage detection level selector
VDD
N-ch
+
–
Detection voltage source (VLVI)
Selector
Internal reset signal
INTLVI
LVIS0
LVION
LVIMD
LVIF
Low-voltage detection level selection register (LVIS)
Low-voltage detection register (LVIM)
Internal bus

24.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)

(1) Low-voltage detection register (LVIM)

The LVIM register is a special register (see **3.2.3 "Special registers" on page 70**). It is used to enable or disable low-voltage detection and sets the operation mode of the low-voltage detector.
This register can be read or written 1-bit or 8-bit units.
Reset input clears this register to 00H.

Figure 24-2: Low-Voltage Detection Register (LVIM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF	FFFFF890H	00H
R/W	R/W	R	R	R	R	R	R/W	R		

LVION	Enables or disables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIMD	Low-voltage detection operation mode selection
0	Generates interrupt request signal INTLVI when supply voltage (V_{DD}) < low-voltage detection (V_{LVI})
1	Generates internal reset signal LVIRES when supply voltage (V_{DD}) < low-voltage detection (V_{LVI})

LVIF	Low-voltage detection flag
0	When supply voltage (V_{DD}) > detection voltage (V_{LVI}), or when operation is disabled
1	When supply voltage (V_{DD}) < detection voltage (V_{LVI})

Cautions:

1. **LVION and LVIMD are cleared to 0 at a reset other than an LVI reset. These are not cleared to 0 at an LVI reset.**
2. **After setting the LVION bit to 1, wait for 0.1ms (TYP) (target value) before checking the voltage using the LVIF bit.**
3. **The value of LVIF flag is output as the interrupt request signal INTLVI when LVION bit = 1 and LVIMD bit = 0.**
4. **The LVIF bit is read-only. Be sure to clear bits 2 to 6 to 0**
5. **To stop LVI, when using 8-bit manipulation instruction: Write 00H to LVIM.**

(2) Low-voltage detection level selection register (LVIS)

The LVIS register is used to select the level of low-voltage to be detected.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 24-3: Low-Voltage Detection Level Selection Register (LVIS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
LVIS	0	0	0	0	0	0	0	LVIS0	FFFFF891H	00H
R/W	R	R	R	R	R	R	R	R/W		

LVIS0	Detection level
0	V_{LVI0} (4.4 V ±0.2 V)
1	V_{LVI1} (4.2 V ±0.2 V)

(3) Internal RAM data status register (RAMS)

The RAMS register is a flag register that indicates whether the internal RAM is valid or not.
This register can be read or written in 8-bit or 1-bit units[Note 1].
Reset input[Note 2] sets this register to 01H.

Notes: 1. This register can be written only in a specific sequence (see **3.2.3 "Special registers" on page 70**).

2. Setting conditions: Detection of voltage lower than specified level
Set by instruction
Generation of reset signal by WDT2
Generation of reset signal while RAM is being accessed
Generation of reset signal by clock monitor
Clearing condition: Writing of 0 in specific sequence

Figure 24-4: Internal RAM Data Status Register (RAMS) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
RAMS	0	0	0	0	0	0	0	RAMF	FFFFF892H	01H
R/W	R	R	R	R	R	R	R	R/W		

RAMF	Internal RAM data valid/invalid
0	valid
1	invalid

(4) Peripheral emulation register 1 (PEMU1)

When an in-circuit emulator is used, the operation of the RAM retention flag (RAMF bit: bit 0 of RAMS register) can be pseudo-controlled and emulated by manipulating this register on the debugger.

This register is valid only in the emulation mode. It is invalid in the normal mode.

Figure 24-5: Peripheral Emulation Register 1 (PEMU1)

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PEMU1	0	0	0	0	0	EVARAMIN	0	0	FFFFF9FEH	00H
R/W	R	R	R	R	R	R/W	R	R		

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.

24.4 Operation of Low-Voltage Detector

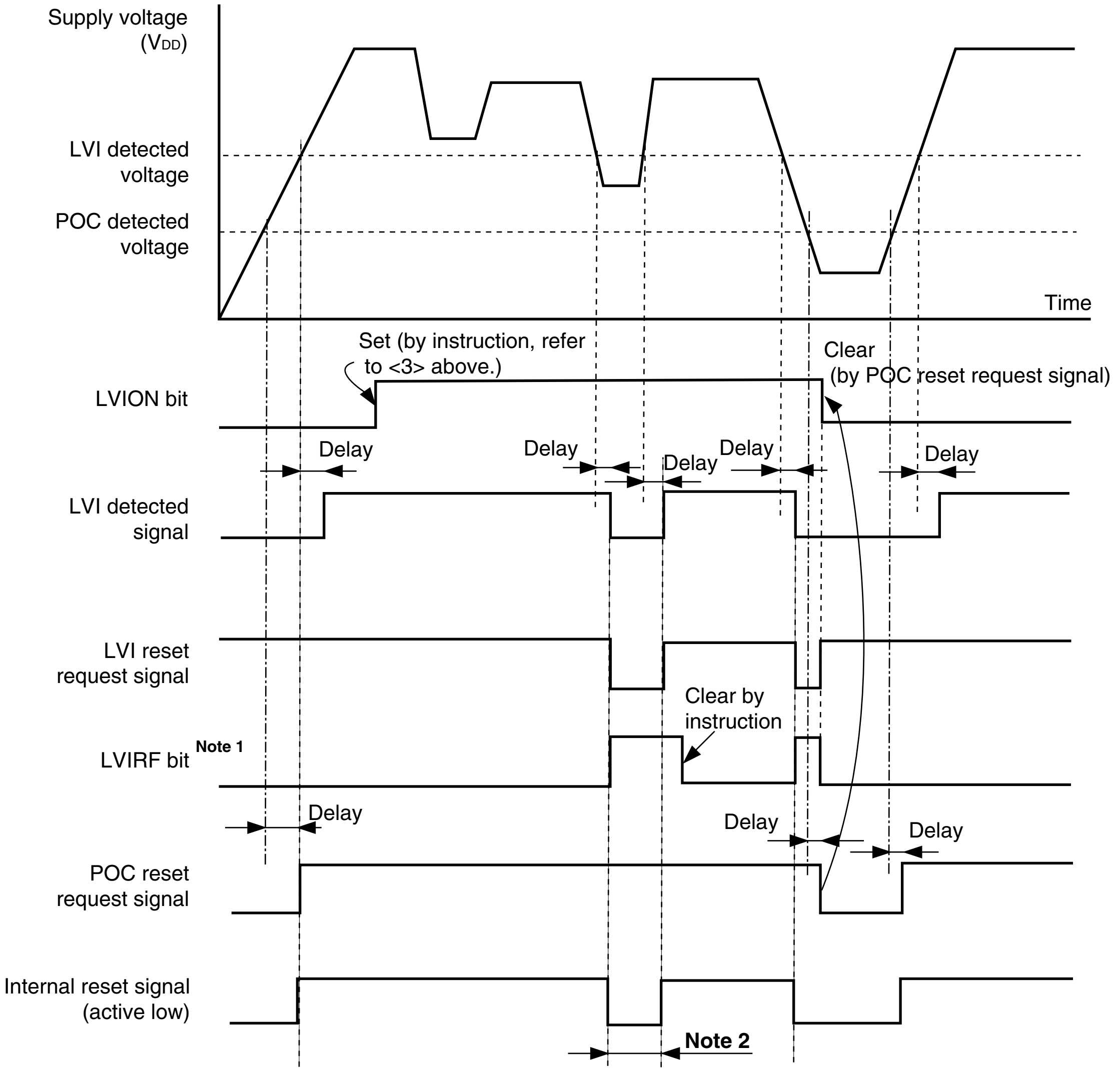
Depending on the setting of the LVIMD bit, an interrupt signal (INTLVI) or an internal reset signal is generated.
How to specify each operation is described below, together with timing charts.

24.4.1 To use for internal reset signal

<To start operation>
<1> Mask the interrupt of LVI.
<2> Select the voltage to be detected by using the LVIS0 bit.
<3> Set the LVION bit to 1 (to enable operation).
<4> Insert a wait cycle of 0.1 ms (TYP) (target value) or more by software.
<5> By using the LVIF bit, check if the supply voltage > detected voltage.
<6> Set the LVIMD bit to 1 (to generate an internal reset signal).

Caution: If LVIMD 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 24-6: Operation Timing of Low-Voltage Detector (LVIMD = 1)



Notes: 1. The LVIRF bit is bit 0 of the reset source flag register (RESF). For details of RESF, refer to **Chapter 19 "RESET Function" on page 739**.

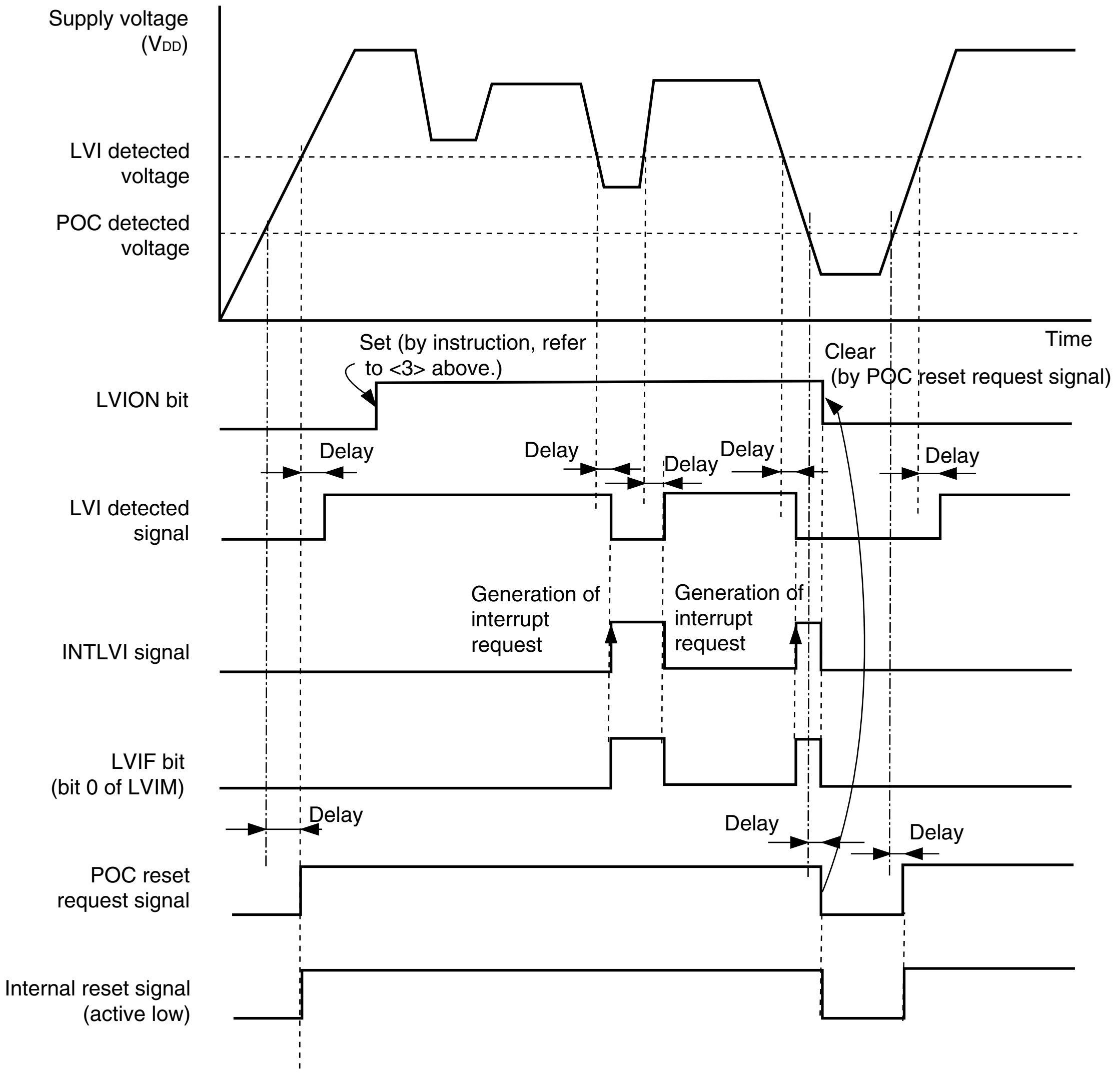
2. During the period in which the supply voltage is the set low voltage or lower, the internal reset signal is retained (internal reset state).

24.4.2 To use for interrupt

<To start operation>
<1> Mask the interrupt of LVI.
<2> Select the voltage to be detected by using the LVIS0 bit.
<3> Set the LVION bit to 1 (to enable operation).
<4> Insert a wait cycle of 0.1 ms (TYP) (target value) or more by software.
<5> By using the LVIF bit, check if the supply voltage > detected voltage.
<6> Clear the interrupt request flag of LVI.
<7> Unmask the interrupt of LVI.

<To stop operation>
Clear the LVION bit to 0.

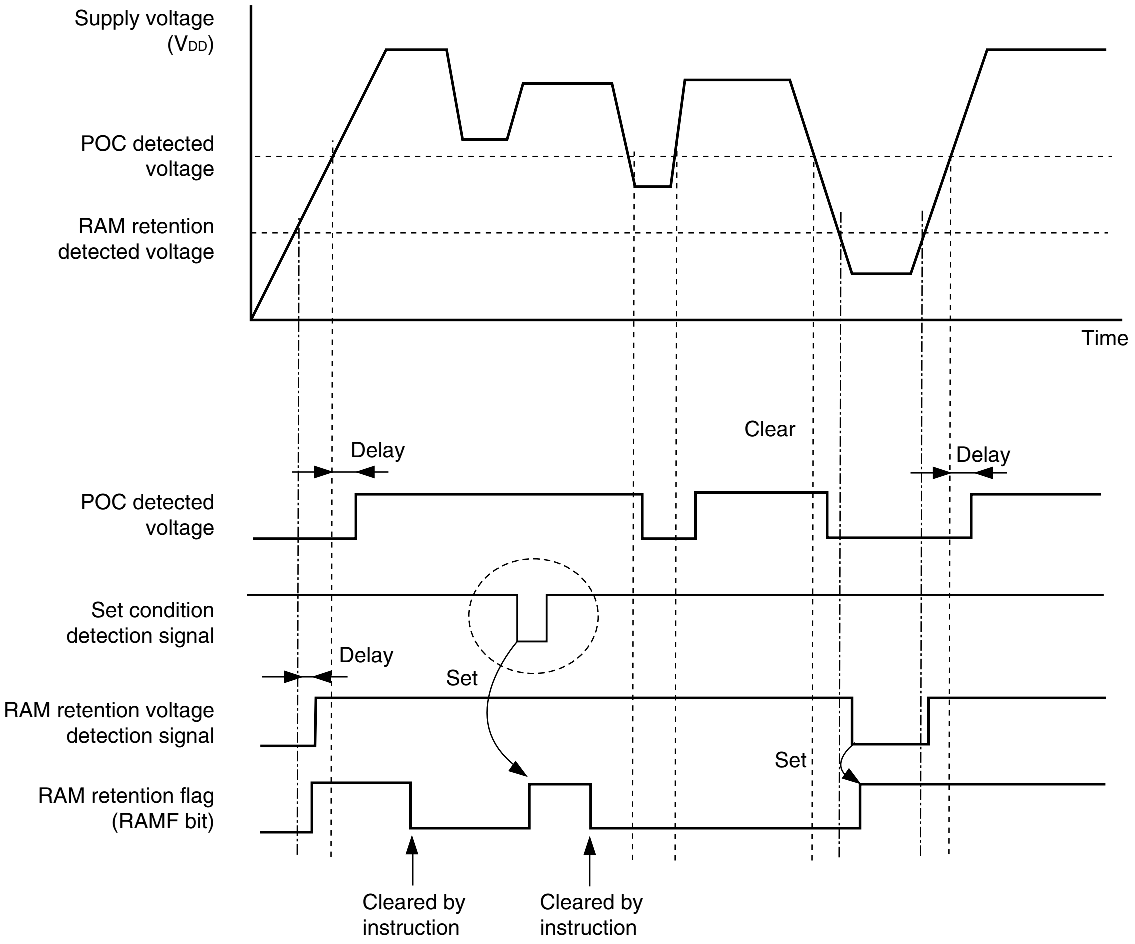
Figure 24-7: Operation Timing of Low-Voltage Detector (LVIMD = 0)



24.5 RAM Retention Voltage Detection Operation

The supply voltage and detected voltage are compared. When the supply voltage drops below the detected voltage (including on power application), the RAMF bit is set.

