NEC

User's Manual

V850ES/Fx2

32-Bit Single-Chip Microcontroller

Hardware

μPD703230(A)	*μPD70(F)3231(A)*	
μPD703230(A1)	*μPD70(F)3231(A1)*	
μPD703230(A2)	*μPD70(F)3231(A2)*	
μPD70(F)3232(A)	*μPD70(F)3233(A)*	
μPD70(F)3232(A1)	*μPD70(F)3233(A1)*	
μPD70(F)3232(A2)	*μPD70(F)3233(A2)*	
μPD70(F)3234(A)	*μPD70(F)3235(A)*	*μPD70F3236(A)*
μPD70(F)3234(A1)	*μPD70(F)3235(A1)*	*μPD70F3236(A1)*
μPD70(F)3234(A2)	*μPD70(F)3235(A2)*	*μPD70F3236(A2)*
μPD70F3237(A)	**μPD70F3238(A)**	**μPD70F3239(A)**
μPD70F3237(A1)	**μPD70F3238(A1)**	**μPD70F3239(A1)**
μPD70F3237(A2)	**μPD70F3238(A2)**	**μPD70F3239(A2)**

Document No. U17830EE1V0UM00
Date Published November 2005-NS CP(K)

© NEC Electronics Corporation 2005
Printed in Europe

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ 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 November, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

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

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

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

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

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

(Note)

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

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

M8E 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J04.1

PREFACE

Readers

For the whole document it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

This manual is intended for users who wish to understand the functions of the V850ES/Fx2 and design application systems using these products.

The target products are as follows.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850ES/Fx2 shown in the **Organization** below.

Organization

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

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

How to Read This Manual

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

To understand the details of an instruction function

→ Refer to the **V850ES Architecture User's Manual**.

Register format

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

→ Regarding the pin functions and Internal peripheral functions of products, please read and change the products as follows.

→ ・μ PD703230 → μ PD703230(A), μ PD703230(A1), μ PD703230(A2)

→ ・μ PD70F3231 → μ PD70(F)3231(A), μ PD70(F)3231(A1), μ PD70(F)3231(A2)

→ ・μ PD70F3232 → μ PD70(F)3232(A), μ PD70(F)3232(A1), μ PD70(F)3232(A2)

→ ・μ PD70F3233 → μ PD70(F)3233(A), μ PD70(F)3233(A1), μ PD70(F)3233(A2)

→ ・μ PD70F3234 → μ PD70(F)3234(A), μ PD70(F)3235(A1), μ PD70(F)3234(A2)

→ ・μ PD70F3235 → μ PD70(F)3235(A), μ PD70(F)3235(A1), μ PD70(F)3235(A2)

→ ・μ PD70F3236 → μ PD70F3236(A), μ PD70F3236(A1), μ PD70F3236(A2)

→ ・μ PD70F3237 → μ PD70F3237(A), μ PD70F3237(A1), μ PD70F3237(A2)

→ ・μ PD70F3238 → μ PD70F3238(A), μ PD70F3238(A1), μ PD70F3238(A2)

→ ・μ PD70F3239 → μ PD70F3239(A), μ PD70F3239(A1), μ PD70F3239(A2)

To understand the overall functions of the V850ES/Fx2

→ Read this manual according to the **CONTENTS**. The mark ★ shows major revised points.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representation:	$\overline{xxx}$ (over score over pin or signal name)	
Memory map address:	Higher addresses on the top and lower addresses on the bottom	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numeric representation:	Binary	... xxxx or xxxxB
	Decimal	... xxxx
	Hexadecimal	... xxxxH

Prefix indicating power of 2 (address space, memory capacity):

K (kilo):	$2^{10} = 1,024$
M (mega):	$2^{20} = 1,024^2$
G (giga):	$2^{30} = 1,024^3$

Related Documents

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

Documents related to V850ES/Fx2 and sub series (V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2)

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
V850ES/Fx2 Hardware User's Manual	U17830EE1V0UM00
V850ES/FE2 Data Sheet	U17834EE1V0DS00
V850ES/FG2 Data Sheet	U17832EE1V0DS00
V850ES/FF2 Data Sheet	U17833EE1V0DS00
V850ES/FJ2 Data Sheet	U17831EE1V0DS00

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-V850ES-G1 (In-Circuit Emulator)		U16313E
IE-703239-G1-EM1 (In-Circuit Emulator Option Board)		SUD-FT-04-0105
CA850 Ver. 2.70 C Compiler Package	Operation	U16053E
	C Language	U16054E
	PM plus	U16055E
	Assembly Language	U16042E
ID850 Ver. 2.51 Integrated Debugger	Operation	U16217E
RX850 Ver. 3.13 or Later Real-Time OS	Fundamental	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.15 Real-Time OS	Fundamental	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.2 System Performance Analyzer		U14410E
PG-FP4 Flash Memory Programmer		U15260E
IE-V850E1-CD-NW(N-wire)		U16647E
QB-V850ESF*x*2(IECUBE)		ZUD-BD-04-0085
SM plus Ver1.00 System Simulation		U16906J

CONTENTS

CHAPTER 1 INTRODUCTION 19

1.1 General 19

1.2 Product Development of V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2 20

1.3 Features 21

1.4 Ordering Information 23

1.5 Applications 26

1.6 Pin Configuration (Top View) 27

1.7 Function Block Configuration 33

1.7.1 Internal block diagram 33

1.7.2 Internal units 38

CHAPTER 2 PIN FUNCTIONS 40

2.1 Pin Function List 40

2.1.1 V850ES/FE2 40

2.1.2 V850ES/FF2 45

2.1.3 V850ES/FG2 50

2.1.4 V850ES/FJ2 55

2.2 Pin Status (V850ES/FJ2) 63

2.3 Description of Pin Functions 64

2.3.1 V850ES/FE2 64

2.3.2 V850ES/FF2 71

2.3.3 V850ES/FG2 78

2.3.4 V850ES/FJ2 84

2.4 Pin I/O Circuit Types and Recommended Connection of Unused Pins 94

2.4.1 V850ES/FE2 94

2.4.2 V850ES/FF2 96

2.4.3 V850ES/FG2 98

2.4.4 V850ES/FJ2 100

2.5 Pin I/O Circuits 104

CHAPTER 3 CPU FUNCTIONS 106

3.1 Features 106

3.2 CPU Register Set 107

3.2.1 Program register set 108

3.2.2 System register set 109

3.3 Operation Modes 115

3.4 Address Space 116

3.4.1 CPU address space 116

3.4.2 Image 117

3.4.3 Wraparound of CPU address space 118

3.4.4 Memory map 119

3.4.5 Areas 121

3.4.6 Recommended use of address space 134

3.4.7 Peripheral I/O registers 137

3.4.8 Programmable peripheral I/O register 176

3.4.9 Special registers 177

3.4.10 Cautions 181

CHAPTER 4 PORT FUNCTIONS 184
4.1 Features 184
4.2 Basic Port Configuration 184
4.2.1 Basic port configuration on V850ES/FE2 184
4.2.2 Port configuration on V850ES/FF2 185
4.2.3 Port configuration on V850ES/FG2 186
4.2.4 Port configuration on V850ES/FJ2 187
4.3 Port Configuration 188
4.3.1 Port 0 194
4.3.2 Port 1 201
4.3.3 Port 3 206
4.3.4 Port 4 216
4.3.5 Port 5 219
4.3.6 Port 6 226
4.3.7 Port 7 236
4.3.8 Port 8 239
4.3.9 Port 9 244
4.3.10 Port 12 257
4.3.11 Port CD 259
4.3.12 Port CM 261
4.3.13 Port CS 265
4.3.14 Port CT 269
4.3.15 Port DL 273
4.3.16 Port pins that function alternately as on-chip debug function 278
4.3.17 Register settings to use port pins as alternate-function pins 279
4.3.18 Operation of port function 286
4.4 Cautions 287
4.4.1 Cautions of setting port pins 287
4.4.2 Cautions of bit manilulation instruction for port register (Pn) 288
4.4.3 Cautions on on-chip debug pins 288
4.4.4 Cautions on P05/INTP2/DRST pin 288

CHAPTER 5 BUS CONTROL FUNCTION 289
5.1 Features 289
5.2 Bus Control Pins 290
5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed 290
5.2.2 Pin status in each operation mode 290
5.3 Memory Block Function 291
5.3.1 Memory space 291
5.3.2 Chip select function 292
5.4 Bus Access 293
5.4.1 Number of clocks for access 293
5.4.2 Bus size setting function 293
5.4.3 Access according to bus size 294
5.5 Wait Function 300
5.5.1 Programmable wait function 300
5.5.2 External wait function 301
5.5.3 Relationship between programmable wait and external wait 301

5.5.4 Programmable address wait function ... 302
5.6 Idle State Insertion Function ... 303
5.7 Bus Hold Function ... 304
5.7.1 Functional outline ... 304
5.7.2 Bus hold procedure ... 305
5.7.3 Operation in power save mode ... 305
5.8 Bus Priority ... 306
5.9 Boundary Operation Conditions ... 306
5.9.1 Program space ... 306
5.9.2 Data space ... 306
5.10 Bus Timing ... 307
5.10.1 Multiplexed bus ... 307

CHAPTER 6 CLOCK GENERATION FUNCTION ... 313
6.1 Overview ... 313
6.2 Configuration ... 314
6.3 Control Registers ... 316
6.4 Operation ... 319
6.4.1 Operation of each clock ... 319
6.4.2 Clock output function ... 320
6.5 PLL Function ... 320
6.5.1 Overview ... 320
6.5.2 Control registers ... 320
6.5.3 Usage ... 323

CHAPTER 7 16-BIT TIMER/EVENT COUNTER P ... 324
7.1 Features ... 324
7.2 Functional Outline ... 324
7.3 Configuration ... 325
7.4 Control Registers ... 330
7.5 Operation ... 339
7.5.1 Anytime write and reload ... 339
7.5.2 Interval timer mode (TPnMD2 to TPnMD0 = 000) ... 344
7.5.3 External event count mode (TPnMD2 to TPnMD0 = 001) ... 347
7.5.4 External trigger pulse mode (TPnMD2 to TPnMD0 = 010) ... 351
7.5.5 One-shot pulse mode (TPnMD2 to TPnMD0 = 011) ... 354
7.5.6 PWM mode (TPnMD2 to TPnMD0 = 100) ... 357
7.5.7 Free-running mode (TPnMD2 to TPnMD0 = 101) ... 362
7.5.8 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110) ... 367
7.6 Timer Synchronized Operation Function ... 369
7.7 Selector Function ... 373
7.8 Cautions ... 377

CHAPTER 8 16-BIT TIMER/EVENT COUNTER Q ... 380
8.1 Features ... 380
8.2 Functional Outline ... 380
8.3 Configuration ... 381
8.4 Control Registers ... 388

8.5 Operation 398
8.5.1 Anytime write and reload 398
8.5.2 Interval timer mode (TQnMD2 to TQnMD0 = 000) 403
8.5.3 External event count mode (TQnMD2 to TQnMD0 = 001) 406
8.5.4 External trigger pulse mode (TQnMD2 to TQnMD0 = 010) 410
8.5.5 One-shot pulse mode (TQnMD2 to TQnMD0 = 011) 413
8.5.6 PWM mode (TQnMD2 to TQnMD0 = 100) 416
8.5.7 Free-running mode (TQnMD2 to TQnMD0 = 101) ..... 42"
8.5.8 Pulse width measurement mode (TQnMD2 to TQnMD0 = 110) 422
8.5.9 Triangular wave PWM mode (TQnMD2 to TQnMD0 = 111) 431
8.6 Timer Synchronized Operation Function 433
8.7 Cautions 437

CHAPTER 9 16-BIT INTERVAL TIMER M 440
9.1 Features 440
9.2 Configuration 441
9.3 Control Register 443
9.4 Operation 445
9.4.1 Interval timer mode 445
9.5 Cautions 446

CHAPTER 10 WATCH TIMER FUNCTIONS 447
10.1 Functions 447
10.2 Configuration 449
10.3 Control Registers 450
10.4 Operation 452
10.4.1 Operation as watch timer 452
10.4.2 Operation as interval timer 452
10.4.3 Cautions 453
10.5 Prescaler 3 454
10.5.1 Control registers 454
10.5.2 Generation of watch timer count clock 455

CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2 456
11.1 Functions 456
11.2 Configuration 457
11.3 Control Registers 457

CHAPTER 12 A/D CONVERTER 461
12.1 Overview 461
12.2 Functions 461
12.3 Configuration 463
12.4 Control Registers 464
12.5 Operation 473
12.5.1 Basic operation 473
12.5.2 Trigger mode 475
12.5.3 Operation mode 477
12.5.4 Power-fail compare mode 481

12.6 Cautions 486
12.7 How to Read A/D Converter Characteristics Table 491

CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE A (UARTA) 495
13.1 Features 496
13.2 Configuration 497
13.2.1 Control registers 499
13.3 Control Registers 500
13.4 Interrupt Request Signals 508
13.5 Operation 509
13.5.1 Data format 509
13.5.2 SBF transmission/reception format 510
13.5.3 SBF transmission 512
13.5.4 SBF reception 513
13.5.5 UART transmission 514
13.5.6 Procedure of continuous transmission 515
13.5.7 UART reception 517
13.5.8 Reception errors 518
13.5.9 Types and operation of parity 520
13.5.10 Noise filter of receive data 521
13.6 Dedicated Baud Rate Generator 522
13.7 Cautions 528

CHAPTER 14 3-WIRE SERIAL INTERFACE (CSIB) 529
14.1 Features 529
14.2 Configuration 530
14.3 Control Registers 533
14.4 Transfer Data Length Change Function 538
14.5 Interrupt Request Signals 539
14.6 Operation 540
14.6.1 Single transfer mode (master mode, transmission/reception mode) 540
14.6.2 Single transfer mode (master mode, reception mode) 541
14.6.3 Continuous mode (master mode, transmission/reception mode) 542
14.6.4 Continuous mode (master mode, reception mode) 543
14.6.5 Continuous reception mode (error) 544
14.6.6 Continuous mode (slave mode, transmission/reception mode) 545
14.6.7 Continuous mode (slave mode, reception mode) 546
14.6.8 Clock timing 547
14.7 Output Pins 549
14.8 Operation Flow 550
14.9 Prescaler 3 556
14.9.1 Control registers of prescaler 3 556
14.9.2 Generation of count clock 557
14.10 Cautions 558

CHAPTER 15 CAN CONTROLLER 559
15.1 Overview 559
15.1.1 Features 559

15.1.2 Overview of Functions....560
15.1.3 Configuration....561
15.2 CAN Protocol....562
15.2.1 Frame format....562
15.2.2 Frame types....563
15.2.3 Data frame and remote frame....563
15.2.4 Error frame....571
15.2.5 Overload frame....572
15.3 Functions....573
15.3.1 Determining bus priority....573
15.3.2 Bit stuffing....573
15.3.3 Multi masters....572
15.3.4 Multi cast....573
15.3.5 CAN sleep mode/CAN stop mode function....574
15.3.6 Error control function....574
15.3.7 Baud rate control function....581
15.4 Connection with Target System....585
15.5 Internal Registers of CAN controller....586
15.5.1 CAN controller configuration....586
15.5.2 Register access type....588
15.5.3 Register bit configuration....656
15.6 Control Registers....660
15.7 Bit Set/Clear Function....694
15.8 CAN Controller Initialization....696
15.8.1 Initialization of CAN module....696
15.8.2 Initialization of message buffer....696
15.8.3 Redefinition of message buffer....695
15.8.4 Transition from initialization mode to operation mode....697
15.8.5 Resetting error counter CnERC of CAN module....698
15.9 Message Reception....699
15.9.1 Message reception....699
15.9.2 Receive history list function....700
15.9.3 Mask function....702
15.9.4 Multi buffer receive block function....704
15.9.5 Remote frame reception....705
15.10 Message Transmission....706
15.10.1 Message transmission....706
15.10.2 Transmit history list function....708
15.10.3 Automatic block transmission (ABT)....710
15.10.4 Transmission abort process....712
15.10.5 Remote frame transmission....712
15.11 Power Saving Modes....713
15.11.1 CAN sleep mode....713
15.11.2 CAN stop mode....714
15.11.3 Example of using power saving modes....715
15.12 Interrupt Function....716
15.13 Diagnosis Functions and Special Operational Modes....717
15.13.1 Receive-only mode....717

15.13.2 Single-shot mode ... 718
15.13.3 Self-test mode ... 719
15.14 Time Stamp Function ... 720
15.14.1 Time stamp function ... 720
15.15 Baud Rate Settings ... 722
15.15.1 Bit rate setting conditions ... 722
15.15.2 Representative examples of baud rate settings ... 726
15.16 Operation of CAN Controller ... 730

CHAPTER 16 DMA CONTROLLER (DMAC) ... 755
16.1 Features ... 755
16.2 Configuration ... 756
16.3 Registers ... 757
16.4 DMA Bus States ... 766
16.4.1 Types of bus states ... 766
16.4.2 DMAC bus cycle state transition ... 767
16.5 Transfer Targets ... 768
16.6 Transfer Modes ... 768
16.7 Transfer Types ... 769
16.8 DMA Channel Priorities ... 770
16.9 Time Related to DMA Transfer ... 770
16.10 DMA Transfer Start Factors ... 771
16.11 DMA Abort Factors ... 772
16.12 End of DMA Transfer ... 772
16.13 Operation Timing ... 772
16.14 Cautions ... 777

CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION ... 782
17.1 Features ... 782
17.2 Non-Maskable Interrupts ... 787
17.2.1 Non-maskable interrupt request signal ... 787
17.2.2 Operation ... 789
17.2.3 Restore ... 790
17.2.4 NP flag ... 792
17.2.5 Eliminating noise on NMI pin ... 792
17.2.6 Function to detect edge of NMI pin ... 792
17.3 Maskable Interrupts ... 794
17.3.1 Operation ... 794
17.3.2 Restore ... 796
17.3.3 Priorities of maskable interrupts ... 797
17.3.4 Interrupt control registers (xxICn) ... 801
17.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR4, IMR5L) ... 804
17.3.6 In-service priority register (ISPR) ... 806
17.3.7 ID flag ... 807
17.3.8 Watchdog timer mode register 2 (WDTM2) ... 807
17.3.9 Eliminating noise on INTP0 to INTP7 pins ... 808
17.4 Software Exceptions ... 818
17.4.1 Operation ... 818

17.4.2 Restore 819
17.4.3 EP flag 820
17.5 Exception Trap 821
17.5.1 Illegal opcode definition 821
17.5.2 Debug trap 823
17.6 Interrupt Acknowledgment Time of CPU 825
17.7 Periods in Which Interrupts Are Not Acknowledged by CPU 826

CHAPTER 18 KEY INTERRUPT FUNCTION 827
18.1 Function 827
18.2 Control Register 828

CHAPTER 19 STANDBY FUNCTION 829
19.1 Overview 829
19.2 HALT Mode 834
19.2.1 Setting and operation status 834
19.2.2 Releasing HALT mode 834
19.3 IDLE1 Mode 836
19.3.1 Setting and operation status 836
19.3.2 Releasing IDLE1 mode 836
19.4 IDLE2 Mode 838
19.4.1 Setting and operation status 838
19.4.2 Releasing IDLE2 mode 838
19.4.3 Securing setup time after release of IDLE2 mode 840
19.5 Software STOP Mode 841
19.5.1 Setting and operation status 841
19.5.2 Releasing software STOP mode 842
19.5.3 Securing setup time after release of software STOP mode 844
19.6 Subclock Operation Mode 845
19.6.1 Setting and operation status 845
19.6.2 Releasing subclock operation mode 845
19.7 Sub-IDLE Mode 847
19.7.1 Setting and operation status 847
19.7.2 Releasing sub-IDLE mode 847
19.8 Control Registers 849

CHAPTER 20 RESET FUNCTION 852
20.1 Overview 852
20.2 Register to Check Reset Source 853
20.3 Operation 854
20.3.1 Reset operation by $\overline{\text{RESET}}$ pin 854
20.3.2 Reset operation by WDT2RES signal 856

CHAPTER 21 CLOCK MONITOR 859
21.1 Function of Clock Monitor 859
21.2 Configuration of Clock Monitor 859
21.3 Register Controlling Clock Monitor 860
21.4 Operation of Clock Monitor 861

CHAPTER 22 POWER-ON CLEAR CIRCUIT 864
22.1 Functions of Power-on Clear Circuit 864
22.2 Configuration of Power-on Clear Circuit 865
22.3 Operation of Power-on Clear Circuit 865

CHAPTER 23 LOW-VOLTAGE DETECTOR 866
23.1 Functions of Low-Voltage Detector 866
23.2 Configuration of Low-Voltage Detector 866
23.3 Registers Controlling Low-Voltage Detector 867
23.4 Operation of Low-Voltage Detector 870
23.4.1 To use for internal reset signal 870
23.4.2 To use for interrupt 871
23.5 RAM Retention Voltage Detection Operation 872

CHAPTER 24 REGULATOR 873
24.1 Overview 873
24.2 Operation 873

CHAPTER 25 FLASH MEMORY 875
25.1 Features 875
25.1.1 Erasure unit 876
25.1.2 Functional outline 877
25.2 Writing with Flash programmer 879
25.3 Programming Environment 879
25.4 Communication Mode 880
25.5 Pin Connection 885
25.5.1 FLMD0 pin 885
25.5.2 FLMD1 pin 886
25.5.3 Serial interface pins 886
25.5.4 $\overline{\text{RESET}}$ pin 888
25.5.5 Port pins (including NMI) 888
25.5.6 Other signal pins 888
25.5.7 Power supply 888
25.6 Programming Method 889
25.6.1 Flash memory control 889
25.6.2 Selecting communication mode 890
25.6.3 Communication commands 891
25.7 Rewriting by Self-Programming 892
25.7.1 Overview 892
25.7.2 Features 893
25.7.3 Standard self-programming flow 894
25.7.4 Flash functions 895
25.7.5 Pin processing 895
25.7.6 Internal resources used 896

CHAPTER 26 OPTION FUNCTION 897
26.1 Mask Options 897

CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT) 899
27.1 Functional Outline 899
27.1.1 Type of on-chip debug unit 899
27.1.2 Debug functions 899
27.2 Connection Circuit Example 901
27.3 Interface Signals 901
27.4 Register 903
27.5 Operation 904
27.6 ROM Security Function 906
27.6.1 Security ID 906
27.6.2 Setting 907
27.7 Connection to N-Wire Emulator 909
27.7.1 KEL connector 909
27.8 Cautions 913

APPENDIX A REGISTER INDEX 914

APPENDIX B INSTRUCTION SET LIST 927
B.1 Conventions 927
B.2 Instruction Set (in Alphabetical Order) 930
B.3 Description of Operating Precautions 937

APPENDIX C REVISION HISTORY 942

CHAPTER 1 INTRODUCTION

The V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 are products of NEC Electronics' V850 Series of single-chip microcontrollers for real-time control.

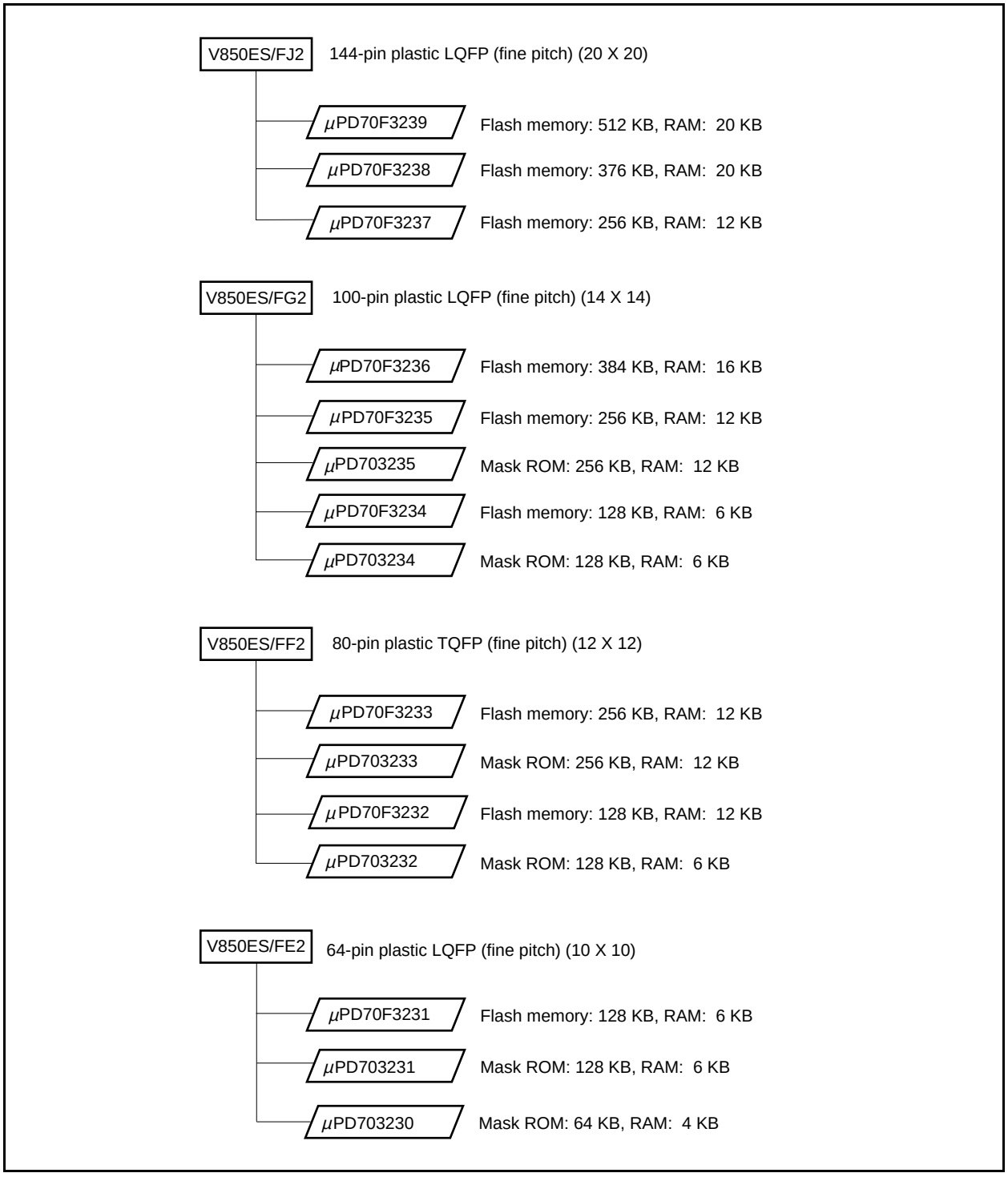
1.1 General

The V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 are 32-bit single-chip microcontroller that include the V850ES CPU core and integrate peripheral functions such as timers/counters, serial interfaces, and an A/D converter. These microcontrollers also incorporate a CAN (Controller Area Network) as an automotive LAN.

In addition to highly real-time responsive, 1-clock-pitch basic instructions, this microcontroller have instructions ideal for digital servo applications, such as multiplication instructions using a hardware multiplier, sum-of-products operation instructions, and bit manipulation instructions. This microcontroller can also realize a real-time control system that is highly cost effective and can be used in automotive instrumentation fields.

V850ES/FE2, V850ES/FF2, V850ES/FG2, are models of the V850ES/FJ2 with reduced I/O, timer/counter, and serial interface functions (See **1.2 Product Development of V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2** and **Table 1-1. Functional Outline of V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2**).

★ 1.2 Product Development of V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2



1.3 Features

- ○ Number of instructions: 83
- ○ Minimum instruction execution time: 50 ns (main clock (f_{XX}) = 20 MHz)
- ○ General-purpose registers: 32 bits × 32
- ○ Power-on clear function
- ○ Low-voltage detection function
- ○ Ring-OSC: 200 kHz (TYP.)
- ○ Internal memory — RAM: 4/6/8/12/16/20 KB (see **Table 1-1**)
  Flash memory: 64/128/256/376/384/512KB (see **Table 1-1**)
- ○ Interrupts/exceptions
  Non-maskable interrupts (see **Table 1-1**)
  Maskable interrupts (see **Table 1-1**)
  Software exceptions： 2 sources
  Exception trap： 1 sources
- ○ I/O lines — I/O ports: 128
- ○ Timer/counters
  16-bit interval timer M (TMM): 1 ch
  16-bit timer/event counter P (TMP): 4 ch
  16-bit timer/event counter Q (TMQ): 1 to 3 ch (see **Table 1-1**)
- ○ Watch timer: 1 ch
- ○ Watchdog timer 2: 1 ch
- ○ Serial interface (SIO)
  Asynchronous serial interface A (UART): 2 to 4 (see **Table 1-1**)
  3-wire variable-length serial interface B (CSIB): 2 to 3ch (see **Table 1-1**)
- ★ ○ CAN controller: 1 to 4 ch (see **Table 1-1**)
- ○ A/D converter — 10-bit resolution: 10 to 24 ch (see **Table 1-1**)
- ○ Clock generator — Main clock/subclock operation
  CPU clock in seven steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})
  Clock-through mode/PLL mode selectable
  Internal oscillator: 200 kHz（TYP.）
- ○ Power save function — HALT/IDLE1/IDLE2/software STOP/subclock/sub-IDLE modes
- ○ Package — 64-pin plastic LQFP (fine pitch) (10 × 10)
  80-pin plastic TQFP (fine pitch) (12 × 12)
  100-pin plastic LQFP (fine pitch) (14 × 14)
  144-pin plastic LQFP (fine pitch) (20 × 20)

★ **Table 1-1. Functional Outline of V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2**

Series name		V850ES/FE2		V850ES/FF2			V850ES/FG2			V850ES/FJ2		
Part number		μPD70(F)3230	μPD70(F)3231	μPD703232	μPD70F3232	μPD70(F)3233	μPD70(F)3234	μPD70(F)3235	μPD70(F)3236	μPD70F)237	μPD70(3238	μPD70F3239
Internal memory	Flash (bytes)	64K	128 K		128 K	256 K	128K	256K	384K	256K	376K	512K
	Mask ROM (bytes)	64K	128K	128K		256K	128K	256K	-	-	-	-
	RAM (bytes)	4K	6K	6K	12K	12K	6K	12K	16K	12K	20K	20K
DMA		None		None			Provided			Provided		
Operating clock	Main (internal)	20 MHz max.		20 MHz max.			20 MHz max.			20 MHz max.		
	Ring-OSC	200 kHz typ.		200 kHz typ.			200 kHz typ.			200 kHz typ.		
	Subclock	RC or crystal		RC or crystal			RC or crystal			RC or crystal		
I/O ports		51		67			84			128		
A/D converter		10 bits × 10 ch		10 bits × 12 ch			10 bits × 16 ch			10 bits × 24 ch		
Timers	TMQ	1 ch		1 ch			2 ch			3 ch		
	TMP	4 ch		4 ch			4 ch			4 ch		
	TMM	1 ch		1 ch			1 ch			1 ch		
	WDT2	1 ch		1 ch			1 ch			1 ch		
	Watch	1 ch		1 ch			1 ch			1 ch		
Serial interfaces	CSI	2 ch		2 ch			2 ch			3 ch		
	UART	2 ch		2 ch			3 ch			3 ch	4 ch	
	CAN	1 ch		1 ch			2 ch			2 ch	4 ch	
Interrupts	External	8 ch		8 ch			11 ch			15 ch		
	Internal	35 ch		35 ch			50 ch			57 ch	67 ch	
	NMI	1 ch		1 ch			1 ch			1 ch		
Other functions	Key return input	8 ch		8 ch			8 ch			8 ch		
	Clock monitor function	Provided		Provided			Provided			Provided		
	POC/LVI function	Provided		Provided			Provided			Provided		
	Clock output function	Provided		Provided			Provided			Provided		
	PCL output function	Provided		Provided			Provided			Provided		
	On-chip debug function	Provided (Note)		Provided (Note)			Provided (Note)			Provided (Note)		
External memory interface		None		None			None			Provided		
Operating voltage		3.5 V to 5.5 V		3.5 V to 5.5 V			3.5 V to 5.5 V			3.5 V to 5.5 V		
Package		64-pin LQFP		80-pin TQFP			100-pin LQFP			144-pin LQFP		

Note On Flash version only (μPD70F323x)

★ 1.4 Ordering Information

- V850ES/FJ2

Part Number **Note**	Package	On-Chip Flash Memory	CAN Buffer	Quality Grade	Remark
μPD70F3237M1GJ(A)-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	256 KB	32 buffer/ch	Special	Without power-on clear function
μPD70F3237M1GJ(A1)-UEN					
μPD70F3237M1GJ(A2)-UEN					
μPD70F3237M2GJ(A)-UEN					With power-on clear function
μPD70F3237M2GJ(A1)-UEN					
μPD70F3237M2GJ(A2)-UEN					
μPD70F3238M1GJ(A)-UEN		376 KB			Without power-on clear function
μPD70F3238M1GJ(A1)-UEN					
μPD70F3238M1GJ(A2)-UEN					
μPD70F3238M2GJ(A)-UEN					With power-on clear function
μPD70F3238M2GJ(A1)-UEN					
μPD70F3238M2GJ(A2)-UEN					
μPD70F3239M1GJ(A)-UEN		512 KB			Without power-on clear function
μPD70F3239M1GJ(A1)-UEN					
μPD70F3239M1GJ(A2)-UEN					
μPD70F3239M2GJ(A)-UEN					With power-on clear function
μPD70F3239M2GJ(A1)-UEN					
μPD70F3239M2GJ(A2)-UEN					

Note: The operating ambient temperature of each quality grades is as follows.
(A) : −40 to +85℃, (A1) : −40 to +110℃, (A2) : −40 to +125℃

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

- V850ES/FG2

Part Number	Package	Internal Memory	Number of CAN Buffers	Quality Grade	Remark
μPD703234GC(A)-xxx-8EA	100-pin plastic LQFP (fine pitch) (14 × 14)	128 KB (Mask ROM)	32 buffers/ch	Special[Note]	–
μPD703234GC(A1)-xxx-8EA					
μPD703234GC(A2)-xxx-8EA					
μPD703235GC(A)-xxx-8EA		256 KB (Mask ROM)			
μPD703235GC(A1)-xxx-8EA					
μPD703235GC(A2)-xxx-8EA					
μPD70F3234M1GC(A)-8EA		128 KB (Flash Memory)			Without power-on clear function
μPD70F3234M1GC(A1)-8EA					
μPD70F3234M1GC(A2)-8EA					
μPD70F3234M2GC(A)-8EA					With power-on clear function
μPD70F3234M2GC(A1)-8EA					
μPD70F3234M2GC(A2)-8EA					
μPD70F3235M1GC(A)-8EA		256 KB (Flash Memory)			Without power-on clear function
μPD70F3235M1GC(A1)-8EA					
μPD70F3235M1GC(A2)-8EA					
μPD70F3235M2GC(A)-8EA					With power-on clear function
μPD70F3235M2GC(A1)-8EA					
μPD70F3235M2GC(A2)-8EA					
μPD70F3236M1GC(A)-8EA		384 KB (Flash Memory)			Without power-on clear function
μPD70F3236M1GC(A1)-8EA					
μPD70F3236M1GC(A2)-8EA					
μPD70F3236M2GC(A)-8EA					With power-on clear function
μPD70F3236M2GC(A1)-8EA					
μPD70F3236M2GC(A2)-8EA					

Note: The operating ambient temperature of each quality grades is as follows.
(A) : −40 to +85℃, (A1) : −40 to +110℃, (A2) : −40 to +125℃
Remark xxx is ROM code number.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

- V850ES/FF2

Part Number	Package	Internal Memory	Number of CAN Buffers	Quality Grade	Remark
μPD703232GK(A)-xxx-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	128 KB (Mask ROM)	32 buffers/ch	Special[Note]	–
μPD703232GK(A1)-xxx-9EU					
μPD703232GK(A2)-xxx-9EU					
μPD703233GK(A)-xxx-9EU		256 KB (Mask ROM)			
μPD703233GK(A1)-xxx-9EU					
μPD703233GK(A2)-xxx-9EU					
μPD70F3232M1GK(A)-9EU		128 KB (Flash Memory)			Without power-on clear function
μPD70F3232M1GK(A1)-9EU					
μPD70F3232M1GK(A2)-9EU					
μPD70F3232M2GK(A)-9EU					With power-on clear function
μPD70F3232M2GK(A1)-9EU					
μPD70F3232M2GK(A2)-9EU					
μPD70F3233M1GK(A)-9EU		256 KB (Flash Memory)			Without power-on clear function
μPD70F3233M1GK(A1)-9EU					
μPD70F3233M1GK(A2)-9EU					
μPD70F3233M2GK(A)-9EU					With power-on clear function
μPD70F3233M2GK(A1)-9EU					
μPD70F3233M2GK(A2)-9EU					

Note: The operating ambient temperature of each quality grades is as follows.
(A) : −40 to +85℃, (A1) : −40 to +110℃, (A2) : −40 to +125℃

Remark xxx is ROM code number.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

- V850ES/FE2

Part Number	Package	Internal Memory	Number of CAN Buffers	Quality Grade	Remark
μPD703230GB(A)-xxx-8EA	64-pin plastic LQFP (fine pitch) (10 × 10)	64 KB (Mask ROM)	32 buffers/ch	Special[Note]	–
μPD703230GB(A1)-xxx-8EA					
μPD703230GB(A2)-xxx-8EA					
μPD703231GB(A)-xxx-8EA		128KB (Mask ROM)			
μPD703231GB(A1)-xxx-8EA					
μPD703231GB(A2)-xxx-8EA					
μPD70F3231M1GB(A)-8EA		128 KB (Flash Memory)			Without power-on clear function
μPD70F3231M1GB(A1)-8EA					
μPD70F3231M1GB(A2)-8EA					
μPD70F3231M2GB(A)-8EA					With power-on clear function
μPD70F3231M2GB(A1)-8EA					
μPD70F3231M2GB(A2)-8EA					

Note: The operating ambient temperature of each quality grades is as follows.
(A) : −40 to +85℃, (A1) : −40 to +110℃, (A2) : −40 to +125℃

Remark xxx is ROM code number.

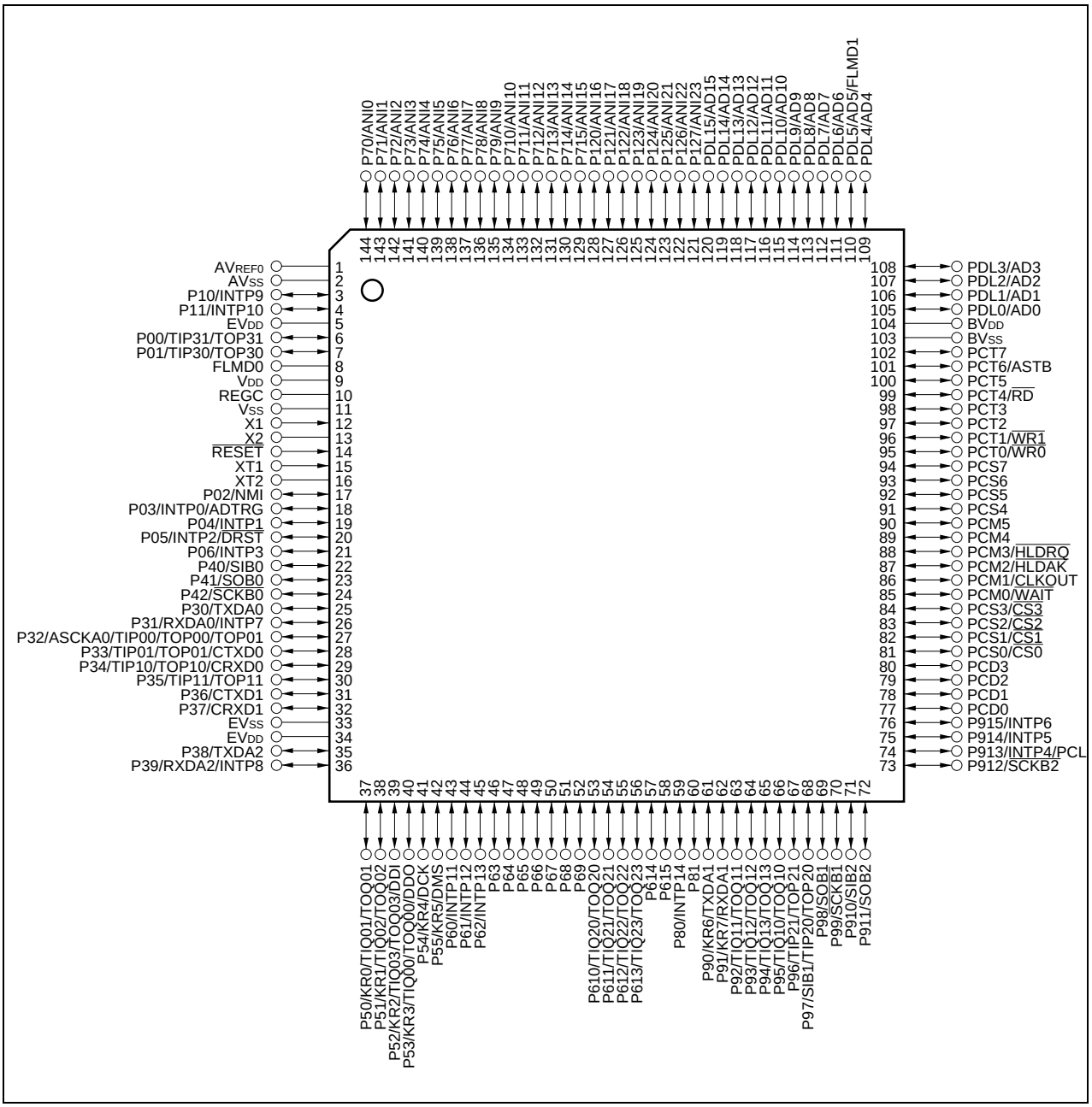
Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

1.5 Applications

Automotive body electrical systems (CAN controller equipped general-purpose products)

★ 1.6 Pin Configuration (Top View)

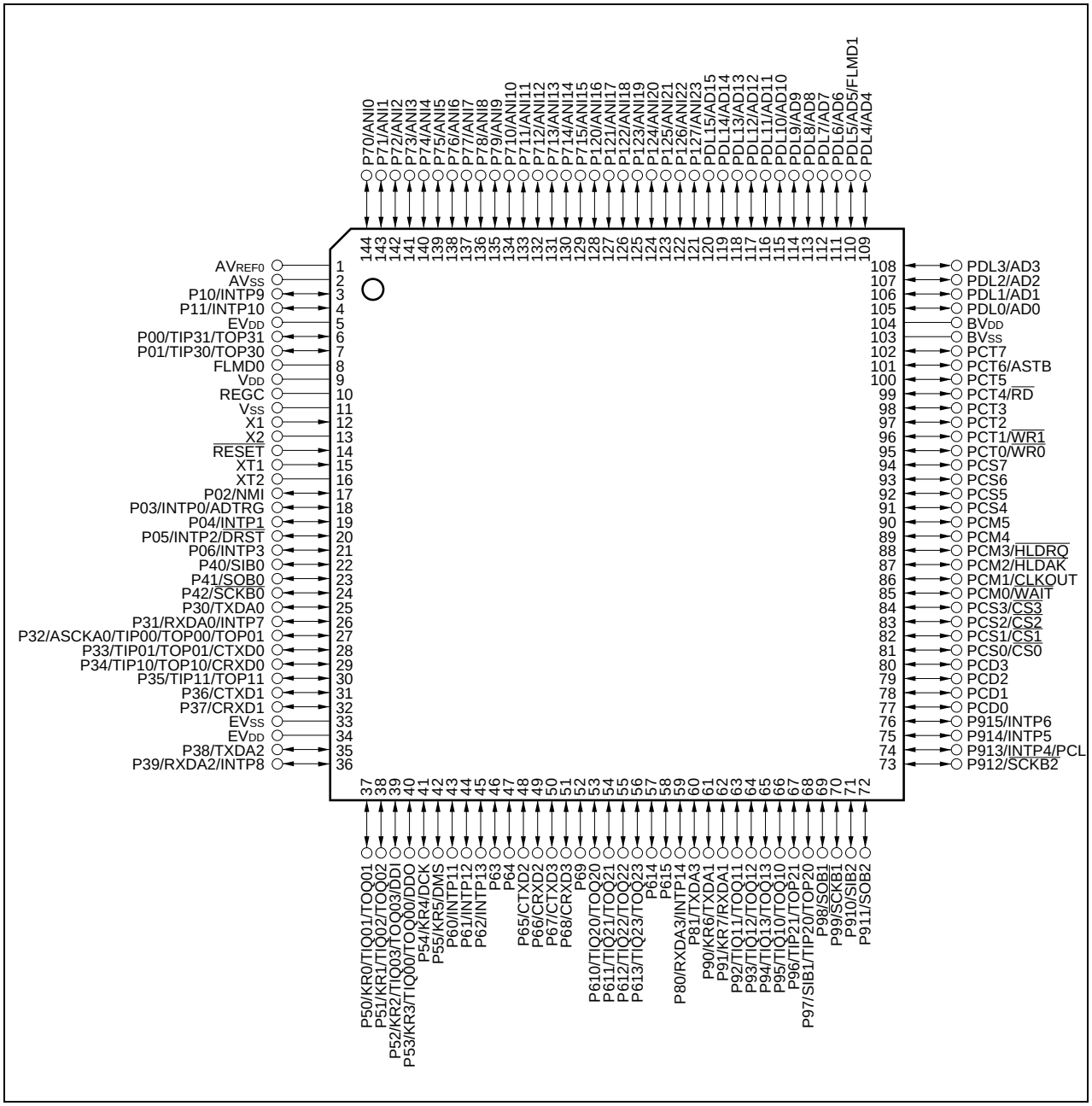
- V850ES/FJ2 (μPD70F3237)
  144-pin plastic LQFP (fine pitch) (20 × 20)



Notes 1 IC: Connect to VSS directly（mask ROM products only）
FLMD0: Connect to VSS in the normal operation mode.（Flash memory versions only）
FLMD1: Flash memory versions only

Notes 2 REGC can be connect to VSS via capacitor of 4.7uF

- V850ES/FJ2 (μPD70F3238, μPD70F3239)
  144-pin plastic LQFP (fine pitch) (20 × 20)



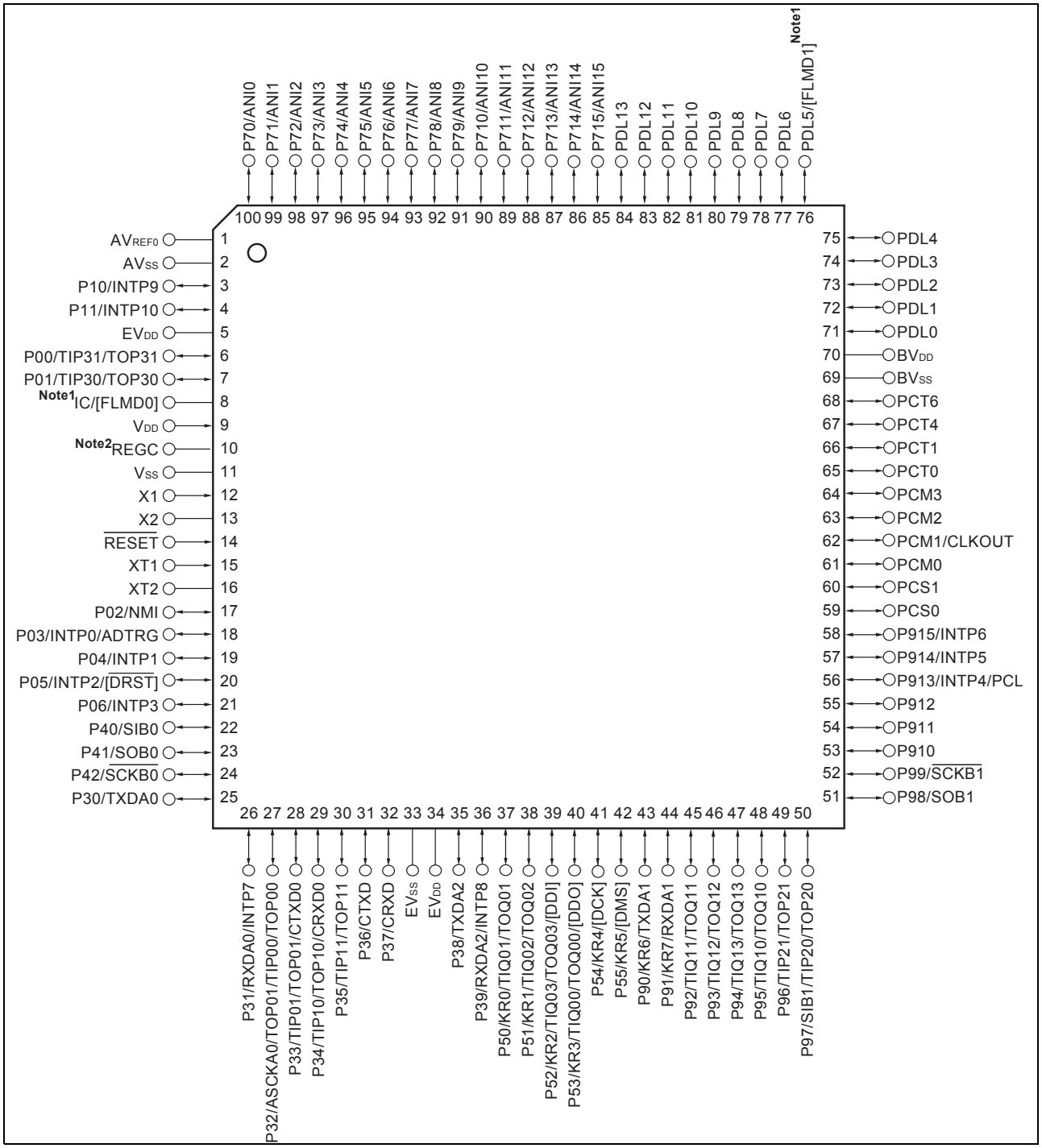
Notes 1 IC: Connect to VSS directly

FLMD0: Connect to VSS in the normal operation mode.（Flash memory versions only）

FLMD1: Flash memory versions only

Notes 2 REGC can be connect to VSS via capacitor of 4.7uF

- V850ES/FG2 (μPD70F3234,μPD70F3235,μPD70F3236)
  100-pin plastic LQFP (fine pitch) (14 × 14)

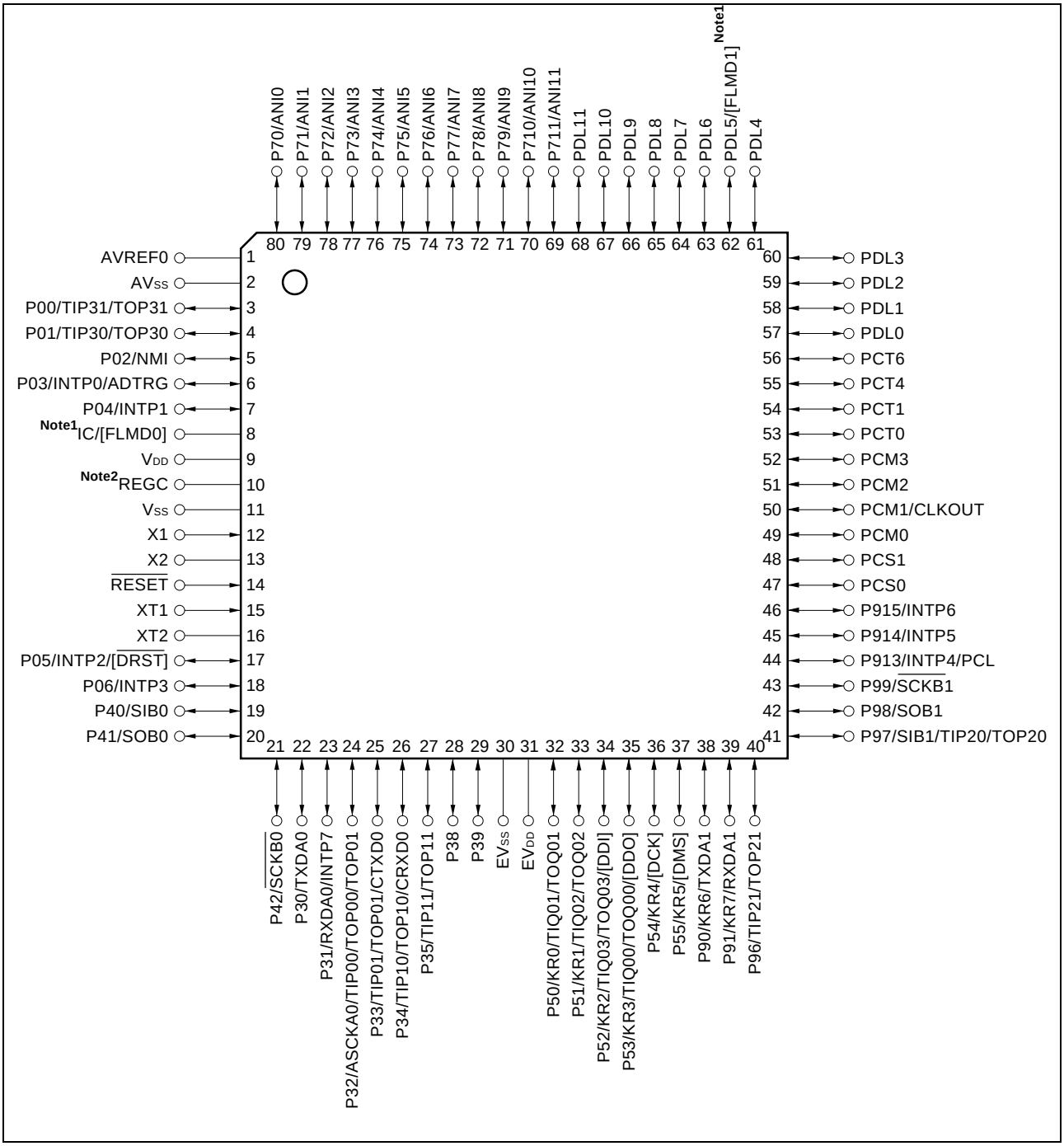


Notes 1 IC: Connect to VSS directly
FLMD0: Connect to VSS in the normal operation mode.（Flash memory versions only）
FLMD1: Flash memory versions only

Notes 2 REGC can be connect to VSS via capacitor of 4.7uF

Remark Pins in brackets are valid only in μ PD70F3234, 70F3235, 70F3236

- V850ES/FF2 (μPD70F3232,μPD70F3233)
  80-pin plastic TQFP (fine pitch) (12 × 12)

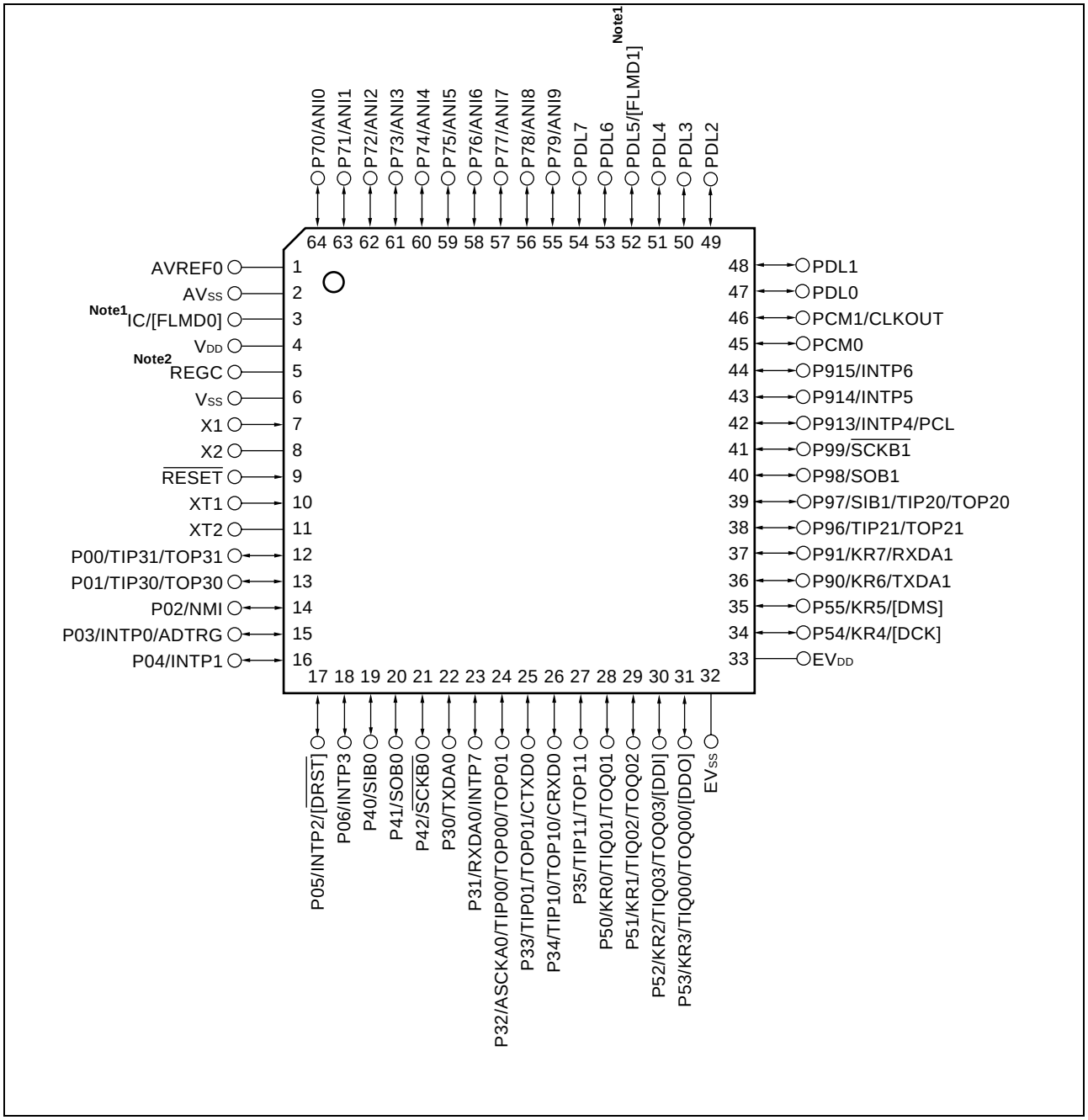


Notes 1 IC: Connect to VSS directly
FLMD0: Connect to VSS in the normal operation mode.（Flash memory versions only）
FLMD1: Flash memory versions only

Notes 2 REGC can be connect to VSS via capacitor of 4.7uF

Remark Pins in brackets are only valid for μ PD70F3232, μ 70F3233

- V850ES/FE2 (μPD703230, μPD70F3231)
  64-pin plastic LQFP (fine pitch) (10 × 10)



Note 1 IC: Connect to VSS directly
FLMD0: Connect to VSS in the normal operation mode.（Flash memory versions only）
FLMD1: Flash memory versions only

Note 2 REGC can be connect to VSS via capacitor of 4.7uF

Remark Pins in brackets are only valid for μ PD70F3231

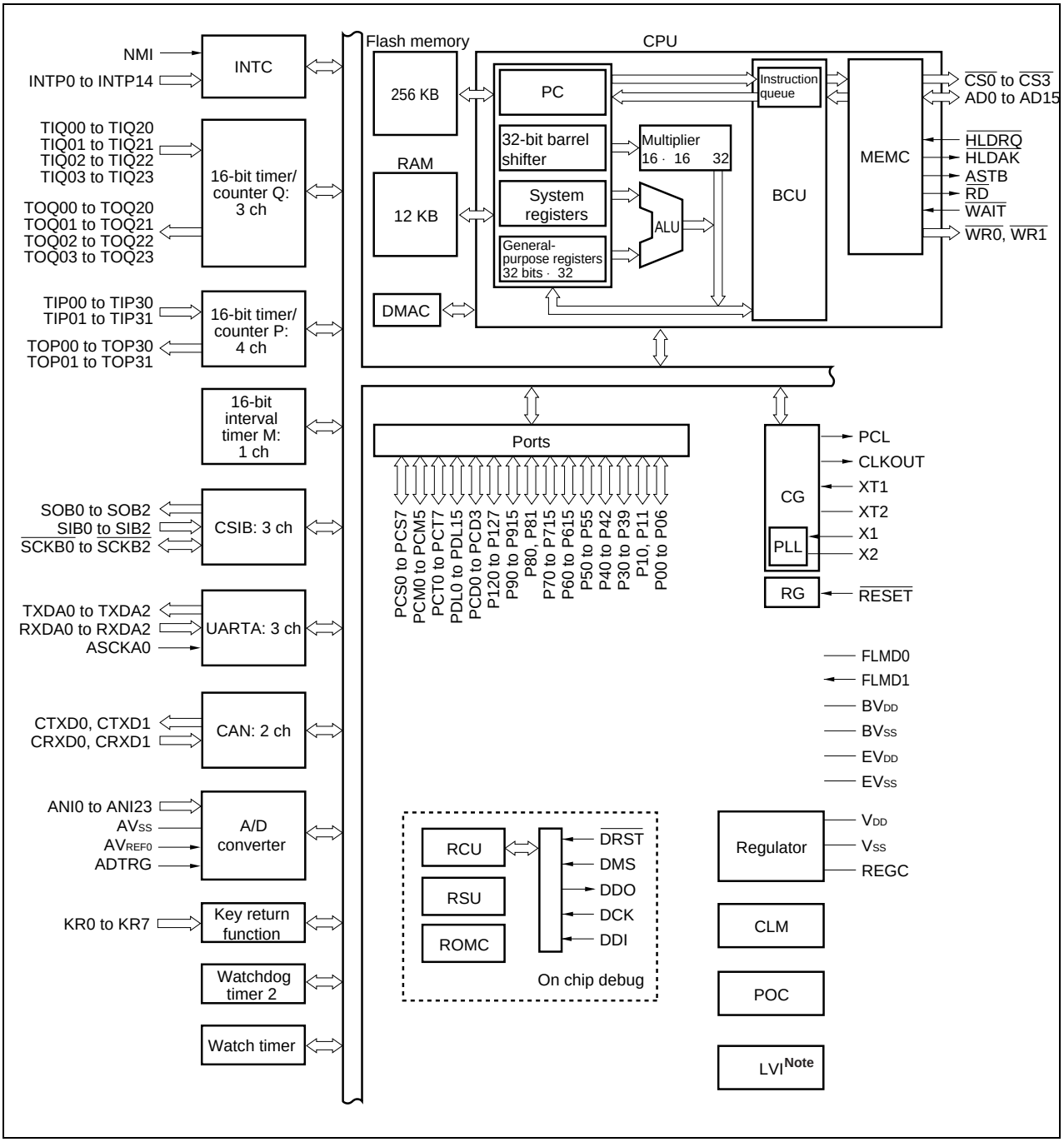
Pin identification

AD0 to AD15:	Address/data bus
ADTRG:	A/D trigger input
ANI0 to ANI23:	Analog input
ASCKA0:	Asynchronous serial clock
ASTB:	Address strobe
AV_{REF0}:	Analog reference voltage
AV_{SS}:	Analog V_{SS}
BV_{DD}:	Power supply for bus interface
BV_{SS}:	Ground for bus interface
CLKOUT:	Clock output
CRXD0 to CRXD3:	CAN receive data
$\overline{CS0}$ to $\overline{CS3}$:	Chip select
CTXD0 to CTXD3:	CAN transmit data
DCK:	Debug clock
DDI:	Debug data input
DDO:	Debug data output
DMS:	Debug mode select
$\overline{DRST}$:	Debug reset
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
FLMD0, FLMD1:	Flash programming mode
$\overline{HLDAK}$:	Hold acknowledge
$\overline{HLDRQ}$:	Hold request
INTP0 to INTP14:	Interrupt request from peripherals
KR0 to KR7:	Key return
NMI:	Non-maskable interrupt request
P00 to P06:	Port 0
P10, P11:	Port 1
P30 to P39:	Port 3
P40 to P42:	Port 4
P50 to P55:	Port 5
P60 to P615:	Port 6
P70 to P715:	Port 7
P80, P81:	Port 8
P90 to P915:	Port 9
P120 to P127:	Port 12
PCD0 to PCD3:	Port CD
PCL:	Programmable clock output
PCM0 to PCM5:	Port CM
PCS0 to PCS7:	Port CS
PCT0 to PCT7:	Port CT
PDL0 to PDL15:	Port DL
$\overline{RD}$:	Read strobe
REGC:	Regulator control
$\overline{RESET}$:	Reset
RXDA0 to RXDA3:	Receive data
$\overline{SCKB0}$ to $\overline{SCKB2}$:	Serial clock
SIB0 to SIB2:	Serial input
SOB0 to SOB2:	Serial output
TIP00, TIP01,	Timer input
TIP10, TIP11,	Timer input
TIP20, TIP21,	Timer input
TIP30, TIP31,	Timer input
TIQ00 to TIQ03,	Timer input
TIQ10 to TIQ13,	Timer input
TIQ20 to TIQ23:	Timer input
TOP00, TOP01,	Timer output
TOP10, TOP11,	Timer output
TOP20, TOP21,	Timer output
TOP30, TOP31,	Timer output
TOQ01 to TOQ03,	Timer output
TOQ11 to TOQ13,	Timer output
TOQ20 to TOQ23:	Timer output
TXDA0 to TXDA3:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
$\overline{WAIT}$:	Wait
$\overline{WR0}$:	Write strobe low level data
$\overline{WR1}$:	Write strobe high level data
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.7 Function Block Configuration

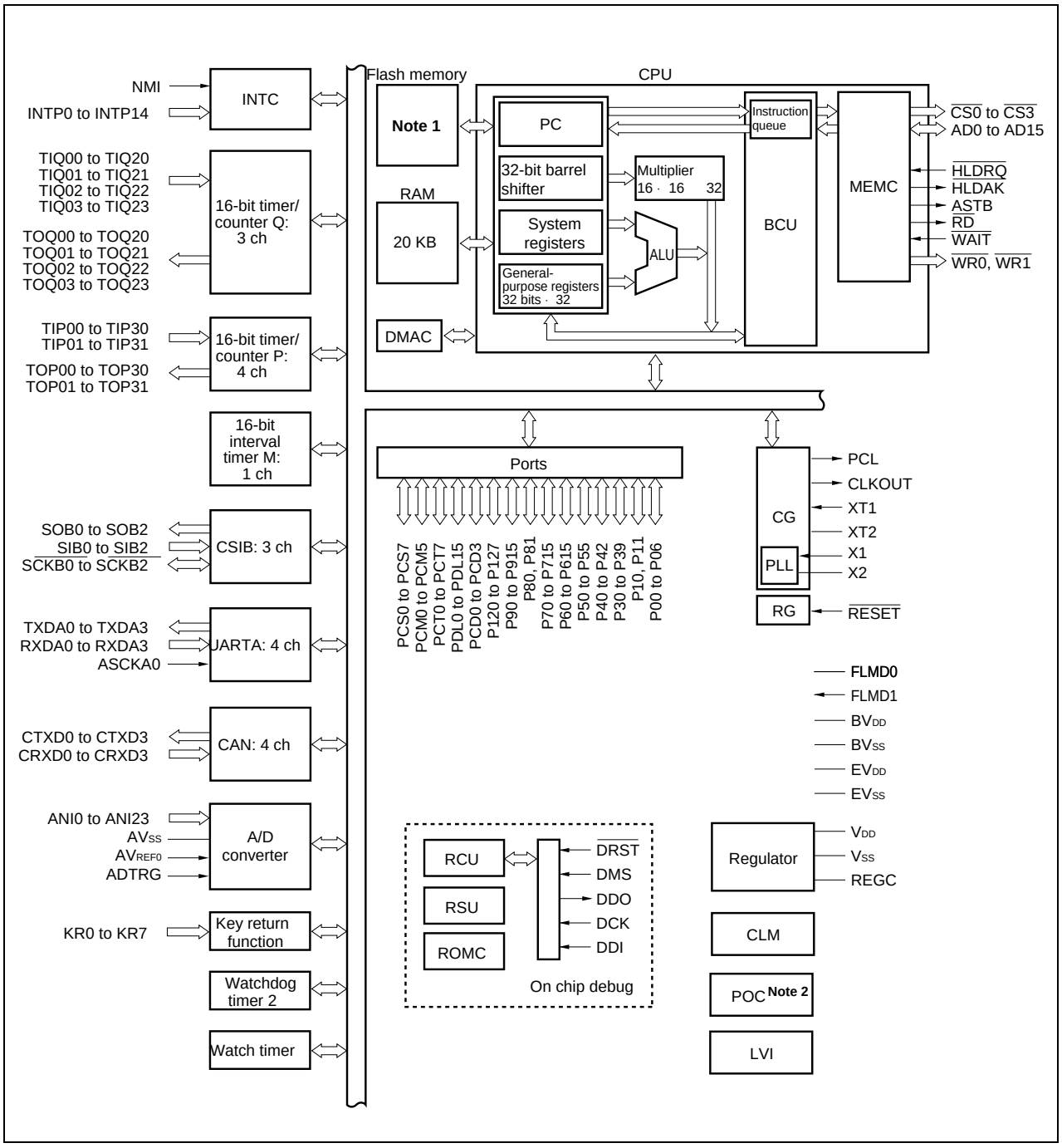
★ 1.7.1 Internal block diagram

- V850ES/FJ2 (μPD70F3237)



Note: only POC version

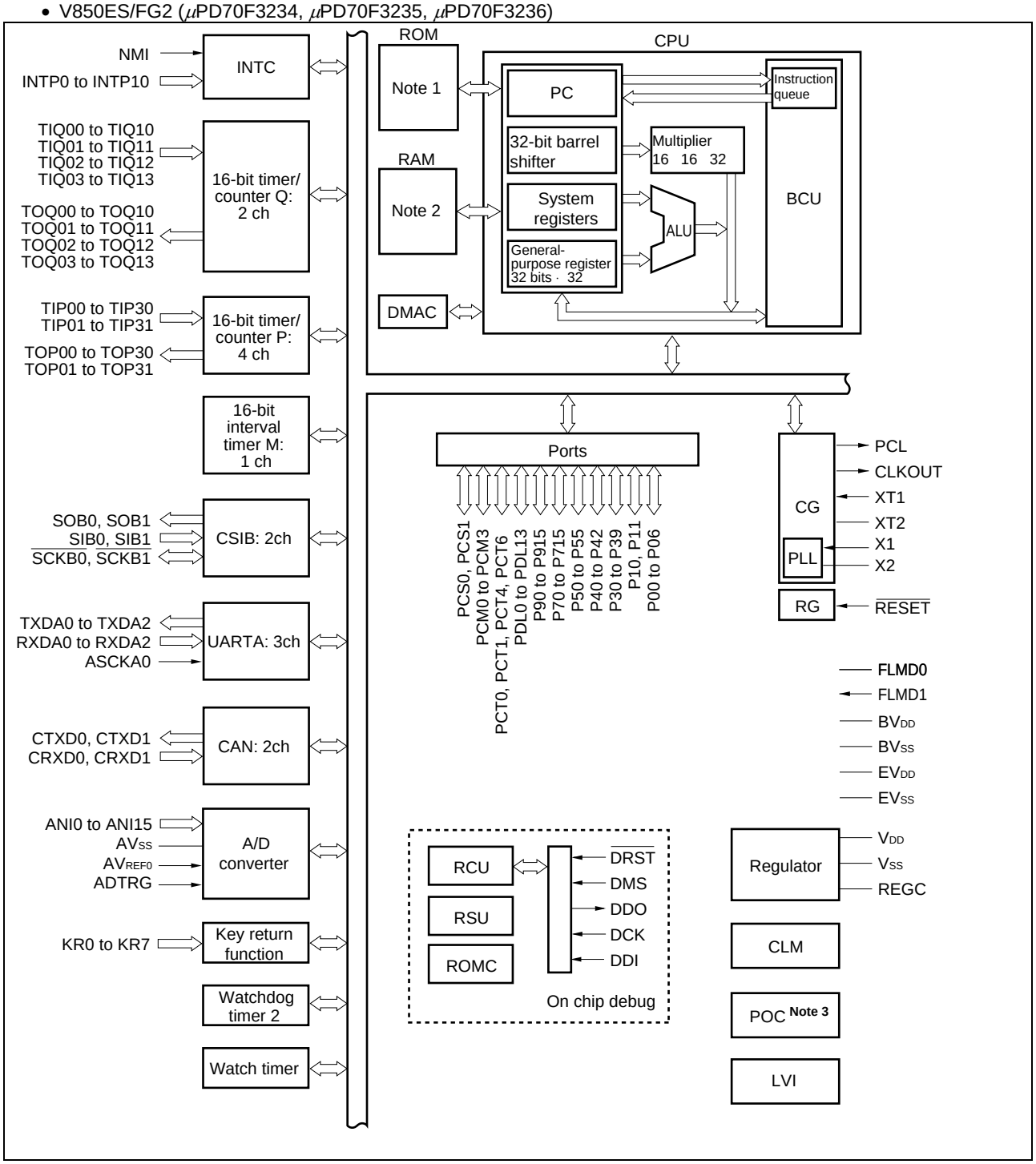
- V850ES/FJ2 (μPD70F3238μPD70F3239)



Notes: 1 376 KB/512 KB (Flash memory see Table1-1)

2. POC version

- V850ES/FG2 (μPD70F3234, μPD70F3235, μPD70F3236)

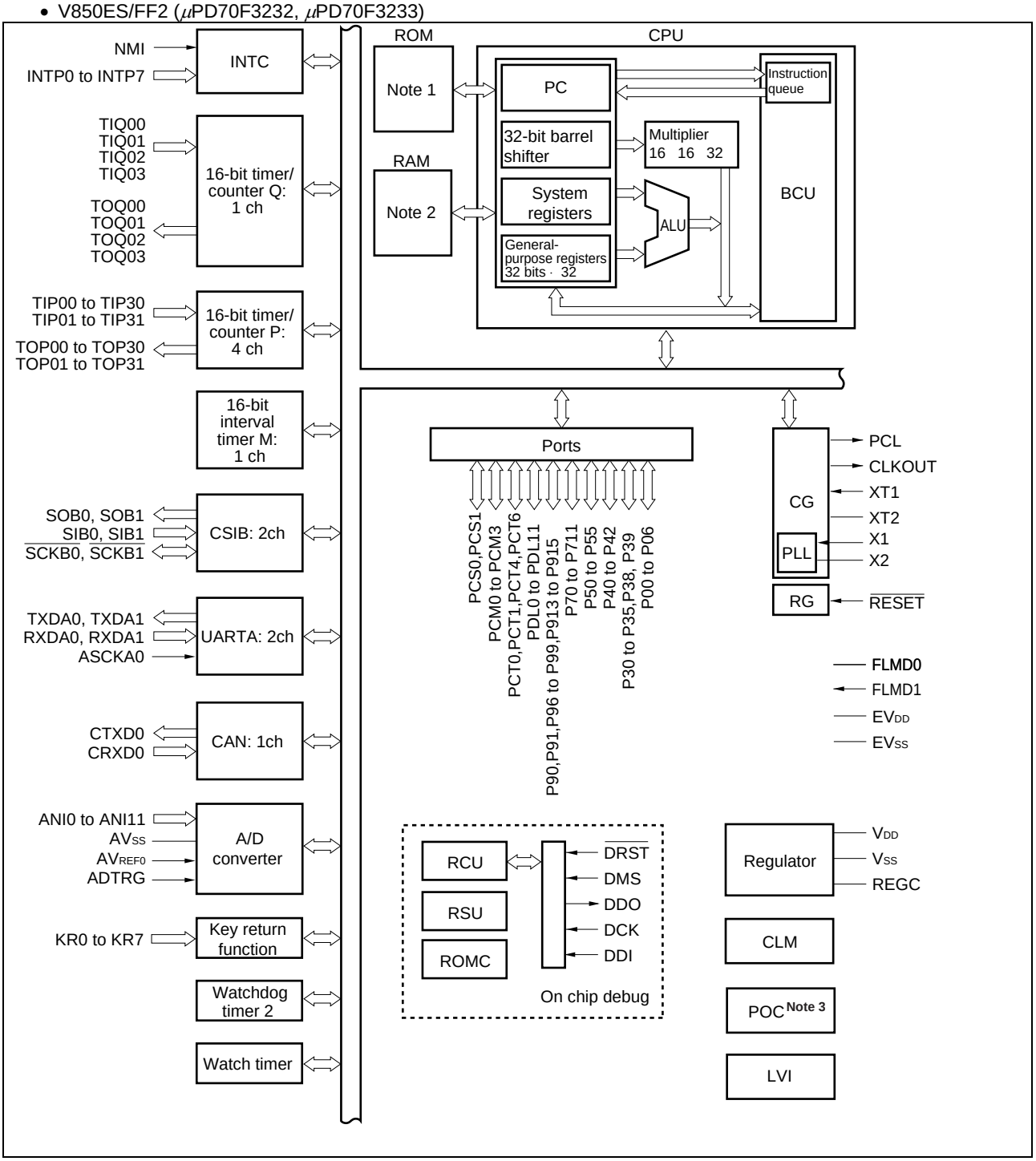


Notes: 1 **128/256/384KB (Flash memory see Table1-1)**
128/256KB (Mask ROM see Table1-1)

2 **6/12/16KB (see Table1-1)**

3 **POC version only**

- V850ES/FF2 (μPD70F3232, μPD70F3233)



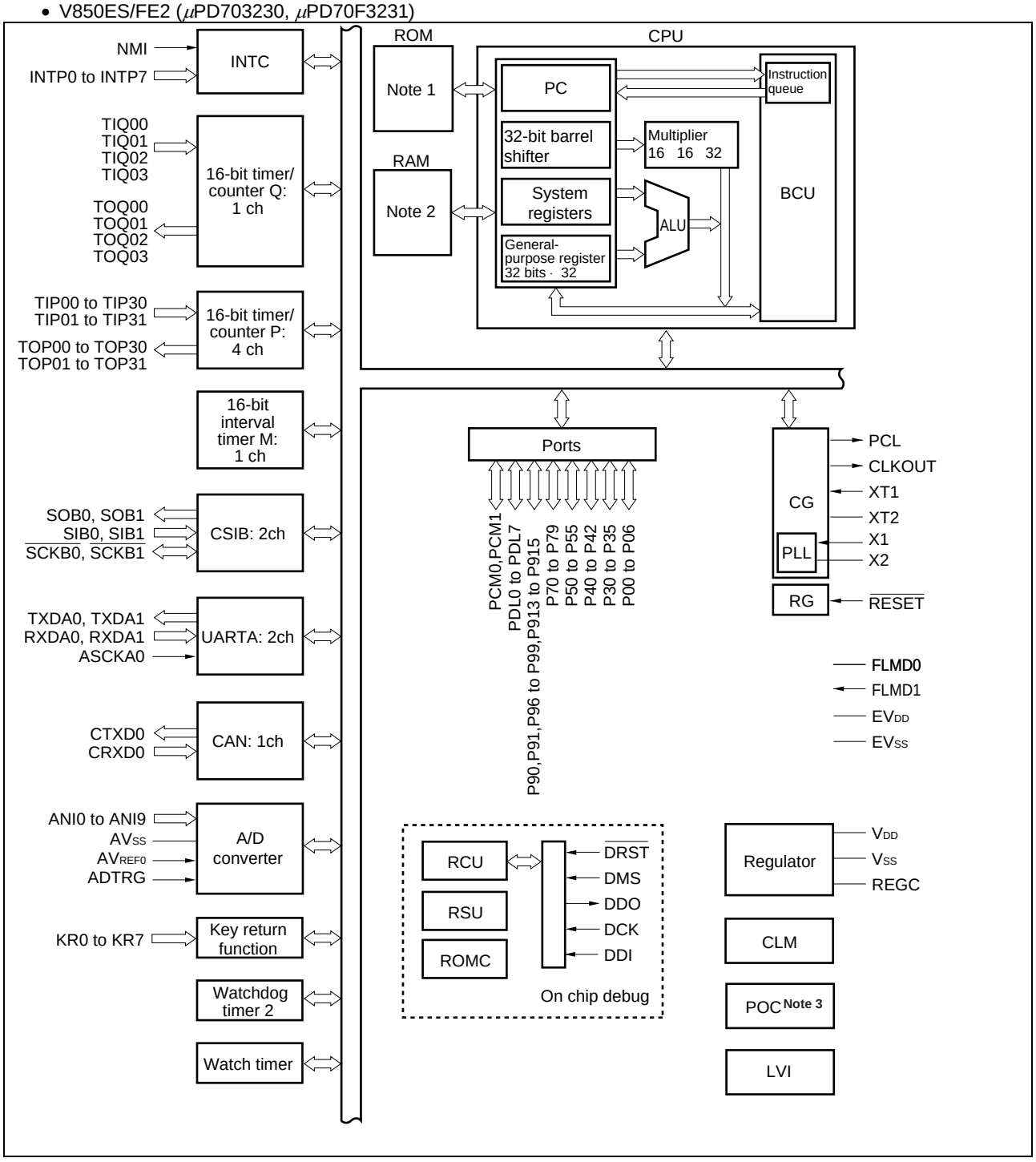
Notes: 1 **128/256 (Flash memory see Table1-1)**

128/256KB (Mask ROM see Table1-1)

2 **6/12KB (see Table1-1)**

3 **POC version only**

- V850ES/FE2 (μPD703230, μPD70F3231)



Notes: 1 128KB (Flash memory see Table1-1)

64/128KB (Mask ROM see Table1-1)

2 4/6KB (see Table1-1)

3 POC version only

1.7.2 Internal units

(1) CPU

The CPU can execute almost all instruction processing, such as address calculation, arithmetic and logic operations, and data transfer, in one clock under control of a five-stage pipeline.

Dedicated hardware units such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) are provided to speed up complicated instruction processing.

(2) External memory control unit (MEMC)

This unit starts necessary external bus cycles based on the physical addresses obtained by the CPU. If the CPU does not request the start of a bus cycle when it fetches an instruction from an external memory area, this unit generates a prefetch address and prefetches an instruction code. The prefetched instruction code is sent to an internal instruction queue.

(3) ROM

This is a flash memory of 512/384/376/256/128/64 KB mapped to addresses 0000000H-007FFFFH/0000000H-005FFFFH/0000000H-005DFFFH/0000000H-003FFFFH/0000000H-001FFFFH/0000000H-000FFFFH. The CPU can access this memory in one clock when it fetches an instruction.

(4) RAM

This is a RAM of 20/16/12/6/4 KB mapped to addresses 3FFA000H-3FFEFFFH/3FFB000H-3FFEFFFH/3FFC000H-3FFEFFFH/3FFD8000H-3FFEFFFH/3FFE000H-3FFEFFFH. The CPU can access this RAM in one clock when it accesses data.

(5) Interrupt controller (INTC)

The interrupt controller processes interrupt requests (NMI and INTP0 up to INTP14 refer to **Table 1-1**) from the on-chip peripheral hardware and external sources. Eight levels of priorities can be specified for these interrupt requests, and multiple servicing control can be performed on interrupt sources.

(6) Clock generator (CG)

Two types of oscillators, a main clock (f_{XX}) and a subclock (f_{XT}), are provided. The clock generator generates seven types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, and f_{XT}), of which one is supplied as the operating clock of the CPU (f_{CPU}).

(7) Ring-OSC

A Ring-OSC oscillator is provided. The oscillation frequency is 200 kHz (TYP.). This Ring-OSC oscillator supplies a clock to watchdog timer 2 and timer M.

(8) Timers/counters

16-bit timer/event counter P (TMP), 16-bit timer/event counter Q (TMQ), and 16-bit interval timer M (TMM) are provided (refer to **Table 1-1**).

(9) Watch timer

This timer counts the reference time for watch counting from the subclock or f_{BRG} from prescaler 3. At the same time, it can also be used as an interval timer that operates on the main clock.

(10) Watchdog timer 2

This watchdog timer is used to detect a program loop and system errors.

★ As the source clock of this timer, Ring-OSC, or main clock can be selected.

When this watchdog timer overflows, it generates a non-maskable interrupt request signal (INTWDT2) or system reset signal (WDT2RES).

(11) Serial interface (SIO)

The V850ES /FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 have asynchronous serial interface A (UARTA) and 3-wire variable-length serial interface B (CSIB) as serial interfaces, and up to seven channels can be used at the same time.

UARTA transfers data by using the TXDAn and RXDAn pins (n = 0 up to 3 refer to **Table 1-1**).

CSIB transfers data by using the SOBm, SIBm, and $\overline{\text{SCKBm}}$ pins (m = 0 up to 2 refer to **Table 1-1**).

UARTA has a dedicated baud rate generator.

(12) CAN controller

The CAN controller is a small-scale digital data transmission system that transfers data between units.

(13) A/D converter

This is a high-speed, high-resolution 10-bit A/D converter with up to 24 analog input pins (refer to **Table 1-1)**. This converter is a successive approximation type.

(14) DMA controller

The V850ES/FG2, V850ES/FJ2 has a four-channel DMA controller that transfers data between the internal RAM, on-chip peripheral I/O, and external memory, in response to interrupt requests from the on-chip peripheral I/O.

(15) Key interrupt function

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

(16) On-chip debug function (Flash memory product only)

An on-chip debug function (Flash memory product only) that uses the communication specifications of JTAG (Joint Test Action Group) and that is used via an N-Wire in-circuit emulator is provided. The normal port function and on-chip debug function are selected by using the input level of a control pin and on-chip debug mode setting register (OCDM).

(17) Ports

General-purpose port functions and control pin functions are available. For details, refer to **CHAPTER 4 PORT FUNCTIONS**.

CHAPTER 2 PIN FUNCTIONS

This section explains the names and functions of the pins of the V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2.

2.1 Pin Function List

2.1.1 V850ES/FE2

Two I/O buffer power supplies, AV_{REF0} and EV_{DD}, are available.The relationship between the power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies (V850ES/FE2)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
EV_{DD}	Port 0, Port 3, Port 4, Port 5, Port 6, Port 8, Port 9, Port CM, Port DL, $\overline{\text{RESET}}$

(1) Port pins

Table 2-2. Pin List (Port Pins V850ES/FE2)

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.	TIP31/TOP31
P01			TIP30/TOP30
P02			NMI
P03			INTP0/ADTRG
P04			INTP1
P05			INTP2/$\overline{\text{DRST}}$
P06			INTP3
P30	I/O	Port 3 6-bit I/O port Input/output can be specified in 1-bit units.	TXDA0
P31			RXDA0/INTP7
P32			ASCKA0/TIP00/TOP00/TOP01
P33			TIP01/TOP01/CTXD0
P34			TIP10/TOP10/CRXD0
P35			TIP11/TOP11
P40	I/O	Port 4 3-bit I/O port Input/output can be specified in 1-bit units.	SIB0
P41			SOB0
P42			$\overline{\text{SCKB0}}$
P50	I/O	Port 5 6-bit I/O port Input/output can be specified in 1-bit units.	KR0/TIQ01/TOQ01
P51			KR1/TIQ02/TOQ02
P52			KR2/TIQ03/TOQ03/DDI
P53			KR3/TIQ00/TOQ00/DDO
P54			KR4/DCK
P55			KR5/DMS
P70 to P79	I/O	Port 7 10-bit I/O port Input/output can be specified in 1-bit units.	ANI0 to ANI9
P90	I/O	Port 9 9-bit I/O port Input/output can be specified in 1-bit units.	KR6/TXDA1
P91			KR7/RXDA1
P96			TIP21/TOP21
P97			SIB1/TIP20/TOP20
P98			SOB1
P99			$\overline{\text{SCKB1}}$
P913			INTP4/PCL
P914			INTP5
P915			INTP6
PCM0	I/O	Port CM 2-bit I/O port Input/output can be specified in 1-bit units.	-
PCM1			CLKOUT
PDL0 to PDL4	I/O	Port DL 8-bit I/O port Input/output can be specified in 1-bit units.	-
PDL5			FLMD1
PDL6, PDL7			-

(2) Non-port pins

★ **Table 2-3. Pin List (Non-Port Pins V850ES/FE2) (1/3)**

Pin Name	I/O	Function	Alternate Function
NMI	Input	External interrupt input (non-maskable, with analog noise eliminated)	P02 Note
INTP0	Input	External interrupt request input (maskable, with analog noise eliminated)	P03/ADTRG
INTP1			P04
INTP2			P05/$\overline{\text{DRST}}$
INTP3			P06
INTP4			P913/PCL
INTP5			P914
INTP6			P915
INTP7			P31/RXDA0
TIP00	Input	External event/clock input (TMP00)	P32/ASCKA0/TOP00/TOP01
TIP01		External event input (TMP01)	P33/TOP01/CTXD0
TIP10		External event/clock input (TMP10)	P34/TOP10/CRXD0
TIP11		External event input (TMP11)	P35/TOP11
TIP20		External event/clock input (TMP20)	P97/SIB1/TOP20
TIP21		External event input (TMP21)	P96/TOP21
TIP30		External event/clock input (TMP30)	P01/TOP30
TIP31		External event input (TMP31)	P00/TOP31
TOP00	Output	Timer output (TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01		Timer output (TMP01)	P32/ASCKA0/TIP00/TOP00
			P33/TIP01/CTXD0
TOP10		Timer output (TMP10)	P34/TIP10/CRXD0
TOP11		Timer output (TMP11)	P35/TIP11
TOP20		Timer output (TMP20)	P97/SIB1/TIP20
TOP21		Timer output (TMP21)	P96/TIP21
TOP30		Timer output (TMP30)	P01/TIP30
TOP31		Timer output (TMP31)	P00/TIP31

Note: The NMI pin and P02 pin are an alternate-function pin. This pin functions as the P02 pin after if has been reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1.The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTR0 registers.

★ **Table 2-3. Pin List (Non-Port Pins V850ES/FE2) (2/3)**

Pin Name	I/O	Function	Alternate Function
TIQ00	Input	External event/clock input (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01		External event input (TMQ01)	P50/KR0/TOQ01
TIQ02		External event input (TMQ02)	P51/KR1/TOQ02
TIQ03		External event input (TMQ03)	P52/KR2/TOQ03/DDI
TOQ00	Output	Timer output (TMQ00)	P53/KR3/TIQ00/DDO
TOQ01		Timer output (TMQ01)	P50/KR0/TIQ01
TOQ02		Timer output (TMQ02)	P51/KR1/TIQ02
TOQ03		Timer output (TMQ03)	P52/KR2/TIQ03/DDI
SIB0	Input	Serial receive data input (CSIB0)	P40
SIB1		Serial receive data input (CSIB1)	P97/TIP20/TOP20
SOB0	Output	Serial transmit data output (CSIB0)	P41
SOB1		Serial transmit data output (CSIB1)	P98
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99
RXDA0	Input	Serial receive data input (UARTA0)	P31/INTP7
RXDA1		Serial receive data input (UARTA1)	P91/KR7
TXDA0	Output	Serial transmit data output (UARTA0)	P30
TXDA1		Serial transmit data output (UARTA1)	P90/KR6
ASCKA0	Input	Baud rate clock input to UARTA0	P32/TIP00/TOP00/TOP01
CRXD0	Input	CAN receive data input (CAN0)	P34/TIP10/TOP10
CTXD0	Output	CAN transmit data output (CAN0)	P33/TIP01/TOP01
ANI0 to ANI9	Input	Analog voltage input to A/D converter	P70 to P79
AV_{REF0}	Input	Reference voltage input to A/D converter, and positive power supply pin for port 7	–
AV_{SS}	–	Ground potential for A/D converter (same potential as V_{SS})	–
ADTRG	Input	A/D converter external trigger input	P03/INTP0
KR0	Input	Key interrupt input	P50/TIQ01/TOQ01
KR1			P51/TIQ02/TOQ02
KR2			P52/TIQ03/TOQ03/DDI
KR3			P53/TIQ00/TOQ00/DDO
KR4			P54/DCK
KR5			P55/DMS
KR6			P90/TXDA1
KR7			P91/RXDA1

Table 2-3. Pin List (Non-Port Pins V850ES/FE2) (2/3)

Pin Name	I/O	Function	Alternate Function
DMS	Input	Debug mode select	P55/KR5
DDI	Input	Debug data input	P52/KR2/TIQ03/TOQ03
DDO	Output	Debug data output	P53/KR3/TIQ00/TOQ00
DCK	Input	Debug clock input	P54/KR4
$\overline{\text{DRST}}$	Input	Debug reset input	P05/INTP2
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	Output	Chip select signal output	PCS0 to PCS3
FLMD0	Input	Flash programming mode setting pins	–
FLMD1			PDL5
CLKOUT	Output	Internal system clock output	PCM1
PCL	Output	Clock output (timing output of X1 input clock and subclock)	P913/INTP4
REGC	–	Regulator output stabilizing capacitor connection	–
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Main clock resonator connection	–
X2	–		–
XT1	Input	Subclock resonator connection	–
XT2	–		–
V_{DD}	–	Positive power supply pin for internal circuitry	–
V_{SS}	–	Ground potential for internal circuitry	–
EV_{DD}	–	Positive power supply pin for external circuitry (same potential as V_{DD})	–
EV_{SS}	–	Ground potential for external circuitry (same potential as V_{SS})	–

2.1.2 V850ES/FF2

Two I/O buffer power supplies, AV_{REF0} and EV_{DD}, are available. The relationship between the power supplies and the pins is shown below.

Table 2-4. Pin I/O Buffer Power Supplies (V850ES/FF2)

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

(1) Port pins

Table 2-5. Pin List (Port Pins V850ES/FF2) (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.	TIP31/TOP31
P01			TIP30/TOP30
P02			NMI
P03			INTP0/ADTRG
P04			INTP1
P05			INTP2/$\overline{DRST}$
P06			INTP3
P30	I/O	Port 3 8-bit I/O port Input/output can be specified in 1-bit units.	TXDA0
P31			RXDA0/INTP7
P32			ASCKA0/TIP00/TOP00/TOP01
P33			TIP01/TOP01/CTXD0
P34			TIP10/TOP10/CRXD0
P35			TIP11/TOP11
P38			-
P39			-
P40	I/O	Port 4 3-bit I/O port Input/output can be specified in 1-bit units.	SIB0
P41			SOB0
P42			$\overline{SCKB0}$
P50	I/O	Port 5 6-bit I/O port Input/output can be specified in 1-bit units.	KR0/TIQ01/TOQ01
P51			KR1/TIQ02/TOQ02
P52			KR2/TIQ03/TOQ03/DDI
P53			KR3/TIQ00/TOQ00/DDO
P54			KR4/DCK
P55			KR5/DMS

Table 2-5. Pin List (Port Pins V850ES/FF2) (2/2)

Pin Name	I/O	Function	Alternate Function
P70 to P711	I/O	Port 7 12-bit I/O port Input/output can be specified in 1-bit units.	ANI0 to ANI11
P90	I/O	Port 9 9-bit I/O port Input/output can be specified in 1-bit units.	KR6/TXDA1
P91			KR7/RXDA1
P96			TIP21/TOP21
P97			SIB1/TIP20/TOP20
P98			SOB1
P99			$\overline{\text{SCKB1}}$
P913			INTP4/PCL
P914			INTP5
P915			INTP6
PCM0	I/O	Port CM 4-bit I/O port Input/output can be specified in 1-bit units.	-
PCM1			CLKOUT
PCM2, PCM3			-
PCS0, PCS1	I/O	Port CS 2-bit I/O port Input/output can be specified in 1-bit units.	-
PCT0, PCT1, PCT4, PCT6	I/O	Port CT 4-bit I/O port Input/output can be specified in 1-bit units.	-
PDL0 to PDL4	I/O	Port DL 8-bit I/O port Input/output can be specified in 1-bit units.	-
PDL5			FLMD1
PDL6 to PDL11			-

(2) Non-port pins

★ **Table 2-6. Pin List (Non-Port Pins V850ES/FF2) (1/3)**

Pin Name	I/O	Function	Alternate Function
NMI	Input	External interrupt input (non-maskable, with analog noise eliminated)	P02 Note
INTP0	Input	External interrupt request input (maskable, with analog noise eliminated)	P03/ADTRG
INTP1			P04
INTP2			P05/$\overline{\text{DRST}}$
INTP3			P06
INTP4			P913/PCL
INTP5			P914
INTP6			P915
INTP7			P31/RXDA0
TIP00	Input	External event/clock input (TMP00)	P32/ASCKA0/TOP00/TOP01
TIP01		External event input (TMP01)	P33/TOP01/CTXD0
TIP10		External event/clock input (TMP10)	P34/TOP10/CRXD0
TIP11		External event input (TMP11)	P35/TOP11
TIP20		External event/clock input (TMP20)	P97/SIB1/TOP20
TIP21		External event input (TMP21)	P96/TOP21
TIP30		External event/clock input (TMP30)	P01/TOP30
TIP31		External event input (TMP31)	P00/TOP31
TOP00	Output	Timer output (TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01		Timer output (TMP01)	P32/ASCKA0/TIP00/TOP00
			P33/TIP01/CTXD0
TOP10		Timer output (TMP10)	P34/TIP10/CRXD0
TOP11		Timer output (TMP11)	P35/TIP11
TOP20		Timer output (TMP20)	P97/SIB1/TIP20
TOP21		Timer output (TMP21)	P96/TIP21
TOP30		Timer output (TMP30)	P01/TIP30
TOP31		Timer output (TMP31)	P00/TIP31

Note: The NMI pin and P02 pin are an alternate-function pin. This pin functions as the P02 pin after if has been reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1.The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTR0 registers.