Figure 24-8: Operation Timing of RAM Retention Voltage Detection Function



Chapter 25 Clock Monitor

25.1 Functions of Clock Monitor

The clock monitor samples the main clock (X1 input clock) by using the on-chip Ring-OSC, 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.
The clock monitor automatically stops under the following conditions.

- While oscillation stabilization time is being counted after software STOP mode is released
- When the main clock (X1 input clock) is stopped
- When the sampling clock is stopped (Ring-OSC)
- When the CPU operates with Ring-OSC

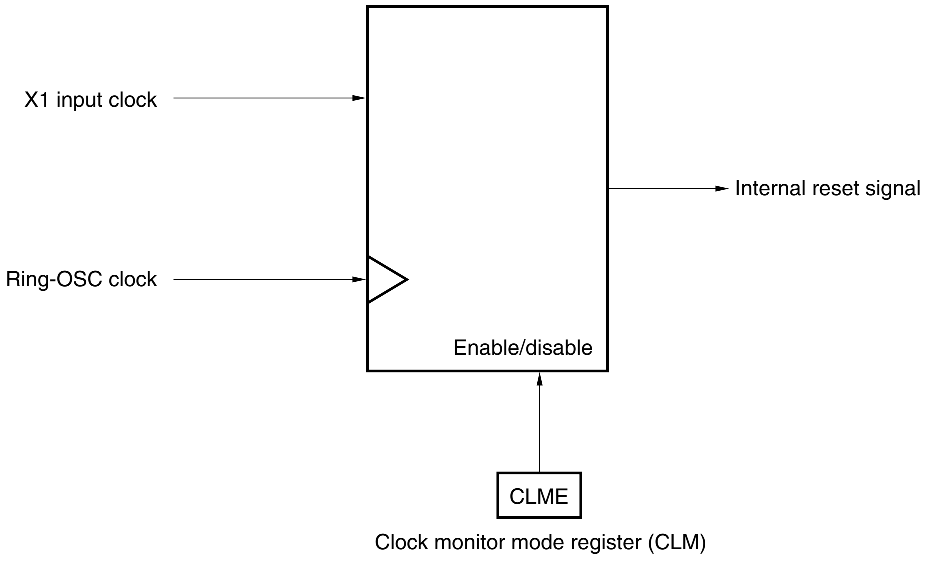
25.2 Configuration of Clock Monitor

Clock monitor consists of the following hardware.

Table 25-1: Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 25-1: Block Diagram of Clock Monitor



25.3 Register Controlling Clock Monitor

The Clock monitor is controlled by the clock monitor mode register (CLM).

(1) Clock monitor mode register (CLM)

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.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 25-2: Clock Monitor Mode Register (CLM) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
CLM	0	0	0	0	0	0	0	CLME	F870H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

CLME	Enables/disables clock monitor operation
0	Disables clock monitor operation
1	Enables 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. If reset is occurred for clock monitor, CLME bit is clear (0), and CLMRF bit of RESF register is set (1) (see 19.2 ”Registers to Check Reset Source” on page 740).

25.4 Operation of Clock Monitor

This section explains the functions of the clock monitor. The start and stop conditions are as follows.

<Start condition>
Enabling operation by setting bit 0 (CLME) of the clock monitor mode register to 1

<Stop condition>

- While oscillation stabilization time is being counted after software STOP mode is released
- When the main clock is stopped
- When the sampling clock is stopped (Ring-OSC)
- When the CPU operates using Ring-OSC

Table 25-2: Operation Status of Clock Monitor (When CLM.CLME Bit = 1, During Ring-OSC Operation) (CKSEL Connected to Ring-OSC)

CPU operation clock	Operation Mode	Status of Main Clock	Status of Ring-OSC Clock	Status of Clock Monitor
Main clock	HALT mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
	IDLE mode			
	STOP mode	Stops		Stops
Ring-OSC clock	-	Stops	Oscillates[Note 1]	Stops
Reset	-	Stops	Stops	Stops

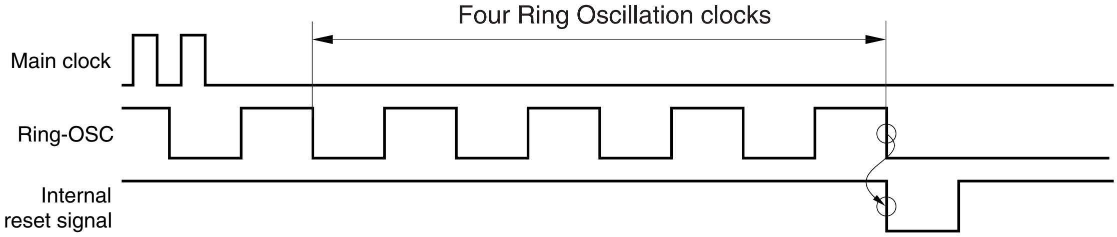
Notes: 1. Ring-OSC can be stopped by setting the RSTOP bit of the RCM register to 1

2. The clock monitor is stopped while Ring-OSC 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 25-3.

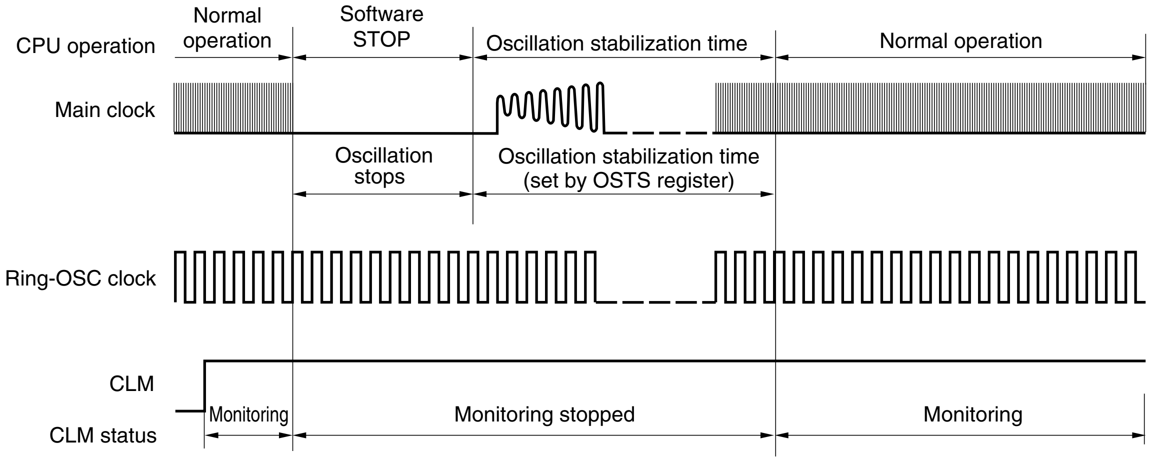
Figure 25-3: When Oscillation of Main Clock Is Stopped



(2) Operation in software STOP mode or after software STOP mode is released

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

Figure 25-4: Operation in Software STOP Mode or After Software STOP Mode Is Released



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the X1 input clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

(3) Operation while CPU is operating on Ring-OSC clock (CCLSF bit of CCLS register = 1)

The monitor operation is not started when the CCLSF bit is 1, even if the CLME bit is set to 1.

Chapter 26 CRC Function

26.1 Functions

- CRC operation circuit for detection of data block errors
- Generation of 16-bit CRC code using a CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) generation polynomial for blocks of data of any length in 8-bit units
- CRC code is set to the CRC data register each time 1-byte data is transferred to the CRCIN register, after the initial value is set to the CRCD register.

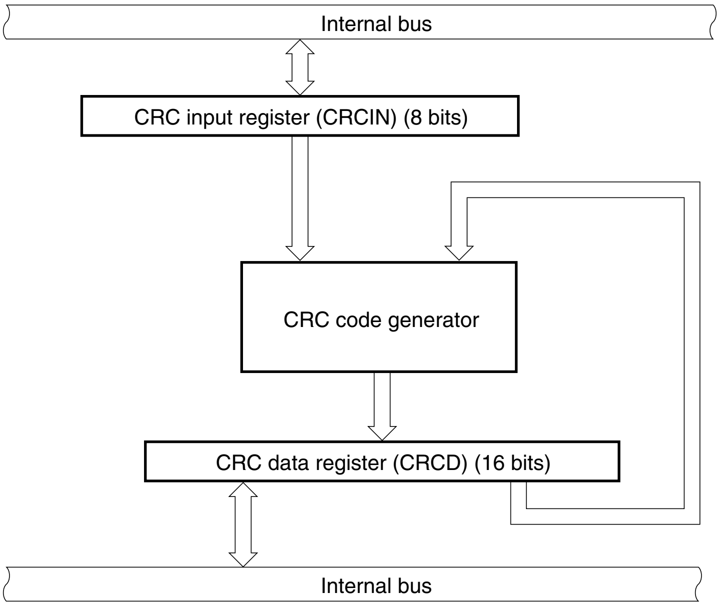
26.2 Configuration

The CRC function includes the following hardware.

Table 26-1: CRC Configuration

Item	Configuration
Control registers	CRC input register (CRCIN) CRC data register (CRCD)

Figure 26-1: Block Diagram of CRC Register


Internal bus
CRC input register (CRCIN) (8 bits)
CRC code generator
CRC data register (CRCD) (16 bits)
Internal bus

26.3 Registers

(1) CRC input register (CRCIN)

The CRCIN register is an 8-bit register for setting data.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 26-2: CRC Input Register (CRCIN) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
CRCIN									FFFFF310H	00H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

(2) CRC data register (CRCD)

The CRCD register is a 16-bit register that stores the CRC-CCITT operation results.
This register can be read or written in 16-bit units.
Reset input clears this register to 0000H.

Cautions: 1. Following write to the CRCD register, do not write the CRC calculator output to the CRCD register during the first write access to the CRCIN register. (Do not load the operation result.)

2. Through the operation of the CRCIN register, the CRC operation results are saved to the CRCD register. Therefore, to write to the CRCD register via the internal bus and then read the written value, read the CRCD register before writing to CRCIN.

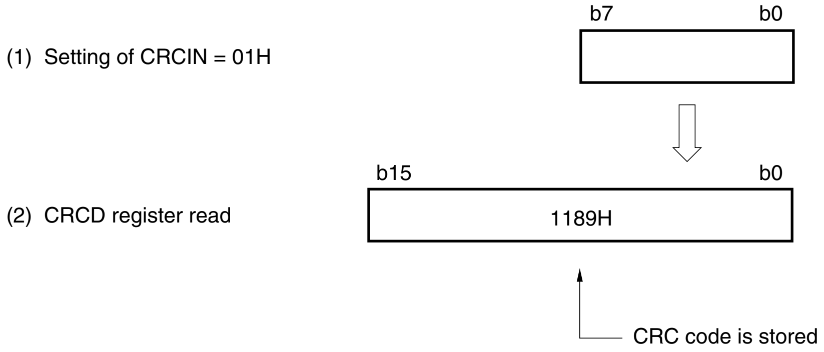
Figure 26-3: CRC Data Register (CRCD) Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset
CRCD									FFFFF312H	0000H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

26.4 Operation

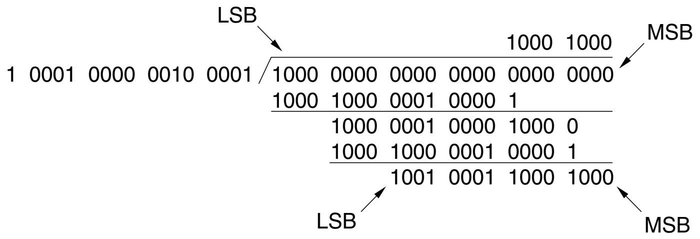
An example of the CRC operation circuit is shown below.

Figure 26-4: CRC Operation Circuit Operation Example (LSB First)



The code when 01H is sent LSB first is (1000 0000). Therefore, the CRC code from generation polynomial $X^{16} + X^{12} + X^5 + 1$ becomes the remainder when (1000 0000) X^{16} is divided by (1 0001 0000 0010 0001) using the modulo-2 operation formula.
The modulo-2 operation is performed based on the following formula.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
− 1 = 1



Therefore, the CRC code becomes

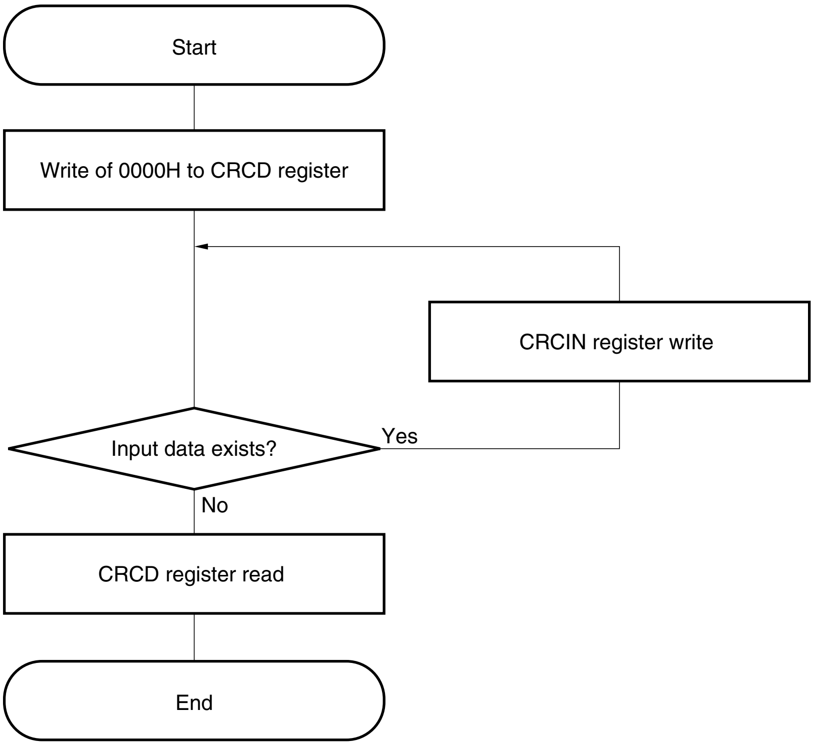
$$\overbrace{1001}^{9}\ \overbrace{0001}^{8}\ \overbrace{1000}^{1}\ \overbrace{1000}^{1}$$

Since LSB first is used, this corresponds to 1189H in hexadecimal notation.

26.5 Usage Method

How to use the CRC operation circuit is described below.

Figure 26-5: CRC Operation Flow

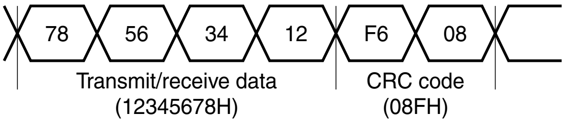


[Basic usage method]

<1> Write 0000H to the CRCD register.
<2> Write the required quantity of data to the CRCIN register.
<3> Read the CRCD register.

Communication errors can easily be detected if the CRC code is transmitted/received along with transmit/receive data when transmitting/receiving data consisting of several bytes.
The following is an illustration using the transmission of 12345678H (0001 0010 0011 0100 0101 0110 0111 1000B) LSB-first as an example.