★ **Table 2-6. Pin List (Non-Port Pins V850ES/FF2) (2/3)**

Pin Name	I/O	Function	Alternate Function
TIQ00	Input	External event/clock input (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01		External event input (TMQ01)	P50/KR0/TOQ01
TIQ02		External event input (TMQ02)	P51/KR1/TOQ02
TIQ03		External event input (TMQ03)	P52/KR2/TOQ03/DDI
TOQ00	Output	Timer output (TMQ00)	P53/KR3/TIQ00/DDO
TOQ01		Timer output (TMQ01)	P50/KR0/TIQ01
TOQ02		Timer output (TMQ02)	P51/KR1/TIQ02
TOQ03		Timer output (TMQ03)	P52/KR2/TIQ03/DDI
SIB0	Input	Serial receive data input (CSIB0)	P40
SIB1		Serial receive data input (CSIB1)	P97/TIP20/TOP20
SOB0	Output	Serial transmit data output (CSIB0)	P41
SOB1		Serial transmit data output (CSIB1)	P98
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99
RXDA0	Input	Serial receive data input (UARTA0)	P31/INTP7
RXDA1		Serial receive data input (UARTA1)	P91/KR7
TXDA0	Output	Serial transmit data output (UARTA0)	P30
TXDA1		Serial transmit data output (UARTA1)	P90/KR6
ASCKA0	Input	Baud rate clock input to UARTA0	P32/TIP00/TOP00/TOP01
CRXD0	Input	CAN receive data input (CAN0)	P34/TIP10/TOP10
CTXD0	Output	CAN transmit data output (CAN0)	P33/TIP01/TOP01
ANI0 to ANI11	Input	Analog voltage input to A/D converter	P70 to P711
AV_{REF0}	Input	Reference voltage input to A/D converter , and positive power supply pin for port 7	–
AV_{SS}	–	Ground potential for A/D converter (same potential as V_{SS})	–
ADTRG	Input	A/D converter external trigger input	P03/INTP0
KR0	Input	Key interrupt input	P50/TIQ01/TOQ01
KR1			P51/TIQ02/TOQ02
KR2			P52/TIQ03/TOQ03/DDI
KR3			P53/TIQ00/TOQ00/DDO
KR4			P54/DCK
KR5			P55/DMS
KR6			P90/TXDA1
KR7			P91/RXDA1

Table 2-6. Pin List (Non-Port Pins V850ES/FF2) (2/3)

Pin Name	I/O	Function	Alternate Function
DMS	Input	Debug mode select	P55/KR5
DDI	Input	Debug data input	P52/KR2/TIQ03/TOQ03
DDO	Output	Debug data output	P53/KR3/TIQ00/TOQ00
DCK	Input	Debug clock input	P54/KR4
$\overline{\text{DRST}}$	Input	Debug reset input	P05/INTP2
FLMD0	Input	Flash programming mode setting pins	–
FLMD1			PDL5
CLKOUT	Output	Internal system clock output	PCM1
PCL	Output	Clock output (timing output of X1 input clock and subclock)	P913/INTP4
REGC	–	Regulator output stabilizing capacitor connection	–
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Main clock resonator connection	–
X2	–		–
XT1	Input	Subclock resonator connection	–
XT2	–		–
V_{DD}	–	Positive power supply pin for internal circuitry	–
V_{SS}	–	Ground potential for internal circuitry	–
EV_{DD}	–	Positive power supply pin for external circuitry (same potential as V_{DD})	–
EV_{SS}	–	Ground potential for external circuitry (same potential as V_{SS})	–

2.1.3 V850ES/FG2

Three I/O buffer power supplies, AV_{REF0}, BV_{DD} and EV_{DD}, are available. The relationship between the power supplies and the pins is shown below.

Table 2-7. Pin I/O Buffer Power Supplies (V850ES/FG2)

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

(1) Port pins

Table 2-8. Pin List (Port Pins V850ES/FG2) (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.	TIP31/TOP31
P01			TIP30/TOP30
P02			NMI
P03			INTP0/ADTRG
P04			INTP1
P05			INTP2/$\overline{\text{DRST}}$
P06			INTP3
P10	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units.	INTP9
P11			INTP10
P30	I/O	Port 3 10-bit I/O port Input/output can be specified in 1-bit units.	TXDA0
P31			RXDA0/INTP7
P32			ASCKA0/TIP00/TOP00/TOP01
P33			TIP01/TOP01/CTXD0
P34			TIP10/TOP10/CRXD0
P35			TIP11/TOP11
P36			CTXD1
P37			CRXD1
P38			TXDA2
P39			RXDA2/INTP8
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 2-8. Pin List (Port Pins V850ES/FG2) (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.	KR0/TIQ01/TOQ01
P51			KR1/TIQ02/TOQ02
P52			KR2/TIQ03/TOQ03/DDI
P53			KR3/TIQ00/TOQ00/DDO
P54			KR4/DCK
P55			KR5/DMS
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.	KR6/TXDA1
P91			KR7/RXDA1
P92			TIQ11/TOQ11
P93			TIQ12/TOQ12
P94			TIQ13/TOQ13
P95			TIQ10/TOQ10
P96			TIP21/TOP21
P97			SIB1/TIP20/TOP20
P98			SOB1
P99			$\overline{\text{SCKB1}}$
P910			–
P911			–
P912			–
P913			INTP4/PCL
P914			INTP5
P915			INTP6
PCM0	I/O	Port CM 4-bit I/O port Input/output can be specified in 1-bit units.	–
PCM1			CLKOUT
PCM2, PCM3			–
PCS0, PCS1	I/O	Port CS 2-bit I/O port Input/output can be specified in 1-bit units.	–
PCT0, PCT1,	I/O	Port CT	–
PCT4, PCT6		4-bit I/O port Input/output can be specified in 1-bit units.	–
PDL0 to PDL4	I/O	Port DL 14-bit I/O port Input/output can be specified in 1-bit units.	–
PDL5			FLMD1
PDL6 to PDL13			–

(2) Non-port pins

★ **Table 2-9. Pin List (Non-Port Pins V850ES/FG2) (1/3)**

Pin Name	I/O	Function	Alternate Function
NMI	Input	External interrupt input (non-maskable, with analog noise eliminated)	P02 Note
INTP0	Input	External interrupt request input (maskable, with analog noise eliminated)	P03/ADTRG
INTP1			P04
INTP2			P05/$\overline{\text{DRST}}$
INTP3			P06
INTP4			P913/PCL
INTP5			P914
INTP6			P915
INTP7			P31/RXDA0
INTP8			P39/RXDA2
INTP9			P10
INTP10			P11
TIP00	Input	External event/clock input (TMP00)	P32/ASCKA0/TOP00/TOP01
TIP01		External event/clock input (TMP01)	P33/TOP01/CTXD0
TIP10		External event/clock input (TMP10)	P34/TOP10/CRXD0
TIP11		External event/clock input (TMP11)	P35/TOP11
TIP20		External event/clock input (TMP20)	P97/SIB1/TOP20
TIP21		External event/clock input (TMP21)	P96/TOP21
TIP30		External event/clock input (TMP30)	P01/TOP30
TIP31		External event/clock input (TMP31)	P00/TOP31
TOP00	Output	Timer output (TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01		Timer output (TMP01)	P32/ASCKA0/TIP00/TOP00
			P33/TIP01/CTXD0
TOP10		Timer output (TMP10)	P34/TIP10/CRXD0
TOP11		Timer output (TMP11)	P35/TIP11
TOP20		Timer output (TMP20)	P97/SIB1/TIP20
TOP21		Timer output (TMP21)	P96/TIP21
TOP30		Timer output (TMP30)	P01/TIP30
TOP31		Timer output (TMP31)	P00/TIP31
TIQ00	Input	External event/clock input (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01		External event input (TMQ01)	P50/KR0/TOQ01
TIQ02		External event input (TMQ02)	P51/KR1/TOQ02
TIQ03		External event input (TMQ03)	P52/KR2/TOQ03/DDI

Note: The NMI pin and P02 pin are an alternate-function pin. This pin functions as the P02 pin after if has been reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1.The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTR0 registers.

★ **Table 2-9. Pin List (Non-Port Pins V850ES/FG2) (2/3)**

Pin Name	I/O	Function	Alternate Function
TIQ10	Input	External event input (TMQ10)	P95/TOQ10
TIQ11		External event input (TMQ11)	P92/TOQ11
TIQ12		External event input (TMQ12)	P93/TOQ12
TIQ13		External event input (TMQ13)	P94/TOQ13
TOQ00	Output	Timer output (TMQ00)	P53/KR3/TIQ00/DDO
TOQ01		Timer output (TMQ01)	P50/KR0/TIQ01
TOQ02		Timer output (TMQ02)	P51/KR1/TIQ02
TOQ03		Timer output (TMQ03)	P52/KR2/TIQ03/DDI
TOQ10		Timer output (TMQ10)	P95/TIQ10
TOQ11		Timer output (TMQ11)	P92/TIQ11
TOQ12		Timer output (TMQ12)	P93/TIQ12
TOQ13		Timer output (TMQ13)	P94/TIQ13
SIB0	Input	Serial receive data input (CSIB0)	P40
SIB1		Serial receive data input (CSIB1)	P97/TIP20/TOP20
SOB0	Output	Serial transmit data output (CSIB0)	P41
SOB1		Serial transmit data output (CSIB1)	P98
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99
RXDA0	Input	Serial receive data input (UARTA0)	P31/INTP7
RXDA1		Serial receive data input (UARTA1)	P91/KR7
RXDA2		Serial receive data input (UARTA2)	P39/INTP8
TXDA0	Output	Serial transmit data output (UARTA0)	P30
TXDA1		Serial transmit data output (UARTA1)	P90/KR6
TXDA2		Serial transmit data output (UARTA2)	P38
ASCKA0	Input	Baud rate clock input to UARTA0	P32/TIP00/TOP00/TOP01
CRXD0	Input	CAN receive data input (CAN0)	P34/TIP10/TOP10
CRXD1		CAN receive data input (CAN1)	P37
CTXD0	Output	CAN transmit data output (CAN0)	P33/TIP01/TOP01
CTXD1		CAN transmit data output (CAN1)	P36
ANI0 to ANI15	Input	Analog voltage input to A/D converter	P70 to P715
AV_{REF0}	Input	Reference voltage input to A/D converter , and positive power supply pin for port 7	–
AV_{SS}	–	Ground potential for A/D converter (same potential as V_{SS})	–
ADTRG	Input	A/D converter external trigger input	P03/INTP0
KR0	Input	Key interrupt input	P50/TIQ01/TOQ01
KR1			P51/TIQ02/TOQ02
KR2			P52/TIQ03/TOQ03/DDI
KR3			P53/TIQ00/TOQ00/DDO
KR4			P54/DCK
KR5			P55/DMS
KR6			P90/TXDA1

KR7			P91/RXDA1

Table 2-9. Pin List (Non-Port Pins V850ES/FG2) (3/3)

Pin Name	I/O	Function	Alternate Function
DMS	Input	Debug mode select	P55/KR5
DDI	Input	Debug data input	P52/KR2/TIQ03/TOQ03
DDO	Output	Debug data output	P53/KR3/TIQ00/TOQ00
DCK	Input	Debug clock input	P54/KR4
$\overline{\text{DRST}}$	Input	Debug reset input	P05/INTP2
FLMD0	Input	Flash programming mode setting pins	–
FLMD1			PDL5
CLKOUT	Output	Internal system clock output	PCM1
PCL	Output	Clock output (timing output of X1 input clock and subclock)	P913/INTP4
REGC	–	Regulator output stabilizing capacitor connection	–
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Main clock resonator connection	–
X2	–		–
XT1	Input	Subclock resonator connection	–
XT2	–		–
V_{DD}	–	Positive power supply pin for internal circuitry	–
V_{SS}	–	Ground potential for internal circuitry	–
EV_{DD}	–	Positive power supply pin for external circuitry (same potential as V_{DD})	–
EV_{SS}	–	Ground potential for external circuitry (same potential as V_{SS})	–
BV_{DD}	–	Positive power supply pin for external circuitry (same potential as V_{DD})	–
BV_{SS}	–	Ground potential for external circuitry (same potential as V_{SS})	–

2.1.4 V850ES/FJ2

Three I/O buffer power supplies, AV_{REF0}, BV_{DD}, and EV_{DD}, are available. The relationship between the power supplies and the pins is shown below.

Table 2-10. Pin I/O Buffer Power Supplies (V850ES/FJ2)

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

(1) Port pins

Table 2-11. Pin List (Port Pins) (1/3)

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.	TIP31/TOP31
P01			TIP30/TOP30
P02			NMI
P03			INTP0/ADTRG
P04			INTP1
P05			INTP2/$\overline{\text{DRST}}$
P06			INTP3
P10	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units.	INTP9
P11			INTP10
P30	I/O	Port 3 10-bit I/O port Input/output can be specified in 1-bit units.	TXDA0
P31			RXDA0/INTP7
P32			ASCKA0/TIP00/TOP00/TOP01
P33			TIP01/TOP01/CTXD0
P34			TIP10/TOP10/CRXD0
P35			TIP11/TOP11
P36			CTXD1
P37			CRXD1
P38			TXDA2
P39			RXDA2/INTP8
P40	I/O	Port 4 3-bit I/O port Input/output can be specified in 1-bit units.	SIB0
P41			SOB0
P42			$\overline{\text{SCKB0}}$
P50	I/O	Port 5 6-bit I/O port Input/output can be specified in 1-bit units.	KR0/TIQ01/TOQ01
P51			KR1/TIQ02/TOQ02
P52			KR2/TIQ03/TOQ03/DDI
P53			KR3/TIQ00/TOQ00/DDO
P54			KR4/DCK
P55			KR5/DMS

Table 2-11. Pin List (Port Pins) (2/3)

Pin Name	I/O	Function	Alternate Function
P60	I/O	Port 6 16-bit I/O port Input/output can be specified in 1-bit units.	INTP11
P61			INTP12
P62			INTP13
P63			–
P64			–
P65			CTXD2[Note 1]
P66			CRXD2[Note 1]
P67			CTXD3[Note 1]
P68			CRXD3[Note 1]
P69			–
P610			TIQ20/TOQ20
P611			TIQ21/TOQ21
P612			TIQ22/TOQ22
P613			TIQ23/TOQ23
P614			–
P615			–
P70 to P715	I/O	Port 7 16-bit I/O port Input/output can be specified in 1-bit units.	ANI0 to ANI15
P80	I/O	Port 8 2-bit I/O port Input/output can be specified in 1-bit units.	RXDA3/INTP14[Note 2]
P81			TXDA3[Note 2]
P90	I/O	Port 9 16-bit I/O port Input/output can be specified in 1-bit units.	KR6/TXDA1
P91			KR7/RXDA1
P92			TIQ11/TOQ11
P93			TIQ12/TOQ12
P94			TIQ13/TOQ13
P95			TIQ10/TOQ10
P96			TIP21/TOP21
P97			SIB1/TIP20/TOP20
P98			SOB1
P99			$\overline{\text{SCKB1}}$
P910			SIB2
P911			SOB2
P912			$\overline{\text{SCKB2}}$
P913			INTP4/PCL
P914			INTP5
P915			INTP6

Notes 1. In the μPD70F3237, alternate functions of the P65 to P68 pins (CTXD2, CRXD2, CTXD3, and CRXD3) are not available.

2. In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available. The alternate function of the P80 pin in the μPD70F3237 is INTP14 only.

Table 2-11. Pin List (Port Pins) (3/3)

Pin Name	I/O	Function	Alternate Function
P120 to P127	I/O	Port 12 8-bit I/O port Input/output can be specified in 1-bit units.	ANI16 to ANI23
PCD0 to PCD3	I/O	Port CD 4-bit I/O port Input/output can be specified in 1-bit units.	–
PCM0	I/O	Port CM 6-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WAIT}}$
PCM1			CLKOUT
PCM2			$\overline{\text{HLDAK}}$
PCM3			$\overline{\text{HLDRQ}}$
PCM4			–
PCM5			–
PCS0 to PCS3	I/O	Port CS 8-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
PCS4 to PCS7			–
PCT0	I/O	Port CT 8-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WR0}}$
PCT1			$\overline{\text{WR1}}$
PCT2			–
PCT3			–
PCT4			$\overline{\text{RD}}$
PCT5			–
PCT6			ASTB
PCT7			–
PDL0 to PDL4	I/O	Port DL 16-bit I/O port Input/output can be specified in 1-bit units.	AD0 to AD4
PDL5			AD5/FLMD1
PDL6 to PDL15			AD6 to AD15

(2) Non-port pins

★ **Table 2-12. Pin List (Non-Port Pins) (1/4)**

Pin Name	I/O	Function	Alternate Function
NMI Note 1	Input	External interrupt input (non-maskable, with analog noise eliminated)	P02 Note 1
INTP0	Input	External interrupt request input (maskable, with analog noise eliminated)	P03/ADTRG
INTP1			P04
INTP2			P05/$\overline{\text{DRST}}$
INTP3			P06
INTP4			P913/PCL
INTP5			P914
INTP6			P915
INTP7			P31/RXDA0
INTP8			P39/RXDA2
INTP9			P10
INTP10			P11
INTP11			P60
INTP12			P61
INTP13			P62
INTP14			P80/RXDA3 Note 2
TIP00	Input	External event/clock input (TMP00)	P32/ASCKA0/TOP00/TOP01
TIP01		External event input (TMP01)	P33/TOP01/CTXD0
TIP10		External event/clock input (TMP10)	P34/TOP10/CRXD0
TIP11		External event input (TMP11)	P35/TOP11
TIP20		External event/clock input (TMP20)	P97/SIB1/TOP20
TIP21		External event input (TMP21)	P96/TOP21
TIP30		External event/clock input (TMP30)	P01/TOP30
TIP31		External event input (TMP31)	P00/TOP31
TOP00	Output	Timer output (TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01		Timer output (TMP01)	P32/ASCKA0/TIP00/TOP00
			P33/TIP01/CTXD0
TOP10		Timer output (TMP10)	P34/TIP10/CRXD0
TOP11		Timer output (TMP11)	P35/TIP11
TOP20		Timer output (TMP20)	P97/SIB1/TIP20
TOP21		Timer output (TMP21)	P96/TIP21
TOP30		Timer output (TMP30)	P01/TIP30
TOP31		Timer output (TMP31)	P00/TIP31

Notes 1. The NMI pin and P02 pin are an alternate-function pin. This pin functions as the P02 pin after if has been reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1.The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTR0 registers.

2. In the µPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available. The alternate function of the P80 pin in the µPD70F3237 is only INTP14.

★ **Table 2-12. Pin List (Non-Port Pins) (2/4)**

Pin Name	I/O	Function	Alternate Function
TIQ00	Input	External event/clock input (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01		External event input (TMQ01)	P50/KR0/TOQ01
TIQ02		External event input (TMQ02)	P51/KR1/TOQ02
TIQ03		External event input (TMQ03)	P52/KR2/TOQ03/DDI
TIQ10		External event input (TMQ10)	P95/TOQ10
TIQ11		External event input (TMQ11)	P92/TOQ11
TIQ12		External event input (TMQ12)	P93/TOQ12
TIQ13		External event input (TMQ13)	P94/TOQ13
TIQ20		External event/clock input (TMQ20)	P610/TOQ20
TIQ21		External event input (TMQ21)	P611/TOQ21
TIQ22		External event input (TMQ22)	P612/TOQ22
TIQ23		External event input (TMQ23)	P613/TOQ23
TOQ00	Output	Timer output (TMQ00)	P53/KR3/TIQ00/DDO
TOQ01		Timer output (TMQ01)	P50/KR0/TIQ01
TOQ02		Timer output (TMQ02)	P51/KR1/TIQ02
TOQ03		Timer output (TMQ03)	P52/KR2/TIQ03/DDI
TOQ10		Timer output (TMQ10)	P95/TIQ10
TOQ11		Timer output (TMQ11)	P92/TIQ11
TOQ12		Timer output (TMQ12)	P93/TIQ12
TOQ13		Timer output (TMQ13)	P94/TIQ13
TOQ20		Timer output (TMQ20)	P610/TIQ20
TOQ21		Timer output (TMQ21)	P611/TIQ21
TOQ22		Timer output (TMQ22)	P612/TIQ22
TOQ23		Timer output (TMQ23)	P613/TIQ23
SIB0	Input	Serial receive data input (CSIB0)	P40
SIB1		Serial receive data input (CSIB1)	P97/TIP20/TOP20
SIB2		Serial receive data input (CSIB2)	P910
SOB0	Output	Serial transmit data output (CSIB0)	P41
SOB1		Serial transmit data output (CSIB1)	P98
SOB2		Serial transmit data output (CSIB2)	P911
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99
$\overline{\text{SCKB2}}$		Serial clock I/O (CSIB2)	P912
RXDA0	Input	Serial receive data input (UARTA0)	P31/INTP7
RXDA1		Serial receive data input (UARTA1)	P91/KR7
RXDA2		Serial receive data input (UARTA2)	P39/INTP8
RXDA3[Note]		Serial receive data input (UARTA3)	P80/INTP14
TXDA0	Output	Serial transmit data output (UARTA0)	P30
TXDA1		Serial transmit data output (UARTA1)	P90/KR6
TXDA2		Serial transmit data output (UARTA2)	P38
TXDA3[Note]		Serial transmit data output (UARTA3)	P81

Note In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available.

Table 2-12. Pin List (Non-Port Pins) (3/4)

Pin Name	I/O	Function	Alternate Function
ASCKA0	Input	Baud rate clock input to UARTA0	P32/TIP00/TOP00/TOP01
CRXD0	Input	CAN receive data input (CAN0)	P34/TIP10/TOP10
CRXD1		CAN receive data input (CAN1)	P37
CRXD2Note		CAN receive data input (CAN2)	P66
CRXD3Note		CAN receive data input (CAN3)	P68
CTXD0	Output	CAN transmit data output (CAN0)	P33/TIP01/TOP01
CTXD1		CAN transmit data output (CAN1)	P36
CTXD2Note		CAN transmit data output (CAN2)	P65
CTXD3Note		CAN transmit data output (CAN3)	P67
ANI0 to ANI15	Input	Analog voltage input to A/D converter	P70 to P715
ANI16 to ANI23			P120 to P127
AV_{REF0}	Input	Reference voltage input to A/D converter, and positive power supply pin for port 7	–
AV_{SS}	–	Ground potential for A/D converter (same potential as VSS)	–
ADTRG	Input	A/D converter external trigger input	P03/INTP0
KR0	Input	Key interrupt input	P50/TIQ01/TOQ01
KR1			P51/TIQ02/TOQ02
KR2			P52/TIQ03/TOQ03/DDI
KR3			P53/TIQ00/TOQ00/DDO
KR4			P54/DCK
KR5			P55/DMS
KR6			P90/TXDA1
KR7			P91/RXDA1
DMS	Input	Debug mode select	P55/KR5
DDI	Input	Debug data input	P52/KR2/TIQ03/TOQ03
DDO	Output	Debug data output	P53/KR3/TIQ00/TOQ00
DCK	Input	Debug clock input	P54/KR4
$\overline{DRST}$	Input	Debug reset input	P05/INTP2
$\overline{CS0}$ to $\overline{CS3}$	Output	Chip select signal output	PCS0 to PCS3
AD0 to AD4	I/O	Address/data bus for external memory	PDL0 to PDL4
AD5			PDL5/FLMD1
AD6 to AD15			PDL6 to PDL15
ASTB	Output	Address strobe signal output to external memory	PCT6
$\overline{HLDRQ}$	Input	Bus hold request input	PCM3
$\overline{HLDAK}$	Output	Bus hold acknowledge output	PCM2
$\overline{RD}$	Output	Read strobe signal output to external memory	PCT4
$\overline{WAIT}$	Input	External wait input	PCM0
$\overline{WR0}$	Output	Write strobe to external memory (lower 8 bits)	PCT0
$\overline{WR1}$		Write strobe to external memory (higher 8 bits)	PCT1

Note In the μPD70F3237, the alternate functions of the P65 to P68 pins (CTXD2, CRXD2, CTXD3, and CRXD3) are not available.

Table 2-12. Pin List (Non-Port Pins) (4/4)

Pin Name	I/O	Function	Alternate Function
FLMD0	Input	Flash programming mode setting pins	–
FLMD1			PDL5/AD5
CLKOUT	Output	Internal system clock output	PCM1
PCL	Output	Clock output (timing output of X1 input clock and subclock)	P913/INTP4
REGC	–	Regulator output stabilizing capacitor connection	–
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Main clock resonator connection	–
X2	–		–
XT1	Input	Subclock resonator connection	–
XT2	–		–
V_{DD}	–	Positive power supply pin for internal circuitry	–
V_{SS}	–	Ground potential for internal circuitry	–
BV_{DD}	–	Positive power supply for bus interface and port	–
BV_{SS}	–	Ground potential for bus interface and port	–
EV_{DD}	–	Positive power supply pin for external circuitry (same potential as V_{DD})	–
EV_{SS}	–	Ground potential for external circuitry (same potential as V_{SS})	–

2.2 Pin Status (V850ES/FJ2)

The V850ES/FJ2 has an external bus interface function that enables connection of external memories, such as ROM and RAM, and I/O.

Table 2-4 shows the operating status of each external bus interface pin in each operation mode.

Table 2-13. Pin Operating Status in Each Operation Mode

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

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

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

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

2.3 Description of Pin Functions

2.3.1 V850ES/FE2

(1) P00 to P06 (port 0) … 3-state I/O

P00 to P06 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as NMI input, external interrupt request signal input, timer/counter I/O, external trigger of the A/D converter, and debug reset input.

This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR0 and INTF0 registers.

An on-chip pull-up resistor can be connected to P00 to P06 by using pull-up resistor option register 0 (PU0).

(a) Port mode

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

(b) Control mode

(i) NMI (Non-maskable interrupt request) … input

This pin inputs a non-maskable interrupt request signal.

(ii) INTP0 to INTP3 (Interrupt request from peripherals) … input

These pins input external interrupt request signals.

(iii) TIP30, TIP31 (Timer input) … input

These pins input to timers P3 (TMP3).

(iv) TOP30, TOP31 (Timer output) … output

These pins output from timers P3 (TMP3).

(v) ADTRG (A/D trigger input) … input

This pin inputs an external trigger to the A/D converter. It is controlled by using A/D converter mode register 0 (ADA0M0).

(vi) $\overline{\text{DRST}}$ (Debug reset) … input

This pin inputs a debug reset signal, a negative-logic signal that asynchronously initializes the on-chip debug circuit. To deassert this signal, reset or invalidate the on-chip debug circuit. Deassert this signal when the debug function is not used.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(2) P30 to P35 (port 3) … 3-state output

P30 to P35 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, serial interface I/O, timer/counter I/O, and CAN data I/O. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by using INTR3 and INTF3 registers.

An on-chip pull-up resistor can be connected to P30 to P35 by using pull-up resistor option register 3 (PU3).

(a) Port mode

P30 to P35 can be set in the input or output mode in 1-bit units, by using port mode register 3 (PM3).

(b) Control mode

(i) RXDA0 (Receive data) … input

These pins input the serial receive data of UARTA0.

(ii) TXDA0 (Transmit data) … output

These pins output the serial transmit data of UARTA0.

(iii) ASCKA0 (Asynchronous serial clock) … input

This pin inputs UARTA0.

(iv) INTP7 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(v) TIP00, TIP01, TIP10, TIP11 (Timer input) … input

These pins input to timers P0 and P1 (TMP0, TMP1).

(vi) TOP00, TOP01, TOP10, TOP11 (Timer output) … output

These pins output from timers P0 and P1 (TMP0, TMP1).

(vii) CRXD0 (CAN receive data) … input

These pins input the receive data of CAN0.

(viii) CTXD0 (CAN transmit data) … output

These pins output the transmit data of CAN0.

(3) P40 to P42 (port 4) … 3-state I/O

P40 to P42 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P40 to P42 by using pull-up resistor option register 4 (PU4).

(a) Port mode

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

(b) Control mode

(i) SIB0 (Serial input) … input

This pin inputs the serial receive data of CSIB0.

(ii) SOB0 (Serial output) … output

This pin outputs the serial transmit data of CSIB0.

(iii) $\overline{\text{SCKB0}}$ (serial clock) … 3-state I/O

This pin inputs/outputs the serial clock of CSIB0.

(4) P50 to P55 (Port 5) … 3-state I/O

P50 to P55 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as an I/O port, but also as timer/counter I/O, debug function I/O, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P50 to P55 by using pull-up resistor option register 5 (PU5).

(a) Port mode

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

(b) Control mode

(i) KR0 to KR5 (Key return) … input

These pins input a key interrupt. Their operation is specified by using the key return mode register (KRM) in the input port mode.

(ii) TIQ00, TIQ01, TIQ02, TIQ03 (Timer input) … input

These pins input to timers Q0 (TMQ0).

(iii) TOQ00, TOQ01, TOQ02, TOQ03 (Timer output) … output

These pins output from timers Q0 (TMQ0).

(iv) DDI (Debug data input) … input

This pin inputs debug data to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(v) DDO (Debug data output) … output

This pin outputs debug data from the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(iv) DCK (Debug clock input) … input

This pin inputs a debug clock to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(vii) DMS (Debug mode select) … input

This pin selects the debug mode of the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(5) P70 to P79 (Port 7) … 3-state I/O

P70 to P79 function as a 10-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as the analog input pins of the A/D converter in the control mode. When using this port as analog input pins, however, set the port in the input mode. At this time, do not read the port.

(a) Port mode

P70 to P79 can be set in the input or output mode in 1-bit units, by using port mode register 7L, H (PM7L, PM7H)

(b) Control mode

P70 to P79 function alternately as the ANI0 to ANI9 pins.

(i) ANI0 to ANI9 (Analog input 0 to 9) … input

These pins input an analog signal to the A/D converter.

(6) P90, P91, P96 to P99, P913 to P915 (Port 9) … 3-state I/O

P90, P91, P96 to P99, P913 to P915 function as a 9-bit I/O port that can be set to input or output in 1-bit units. Besides functioning as an I/O port, these pins operate as serial interface I/O, timer/counter I/O, clock output, external interrupt request signal input, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units. The valid edge of P913 to P915 is specified by using INTF9H register.

An on-chip pull-up resistor can be connected to P90, P91, P96 to P99, P913 to P915 by using pull-up resistor option register 9 (PU9).

(a) Port mode

P90, P91, P96 to P99, P913 to P915 can be set in the input or output mode in 1-bit units, by using port 9 mode register (PM9).

(b) Control mode

(i) SIB1 (Serial input) … input

These pins input the serial receive data of CSIB1.

(ii) SOB1 (Serial output) … output

These pins output the serial receive data of CSIB1.

(iii) $\overline{\text{SCKB1}}$ (Serial clock) … 3-state I/O

These pins input/output the serial clock of CSIB1.

(iv) RXDA1 (Receive data) … input

This pin inputs the serial receive data of UARTA1.

(v) TXDA1 (Transmit data) … output

This pin outputs the serial transmit data of UARTA1.

(vi) TIP20, TIP21 (Timer input) … input

These pins input to timers P2 (TMP2).

(vii) TOP20, TOP21 (Timer output) … output

These pins output from timers P2 (TMP2).

(viii) PCL (Clock output) … output

This pin outputs a clock.

(ix) INTP4 to INTP6 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(x) KR6, KR7 (Key return) … input

These pins input a key interrupt. Their operation is specified by the key return mode register (KRM) in the input port mode.

(7) PCM0, PCM1 (port CM) … 3-state I/O

PCM0, PCM1 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, and bus clock output.

(a) Port mode

PCM0, PCM1 can be set in the input or output mode in 1-bit units, by using port mode register CM (PMCM).

(b) Control mode

(i) CLKOUT (Clock output) … output

This pin outputs an internally generated bus clock.

(8) PDL0 to PDL7 (port DL) … 3-state I/O

PDL0 to PDL7 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

PDL5 also functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL7 can be set in the input or output mode in 1-bit units, by using port mode register DL (PMDL).

(9) $\overline{\text{RESET}}$ (Reset) … input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal with a fixed low level width is input to the $\overline{\text{RESET}}$ pin regardless of the operating clock, the system is reset, taking precedence over all the other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP), as well as for normal initialization/start.

(10) X1, X2 (Crystal for main clock)

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

(11) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

(12) AV_{SS} (Ground for analog)

This is a ground pin for the A/D converter, and alternate-function ports.

(13) AV_{REF0} (Analog reference voltage) … input

This pin supplies positive analog power to the A/D converter and alternate-function ports.

It also supplies a reference voltage to the A/D converter.

(14) EV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(15) EV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(16) V_{DD} (Power supply)

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

(17) V_{SS} (Ground)

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

★ **(18) FLMD0 (Flash programming mode) Input**

This is a signal input pin for flash memory programming mode. Connect this pin to VSS in the normal operation mode.

(19) REGC (Regulator control) … input

This pin connects a capacitor for the regulator.

2.3.2 V850ES/FF2

(1) P00 to P06 (port 0) … 3-state I/O

P00 to P06 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as NMI input, external interrupt request signal input, timer/counter I/O, external trigger of the A/D converter, and debug reset input.

This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR0 and INTF0 registers.

An on-chip pull-up resistor can be connected to P00 to P06 by using pull-up resistor option register 0 (PU0).

(a) Port mode

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

(b) Control mode

(i) NMI (Non-maskable interrupt request) … input

This pin inputs a non-maskable interrupt request signal.

(ii) INTP0 to INTP3 (Interrupt request from peripherals) … input

These pins input external interrupt request signals.

(iii) TIP30, TIP31 (Timer input) … input

These pins input to timers P3 (TMP3).

(iv) TOP30, TOP31 (Timer output) … output

These pins output from timers P3 (TMP3).

(v) ADTRG (A/D trigger input) … input

This pin inputs an external trigger to the A/D converter. It is controlled by using A/D converter mode register 0 (ADA0M0).

(vi) $\overline{\text{DRST}}$ (Debug reset) … input

This pin inputs a debug reset signal, a negative-logic signal that asynchronously initializes the on-chip debug circuit. To deassert this signal, reset or invalidate the on-chip debug circuit. Deassert this signal when the debug function is not used.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(2) P30 to P35, P38, P39 (port 3) … 3-state output

P30 to P35, P38, P39 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, serial interface I/O, timer/counter I/O, and CAN data I/O. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by using INTR3 and INTF3 registers.

An on-chip pull-up resistor can be connected to P30 to P35, P38, P39 by using pull-up resistor option register 3 (PU3).

(a) Port mode

P30 to P35, P38, P39 can be set in the input or output mode in 1-bit units, by using port mode register 3 (PM3).

(b) Control mode

(i) RXDA0 (Receive data) … input

These pins input the serial receive data of UARTA0.

(ii) TXDA0 (Transmit data) … output

These pins output the serial transmit data of UARTA0.

(iii) ASCKA0 (Asynchronous serial clock) … input

This pin inputs of UARTA0.

(iv) INTP7 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(v) TIP00, TIP01, TIP10, TIP11 (Timer input) … input

These pins input to timers P0, P1 (TMP0, TMP1).

(vi) TOP00, TOP01, TOP10, TOP11 (Timer output) … output

These pins output from timers P0, P1 (TMP0, TMP1).

(vii) CRXD0 (CAN receive data) … input

These pins input the receive data of CAN0.

(viii) CTXD0 (CAN transmit data) … output

These pins output the transmit data of CAN0.

(3) P40 to P42 (port 4) … 3-state I/O

P40 to P42 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P40 to P42 by using pull-up resistor option register 4 (PU4).

(a) Port mode

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

(b) Control mode

(i) SIB0 (Serial input) … input

This pin inputs the serial receive data of CSIB0.

(ii) SOB0 (Serial output) … output

This pin outputs the serial transmit data of CSIB0.

(iii) $\overline{\text{SCKB0}}$ (serial clock) … 3-state I/O

This pin inputs/outputs the serial clock of CSIB0.

(4) P50 to P55 (Port 5) … 3-state I/O

P50 to P55 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as an I/O port, but also as timer/counter I/O, debug function I/O, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P50 to P55 by using pull-up resistor option register 5 (PU5).

(a) Port mode

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

(b) Control mode

(i) KR0 to KR5 (Key return) … input

These pins input a key interrupt. Their operation is specified by using the key return mode register (KRM) in the input port mode.

(ii) TIQ00, TIQ01, TIQ02, TIQ03 (Timer input) … input

These pins input to timers Q0 (TMQ0).

(iii) TOQ00, TOQ01, TOQ02, TOQ03 (Timer output) … output

These pins output from timers Q0 (TMQ0).

(iv) DDI (Debug data input) … input

This pin inputs debug data to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(v) DDO (Debug data output) … output

This pin outputs debug data from the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(iv) DCK (Debug clock input) … input

This pin inputs a debug clock to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(vii) DMS (Debug mode select) … input

This pin selects the debug mode of the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(5) P70 to P711 (Port 7) … 3-state I/O

P70 to P711 function as a 12-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as the analog input pins of the A/D converter in the control mode. When using this port as analog input pins, however, set the port in the input mode. At this time, do not read the port.

(a) Port mode

P70 to P711 can be set in the input or output mode in 1-bit units, by using port mode register 7L, H (PM7L, PM7H).

(b) Control mode

P70 to P711 function alternately as the ANI0 to ANI11 pins.

(i) ANI0 to ANI11 (Analog input 0 to 11) … input

These pins input an analog signal to the A/D converter.

(6) P90, P91, P96 to P99, P913 to P915 (Port 9) … 3-state I/O

P90, P91, P96 to P99, P913 to P915 function as a 9-bit I/O port that can be set to input or output in 1-bit units. Besides functioning as an I/O port, these pins operate as serial interface I/O, timer/counter I/O, clock output, external interrupt request signal input, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units. The valid edge of P913 to P915 is specified by using INTF9H register.

An on-chip pull-up resistor can be connected to P90, P91, P96 to P99, P913 to P915 by using pull-up resistor option register 9 (PU9).

(a) Port mode

P90, P91, P96 to P99, P913 to P915 can be set in the input or output mode in 1-bit units, by using port 9 mode register (PM9).

(b) Control mode

(i) SIB1 (Serial input) … input

These pins input the serial receive data of CSIB1.

(ii) SOB1 (Serial output) … output

These pins output the serial receive data of CSIB1.

(iii) $\overline{\text{SCKB1}}$ (Serial clock) … 3-state I/O

These pins input/output the serial clock of CSIB1.

(iv) RXDA1 (Receive data) … input

This pin inputs the serial receive data of UARTA1.

(v) TXDA1 (Transmit data) … output

This pin outputs the serial transmit data of UARTA1.

(vi) TIP20, TIP21 (Timer input) … input

These pins input to timers P2 (TMP2).

(vii) TOP20, TOP21 (Timer output) … output

These pins output from timers P2 (TMP2).

(viii) PCL (Clock output) … output

This pin outputs a clock.

(ix) INTP4 to INTP6 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(x) KR6, KR7 (Key return) … input

These pins input a key interrupt. Their operation is specified by the key return mode register (KRM) in the input port mode.

(7) PCM0 to PCM3 (port CM) … 3-state I/O

PCM0 to PCM3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, and bus clock output.

(a) Port mode

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

(b) Control mode

(i) CLKOUT (Clock output) … output

This pin outputs an internally generated bus clock.

(13) PCS0, PCS1 (port CS) … 3-state I/O

PCS0, PCS1 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port.

(a) Port mode

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

(14) PCT0, PCT1, PCT4, PCT6 (port CT) … 3-state I/O

PCT0, PCT1, PCT4, PCT6 function as an 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port.

(a) Port mode

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

(8) PDL0 to PDL11 (port DL) … 3-state I/O

PDL0 to PDL11 function as a 12-bit I/O port that can be set to input or output in 1-bit units.

PDL5 also functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL11 can be set in the input or output mode in 1-bit units, by using port mode register DL (PMDL).

(9) $\overline{\text{RESET}}$ (Reset) … input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal with a fixed low level width is input to the $\overline{\text{RESET}}$ pin regardless of the operating clock, the system is reset, taking precedence over all the other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP), as well as for normal initialization/start.

(10) X1, X2 (Crystal for main clock)

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

(11) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

(12) AV_{SS} (Ground for analog)

This is a ground pin for the A/D converter, and alternate-function ports.

(13) AV_{REF0} (Analog reference voltage) … input

This pin supplies positive analog power to the A/D converter and alternate-function ports.

It also supplies a reference voltage to the A/D converter.

(14) EV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(15) EV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(16) V_{DD} (Power supply)

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

(17) V_{SS} (Ground)

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

★ **(18) FLMD0 (Flash programming mode) Input**

This is a signal input pin for flash memory programming mode. Connect this pin to VSS in the normal operation mode.

(19) REGC (Regulator control) … input

This pin connects a capacitor for the regulator.

2.3.3 V850ES/FG2

(1) P00 to P06 (port 0) … 3-state I/O

P00 to P06 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as NMI input, external interrupt request signal input, timer/counter I/O, external trigger of the A/D converter, and debug reset input.

This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR0 and INTF0 registers.

An on-chip pull-up resistor can be connected to P00 to P06 by using pull-up resistor option register 0 (PU0).

(a) Port mode

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

(b) Control mode

(i) NMI (Non-maskable interrupt request) … input

This pin inputs a non-maskable interrupt request signal.

(ii) INTP0 to INTP3 (Interrupt request from peripherals) … input

These pins input external interrupt request signals.

(iii) TIP30, TIP31 (Timer input) … input

These pins input an external count clock to timer P3 (TMP3).

(iv) TOP30, TOP31 (Timer output) … output

These pins output a pulse signal from timer P3 (TMP3).

(v) ADTRG (A/D trigger input) … input

This pin inputs an external trigger to the A/D converter. It is controlled by using A/D converter mode register 0 (ADA0M0).

(vi) $\overline{\text{DRST}}$ (Debug reset) … input

This pin inputs a debug reset signal, a negative-logic signal that asynchronously initializes the on-chip debug circuit. To deassert this signal, reset or invalidate the on-chip debug circuit. Deassert this signal when the debug function is not used.

For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(2) P10, P11 (port 1) … 3-state I/O

P10 and P11 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input in the control mode. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by INTR1 and INTF1 registers.

An on-chip pull-up resistor can be connected to P10 and P11 by using pull-up resistor option register 1 (PU1).

(a) Port mode

P10 and P11 can be set in the input or output mode in 1-bit units, by using port mode register 1 (PM1).

(b) Control mode

(i) INTP9, INTP10 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(3) P30 to P39 (port 3) … 3-state I/O

P30 to P39 function as a 10-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, serial interface I/O, timer/counter I/O, and CAN data I/O. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by using INTR3 and INTF3 registers.

An on-chip pull-up resistor can be connected to P30 to P39 by using pull-up resistor option register 3 (PU3).

(a) Port mode

P30 to P39 can be set in the input or output mode in 1-bit units, by using port mode register 3 (PM3).

(b) Control mode

(i) RXDA0, RXDA2 (Receive data) … input

This pin inputs the serial receive data of UARTA0.

(ii) TXDA0, TXDA2 (Transmit data) … output

This pin outputs the serial transmit data of UARTA0.

(iii) ASCKA0 (Asynchronous serial clock) … input

This pin inputs of UARTA0.

(iv) INTP7, INTP8 (Interrupt request from peripherals) … input

This pin inputs an external interrupt request signal.

(v) TIP00, TIP01, TIP10, TIP11 (Timer input) … input

These pins input an external count clock to timers P0, P1 (TMP0, TMP1).

(vi) TOP00, TOP01, TOP10, TOP11 (Timer output) … output

These pins output a pulse signal from timers P0, P1 (TMP0, TMP1).

(vii) CRXD0, CRXD1 (CAN receive data) … input

These pins input the receive data of CAN0 and CAN1.

(viii) CTXD0, CTXD1 (CAN transmit data) … output

These pins output the transmit data of CAN0 and CAN1.

(4) P40 to P42 (port 4) … 3-state I/O

P40 to P42 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P40 to P42 by using pull-up resistor option register 4 (PU4).

(a) Port mode

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

(b) Control mode

(i) SIB0 (Serial input) … input

This pin inputs the serial receive data of CSIB0.

(ii) SOB0 (Serial output) … output

This pin outputs the serial transmit data of CSIB0.

(iii) $\overline{\text{SCKB0}}$ (serial clock) … 3-state I/O

This pin inputs/outputs the serial clock of CSIB0.

(5) P50 to P55 (Port 5) … 3-state I/O

P50 to P55 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as an I/O port, but also as timer/counter I/O, debug function I/O, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P50 to P55 by using pull-up resistor option register 5 (PU5).

(a) Port mode

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

(b) Control mode

(i) KR0 to KR5 (Key return) … input

These pins input a key interrupt. Their operation is specified by using the key return mode register (KRM) in the input port mode.

(ii) TIQ00, TIQ01, TIQ02, TIQ03 (Timer input) … input

These pins input an external count clock to timer Q0 (TMQ0).

(iii) TOQ00, TOQ01, TOQ02, TOQ03 (Timer output) … output

These pins output a pulse signal from timer Q0 (TMQ0).

(iv) DDI (Debug data input) … input

This pin inputs debug data to the on-chip debug circuit.

For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(v) DDO (Debug data output) … output

This pin outputs debug data from the on-chip debug circuit.

For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(iv) DCK (Debug clock input) … input

This pin inputs a debug clock to the on-chip debug circuit.

For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(vii) DMS (Debug mode select) … input

This pin selects the debug mode of the on-chip debug circuit.

For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(6) P70 to P715 (Port 7) … 3-state I/O

P70 to P715 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as the analog input pins of the A/D converter in the control mode. When using this port as analog input pins, however, set the port in the input mode. At this time, do not read the port.

(a) Port mode

P70 to P715 can be set in the input or output mode in 1-bit units, by using port mode register 7L, H (PM7L, PM7H).

(b) Control mode

P70 to P715 function alternately as the ANI0 to ANI15 pins.

(i) ANI0 to ANI15 (Analog input 0 to 15) … input

These pins input an analog signal to the A/D converter.

(7) P90 to P915 (Port 9) … 3-state I/O

P90 to P915 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O, timer/counter I/O, clock output, external interrupt request signal input, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units. The valid edge of P913 to P915 is specified by using INTR9H and INTF9H registers.

An on-chip pull-up resistor can be connected to P90 to P915 by using pull-up resistor option register 9 (PU9).

(a) Port mode

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

(b) Control mode

(i) SIB1 (Serial input) … input

This pin inputs the serial receive data of CSIB1.

(ii) SOB1 (Serial output) … output

This pin outputs the serial receive data of CSIB1.

(iii) $\overline{\text{SCKB1}}$ (Serial clock) … 3-state I/O

This pin inputs/outputs the serial clock of CSIB1.

(iv) RXDA1 (Receive data) … input

This pin inputs the serial receive data of UARTA1.

(v) TXDA1 (Transmit data) … output

This pin outputs the serial transmit data of UARTA1.

(vi) TIP20, TIP21 (Timer input) … input

These pins input timers P2 (TMP2).

(vii) TOP20, TOP21 (Timer output) … output

These pins output a pulse signal from timers P2 (TMP2).

(viii) TIQ10, TIQ11, TIQ12, TIQ13 (Timer input) … input

These pins input to timers Q1 (TMQ1), respectively.

(ix) TOQ10, TOQ11, TOQ12, TOQ13 (Timer output) … output

These pins output a pulse signal from timers Q1 (TMQ1), respectively.

(x) PCL (Clock output) … output

This pin outputs a clock.

(xi) INTP4 to INTP6 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(xii) KR6, KR7 (Key return) … input

These pins input a key interrupt. Their operation is specified by key return mode register (KRM) in the input port mode.

(8) PCM0 to PCM3 (port CM) … 3-state I/O

PCM0 to PCM3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.
Besides functioning as an I/O port, these pins operate as bus clock output.

(a) Port mode

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

(b) Control mode

(i) CLKOUT (Clock output) … output

This pin outputs an internally generated bus clock.

(9) PCS0, PCS1 (Port CS) … 3-state I/O

PCS0 and PCS1 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

(a) Port mode

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

(10) PCT0, PCT1, PCT4, PCT6 (Port CT) … 3-state I/O

PCT0, PCT1, PCT4, and PCT6 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

(a) Port mode

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

(11) PDL0 to PDL13 (port DL) … 3-state I/O

PDL0 to PDL13 function as a 14-bit I/O port that can be set to input or output in 1-bit units.

PDL5 also functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLMD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

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

(12) $\overline{\text{RESET}}$ (Reset) … input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal with a fixed low level width is input to the $\overline{\text{RESET}}$ pin regardless of the operating clock, the system is reset, taking precedence over all the other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP), as well as for normal initialization/start.

(13) X1, X2 (Crystal for main clock)

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

(14) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

(15) AV_{SS} (Ground for analog)

This is a ground pin for the A/D converter, and alternate-function ports.

(16) AV_{REF0} (Analog reference voltage) … input

This pin supplies positive analog power to the A/D converter and alternate-function ports.

It also supplies a reference voltage to the A/D converter.

(17) EV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(18) EV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(19) V_{DD} (Power supply)

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

(20) V_{SS} (Ground)

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

★ **(21) FLMD0 (Flash programming mode) Input**

This is a signal input pin for flash memory programming mode. Connect this pin to VSS in the normal operation mode.

(22) BV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(23) BV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(24) REGC (Regulator control) … input

This pin connects a capacitor for the regulator.

2.3.4 V850ES/FJ2

(1) P00 to P06 (port 0) … 3-state I/O

P00 to P06 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as NMI input, external interrupt request signal input, timer/counter I/O, external trigger of the A/D converter, and debug reset input.

This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR0 and INTF0 registers.

An on-chip pull-up resistor can be connected to P00 to P06 by using pull-up resistor option register 0 (PU0).

(a) Port mode

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

(b) Control mode

(i) NMI (Non-maskable interrupt request) … input

This pin inputs a non-maskable interrupt request signal.

(ii) INTP0 to INTP3 (Interrupt request from peripherals) … input

These pins input external interrupt request signals.

(iii) TIP30, TIP31 (Timer input) … input

These pins input to timers P3 (TMP3).

(iv) TOP30, TOP31 (Timer output) … output

These pins output from timers P3 (TMP3).

(v) ADTRG (A/D trigger input) … input

This pin inputs an external trigger to the A/D converter. It is controlled by using A/D converter mode register 0 (ADA0M0).

(vi) $\overline{\text{DRST}}$ (Debug reset) … input

This pin inputs a debug reset signal, a negative-logic signal that asynchronously initializes the on-chip debug circuit. To deassert this signal, reset or invalidate the on-chip debug circuit. Deassert this signal when the debug function is not used.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(2) P10, P11 (port 1) … 3-state I/O

P10 and P11 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request input in the control mode. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by INTR1 and INTF1 registers.