Figure 26-6: CRC Transmission Example



- Setting procedure on transmitting side

<1> Write the initial value 0000H to the CRCD register.
<2> Write the 1 byte of data to be transmitted first to the transmit buffer register. (At this time, also write the same data to the CRCIN register.)
<3> When transmitting several bytes of data, write the same data to the CRCIN register each time transmit data is written to the transmit buffer register.
<4> After all the data has been transmitted, write the contents of the CRCD register (CRC code) to the transmit buffer register and transmit them. (Since this is LSB first, transmit the data starting from the lower bytes, then the higher bytes.)

- Setting procedure on receiving side

<1> Write the initial value 0000H to the CRCD register.
<2> When reception of the first 1 byte of data is complete, write that receive data to the CRCIN register.
<3> If receiving several bytes of data, write the receive data to the CRCIN register upon every reception completion. (In the case of normal reception, when all the receive data has been written to the CRCIN register, the contents of the CRCD register on the receiving side and the contents of the CRCD register on the transmitting side are the same.)
<4> Next, the CRC code is transmitted from the transmitting side, so write this data to the CRCIN register similarly to receive data.
<5> When reception of all the data, including the CRC code, has been completed, reception was normal if the contents of the CRCD register are 0000H. If the contents of the CRCD register are other than 0000H, this indicates a communication error, so transmit a resent request to the transmitting side.

[MEMO]

Chapter 27 Electrical Specification

27.1 General

The following electrical specification characterizes the V850E/RS1. Unless noted otherwise, the data provided applies to all versions device (µPD70F3402, µPD70F3403, µPD70F3403A). Data that is unique to any of the versions will be indicated in the header of the subsection.

27.2 Absolute Maximum Ratings

(T_A = +25°C)

Parameter	Symbol	Condition	Rating	Unit
Power supply voltage, Analog reference input voltage	V_{DD}	V_{DD0}, V_{DD1}, BV_{DD} pins $V_{DD0} = V_{DD1} = BV_{DD}$	-0.5 to +6.5	V
	AV_{REF}	AV_{REF0} pin	-0.5 to +6.5	V
	V_{SS}	V_{SS0}, $V_{SS1,}$ BV_{SS} pins $V_{SS0} = V_{SS1} = BV_{SS}$	-0.5 to +0.5	V
	AV_{SS}	AV_{SS} pin	-0.5 to +0.5	V
Input voltage	V_{I1}	P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P913	-0.5 to V_{DD} +0.5 ≤ 6.5 V	V
Input voltage	V_{I2}	P914-P915, PCS0-PCS1, PCM0-PCM3, PCT0, PCT1, PCT4, PCT6, PDL0-PDL13	-0.5 to BV_{DD} +0.5 ≤ 6.5 V	V
Input voltage	V_{IAN}	P70-P715	-0.5 to AV_{REF} +0.5 ≤ 6.5 V	V
Clock Input Voltage	V_K	X1, X2	-0.5 to V_{RO} +0.5 **Note**	V
Low-level output current	I_{OL}	1 pin	4.0	mA
		Total for all pins	50	mA
High-level output current	I_{OH}	1 pin	-4.0	mA
		Total for all pins	-50	mA
Storage temperature	T_{STG}		-40 to +125	°C

Note: V_{RO} is the output of the on-chip voltage regulator.

Cautions: 1. Be sure to avoid direct connections among the IC device output (or I/O) pins and between V_{DD} or GND.

2. If the absolute maximum rating for any of the above parameters is exceeded even momentarily, it may adversely affect the quality of this product. In other words, these absolute maximum ratings have been set to prevent physical damage to the product. Do not use the product in such a way as to exceed any of these ratings. The ratings and conditions shown below for DC characteristics and AC characteristics are within range for normal operation and quality assurance.

27.2.1 Absolute maximum ratings (μPD70F3402)

Parameter	Symbol	Condition	Rating	Unit
Operating temperature	T_A	Normal operation mode	-40 to +110	°C
		Flash programming mode	-40 to +85	°C

27.2.2 Absolute maximum ratings (μPD70F3403, μPD70F3403A)

Parameter	Symbol	Condition	Rating	Unit
Operating temperature	T_A	Normal operation mode	-40 to +85	°C
		Flash programming mode	-40 to +85	°C

27.3 Capacitance

(T_A = 25°C, V_{DD} = V_{SS} = 0 V)

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz All pins are 0 V except for testing pin				15	pF
I/O capacitance	C_{IO}					15	pF
Output capacitance	C_O					15	pF
Analog input capacitance	C_{AI}		ANI15-ANI0			20	pF

27.4 Recommended Operating Conditions

27.4.1 Recommended operating conditions (μPD70F3402)

Operation Mode	International Operating Clock Frequency	Operating Ambient Temperature (T_A)	Digital Power Supply (V_{DD0}, V_{DD1}, BV_{DD})	Analog Power Supply (AV_{REF0})
Normal Mode	24 MHz, 32 MHz REGC1: Capacity=4.7 μF REGC0: Capacity=1.0 μF	-40°C to +110°C	4.0 V – 5.5 V	4.5 V – 5.5 V

27.4.2 Recommended operating conditions (μPD70F3403, μPD70F3403A)

Operation Mode	International Operating Clock Frequency	Operating Ambient Temperature (T_A)	Digital Power Supply (V_{DD0}, V_{DD1}, BV_{DD})	Analog Power Supply (AV_{REF0})
Normal Mode	24 MHz, 32 MHz, 40 MHz REGC1: Capacity=4.7 μF REGC0: Capacity=1.0 μF	-40°C to +85°C	4.0 V – 5.5 V	4.5 V – 5.5 V

27.5 Oscillator Characteristics

27.5.1 General condition

$-40°C \le T_A \le +110°C$ (μPD70F3402)
$-40°C \le T_A \le +85°C$ (μPD70F3403, μPD70F3403A)
$V_{SS0}=V_{SS1}=AV_{SS}=BV_{SS}=0$ V
$4.0 V \le V_{DD0} \le V_{DD1} \le 5.5$ V

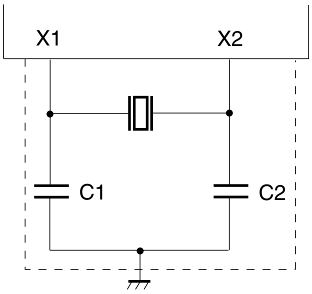
27.5.2 Oscillator timing and recommended oscillator connection

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X			4	-	8	MHz
Oscillation stabilization time	T_{OST}		After $\overline{\text{RESET}}$	-	**Note 1**	-	ms
			After STOP mode	0.5	**Note 2**	-	ms

Notes: 1. The on-chip oscillator starts its operation after the operating voltage is supplied to the system independent of the state of the $\overline{\text{RESET}}$ signal. After release of the $\overline{\text{RESET}}$ signal, the system applies a delay of 2^{13}/fx before starting its operation. If this delay is not sufficient to ensure proper and stable oscillation of an external resonator, the user needs to adjust the active period of the $\overline{\text{RESET}}$ signal after applying the operating voltage to meet the stabilization timing of the external resonator. Please contact the resonator manufacturer for details.

2. The timing depends on the setting of the oscillation stabilization time (OSTS register).

Figure 27-1: Recommended Oscillator Connection (Ceramic or Crystal Resonator)



Cautions: 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Don’t write any other signal lines in the area indicated by broken lines.

3. For the resonator selection and oscillator constant, customers are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

27.6 Voltage Regulator Characteristics

27.6.1 General condition

-40°C ≤ T_A ≤ +110°C (µPD70F3402)
-40°C ≤ T_A ≤ +85°C (µPD70F3403, µPD70F3403A)
V_{SS0}=V_{SS1}=AV_{SS}=BV_{SS}=0 V

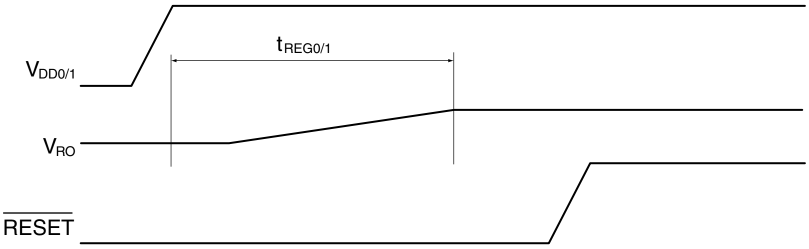
27.6.2 Regulator 0

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage stabilization time	t_{REG0}	After V_{DD} reaches 4.0 V, C= 1.0 µF	-	-	1	ms

27.6.3 Regulator 1

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage stabilization time	t_{REG1}	After V_{DD} reaches 4.0 V, C= 4.7 µF	-	-	1	ms

Figure 27-2: Voltage Regulator Startup Timing



Notes: 1. Make sure that $\overline{\text{RESET}}$ = V_{SS} = 0 V when starting $V_{DD0/1}$.

2. The on-chip POC function generates a reset signal during the regulator stabilization time (no external control of the $\overline{\text{RESET}}$ signal pin required)

27.7 DC Characteristics

27.7.1 General condition

-40°C ≤ T_A ≤ +110°C (μPD70F3402)
-40°C ≤ T_A ≤ +85°C (μPD70F3403, μPD70F3403A)
4.0 V ≤ V_{DD0}=V_{DD1}=BV_{DD} ≤ 5.5 V
4.5 V ≤ AV_{REF0} ≤ 5.5 V
V_{SS0}=V_{SS1}=AV_{SS}=BV_{SS}=0 V

27.7.2 DC input/output level

(1/2)

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High-level input voltage Note 1	V_{IH1}	P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P913, FLMD0, NMI, $\overline{\text{RESET}}$		0.7 V_{DD1}	-	V_{DD1}	V
	V_{IH2}	P914-P915		0.7 BV_{DD}	-	BV_{DD}	V
	V_{IH3}	PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6 PDL0-PDL13		2.2		BV_{DD}	V
	V_{IH4}	P70-P715		0.7 AV_{REF0}	-	AV_{REF0}	V
Low-level input voltage Note 1	V_{IL1}	P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P913, FLMD0, NMI, $\overline{\text{RESET}}$		V_{SS1}	-	0.3 V_{DD1}	V
	V_{IL2}	P914-P915		BV_{SS}		0.3 BV_{DD}	V
	V_{IL3}	PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6 PDL0-PDL13		BV_{SS}		0.8	V
	V_{IL4}	P70-P715		AV_{SS}	-	0.3 AV_{REF0}	V
High-level output voltage	V_{OH1}	P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P913	I_{OH} = -1.0 mA	V_{DD1} -1.0	-	-	V
			I_{OH} = -0.1 mA	V_{DD1} -0.5	-	-	V
	V_{OH2}	P914-P915, PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6 PDL0-PDL13	I_{OH} = -1.0 mA	BV_{DD} -1.0	-	-	V
			I_{OH} = -0.1 mA	BV_{DD} -0.5	-	-	V
	V_{OH3}	P70-P715	I_{OH} = -1.0 mA	AV_{REF0} -1.0	-	-	V
			I_{OH} =-0.1 mA	AV_{REF0} -0.5	-	-	V
Low-level output voltage	V_{OL1}	P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P913	I_{OL} = 1.0 mA	-	-	1.0	V
			I_{OL} = 0.1 mA	-	-	0.5	V
	V_{OL2}	P914-P915, PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6 PDL0-PDL13	I_{OL} = 1.0 mA	-	-	1.0	V
			I_{OL} = 0.1 mA	-	-	0.5	V
	V_{OL3}	P70-P715	I_{OL} = 1.0 mA	-	-	1.0	V
			I_{OL} = 0.1 mA	-	-	0.5	V
High-level input leakage current	I_{LIH}	V_I = V_{DD}	Analog pins	-	-	1	μA
			Other pins	-	-	5	μA
Low-level input leakage current	I_{LIL}	V_I = 0 V	Analog pins	-	-	-1	μA
			Other pins	-	-	-5	μA

(2/2)

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High-level output leakage current	I_{LOH}	$V_O = V_{DD}$	Analog pins	-	-	1	µA
			Other pins	-	-	5	µA
Low-level output leakage current	I_{LOL}	$V_O = 0$ V	Analog pins	-	-	-1	µA
			Other pins	-	-	-5	µA
Pull up resistance	R_{LU}	**Note 2**		10	30	100	kΩ
Pull down resistance	R_{LD}	**Note 3**		10	30	100	kΩ

Notes: 1. Schmitt-Trigger type inputs are pins P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P915, FLMD0, NMI and $\overline{\text{RESET}}$
TTL type inputs are DCK, DDI, DMS, $\overline{\text{DRST}}$, PCM0-PCM3, PCS0-PCS1, PCT0/1/4/6, PDL0-PDL13.
Analog type inputs are P70-P715

2. Pull-up resistors are available for P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P915, PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6.

3. Pull-down resistors are available for P00-P06, P10-P12, P30-P38, P40-P42, P50-P55, P90-P99, P911-P915, PCS0-PCS1, PCM0-PCM3, PCT0/1/4/6.

Remark: Typical values are reference values for T_A = 25°C and V_{DD0}=V_{DD1}=BV_{DD}=5.0 V.

27.7.3 DC power supply current (µPD70F3402)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Normal operation	I_{DD1}	PLL mode, f_{CPU} = 32 MHz All functions are operating	-	60	80	mA
HALT mode	I_{DD2}	PLL mode, f_{CPU} = 32 MHz All functions are operating	-	27	55	mA
IDLE1 mode	I_{DD3}	PLL is on	-	9	18	mA
		PLL is off	-	6	12	mA
IDLE2 mode	I_{DD4}		-	6	12	mA
STOP mode	I_{DD5}		-	30	600	µA

Remarks: 1. Typical values are reference values for T_A = 25°C and V_{DD0}=V_{DD1}=BV_{DD}= 5.0 V.

2. The supply current does not include AV_{REF0}

27.7.4 DC power supply current (µPD70F3403, µPD70F3403A)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Normal operation	I_{DD1}	PLL mode, f_{CPU} = 40 MHz All functions are operating Note	-	75	100	mA
		PLL mode, f_{CPU} = 32 MHz All functions are operating Note	-	60	80	mA
HALT mode	I_{DD2}	PLL mode, f_{CPU} = 40 MHz All functions are operating Note	-	33	70	mA
		PLL mode, f_{CPU} = 32 MHz All functions are operating Note	-	27	55	mA
IDLE1 mode	I_{DD3}	PLL is on	-	9	18	mA
		PLL is off	-	6	12	mA
IDLE2 mode	I_{DD4}		-	6	12	mA
STOP mode	I_{DD5}		-	30	600	µA

Note: All functions are operating except the external bus interface.

Remarks: 1. Typical values are reference values for T_A = 25°C and V_{DD0}=V_{DD1}=BV_{DD}= 5.0 V.