An on-chip pull-up resistor can be connected to P10 and P11 by using pull-up resistor option register 1 (PU1).

(a) Port mode

P10 and P11 can be set in the input or output mode in 1-bit units, by using port mode register 1 (PM1).

(b) Control mode

(i) INTP9, INTP10 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(3) P30 to P39 (port 3) … 3-state output

P30 to P39 function as a 10-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, serial interface I/O, timer/counter I/O, and CAN data I/O. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by using INTR3 and INTF3 registers.

An on-chip pull-up resistor can be connected to P30 to P39 by using pull-up resistor option register 3 (PU3).

(a) Port mode

P30 to P39 can be set in the input or output mode in 1-bit units, by using port mode register 3 (PM3).

(b) Control mode

(i) RXDA0, RXDA2 (Receive data) … input

These pins input the serial receive data of UARTA0 and UARTA2.

(ii) TXDA0, TXDA2 (Transmit data) … output

These pins output the serial transmit data of UARTA0 and UARTA2.

(iii) ASCKA0 (Asynchronous serial clock) … input

This pin inputs of UARTA0.

(iv) INTP7, INTP8 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(v) TIP00, TIP01, TIP10, TIP11 (Timer input) … input

These pins input to timers P0 and P1 (TMP0, TMP1).

(vi) TOP00, TOP01, TOP10, TOP11 (Timer output) … output

These pins output from timers P0 and P1 (TMP0, TMP1).

(vii) CRXD0, CRXD1 (CAN receive data) … input

These pins input the receive data of CAN0 and CAN1.

(viii) CTXD0, CTXD1 (CAN transmit data) … output

These pins output the transmit data of CAN0 and CAN1.

(4) P40 to P42 (port 4) … 3-state I/O

P40 to P42 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P40 to P42 by using pull-up resistor option register 4 (PU4).

(a) Port mode

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

(b) Control mode

(i) SIB0 (Serial input) … input

This pin inputs the serial receive data of CSIB0.

(ii) SOB0 (Serial output) … output

This pin outputs the serial transmit data of CSIB0.

(iii) $\overline{\text{SCKB0}}$ (serial clock) … 3-state I/O

This pin inputs/outputs the serial clock of CSIB0.

(5) P50 to P55 (Port 5) … 3-state I/O

P50 to P55 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as an I/O port, but also as timer/counter I/O, debug function I/O, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P50 to P55 by using pull-up resistor option register 5 (PU5).

(a) Port mode

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

(b) Control mode

(i) KR0 to KR5 (Key return) … input

These pins input a key interrupt. Their operation is specified by using the key return mode register (KRM) in the input port mode.

(ii) TIQ00, TIQ01, TIQ02, TIQ03 (Timer input) … input

These pins input to timers Q0 (TMQ0).

(iii) TOQ00, TOQ01, TOQ02, TOQ03 (Timer output) … output

These pins output from timers Q0 (TMQ0).

(iv) DDI (Debug data input) … input

This pin inputs debug data to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(v) DDO (Debug data output) … output

This pin outputs debug data from the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(iv) DCK (Debug clock input) … input

This pin inputs a debug clock to the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(vii) DMS (Debug mode select) … input

This pin selects the debug mode of the on-chip debug circuit.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

(6) P60 to P615 … 3-state I/O

P60 to P615 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, timer/counter I/O, and CAN data I/O. P60 to P62 can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by INTR6L and INTF6L registers.

(a) Port mode

P60 to P65 can be set in the input or output mode in 1-bit units, by using port mode register 6 (PM6).

(b) Control mode

(i) INTP11 to INTP13 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(ii) TIQ20, TIQ21, TIQ22, TIQ23 (Timer input) … input

These pins input to timers Q2 (TMQ2).

(iii) TOQ20, TOQ21, TOQ22, TOQ23 (Timer output) … input

These pins output from timers Q2 (TMQ2).

(iv) CRXD2, CRXD3 (CAN receive data)[Note] … input

These pins input the receive data of CAN2 and CAN3.

(v) CTXD2, CTXD3 (CAN transmit data)[Note] … output

These pins output the transmit data of CAN2 and CAN3.

Note In the μPD70F3237, the alternate functions of the P65 to P68 pins (CTXD2, CRXD2, CTXD3, and CRXD3) are not available.

(7) P70 to P715 (Port 7) … 3-state I/O

P70 to P715 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as the analog input pins of the A/D converter in the control mode. When using this port as analog input pins, however, set the port in the input mode. At this time, do not read the port.

(a) Port mode

P70 to P715 can be set in the input or output mode in 1-bit units, by using port mode register 7L, H (PM7L, PM7H).

(b) Control mode

P70 to P715 function alternately as the ANI0 to ANI15 pins.

(i) ANI0 to ANI15 (Analog input 0 to 15) … input

These pins input an analog signal to the A/D converter.

(8) P80 and P81 (port 8) … 3-state I/O

P80 and P81 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as external interrupt request signal input, and serial interface I/O. P80 and P81 can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by INTR8 and INTF8 registers.

An on-chip pull-up resistor can be connected to P80 and P81 by using pull-up resistor option register 8 (PU8).

(a) Port mode

P80 and P81 can be set in the input or output mode in 1-bit units, by using port mode register 8 (PM8).

(b) Control mode

(i) INTP14 (Interrupt request from peripherals) … input

This pin inputs an external interrupt request signal.

(ii) RXDA3 (Receive data)[Note] … input

This pin inputs the serial receive data of UARTA3.

(iii) TXDA3 (Transmit data)[Note] … output

This pin outputs the serial transmit data of UARTA3.

Note In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available. The alternate function of the P80 pin in the μPD70F3237 is INTP14 only.

(9) P90 to P915 (Port 9) … 3-state I/O

P90 to P915 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O, timer/counter I/O, clock output, external interrupt request signal input, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units. The valid edge of P913 to P915 is specified by using INTF9H register.

An on-chip pull-up resistor can be connected to P90 to P915 by using pull-up resistor option register 9 (PU9).

(a) Port mode

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

(b) Control mode

(i) SIB1, SIB2 (Serial input) … input

These pins input the serial receive data of CSIB1 and CSIB2.

(ii) SOB1, SOB2 (Serial output) … output

These pins output the serial receive data of CSIB1 and CSIB2.

(iii) $\overline{\text{SCKB1}}$, $\overline{\text{SCKB2}}$ (Serial clock) … 3-state I/O

These pins input/output the serial clock of CSIB1 and CSIB2.

(iv) RXDA1 (Receive data) … input

This pin inputs the serial receive data of UARTA1.

(v) TXDA1 (Transmit data) … output

This pin outputs the serial transmit data of UARTA1.

(vi) TIP20, TIP21 (Timer input) … input

These pins input timers P2 (TMP2).

(vii) TOP20, TOP21 (Timer output) … output

These pins output from timers P2 (TMP2).

(viii) TIQ10, TIQ11, TIQ12, TIQ13 (Timer input) … input

These pins input an external count clock to timers Q1.

(ix) TOQ10, TOQ11, TOQ12, TOQ13 (Timer output) … output

These pins output a pulse signal from timers Q1.

(x) PCL (Clock output) … output

This pin outputs a clock.

(xi) INTP4 to INTP6 (Interrupt request from peripherals) … input

These pins input an external interrupt request signal.

(xii) KR6, KR7 (Key return) … input

These pins input a key interrupt. Their operation is specified by the key return mode register (KRM) in the input port mode.

(10) P120 to P127 (Port 12) … 3-state I/O

P120 to P127 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as the analog input pins of the A/D converter in the control mode. When using this port as analog input pins, however, set the port in the input mode. At this time, do not read the port.

(a) Port mode

P120 to P127 can be set in the input or output mode in 1-bit units, by using port mode register 12 (PM12).

(b) Control mode

P120 to P127 function alternately as the ANI16 to ANI23 pins.

(i) ANI16 to ANI23 (Analog input 16 to 23) … input

These pins input an analog signal to the A/D converter.

(11) PCD0 to PCD3 (port CD) … 3-state I/O

PCD0 to PCD3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

(a) Port mode

PCD0 to PCD3 can be set in the input or output mode in 1-bit units, by using port mode register CD (PMCD).

(12) PCM0 to PCM5 (port CM) … 3-state I/O

PCM0 to PCM5 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as bus hold control signal I/O, bus clock output, and a control signal that inserts a wait cycle in the bus cycle ($\overline{\text{WAIT}}$), in the control mode.

(a) Port mode

PCM0 to PCM5 can be set in the input or output mode in 1-bit units, by using port mode register CM (PMCM).

(b) Control mode

(i) $\overline{\text{HLDAK}}$ (Hold acknowledge) … output

This pin outputs an acknowledge signal that indicates that the V850ES/FJ2 has placed the address bus, data bus, and control bus in a high-impedance state in response to a bus hold request.

While this signal is active, the address bus, data bus, and control bus go into a high-impedance state.

(ii) $\overline{\text{HLDRQ}}$ (Hold request) … input

An external device uses this input pin to request the V850ES/FJ2 to release the address bus, data bus, and control bus. A signal can be input to this pin asynchronously to CLKOUT. When this pin is asserted, the V850ES/FJ2 places the address bus, data bus, and control bus in a high-impedance state after completion of a bus cycle under execution, if any, or immediately if no such bus cycle is under execution. The V850ES/FJ2 then asserts the $\overline{\text{HLDAK}}$ signal and releases the buses.

(iii) CLKOUT (Clock output) … output

This pin outputs an internally generated bus clock.

(iv) $\overline{\text{WAIT}}$ (Wait) … input

This is a control signal input pin that inserts a data wait state in the bus cycle. A signal can be input to this pin asynchronously to the CLKOUT signal. The signal input to this pin is sampled at the falling edge of the CLKOUT signal in the T2 and TW states of the bus cycle in the multiplexed mode. No wait state may be inserted if the setup/hold time of the sampling timing is not satisfied.

The wait function is set to on or off by port mode control register CM (PMCCM).

(13) PCS0 to PCS7 (port CS) … 3-state I/O

PCS0 to PCS7 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as chip select signal output in the control mode.

(a) Port mode

PCS0 to PCS7 can be set in the input or output mode in 1-bit units, by using port mode register CS (PMCS).

(b) Control mode

(i) $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ (Chip select input) … output

These pins output a chip select signal to external memory and external peripheral I/O.

The $\overline{\text{CS}}$n signal is assigned to memory block n (n = 0 to 3).

This signal is asserted while a bus cycle for accessing the corresponding memory block is being executed.

This signal is deasserted in the idle state (TI).

(14) PCT0 to PCT7 (port CT) … 3-state I/O

PCT0 to PCT7 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as control signal output in the control mode when memory is externally expanded.

(a) Port mode

PCT0 to PCT7 can be set in the input or output mode in 1-bit units, by using port mode register CT (PMCT).

(b) Control mode

(i) $\overline{\text{WR0}}$ (Lower byte write strobe) … output

This pin outputs the write strobe signal of the lower data of the external 16-bit data bus.

(ii) $\overline{\text{WR1}}$ (Upper byte write strobe) … output

This pin outputs the write strobe signal of the higher data of the external 16-bit data bus.

(iii) $\overline{\text{RD}}$ (Read strobe) … output

This pin outputs the read strobe signal of the external 16-bit data bus.

(iv) ASTB (Address strobe) … output

This pin outputs the latch strobe signal of the external address bus. The signal output from this pin goes low at the falling edge of the T1 state of the bus cycle, and goes high at the falling edge of the T3 state. It goes high while the bus cycle is not active.

(15) PDL0 to PDL15 (port DL) … 3-state I/O

PDL0 to PDL15 function as a 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as a time-division address/data bus (AD0 to AD15) when the memory is externally expanded.

PDL5/AD5 also functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL15 can be set in the input or output mode in 1-bit units, by using port mode register DL (PMDL).

(b) Control mode

(i) AD0 to AD15 (Address/Data Bus 0 to 15) … 3-state I/O

This is a multiplexed address/data bus for external access.

(16) $\overline{\text{RESET}}$ (Reset) … input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal with a fixed low level width is input to the $\overline{\text{RESET}}$ pin regardless of the operating clock, the system is reset, taking precedence over all the other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP), as well as for normal initialization/start.

(17) X1, X2 (Crystal for main clock)

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

(18) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

(19) AV_{SS} (Ground for analog)

This is a ground pin for the A/D converter, and alternate-function ports.

(20) AV_{REF0} (Analog reference voltage) … input

This pin supplies positive analog power to the A/D converter and alternate-function ports.

It also supplies a reference voltage to the A/D converter.

(21) EV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(22) EV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(23) V_{DD} (Power supply)

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

(24) V_{SS} (Ground)

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

★ **(25) FLMD0 (Flash programming mode) Input**

This is a signal input pin for flash memory programming mode. Connect this pin to VSS in the normal operation mode.

(26) BV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(27) BV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(28) REGC (Regulator control) … input

This pin connects a capacitor for the regulator.

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

2.4.1 V850ES/FE2

(1/2)

Pin	I/O Circuit Type	Recommended Connection
P00/TIP31/TOP31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P01/TIP30/TOP30		
P02/NMI		
P03/INTP0/ADTRG		
P04/INTP1		
P05/INTP2/$\overline{\text{DRST}}$	5-AF	Input: Independently connect to EV_{SS} Output: Leave open
P06/INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P30/TXDA0	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P31/RXDA0/INTP7	5-W	
P32/ASCKA0/TIP00/TOP00/ TOP01		
P33/TIP01/TOP01/CTXD0		
P34/TIP10/TOP10/CRXD0		
P35/TIP11/TOP11		
P40/SIB0	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P41/SOB0	5-A	
P42/$\overline{\text{SCKB0}}$	5-W	
P50/KR0/TIQ01/TOQ01	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P51/KR1/TIQ02/TOQ02		
P52/KR2/TIQ03/TOQ03/DDI		
P53/KR3/TIQ00/TOQ00/DDO		
P54/KR4/DCK		
P55/KR5/DMS		

(2/2)

Pin	I/O Circuit Type	Recommended Connection
P70/ANI0 to P79/ANI9	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open
P90/KR6/TXDA1	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P91/KR7/RXDA1		
P96/TIP21/TOP21		
P97/SIB1/TIP20/TOP20		
P98/SOB1	5-A	
P99/SCKB1	5-W	
P913/INTP4/PCL	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P914/INTP5		
P915/INTP6		
PCM0	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PCM1/CLKOUT		
PDL0 to PDL4	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PDL5/FLMD1		
PDL6, PDL7		
AV_{REF0}	–	Directly connect to V_{DD}
AV_{SS}	–	–
FLMD0[Note]	–	Directly connect to V_{SS}
REGC	–	–
$\overline{\text{RESET}}$	2	–
X1	–	–
X2	–	–
XT1	16	Connect to V_{SS} via a resistor
XT2	16	Leave open
V_{DD}	–	–
V_{SS}	–	–
EV_{DD}	–	–
EV_{SS}	–	–

Note If noise that exceeds the noise elimination width is input to the $\overline{\text{RESET}}$ pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin.

2.4.2 V850ES/FF2

(1/2)

Pin	I/O Circuit Type	Recommended Connection
P00/TIP31/TOP31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P01/TIP30/TOP30		
P02/NMI		
P03/INTP0/ADTRG		
P04/INTP1		
P05/INTP2/$\overline{\text{DRST}}$	5-AF	Input: Independently connect to EV_{SS} Output: Leave open
P06/INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P30/TXDA0	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P31/RXDA0/INTP7	5-W	
P32/ASCKA0/TIP00/TOP00/ TOP01		
P33/TIP01/TOP01/CTXD0		
P34/TIP10/TOP10/CRXD0		
P35/TIP11/TOP11		
P38	5-A	
P39	5-W	
P40/SIB0	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P41/SOB0	5-A	
P42/$\overline{\text{SCKB0}}$	5-W	
P50/KR0/TIQ01/TOQ01	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P51/KR1/TIQ02/TOQ02		
P52/KR2/TIQ03/TOQ03/DDI		
P53/KR3/TIQ00/TOQ00/DDO		
P54/KR4/DCK		
P55/KR5/DMS		

(2/2)

Pin	I/O Circuit Type	Recommended Connection
P70/ANI0 to P711/ANI11	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open
P90/KR6/TXDA1	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P91/KR7/RXDA1		
P96/TIP21/TOP21		
P97/SIB1/TIP20/TOP20		
P98/SOB1	5-A	
P99/SCKB1	5-W	
P913/INTP4/PCL	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P914/INTP5		
P915/INTP6		
PCM0	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PCM1/CLKOUT		
PCM2, PCM3		
PCS0, PCS1	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PCT0. PCT1, PCT4, PCT6	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PDL0 to PDL4	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
PDL5/FLMD1		
PDL6 to PDL11		
AV_{REF0}	–	Directly connect to V_{DD}
AV_{SS}	–	–
FLMD0[Note]	–	Directly connect to V_{SS}
REGC	–	–
$\overline{\text{RESET}}$	2	–
X1	–	–
X2	–	–
XT1	16	Connect to V_{SS} via a resistor
XT2	16	Leave open
V_{DD}	–	–
V_{SS}	–	–
EV_{DD}	–	–
EV_{SS}	–	–

Note If noise that exceeds the noise elimination width is input to the $\overline{\text{RESET}}$ pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin.

2.4.3 V850ES/FG2

(1/2)

Pin	I/O Circuit Type	Recommended Connection
P00/TIP31/TOP31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P01/TIP30/TOP30		
P02/NMI		
P03/INTP0/ADTRG		
P04/INTP1		
P05/INTP2/$\overline{\text{DRST}}$	5-AF	Input: Independently connect to EV_{SS} via a resistor Output: Leave open
P06/INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P10/INTP9	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P11/INTP10		
P30/TXDA0	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P31/RXDA0/INTP7	5-W	
P32/ASCKA0/TIP00/TOP00/TOP01		
P33/TIP01/TOP01/CTXD0		
P34/TIP10/TOP10/CRXD0		
P35/TIP11/TOP11		
P36/CTXD1	5-A	
P37/CRXD1	5-W	
P38/TXDA2	5-A	
P39/RXDA2/INTP8	5-W	
P40/SIB0	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P41/SOB0	5-A	
P42/$\overline{\text{SCKB0}}$	5-W	
P50/KR0/TIQ01/TOQ01		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P51/KR1/TIQ02/TOQ02		
P52/KR2/TIQ03/TOQ03/DDI		
P53/KR3/TIQ00/TOQ00/DDO		
P54/KR4/DCK		
P55/KR5/DMS		
P70/ANI0 to P711/ANI11	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open
P712/ANI12 to P715/ANI15		
P90/KR6/TXDA1	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P91/KR7/RXDA1		
P92/TIQ11/TOQ11		
P93/TIQ12/TOQ12		
P94/TIQ13/TOQ13		
P95/TIQ10/TOQ10		

(2/2)

Pin	I/O Circuit Type	Recommended Connection
P96/TIP21/TOP21	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P97/SIB1/TIP20/TOP20		
P98/SOB1	5-A	
P99/$\overline{\text{SCKB1}}$	5-W	
P910	5-A	
P911		
P912		
P913/INTP4/PCL	5-W	
P914/INTP5		
P915/INTP6		
PCM0	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCM1/CLKOUT		
PCM2, PCM3		
PCS0, PCS1	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCT0, PCT1, PCT4, PCT6	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PDL0 to PDL4	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PDL5/ FLMD1		
PDL6 to AD13		
AV_{REF0}	–	Directly connect to V_{DD}
AV_{SS}	–	–
FLMD0[Note]	–	Directly connect to V_{SS}
REGC	–	–
$\overline{\text{RESET}}$	2	–
X1	–	–
X2	–	–
XT1	16	Connect to V_{SS} via a resistor
XT2	16	Leave open
V_{DD}	–	–
V_{SS}	–	–
BV_{DD}	–	–
BV_{SS}	–	–
EV_{DD}	–	–
EV_{SS}	–	–

Note If noise that exceeds the noise elimination width is input to the $\overline{\text{RESET}}$ pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin.

2.4.4 V850ES/FJ2

(1/4)

Pin	I/O Circuit Type	Recommended Connection
P00/TIP31/TOP31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P01/TIP30/TOP30		
P02/NMI		
P03/INTP0/ADTRG		
P04/INTP1		
P05/INTP2/$\overline{\text{DRST}}$	5-AF	Input: Independently connect to EV_{SS} Output: Leave open
P06/INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P10/INTP9	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P11/INTP10		
P30/TXDA0	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P31/RXDA0/INTP7	5-W	
P32/ASCKA0/TIP00/TOP00/ TOP01		
P33/TIP01/TOP01/CTXD0		
P34/TIP10/TOP10/CRXD0		
P35/TIP11/TOP11		
P36/CTXD1	5-A	
P37/CRXD1	5-W	
P38/TXDA2[Note]	5-A	
P39/RXDA2/INTP8[Note]	5-W	
P40/SIB0	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P41/SOB0	5-A	
P42/$\overline{\text{SCKB0}}$	5-W	
P50/KR0/TIQ01/TOQ01	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P51/KR1/TIQ02/TOQ02		
P52/KR2/TIQ03/TOQ03/DDI		
P53/KR3/TIQ00/TOQ00/DDO		
P54/KR4/DCK		
P55/KR5/DMS		

(2/4)

Pin	I/O Circuit Type	Recommended Connection
P60/INTP11	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P61/INTP12		
P62/INTP13		
P63	5-A	
P64		
P65/CTXD2[Note1]		
P66/CRXD2[Note1]	5-W	
P67/CTXD3[Note1]	5-A	
P68/CRXD3[Note1]	5-W	
P69	5-A	
P610/TIQ20/TOQ20	5-W	
P611/TIQ21/TOQ21		
P612/TIQ22/TOQ22		
P613/TIQ23/TOQ23		
P614	5-A	
P615		
P70/ANI0 to P79/ANI9	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open
P710/ANI10, P11/ANI11		
P712/ANI12 to P715/ANI15		
P80/RXDA3/INTP14[Note2]	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P81/TXDA3[Note2]	5-A	
P90/KR6/TXDA1	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P91/KR7/RXDA1		
P92/TIQ11/TOQ11		
P93/TIQ12/TOQ12		
P94/TIQ13/TOQ13		
P95/TIQ10/TOQ10		
P96/TIP21/TOP21		
P97/SIB1/TIP20/TOP20		
P98/SOB1	5-A	
P99/$\overline{\text{SCKB1}}$	5-W	
P910/SIB2		
P911/SOB2	5-A	
P912/$\overline{\text{SCKB2}}$	5-W	

Notes 1. In the μPD70F3237, the alternate functions of the P65 to P68 pins (CTXD2, CRXD2, CTXD3, and CRXD3) are not available.

2. In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available.

The alternate function of the P80 pin in the μPD70F3237 is only INTP14.

(3/4)

Pin	I/O Circuit Type	Recommended Connection
P913/INTP4/PCL	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P914/INTP5		
P915/INTP6		
P120/ANI16 to P127/ANI23	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open
PCD0 to PCD3	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCM0/$\overline{\text{WAIT}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCM1/CLKOUT		
PCM2/$\overline{\text{HLDAK}}$		
PCM3/$\overline{\text{HLDRQ}}$		
PCM4		
PCM5		
PCS0/$\overline{\text{CS0}}$, PCS3/$\overline{\text{CS1}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCS0/$\overline{\text{CS2}}$, PCS3/$\overline{\text{CS3}}$, PCS4 to PCS7		
PCT0/$\overline{\text{WR0}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PCT1/$\overline{\text{WR1}}$		
PCT2		
PCT3		
PCT4/$\overline{\text{RD}}$		
PCT5		
PCT6/ASTB		
PCT7		
PDL0/AD0 to PDL4/AD4	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
PDL5/AD5/FLMD1		
PDL6/AD6, PDL7/AD7		
PDL8/AD8 to PDL11/AD11		
PDL12/AD12, PDL13/AD13		
PDL14/AD14, PDL15/AD15		

(4/4)

Pin	I/O Circuit Type	Recommended Connection
AV_{REF0}	–	Directly connect to V_{DD}
AV_{SS}	–	–
FLMD0[Note]	–	Directly connect to V_{SS}
REGC	–	–
$\overline{RESET}$	2	–
X1	–	–
X2	–	–
XT1	16	Connect to V_{SS} via a resistor
XT2	16	Leave open
V_{DD}	–	–
V_{SS}	–	–
BV_{DD}	–	–
BV_{SS}	–	–
EV_{DD}	–	–
EV_{SS}	–	–

Note If noise that exceeds the noise elimination width is input to the $\overline{RESET}$ pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin.

2.5 Pin I/O Circuits

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

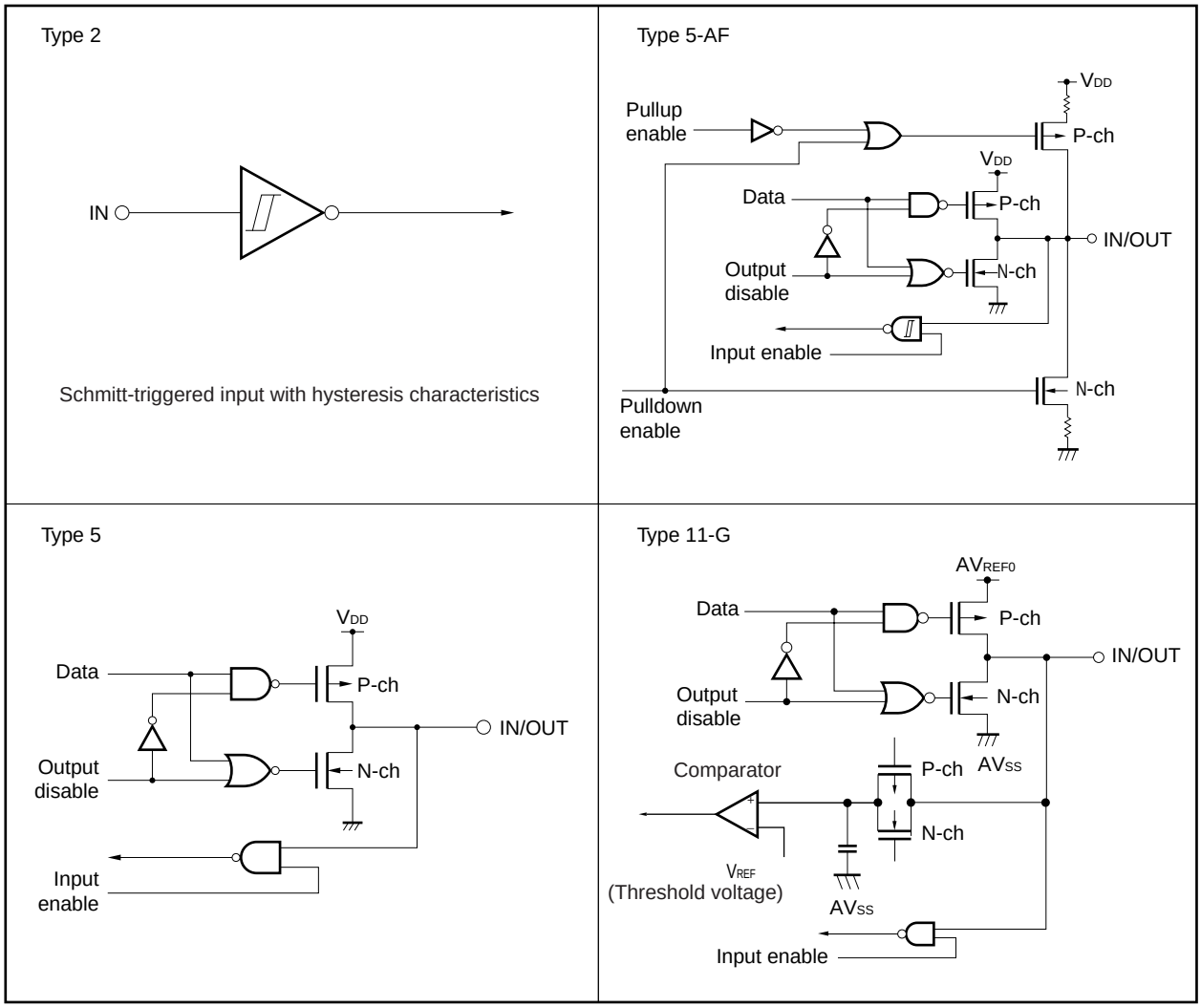
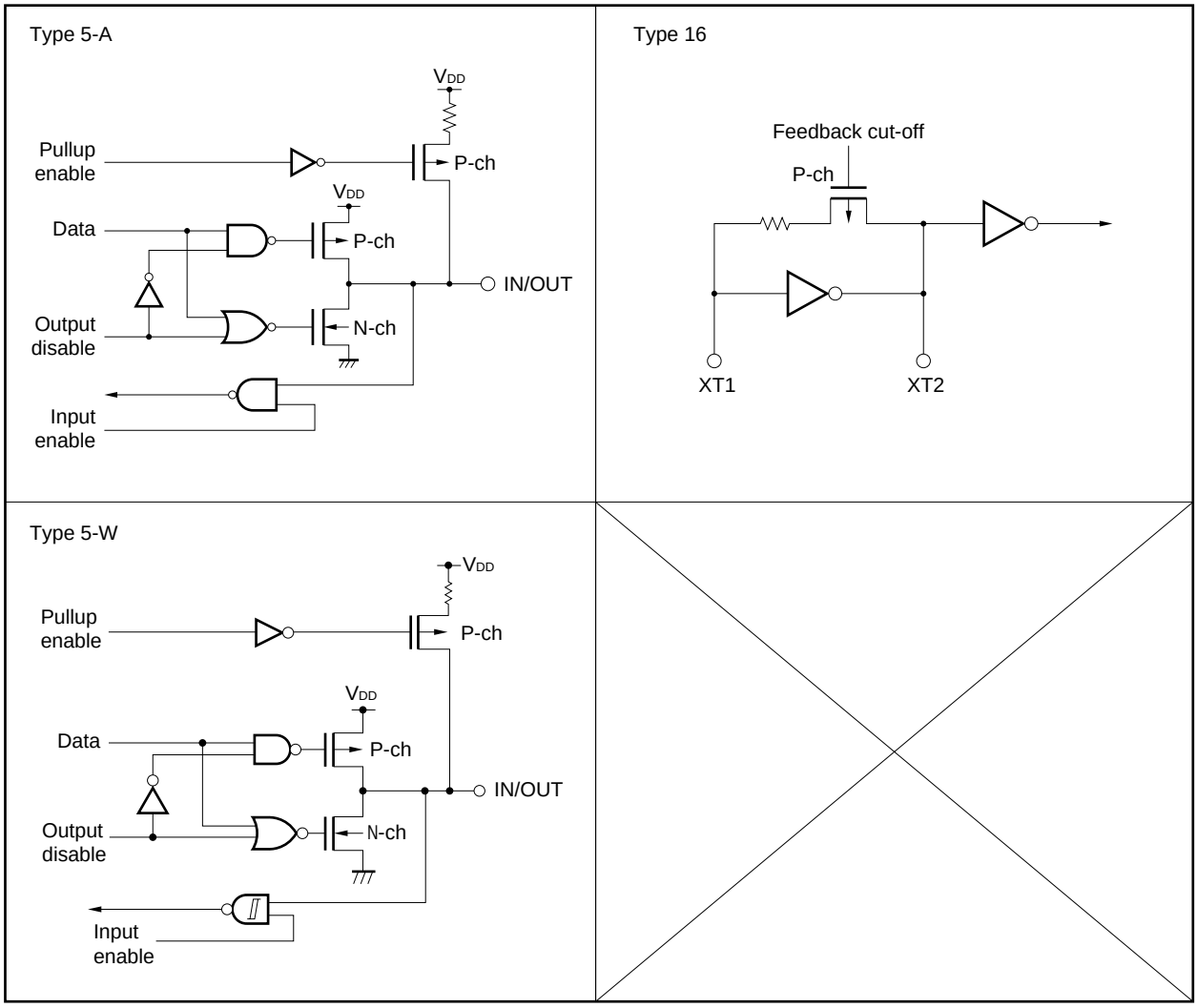


Figure 2-1. Pin I/O Circuit Types (2/2)



CHAPTER 3 CPU FUNCTIONS

Based on the RISC architecture, the CPU of the V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 executes most of the instructions in one clock under control of a five-stage pipeline.

3.1 Features

- Minimum instruction execution time: 50 ns (at 20 MHz operation)
- Memory space Program space: 64 MB, linear
  Data space: 4 GB, linear
- General-purpose registers: 32 bits × 32
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturation operation instructions
- 32-bit shift instructions: 1 clock
- Load/store instructions with long/short format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 can be classified into two types: general-purpose program registers and dedicated system registers. All the registers are 32 bits wide.
For details, refer to **V850ES Architecture User's Manual**.

(1) Program register set

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

31		0
PC	(Program counter)	

(2) System register set

31		0
EIPC	(Status saving register on interrupt)	
EIPSW	(Status saving register on interrupt)	
FEPC	(Status saving register on NMI)	
FEPSW	(Status saving register on NMI)	
ECR	(Interrupt source register)	
PSW	(Program status word)	
CTPC	(Status saving register on CALLT execution)	
CTPSW	(Status saving register on CALLT execution)	
DBPC	(Status saving register on exception/debug trap)	
DBPSW	(Status saving register on exception/debug trap)	
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 for a data variable or address variable.

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. Register r0 always holds 0, and is used for an operation using 0 or addressing with offset 0. Register r30 is used as a base pointer when the SLD or SST instruction is used to access the memory. Registers r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. When using these registers, therefore, their contents must be saved so that they are not lost, and later restored to the registers. Register r2 may be used by a real-time OS. If the real-time OS used does not use r2, r2 can be used as a register for variables.

Table 3-1. Program Register List

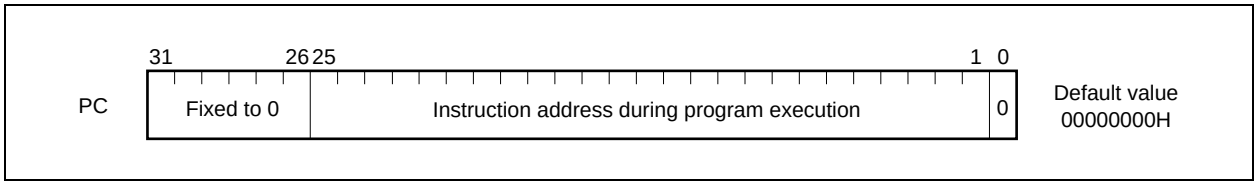
Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as a working register for creating 32-bit immediate
r2	Register for address/data variable (if the real-time OS used does not use r2)	
r3	Stack pointer	Used to generate a stack frame when a function is called
r4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

Remark For further details on the r1, r3 to r5, and r31that are used in the assembler and C compiler, refer to the CA850 (C Compiler Package) Assembly Language of the user's Manual.

(2) Program counter (PC)

The program counter holds an instruction address during program execution. The lower 26 bits of this counter are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to bit 26 is ignored.

Bit 0 is always fixed to 0, and execution cannot branch to an odd address.



3.2.2 System register set

The system registers are used to control the status of the CPU or to hold interrupt information.

Data can be read from or written to system registers by setting one of the system register numbers listed below using a system register load or store (LDSR or STSR) instruction.

Table 3-2. System Register Numbers

Register No.	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)	√	√
17	CALLT execution status saving register (CTPSW)	√	√
18	Exception/debug trap status saving register (DBPC)	√[Note 2]	√
19	Exception/debug trap status saving register (DBPSW)	√[Note 2]	√
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion. (Operation is not guaranteed if these registers are accessed.)	×	×

Notes 1. Because only one pair of these registers is provided, the contents of these registers must be saved by program when multiple interrupt servicing is enabled.

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

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

Remark √: Accessible

×: Access prohibited

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

EIPC and EIPSW are interrupt status saving registers.

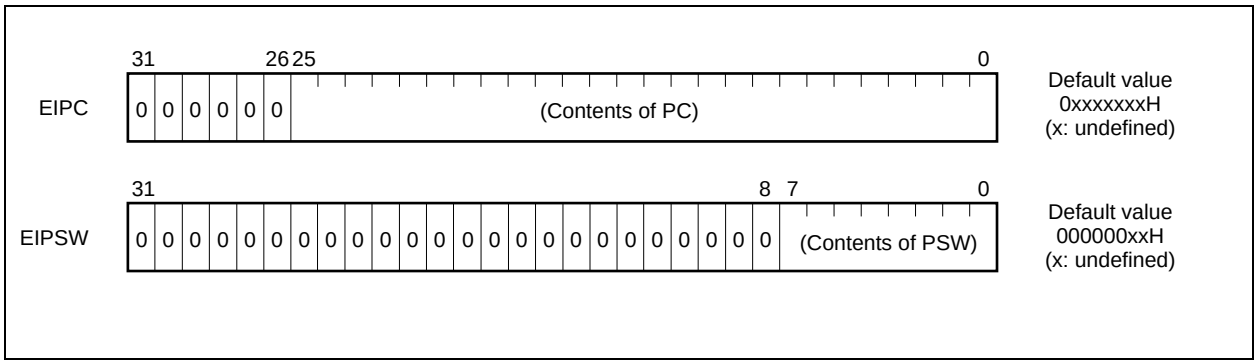
If a software exception or a maskable interrupt occurs, the contents of the program counter (PC) are saved to EIPC, and the contents of the program status word (PSW) are saved to EIPSW. (The contents of the PC and PSW are saved to FEPC and FEPSW (NMI status saving registers) if a non-maskable interrupt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

The address of the instruction next to the one under execution, except some instructions (see 17.7 Periods in Which Interrupts Are Not Acknowledged by CPU), is saved to EIPC when a software exception or a maskable interrupt occurs.

The current contents of the PSW are saved to EIPSW.

Because only one pair of interrupt status saving registers is available, the contents of these registers must be saved by program if multiple interrupt servicing is enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved for future function expansion (these bits are fixed to 0). The values of EIPC restore the PC, and the values of EIPSW do to the PSW by the RETI instruction.



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

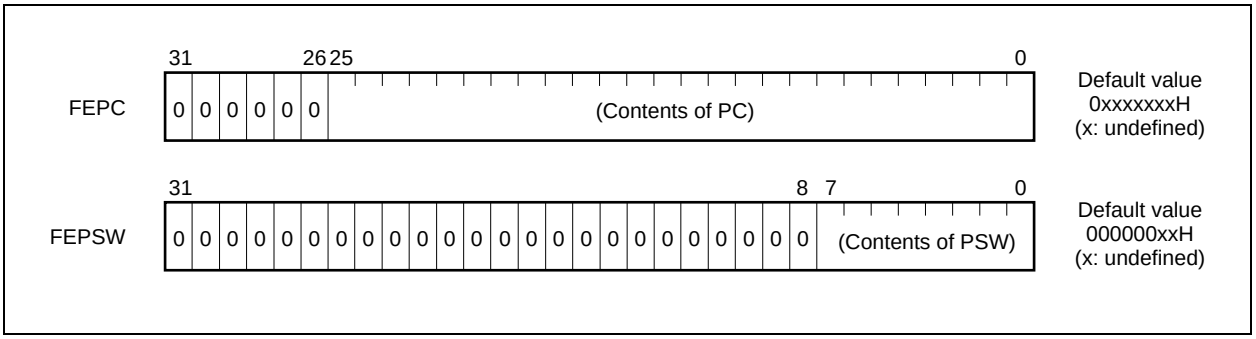
FEPC and FEPSW are NMI status saving registers.

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

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

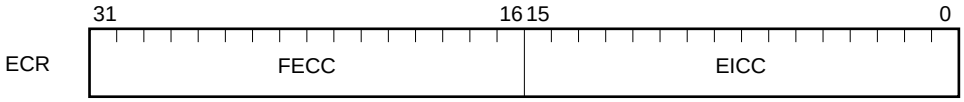
The current contents of the PSW are saved to FEPSW.

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



(3) Interrupt source register (ECR)

The interrupt source register ECR holds the source of an exception or an interrupt that has occurred. The value ECR is to hold is an exception code for each interrupt source. This register is a read-only register. No data can be written to this register by using the LDSR instruction.



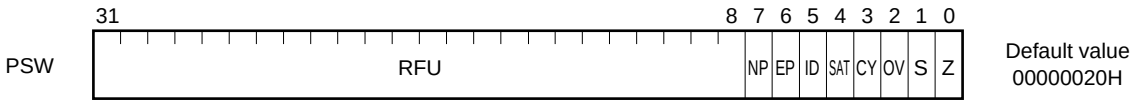
Bit Position	Bit Name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

(4) Program status word (PSW)

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

If the contents of any bit of this register are changed by using the LDSR instruction, the new contents become valid immediately after execution of the LDSR instruction. When the ID flag is set to 1, however, acknowledgment of an interrupt request is disabled from when the LDSR instruction is still under execution.

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



Bit Position	Flag Name	Meaning
31 to 8	RFU	Reserved field. Always fixed to 0
7	NP	Indicates that a non-maskable interrupt (NMI) is being serviced. This flag is set to 1, disabling multiple interrupt servicing, when an NMI request is acknowledged. 0: NMI is not being serviced. 1: NMI is being serviced.
6	EP	Indicates that exception processing is in progress. This flag is set to 1 when an exception is generated. Interrupt requests are acknowledged even if this bit is set. 0: Exception is not being processed. 1: Exception is being processed.
5	ID	Indicates whether a maskable interrupt request can be acknowledged. 0: Interrupts enabled 1: Interrupts disabled
4	SAT[Note]	Indicates that the result of an operation of a saturation operation instruction overflows and that the operation result is saturated. Because this is a cumulative flag, it is set to 1 if the operation result of a saturation operation instruction is saturated, and is not cleared to 0 even if the operation result of the subsequent instructions is not saturated. This flag is cleared to 0 by the LDSR instruction. When an arithmetic operation instruction is executed, this flag is neither set to 1 nor cleared to 0. 0: Not saturated 1: Saturated
3	CY	Indicates whether a carry or a borrow occurred as a result of an operation. 0: Carry or borrow did not occur. 1: Carry or borrow occurred.
2	OV[Note]	Indicates whether an overflow occurred during an operation. 0: Overflow did not occur. 1: Overflow occurred.
1	S[Note]	Indicates whether the result of an operation is negative or not. 0: Operation result is positive or 0. 1: Operation result is negative.
0	Z	Indicates whether the result of an operation is 0 or not. 0: Result of operation is not 0. 1: Result of operation is 0.

Remark Refer to the next page for the explanation of **Note**.

(2/2)

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

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 not exceeded)	Holds value before operation.	0	0	Operation result itself
Negative (maximum value not exceeded)			1	

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

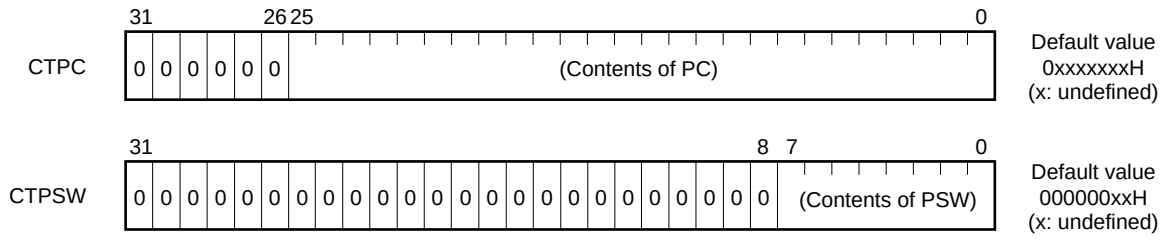
CTPC and CTPSW are CALLT execution status saving registers.

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

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

The current contents of the PSW are saved to CTPSW.

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



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

DBPC and DBPSW are exception/debug trap status saving registers.

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

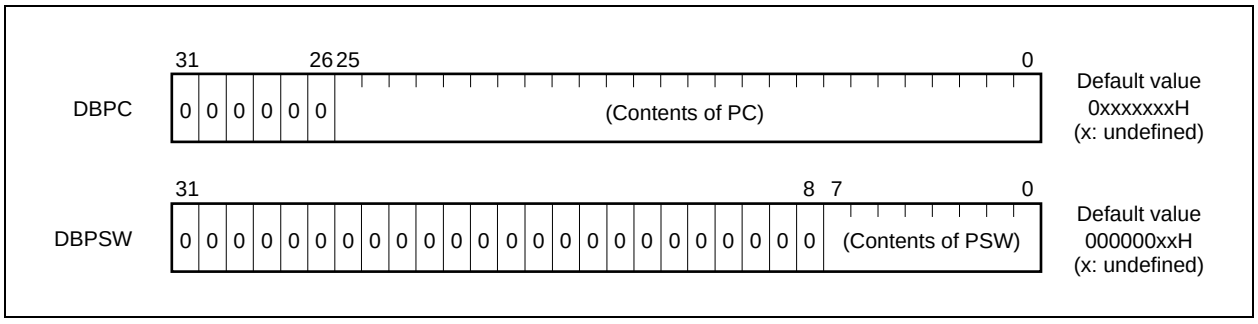
The contents saved to DBPC are the address of the instruction next to the one under execution when an exception trap or a debug trap has occurred.

The current contents of the PSW are saved to DBPSW.

These registers can be accessed only between DBTRAP or the illegal instruction execution and DBRET instruction execution.

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

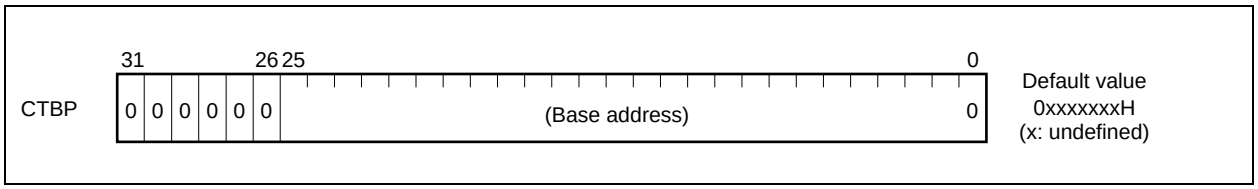
The values of DBPC restore PC, and the values of DBPSW do to PSW by DBRET instruction.



(7) CALLT base pointer (CTBP)

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

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



3.3 Operation Modes

The V850ES/Fx2 have the following operation modes.

FLMD0	FLMD1	Operation Mode
0	X	Normal operation mode
1	0	Flash memory programming mode
1	1	Setting prohibited

Remark x: don't care

(1) Normal operation mode

After system reset is released, each pin related to the bus interface is set in the port mode, execution branches to the reset entry address of the internal ROM, and instruction processing is started. When the PMCDL, PMCCM, PMCCS, and PMCCT registers are set in the control mode by software, an external device can be connected to the external memory area.

(2) Flash memory programming mode

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

(3) On-chip debug mode

The V850ES/FJ2 is provided with an on-chip debug function that employ the JTAG (Joint Test Action Group) communication specifications and that is executed via an N-Wire emulator. For details, see CHAPTER 27 ON-CHIP DEBUG FUNCTION."

3.3.1 Specifying operation mode

Specify the operation mode by using the FLMD0 and FLMD1 pins. In the normal mode, make sure that the FLMD0/IC pin goes low when reset is released.

In the flash memory programming mode, a high level is input to the FLMD0 pin from the flash programmer if a flash programmer is connected, but it must be input from an external circuit in the self-programming mode.

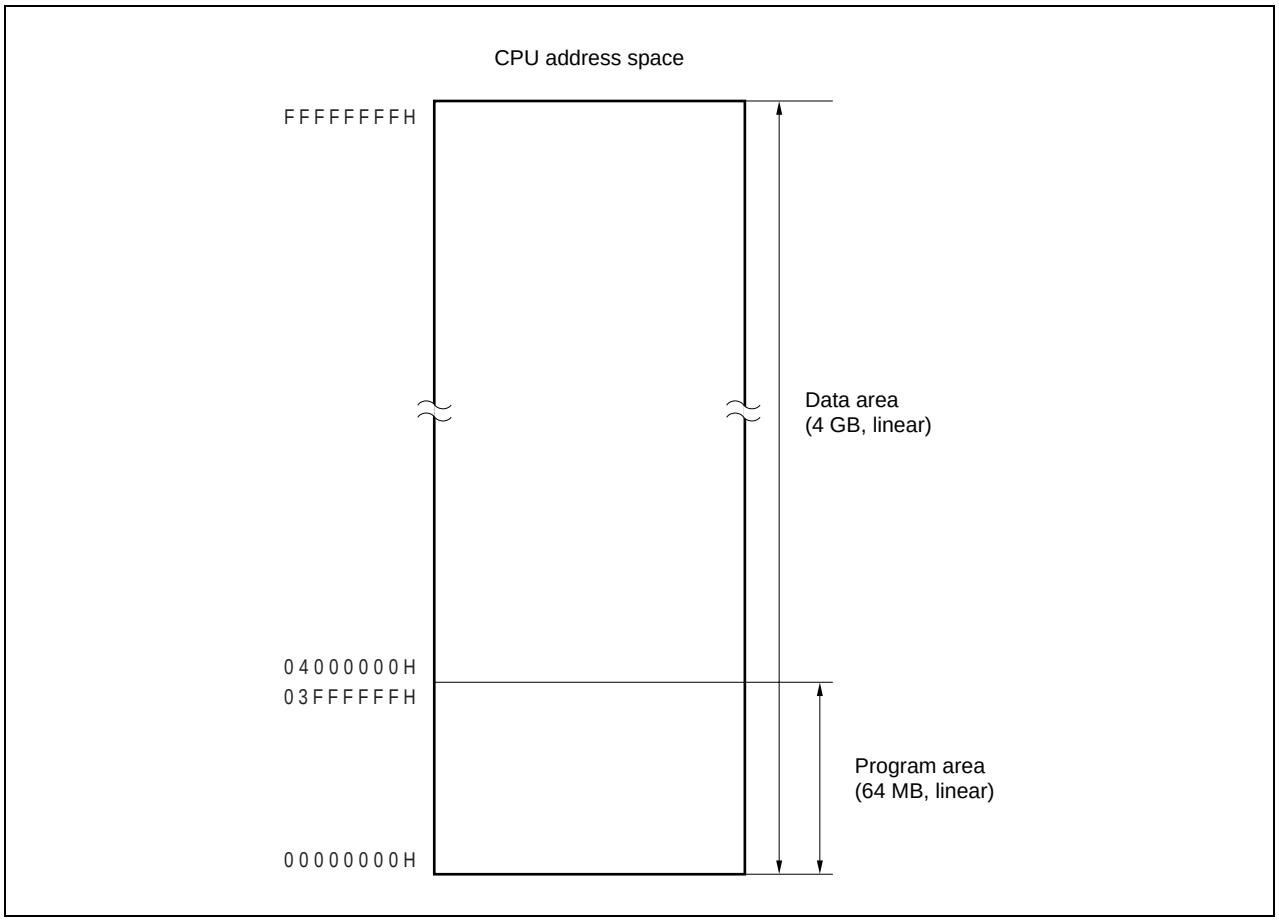
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850ES/Fx2 has 32-bit architecture, and supports a linear address space (data space) of up to 4 GB for operand addressing (data access). It also supports a linear address space (program space) of up to 64 MB for addressing instruction addresses. However, both the program and data spaces have areas prohibited from being used. For details, refer to **Figure 3-2**.

Figure 3-1 illustrates the CPU address space.

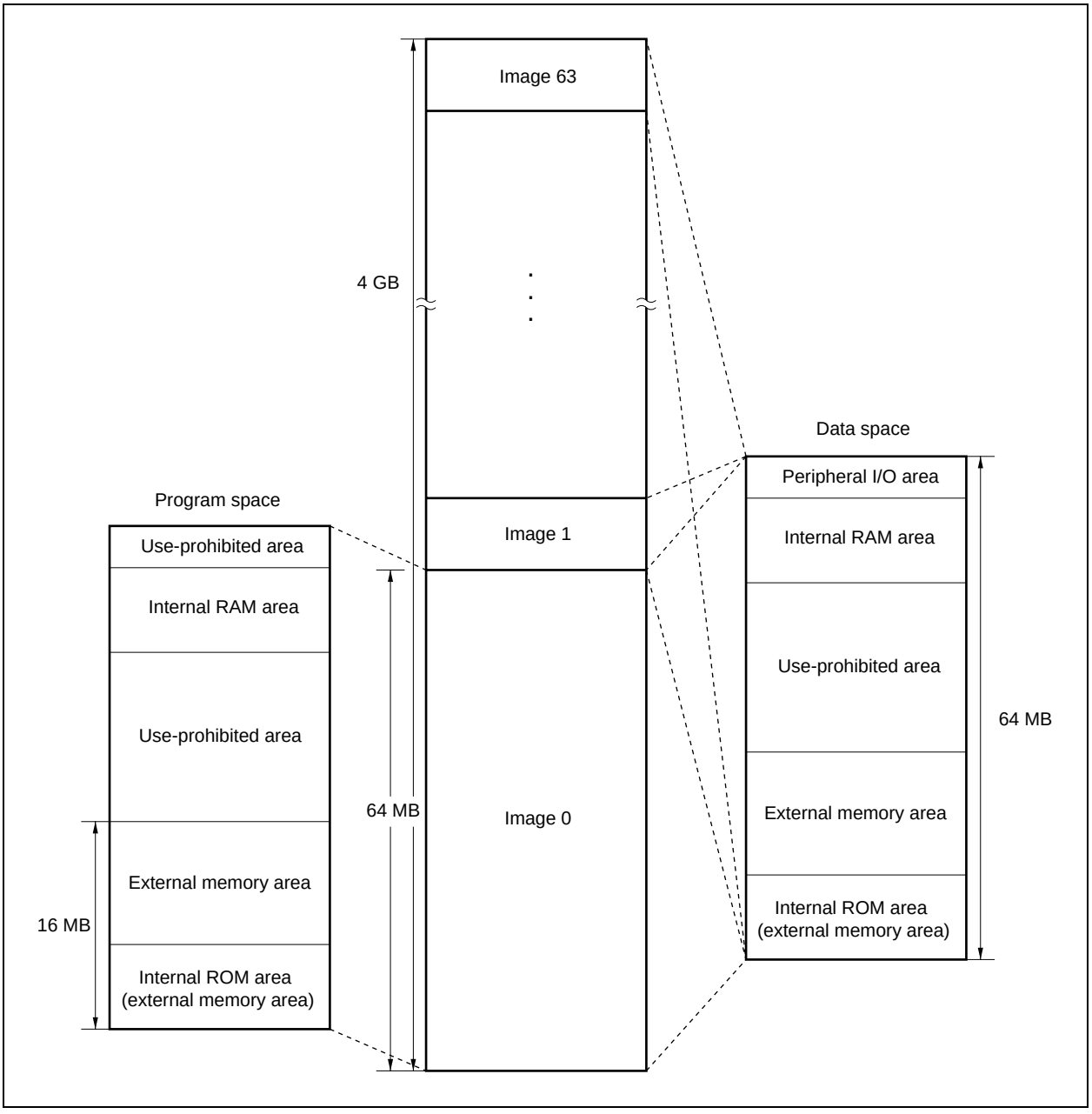
Figure 3-1. CPU Address Space



3.4.2 Image

Up to 16 MB of external memory area, internal ROM area, and internal RAM area of up to 16 MB of linear address space (program space) are supported for addressing of instruction addresses. Up to 4 GB of linear address space (data space) are supported for operand addressing (data access). Note, however, that there seems to be sixty-four 64 MB physical address spaces on the 4 GB address space. This means that the same 64 MB physical address space is accessed regardless of the value of bits 31 to 26.

Figure 3-2. Image on Address Space



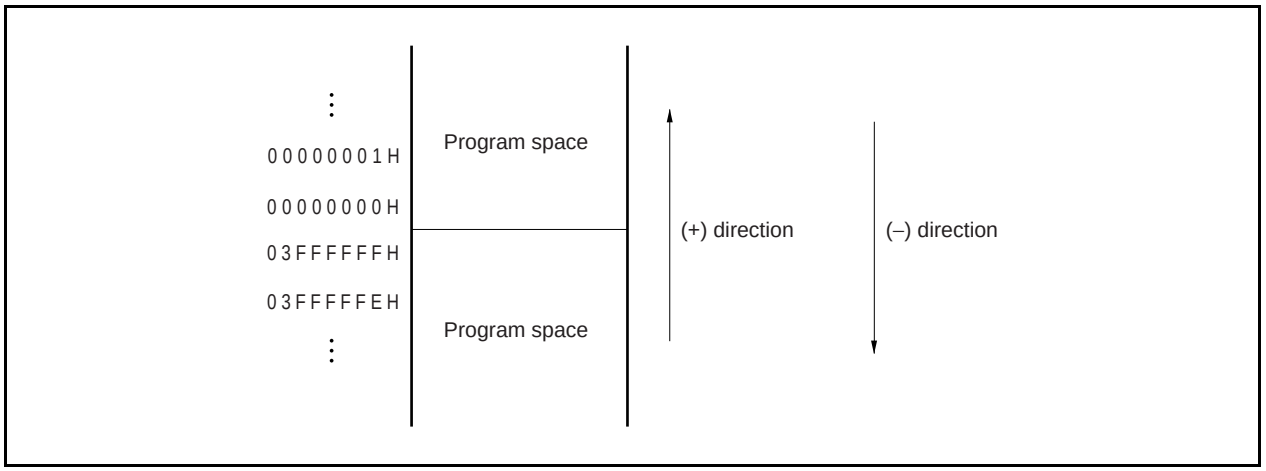
3.4.3 Wraparound of CPU address space

(1) Program space

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

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

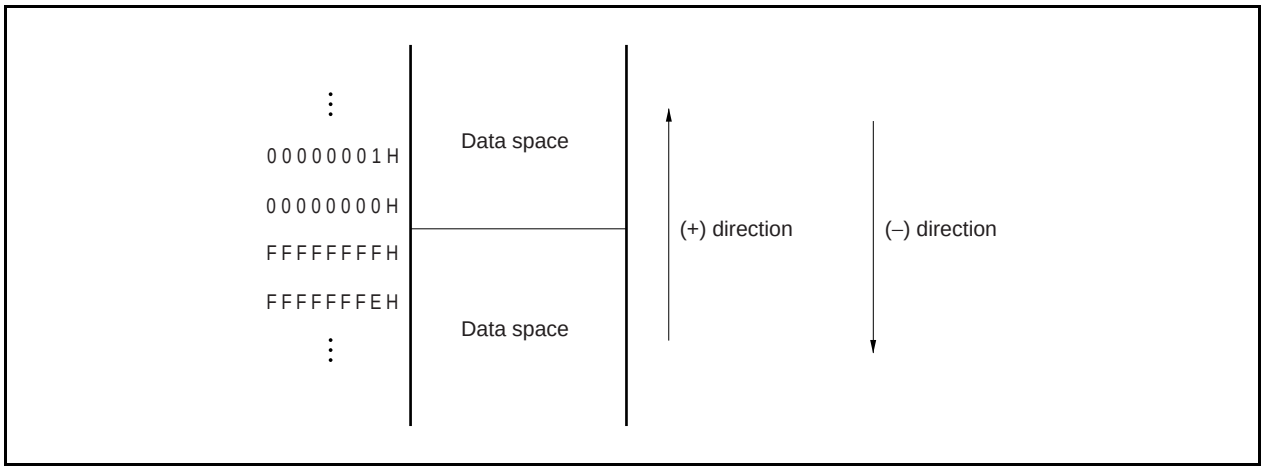
Caution Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is an on-chip peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.



(2) Data space

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

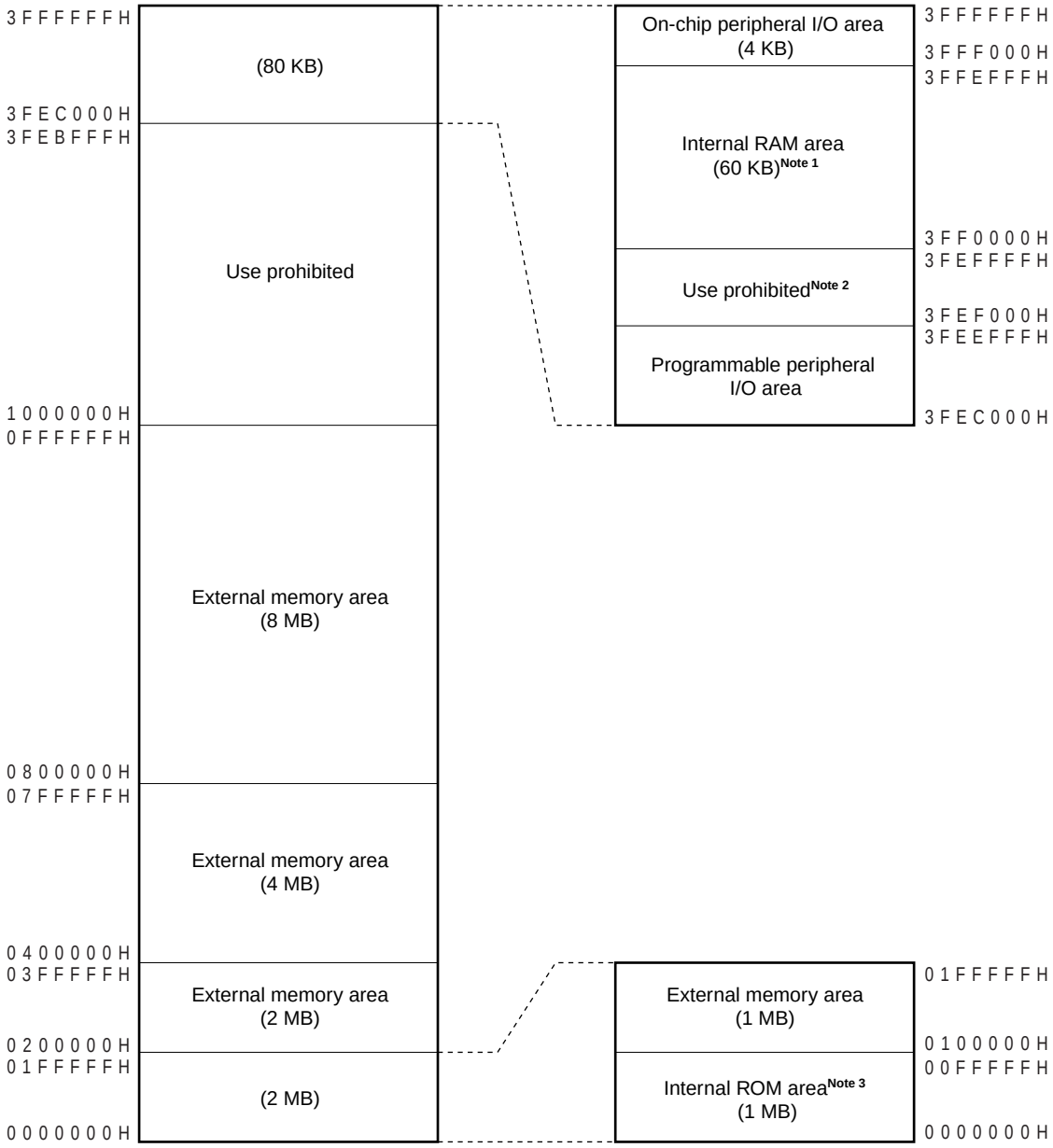
Therefore, the highest address of the data space, FFFFFFFFH, and the lowest address, 00000000H, are contiguous, and wraparound occurs at the boundary of these addresses.



3.4.4 Memory map

The V850ES/Fx2 reserves the areas shown in **Figure 3-3**.

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

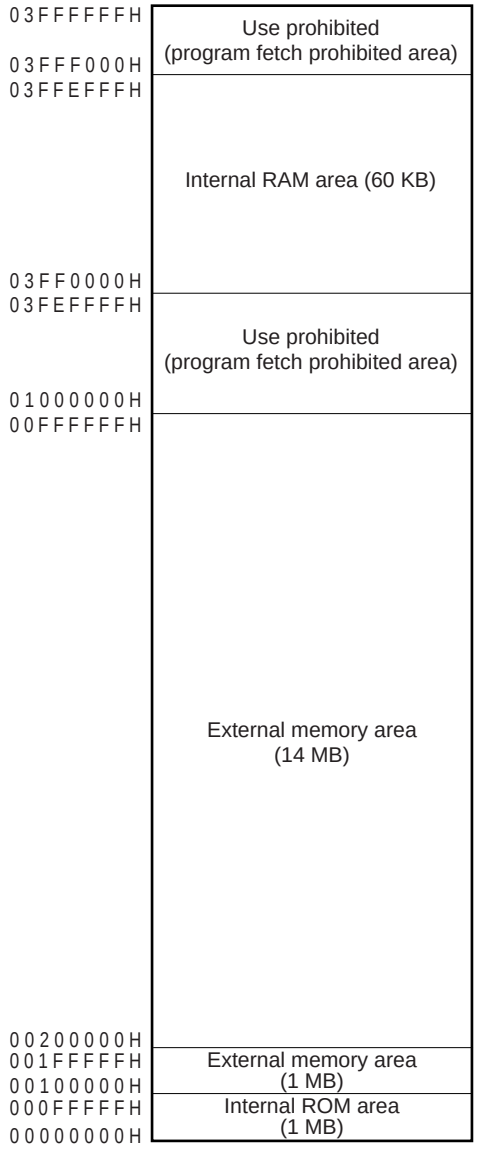


Notes 1. V850ES/FE2: µPD703230: 4K, µPD70F3231: 6K are provided.
V850ES/FF2: µPD703232: 6K , µPD70F3232: 12K , µPD703232: 12K are provided
V850ES/FG2: µPD70F3234: 6K , µPD70F3235: 12K , µPD70F3236: 16K are provided
V850ES/FJ2: µPD70F3237: 12KB, µPD70F3238: 20KB, µPD703239: 20KB are provided.

2. Use of addresses 3FEF000H to 3FEFFFFH is prohibited because these addresses are in the same area as the on-chip peripheral I/O area.

3. A fetch access and a read access to addresses 0000000H to 00FFFFFH is made to the internal ROM area, but these addresses are used as an external memory area when a data write access is made.

Figure 3-4. Program Memory Map



Remark Instructions can be executed in the external memory area without a branch from the internal ROM area to the external memory area. For details, refer to 3.4.5 (2) Internal RAM area

3.4.5 Areas

(1) Internal flash memory area

Up to 1 MB is reserved as an internal flash memory area.

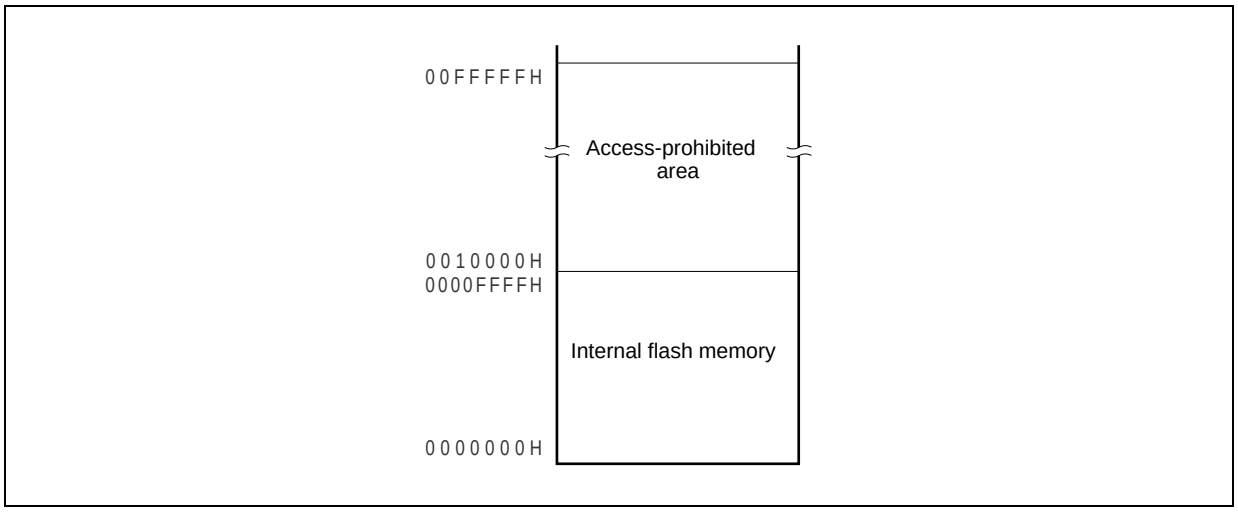
(a) Internal Mask ROM memory area (64 KB)

64 KB, addresses 0000000H to 000FFFFH, are provided in the following products as an internal flash memory area.

Accessing addresses 0001000H to 00FFFFFH is prohibited.

- V850ES/FE2: μPD703230

Figure 3-5. Internal Flash Memory Area (64 KB)



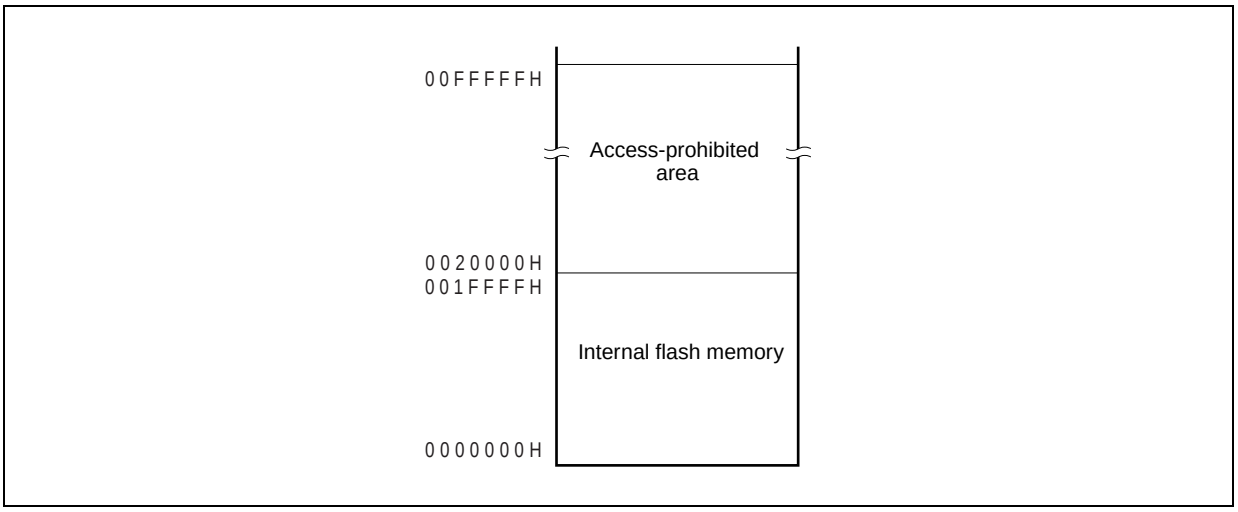
(b) Internal flash memory area (128 KB)

128 KB, addresses 0000000H to 001FFFFH, are provided in the following products as an internal flash memory area.

Accessing addresses 0020000H to 00FFFFFH is prohibited.

- V850ES/FE2: μPD70F3231
- V850ES/FF2: μPD70F3232
- V850ES/FG2: μPD70F3234

Figure 3-6. Internal Flash Memory Area (128 KB)



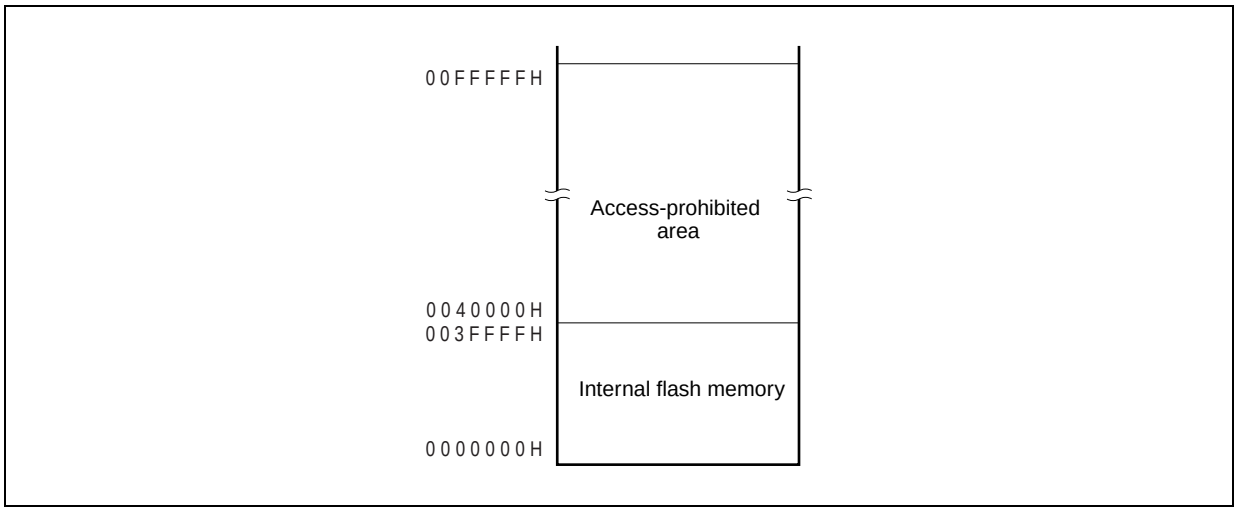
(c) Internal flash memory area (256 KB)

256 KB, addresses 0000000H to 003FFFFH, are provided in the following products as an internal flash memory area.

Accessing addresses 0040000H to 00FFFFFH is prohibited.

- V850ES/FF2: μPD70F3233
- V850ES/FG2: μPD70F3235
- V850ES/FJ2: μPD70F3237

Figure 3-7. Internal Flash Memory Area (256 KB)



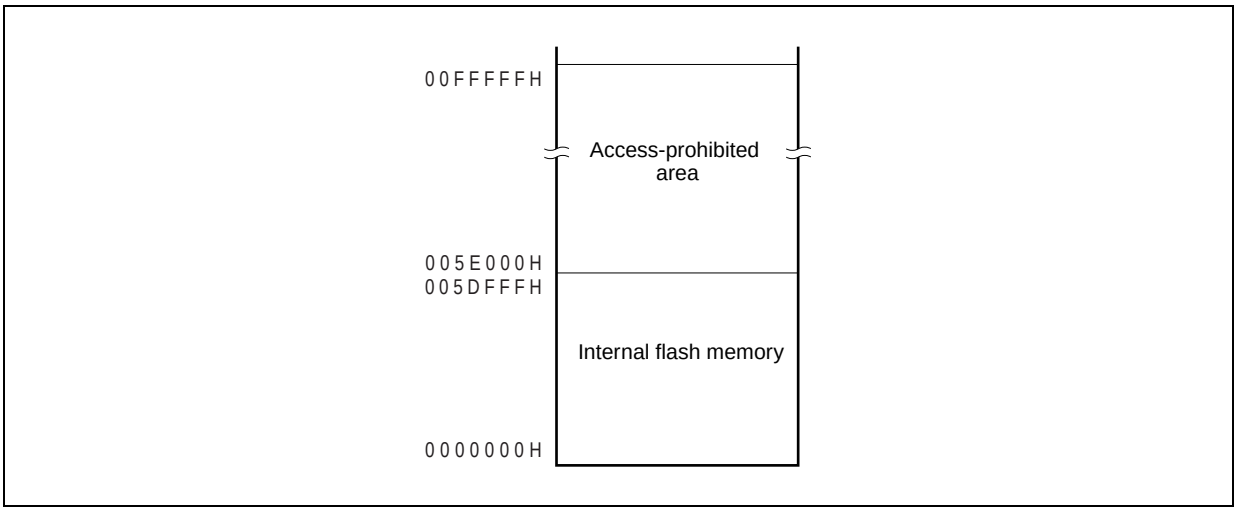
(d) Internal flash memory area (376 KB)

The following products have a 376 KB area of 00000000H to 0005DFFFH.

Use of addresses 005E000H to 00FFFFFH is prohibited.

- V850ES/FJ2: μPD70F3238

Figure 3-8. Internal Flash Memory Area (376 KB)



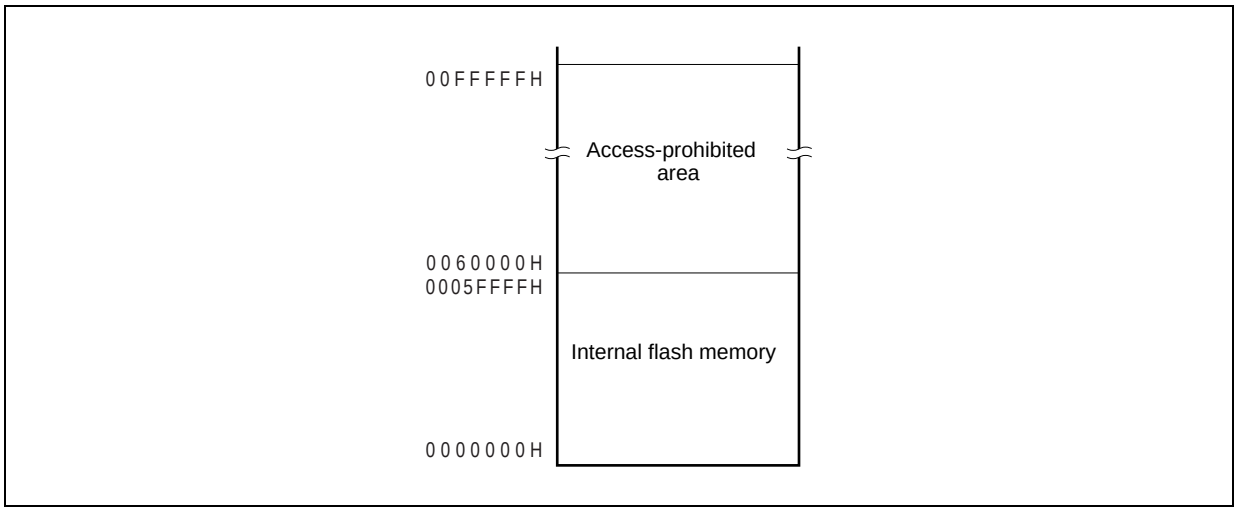
(e) Internal flash memory area (384 KB)

The following products have a 384 KB area of 00000000H to 0005FFFFH.

Use of addresses 0060000H to 00FFFFFH is prohibited.

- V850ES/FG2: μPD70F3236

Figure 3-9. Internal Flash Memory Area (384 KB)



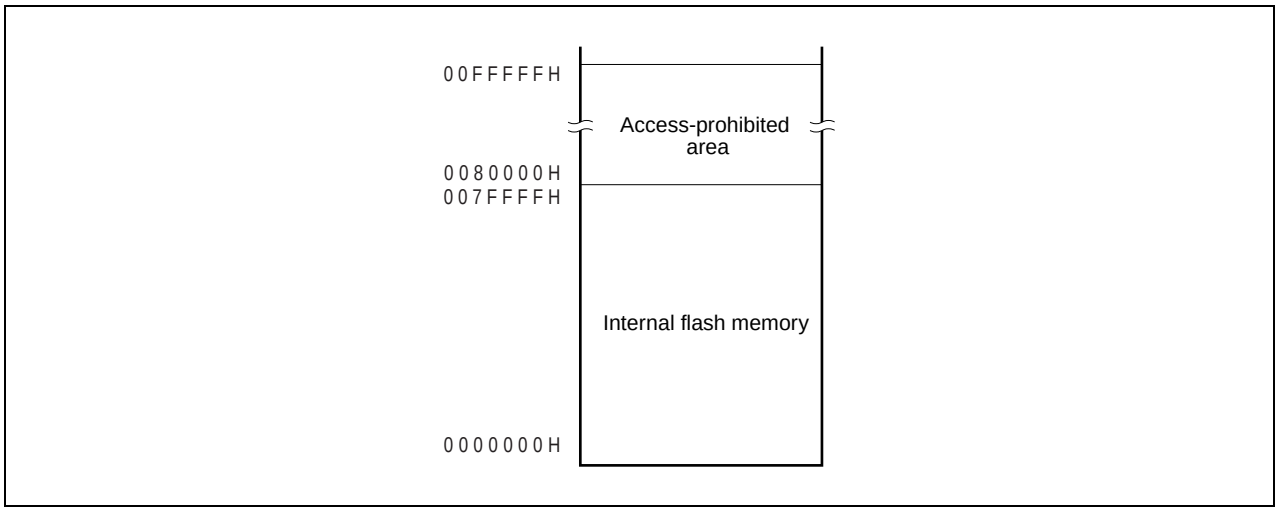
(f) Internal flash memory area (512 KB)

512 KB, addresses 0000000H to 007FFFFH, are provided in the following product as an internal flash memory area.

Accessing addresses 0080000H to 00FFFFFH is prohibited.

- V850ES/FJ2: μPD70F3239

Figure 3-10. Internal Flash Memory Area (512 KB)



(2) Internal RAM area

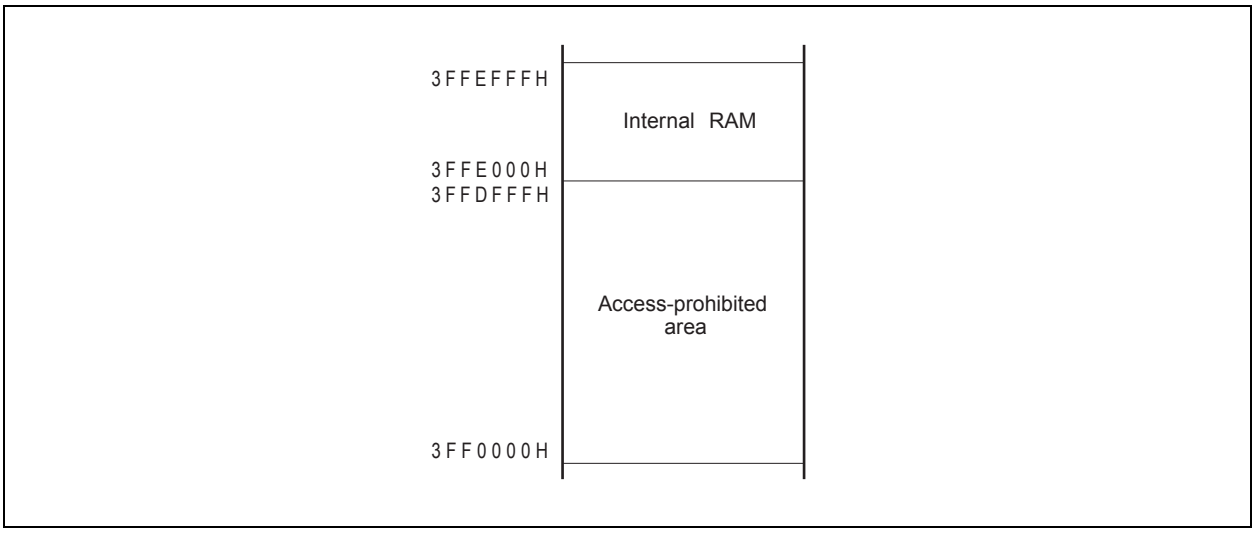
Up to 60 KB are reserved as an internal RAM area.

(a) Internal RAM (4 KB)

The following product has a 4 KB area from addresses 3FFE000H to 3FFEFFFH. Use of addresses 3FF0000H to 3FFDFFFH is prohibited.

- V850ES/FE2: μPD703230

Figure 3-11. Internal RAM Area (4 KB)

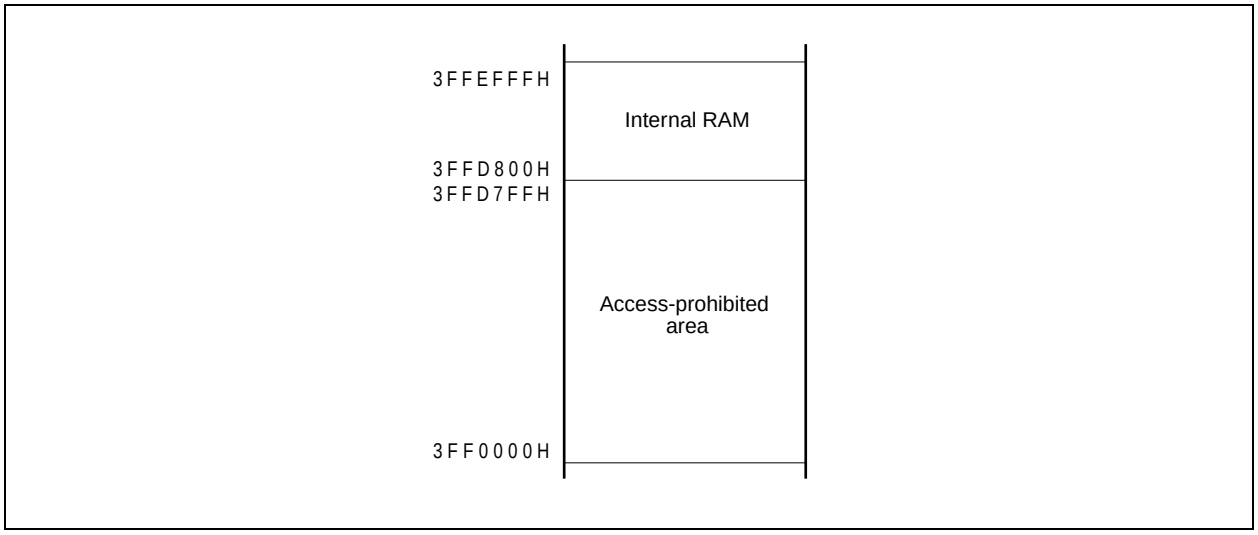


(b) Internal RAM (6 KB)

The following products have a 6 KB area from addresses 3FFD800H to 3FFEFFFH. Use of addresses 3FF0000H to 3FFD7FFH is prohibited.

- V850ES/FE2: μPD70F3231
- V850ES/FF2: μPD703232
- V850ES/FG2: μPD70F3234

Figure 3-12. Internal RAM Area (6 KB)

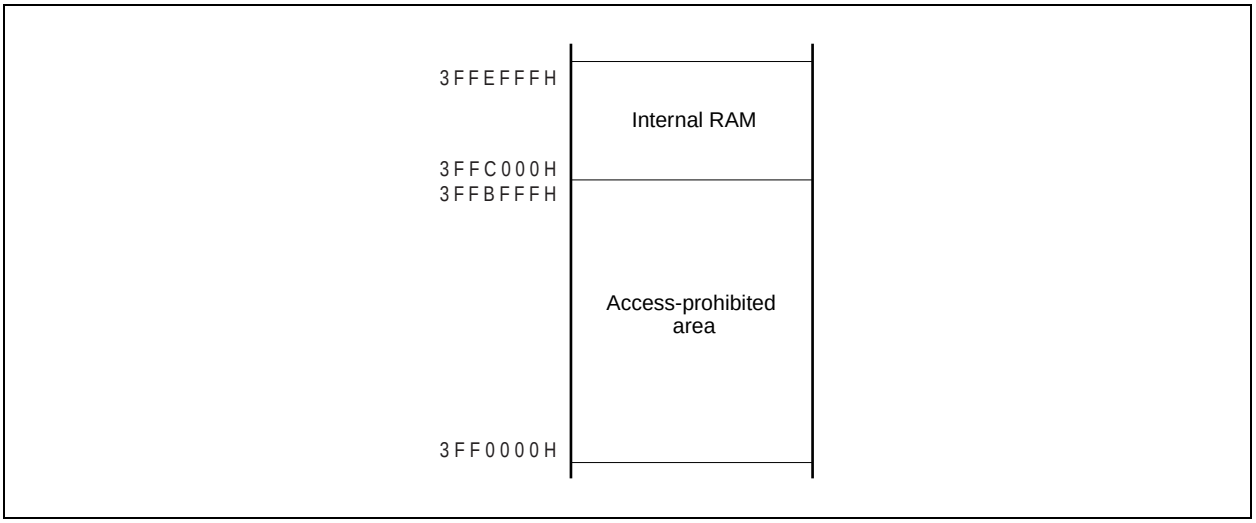


(c) Internal RAM (12 KB)

The following products have a 12 KB area from addresses 3FFC000H to 3FFEFFFH. Use of addresses 3FF0000H to 3FFBFFFH is prohibited.

- V850ES/FF2: μPD70F3232, µPD70F3233
- V850ES/FG2: μPD70F3235
- V850ES/FJ2: μPD70F3237

Figure 3-13. Internal RAM Area (12 KB)

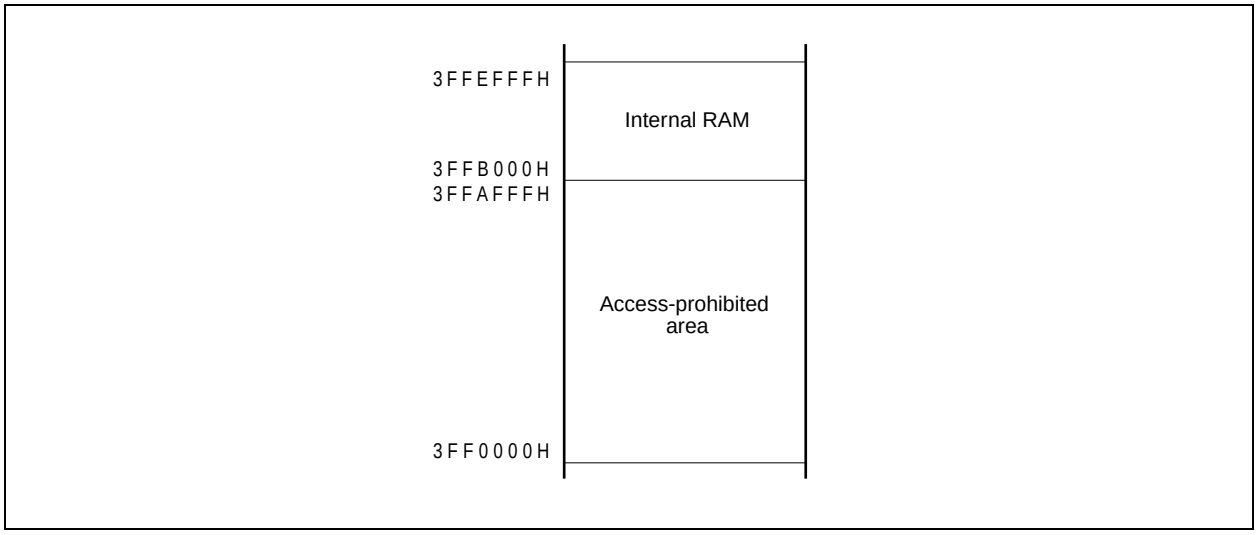


(d) Internal RAM (16 KB)

The following product has a 16 KB area from addresses 3FFb000H to 3FFEFFFH. Use of addresses 3FF0000H to 3FFAFFFH is prohibited.

- V850ES/FG2: μPD70F3236

Figure 3-14. Internal RAM Area (16 KB)

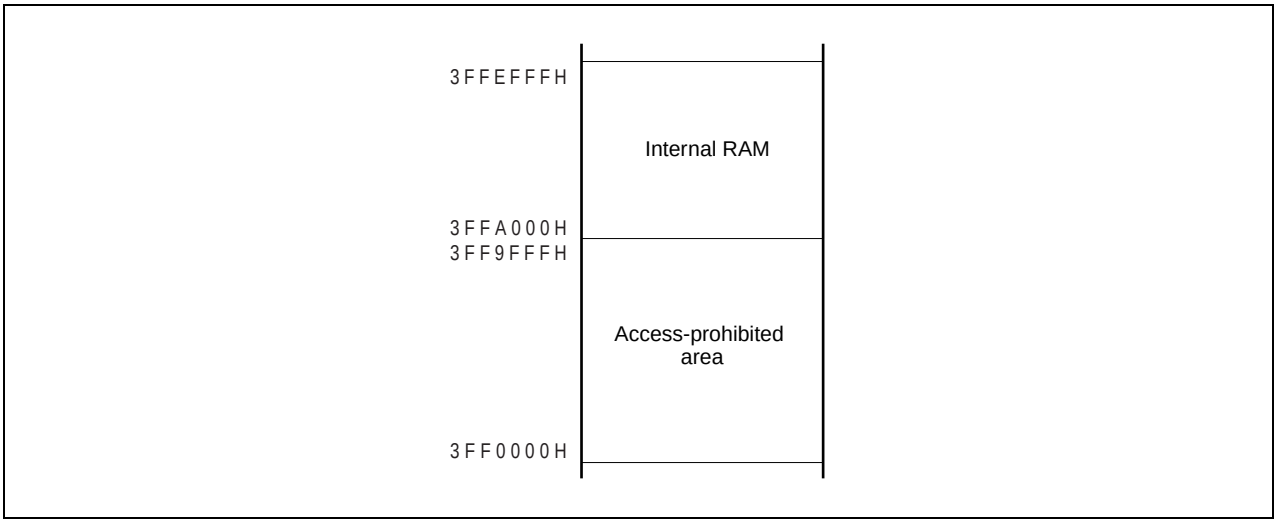


(e) Internal RAM (20 KB)

The following products have a 20 KB area from addresses 3FFA000H to 3FFEFFFH. Use of addresses 3FF0000H to 3FF9FFFH is prohibited.

- V850ES/FJ2: μPD70F3238, μPD70F3239

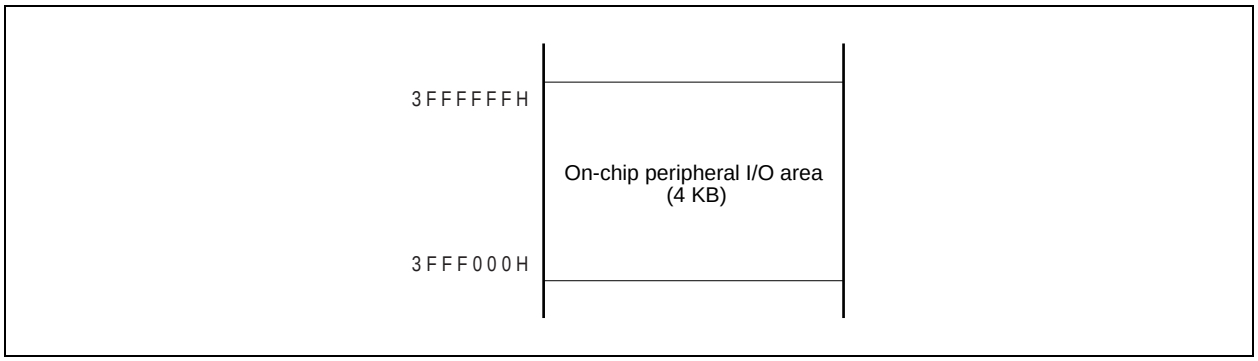
Figure 3-15. Internal RAM Area (20 KB)



(3) On-chip peripheral I/O area

4 KB, addresses 3FFF000H to 3FFFFFFH, are reserved as an on-chip peripheral I/O area.

Figure 3-16. On-Chip Peripheral I/O Area



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

Cautions 1. If a register is accessed in word units, a word area with the lower 2 bits of an address ignored is accessed in halfword units in the order of the lower halfword and the higher halfword.

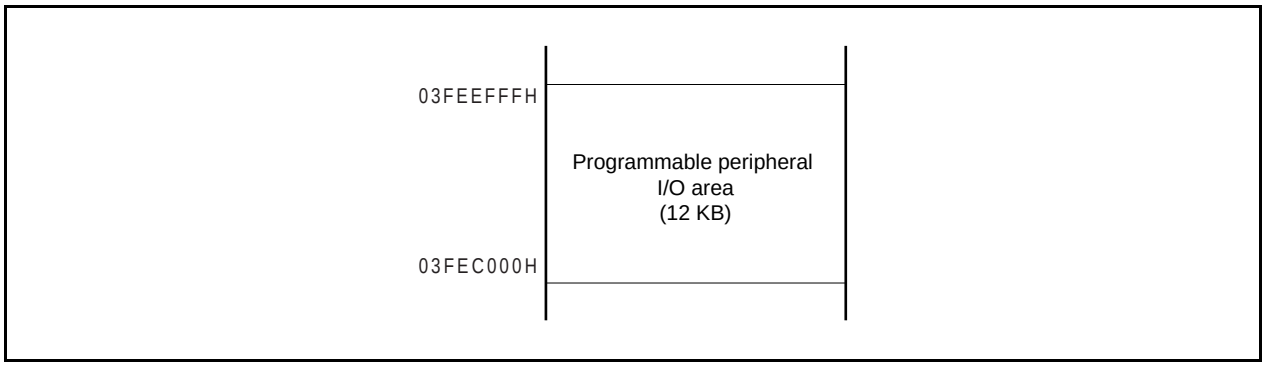
2. If a register that can be accessed in bytes is accessed in halfwords, the higher 8 bits are undefined when the register is read. Data is written to the lower 8 bits when a write access is made to the register.

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

(4) Programmable peripheral I/O area

12 KB of addresses 03FEC000H to 03FEEFFFH are reserved as the programmable peripheral I/O area.

Figure 3-17. Programmable Peripheral I/O Area



Caution The programmable peripheral I/O area is seen as images of 256 MB each in the 4 GB address space.

(5) External memory area

An external memory area of 15 MB (0100000H to 0FFFFFFH) is available. For details, refer to **CHAPTER 5 BUS CONTROL FUNCTION**.

3.4.6 Recommended use of address space

With the architecture of the V850ES/Fx2, a register that serves as a pointer must be secured for address generation when operand data in the data space is to be accessed. The ±32 KB of this pointer register can be directly accessed from an instruction for operand data. However, the number of general-purpose registers that can be used as a pointer is limited. The number of general-purpose registers that can be allocated for variables can be maximized and the program size can be reduced by suppressing a drop in performance due to address calculation when a pointer value is to be changed.

(1) Program space

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

When using the internal RAM area as a program space, access the following addresses.

Caution The prefetch operation (invalid fetch) across internal peripheral I/O area is not generated if there is a branch instruction in the upper-limit addresses of the internal RAM area.

RAM Size	Addresses to Be Accessed
20 KB	3FFA000H to 3FFEFFFH
16 KB	3FFB000H to 3FFEFFFH
12 KB	3FFC000H to3FFEFFFH
6 KB	3FFD800H to3FFEFFFH
4 KB	3FFE000H to3FFEFFFH

(2) Data space

On the 4 GB CPU address space of the V850ES/Fx2, there seems to be sixty-four 64 MB physical address spaces. Therefore, 26-bit addresses with the most significant bit (bit 25) sign-extended up to 32 bits in length are allocated.

(a) Application example of wraparound

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

The zero register (r0) is fixed to 0 by hardware and a register used for a pointer is basically unnecessary.