2. The supply current does not include AV_{REF0}

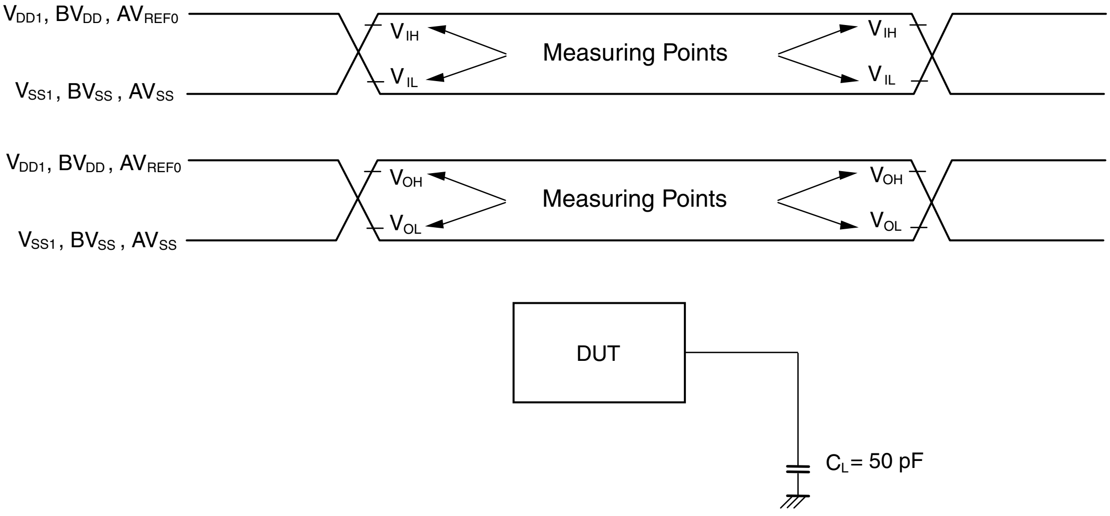
27.8 AC Characteristics

27.8.1 General condition

-40°C ≤ T_A ≤ +110°C (µPD70F3402)
-40°C ≤ T_A ≤ +85°C (µPD70F3403, µPD70F3403A)
4.0 V ≤ V_{DD0}=V_{DD1}=BV_{DD} ≤ 5.5 V
4.5 V ≤ AV_{REF0} ≤ 5.5 V
V_{SS0}=V_{SS1}=AV_{SS}=BV_{SS}=0 V
C_L=50 pF

27.8.2 AC test input waveform

Figure 27-3: AC Test Conditions



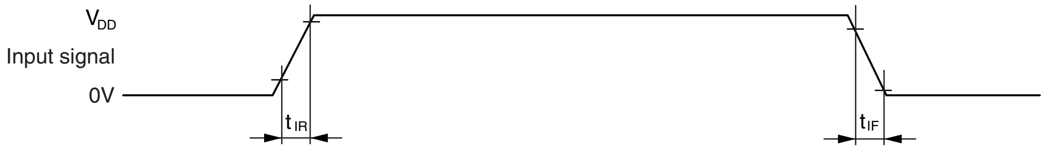
Remark: DUT: Device Under Testing

Caution: In cases where the load capacitance is greater than 50 pF due to the circuit configuration, insert a buffer or other element to reduce the device's load capacitance to below 50 pF.

27.8.3 Input waveform

Parameter	Symbol	Condition	MIN.	MAX.	Unit
Input rising time	t_{IR}		-	20	ns
Input falling time	t_{IF}		-	20	ns

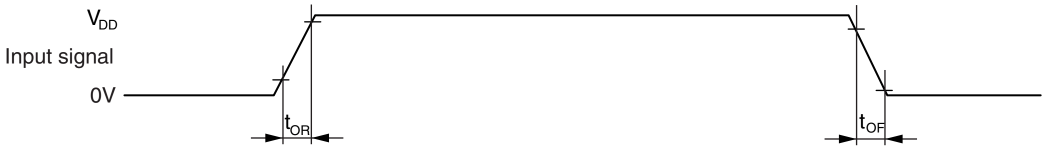
Figure 27-4: Input Rise and Fall Time



27.8.4 Output waveform

Parameter	Symbol	Condition	MIN.	MAX.	Unit
Output rising time	t_{OR}		-	10	ns
Output falling time	t_{OF}		-	10	ns

Figure 27-5: Output Rise and Fall Time



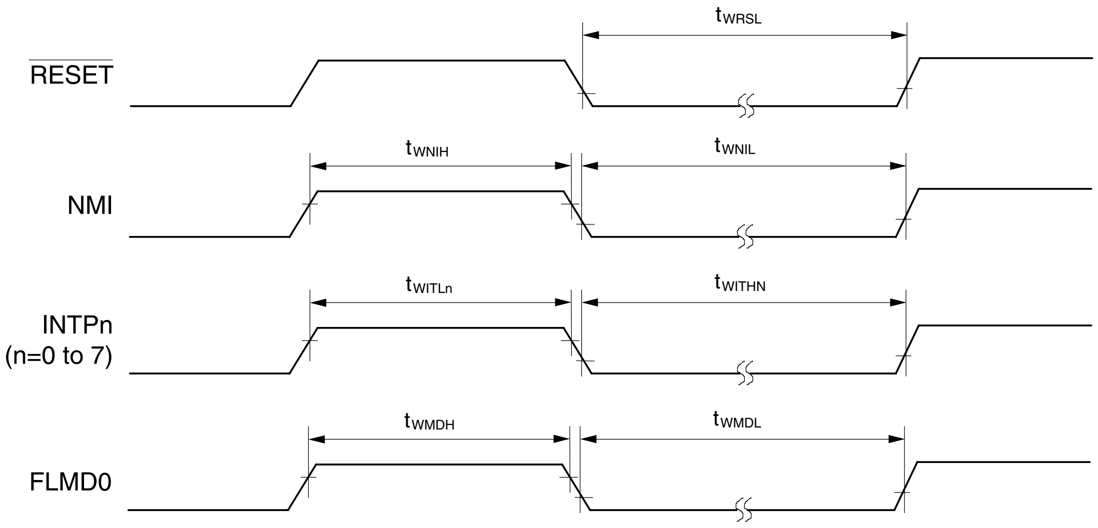
27.8.5 $\overline{\text{RESET}}$, NMI, Interrupt and FLMD0 timing

Parameter	Symbol	Condition	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low level width	t_{WRSL}	When power supply is ON	500 + t_{REG} + t_{OST}	-	ns
		When STOP mode is released	500 + t_{OST}	-	ns
		Other than when power supply is ON nor STOP mode has been released	500	-	ns
NMI high level width	t_{WNIH}	Analog filter	500	-	ns
NMI low level width	t_{WNIL}	Analog filter	500	-	ns
INTPn high level width	t_{WITLn}	Analog filter (n=0 to 7)	500	-	ns
		Digital filter (n=3)	Note	-	ns
INTPn low level width	t_{WITHn}	Analog filter (n=0 to 7)	500	-	ns
		Digital filter (n=3)	Note	-	ns
FLMD0 high level width	t_{WMDH}		500	-	ns
FLMD0 low level width	t_{WMDL}		500	-	ns

Note: 2 t_{SAMP} + 20 or 3 t_{SAMP} + 20 (t_{SAMP} is the noise reject sampling clock)

Remark: t_{REG}: Regulator output voltage stabilization time
t_{OST}: Oscillation stabilization time

Figure 27-6: RESET, Interrupt, NMI and FLMD0 Timing



27.8.6 Timer P and Timer Q input/output

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TIPmn (m=0-3, n=0-1) TIQmn (m=0-1, n=0-3) Input high level width	t_{TIH}		**Note**		ns
TOPmn (m=0-3, n=0-1) TOQmn (m=0-1, n=0-3) Input low level width	t_{TIL}		**Note**		ns

Note: $2t_{SAMP} + 20$ or $3t_{SAMP} + 20$ (t_{SAMP} is the noise reject sampling clock)

27.8.7 Bus interface timing (μPD70F3403, μPD70F3403A only)

CLKOUT Asynchronous (4.5 V ≤ V_{DD0}=V_{DD1}=BV_{DD} ≤ 5.5 V)

Parameter	Symbol		Test Conditions	MIN.	MAX.	Unit
Address setting time (to ASTB↓)	<14>	T_{SAST}		0.5T-10	-	ns
Address maintenance time (to ASTB↓)	<15>	T_{HSTA}		0.5T-10	-	ns
$\overline{\text{RD}}$↓ → Address float delay time	<16>	T_{FRDA}		-	0	ns
Address → Data input setting time	<17>	T_{SAID}		-	(2+n)T-25	ns
$\overline{\text{RD}}$↓ → Data input setting time	<18>	T_{SRDID}		-	(1+n)T-20	ns
ASTB↓ → $\overline{\text{RD}}$, $\overline{\text{WRn}}$↓ delay time	<19>	$T_{DSTRDWR}$		0.5T-10	-	ns
Data input hold time (to $\overline{\text{RD}}$↑)	<20>	T_{HRDID}		0	-	ns
$\overline{\text{RD}}$↑ → Address output time	<21>	T_{DRDA}		(1+i)T-10	-	ns
$\overline{\text{RD}}$, $\overline{\text{WRn}}$↑ → ASTB↑ delay time	<22>	$T_{DRDWRST}$		0.5T-10	-	ns
$\overline{\text{RD}}$↑ → ASTB↓ delay time	<23>	T_{DRDST}		(1.5+i)T -10	-	ns
$\overline{\text{RD}}$, $\overline{\text{WRn}}$ low level width	<24>	T_{WRDWRL}		(1+n)T-15	-	ns
ASTB high level width	<25>	T_{WSTH}		T-10	-	ns
$\overline{\text{WRn}}$↓ → Data output time	<26>	T_{DWROD}		-	10	ns
Data output setting time (to $\overline{\text{WRn}}$↑)	<27>	T_{SODWR}		(1+n)T-15	-	ns
Data output hold time (to $\overline{\text{WRn}}$↑)	<28>	T_{HWROD}		T-10	-	ns
$\overline{\text{WAIT}}$ setting time (to Address)	<29>	T_{SAWT1}	n ≥ 1	-	1.5T-30	ns
	<30>	T_{SAWT2}		-	(1.5+n)T-30	ns
$\overline{\text{WAIT}}$ maintenance time (to Address)	<31>	T_{HAWT1}	n ≥ 1	(0.5+n)T	-	ns
	<32>	T_{HAWT2}		(1.5+n)T	-	ns
$\overline{\text{WAIT}}$ setting time (to ASTB↓)	<33>	T_{SSTWT1}	n ≥ 1	-	T-20	ns
	<34>	T_{SSTWT2}		-	(1+n)T-20	ns
$\overline{\text{WAIT}}$ maintenance time (to ASTB↓)	<35>	T_{HSTWT1}	n ≥ 1	NT	-	ns
	<36>	T_{HSTWT2}		(1+n)T	-	ns
$\overline{\text{HLDRQ}}$ high level width	<37>	T_{WHQH}		T+10	-	ns
$\overline{\text{HLDAK}}$ low level width	<38>	T_{WHAL}		T-10	-	ns
$\overline{\text{HLDAK}}$↑ → Bus output delay time	<40>	T_{DHAC}		-3	-	ns
$\overline{\text{HLDRQ}}$↓ → $\overline{\text{HLDAK}}$↓ delay time	<41>	T_{DHQHA1}		-	(2.5+n)T+20	ns
$\overline{\text{HLDRQ}}$↑ → $\overline{\text{HLDAK}}$↑ delay time	<42>	T_{DHQHA2}		0.5T	1.5T+20	ns

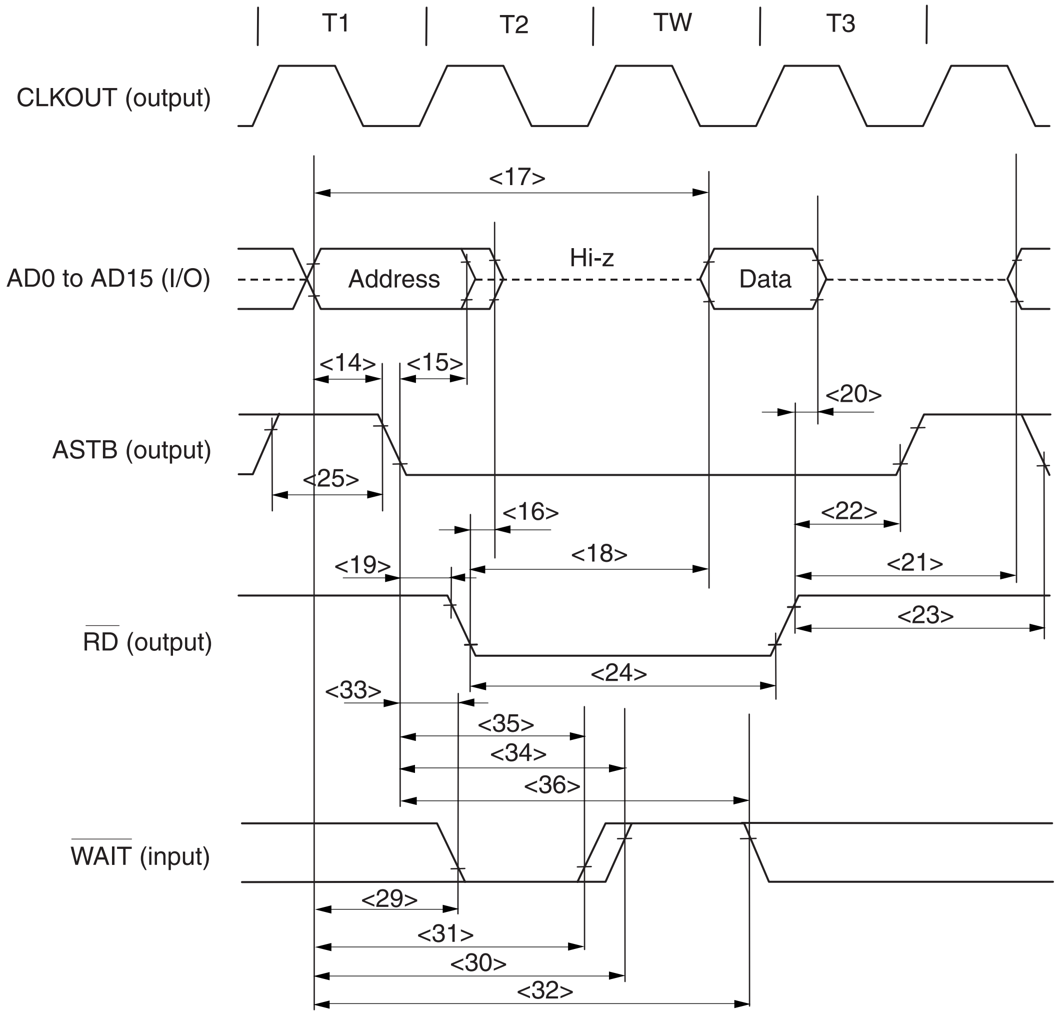
Notes: 1. T= t_{CYK}

2. n shows the number of the wait clocks which is inserted in the bus cycle.
In programmable wait insertion, sampling timing changes.

3. i shows idle state's number (0 or 1) which is inserted after the lead cycle.