Example: μPD70F3239

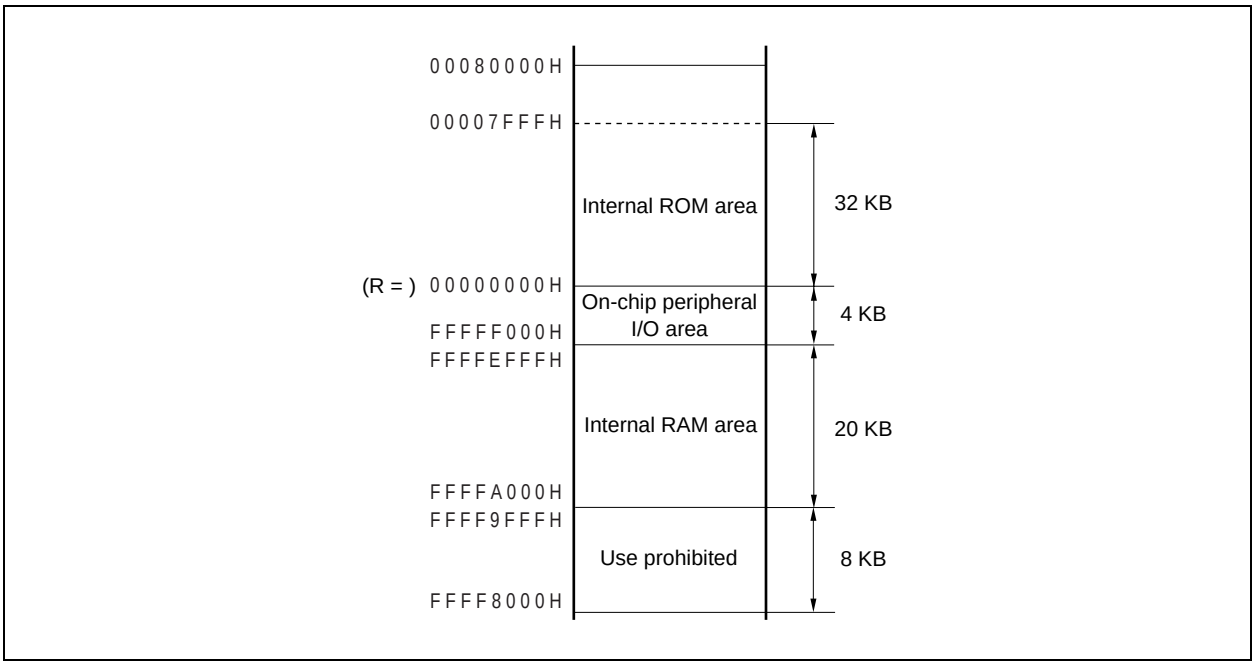
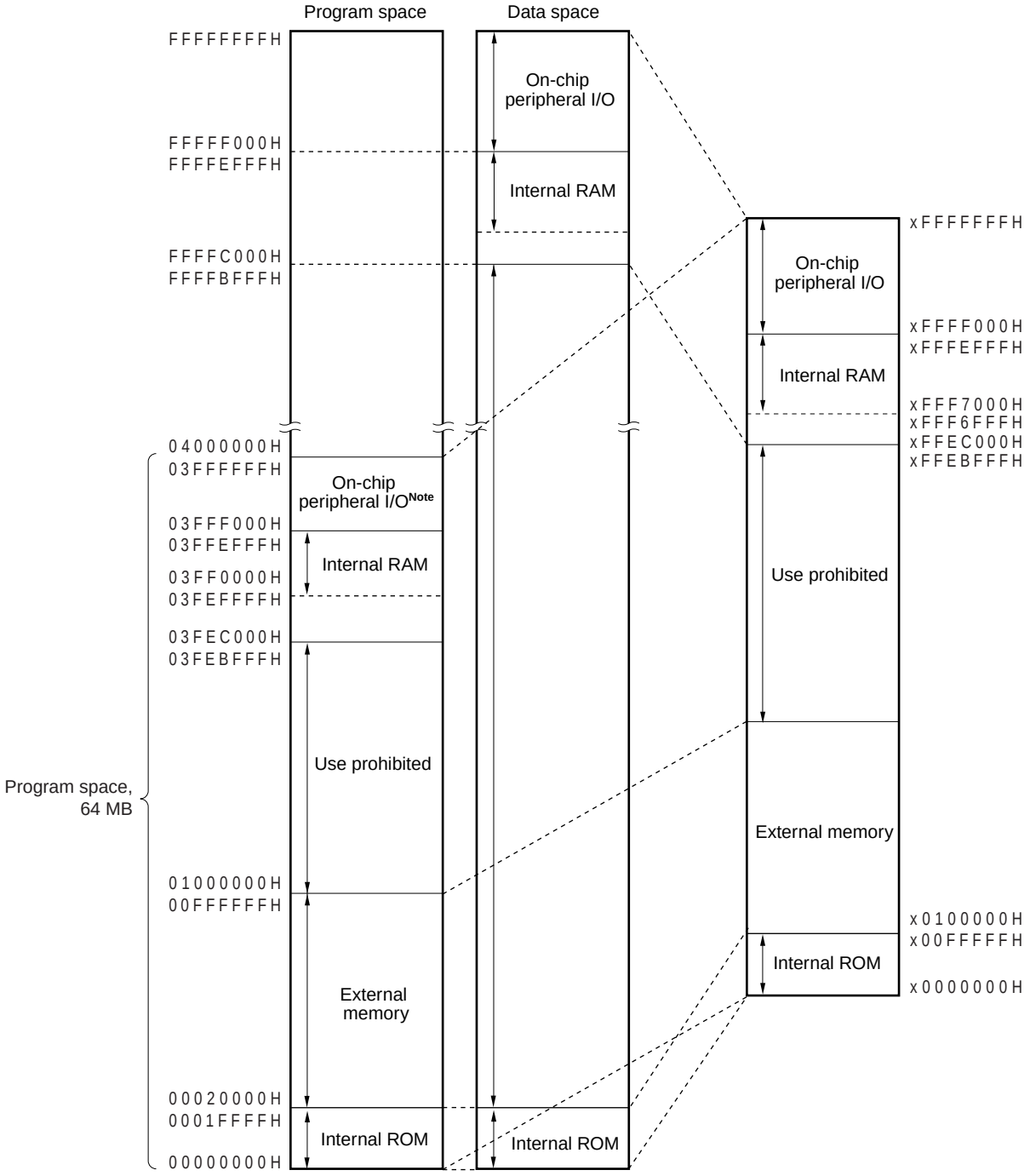


Figure 3-18. Recommended Memory Map



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

Remarks 1. The upper and lower arrows indicate the area recommended to be used.

2. This figure is the recommended memory map of the uPD70F3238 and uPD79F0239.

3.4.7 Peripheral I/O registers

3.4.7.1 V850ES/FE2

(1/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL	PDL	R/W			√	Undefined
FFFFF004H	Port DLL	PDLL		√	√		Undefined
FFFFF005H	Port DLH	PDLH		√	√		Undefined
FFFFF00CH	Port CM	PCM		√	√		Undefined
FFFFF024H	Port mode register DL	PMDL				√	FFFFH
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH
FFFFF025H	Port mode register DLH	PMDLH		√	√		FFH
FFFFF02CH	Port mode register CM	PMCM		√	√		FFH
FFFFF04CH	Port mode control register CM	PMCCM		√	√		00H
FFFFF064H	Peripheral I/O area select control register	BPC				√	0000H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF108H	Interrupt mask register 4	IMR4				√	FFFFH
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH

(2/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF10AH	Interrupt mask register 5L	IMR5L	R/W	√	√		FFH
FFFFF110H	Interrupt control register	LVIIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		47H
FFFFF114H	Interrupt control register	PIC1		√	√		47H
FFFFF116H	Interrupt control register	PIC2		√	√		47H
FFFFF118H	Interrupt control register	PIC3		√	√		47H
FFFFF11AH	Interrupt control register	PIC4		√	√		47H
FFFFF11CH	Interrupt control register	PIC5		√	√		47H
FFFFF11EH	Interrupt control register	PIC6		√	√		47H
FFFFF120H	Interrupt control register	PIC7		√	√		47H
FFFFF122H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC1		√	√		47H
FFFFF128H	Interrupt control register	TQ0CCIC2		√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12CH	Interrupt control register	TP0OVIC		√	√		47H
FFFFF12EH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF130H	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF132H	Interrupt control register	TP1OVIC		√	√		47H
FFFFF134H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF136H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF138H	Interrupt control register	TP2OVIC		√	√		47H
FFFFF13AH	Interrupt control register	TP2CCIC0		√	√		47H
FFFFF13CH	Interrupt control register	TP2CCIC1		√	√		47H

(3/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF13EH	Interrupt control register	TP3OVIC	R/W	√	√		47H
FFFFF140H	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF142H	Interrupt control register	TP3CCIC1		√	√		47H
FFFFF144H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF146H	Interrupt control register	CB0RIC		√	√		47H
FFFFF148H	Interrupt control register	CB0TIC		√	√		47H
FFFFF14AH	Interrupt control register	CB1RIC		√	√		47H
FFFFF14CH	Interrupt control register	CB1TIC		√	√		47H
FFFFF14EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF150H	Interrupt control register	UA0TIC		√	√		47H
FFFFF152H	Interrupt control register	UA1RIC		√	√		47H
FFFFF154H	Interrupt control register	UA1TIC		√	√		47H
FFFFF156H	Interrupt control register	ADIC		√	√		47H
FFFFF158H	Interrupt control register	C0ERRIC		√	√		47H
FFFFF15AH	Interrupt control register	C0WUPIC		√	√		47H
FFFFF15CH	Interrupt control register	C0RECIC		√	√		47H
FFFFF15EH	Interrupt control register	C0TRXIC		√	√		47H
FFFFF160H	Interrupt control register	KRIC		√	√		47H
FFFFF162H	Interrupt control register	WTIIC		√	√		47H
FFFFF164H	Interrupt control register	WTIC		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter channel specification register 0	ADA0S		√	√		00H
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H
FFFFF204H	Power-fail comparison mode register	ADA0PFM		√	√		00H
FFFFF205H	Power-fail comparison threshold value register	ADA0PFT		√	√		00H

(4/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF210H	A/D conversion result register 0	ADA0CR0	R			√	Undefined
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		Undefined
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	Undefined
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		Undefined
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	Undefined
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		Undefined
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	Undefined
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		Undefined
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	Undefined
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		Undefined
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	Undefined
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		Undefined
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	Undefined
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		Undefined
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	Undefined
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		Undefined
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	Undefined
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		Undefined
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	Undefined
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		Undefined
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H
FFFFF318H	Noise elimination control register	NFC		√	√		00H
FFFFF400H	Port 0	P0		√	√		Undefined
FFFFF406H	Port 3	P3				√	Undefined
FFFFF406H	Port 3L	P3L		√	√		Undefined
FFFFF408H	Port 4	P4		√	√		Undefined
FFFFF40AH	Port 5	P5		√	√		Undefined
FFFFF40EH	Port 7L	P7L		√	√		Undefined
FFFFF40FH	Port 7H	P7H		√	√		Undefined
FFFFF412H	Port 9	P9				√	Undefined
FFFFF412H	Port 9L	P9L		√	√		Undefined
FFFFF413H	Port 9H	P9H		√	√		Undefined

(5/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF420H	Port mode register 0	PM0	R/W	√	√		FFH
FFFFF426H	Port mode register 3	PM3				√	FFFFH
FFFFF426H	Port mode register 3L	PM3L		√	√		FFH
FFFFF428H	Port mode register 4	PM4		√	√		FFH
FFFFF42AH	Port mode register 5	PM5		√	√		FFH
FFFFF42EH	Port mode register 7	PM7				√	FFFFH
FFFFF42EH	Port mode register 7L	PM7L		√	√		FFH
FFFFF42FH	Port mode register 7H	PM7H		√	√		FFH
FFFFF432H	Port mode register 9	PM9				√	FFFFH
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH
FFFFF433H	Port mode register 9H	PM9H		√	√		FFH
FFFFF440H	Port mode control register 0	PMC0		√	√		00H
FFFFF446H	Port mode control register 3	PMC3				√	0000H
FFFFF446H	Port mode control register 3L	PMC3L		√	√		00H
FFFFF447H	Port mode control register 3H	PMC3H		√	√		00H
FFFFF448H	Port mode control register 4	PMC4		√	√		00H
FFFFF44AH	Port mode control register 5	PMC5		√	√		00H
FFFFF452H	Port mode control register 9	PMC9				√	0000H
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H
FFFFF453H	Port mode control register 9H	PMC9H		√	√		00H
FFFFF460H	Port function control register 0	PFC0		√	√		00H
FFFFF466H	Port function control register 3L	PFC3L		√	√		00H
FFFFF46AH	Port function control register 5	PFC5		√	√		00H
FFFFF472H	Port function control register 9	PFC9				√	0000H
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H
FFFFF473H	Port function control register 9H	PFC9H		√	√		00H
FFFFF540H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H

(6/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2	R/W	√	√		00H
FFFFF545H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O control register 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H

(7/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF610H	TMQ1 timer control register 0	TQ1CTL0	R/W	√	√		00H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF706H	Port function control expansion register 3L	PFCE3L		√	√		00H
FFFFF70AH	Port function control expansion register 5	PFCE5		√	√		00H
FFFFF712H	Port function control expansion register 9	PFCE9				√	0000H
FFFFF712H	Port function control expansion register 9L	PFCE9L		√	√		00H
FFFFF713H	Port function control expansion register 9H	PFCE9H		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Ring OSC mode register	RCM		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		00H
FFFFF828H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF82EH	CPU operating clock status register	CCLS	R	√	√		00H
FFFFF82FH	Programmable clock mode register	PCLM	R/W	√	√		00H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFF892H	Internal RAM data status register	RAMS		√	√		01H
FFFFF8B0H	Prescaler mode register 0	PRSM0			√		00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√		00H
FFFFF9FCH	On-chip debug mode register	OCDM		√	√		01H
FFFFF9FEH	Peripheral emulation register 1	PEMU1		√	√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH

Caution For OCDM details, refer to CHAPTER 25 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT).

(8/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 receive data register	UA1TX	R/W		√		FFH
FFFFFB00H	TIP00 noise eliminator control register	P00NFC	R/W	√	√		00H
FFFFFB04H	TIP01 noise eliminator control register	P01NFC		√	√		00H
FFFFFB08H	TIP10 noise eliminator control register	P10NFC		√	√		00H
FFFFFB0CH	TIP11 noise eliminator control register	P11NFC		√	√		00H
FFFFFB10H	TIP20 noise eliminator control register	P20NFC		√	√		00H
FFFFFB14H	TIP21 noise eliminator control register	P21NFC		√	√		00H
FFFFFB18H	TIP30 noise eliminator control register	P30NFC		√	√		00H
FFFFFB1CH	TIP31 noise eliminator control register	P31NFC		√	√		00H
FFFFFB50H	TIQ00 noise eliminator control register	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 noise eliminator control register	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 noise eliminator control register	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 noise eliminator control register	Q03NFC		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3				√	0000H
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H
FFFFFC07H	External interrupt falling edge specification register 3H	INTF3H		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3				√	0000H
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H
FFFFFC27H	External interrupt rising edge specification register 3H	INTR3H		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H

(9/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFC46H	Pull-up resistor option register 3L	PU3L	R/W	√	√		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H

3.4.1.2 V850ES/FF2

(1/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL	PDL	R/W			√	Undefined
FFFFF004H	Port DLL	PDLL		√	√		Undefined
FFFFF005H	Port DLH	PDLH		√	√		Undefined
FFFFF008H	Port CS	PCS		√	√		Undefined
FFFFF00AH	Port CT	PCT		√	√		Undefined
FFFFF00CH	Port CM	PCM		√	√		Undefined
FFFFF024H	Port mode register DL	PMDL				√	FFFFH
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH
FFFFF025H	Port mode register DLH	PMDLH		√	√		FFH
FFFFF02CH	Port mode register CM	PMCM		√	√		FFH
FFFFF04CH	Port mode control register CM	PMCCM		√	√		00H
FFFFF064H	Peripheral I/O area select control register	BPC				√	0000H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF108H	Interrupt mask register 4	IMR4				√	FFFFH
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH

(2/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF10AH	Interrupt mask register 5L	IMR5L	R/W	√	√		FFH
FFFFF110H	Interrupt control register	LVIIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		47H
FFFFF114H	Interrupt control register	PIC1		√	√		47H
FFFFF116H	Interrupt control register	PIC2		√	√		47H
FFFFF118H	Interrupt control register	PIC3		√	√		47H
FFFFF11AH	Interrupt control register	PIC4		√	√		47H
FFFFF11CH	Interrupt control register	PIC5		√	√		47H
FFFFF11EH	Interrupt control register	PIC6		√	√		47H
FFFFF120H	Interrupt control register	PIC7		√	√		47H
FFFFF122H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC1		√	√		47H
FFFFF128H	Interrupt control register	TQ0CCIC2		√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12CH	Interrupt control register	TP0OVIC		√	√		47H
FFFFF12EH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF130H	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF132H	Interrupt control register	TP1OVIC		√	√		47H
FFFFF134H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF136H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF138H	Interrupt control register	TP2OVIC		√	√		47H
FFFFF13AH	Interrupt control register	TP2CCIC0		√	√		47H
FFFFF13CH	Interrupt control register	TP2CCIC1		√	√		47H

(3/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF13EH	Interrupt control register	TP3OVIC	R/W	√	√		47H
FFFFF140H	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF142H	Interrupt control register	TP3CCIC1		√	√		47H
FFFFF144H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF146H	Interrupt control register	CB0RIC		√	√		47H
FFFFF148H	Interrupt control register	CB0TIC		√	√		47H
FFFFF14AH	Interrupt control register	CB1RIC		√	√		47H
FFFFF14CH	Interrupt control register	CB1TIC		√	√		47H
FFFFF14EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF150H	Interrupt control register	UA0TIC		√	√		47H
FFFFF152H	Interrupt control register	UA1RIC		√	√		47H
FFFFF154H	Interrupt control register	UA1TIC		√	√		47H
FFFFF156H	Interrupt control register	ADIC		√	√		47H
FFFFF158H	Interrupt control register	C0ERRIC		√	√		47H
FFFFF15AH	Interrupt control register	C0WUPIC		√	√		47H
FFFFF15CH	Interrupt control register	C0RECIC		√	√		47H
FFFFF15EH	Interrupt control register	C0TRXIC		√	√		47H
FFFFF160H	Interrupt control register	KRIC		√	√		47H
FFFFF162H	Interrupt control register	WTIIC		√	√		47H
FFFFF164H	Interrupt control register	WTIC		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter channel specification register 0	ADA0S		√	√		00H
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H
FFFFF204H	Power-fail comparison mode register	ADA0PFM		√	√		00H
FFFFF205H	Power-fail comparison threshold value register	ADA0PFT		√	√		00H

(4/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF210H	A/D conversion result register 0	ADA0CR0	R			√	00H
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		00H
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	00H
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		00H
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	00H
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		00H
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	00H
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		00H
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	00H
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		00H
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	00H
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		00H
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	00H
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		00H
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	00H
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		00H
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	00H
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		00H
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	00H
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		00H
FFFFF224H	A/D conversion result register 10	ADA0CR10				√	00H
FFFFF225H	A/D conversion result register 10H	ADA0CR10H			√		00H
FFFFF226H	A/D conversion result register 11	ADA0CR11				√	00H
FFFFF227H	A/D conversion result register 11H	ADA0CR11H			√		00H
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H
FFFFF318H	Noise elimination control register	NFC		√	√		00H
FFFFF400H	Port 0	P0		√	√		Undefined
FFFFF406H	Port 3	P3				√	Undefined
FFFFF406H	Port 3L	P3L		√	√		Undefined
FFFFF407H	Port 3H	P3H		√	√		Undefined
FFFFF408H	Port 4	P4		√	√		Undefined
FFFFF40AH	Port 5	P5		√	√		Undefined
FFFFF40EH	Port 7L	P7L		√	√		Undefined
FFFFF40FH	Port 7H	P7H		√	√		Undefined
FFFFF412H	Port 9	P9				√	Undefined
FFFFF412H	Port 9L	P9L		√	√		Undefined
FFFFF413H	Port 9H	P9H		√	√		Undefined

(5/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF420H	Port mode register 0	PM0	R/W	√	√		FFH
FFFFF426H	Port mode register 3	PM3				√	FFFFH
FFFFF426H	Port mode register 3L	PM3L		√	√		FFH
FFFFF427H	Port mode register 3H	PM3H		√	√		FFH
FFFFF428H	Port mode register 4	PM4		√	√		FFH
FFFFF42AH	Port mode register 5	PM5		√	√		FFH
FFFFF42EH	Port mode register 7	PM7				√	FFFFH
FFFFF42EH	Port mode register 7L	PM7L		√	√		FFH
FFFFF42FH	Port mode register 7H	PM7H		√	√		FFH
FFFFF432H	Port mode register 9	PM9				√	FFFFH
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH
FFFFF433H	Port mode register 9H	PM9H		√	√		FFH
FFFFF440H	Port mode control register 0	PMC0		√	√		00H
FFFFF446H	Port mode control register 3	PMC3				√	0000H
FFFFF446H	Port mode control register 3L	PMC3L		√	√		00H
FFFFF447H	Port mode control register 3H	PMC3H		√	√		00H
FFFFF448H	Port mode control register 4	PMC4		√	√		00H
FFFFF44AH	Port mode control register 5	PMC5		√	√		00H
FFFFF452H	Port mode control register 9	PMC9				√	0000H
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H
FFFFF453H	Port mode control register 9H	PMC9H		√	√		00H
FFFFF460H	Port function control register 0	PFC0		√	√		00H
FFFFF466H	Port function control register 3L	PFC3L		√	√		00H
FFFFF46AH	Port function control register 5	PFC5		√	√		00H
FFFFF472H	Port function control register 9	PFC9				√	0000H
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H
FFFFF473H	Port function control register 9H	PFC9H		√	√		00H
FFFFF540H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H

(6/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2	R/W	√	√		00H
FFFFF545H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O control register 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H

(7/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF706H	Port function control expansion register 3L	PFCE3L		√	√		00H
FFFFF70AH	Port function control expansion register 5	PFCE5		√	√		00H
FFFFF712H	Port function control expansion register 9	PFCE9				√	0000H
FFFFF712H	Port function control expansion register 9L	PFCE9L		√	√		00H
FFFFF713H	Port function control expansion register 9H	PFCE9H		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Ring OSC mode register	RCM		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		00H
FFFFF828H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF82EH	CPU operating clock status register	CCLS	R	√	√		00H
FFFFF82FH	Programmable clock mode register	PCLM	R/W	√	√		00H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFF892H	Internal RAM data status register	RAMS		√	√		01H
FFFFF8B0H	Prescaler mode register 0	PRSM0			√		00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√		00H
FFFFF9FCH	On-chip debug mode register	OCDM		√	√		01H
FFFFF9FEH	Peripheral emulation register 1	PEMU1		√	√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH

Caution For OCDM details, refer to CHAPTER 25 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT).

(8/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 receive data register	UA1TX	R/W		√		FFH
FFFFFB00H	TIP00 noise eliminator control register	P00NFC	R/W	√	√		00H
FFFFFB04H	TIP01 noise eliminator control register	P01NFC		√	√		00H
FFFFFB08H	TIP10 noise eliminator control register	P10NFC		√	√		00H
FFFFFB0CH	TIP11 noise eliminator control register	P11NFC		√	√		00H
FFFFFB10H	TIP20 noise eliminator control register	P20NFC		√	√		00H
FFFFFB14H	TIP21 noise eliminator control register	P21NFC		√	√		00H
FFFFFB18H	TIP30 noise eliminator control register	P30NFC		√	√		00H
FFFFFB1CH	TIP31 noise eliminator control register	P31NFC		√	√		00H
FFFFFB50H	TIQ00 noise eliminator control register	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 noise eliminator control register	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 noise eliminator control register	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 noise eliminator control register	Q03NFC		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3				√	0000H
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H
FFFFFC07H	External interrupt falling edge specification register 3H	INTF3H		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3				√	0000H
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H
FFFFFC27H	External interrupt rising edge specification register 3H	INTR3H		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H

(9/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFC46H	Pull-up resistor option register 3L	PU3L	R/W	√	√		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H

3.4.1.3 V850ES/FG2

(1/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL	PDL	R/W			√	Undefined
FFFFF004H	Port DLL	PDLL		√	√		Undefined
FFFFF005H	Port DLH	PDLH		√	√		Undefined
FFFFF008H	Port CS	PCS		√	√		Undefined
FFFFF00AH	Port CT	PCT		√	√		Undefined
FFFFF00CH	Port CM	PCM		√	√		Undefined
FFFFF024H	Port mode register DL	PMDL				√	FFFFH
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH
FFFFF025H	Port mode register DLH	PMDLH		√	√		FFH
FFFFF028H	Port mode register CS	PMCS		√	√		FFH
FFFFF02AH	Port mode register CT	PMCT		√	√		FFH
FFFFF02CH	Port mode register CM	PMCM		√	√		FFH
FFFFF04CH	Port mode control register CM	PMCCM		√	√		00H
FFFFF064H	Peripheral I/O area select control register	BPC				√	0000H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2				√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H

(2/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF0E0H	DMA channel control register 0	DCHC0	R/W	√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		00H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF110H	Interrupt control register	LVIIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		47H
FFFFF114H	Interrupt control register	PIC1		√	√		47H
FFFFF116H	Interrupt control register	PIC2		√	√		47H
FFFFF118H	Interrupt control register	PIC3		√	√		47H
FFFFF11AH	Interrupt control register	PIC4		√	√		47H
FFFFF11CH	Interrupt control register	PIC5		√	√		47H
FFFFF11EH	Interrupt control register	PIC6		√	√		47H
FFFFF120H	Interrupt control register	PIC7		√	√		47H
FFFFF122H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC1		√	√		47H
FFFFF128H	Interrupt control register	TQ0CCIC2		√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12CH	Interrupt control register	TP0OVIC		√	√		47H
FFFFF12EH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF130H	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF132H	Interrupt control register	TP1OVIC		√	√		47H
FFFFF134H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF136H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF138H	Interrupt control register	TP2OVIC		√	√		47H
FFFFF13AH	Interrupt control register	TP2CCIC0		√	√		47H
FFFFF13CH	Interrupt control register	TP2CCIC1		√	√		47H
FFFFF13EH	Interrupt control register	TP3OVIC		√	√		47H
FFFFF140H	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF142H	Interrupt control register	TP3CCIC1		√	√		47H

(3/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF144H	Interrupt control register	TM0EQIC0	R/W	√	√		47H
FFFFF146H	Interrupt control register	CB0RIC		√	√		47H
FFFFF148H	Interrupt control register	CB0TIC		√	√		47H
FFFFF14AH	Interrupt control register	CB1RIC		√	√		47H
FFFFF14CH	Interrupt control register	CB1TIC		√	√		47H
FFFFF14EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF150H	Interrupt control register	UA0TIC		√	√		47H
FFFFF152H	Interrupt control register	UA1RIC		√	√		47H
FFFFF154H	Interrupt control register	UA1TIC		√	√		47H
FFFFF156H	Interrupt control register	ADIC		√	√		47H
FFFFF158H	Interrupt control register	C0ERRIC		√	√		47H
FFFFF15AH	Interrupt control register	C0WUPIC		√	√		47H
FFFFF15CH	Interrupt control register	C0RECIC		√	√		47H
FFFFF15EH	Interrupt control register	C0TRXIC		√	√		47H
FFFFF160H	Interrupt control register	KRIC		√	√		47H
FFFFF162H	Interrupt control register	WTIIC		√	√		47H
FFFFF164H	Interrupt control register	WTIC		√	√		47H
FFFFF166H	Interrupt control register	PIC8		√	√		47H
FFFFF168H	Interrupt control register	PIC9		√	√		47H
FFFFF16AH	Interrupt control register	PIC10		√	√		47H
FFFFF16CH	Interrupt control register	TQ1OVIC		√	√		47H
FFFFF16EH	Interrupt control register	TQ1CCIC0		√	√		47H
FFFFF170H	Interrupt control register	TQ1CCIC1		√	√		47H
FFFFF172H	Interrupt control register	TQ1CCIC2		√	√		47H
FFFFF174H	Interrupt control register	TQ1CCIC3		√	√		47H
FFFFF176H	Interrupt control register	UA2RIC		√	√		47H
FFFFF178H	Interrupt control register	UA2TIC		√	√		47H
FFFFF17AH	Interrupt control register	C1ERRIC		√	√		47H
FFFFF17CH	Interrupt control register	C1WUPIC		√	√		47H
FFFFF17EH	Interrupt control register	C1RECIC		√	√		47H
FFFFF180H	Interrupt control register	C1TRXIC		√	√		47H
FFFFF182H	Interrupt control register	DMAIC0		√	√		47H
FFFFF184H	Interrupt control register	DMAIC1		√	√		47H
FFFFF186H	Interrupt control register	DMAIC2		√	√		47H
FFFFF188H	Interrupt control register	DMAIC3		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H

(4/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF200H	A/D converter mode register 0	ADA0M0	R/W	√	√		00H
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter channel specification register 0	ADA0S		√	√		00H
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H
FFFFF204H	Power-fail comparison mode register	ADA0PFM		√	√		00H
FFFFF205H	Power-fail comparison threshold value register	ADA0PFT		√	√		00H
FFFFF210H	A/D conversion result register 0	ADA0CR0	R			√	00H
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		00H
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	00H
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		00H
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	00H
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		00H
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	00H
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		00H
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	00H
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		00H
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	00H
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		00H
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	00H
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		00H
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	00H
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		00H
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	00H
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		00H
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	00H
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		00H
FFFFF224H	A/D conversion result register 10	ADA0CR10				√	00H
FFFFF225H	A/D conversion result register 10H	ADA0CR10H			√		00H
FFFFF226H	A/D conversion result register 11	ADA0CR11				√	00H
FFFFF227H	A/D conversion result register 11H	ADA0CR11H			√		00H
FFFFF228H	A/D conversion result register 12	ADA0CR12				√	00H
FFFFF229H	A/D conversion result register 12H	ADA0CR12H			√		00H
FFFFF22AH	A/D conversion result register 13	ADA0CR13				√	00H
FFFFF22BH	A/D conversion result register 13H	ADA0CR13H			√		00H
FFFFF22CH	A/D conversion result register 14	ADA0CR14				√	00H
FFFFF22DH	A/D conversion result register 14H	ADA0CR14H			√		00H
FFFFF22EH	A/D conversion result register 15	ADA0CR15				√	00H
FFFFF22FH	A/D conversion result register 15H	ADA0CR15H			√		00H
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H
FFFFF318H	Noise elimination control register	NFC		√	√		00H

(5/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF400H	Port 0	P0	R/W	√	√		Undefined
FFFFF402H	Port 1	P1		√	√		Undefined
FFFFF406H	Port 3	P3				√	Undefined
FFFFF406H	Port 3L	P3L		√	√		Undefined
FFFFF407H	Port 3H	P3H		√	√		Undefined
FFFFF408H	Port 4	P4		√	√		Undefined
FFFFF40AH	Port 5	P5		√	√		Undefined
FFFFF40EH	Port 7L	P7L		√	√		Undefined
FFFFF40FH	Port 7H	P7H		√	√		Undefined
FFFFF412H	Port 9	P9				√	Undefined
FFFFF412H	Port 9L	P9L		√	√		Undefined
FFFFF413H	Port 9H	P9H		√	√		Undefined
FFFFF420H	Port mode register 0	PM0		√	√		FFH
FFFFF422H	Port mode register 1	PM1		√	√		FFH
FFFFF426H	Port mode register 3	PM3				√	FFFFH
FFFFF426H	Port mode register 3L	PM3L		√	√		FFH
FFFFF427H	Port mode register 3H	PM3H		√	√		FFH
FFFFF428H	Port mode register 4	PM4		√	√		FFH
FFFFF42AH	Port mode register 5	PM5		√	√		FFH
FFFFF42EH	Port mode register 7L	PM7L		√	√		FFH
FFFFF42FH	Port mode register 7H	PM7H		√	√		FFH
FFFFF432H	Port mode register 9	PM9				√	FFFFH
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH
FFFFF433H	Port mode register 9H	PM9H		√	√		FFH
FFFFF440H	Port mode control register 0	PMC0		√	√		00H
FFFFF442H	Port mode control register 1	PMC1		√	√		00H
FFFFF446H	Port mode control register 3	PMC3				√	0000H
FFFFF446H	Port mode control register 3L	PMC3L		√	√		00H
FFFFF447H	Port mode control register 3H	PMC3H		√	√		00H
FFFFF448H	Port mode control register 4	PMC4		√	√		00H
FFFFF44AH	Port mode control register 5	PMC5		√	√		00H
FFFFF452H	Port mode control register 9	PMC9				√	0000H
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H
FFFFF453H	Port mode control register 9H	PMC9H		√	√		00H
FFFFF460H	Port function control register 0	PFC0		√	√		00H
FFFFF466H	Port function control register 3L	PFC3L		√	√		00H
FFFFF46AH	Port function control register 5	PFC5		√	√		00H
FFFFF472H	Port function control register 9	PFC9				√	0000H
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H
FFFFF473H	Port function control register 9H	PFC9H		√	√		00H

(6/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF540H	TMQ0 control register 0	TQ0CTL0	R/W	√	√		00H
FFFFF541H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2		√	√		00H
FFFFF545H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H

(7/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5C4H	TMP3 I/O control register 2	TP3IOC2	R/W	√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 0	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF610H	TMQ1 control register 0	TQ1CTL0	R/W	√	√		00H
FFFFF611H	TMQ1 control register 1	TQ1CTL1		√	√		00H
FFFFF612H	TMQ1 I/O control register 0	TQ1IOC0		√	√		00H
FFFFF613H	TMQ1 I/O control register 1	TQ1IOC1		√	√		00H
FFFFF614H	TMQ1 I/O control register 2	TQ1IOC2		√	√		00H
FFFFF615H	TMQ1 timer option register 0	TQ1OPT0		√	√		00H
FFFFF616H	TMQ1 capture/compare register 0	TQ1CCR0				√	0000H
FFFFF618H	TMQ1 capture/compare register 1	TQ1CCR1				√	0000H
FFFFF61AH	TMQ1 capture/compare register 2	TQ1CCR2				√	0000H
FFFFF61CH	TMQ1 capture/compare register 3	TQ1CCR3				√	0000H
FFFFF61EH	TMQ1 counter read buffer register	TQ1CNT	R			√	0000H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF706H	Port function control expansion register 3L	PFCE3L		√	√		00H
FFFFF70AH	Port function control expansion register 5	PFCE5		√	√		00H
FFFFF712H	Port function control expansion register 9	PFCE9				√	0000H
FFFFF712H	Port function control expansion register 9L	PFCE9L		√	√		00H
FFFFF713H	Port function control expansion register 9H	PFCE9H		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Ring OSC mode register	RCM		√	√		00H
FFFFF810H	DMA trigger source register 0	DTFR0		√	√		00H
FFFFF812H	DMA trigger source register 1	DTFR1		√	√		00H
FFFFF814H	DMA trigger source register 2	DTFR2		√	√		00H
FFFFF816H	DMA trigger source register 3	DTFR3		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		00H
FFFFF828H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF82EH	CPU operating clock status register	CCLS	R	√	√		00H
FFFFF82FH	Programmable clock mode register	PCLM	R/W	√	√		00H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H

(8/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF890H	Low-voltage detection register	LVIM	R/W	√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFF892H	Internal RAM data status register	RAMS		√	√		01H
FFFFF8B0H	Prescaler mode register 0	PRSM0			√		00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√		00H
FFFFF9FCH	On-chip debug mode register	OCDM		√	√		01H
FFFFF9FEH	Peripheral emulation register 1	PEMU1		√	√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 receive data register	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 status register	UA2STR		√	√		00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√		FFH
FFFFFB00H	TIP00 noise eliminator control register	P00NFC		√	√		00H
FFFFFB04H	TIP01 noise eliminator control register	P01NFC		√	√		00H
FFFFFB08H	TIP10 noise eliminator control register	P10NFC		√	√		00H
FFFFFB0CH	TIP11 noise eliminator control register	P11NFC		√	√		00H
FFFFFB10H	TIP20 noise eliminator control register	P20NFC		√	√		00H
FFFFFB14H	TIP21 noise eliminator control register	P21NFC		√	√		00H
FFFFFB18H	TIP30 noise eliminator control register	P30NFC		√	√		00H
FFFFFB1CH	TIP31 noise eliminator control register	P31NFC		√	√		00H
FFFFFB50H	TIQ00 noise eliminator control register	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 noise eliminator control register	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 noise eliminator control register	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 noise eliminator control register	Q03NFC		√	√		00H

Caution For OCDM details, refer to CHAPTER 26 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT).

(9/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFB60H	TIQ10 noise eliminator control register	Q10NFC	R/W	√	√		00H
FFFFFB64H	TIQ11 noise eliminator control register	Q11NFC		√	√		00H
FFFFFB68H	TIQ12 noise eliminator control register	Q12NFC		√	√		00H
FFFFFB6CH	TIQ13 noise eliminator control register	Q13NFC		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3				√	0000H
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H
FFFFFC07H	External interrupt falling edge specification register 3H	INTF3H		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3				√	0000H
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H
FFFFFC27H	External interrupt rising edge specification register 3H	INTR3H		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H
FFFFFC46H	Pull-up resistor option register 3L	PU3L		√	√		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H

3.4.1.4 V850ES/FJ2

Note The on-chip peripheral I/O differs for µPD70F3237 and µPD70F3238/µPD70F3239.

(1/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL	PDL	R/W			√	Undefined
FFFFF004H	Port DLL	PDLL		√	√		Undefined
FFFFF005H	Port DLH	PDLH		√	√		Undefined
FFFFF008H	Port CS	PCS		√	√		Undefined
FFFFF00AH	Port CT	PCT		√	√		Undefined
FFFFF00CH	Port CM	PCM		√	√		Undefined
FFFFF00EH	Port CD	PCD		√	√		Undefined
FFFFF024H	Port mode register DL	PMDL				√	FFFFH
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH
FFFFF025H	Port mode register DLH	PMDLH		√	√		FFH
FFFFF028H	Port mode register CS	PMCS		√	√		FFH
FFFFF02AH	Port mode register CT	PMCT		√	√		FFH
FFFFF02CH	Port mode register CM	PMCM		√	√		FFH
FFFFF02EH	Port mode register CD	PMCD		√	√		FFH
FFFFF044H	Port mode control register DL	PMCDL				√	0000H
FFFFF044H	Port mode control register DLL	PMCDLL		√	√		00H
FFFFF045H	Port mode control register DLH	PMCDLH		√	√		00H
FFFFF048H	Port mode control register CS	PMCCS		√	√		00H
FFFFF04AH	Port mode control register CT	PMCCT		√	√		00H
FFFFF04CH	Port mode control register CM	PMCCM		√	√		00H
FFFFF064H	Peripheral I/O area select control register	BPC				√	0000H
FFFFF066H	Bus size configuration register bus	BSC				√	5555H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined

(2/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF0C0H	DMA transfer count register 0	DBC0	R/W			√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2				√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		00H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF108H	Interrupt mask register 4	IMR4				√	FFFFH
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH
FFFFF10AH	Interrupt mask register 5L	IMR5L		√	√		FFH
FFFFF110H	Interrupt control register	LVIIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		47H
FFFFF114H	Interrupt control register	PIC1		√	√		47H
FFFFF116H	Interrupt control register	PIC2		√	√		47H
FFFFF118H	Interrupt control register	PIC3		√	√		47H
FFFFF11AH	Interrupt control register	PIC4		√	√		47H
FFFFF11CH	Interrupt control register	PIC5		√	√		47H
FFFFF11EH	Interrupt control register	PIC6		√	√		47H
FFFFF120H	Interrupt control register	PIC7		√	√		47H
FFFFF122H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC1		√	√		47H

(3/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF128H	Interrupt control register	TQ0CCIC2	R/W	√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12CH	Interrupt control register	TP0OVIC		√	√		47H
FFFFF12EH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF130H	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF132H	Interrupt control register	TP1OVIC		√	√		47H
FFFFF134H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF136H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF138H	Interrupt control register	TP2OVIC		√	√		47H
FFFFF13AH	Interrupt control register	TP2CCIC0		√	√		47H
FFFFF13CH	Interrupt control register	TP2CCIC1		√	√		47H
FFFFF13EH	Interrupt control register	TP3OVIC		√	√		47H
FFFFF140H	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF142H	Interrupt control register	TP3CCIC1		√	√		47H
FFFFF144H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF146H	Interrupt control register	CB0RIC		√	√		47H
FFFFF148H	Interrupt control register	CB0TIC		√	√		47H
FFFFF14AH	Interrupt control register	CB1RIC		√	√		47H
FFFFF14CH	Interrupt control register	CB1TIC		√	√		47H
FFFFF14EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF150H	Interrupt control register	UA0TIC		√	√		47H
FFFFF152H	Interrupt control register	UA1RIC		√	√		47H
FFFFF154H	Interrupt control register	UA1TIC		√	√		47H
FFFFF156H	Interrupt control register	ADIC		√	√		47H
FFFFF158H	Interrupt control register	C0ERRIC		√	√		47H
FFFFF15AH	Interrupt control register	C0WUPIC		√	√		47H
FFFFF15CH	Interrupt control register	C0RECIC		√	√		47H
FFFFF15EH	Interrupt control register	C0TRXIC		√	√		47H
FFFFF160H	Interrupt control register	KRIC		√	√		47H
FFFFF162H	Interrupt control register	WTIIC		√	√		47H
FFFFF164H	Interrupt control register	WTIC		√	√		47H
FFFFF166H	Interrupt control register	PIC8		√	√		47H
FFFFF168H	Interrupt control register	PIC9		√	√		47H
FFFFF16AH	Interrupt control register	PIC10		√	√		47H
FFFFF16CH	Interrupt control register	TQ1OVIC		√	√		47H
FFFFF16EH	Interrupt control register	TQ1CCIC0		√	√		47H
FFFFF170H	Interrupt control register	TQ1CCIC1		√	√		47H
FFFFF172H	Interrupt control register	TQ1CCIC2		√	√		47H
FFFFF174H	Interrupt control register	TQ1CCIC3		√	√		47H
FFFFF176H	Interrupt control register	UA2RIC		√	√		47H

(4/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF178H	Interrupt control register	UA2TIC	R/W	√	√		47H
FFFFF17AH	Interrupt control register	C1ERRIC		√	√		47H
FFFFF17CH	Interrupt control register	C1WUPIC		√	√		47H
FFFFF17EH	Interrupt control register	C1RECIC		√	√		47H
FFFFF180H	Interrupt control register	C1TRXIC		√	√		47H
FFFFF182H	Interrupt control register	DMAIC0		√	√		47H
FFFFF184H	Interrupt control register	DMAIC1		√	√		47H
FFFFF186H	Interrupt control register	DMAIC2		√	√		47H
FFFFF188H	Interrupt control register	DMAIC3		√	√		47H
FFFFF18AH	Interrupt control register	PIC11		√	√		47H
FFFFF18CH	Interrupt control register	PIC12		√	√		47H
FFFFF18EH	Interrupt control register	PIC13		√	√		47H
FFFFF190H	Interrupt control register	PIC14		√	√		47H
FFFFF192H	Interrupt control register	TQ2OVIC		√	√		47H
FFFFF194H	Interrupt control register	TQ2CCIC0		√	√		47H
FFFFF196H	Interrupt control register	TQ2CCIC1		√	√		47H
FFFFF198H	Interrupt control register	TQ2CCIC2		√	√		47H
FFFFF19AH	Interrupt control register	TQ2CCIC3		√	√		47H
FFFFF19CH	Interrupt control register	CB2RIC		√	√		47H
FFFFF19EH	Interrupt control register	CB2TIC		√	√		47H
FFFFF1A0H	Interrupt control register	UA3RIC		√	√		47H
FFFFF1A2H	Interrupt control register	UA3TIC		√	√		47H
FFFFF1A4H	Interrupt control register	C2ERRIC		√	√		47H
FFFFF1A6H	Interrupt control register	C2WUPIC		√	√		47H
FFFFF1A8H	Interrupt control register	C2RECIC		√	√		47H
FFFFF1AAH	Interrupt control register	C2TRXIC		√	√		47H
FFFFF1ACH	Interrupt control register	C3ERRIC		√	√		47H
FFFFF1AEH	Interrupt control register	C3WUPIC		√	√		47H
FFFFF1B0H	Interrupt control register	C3RECIC		√	√		47H
FFFFF1B2H	Interrupt control register	C3TRXIC		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter channel specification register 0	ADA0S		√	√		00H
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H
FFFFF204H	Power-fail comparison mode register	ADA0PFM		√	√		00H
FFFFF205H	Power-fail comparison threshold value register	ADA0PFT		√	√		00H

(5/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF210H	A/D conversion result register 0	ADA0CR0	R			√	00H
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		00H
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	00H
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		00H
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	00H
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		00H
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	00H
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		00H
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	00H
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		00H
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	00H
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		00H
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	00H
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		00H
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	00H
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		00H
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	00H
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		00H
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	00H
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		00H
FFFFF224H	A/D conversion result register 10	ADA0CR10				√	00H
FFFFF225H	A/D conversion result register 10H	ADA0CR10H			√		00H
FFFFF226H	A/D conversion result register 11	ADA0CR11				√	00H
FFFFF227H	A/D conversion result register 11H	ADA0CR11H			√		00H
FFFFF228H	A/D conversion result register 12	ADA0CR12				√	00H
FFFFF229H	A/D conversion result register 12H	ADA0CR12H			√		00H
FFFFF22AH	A/D conversion result register 13	ADA0CR13				√	00H
FFFFF22BH	A/D conversion result register 13H	ADA0CR13H			√		00H
FFFFF22CH	A/D conversion result register 14	ADA0CR14				√	00H
FFFFF22DH	A/D conversion result register 14H	ADA0CR14H			√		00H
FFFFF22EH	A/D conversion result register 15	ADA0CR15				√	00H
FFFFF22FH	A/D conversion result register 15H	ADA0CR15H			√		00H
FFFFF230H	A/D conversion result register 16	ADA0CR16				√	00H
FFFFF231H	A/D conversion result register 16H	ADA0CR16H			√		00H
FFFFF232H	A/D conversion result register 17	ADA0CR17				√	00H
FFFFF233H	A/D conversion result register 17H	ADA0CR17H			√		00H
FFFFF234H	A/D conversion result register 18	ADA0CR18				√	00H
FFFFF235H	A/D conversion result register 18H	ADA0CR18H			√		00H
FFFFF236H	A/D conversion result register 19	ADA0CR19				√	00H
FFFFF237H	A/D conversion result register 19H	ADA0CR19H			√		00H

(6/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF238H	A/D conversion result register 20	ADA0CR20	R			√	00H
FFFFF239H	A/D conversion result register 20H	ADA0CR20H			√		00H
FFFFF23AH	A/D conversion result register 21	ADA0CR21				√	00H
FFFFF23BH	A/D conversion result register 21H	ADA0CR21H			√		00H
FFFFF23CH	A/D conversion result register 22	ADA0CR22				√	00H
FFFFF23DH	A/D conversion result register 22H	ADA0CR22H			√		00H
FFFFF23EH	A/D conversion result register 23	ADA0CR23				√	00H
FFFFF23FH	A/D conversion result register 23H	ADA0CR23H			√		00H
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H
FFFFF30AH	Selector operation control register 1	SELCNT1		√	√		00H
FFFFF318H	Noise elimination control register	NFC		√	√		00H
FFFFF400H	Port 0	P0		√	√		Undefined
FFFFF402H	Port 1	P1		√	√		Undefined
FFFFF406H	Port 3	P3				√	Undefined
FFFFF406H	Port 3L	P3L		√	√		Undefined
FFFFF407H	Port 3H	P3H		√	√		Undefined
FFFFF408H	Port 4	P4		√	√		Undefined
FFFFF40AH	Port 5	P5		√	√		Undefined
FFFFF40CH	Port 6	P6				√	Undefined
FFFFF40CH	Port 6L	P6L		√	√		Undefined
FFFFF40DH	Port 6H	P6H		√	√		Undefined
FFFFF40EH	Port 7L	P7L		√	√		Undefined
FFFFF40FH	Port 7H	P7H		√	√		Undefined
FFFFF410H	Port 8	P8		√	√		Undefined
FFFFF412H	Port 9	P9				√	Undefined
FFFFF412H	Port 9L	P9L		√	√		Undefined
FFFFF413H	Port 9H	P9H		√	√		Undefined
FFFFF418H	Port 12	P12		√	√		Undefined
FFFFF420H	Port mode register 0	PM0		√	√		FFH
FFFFF422H	Port mode register 1	PM1		√	√		FFH
FFFFF426H	Port mode register 3	PM3				√	FFFFH
FFFFF426H	Port mode register 3L	PM3L		√	√		FFH
FFFFF427H	Port mode register 3H	PM3H		√	√		FFH
FFFFF428H	Port mode register 4	PM4		√	√		FFH
FFFFF42AH	Port mode register 5	PM5		√	√		FFH
FFFFF42CH	Port mode register 6	PM6				√	FFFFH

(7/11)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF42CH	Port mode register 6L	PM6L	R/W	√	√		FFH
FFFFF42DH	Port mode register 6H	PM6H		√	√		FFH
FFFFF42EH	Port mode register 7L	PM7L		√	√		FFH
FFFFF42FH	Port mode register 7H	PM7H		√	√		FFH
FFFFF430H	Port mode register 8	PM8		√	√		FFH
FFFFF432H	Port mode register 9	PM9				√	FFFFH
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH
FFFFF433H	Port mode register 9H	PM9H		√	√		FFH
FFFFF438H	Port mode register 12	PM12		√	√		FFH
FFFFF440H	Port mode control register 0	PMC0		√	√		00H
FFFFF442H	Port mode control register 1	PMC1		√	√		00H
FFFFF446H	Port mode control register 3	PMC3				√	0000H
FFFFF446H	Port mode control register 3L	PMC3L		√	√		00H
FFFFF447H	Port mode control register 3H	PMC3H		√	√		00H
FFFFF448H	Port mode control register 4	PMC4		√	√		00H
FFFFF44AH	Port mode control register 5	PMC5		√	√		00H
FFFFF44CH	Port mode control register 6	PMC6				√	0000H
FFFFF44CH	Port mode control register 6L	PMC6L		√	√		00H
FFFFF44DH	Port mode control register 6H	PMC6H		√	√		00H
FFFFF450H	Port mode control register 8	PMC8		√	√		00H
FFFFF452H	Port mode control register 9	PMC9				√	0000H
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H
FFFFF453H	Port mode control register 9H	PMC9H		√	√		00H
FFFFF460H	Port function control register 0	PFC0		√	√		00H
FFFFF466H	Port function control register 3L	PFC3L		√	√		00H
FFFFF46AH	Port function control register 5	PFC5		√	√		00H
FFFFF46CH	Port function control register 6	PFC6				√	0000H
FFFFF46CH	Port function control register 6L	PFC6L		√	√		00H
FFFFF46DH	Port function control register 6H	PFC6H		√	√		00H
FFFFF472H	Port function control register 9	PFC9				√	0000H
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H
FFFFF473H	Port function control register 9H	PFC9H		√	√		00H
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF540H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H

(8/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2	R/W	√	√		00H
FFFFF545H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O control register 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H

(9/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF610H	TMQ1 timer control register 0	TQ1CTL0	R/W	√	√		00H
FFFFF611H	TMQ1 control register 1	TQ1CTL1		√	√		00H
FFFFF612H	TMQ1 I/O control register 0	TQ1IOC0		√	√		00H
FFFFF613H	TMQ1 I/O control register 1	TQ1IOC1		√	√		00H
FFFFF614H	TMQ1 I/O control register 2	TQ1IOC2		√	√		00H
FFFFF615H	TMQ1 timer option register	TQ1OPT0		√	√		00H
FFFFF616H	TMQ1 capture/compare register 0	TQ1CCR0				√	0000H
FFFFF618H	TMQ1 capture/compare register 1	TQ1CCR1				√	0000H
FFFFF61AH	TMQ1 capture/compare register 2	TQ1CCR2				√	0000H
FFFFF61CH	TMQ1 capture/compare register 3	TQ1CCR3				√	0000H
FFFFF61EH	TMQ1 counter read buffer register	TQ1CNT	R			√	0000H
FFFFF620H	TMQ2 control register 0	TQ2CTL0	R/W	√	√		00H
FFFFF621H	TMQ2 control register 1	TQ2CTL1		√	√		00H
FFFFF622H	TMQ2 I/O control register 0	TQ2IOC0		√	√		00H
FFFFF623H	TMQ2 I/O control register 1	TQ2IOC1		√	√		00H
FFFFF624H	TMQ2 I/O control register 2	TQ2IOC2		√	√		00H
FFFFF625H	TMQ2 option register	TQ2OPT0				√	00H
FFFFF626H	TMQ2 capture/compare register 0	TQ2CCR0				√	0000H
FFFFF628H	TMQ2 capture/compare register 1	TQ2CCR1				√	0000H
FFFFF62AH	TMQ2 capture/compare register 2	TQ2CCR2				√	0000H
FFFFF62CH	TMQ2 capture/compare register 3	TQ2CCR3				√	0000H
FFFFF62EH	TMQ2 counter read buffer register	TQ2CNT	R			√	0000H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF706H	Port function control expansion register 3L	PFCE3L		√	√		00H
FFFFF70AH	Port function control expansion register 5	PFCE5		√	√		00H
FFFFF712H	Port function control expansion register 9	PFCE9				√	0000H
FFFFF712H	Port function control expansion register 9L	PFCE9L		√	√		00H
FFFFF713H	Port function control expansion register 9H	PFCE9H		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Ring OSC mode register	RCM		√	√		00H
FFFFF810H	DMA trigger source register 0	DTFR0		√	√		00H
FFFFF812H	DMA trigger source register 1	DTFR1		√	√		00H
FFFFF814H	DMA trigger source register 2	DTFR2		√	√		00H
FFFFF816H	DMA trigger source register 3	DTFR3		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		00H

(10/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF828H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF82EH	CPU operating clock status register	CCLS	R	√	√		00H
FFFFF82FH	Programmable clock mode register	PCLM	R/W	√	√		00H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFF892H	Internal RAM data status register	RAMS		√	√		01H
FFFFF8B0H	Prescaler mode register 0	PRSM0			√		00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√		00H
FFFFF9FCH	On-chip debug mode register	OCDM		√	√		01H
FFFFF9FEH	Peripheral emulation register 1	PEMU1		√	√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 receive data register	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 status register	UA2STR		√	√		00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√		FFH
FFFFFA30H	UARTA3 control register 0	UA3CTL0		√	√		10H
FFFFFA31H	UARTA3 control register 1	UA3CTL1			√		00H
FFFFFA32H	UARTA3 control register 2	UA3CTL2			√		FFH
FFFFFA33H	UARTA3 option control register 0	UA3OPT0		√	√		14H
FFFFFA34H	UARTA3 status register	UA3STR		√	√		00H
FFFFFA36H	UARTA3 receive data register	UA3RX	R		√		FFH
FFFFFA37H	UARTA3 transmit data register	UA3TX	R/W		√		FFH

Caution For OCDM details, refer to CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT).

(11/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFB00H	TIP00 noise eliminator control register	P00NFC	R/W	√	√		00H
FFFFFB04H	TIP01 noise eliminator control register	P01NFC		√	√		00H
FFFFFB08H	TIP10 noise eliminator control register	P10NFC		√	√		00H
FFFFFB0CH	TIP11 noise eliminator control register	P11NFC		√	√		00H
FFFFFB10H	TIP20 noise eliminator control register	P20NFC		√	√		00H
FFFFFB14H	TIP21 noise eliminator control register	P21NFC		√	√		00H
FFFFFB18H	TIP30 noise eliminator control register	P30NFC		√	√		00H
FFFFFB1CH	TIP31 noise eliminator control register	P31NFC		√	√		00H
FFFFFB50H	TIQ00 noise eliminator control register	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 noise eliminator control register	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 noise eliminator control register	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 noise eliminator control register	Q03NFC		√	√		00H
FFFFFB60H	TIQ10 noise eliminator control register	Q10NFC		√	√		00H
FFFFFB64H	TIQ11 noise eliminator control register	Q11NFC		√	√		00H
FFFFFB68H	TIQ12 noise eliminator control register	Q12NFC		√	√		00H
FFFFFB6CH	TIQ13 noise eliminator control register	Q13NFC		√	√		00H
FFFFFB70H	TIQ20 noise eliminator control register	Q20NFC		√	√		00H
FFFFFB74H	TIQ21 noise eliminator control register	Q21NFC		√	√		00H
FFFFFB78H	TIQ22 noise eliminator control register	Q22NFC		√	√		00H
FFFFFB7CH	TIQ23 noise eliminator control register	Q23NFC		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3				√	0000H
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H
FFFFFC07H	External interrupt falling edge specification register 3H	INTF3H		√	√		00H
FFFFFC0CH	External interrupt falling edge specification register 6L	INTF6L		√	√		00H
FFFFFC10H	External interrupt falling edge specification register 8	INTF8		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3				√	0000H
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H
FFFFFC27H	External interrupt rising edge specification register 3H	INTR3H		√	√		00H
FFFFFC2CH	External interrupt rising edge specification register 6L	INTR6L		√	√		00H
FFFFFC30H	External interrupt rising edge specification register 8	INTR8		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H

(12/12)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFC46H	Pull-up resistor option register 3L	PU3L	R/W	√	√		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H
FFFFFC4CH	Pull-up resistor option register 6	PU6				√	0000H
FFFFFC4CH	Pull-up resistor option register 6L	PU6L		√	√		00H
FFFFFC4DH	Pull-up resistor option register 6H	PU6H		√	√		00H
FFFFFC50H	Pull-up resistor option register 8	PU8		√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H
FFFFFD20H	CSIB2 control register 0	CB2CTL0		√	√		01H
FFFFFD21H	CSIB2 control register 1	CB2CTL1		√	√		00H
FFFFFD22H	CSIB2 control register 2	CB2CTL2			√		00H
FFFFFD23H	CSIB2 status register	CB2STR		√	√		00H
FFFFFD24H	CSIB2 receive data register	CB2RX	R			√	0000H
FFFFFD24H	CSIB2 receive data register L	CB2RXL			√		00H
FFFFFD26H	CSIB2 transmit data register	CB2TX	R/W			√	0000H
FFFFFD26H	CSIB2 transmit data register L	CB2TXL			√		00H

3.4.8 Programmable peripheral I/O register

The peripheral I/O area select control register (BPC) is used to select a programmable peripheral I/O register area.

Peripheral I/O registers for the CAN controller are allocated to addresses 03FEC000H to 03FEE6EFH of the programmable peripheral I/O register area. For details, see CHAPTER 15 CAN CONTROLLER.

(1) Peripheral I/O select control register (BPC)

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF064H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPC	PA15	0	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

PA15	Enable or disable of use of programmable peripheral I/O area
0	Disable use of programmable peripheral I/O area.
1	Enable use of programmable peripheral I/O area.

PA13 to PA0	Setting of resumption address of programmable peripheral I/O area (correspond to A27 to A14).

Caution Be sure to set the BPC register to 8FFBH when the PA15 bit is set to 1.
Be sure to set the BPC register to 0000H when the PA15 bit is cleared to 0.

3.4.9 Special registers

Special registers are protected so that no illegal data is written to them in case of a program loop. The V850ES/Fx2 have the following seven special registers.

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

A command register (PRCMD) is provided as a register that protects the special registers from a write operation so that the application system does not stop inadvertently in case of a program loop. A write access to a special register is performed in a specific sequence, and an illegal store operation is reported to the system status register (SYS) (if an operation to read option data (address: 007AH) is illegal due to noise or instantaneous voltage drop, it is also reported to the system status register (SYS)).

(1) Setting data to special register

Data is set to the special registers in the following sequence.

<1> Disable the DMA operation.
<2> Prepare data to be set to a special register, in a general-purpose register.
<3> Write the data prepared in <2> to the command register (PRCMD).
<4> Write the 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 (five instructions).
<10> Enable DMA operation if necessary.

[Example] To set data to the PSC register (to set standby mode)

```
    ST.B r11, PSMR [r0]    ; Setting PSMR register (setting IDLE1, IDLE2, or software STOP mode).
<1> CLR1 0, DCHCn [r0]     ; Disabling DMA operation.  n = 0 to 3
<2> MOV 0x02, r10
<3> ST.B r10, PRCMD [r0]   ; Writing PRCMD register.
<4> ST.B r10, PSC [r0]     ; Setting 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]    ; Enabling DMA operation.  n = 0 to 3
(next instruction)
```

No specific sequence is necessary for reading a special register.

Cautions 1. The instruction that stores data in the command register does not acknowledge an interrupt. This is because it is assumed that steps <3> and <4> above are executed by successive store instructions. If an other instruction is written between <3> and <4>, and if that instruction acknowledges an interrupt, the above sequence may not be established, causing malfunction.

2. Dummy data is written to the PRCMD register. Use the same general-purpose register as the one used to set the special register (<4> in the above example) for writing the PRCMD register (<3>). The same applies when using a general-purpose register for addressing.

3. When shifting to IDLE1, IDLE2, software STOP mode and sub-IDLE mode (STP bit of PSC register = 1), five NOP instructions or more must be inserted immediately after entering that mode.

(2) Command register (PRCMD)

The command register (PRCMD) is an 8-bit register that is used to protect a register that may seriously affect the system from a write operation, so that the application system does not stop inadvertently in case of a program loop. Only the first writing of a special register is valid after a write operation is performed on the PRCMD register in advance. The value written to the PRCMD register can be rewritten only in a specific sequence, so that an illegal write operation cannot be executed.

The PRCMD register is write-only; in 8-bit units (if this register is read, illegal data is read).

This register becomes undefined at reset.

After reset: Undefined W Address: FFFFF1FCH

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

(3) System status register (SYS)

A status flag that indicates the overall operating status of the system is allocated to this register.

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

This register becomes 00H at reset.

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR

PRERR	Detection of protection error
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Setting condition (PRERR flag = 1)

(i) When a write operation is not performed on the PRCMD register and an operation to write a special register is performed (when <4> in the example in **3.4.9 (1) Setting data to special register** is executed without <3>)

(ii) If a write operation (including a bit manipulation instruction) is performed on an on-chip peripheral I/O register other than a special register after a write operation to the PRCMD register (when <4> in the example in **3.4.9 (1) Setting data to special register** is not for a special register)

Remark Even if an on-chip peripheral I/O register is read (including a bit manipulation instruction) between writing the PRCMD register and writing a special register such as an access to the internal RAM, the PRERR flag is not set, and data can be written to the special register.

(b) Clearing condition (PRERR flag = 0)

(i) When 0 is written to the PRERR flag of the SYS register

(ii) When system reset is executed

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

2. If a write operation is performed on the PRCMD register, which is not a special register immediately after a write operation to the PRCMD register, the PRERR bit is set to 1.

3.4.10 Cautions

(1) Registers to be set first

Be sure to set the following registers first when using the V850ES/Fx2.

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

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

When using the external bus, place each pin in the control mode by setting the port-related registers immediately after setting the above registers.

(a) System wait control register (VSWC)

The VSWC register is used to control the wait cycle of a bus access to an on-chip peripheral I/O register.

Three clocks are required to access an on-chip peripheral I/O register (when no wait cycle is used). The V850ES/Fx2 requires a wait cycle depending on the operating frequency used. Set the following value to the VSWC register.

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

Operating Frequency (f_{CLK})	Set Value of VSWC	Number of Wait Cycles
32 kHz ≤ f_{CLK} < 16.6 MHz	00H	0
16.6 MHz ≤ f_{CLK} ≤ 20 MHz	01H	1

Remark If an attempt to change the contents of one of the following registers by hardware conflicts with a CPU access to the register, the register access is kept waiting. Consequently, an access to an on-chip peripheral I/O register may take a longer time than usual.

Peripheral Function	Register Name
Timer P (n = 0 to 3)	TPnCCR0, TPnCCR1, TPnCNT
Timer P (n = 1, 2)	TQnCCR0, TQnCCR1, TQnCCR2, TQnCCR3, TQnCNT
Watchdog timer 2	WDTM2
A/D converter (n = 0 to 23)	ADA0M0, ADA0CRn, ADA0CRnH
CAN controller	Each control register, each message buffer register

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

For details, see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

(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**.

(2) Accessing specific on-chip peripheral I/O registers

This product has two types of internal system buses.

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

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

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

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

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

(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, TPnCCR1	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
16-bit timer/event counter Q (TMQ) (n= 0 (V850ES/FE2, V850ES/FF2)) (n= 0, 1 (V850ES/FG2)) (n= 0 to 3 (V850ES/FJ2))	TQnCNT	Read	1 or 2
	TQnCCR0 to TQnCCR3	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 (n= 9 (V850ES/FE2)) (n= 11 (V850ES/FF2)) (n= 15 (V850ES/FG2)) (n= 23 (V850ES/FJ2))	ADA0M0	Read	1 or 2
	ADA0CR0 to ADA0CRn	Read	1 or 2
	ADA0CR0H to ADA0CRnH	Read	1 or 2

(2/2)

Peripheral Function	Register Name	Access	k
CAN controller (n= 0 (μPD703230)) (n= 0 (μPD70F3231)) (n= 0 (μPD703232)) (n= 0 (μPD703233)) (n = 0, 1 (μPD70F3234)) (n = 0, 1 (μPD70F3235)) (n = 0, 1 (μPD70F3236)) (n = 0, 1 (μPD70F3237)) (n = 0 to 3 (μPD70F3238)) (n = 0 to 3 (μPD70F3239)) (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	$(f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(2 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
	CnRGPT, CnTGPT	Write	$(f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(2 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
		Read	$(3 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(4 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
	CnLIPT, CnLOPT	Read	$(3 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(4 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
	CnMDATA01m, CnMDATA0m, CnMDATA1m, CnMDATA23m, CnMDATA2m, CnMDATA3m, CnMDATA45m, CnMDATA4m, CnMDATA5m, CnMDATA67m, CnMDATA6m, CnMDATA7m, CnMDLCm, CnMCONFm, CnMIDLm, CnMIDHm, CnMCTRLm	Write (8 bits)	$(4 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(5 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
		Write (16 bits)	$(2 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(3 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]
		Read (8/16 bits)	$(3 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.)[Note] $(4 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.)[Note]

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

Note Digits below the decimal point are rounded up.

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

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

Remark f_{XX}: Main clock frequency = f_{XX}
f_{CANMOD}: 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

CHAPTER 4 PORT FUNCTIONS

4.1 Features

- O I/O ports: 51 (V850ES/FE2), 67 (V850ES/FF2), 84 (V850ES/FG2), or 128 (V850ES/FJ2)
- O Port pins function alternately as other peripheral-function I/O pins
- O Can be set in input or output mode in 1-bit units.

4.2 Basic Port Configuration

4.2.1 Basic Port Configuration on V850ES/FE2

The V850ES/FE2 has a total of 51 I/O ports, ports 0, 3 to 5, 7, 9, CM and DL. The port configuration is shown below.

Figure 4-1. Port Configuration (V850ES/FE2)

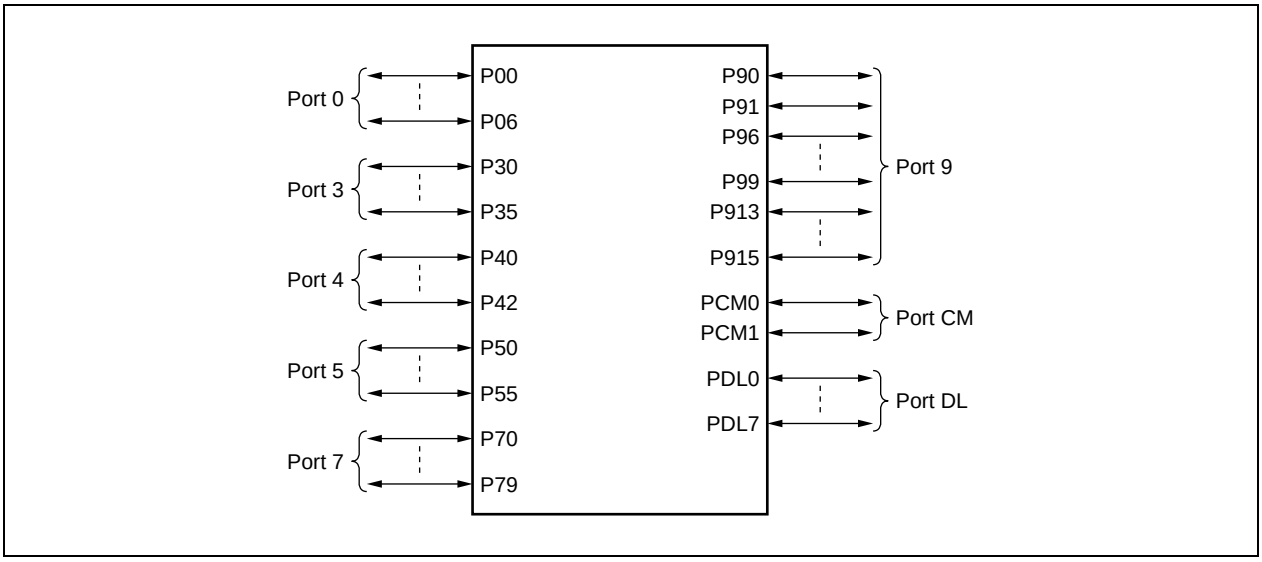


Table 4-1 Pin I/O Buffer Power Supplies (V850ES/FE2)

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

4.2.2 Port Configuration on V850ES/FF2

The V850ES/FF2 has a total of 67 I/O ports, ports 0, 3 to 5, 7, 9, CM, CS, CT and DL. The port configuration is shown below.

Figure 4-2. Port Configuration (V850ES/FF2)

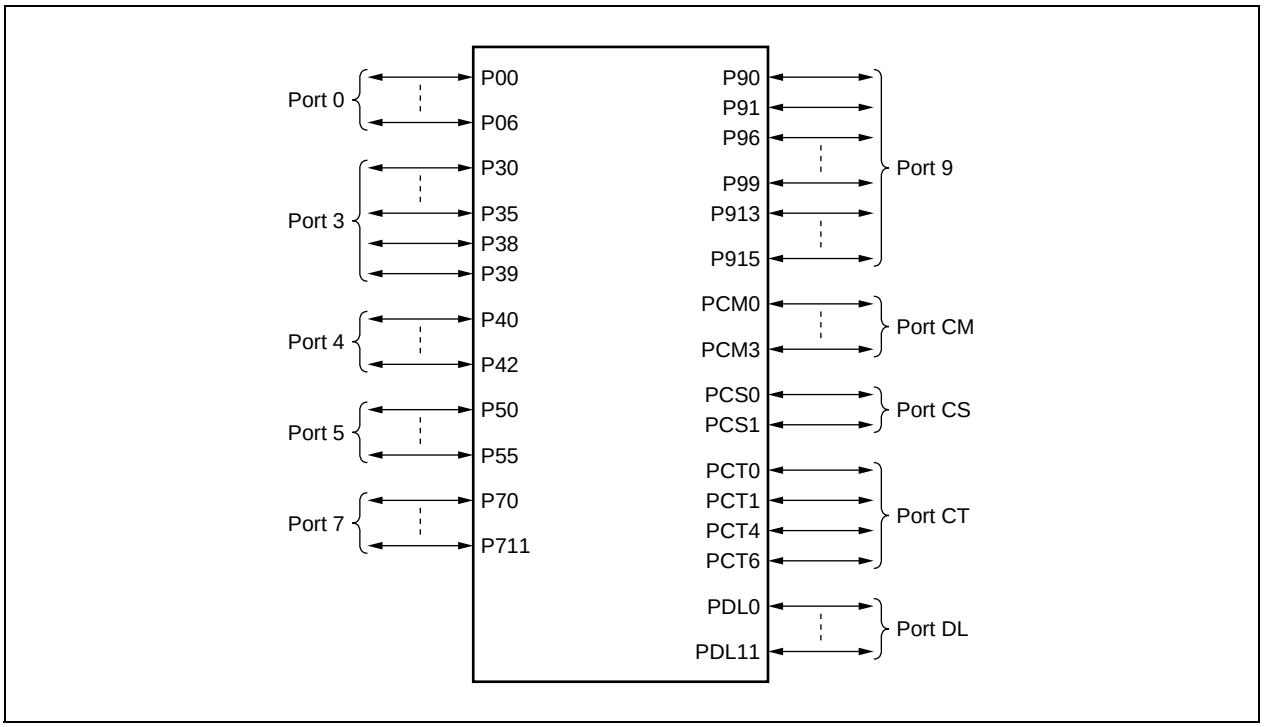


Table 4-2 Pin I/O Buffer Power Supplies (V850ES/FF2)

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

4.2.3 Port Configuration on V850ES/FG2

The V850ES/FG2 has a total of 84 I/O ports, ports 0, 1, 3 to 5, 7, 9, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-3. Port Configuration (V850ES/FG2)

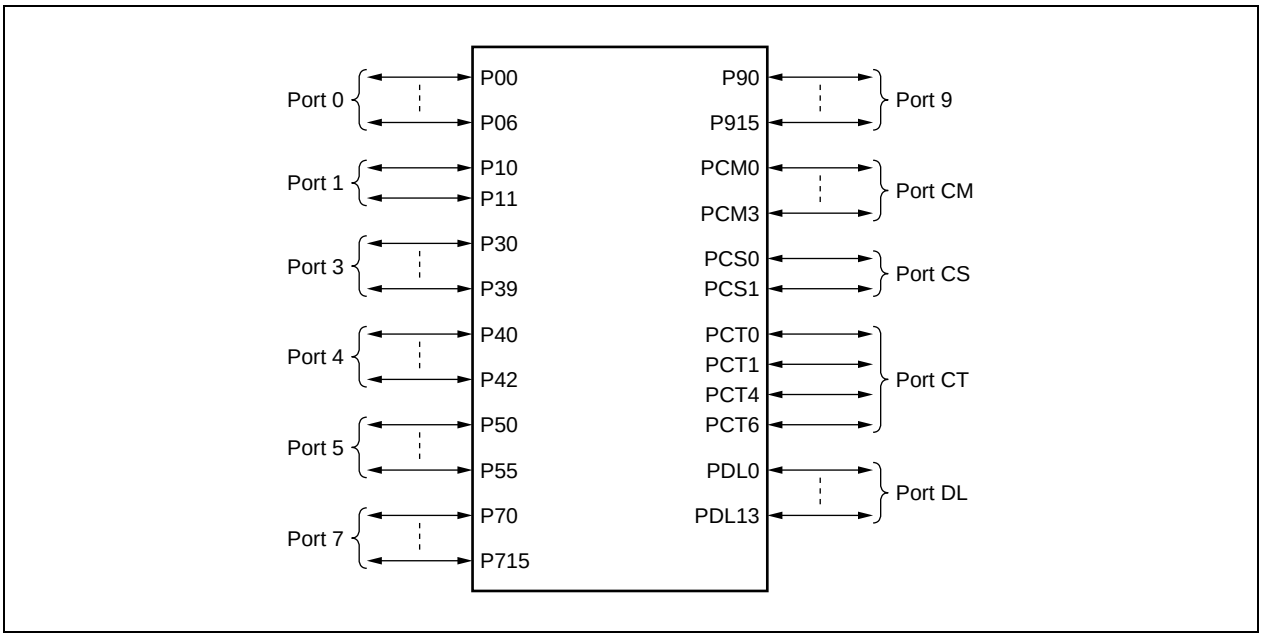


Table 4-3 Pin I/O Buffer Power Supplies (V850ES/FG2)

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

4.2.4 Port Configuration on V850ES/FJ2

The V850ES/FJ2 features a total of 128 I/O ports consisting of ports 0, 1, 3 to 9, 12, CD, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-4. Port Configuration (V850ES/FJ2)

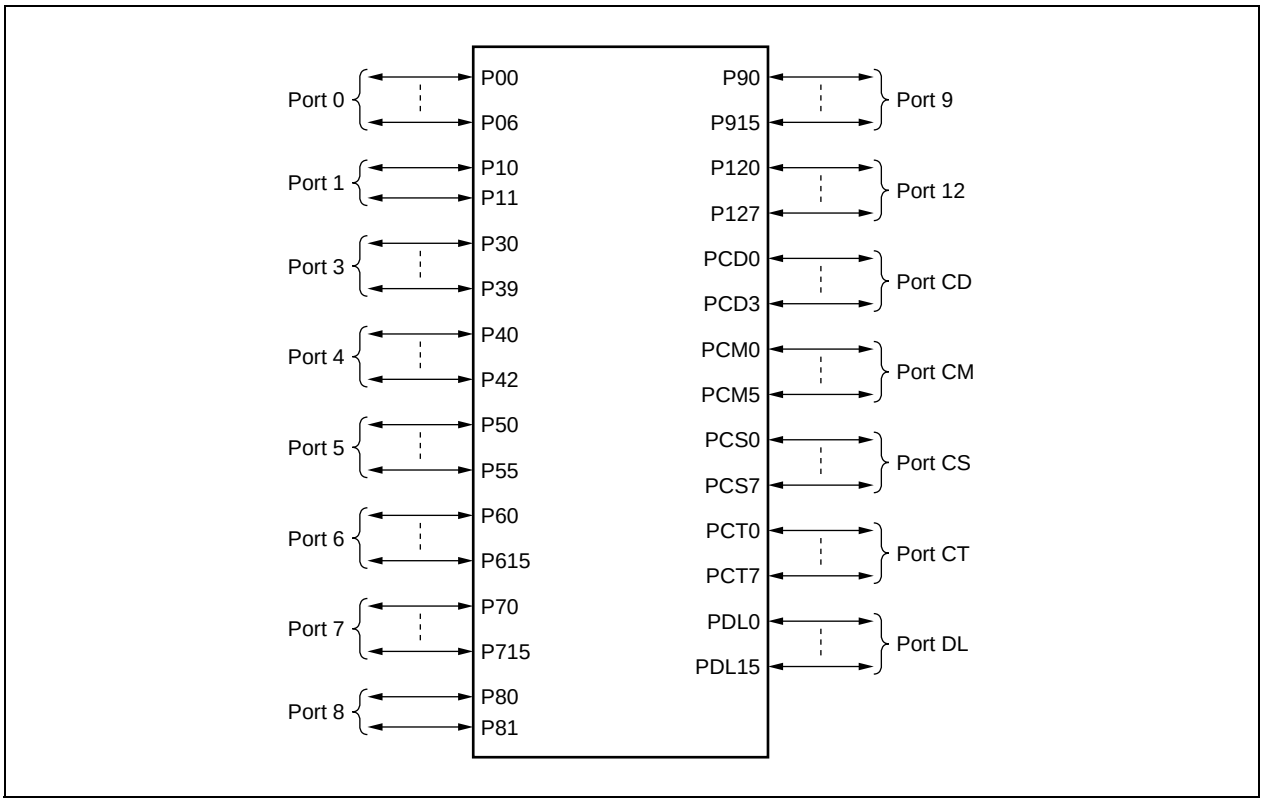


Table 4-4 Pin I/O Buffer Power Supplies (V850ES/FJ2)

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

4.3 Port Configuration

The following tables give the relevant registers to configure ports on V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2.

Table 4-5 Port Configuration (V850ES/FE2)

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

Table 4-6 Port Configuration (V850ES/FF2)

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

Table 4-7 Port Configuration (V850ES/FG2)

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

Table 4-8 Port Configuration (V850ES/FJ2)

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

(1) Port n register (Pn)

Data is input from or output to an external device by writing or reading the Pn register.

The Pn register consists of a port latch that holds output data, and a circuit that reads the status of pins.

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

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

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

Pnm	Control of output data (in output mode)
0	Outputs 0
1	Outputs 1

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

Table 4-9 Writing/Reading Pn Register

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

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

(2) Port n mode register (PMn)

The PMn register specifies the input or output mode of the corresponding port pin.

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

After reset: FFH R/W

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

PMnm	Control of input/output mode
0	Output mode
1	Input mode

(3) Port n mode control register (PMCn)

The PMCn register specifies the port mode or alternate function.

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

After reset: 00H R/W

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

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

(4) Port n function control register (PFCn)

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

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

After reset: 00H R/W

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

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

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

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

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

After reset: 00H R/W

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

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

(6) Port n function register (PFn)

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

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

After reset: 00H R/W

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

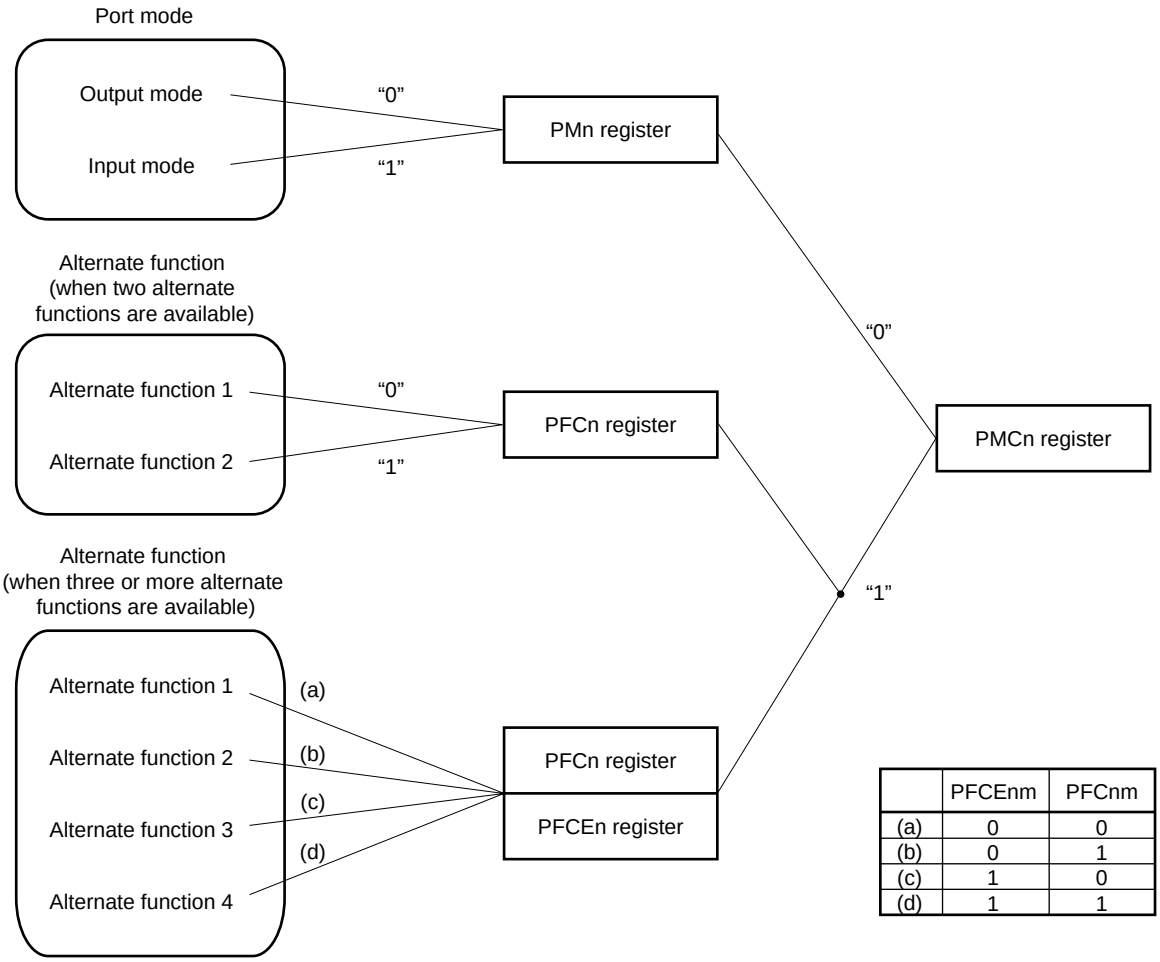
PFnm[Note]	Control of normal output/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output

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

(7) Port setting

Set a port as illustrated below.

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



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

Remark Set the alternate functions in the following sequence.

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

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

4.3.1 Port 0

Port 0 is a 7-bit (P00 to P06) port for which I/O settings can be controlled in 1-bit units.
The number of I/O port pins is the same for all products.

Product	Number of I/O Port Pins
V850ES/FE2	7-bit I/O port (P00 to P06)
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	

(1) Functions of port 0

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 0 (P0)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 0 (PM0)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 0 (PMC0)
- Control mode 1 or control mode 2 can be specified in 1-bit units.
  Specified by port function control register 0 (PFC0)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 0 (PU0)
- The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 0 (INTF0) and external interrupt rising edge specification register 0 (INTR0)

Port 0 includes the following alternate-function pins.

Table 4-10 Alternate-Function Pins of Port 0

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 0	P00	TP31/TOP31	I/O		G-1
	P01	TP30/TOP30			G-1
	P02	NMI Note 1			L-1
	P03	INTP0/ADTRG			N-1
	P04	INTP1			L-1
	P05	INTP2/$\overline{\text{DRST}}$ Note 2			AA-1
	P06	INTP3			L-2

Notes 1. The NMI pin is used in combination with the P02 pin.

(1) After reset the P02 pin function is active. Set the PMC0.PMC02 bit when you make NMI effective.

(2) Moreover, the NMI pin initialization is "No edge detection". Select an effective edge of the NMI pin by the INTF0 and the INTR0 register.

2. The $\overline{\text{DRST}}$ pin is for on-chip debugging (flash memory version only).

If on-chip debugging is not used, fix the P05/INTP2/$\overline{\text{DRST}}$ pin to low level between when the reset signal of the RESET pin is released and when the OCDM.OCDM0 bit is cleared (0). Although the mask ROM versions do not support the on-chip debug mode, handle the P05/INTP2 pin the same as in flash memory versions.

For details, see **4.4.3 Cautions on on-chip debug pins.**

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

(2) Registers

(a) Port register 0 (P0)

Port register 0 (P0) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF400H

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

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

(b) Port mode register 0 (PM0)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF420H

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

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

(c) Port mode control register 0 (PMC0)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF440H

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

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

PMC05	Specification of operation mode of P05 pin
0	I/O port
1	INTP2/$\overline{\text{DRST}}$ input

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

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

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

PMC01	Specification of operation mode of P01 pin
0	I/O port
1	TIP30/TOP30 I/O

PMC00	Specification of operation mode of P00 pin
0	I/O port
1	TIP31/TOP31 I/O

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

(d) Port function control register 0 (PFC0)

This is an 8-bit register that specifies control mode 1 or control mode 2. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF460H

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

PFC03	Specification of operation mode when P03 pin is in control mode
0	INTP0 input
1	ADTRG input

PFC01	Specification of operation mode when P01 pin is in control mode
0	TIP30 input
1	TOP30 output

PFC00	Specification of operation mode when P00 pin is in control mode
0	TIP31 input
1	TOP31 output

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC40H

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

PU0n	Control of on-chip pull-up resistor connection (n = 0 to 6)
0	Not connected
1	Connected

(f) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF0n and INTR0n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

3. For how to set the internal noise filter (analog delay/digital delay) of INTP3, refer to CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION.

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0

Remark Refer to **Table 4-11** for how to specify a valid edge.

(g) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF0n and INTR0n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

3. For how to set the internal noise filter (analog delay/digital delay) of INTP3, refer to CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION.

After reset: 00H R/W Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0

Remark Refer to **Table 4-11** for how to specify a valid edge.

Table 4-11 Valid Edge Specification

INTF0n Bit	INTR0n Bit	Valid Edge Specification (n = 2 to 6)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 2: Control of NMI pin
n = 3: Control of INTP0 pin
n = 4: Control of INTP1 pin
n = 5: Control of INTP2 pin
n = 6: Control of INTP3 pin

4.3.2 Port 1

Port 1 is a 2-bit port (P10 and P11) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	
V850ES/FG2	2-bit I/O port (P10 and P11)
V850ES/FJ2	

(1) Functions of port 1

- O The input/output data of the port can be specified in 1-bit units.
  Specified by port register 1 (P1)
- O The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 1 (PM1)
- O Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 1 (PMC1)
- O An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 1 (PU1)
- O The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 1 (INTF1) and external interrupt rising edge specification register 1 (INTR1)

Port 1 functions alternately as the following pins.

Table 4-12 Alternate-Function Pins of Port 1

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 1	P10	INTP9	I/O	–	L-1
	P11	INTP10			L-1

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

(2) Registers

(a) Port register 1 (P1)

Port register 1 (P1) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF402H

(i) V850ES/FG2, V850ES/FJ2

	7	6	5	4	3	2	1	0
P1	0	0	0	0	0	0	P11	P10

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

(b) Port mode register 1 (PM1)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF422H

(i) V850ES/FG2, V850ES/FJ2

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

PM1n	Control of input/output mode (n = 0, 1)
0	Output mode
1	Input mode

(c) Port mode control register 1 (PMC1)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF442H

(i) V850ES/FG2, V850ES/FJ2

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

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

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

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC42H

(i) V850ES/FG2, V850ES/FJ2

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

PU1n	Control of on-chip pull-up resistor connection (n = 0, 1)
0	Not connected
1	Connected

(e) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF1n and INTR1n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC02H

(i) V850ES/FG2, V850ES/FJ2

	7	6	5	4	3	2	1	0
INTF1	0	0	0	0	0	0	INTF11	INTF10

Remark Refer to **Table 4-13** for how to specify a valid edge.

(f) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF1n and INTR1n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC22H

(i) V850ES/FG2, V850ES/FJ2

	7	6	5	4	3	2	1	0
INTR1	0	0	0	0	0	0	INTR11	INTR10

Remark Refer to **Table 4-13** for how to specify a valid edge.

Table 4-13 Valid Edge Specification

INTF1n Bit	INTR1n Bit	Valid Edge Specification (n = 0, 1)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 0: Control of INTP9 pin
n = 1: Control of INTP10 pin

4.3.3 Port 3

Port 3 is a 10-bit port (P30 to P39) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	6-bit I/O port (P30 to P35)
V850ES/FF2	8-bit I/O port (P30 to P35, P38, P39)Note
V850ES/FG2	10-bit I/O port (P30 to P39)
V850ES/FJ2	

Note In the V850ES/FF2, the alternate functions of the P38 and P39 pins (TXDA2, RXDA2/INTP8) are not available.

(1) Function of port 3

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 3 (P3)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 3 (PM3)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 3 (PMC3)
- Control mode can be specified in 1-bit units.
  Specified by port function control register 3 (PFC3) and port function control expansion register 3L (PFCE3L)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 3 (PU3)
- The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 3 (INTF3) and external interrupt rising edge specification register 3 (INTR3)

Port 3 functions alternately as the following pins.

Table 4-14 Alternate-Function Pins of Port 3

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 3	P30	TXDA0	I/O		E-2
	P31	RXDA0/INTP7			L-2
	P32	ASCKA0/TIP00/TOP00/TOP01			U-13
	P33	TIP01/TOP01/CTXD0			U-3
	P34	TIP10/TOP10/CRXD0			U-2
	P35	TIP11/TOP11			G-1
	P36	CTXD1			E-2
	P37	CRXD1			E-1
	P38	TXDA2			E-2
	P39	RXDA2/INTP8			L-2

Caution: The P31 to P35, P37, P39 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(2) Registers

(a) Port register 3 (P3)

Port register 3 (P3) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P3 register are used as the P3H register, and the lower 8 bits as the P3L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF406H, FFFFF407H

(i) V850ES/FE2

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

(ii) V850ES/FF2

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

(iii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
P3 (P3H[Note])	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Control of output data (in output mode) (n = 0 to 9)
0	Output 0.
1	Output 1.

Note To read or write bits 8 to 15 of the P3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the P3H register. Note that the V850ES/FE2 is not provided with a P3H register. Therefore, the P3 register can be used only as the P3L register in the V850ES/FE2.

(b) Port mode register 3 (PM3)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PM3 register are used as the PM3H register, and the lower 8 bits as the PM3L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: FFFFH R/W Address: FFFFF426H, FFFFF427H

(i) V850ES/FE2

	15	14	13	12	11	10	9	8
PM3 (PM3H[Note])	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
(PM3L)	1	1	PM35	PM34	PM33	PM32	PM31	PM30

(ii) V850ES/FF2

	15	14	13	12	11	10	9	8
PM3 (PM3H[Note])	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	1	1	PM35	PM34	PM33	PM32	PM31	PM30

(iii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PM3 (PM3H[Note])	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

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

Note To read or write bits 8 to 15 of the PM3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PM3H register. Note that the V850ES/FE2 is not provided with a PM3H register. Therefore, the PM3 register can be used only as the PM3L register in the V850ES/FE2.

(c) Port mode control register 3 (PMC3)

This is a 16-bit register that specifies the port mode or control mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PMC3 register are used as the PMC3H register, and the lower 8 bits as the PMC3L register, however, these registers can be read or written in 8-bit or 1-bit units.

(1/2)

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

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PMC3 (PMC3H[Note 1])	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PMC3L)	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PMC3 (PMC3H[Note 1])	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
(PMC3L)	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	Specification of operation mode of P39 pin
0	I/O port
1	RXDA2/INTP8 input[Note 2]

PMC38	Specification of operation mode of P38 pin
0	I/O port
1	TXDA2 output

Notes 1. To read or write bits 8 to 15 of the PMC3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMC3H register. Note that the V850ES/FE2, V850ES/FF2 are not provided with a PMC3H register. Therefore, the PMC3 register can be used only as the PMC3L register in the V850ES/FE2, V850ES/FF2.

2. The INTP8 pin functions alternately as the RXDA2 pin. To use as the RXDA2 pin, invalidate the edge detection function of the alternate-function INTP8 pin (by fixing the INTF39 bit of the INTF3 register to 0 and the INTR39 bit of the INTR3 register to 0). To use as the INTP8 pin, stop the reception operation of UARTA2 (by clearing the UA2RXE bit of the UA2CTL0 register to 0).

(2/2)

PMC37	Specification of operation mode of P37 pin
0	I/O port
1	CRXD1 input

PMC36	Specification of operation mode of P36 pin
0	I/O port
1	CTXD1 output

PMC35	Specification of operation mode of P35 pin
0	I/O port
1	TIP11/TOP11 I/O

PMC34	Specification of operation mode of P34 pin
0	I/O port
1	TIP10/TOP10/CRXD0 I/O

PMC33	Specification of operation mode of P33 pin
0	I/O port
1	TIP01/TOP01/CTXD0 I/O

PMC32	Specification of operation mode of P32 pin
0	I/O port
1	ASCKA0/TIP00/TOP00/TOP01 I/O

PMC31	Specification of operation mode of P31 pin
0	I/O port
1	RXDA0/INTP7 input[Note]

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

Note The INTP7 pin functions alternately as the RXDA0 pin. To use as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by fixing the INTF31 bit of the INTF3 register to 0 and the INTR31 bit of the INTR3 register to 0). To use as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0RXE bit of the UA0CTL0 register to 0).

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF466H

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

Remark For how to specify a control mode, refer to **4.3.4 (2) (f) Setting of control mode of P3 pin**.

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	PFCE34	PFCE33	PFCE32	0	0

Remark For how to specify a control mode, refer to **4.3.4 (2) (f) Setting of control mode of P3 pin**.

(f) Setting of control mode of P3 pin

PFC35	Specification of control mode of P35 pin
0	TIP11 input
1	TOP11 output

PFCE34	PFC34	Specification of control mode of P34 pin
0	0	TIP10 input
0	1	TOP10 output
1	0	CRXD0 input
1	1	Setting prohibited

PFCE33	PFC33	Specification of control mode of P33 pin
0	0	TIP01 input
0	1	TOP01 output
1	0	CTXD0 output
1	1	Setting prohibited

PFCE32	PFC32	Specification of control mode of P32 pin
0	0	ASCKA0 input
0	1	TOP01 output
1	0	TIP00 input
1	1	TOP00 output

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

This is a 16-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 16- or 1-bit units.

If the higher 8 bits of the PU3 register are used as the PU3H register, and the lower 8 bits as the PU3L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC46H, FFFFFC47H

(i) V850ES/FE2

	15	14	13	12	11	10	9	8
PU3 (PU3H[Note])	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PU3L)	0	0	PU35	PU34	PU33	PU32	PU31	PU30

(ii) V850ES/FF2

	15	14	13	12	11	10	9	8
PU3 (PU3H[Note])	0	0	0	0	0	0	PU39	PU38
	7	6	5	4	3	2	1	0
(PU3L)	0	0	PU35	PU34	PU33	PU32	PU31	PU30

(iii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PU3 (PU3H[Note])	0	0	0	0	0	0	PU39	PU38
	7	6	5	4	3	2	1	0
(PU3L)	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 9)
0	Not connected
1	Connected

Note To read/write bits 8 to 15 of the PU3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PU3H register. Note that the V850ES/FE2 is not provided with a PU3H register. Therefore, the PU3 register can be used only as the PU3L register in the V850ES/FE2.

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

This is a 16-bit register that specifies detection of the falling edge of the external interrupt pin. It can be read or written in 16-bit units.

If the higher 8 bits of the INTF3 register are used as the INTF3H register, and the lower 8 bits as the INTF3L register, however, these registers can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF3n and INTR3n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC06H, FFFFFC07H

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
INTF3 (INTF3H[Note])	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(INTF3L)	0	0	0	0	0	0	INTF31	0

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
INTF3 (INTF3H[Note])	0	0	0	0	0	0	INTF39	0
	7	6	5	4	3	2	1	0
(INTF3L)	0	0	0	0	0	0	INTF31	0

Note To read/write bits 8 to 15 of the INTF3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the INTF3H register. Note that the V850ES/FE2, V850ES/FF2 is not provided with an INTF3H register. Therefore, the INTF3 register can be used only as the INTF3L register in the V850ES/FE2, V850ES/FE2.

Remark Refer to **Table 4-15** for how to specify a valid edge.

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

This is a 16-bit register that specifies detection of the rising edge of the external interrupt pin. It can be read or written in 16-bit units.

If the higher 8 bits of the INTR3 register are used as the INTR3H register, and the lower 8 bits as the INTR3L register, however, these registers can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF3n and INTR3n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC26H, FFFFFC27H

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
INTR3 (INTR3H[Note])	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(INTR3L)	0	0	0	0	0	0	INTR31	0

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
INTR3 (INTR3H[Note])	0	0	0	0	0	0	INTR39	0
	7	6	5	4	3	2	1	0
(INTR3L)	0	0	0	0	0	0	INTR31	0

Note To read/write bits 8 to 15 of the INTR3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the INTR3H register. Note that the V850ES/FE2, V850ES/FF2 is not provided with an INTR3H register. Therefore, the INTR3 register can be used only as the INTR3L register in the V850ES/FE2, V850ES/FF2.