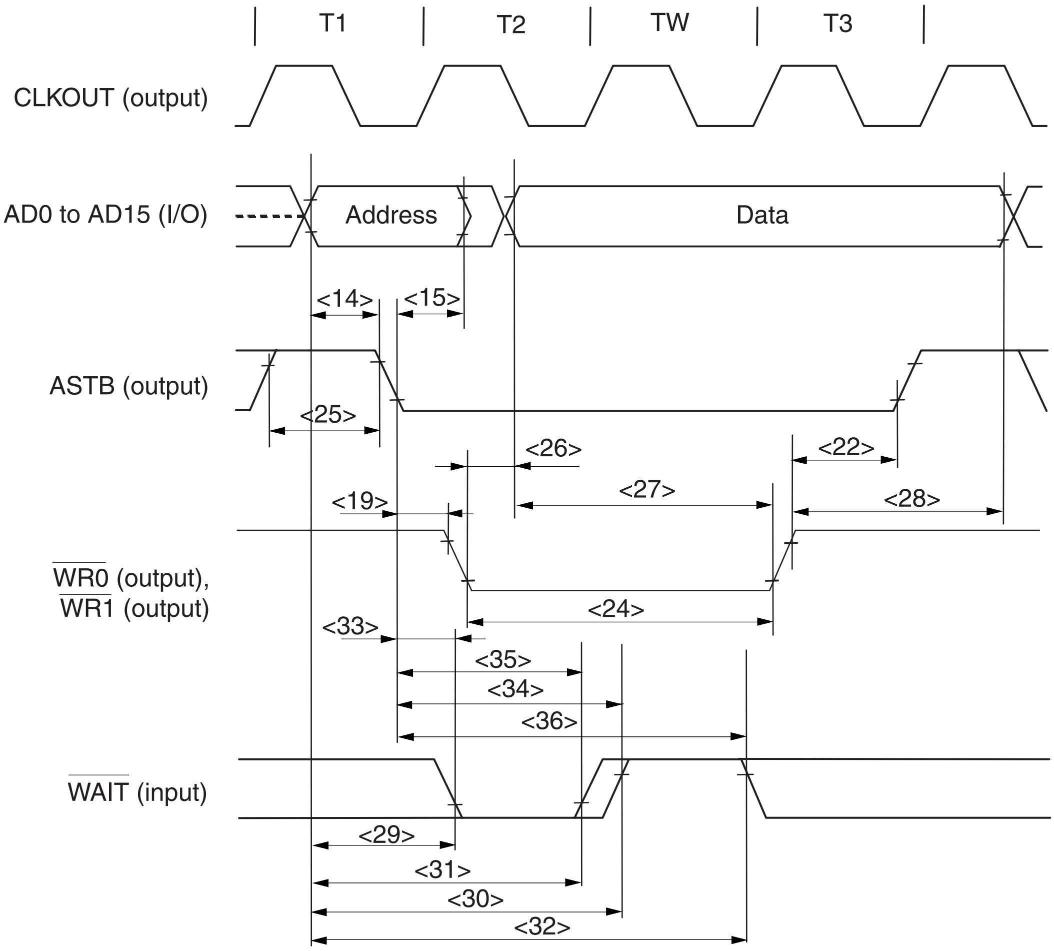
4. This product doesn't support CLKOUT synchronous mode.

Figure 27-7: Read Cycle (CLKOUT Asynchronous, 1 Wait)



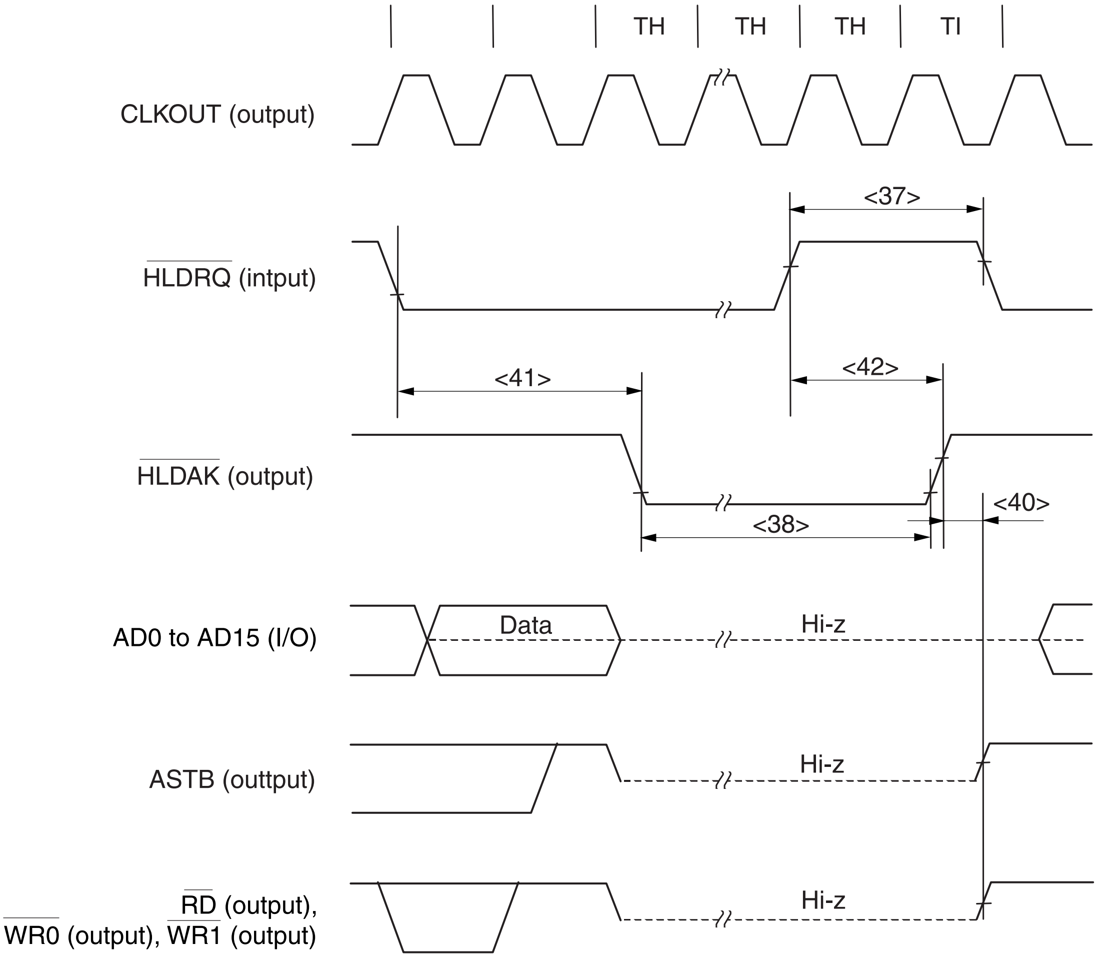
Remark: $\overline{\text{WR0}}$ and $\overline{\text{WR1}}$ are high level.

Figure 27-8: Write Cycle (CLKOUT Asynchronous, 1 WAIT)



Remark: $\overline{\text{RD}}$ is high level.

Figure 27-9: Bus Hold



27.8.8 CSIBn timing

(a) Master mode

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCK}}$Bn cycle	t_{KCYM}	Output	125	-	ns
$\overline{\text{SCK}}$Bn high-level width	t_{KWHM}	Output	$0.5t_{KCYMn}$-10	-	ns
$\overline{\text{SCK}}$Bn low-level width	t_{KWLM}	Output	$0.5t_{KCYMn}$-10	-	ns
SIBn setup time (vs. $\overline{\text{SCK}}$Bn)	t_{SSIM}		30	-	ns
SIBn hold time (vs. $\overline{\text{SCK}}$Bn)	t_{HSIM}		30	-	ns
SOBn output delay time (vs. $\overline{\text{SCK}}$Bn)	t_{HSOM}		-	30	ns
SOBn output hold time (vs. $\overline{\text{SCK}}$Bn)	t_{HSOM}		$0.5t_{KCYMn}$-20	-	ns

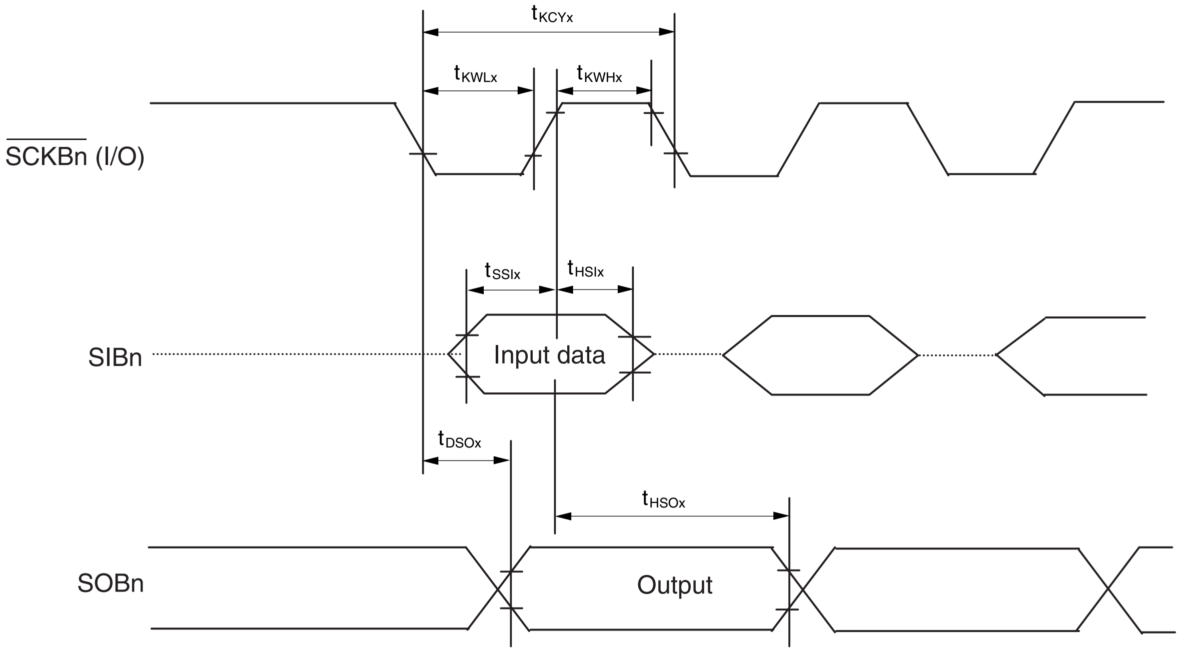
Remark: n = 0, 1

(b) Slave mode

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCK}}$Bn cycle	t_{KCYS}	Input	125		ns
$\overline{\text{SCK}}$Bn high-level width	t_{KWHS}	Input	$0.5t_{KCYSn}$-10		ns
$\overline{\text{SCK}}$Bn low-level width	t_{KWLS}	Input	$0.5t_{KCYSn}$-10		ns
SIBn setup time (vs. $\overline{\text{SCK}}$Bn)	t_{SSIS}		30		ns
SIBn hold time (vs. $\overline{\text{SCK}}$Bn)	t_{HSIS}		30		ns
SOBn output delay time (vs. $\overline{\text{SCK}}$Bn)	t_{DSOS}			30	ns
SOBn output hold time (vs. $\overline{\text{SCK}}$Bn)	t_{HSOS}		$0.5t_{KCYSn}$-10		ns

Remark: n = 0, 1

Figure 27-10: CSIBn Timing (CBnCKP=0, CBnDAP=0)



Remarks: 1. Broken line indicates high impedance.

2. “x” stands for either “M” (master mode) or “S” (slave mode)

27.8.9 UARTAn timing

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Baud rate			-	312.5	kbps
ASCK0 frequency			-	10	MHz

Remark: n = 0, 1

27.8.10 CAN Interface timing (µPD70F3402)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
CAN baud rate			-	83.333	kbps
Internal transmit to receive data delay		$t_{CTXDn} + t_{CRXDn}$	-	100	ns

Remark: n = 0

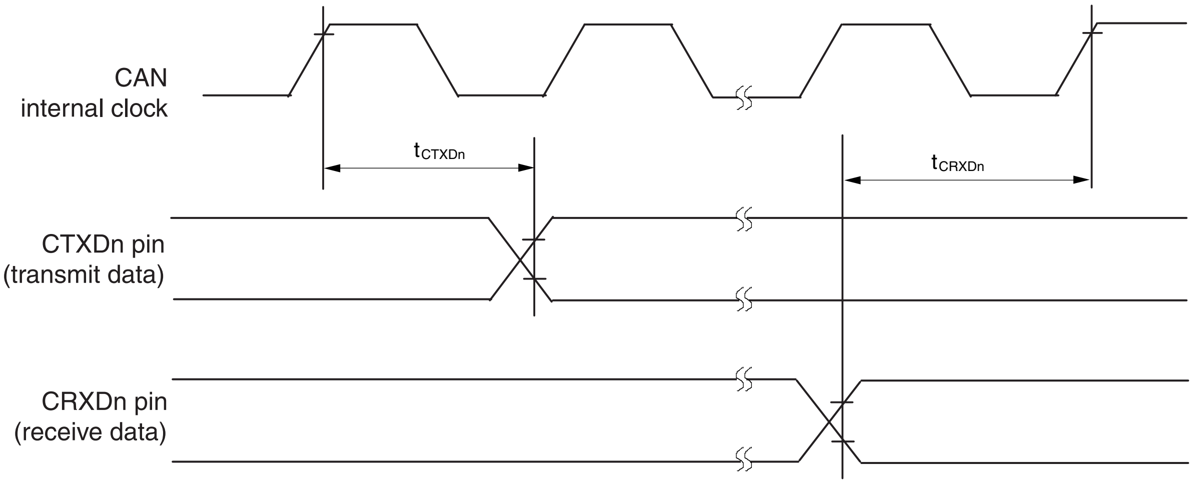
27.8.11 CAN Interface timing (µPD70F3403, µPD70F3403A)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
CAN baud rate			-	1	Mbps
Internal transmit to receive data delay		$t_{CTXDn} + t_{CRXDn}$	-	100	ns

Remark: n = 0, 1

27.8.12 CAN Interface timing diagram

Figure 27-11: CAN Internal Timing



27.8.13 CSI3n timing

(a) Master mode: ($\overline{\text{SCK3n}}$: output)

Parameter	Symbol	MIN.	MAX.	Unit
Macro operation clock cycle time	t_{KCY3}	25	-	ns
$\overline{\text{SCK3n}}$ cycle time	t_{KCY3M}	100	-	ns
$\overline{\text{SCK3n}}$ high, low width	t_{KWH3M}, t_{KWL3M}	0.5t_{KCY3M}-10.0	-	ns
SI3n setup time (vs. $\overline{\text{SCK3n}}$)	t_{SSI3M}	20.0	-	ns
SI3n hold time (vs. $\overline{\text{SCK3n}}$)	t_{HSI3M}	10.0	-	ns
SO3n output delay (vs. $\overline{\text{SCK3n}}$)	t_{DSO3M}	-	10	ns
SO3n hold time (vs. $\overline{\text{SCK3n}}$)	t_{HSO3M}	0.5t_{KCY3M}-10.0	-	ns
CS3nm inactive (High) width (vs. $\overline{\text{SCK3n}}$)	t_{WSCSB}	0.5t_{KCY3M}-10.0	-	ns
CS3nm setup time (vs. $\overline{\text{SCK3n}}$)	t_{SSCSB0}	t_{KCY3}-10.0	-	ns
	t_{SSCSB1}	t_{KCY3M}-10.0	-	ns
	t_{SSCSB2}	t_{KCY3M}-t_{KCY3}-10.0	-	ns
CS3nm hold time (vs. $\overline{\text{SCK3n}}$)	t_{HSCSB0}	t_{KCY3}-10.0	-	ns
	t_{HSCSB1}	0.5t_{KCY3}-10.0	-	ns

Remark: n = 0 to 1, m = 0 to 3

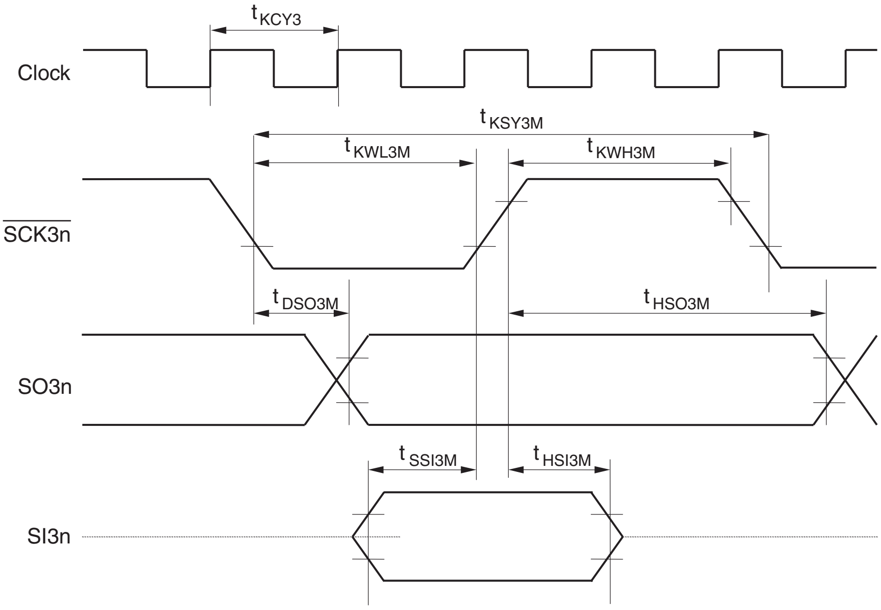
(b) Slave mode: ($\overline{\text{SCK3n}}$ is input mode)

Parameter	Symbol	MIN.	MAX.	Unit
Macro operation clock cycle time	t_{KCY3}	25	-	ns
$\overline{\text{SCK3n}}$ cycle time	t_{KCY3S}	100	-	ns
$\overline{\text{SCK3n}}$ high, low width	t_{KWH3S}, t_{KWL3S}	0.5t_{KCY3S} - 10.0	-	ns
SI3n setup time (vs. $\overline{\text{SCK3n}}$)	t_{SSI3S}	10.0	-	ns
SI3n hold time (vs. $\overline{\text{SCK3n}}$)	t_{HSI3S}	1.5t_{KCY3} + 10.0	-	ns
SO3n output delay (vs. $\overline{\text{SCK3n}}$)	t_{DSO3S}	-	20	ns
SO3n hold time (vs. $\overline{\text{SCK3n}}$)	t_{HSO3S}	0.5t_{KCY3S} - 10.0	-	ns

Remark: n = 0 to 1, m = 0 to 3

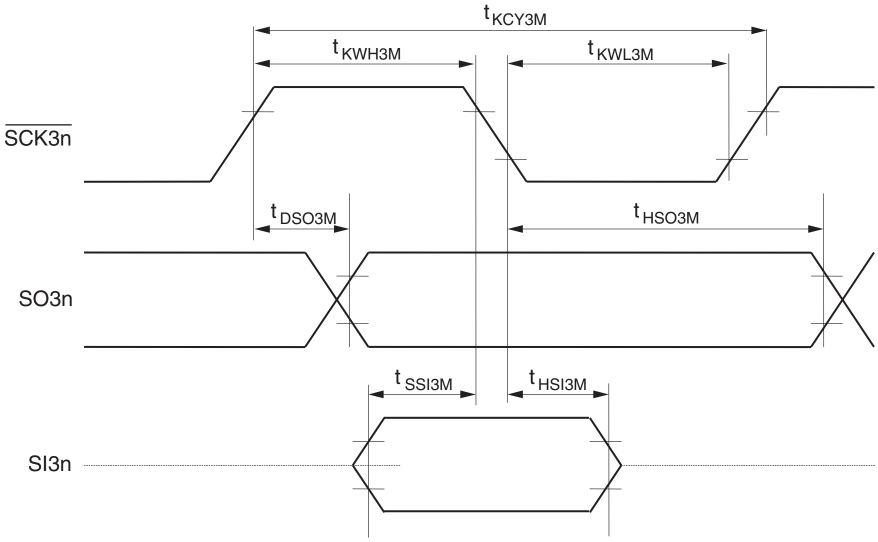
Figure 27-12: CSI3n Timings (1/2)

(a) [$\overline{\text{SCK3n}}$/SI3n/SO3n] pins in master mode: (CSIM:CKP/DAP=0/0 or 1/1)



Remark: n = 0 to 1

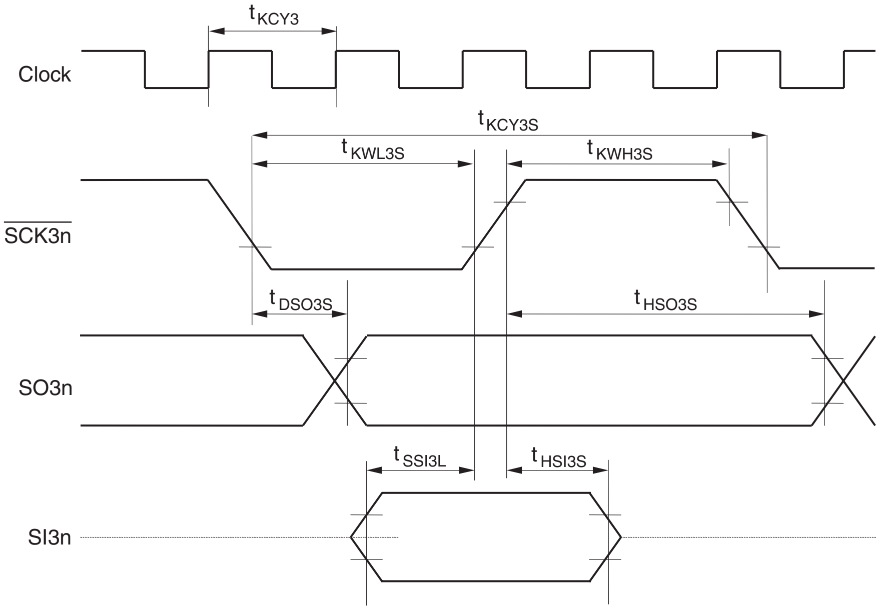
(b) [$\overline{\text{SCK3n}}$/SI3n/SO3n] pins in master mode: (CSIM:CKP/DAP=1/0 or 0/1)



Remark: n = 0 to 1

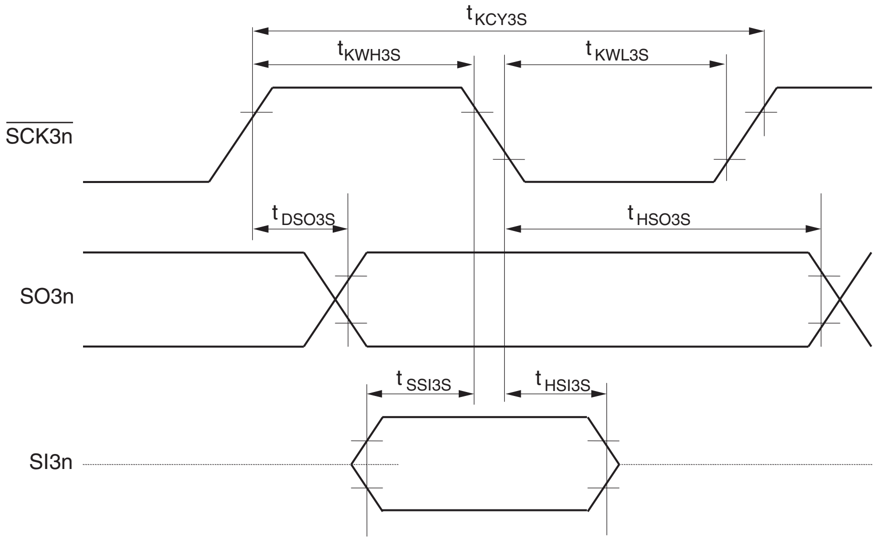
Figure 27-12: CSI3n Timings (2/2)

(c) [$\overline{\text{SCK3n}}$/SI3n/SO3n] pins slave mode: (CSIM:CKP/DAP=0/0 or 1/1)



Remark: n = 0 to 1

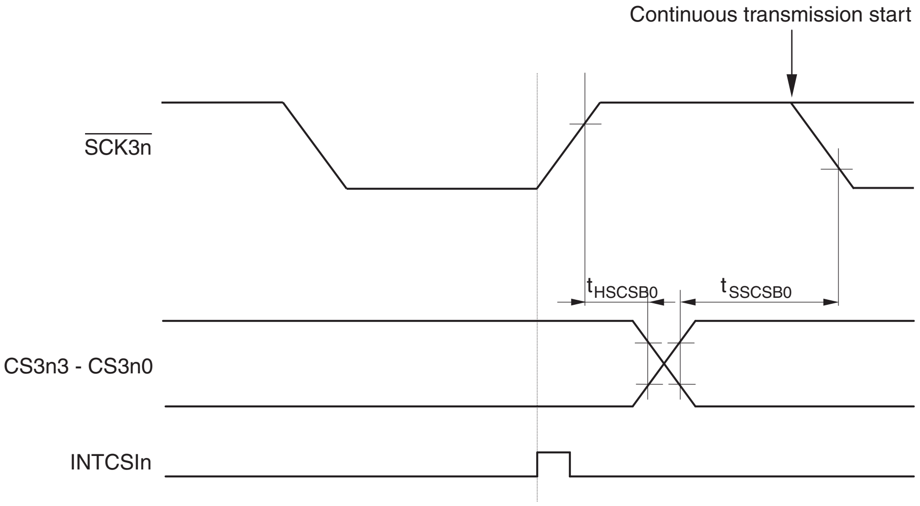
(d) [$\overline{\text{SCK3n}}$/SI3n/SO3n] pins in slave mode: (CSIM:CKP/DAP=1/0 or 0/1)



Remark: n = 0 to 1

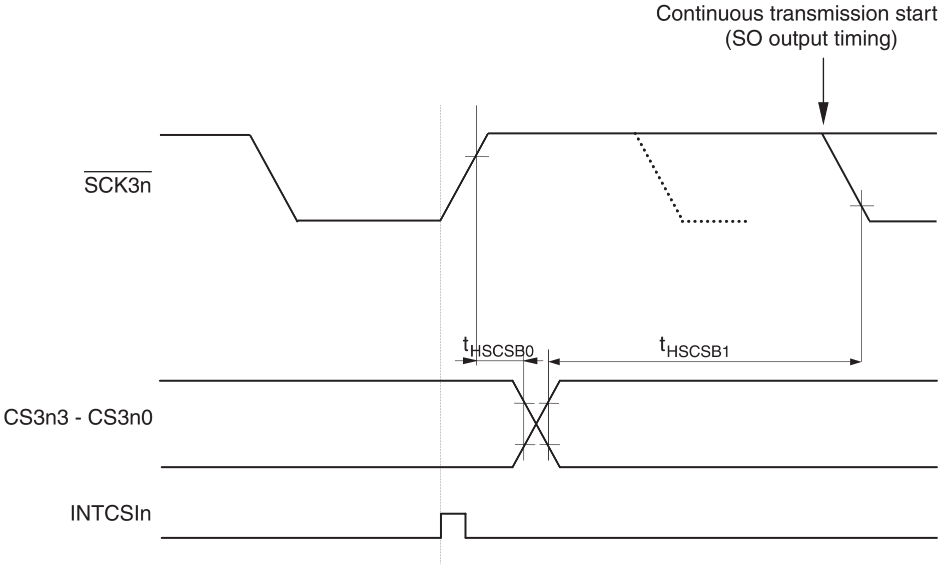
Figure 27-13: CS3n3 - CS3n0 Pins Timings (1/3)

(a) Only in master mode (CSIMn: CSIT = 0 & CSWE/CSMD = 0/0)



Remark: n = 0 to 1

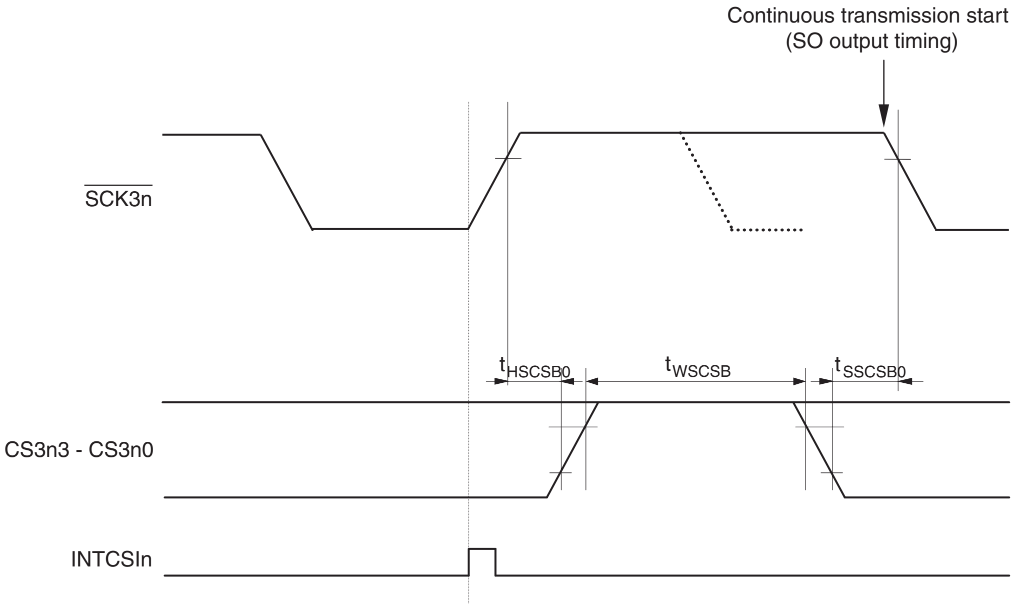
(b) Only in master mode (CSIMn: CSIT = 0 & CSWE/CSMD = 1/0)



Remark: n = 0 to 1

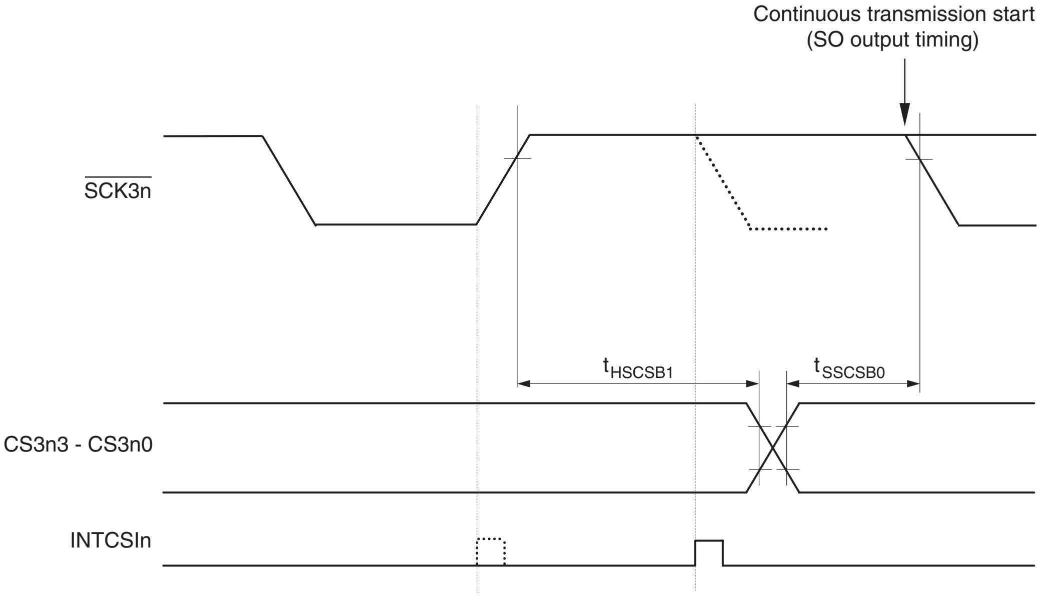
Figure 27-13: CS3n3 - CS3n0 Pins Timings (2/3)

(c) Only in master mode (CSIMn: CSIT = 0 & CSWE/CSMD = 1/1)



Remark: n = 0 to 1

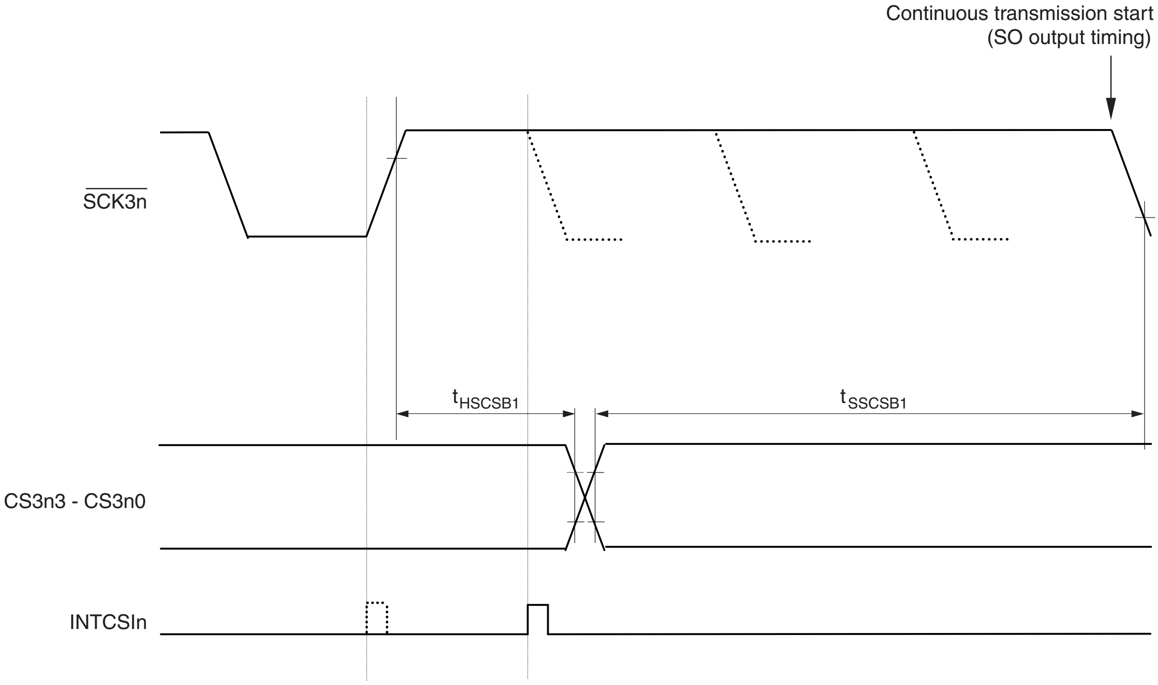
(d) Only in Master mode, (CSIMn: CSIT = 1 & CSWE/CSMD = 0/0)



Remark: n = 0 to 1

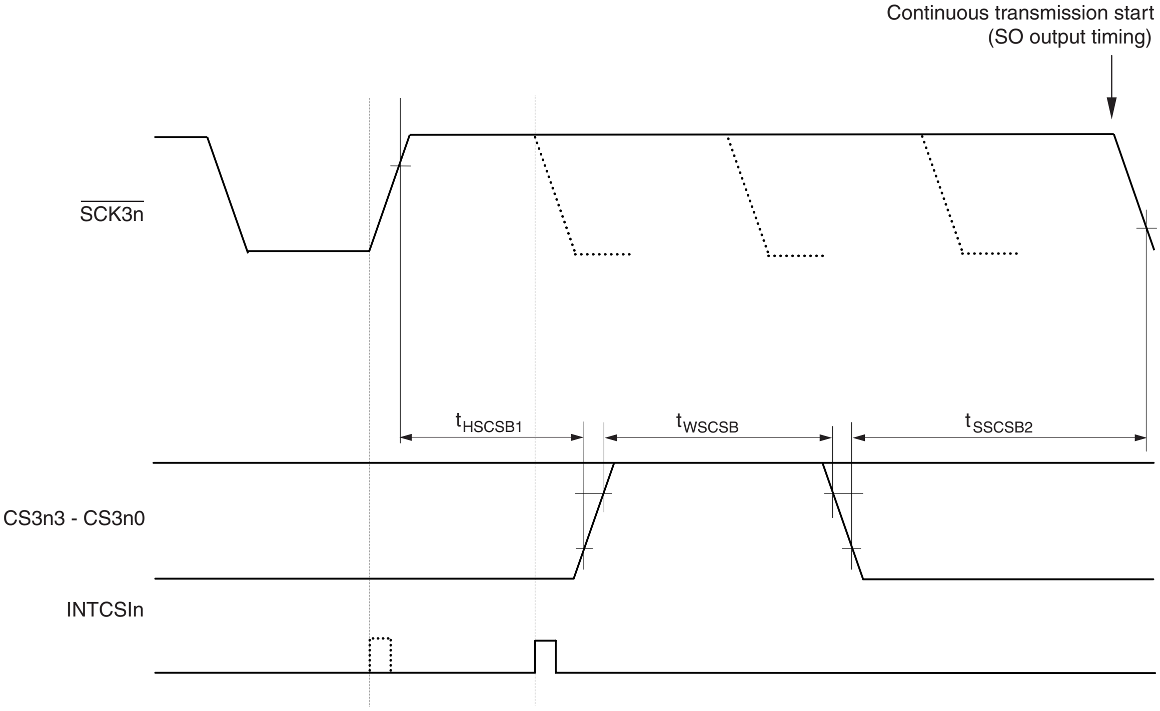
Figure 27-13: CS3n3 - CS3n0 Pins Timings (3/3)

(e) Only in Master mode, (CSIMn: CSIT = 1 & CSWE/CSMD = 1/0)



Remark: n = 0 to 1

(f) In Master mode, (CSIMn: CSIT = 1 & CSWE/CSMD = 1/1)



Remark: n = 0 to 1

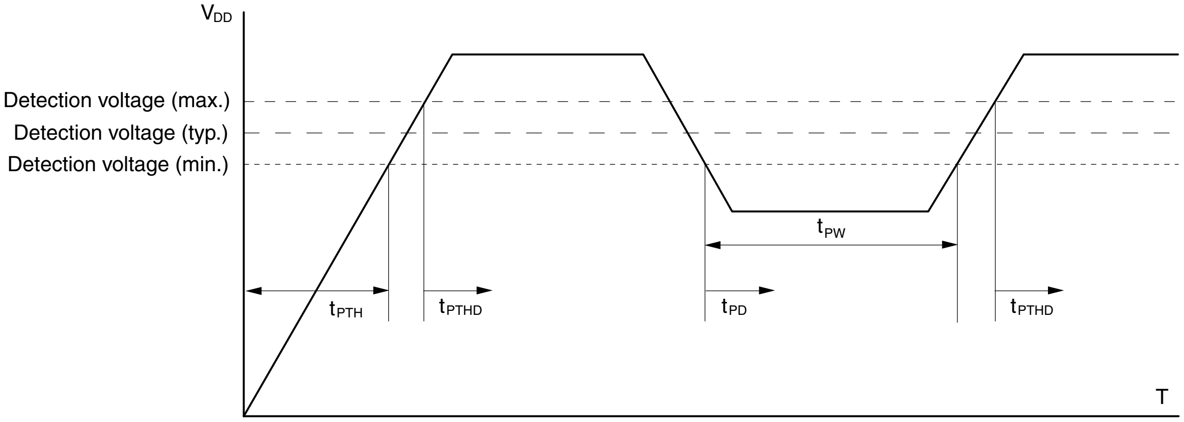
27.8.14 A/D Converter characteristics

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Resolution	-		10	10	10	bit
Total error	-		-	-	±3	LSB
Quantization error	-		-	-	±0.5	LSB
Conversion time	t_{CONV}		2.0	-	15	µs
Recovery time from power-down mode	t_{DPU}		1	-	-	µs
Zero scale error	ZSE	Sampling error not included	-	-	±2	LSB
Full scale error	FSE	Sampling error not included	-	-	±2	LSB
Integral non-linearity error	INL	Sampling error not included	-	-	±3	LSB
Differential non-linearity error	DNL	Sampling error not included	-	-	±2.5	LSB
Analog input voltage	V_{AIN}		-0.2	-	AV_{REF}+0.2	V
Reference voltage	AV_{REF}		4.5	-	5.5	V
AV_{REF} power supply current	AI_{REF}		-	4.5	10	mA

27.8.15 Power-on-clear (POC)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		3.5	3.7	3.9	V
Supply voltage rise time	t_{PTH}	Rise of V_{DD}=0 V to V_{DD}=3.5 V	0.002	-	-	ms
Response time 1	t_{PTHD}	Timing at power-on condition after V_{DD} reaches 3.9V	-	-	3.0	ms
Response time 2	t_{PD}	Timing for power-down after V_{DD} dropped below 3.5V	-	-	1.0	ms
V_{DD} drop minimum width	t_{PW}		0.2	-	-	ms

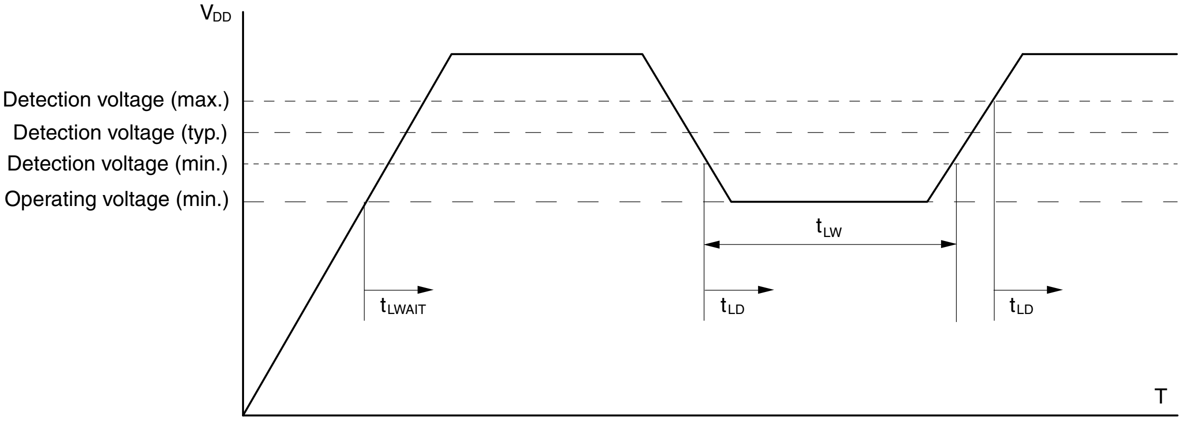
Figure 27-14: Power-on-clear (POC) Waveform



27.8.16 Low-Voltage Indicator (LVI)

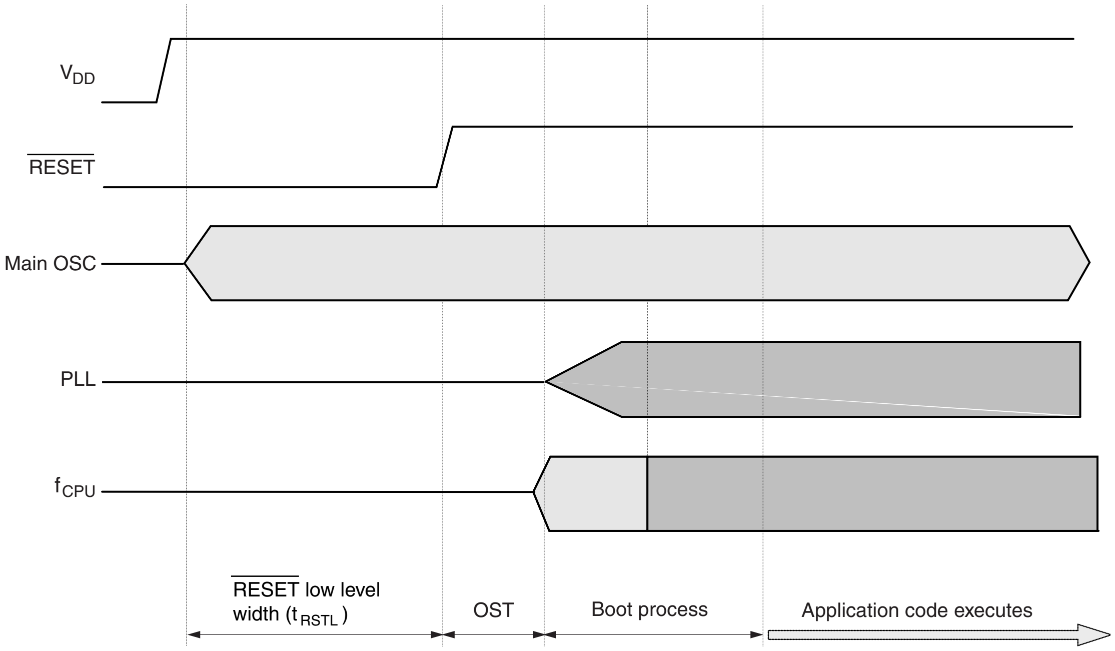
Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.2	4.4	4.6	V
	V_{LVI1}		4.0	4.2	4.4	V
Response time 1	t_{LD}	After V_{DD} reached $V_{LVI0/1}$ (max.) or dropped below $V_{LVI0/1}$ (min.)	-	0.2	2.0	ms
V_{DD} drop minimum width	t_{LW}		0.2	-	-	ms
Reference voltage stabilization time	t_{LWAIT}	After V_{DD} reached the minimum operating voltage and LVION bit (LVIM.bit7) is set to 1.	-	0.1	0.2	ms

Figure 27-15: Low-Voltage Indicator (LVI) Waveform



27.9 Power on Sequence

Figure 27-16: Power on Sequence


V_{DD}
$\overline{RESET}$
Main OSC
PLL
f_{CPU}
$\overline{RESET}$ low level width (t_{RSTL})
OST
Boot process
Application code executes

$\overline{RESET}$ Low level width	f_X	PLL Factor	f_{CPU}	OST	Boot Time	Total Start-up Time
t_{RSTL}	8 MHz	X4	32 MHz	1.024 ms	2.029ms	t_{RSTL} + 3.053 ms

27.10 Flash Memory Characteristics

27.10.1 General condition

-40°C ≤ T_A ≤ +110°C (µPD70F3402)
-40°C ≤ T_A ≤ +85°C (µPD70F3403, µPD70F3403A)
4.0 V ≤ V_{DD0}=V_{DD1}=BV_{DD} ≤ 5.5 V
4.5 V ≤ AV_{REF0} ≤ 5.5 V
V_{SS0}=V_{SS1}=AV_{SS}=BV_{SS}=0 V
C_L=50 pF

27.10.2 Basic Flash characteristics (µPD70F3402)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_X		4	-	32	MHz
Supply voltage	$V_{DD0/1}$		4.0	-	5.5	V
Maximum times of reprogramming Note	C_{WRT}		-	-	100	times
Programming temperature	T_{PRG}		-40	-	+85	°C

Note: Any erase+program or program-only operation is counting as one reprogramming cycle
Example: program -> erase+program -> erase+program: 3 reprogramming times
erase+program -> erase+program: 2 reprogramming times

27.10.3 Basic Flash characteristics (µPD70F3403, µPD70F3403A)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_X		4	-	40	MHz
Supply voltage	$V_{DD0/1}$		4.0	-	5.5	V
Maximum times of reprogramming Note	C_{WRT}		-	-	100	times
Programming temperature	T_{PRG}		-40	-	+85	°C

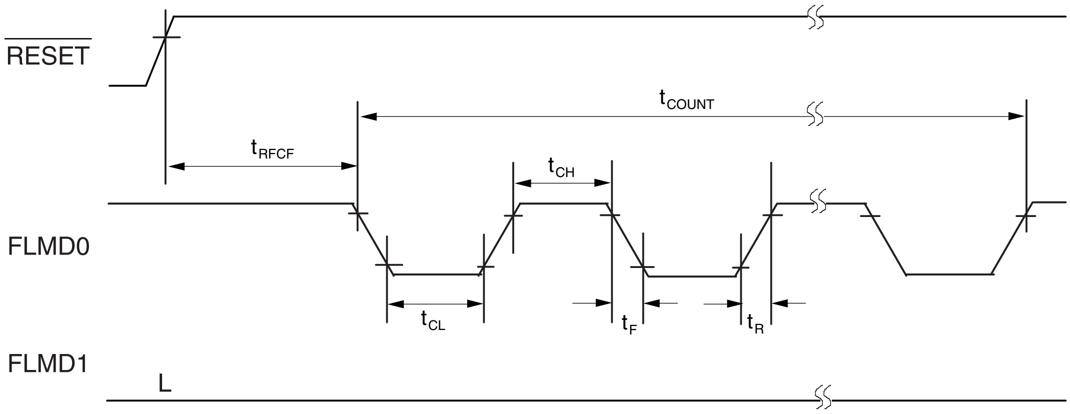
Note: Any erase+program or program-only operation is counting as one reprogramming cycle
Example: program -> erase+program -> erase+program: 3 reprogramming times
erase+program -> erase+program: 2 reprogramming times

27.10.4 External Flash programmer serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (from release of $\overline{\text{RESET}}$ signal)	t_{RFCF}		$16581/f_X+\alpha$ Note	-	-	s
FLMD0 high level width	t_{CH}		10	-	100	µs
FLMD0 low level width	t_{CL}		10	-	100	µs
FLMD0 rise time	t_{RF}		-	-	50	ns
FLMD0 fall time	t_F		-	-	50	ns

Note: α = oscillation stabilization time

Figure 27-17: Flash EPROM Serial Programming Operation Characteristics



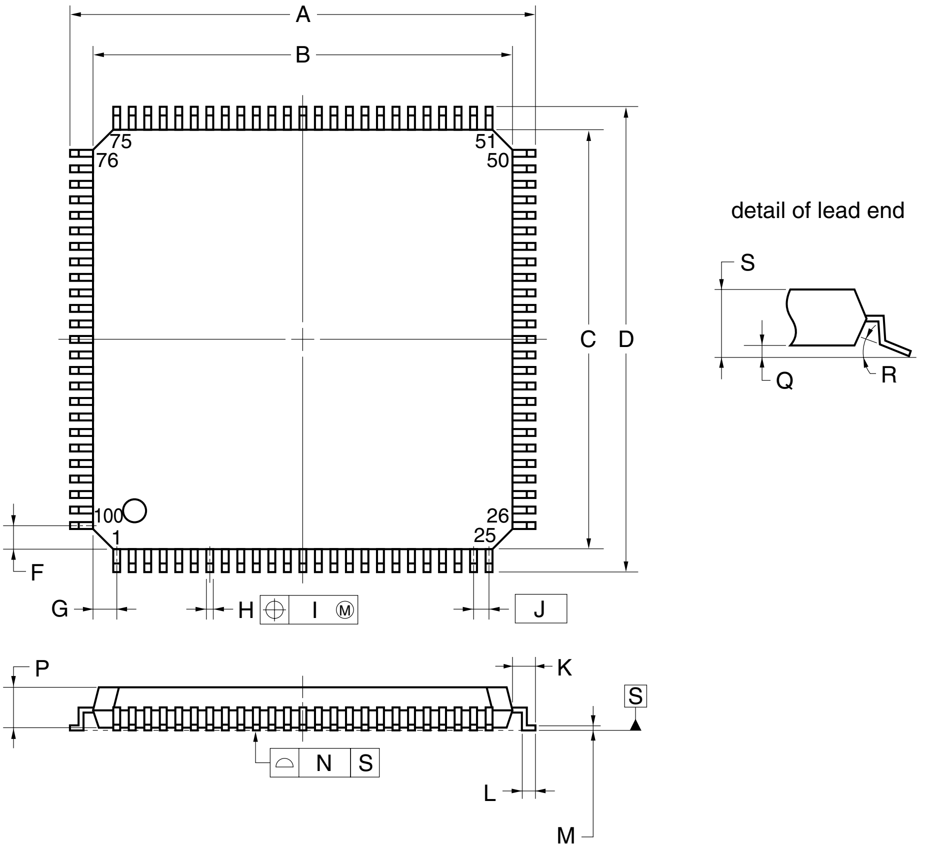
27.10.5 Flash programming characteristics

Parameter	Operating frequency	Conditions	MIN.	TYP.	MAX.	Unit
Block erase	f_{XX}=32 MHz	8 KB	-	651	-	ms
		56 KB	-	3082	-	ms
Write	f_{XX}=32 MHz	256 bytes	-	8.7	-	ms
Block verification	f_{XX}=32 MHz	8 KB	-	47.8	-	ms
		56 KB	-	335	-	ms
Block blank check	f_{XX}=32 MHz	8 KB	-	22.2	-	ms
		56 KB	-	156	-	ms
Flash information setting	f_{XX}=32 MHz		-	0.4	-	ms

Chapter 28 Package Drawing

Figure 28-1: Package Drawing

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3°^{+7°}_{-3°}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

[MEMO]

Chapter 29 Recommended Soldering Conditions

V850E/RS1 should be soldered and mounted under the following recommended conditions. For soldering methods and condition other than those recommended below, contact an NEC Electronics sales representative. For technical information, see the follow website:

http://www.ee.nec.de

µPD70F3402GC(A1)-8EA	100-pin plastic LQFP (Fine pitch) (14 × 14)
µPD70F3403GC(A)-8EA	100-pin plastic LQFP (Fine pitch) (14 × 14)
µPD70F3403AGC(A)-8EA	100-pin plastic LQFP (Fine pitch) (14 × 14)

Table 29-1: Soldering Conditions

Soldering Method	Soldering Condition	Symbol of Recommended Soldering Condition
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: 3 times or less, Exposure limits: 7days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR35-207-3
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds max. (at 200°C or higher), Count: 3 times or less, Exposure limits: 7days[Note] (after that, prebake at 125°C for 20 to 72 hours)	VP15-207-3
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.

[MEMO]

Appendix A   Instruction Set List

A.1 Convention

(a) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General registers: Used as source registers
reg2	General registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General registers: Used mainly to store the remainders of division results and the higher order 3 bits of multiplication results.
bit#3	33-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (SP)
ep	Element pointer (r30)
listX	X item register list

(b) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list
S	1-bit data that specifies a system register in the register list

(c) Register symbols used in operation

Register Symbol	Explanation
←	Input for
GR [ ]	General 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)
Half-word	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

(d) Register symbols used in an 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).

(e) 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.

(f) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	—	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

A.2 Instruction Set (In Alphabetical Order)

(1/4)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2] + GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2] + sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] + sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2] AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] AND zero-extend(imm16)		1	1	1		0	0	×	×
Bcond	disp9	ddddd1011dddcccc Note 1	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	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) ‖ GR[reg2] (31: 24) ‖ GR[reg2] (7: 0) ‖ GR[reg2] (15: 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7: 0) ‖ GR[reg2] (15: 8) ‖ GR[reg2] (23: 16) ‖ 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,Half-word))		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	rrrrr111111RRRRR 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
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 are loaded		N+1 Note 4	N+1 Note 4	N+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR Note 5	sp←sp + zero-extend(imm5 logically shift left by 2) R[reg in list12]←Load-memory(sp,Word) sp←sp + 4 repeat 2 steps above until all regs in list12 are 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) ‖ GR[reg2] (31: 16)		1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 Note 7	GR[reg2]←PC + 4 PC←PC + sign-extend(disp22)		2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]		3	3	3					

(2/4)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
JR	disp22	0000011110dddddd ddddddddddddddd0 Note 7	PC←PC + sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	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					
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 Note 8	adr←GR[reg1] + sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 Note 12	SR[regID]←GR[reg2] — Other than regID = PSW	1	1	1					
			SR[regID]←GR[reg2] — regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 Note 8	adr←GR[reg1]+sign-exend(disp16) GR[reg2]←zero-extend(Load-memory(adr,half-word)	1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 Note 8	adr←GR[reg1] + sign-exend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)	1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] + sign-extend(imm16)	1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] + (imm16 ‖ 0^{16})	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ‖ GR[reg2]←GR[reg2] × GR[reg1]	1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00	GR[reg3] ‖ GR[reg2]←GR[reg2] × sign-extend(imm9) **Note 13**	1	2 Note 14	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]$^{\text{Note 6}}$ × GR[reg1]$^{\text{Note 6}}$	1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2]$^{\text{Note 6}}$ × sign-extend(imm5)	1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]$^{\text{Note 6}}$ × imm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ‖ GR[reg2]←GR[reg2]xGR[reg1]	1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10	GR[reg3] ‖ GR[reg2]←GR[reg2]xzero-extend(imm9) Note 13	1	2 Note 14	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	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				×	
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 are stored sp←sp – zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm$^{\text{Note 15}}$	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 are 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					

(3/4)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
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					
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1] + sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep + zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd Note 18	adr←ep + zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd Note 19	adr←ep + zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	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,Half-word))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 Note 21	adr←ep + zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep + zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd Note 19	adr←ep + zero-extend(disp8) Store-memory(adr,GR[reg2],Half-word)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 Note 21	adr←ep + zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1] + sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 Note 8	adr←GR[reg1] + sign-extend(disp16) Store-memory (adr,GR[reg2], Half-word)	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					
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	×	×	×	×	

(4/4)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Half-word))) 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	00000111111iiiii 0000000100000000	EIPC←PC+4 (Return 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 dddddddddddddddd	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 iiiiiiiiiiiiiiii	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. dddddddd: Higher 8 bits of disp9.

2. 3 clocks if the final instruction includes PSW write access.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list X load registers. (According to the number of wait states. Also, if there are no wait states, n is the number of list X registers.)

5. RRRRR: other than 00000.

6. The lower half word data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

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: Lower 4 bits of imm9.

14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.
10: Load sign expanded 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.

Appendix B Index

A
A/D conversion result register n . . . 371
A/D converter mode register 0H . . . 374
A/D converter mode register 0L . . . 375
A/D discharge and diagnostic mode register . . . 378
AD converter external trigger selection register . . . 379
AD converter mode register 1H . . . 377
ADA0CRDD . . . 371
ADA0CRn . . . 371
ADA0CRSS . . . 371
Address wait control register . . . 221
ADSCM0H . . . 374
ADSCM0L . . . 375
ADSCM1H . . . 377
ADVMS0 . . . 378
AWC . . . 221

B
BCC . . . 222
BSC . . . 212
Bus cycle control register . . . 222
Bus size configuration register . . . 212

C
Capture/compare register 0 . . . 258, 306
Capture/compare register 1 . . . 259, 307
Capture/compare register 2 . . . 308
Capture/compare register 3 . . . 309
CBnCTL0 . . . 434
CBnCTL1 . . . 436
CBnCTL2 . . . 438
CBnRX . . . 440
CBnSTR . . . 439
CBnTX . . . 440
CCLS . . . 240
Chip select data buffer registers . . . 471
CKC . . . 247
CLM . . . 790
Clock monitor mode register . . . 790
Clock selection register 0 . . . 248
Clock selection register 1 . . . 249
Clock selection register 2 . . . 242
Clock selection register 3 . . . 251
CPU operation clock status register . . . 240
CRC data register . . . 794
CRC input register . . . 794
CRCD . . . 794
CRCIN . . . 794
CSIBn control register 0 . . . 434
CSIBn control register 1 . . . 436
CSIBn control register 2 . . . 438
CSIBn receive data register . . . 440
CSIBn status register . . . 439
CSIBn transmit data register . . . 440
CSIC0 . . . 468

CSIC1 468
CSIL0 . . . . . 474, 475
CSIL1 . . . . . 474, 475
CSIM0 466
CSIM1 466

D
Data wait control register 0 219
DMA addressing control register 511
DMA channel control register 513
DMA channel status flag register 507
DMA control register 505
DMA destination address register 509
DMA source address register 507
DMA transfer count register 510
DMA trigger factor register 517
DMADCn 511
DMBCn 510
DMC 505
DMCHCn 513
DMDAn 509
DMSAn 507
DMSF 507
DTFRn 517
DWC0 219

E
EP 716
Exception status flag 716
EXCKSEL 237
Extension clock select register 237
External interrupt falling edge specification register 0 709
External interrupt falling edge specification register 1 710
External interrupt falling edge specification register 3 711
External interrupt falling edge specification register 9H 712
External interrupt rising edge specification register 1 710
External interrupt rising edge specification register 9H 712

I
IMR0 to IMR3 704
In-service priority register 705
Interrupt control register 701
Interrupt mask registers 0 to 3 704
INTF0 709
INTF1 710
INTF3 711
INTF9H 712
INTR1 710
INTR3 711
INTR9H 712
ISPR 705

L
Low-voltage detection level selection register 783
Low-voltage detection register 782
LVIM 782
LVIS 783

M
Main peripheral clock control register 235
MPCCTL 235

N
Noise elimination control register 713

O
OCDM 771
OCKS0 248
OCKS1 249
OCKS2 242
OCKS3 251
On-chip debug mode register 771
Oscillation stabilization time select register 241, 360
OSTS 241, 360

P
PCC 239
PCLM 250
PLL control register 0 245
PLL control register 1 245
PLLCTL0 245
PLLCTL1 245
Power save control register 737
Power save mode register 238, 738
Prescaler compare register 0 462
Prescaler mode register 0 461
Processor clock control register 239
Programmable clock mode register 250
PRSCM0 462
PRSM0 461
PSC 737
PSMR 238, 738

Q
QSPI clock selection registers 468
QSPI operation mode registers 466
QSPI transmit length selection registers 474

R
RCM 240
Receive data buffer registers 471
Reset source flag register 740
RESF 740
Ring-OSC mode register 240

S

SAR 369
SELCNT1 379
SFA0 472
SFA1 472
SFCS0 471
SFCS1 471
SFDB0 472
SFDB1 472
SIRB0 471
SIRB1 471
Successive approximation register 369

T

Timer Q control register 1 312
Timer Q dedicated I/O control register 0 314
Timer Q dedicated I/O control register 1 315
Timer Q dedicated I/O control register 2 316
Timer Q option register 0 317
Timer Q0 control register 0 311
Timer read buffer register 310
TMMn compare register 0 356
TMMn timer control register 357
TMnCMP0 356
TMnCTL0 357
TMP dedicated I/O control register 2 266
TMPn control register 0 261
TMPn control register 1 262
TMPn dedicated I/O control register 0 264
TMPn dedicated I/O control register 1 265
TMPn option register 0 267
TMPn timer read buffer register 260
TPnCCR0 258
TPnCCR1 259
TPnCNT 260
TPnCTL0 261
TPnCTL1 262
TPnIOC0 264
TPnIOC1 265
TPnIOC2 266
TPnOPT0 267
TQnCCR0 306
TQnCCR1 307
TQnCCR2 308
TQnCCR3 309
TQnCNT 310
TQnCTL0 311
TQnCTL1 312
TQnIOC0 314
TQnIOC1 315
TQnIOC2 316
TQnOPT0 317
Transmission data buffer registers 472
Transmit buffer status registers 472

U
UAnCTL0 407
UAnCTL1 409
UAnOPT0 411
UAnRX 414
UAnSTR 412
UARTAn control register 0 407
UARTAn control register 1 409
UARTAn option control register 0 411
UARTAn receive data register 414
UARTAn status register 412

W
Watchdog timer enable register 363
Watchdog timer mode register 2 . . . . . 361, 707
WDTE 363
WDTM2 . . . . . 361, 707

X
xxICn 701

Z
External interrupt rising edge specification register 3 711

Appendix C Revision History

The following shows the revision history up to present. Application portions signifies the chapter of each edition.

(1/2)

Edition No.	Major items revised	Revised Sections
EE3V2	Figure 15-6 “DMA Transfer Count Register (DMBCn) Format”, Note, bit ACF changed to bit EN	15.3 (5), p.510
	Figure 15-10 “Maskable Interrupt Status Flag Format”, bit ACF description changed, 3 timings for ACF bit status added	15.3 (7), p.514-516
	Figure 15-10 “Maskable Interrupt Status Flag Format”, bit EN description changed	15.3 (7), p.516
	Section 15.4.4 “DMA transfer end“, description changed	15.4.4, p.521

(2/2)

Edition No.	Major items revised	Revised Sections

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics America Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Marketing Services & Publishing
Fax: +49(0)-211-6503-1344

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-6250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: ______________________________

Document number: ______________________ Page number: ____________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	□	□	□	□
Technical Accuracy	□	□	□	□
Organization	□	□	□	□