Remark Refer to **4-15** for how to specify a valid edge.

Table 4-15 Valid Edge Specification

INTF3n Bit	INTR3n Bit	Valid Edge Specification (n = 1, 9)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 1: Control of INTP7 pin
n = 9: Control of INTP8 pin

4.3.4 Port 4

Port 4 is a 3-bit port (P40 to P42) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins is the same for all products.

Product	Number of I/O Port Pins
V850ES/FE2	3-bit I/O port (P40 to P42)
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	

(1) Functions of port 4

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 4 (P4)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 4 (PM4)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 4 (PMC4)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 4 (PU4)

Port 4 functions alternately as the following pins.

Table 4-16 Alternate-Function Pins of Port 4

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 4	P40	SIB0	I/O	–	E-1
	P41	SOB0			E-2
	P42	$\overline{\text{SCKB0}}$			E-3

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

(2) Registers

(a) Port register 4 (P4)

Port register 4 (P4) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF408H

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

P4n	Control of output data (in output mode) (n = 0 to 2)
0	Output 0.
1	Output 1.

★

(b) Port mode register 4 (PM4)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF428H

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

PM4n	Control of input/output mode (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port mode control register 4 (PMC4)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF448H

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

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

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

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

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC48H

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

PU4n	Control of on-chip pull-up resistor connection (n = 0 to 2)
0	Not connected
1	Connected

4.3.5 Port 5

Port 5 is a 6-bit port (P50 to P55) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins is the same for all products.

Product	Number of I/O Port Pins
V850ES/FE2	6-bit I/O port (P50 to P55)
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	

(1) Functions of port 5

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 5 (P5)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 5 (PM5)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 5 (PMC5)
- Control mode can be specified in 1-bit units.
  Specified by port function control register 5 (PFC5) or port function control expansion register 5 (PFCE5)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 5 (PU5)

Port 5 functions alternately as the following pins.

Table 4-17 Alternate-Function Pins of Port 5

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 5	P50	KR0/TIQ01/TOQ01	I/O	–	U-4
	P51	KR1/TIQ02/TOQ02			U-4
	P52	KR2/TIQ03/TOQ03/DDI[Note]			U-5
	P53	KR3/TIQ00/TOQ00/DDO[Note]			U-6
	P54	KR4/DCK[Note]			G-2
	P55	KR5/DMS[Note]			G-2

Caution: The P50 to P55 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

Note The DDI, DDO, DCK, and DMS pins are for the on-chip debug function. To use the DDI, DDO, DCK, and DMS pins as port pins, not as on-chip debug pins, the following actions must be taken.

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

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

(2) Registers

(a) Port register 5 (P5)

Port register 5 (P5) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50

P5n	Control of output data (in output mode) (n = 0 to 5)
0	Output 0.
1	Output 1.

(b) Port mode register 5 (PM5)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF42AH

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

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

(c) Port mode control register 5 (PMC5)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC5 register when the PFC5n bit of the PFC5 register and the PFCE5n bit of the PFCE5 register are the default values (0), the output becomes undefined.
For this reason, first set the PFC5n bit of the PFC5 register and the PFCE5n bit of the PFCE5 register, and then set the PMC5n bit to 1 to set the control mode.

After reset: 00H R/W Address: FFFFF44AH

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

PMC55	Specification of operation mode of P55 pin
0	I/O port
1	KR5 input

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

PMC53	Specification of operation mode of P53 pin
0	I/O port
1	KR3/TIQ00/TOQ00 I/O

PMC52	Specification of operation mode of P52 pin
0	I/O port
1	KR2/TIQ03/TOQ03 I/O

PMC51	Specification of operation mode of P51 pin
0	I/O port
1	KR1/TIQ02/TOQ02 I/O

PMC50	Specification of operation mode of P50 pin
0	I/O port
1	KR0/TIQ01/TOQ01 I/O

(d) Port function control register 5 (PFC5)

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF46AH

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

Remark For how to specify a control mode, refer to **4.3.6 (2) (f) Setting of control mode of P5 pin**.

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF70AH

	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	PFCE53	PFCE52	PFCE51	PFCE50

Remark For how to specify a control mode, refer to **4.3.6 (2) (f) Setting of control mode of P5 pin**.

(f) Setting of control mode of P5 pin

Caution If the control mode is specified by using the PMC5 register when the PFC5n bit of the PFC5 register and PFCE5n bit of the PFCE5 register are the default values (0), the output becomes undefined.
For this reason, first set the PFC5n bit of the PFC5 register and the PFCE5n bit of the PFCE5 register, and then set the PMC5n bit to 1 to set the control mode.

PFC55	Specification of control mode of P55 pin
0	Setting prohibited
1	KR5 input

PFC54	Specification of control mode of P54 pin
0	Setting prohibited
1	KR4 input

PFCE53	PFC53	Specification of control mode of P53 pin
0	0	Setting prohibited
0	1	TIQ00/KR3[Note] input
1	0	TOQ00 output
1	1	Setting prohibited

PFCE52	PFC52	Specification of control mode of P52 pin
0	0	Setting prohibited
0	1	TIQ03/KR2[Note] input
1	0	TOQ03 output
1	1	Setting prohibited

PFCE51	PFC51	Specification of control mode of P51 pin
0	0	Setting prohibited
0	1	TIQ02/KR1[Note] input
1	0	TOQ02 output
1	1	Setting prohibited

PFCE50	PFC50	Specification of control mode of P50 pin
0	0	Setting prohibited
0	1	TIQ01/KR0[Note] input
1	0	TOQ01 output
1	1	Setting prohibited

Note The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRMn bit of the KRM register to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin (n = 0 to 3, m = 0 to 3).

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

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC4AH

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

PU5n	Control of on-chip pull-up resistor connection (n = 0 to 5)
0	Not connected
1	Connected

4.3.6 Port 6

Port 6 is a 16-bit port (P60 to P615) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	16-bit I/O port (P60 to P615)[Note]

Note In the μPD70F3237, the alternate functions of the P65 to P68 pins (CTXD2, CRXD2, CTXD3, CRXD3) are not available.

(1) Functions of port 6

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 6 (P6)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 6 (PM6)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 6 (PMC6)
- Control mode 1 or control mode 2 can be specified in 1-bit units.
  Specified by port function control register 6 (PFC6)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 6 (PU6)
- The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 6L (INTF6L) and external interrupt rising edge specification register 6L (INTR6L)

Port 6 functions alternately as the following pins.

Table 4-18 Alternate-Function Pins of Port 6

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 6	P60	INTP11	I/O	–	N-2
	P61	INTP12			N-2
	P62	INTP13			N-2
	P63	–			C-1
	P64	–			C-1
	P65	CTXD2			G-3
	P66	CRXD2			G-4
	P67	CTXD3			G-3
	P68	CRXD3			G-4
	P69	–			C-1
	P610	TIQ20/TOQ20			G-1
	P611	TIQ21/TOQ21			G-1
	P612	TIQ22/TOQ22			G-1
	P613	TIQ23/TOQ23			G-1
	P614	–			C-1
	P615	–			C-1

Caution: The P60 to P62, P66, P68, P610 to P613 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode. (P66 and P68 only μPD70F3238, μPD70F3239)

(2) Registers

(a) Port register 6 (P6)

Port register 6 (P6) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P6 register are used as the P6H register, and the lower 8 bits as the P6L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF40CH, FFFFF40DH

(i) V850ES/FJ2

	15	14	13	12	11	10	9	8
P6 (P6H[Note])	P615	P614	P613	P612	P611	P610	P69	P68
	7	6	5	4	3	2	1	0
(P6L)	P67	P66	P65	P64	P63	P62	P61	P60

P6n	Control of output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

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

(b) Port mode register 6 (PM6)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PM6 register are used as the PM6H register, and the lower 8 bits as the PM6L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF42CH, FFFFF42DH

(i) V850ES/FJ2

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

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

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

(c) Port mode control register 6 (PMC6)

This is a 16-bit register that specifies the port mode or control mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PMC6 register are used as the PMC6H register, and the lower 8 bits as the PMC6L register, however, these registers can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC6 register when the PFC6n bit of the PFC6 register is the default value (0), the output becomes undefined (n = 0 to 8).
For this reason, first set the PFC6n bit of the PFC6 register to 1, and then set the PMC6n bit to 1 to set the control mode.

(1/2)

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

(i) V850ES/FJ2 (μPD70F3237)

	15	14	13	12	11	10	9	8
PMC6 (PMC6H[Note])	0	0	PMC613	PMC612	PMC611	PMC610	0	0
	7	6	5	4	3	2	1	0
(PMC6L)	0	0	0	0	0	PMC62	PMC61	PMC60

(ii) V850ES/FJ2 (μPD70F3238,μPD70F3239)

	15	14	13	12	11	10	9	8
PMC6 (PMC6H[Note])	0	0	PMC613	PMC612	PMC611	PMC610	0	PMC68
	7	6	5	4	3	2	1	0
(PMC6L)	PMC67	PMC66	PMC65	0	0	PMC62	PMC61	PMC60

PMC613	Specification of operation mode of P613 pin
0	I/O port
1	TIQ23/TOQ23 I/O

PMC612	Specification of operation mode of P612 pin
0	I/O port
1	TIQ22/TOQ22 I/O

PMC611	Specification of operation mode of P611 pin
0	I/O port
1	TIQ21/TOQ21 I/O

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

(2/2)

PMC610	Specification of operation mode of P610 pin
0	I/O port
1	TIQ20/TOQ20 I/O

PMC68	Specification of operation mode of P68 pin
0	I/O port
1	CRXD3 input

PMC67	Specification of operation mode of P67 pin
0	I/O port
1	CTXD3 output

PMC66	Specification of operation mode of P66 pin
0	I/O port
1	CRXD2 input

PMC65	Specification of operation mode of P65 pin
0	I/O port
1	CTXD2 output

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

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

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

(d) Port function control register 6 (PFC6)

This is a 16-bit register that specifies control mode 1 or 2. It can be read or written in 16-bit units.

If the higher 8 bits of the PFC6 register are used as the PFC6H register, and the lower 8 bits as the PFC6L register, however, these registers can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC6 register when the PFC6n bit of the PFC6 register is the default value (0), the output becomes undefined (n = 0 to 8).
For this reason, first set the PFC6n bit of the PFC6 register to 1, and then set the PMC6n bit to 1 to set the control mode.

(1/2)

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

(i) V850ES/FJ2 (μPD70F3237)

	15	14	13	12	11	10	9	8
PFC6 (PFC6H[Note])	0	0	PFC613	PFC612	PFC611	PFC610	0	0
	7	6	5	4	3	2	1	0
(PFC6L)	0	0	0	0	0	PFC62	PFC61	PFC60

(ii) V850ES/FJ2 (μPD70F3238,μPD70F3239)

	15	14	13	12	11	10	9	8
PFC6 (PFC6H[Note])	0	0	PFC613	PFC612	PFC611	PFC610	0	PFC68
	7	6	5	4	3	2	1	0
(PFC6L)	PFC67	PFC66	PFC65	0	0	PFC62	PFC61	PFC60

PFC613	Specification of control mode of P613 pin
0	TIQ23 input
1	TOQ23 output

PFC612	Specification of control mode of P612 pin
0	TIQ22 input
1	TOQ22 output

PFC611	Specification of control mode of P611 pin
0	TIQ21 input
1	TOQ21 output

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

(2/2)

PFC610	Specification of control mode of P610 pin
0	TIQ20 input
1	TOQ20 output

PFC68	Specification of control mode of P68 pin
0	Setting prohibited
1	CRXD3 input

PFC67	Specification of control mode of P67 pin
0	Setting prohibited
1	CTXD3 output

PFC66	Specification of control mode of P66 pin
0	Setting prohibited
1	CRXD2 input

PFC65	Specification of control mode of P65 pin
0	Setting prohibited
1	CTXD2 output

PFC62	Specification of control mode of P62 pin
0	Setting prohibited
1	INTP13 input

PFC61	Specification of control mode of P61 pin
0	Setting prohibited
1	INTP12 input

PFC60	Specification of control mode of P60 pin
0	Setting prohibited
1	INTP11 input

(e) Pull-up resistor option register 6 (PU6)

This is a 16-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 16-bit units.

If the higher 8 bits of the PU6 register are used as the PU6H register, and the lower 8 bits as the PU6L register, however, these registers can be read or written in 8-bit or 1-bit units.

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

(i) V850ES/FJ2

	15	14	13	12	11	10	9	8
PU6 (PU6H[Note])	PU615	PU614	PU613	PU612	PU611	PU610	PU69	PU68
	7	6	5	4	3	2	1	0
(PU6L)	PU67	PU66	PU65	PU64	PU63	PU62	PU61	PU60

PU6n	Control of on-chip pull-up resistor connection (n = 0 to 15)
0	Not connected
1	Connected

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

(f) External interrupt falling edge specification register 6L (INTF6L)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF6n and INTR6n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC0CH

(i) V850ES/FJ2

	7	6	5	4	3	2	1	0
INTF6L	0	0	0	0	0	INTF62	INTF61	INTF60

Remark Refer to **Table 4-19** for how to specify a valid edge.

(g) External interrupt rising edge specification register 6L (INTR6L)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF6n and INTR6n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC2CH

(i) V850ES/FJ2

	7	6	5	4	3	2	1	0
INTR6L	0	0	0	0	0	INTR62	INTR61	INTR60

Remark Refer to **Table 4-19** for how to specify a valid edge.

Table 4-19 Valid Edge Specification

INTF6n Bit	INTR6n Bit	Valid Edge Specification (n = 0 to 2)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 0: Control of INTP11 pin
n = 1: Control of INTP12 pin
n = 2: Control of INTP13 pin

4.3.7 Port 7

Port 7 is a 10-bit, 12-bit or16-bit port (P70 to P715) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	10-bit I/O port (P70 to P79)
V850ES/FF2	12-bit I/O port (P70 to P711)
V850ES/FG2	16-bit I/O port (P70 to P715)
V850ES/FJ2	

(1) Functions of port 7

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 7 (P7)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 7L, 7H (P7L, P7H)

Port 7 functions alternately as the following pins.

Table 4-20 Alternate-Function Pins of Port 7

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 7	P70	ANI0	I/O	–	A-1
	P71	ANI1			A-1
	P72	ANI2			A-1
	P73	ANI3			A-1
	P74	ANI4			A-1
	P75	ANI5			A-1
	P76	ANI6			A-1
	P77	ANI7			A-1
	P78	ANI8			A-1
	P79	ANI9			A-1
	P710	ANI10			A-1
	P711	ANI11			A-1
	P712	ANI12			A-1
	P713	ANI13			A-1
	P714	ANI14			A-1
	P715	ANI15			A-1

(2) Registers

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

Port registers 7H and 7L (P7H and P7L) are 8-bit registers that control reading the pin level and writing the output level. These registers can be read or written in 8-bit or 1-bit units.

They cannot be accessed in 16-bit units.

After reset: Undefined R/W Address: FFFFF40FH, FFFFF40EH

(i) V850ES/FE2

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	0	0	P79	P78
	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

(ii) V850ES/FF2

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

(iii) V850ES/FG2, V850ES/FJ2

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

P7n	Control of output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

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

(b) Port mode registers 7H, 7L (PM7H, PM7L)

These are 8-bit registers that specify an input or output mode. They can be read or written in 8-bit or 1-bit units.

These registers cannot be accessed in 16-bit units.

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

(i) V850ES/FE2

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

(ii) V850ES/FF2

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

(iii) V850ES/FG2, V850ES/FJ2

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

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

Caution To use the alternate function of P7n (ANIn), set PM7n to 1.

4.3.8 Port 8

Port 8 is a 2-bit port (P80, P81) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	2-bit I/O port (P80, P81)[Note]

Note In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available. The alternate function of the P80 pin in the μPD70F3237 is INTP14 only.

(1) Functions of port 8

- O The input/output data of the port can be specified in 1-bit units.
  Specified by port register 8 (P8)
- O The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 8 (PM8)
- O Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 8 (PMC8)
- O An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 8 (PU8)
- O The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 8 (INTF8) and external interrupt rising edge specification register 8 (INTR8)

Port 8 functions alternately as the following pins.

Table 4-21 Alternate-Function Pins of Port 8

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 8	P80	RXDA3/INTP14	I/O	–	L-1[Note]
	P81	TXDA3			C-1[Note]

Note In the μPD70F3237, the alternate functions of the P80 and P81 pins (RXDA3 and TXDA3) are not available. Moreover, the port type becomes P80: L-1 and P81: C-1.

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

(2) Registers

(a) Port register 8 (P8)

Port register 8 (P8) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF410H

(i) V850ES/FJ2

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

P8n	Control of output data (in output mode) (n = 0, 1)
0	Output 0.
1	Output 1.

(b) Port mode register 8 (PM8)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF430H

(i) V850ES/FJ2

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

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

(c) Port mode control register 8 (PMC8)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF450H

(i) V850ES/FJ2 (μPD70F3237)

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

(ii) V850ES/FJ2 (μPD70F3238,μPD70F3239)

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

PMC81	Specification of operation mode of P81 pin
0	I/O port
1	TXDA3 output

PMC80	Specification of operation mode of P80 pin
0	I/O port
1	RXDA3/INTP14 input[Note]

Note The μPD70F3237 does not have RXDA3.

The INTP14 pin of the μPD70F3238, μPD70F3239 functions alternately as the RXDA3 pin. To use this pin as the RXDA3 pin, invalidate the edge detection function of the alternate-function INTP14 pin (by clearing the INTF80 bit of the INTF8 register to 0 and the INTR80 bit of the INTR8 register to 0). To use this pin as the INTP14 pin, stop the reception operation of UARTA3 (by clearing the UA3RXE bit of the UA3CTL0 register to 0).

(d) Pull-up resistor option register 8 (PU8)

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC50H

(i) V850ES/FJ2

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

PU8n	Control of on-chip pull-up resistor connection (n = 0, 1)
0	Not connected
1	Connected

(e) External interrupt falling edge specification register 8 (INTF8)

This is an 8-bit register that specifies detection of the falling edge of the external interrupt pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF80 and INTR80 bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address FFFFFC10H

(i) V850ES/FJ2

	7	6	5	4	3	2	1	0
INTF8	0	0	0	0	0	0	0	INTF80

Remark Refer to **Table 4-22** or how to specify a valid edge.

(f) External interrupt rising edge specification register 8 (INTR8)

This is an 8-bit register that specifies detection of the rising edge of the external interrupt pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF80 and INTR80 bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address FFFFFC30H

(i) V850ES/FJ2

	7	6	5	4	3	2	1	0
INTR8	0	0	0	0	0	0	0	INTR80

Remark Refer to **Table 4-22** for how to specify a valid edge.

Table 4-22 Valid Edge Specification

INTF80 Bit	INTR80 Bit	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark Control of INTP14 pin

4.3.9 Port 9

Port 9 is a 9-bit or 16-bit port (P90 to P915) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	9-bit I/O port (P90, P91, P96 to P99, P913 to P915)
V850ES/FF2	
V850ES/FG2	16-bit I/O port (P90 to P915)[Note]
V850ES/FJ2	

Note In the V850ES/FG2, the alternate functions of the P910 to P912 pins (SIB2, SOB2, SCKB2) are not available.

(1) Functions of port 9

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 9 (P9)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 9 (PM9)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register 9 (PMC9)
- Control mode can be specified in 1-bit units.
  Specified by port function control register 9 (PFC9) and port function control expansion register 9 (PFCE9)
- An on-chip pull-up resistor can be connected in 1-bit units.
  Specified by pull-up resistor option register 9 (PU9)
- The valid edge of the external interrupt (alternate function) can be specified in 1-bit units.
  Specified by external interrupt falling edge specification register 9H (INTF9H) and external interrupt rising edge specification register 9H (INTR9H)

Port 9 functions alternately as the following pins.

Table 4-23 Alternate-Function Pins of Port 9

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 9	P90	KR6/TXDA1	I/O	–	U-12
	P91	KR7/RXDA1			U-7
	P92	TIQ11/TOQ11			U-11
	P93	TIQ12/TOQ12			U-11
	P94	TIQ13/TOQ13			U-11
	P95	TIQ10/TOQ10			U-11
	P96	TIP21/TOP21			U-9
	P97	SIB1/TIP20/TOP20			U-8
	P98	SOB1			G-3
	P99	$\overline{\text{SCKB1}}$			G-5
	P910	SIB2			G-4
	P911	SOB2			G-3
	P912	$\overline{\text{SCKB2}}$			G-5
	P913	INTP4/PCL			W-1
	P914	INTP5			N-2
	P915	INTP6			N-2

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

(2) Registers

(a) Port register 9 (P9)

Port register 9 (P9) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P9 register are used as the P9H register, and the lower 8 bits as the P9L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF412H, FFFFF413H

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
P9 (P9H[Note])	P915	P914	P913	0	0	0	P99	P98
	7	6	5	4	3	2	1	0
(P9L)	P97	P96	0	0	0	0	P91	P90

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
P9 (P9H[Note])	P915	P914	P913	P912	P911	P910	P99	P98
	7	6	5	4	3	2	1	0
(P9L)	P97	P96	P95	P94	P93	P92	P91	P90

P9n	Control of output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

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

(b) Port mode register 9 (PM9)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PM9 register are used as the PM9H register, and the lower 8 bits as the PM9L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: FFFFH R/W Address: FFFFF432H, FFFFF433H

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PM9 (PM9H[Note])	PM915	PM914	PM913	1	1	1	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	1	1	1	1	PM91	PM90

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PM9 (PM9H[Note])	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

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

Note To read or write bits 8 to 15 of the PM9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PM9H register.

(c) Port mode control register 9 (PMC9)

This is a 16-bit register that specifies the port mode or control mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PMC9 register are used as the PMC9H register, and the lower 8 bits as the PMC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC9 register when the PFC9n bit of the PFC9 register and the PFCE9n bit of the PFCE9 register are the default values (0), the output becomes undefined.
For this reason, first set the PFC9n bit of the PFC9 register and the PFCE9n bit of the PFCE9 register to 1, and then set the PMC9n bit to 1 to set the control mode.

(1/3)

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

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PMC9 (PMC9H[Note])	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98
	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

(ii) V850ES/FG2

	15	14	13	12	11	10	9	8
PMC9 (PMC9H[Note])	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98
	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

(iii) V850ES/FJ2

	15	14	13	12	11	10	9	8
PMC9 (PMC9H[Note])	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

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

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

Note To read or write bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMC9H register.

(2/3)

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

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

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

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

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

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

PMC97	Specification of operation mode of P97 pin
0	I/O port
1	SIB1/TIP20/TOP20 I/O

PMC96	Specification of operation mode of P96 pin
0	I/O port
1	TIP21/TOP21 I/O

PMC95	Specification of operation mode of P95 pin
0	I/O port
1	TIQ10/TOQ10 I/O

PMC94	Specification of operation mode of P94 pin
0	I/O port
1	TIQ13/TOQ13 I/O

(3/3)

PMC93	Specification of operation mode of P93 pin
0	I/O port
1	TIQ12/TOQ12 I/O

PMC92	Specification of operation mode of P92 pin
0	I/O port
1	TIQ11/TOQ11 I/O

PMC91	Specification of operation mode of P91 pin
0	I/O port
1	KR7/RXDA1 input

PMC90	Specification of operation mode of P90 pin
0	I/O port
1	KR6/TXDA1 I/O

(d) Port function control register 9 (PFC9)

This is a 16-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 16-bit units.

If the higher 8 bits of the PFC9 register are used as the PFC9H register, and the lower 8 bits as the PFC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

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

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PFC9 (PFC9H[Note])	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

(ii) V850ES/FG2

	15	14	13	12	11	10	9	8
PFC9 (PFC9H[Note])	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

(iii) V850ES/FJ2

	15	14	13	12	11	10	9	8
PFC9 (PFC9H[Note])	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

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

Remark For how to specify a control mode, refer to **4.3.10 (2) (f) Setting of control mode of P9 pin**.

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

This is a 16-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 16-bit units.

If the higher 8 bits of the PFC9 register are used as the PFC9H register, and the lower 8 bits as the PFC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

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

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H[Note])	0	0	PFCE913	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H[Note])	0	0	PFCE913	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

Note To read or write bits 8 to 15 of the PFCE9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFCE9H register.

Remark For how to specify a control mode, refer to **4.3.10 (2) (f) Setting of control mode of P9 pin**.

(f) Setting of control mode of P9 pin

Caution If the control mode is specified by using the PFC9 register when the PFC9n bit of the PFC9 register and PFCE9n bit of the PFCE9 register are the default values (0), the output becomes undefined.

For this reason, first set the PFC9n bit of the PFC9 register and the PFCE9n bit of the PFCE9 register, and then set the PMC9n bit to 1 to set the control mode.

PFC915	Specification of control mode of P915 pin
0	Setting prohibited
1	INTP6 input

PFC914	Specification of control mode of P914 pin
0	Setting prohibited
1	INTP5 input

PFCE913	PFC913	Specification of control mode of P913 pin
0	0	Setting prohibited
0	1	INTP4 input
1	0	PCL output
1	1	Setting prohibited

PFC912	Specification of control mode of P912 pin
0	Setting prohibited
1	$\overline{\text{SCKB2}}$ I/O

PFC911	Specification of control mode of P911 pin
0	Setting prohibited
1	SOB2 output

PFC910	Specification of control mode of P910 pin
0	Setting prohibited
1	SIB2 input

PFC99	Specification of control mode of P99 pin
0	Setting prohibited
1	$\overline{\text{SCKB1}}$ I/O

PFC98	Specification of control mode of P98 pin
0	Setting prohibited
1	SOB1 input

PFCE97	PFC97	Specification of control mode of P97 pin
0	0	Setting prohibited
0	1	SIB1 input
1	0	TIP20 input
1	1	TOP20 output

PFCE96	PFC96	Specification of control mode of P96 pin
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIP21 input
1	1	TOP21 output

PFCE95	PFC95	Specification of control mode of P95 pin
0	0	Setting prohibited
0	1	TIQ10 input
1	0	TOQ10 output
1	1	Setting prohibited

PFCE94	PFC94	Specification of control mode of P94 pin
0	0	Setting prohibited
0	1	TIQ13 input
1	0	TOQ13 output
1	1	Setting prohibited

PFCE93	PFC93	Specification of control mode of P93 pin
0	0	Setting prohibited
0	1	TIQ12 input
1	0	TOQ12 output
1	1	Setting prohibited

PFCE92	PFC92	Specification of control mode of P92 pin
0	0	Setting prohibited
0	1	TIQ11 input
1	0	TOQ11 output
1	1	Setting prohibited

PFCE91	PFC91	Specification of control mode of P91 pin
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDA1 input
1	1	Setting prohibited

★

PFCE90	PFC90	Specification of control mode of P90 pin
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDA1 output
1	1	Setting prohibited

Note KR7 and RXDA1 pins are using combined.

Invalidate the key return detection of KR7 pins when you use the pins as an RXDA1pin ("0" is set to the KRM7 bit of the KRM register). Moreover, it is recommended to set it to PFC91 bit = 1, PFCE91 bit =0 when using it as KR7 pin.

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

This is a 16-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 16-bit units.

If the higher 8 bits of the PU9 register are used as the PU9H register, and the lower 8 bits as the PU9L register, however, these registers can be read or written in 8-bit or 1-bit units.

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

(i) V850ES/FE2, V850ES/FF2

	15	14	13	12	11	10	9	8
PU9 (PU9H[Note])	PU915	PU914	PU913	0	0	0	PU99	PU98
	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	0	0	0	0	PU91	PU90

(ii) V850ES/FG2, V850ES/FJ2

	15	14	13	12	11	10	9	8
PU9 (PU9H[Note])	PU915	PU914	PU913	PU912	PU911	PU910	PU99	PU98
	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection (n = 0 to 15)
0	Not connected
1	Connected

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

(h) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF9n and INTR9n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

After reset: 00H R/W Address: FFFFFC13H

	7	6	5	4	3	2	1	0
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0

Remark Refer to **Table 4-24** or how to specify a valid edge.

(i) 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 pin. It can be read or written in 8-bit or 1-bit units.

Cautions 1. When the external interrupt function (alternate function) is switched to the port function, an edge may be detected. Set the port mode after clearing the INTF9n and INTR9n bits to 0.

2. An analog-delay-based noise eliminator is connected to the external interrupt input pin.

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 Refer to **Table 4-24** or how to specify a valid edge.

Table 4-24 Valid Edge Specification

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

Remark n = 13: Control of INTP4 pin
n = 14: Control of INTP5 pin
n = 15: Control of INTP6 pin

4.3.10 Port 12

Port 12 is an 8-bit port (P120 to P127) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	8-bit I/O port (P120 to P127)

(1) Functions of port 12

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register 12 (P12)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register 12 (PM12)

Port 12 functions alternately as the following pins.

Table 4-25 Alternate-Function Pins of Port 12

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port 12	P120	ANI16	I/O	–	A-1
	P121	ANI17			A-1
	P122	ANI18			A-1
	P123	ANI19			A-1
	P124	ANI20			A-1
	P125	ANI21			A-1
	P126	ANI22			A-1
	P127	ANI23			A-1

(2) Registers

(a) Port register 12 (P12)

Port register 12 (P12) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF418H

(i) V850ES/FJ2

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

P12n	Control of output data (in output mode) (n = 0 to 7)
0	Output 0.
1	Output 1.

(b) Port mode register 12 (PM12)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF438H

(i) V850ES/FJ2

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

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

Caution To use the alternate function of P12n (ANIn), set PM12n to 1.

4.3.11 Port CD

Port CD is a 4-bit port (PCD0 to PCD3) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	4-bit I/O port (PCD0 to PCD3)

(1) Functions of port CD

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register CD (PCD)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register CD (PMCD)

Port CD functions alternately as the following pins.

Table 4-26. Alternate-Function Pins of Port CD

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port CD	PCD0	–	I/O	–	B-1
	PCD1	–			B-1
	PCD2	–			B-1
	PCD3	–			B-1

(2) Registers

(a) Port register CD (PCD)

Port register CD (PCD) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00EH

(i) V850ES/FJ2

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

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

(b) Port mode register CD (PMCD)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF02EH

(i) V850ES/FJ2

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

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

4.3.12 Port CM

Port CM is a 2-bit, 4-bit, or 6-bit port (PCM0 to PCM5) for which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	2-bit I/O port (PCM0, PCM1)Note 1
V850ES/FF2	4-bit I/O port (PCM0 to PCM3)Note 2
V850ES/FG2	
V850ES/FJ2	4-bit I/O port (PCD0 to PCD3)

Notes 1. In the V850ES/FE2, the alternate function of the PCM0 pin (WAIT) is not available.

2. In the V850ES/FF2 and V850ES/FG2, the alternate functions of the PCM0, PCM2, and PCM3 pins. (WAIT, HLDAK, HLDRQ) are not available.

(1) Functions of port CM

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register CM (PCM)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register CM (PMCM)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register CM (PMCCM)

Port CM functions alternately as the following pins.

Table 4-27 Alternate-Function Pins of Port CM

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port CM	PCM0	$\overline{\text{WAIT}}$	I/O	–	D-1
	PCM1	CLKOUT			D-2
	PCM2	$\overline{\text{HLDAK}}$			D-2
	PCM3	$\overline{\text{CLDRQ}}$			D-1
	PCM4	–			B-1
	PCM5	–			B-1

(2) Registers

(a) Port register CM (PCM)

Port register CM (PCM) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00CH

(i) V850ES/FE2

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

(ii) V850ES/FF2, V850ES/FG2

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

(iii) V850ES/FJ2

	7	6	5	4	3	2	1	0
PCM	0	0	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0

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

(b) Port mode register CM (PMCM)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF02CH

(i) V850ES/FE2

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	1	1	PMCM1	PMCM0

(ii) V850ES/FF2, V850ES/FG2

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

(iii) V850ES/FJ2

	7	6	5	4	3	2	1	0
PMCM	1	1	PMCM5	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0

PMCMn	Control of I/O mode (n = 0 to 5)
0	Output mode
1	Input mode

(c) Port mode control register CM (PMCCM)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF04CH

(i) V850ES/FE2, V850ES/FF2, V850ES/FG2

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	0

(ii) V850ES/FJ2

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

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

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

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

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

4.3.13 Port CS

Port CS is a 2-bit or 8-bit port (PCS0 to PCS7) or which I/O settings can be controlled in 1-bit units.
The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	2-bit I/O port (PCS0, PCS1)Note
V850ES/FG2	
V850ES/FJ2	8-bit I/O port (PCS0 to PCS7)

Note In the V850ES/FF2 and V850ES/FG2, the alternate functions of the PCS0 and PCS1 pins (CS0, CS1) are not available.

(1) Functions of port CS

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register CS (PCS)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register CS (PMCS)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register CS (PMCCS)

Port CS functions alternately as the following pins.

Table 4-28 Alternate-Function Pins of Port CS

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port CS	PCS0	$\overline{CS0}$	I/O	–	D-2
	PCS1	$\overline{CS1}$			D-2
	PCS2	$\overline{CS2}$			D-2
	PCS3	$\overline{CS3}$			D-2
	PCS4	–			B-1
	PCS5	–			B-1
	PCS6	–			B-1
	PCS7	–			B-1

(2) Registers

(a) Port register CS (PCS)

Port register CS (PCS) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF008H

(i) V850ES/FF2, V850ES/FG2

	7	6	5	4	3	2	1	0
PCS	0	0	0	0	0	0	PCS1	PCS0

(ii) V850ES/FJ2

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

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

(b) Port mode register CS (PMCS)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF028H

(i) V850ES/FF2, V850ES/FG2

	7	6	5	4	3	2	1	0
PMCS	1	1	1	1	1	1	PMCS1	PMCS0

(ii) V850ES/FJ2

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

PMCSn	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(c) Port mode control register CS (PMCCS)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF048H

(i) V850ES/FJ2

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	PMCCS3	PMCCS2	PMCCS1	PMCCS0

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

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

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

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

4.3.14 Port CT

Port CT is a 4-bit or 8-bit port (PCT0 to PCT7) or which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	-
V850ES/FF2	4-bit I/O port (PCT0, PCT1, PCT4, PCT6)[Note]
V850ES/FG2	
V850ES/FJ2	8-bit I/O port (PCT0 to PCT7)

Note In the V850ES/FF2 and V850ES/FG2, the alternate functions of the PCT0, PCT1, PCT4, and PCT6 pins (WR0, WR1, RD, ASTB) are not available.

(1) Functions of port CT

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register CT (PCT)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register CT (PMCT)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register CT (PMCCT)

Port CT functions alternately as the following pins.

Table 4-29 Alternate-Function Pins of Port CT

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port CT	PCT0	$\overline{WR0}$	I/O	–	D-2
	PCT1	$\overline{WR1}$			D-2
	PCT2	–			B-1
	PCT3	–			B-1
	PCT4	$\overline{RD}$			D-2
	PCT5	–			B-1
	PCT6	ASTB			D-2
	PCT7	–			B-1

(2) Registers

(a) Port register CT (PCT)

Port register CT (PCT) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00AH

(i) V850ES/FF2, V850ES/FG2

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

(ii) V850ES/FJ2

	7	6	5	4	3	2	1	0
PCT	PCT7	PCT6	PCT5	PCT4	PCT3	PCT2	PCT1	PCT0

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

(b) Port mode register CT (PMCT)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF02AH

(i) V850ES/FF2, V850ES/FG2

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

(ii) V850ES/FJ2

	7	6	5	4	3	2	1	0
PMCT	PMCT7	PMCT6	PMCT5	PMCT4	PMCT3	PMCT2	PMCT1	PMCT0

PMCTn	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(c) Port mode control register CT (PMCCT)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF04AH

(i) V850ES/FJ2

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

PMCCT6	Specification of operation mode of PCT3 pin
0	I/O port
1	ASTB output

PMCCT4	Specification of operation mode of PCT2 pin
0	I/O port
1	$\overline{RD}$ output

PMCCT1	Specification of operation mode of PCT1 pin
0	I/O port
1	$\overline{WR1}$ output

PMCCT0	Specification of operation mode of PCT0 pin
0	I/O port
1	$\overline{WR0}$ output

4.3.15 Port DL

Port DL is an 8-bit, 12-bit, 14-bit, or 16-bit port (PDL0 to PDL15) or which I/O settings can be controlled in 1-bit units.

The number of I/O port pins differs depending on the product.

Product	Number of I/O Port Pins
V850ES/FE2	8-bit I/O port (PDL0 to PDL7)Note
V850ES/FF2	12-bit I/O port (PDL0 to PDL11)Note
V850ES/FG2	14-bit I/O port (PDL0 to PDL13)Note
V850ES/FJ2	16-bit I/O port (PDL0 to PDL15)

Note In the V850ES/FE2, V850ES/FF2, and V850ES/FG2, the alternate function of the PDLn pin (ADn) is not available. The alternate function of the PDL5 pin in the V850ES/FE2, V850ES/FF2, and V850ES/FG2 is FLMD1 only.

(1) Function of port DL

- The input/output data of the port can be specified in 1-bit units.
  Specified by port register DL (PDL)
- The input/output mode of the port can be specified in 1-bit units.
  Specified by port mode register DL (PMDL)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
  Specified by port mode control register DL (PMCDL)

Port DL functions alternately as the following pins.

Table 4-30 Alternate-Function Pins of Port DL

Pin Name		Alternate-Function Pin Name	I/O	Remark	Block Type
Port DL	PDL0	AD0	I/O	–	D-3
	PDL1	AD1			D-3
	PDL2	AD2			D-3
	PDL3	AD3			D-3
	PDL4	AD4			D-3
	PDL5	AD5/FLMD1[Note]			D-3
	PDL6	AD6			D-3
	PDL7	AD7			D-3
	PDL8	AD8			D-3
	PDL9	AD9			D-3
	PDL10	AD10			D-3
	PDL11	AD11			D-3
	PDL12	AD12			D-3
	PDL13	AD13			D-3
	PDL14	AD14			D-3
	PDL15	AD15			D-3

Note Because the FLMD1 pin is used in the flash programming mode, it does not have to be manipulated by using a port control register. For details, refer to **CHAPTER 25 FLASH MEMORY**.

(2) Registers

(a) Port register DL (PDL)

Port register DL (PDL) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

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

After reset: Undefined R/W Address: FFFFF004H, FFFFF005H

(i) V850ES/FE2

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

(ii) V850ES/FF2

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	0	0	0	0	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

(iii) V850ES/FG2

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	0	0	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

(iv) V850ES/FJ2

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

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

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

(b) Port mode register DL (PMDL)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

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

After reset: FFFFH R/W Address: FFFFF024H, FFFFF025H

(i) V850ES/FE2

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

(ii) V850ES/FF2

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note])	1	1	1	1	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

(iii) V850ES/FG2

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note])	1	1	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

(iv) V850ES/FJ2

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note])	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	Control of I/O mode (n = 0 to 15)
0	Output mode
1	Input mode

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

(c) Port mode control register DL (PMCDL)

This is a 16-bit register that specifies the port mode or control mode. It can be read or written in 16-bit units.

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

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

(i) V850ES/FJ2

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH[Note])	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8
	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	Specification of operation mode of PDL15 pin (n = 0 to 15)
0	I/O port
1	ADn I/O (address/data bus I/O)

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

4.3.16 Port pins that function alternately as on-chip debug function

The pins shown in Table 4-31 function alternately as on-chip debug pins. After an external reset, these pins are initialized as on-chip debug pins ($\overline{\text{DRST}}$, DDI, DDO, DCK, and DMS).

Table 4-31 On-Chip Debug Pins

Pin Name	Alternate Function Pin
P05	INTP2/$\overline{\text{DRST}}$
P52	KR2/TIQ03/TOQ03/DDI
P53	KR3/TIQ00/TOQ00/DDO
P54	KR4/DCK
P55	KR5/DMS

To use these pins as port pins, not as on-chip debug pins, the following actions must be taken after an external reset.

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

When the on-chip debug function is not used, inputting a high level to the $\overline{\text{DRST}}$ pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.

When a high level is not input to the P05/INTP2/$\overline{\text{DRST}}$ pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM0 bit of the OCDM register.

Because a pull-down resistor (30 kΩ TYP) is connected to the buffer of the P05/INTP2/$\overline{\text{DRST}}$ pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.

For details, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

4.3.17 Register settings to use port pins as alternate-function pins

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (1/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P00	TIP31	Input	Setting not required	PMC00 = 1	PFC00 = 0	–	
	TOP31	Output	Setting not required	PMC00 = 1	PFC00 = 1	–	
P01	TIP30	Input	Setting not required	PMC01 = 1	PFC01 = 0	–	
	TOP30	Output	Setting not required	PMC01 = 1	PFC01 = 1	–	
P02	NMI	Input	Setting not required	PMC02 = 1	–	–	
P03	INTP0	Input	Setting not required	PMC03 = 1	PFC03 = 0	–	INTx03 (INTx0)
	ADTRG	Output	Setting not required	PMC03 = 1	PFC03 = 1	–	
P04	INTP1	Input	Setting not required	PMC04 = 1	–	–	INTx04 (INTx0)
P05[Note]	INTP2	Input	Setting not required	PMC05 = 1	–	–	INTx05 (INTx0)
	$\overline{\text{DRST}}$	Input	Setting not required	Setting not required	–	–	OCDM0 (OCDM) = 1
P06	INTP3	Input	Setting not required	PMC06 = 1	–	–	INTx06 (INTx0)
P10	INTP9	Input	Setting not required	PMC10 = 1	–	–	INTx10 (INTx1)
P11	INTP10	Input	Setting not required	PMC11 = 1	–	–	INTx11 (INTx1)

Note After an external reset, the P05/INTP2/$\overline{\text{DRST}}$ pin is initialized as an on-chip debug pin ($\overline{\text{DRST}}$). To not use the P05/INTP2/$\overline{\text{DRST}}$ pin as an on-chip debug pin, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

2. INTxn = INTFn, INTRn

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (2/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P30	TXDA0	Output	Setting not required	PMC30 = 1	–	–	
P31	RXDA0	Input	Setting not required	PMC31 = 1	–	–	**Note 1**
	INTP7	Input	Setting not required	PMC31 = 1	–	–	**Note 1**, INTx31 (INTx3)
P32	ASCKA0	Input	Setting not required	PMC32 = 1	PFC32 = 0	PFCE32 = 0	
	TOP01	Output	Setting not required	PMC32 = 1	PFC32 = 1	PFCE32 = 0	
	TIP00	Input	Setting not required	PMC32 = 1	PFC32 = 0	PFCE32 = 1	
	TOP00	Output	Setting not required	PMC32 = 1	PFC32 = 1	PFCE32 = 1	
P33	TIP01	Input	Setting not required	PMC33 = 1	PFC33 = 0	PFCE33 = 0	
	TOP01	Output	Setting not required	PMC33 = 1	PFC33 = 1	PFCE33 = 0	
	CTXD0	Output	Setting not required	PMC33 = 1	PFC33 = 0	PFCE33 = 1	
P34	TIP10	Input	Setting not required	PMC34 = 1	PFC34 = 0	PFCE34 = 0	
	TOP10	Output	Setting not required	PMC34 = 1	PFC34 = 1	PFCE34 = 0	
	CRXD0	Input	Setting not required	PMC34 = 1	PFC34 = 0	PFCE34 = 1	
P35	TIP11	Input	Setting not required	PMC35 = 1	PFC35 = 0	–	
	TOP11	Output	Setting not required	PMC35 = 1	PFC35 = 1	–	
P36	CTXD1	Output	Setting not required	PMC36 = 1	–	–	
P37	CRXD1	Input	Setting not required	PMC37 = 1	–	–	
P38	TXDA2	Output	Setting not required	PMC38 = 1	–	–	
P39	RXDA2	Input	Setting not required	PMC39 = 1	–	–	**Note 2**
	INTP8	Input	Setting not required	PMC39 = 1	–	–	**Note 2**, INTx39 (INTx3)
P40	SIB0	Input	Setting not required	PMC40 = 1	–	–	
P41	SOB0	Output	Setting not required	PMC41 = 1	–	–	
P42	$\overline{\text{SCKB0}}$	I/O	Setting not required	PMC42 = 1	–	–	

Notes 1. The INTP7 pin functions alternately as the RXDA0 pin. To use this pin as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by clearing the INTF31 bit of the INTF3 register to 0 and the INTR31 bit of the INTR3 register to 0). To use this pin as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0RXE bit of the UA0CTL0 register to 0).

2. The INTP8 pin functions alternately as the RXDA2 pin. To use this pin as the RXDA2 pin, invalidate the edge detection function of the alternate-function INTP8 pin (by clearing the INTF39 bit of the INTF3 register to 0 and the INTR39 bit of the INTR3 register to 0). To use this pin as the INTP8 pin, stop the reception operation of UARTA2 (by clearing the UA2RXE bit of the UA2CTL0 register to 0).

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

2. INTxn = INTFn, INTRn

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (3/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P50	KR0	Input	Setting not required	PMC50 = 1	PFC50 = 1	PFCE50 = 0	**Note 1**
	TIQ01	Input	Setting not required	PMC50 = 1	PFC50 = 1	PFCE50 = 0	**Note 1**
	TOQ01	Output	Setting not required	PMC50 = 1	PFC50 = 0	PFCE50 = 1	
P51	KR1	Input	Setting not required	PMC51 = 1	PFC51 = 1	PFCE54 = 0	**Note 1**
	TIQ02	Input	Setting not required	PMC51 = 1	PFC51 = 1	PFCE51 = 0	**Note 1**
	TOQ02	Output	Setting not required	PMC51 = 1	PFC51 = 0	PFCE51 = 1	
P52	KR2	Input	Setting not required	PMC52 = 1	PFC52 = 1	PFCE52 = 0	**Note 1**
	TIQ03	Input	Setting not required	PMC52 = 1	PFC52 = 1	PFCE52 = 0	**Note 1**
	TOQ03	Output	Setting not required	PMC52 = 1	PFC52 = 0	PFCE52 = 1	
	DDI[Note 2]	Input	Setting not required	Setting not required	Setting not required	Setting not required	OCDM0 (OCDM) = 1
P53	KR3	Input	Setting not required	PMC53 = 1	PFC53 = 1	PFCE53 = 0	**Note 1**
	TIQ00	Input	Setting not required	PMC53 = 1	PFC53 = 1	PFCE53 = 0	**Note 1**
	TOQ00	Output	Setting not required	PMC53 = 1	PFC53 = 0	PFCE53 = 1	
	DDO[Note 2]	Output	Setting not required	Setting not required	Setting not required	Setting not required	OCDM0 (OCDM) = 1
P54	KR4	Input	Setting not required	PMC54 = 1	PFC54 = 1	–	
	DCK[Note 2]	Output	Setting not required	Setting not required	Setting not required	–	OCDM0 (OCDM) = 1
P55	KR5	Input	Setting not required	PMC55 = 1	PFC55 = 1	–	
	DMS[Note 2]	Output	Setting not required	Setting not required	Setting not required	–	OCDM0 (OCDM) = 1

Notes 1. The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRMn bit of the KRM register to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin (n = 0 to 3, m = 0 to 3).

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

2. The DDI, DDO, DCK, and DMS pins are on-chip debug pins. To not use these pins as on-chip debug pins after an external reset, refer to **CHAPTER 27 ON-CHIP DEBUG FUNCTION (ON-CHIP DEBUG UNIT)**.

Caution If the control mode is specified by using the PMC5 register when the PFC5n bit of the PFC5 register and the PFCE5n bit of the PFCE5 register are the default values (0), the output becomes undefined. For this reason, first set the PFC5n bit of the PFC5 register and the PFCE5n bit of the PFCE5 register, and then set the PMC5n bit to 1 to set the control mode.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

2. INTxn = INTFn, INTRn

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (4/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P60	INTP11	Input	Setting not required	PMC60 = 1	PFC60 = 1	–	INTx60 (INTx6L)
P61	INTP12	Input	Setting not required	PMC61 = 1	PFC61 = 1	–	INTx61 (INTx6L)
P62	INTP13	Input	Setting not required	PMC62 = 1	PFC62 = 1	–	INTx62 (INTx6L)
P65	CTXD2	Output	Setting not required	PMC65 = 1	PFC65 = 1	–	
P66	CRXD2	Input	Setting not required	PMC66 = 1	PFC66 = 1	–	
P67	CTXD3	Output	Setting not required	PMC67 = 1	PFC67 = 1	–	
P68	CRXD3	Input	Setting not required	PMC68 = 1	PFC68 = 1	–	
P610	TIQ20	Input	Setting not required	PMC610 = 1	PFC610 = 0	–	
	TOQ20	Output	Setting not required	PMC610 = 1	PFC610 = 1	–	
P611	TIQ21	Input	Setting not required	PMC611 = 1	PFC611 = 0	–	
	TOQ21	Output	Setting not required	PMC611 = 1	PFC611 = 1	–	
P612	TIQ22	Input	Setting not required	PMC612 = 1	PFC612 = 0	–	
	TOQ22	Output	Setting not required	PMC612 = 1	PFC612 = 1	–	
P613	TIQ23	Input	Setting not required	PMC613 = 1	PFC613 = 0	–	
	TOQ23	Output	Setting not required	PMC613 = 1	PFC613 = 1	–	
P70	ANI0	Input	PM70 = 1 Note	–	–	–	
P71	ANI1	Input	PM71 = 1 Note	–	–	–	
P72	ANI2	Input	PM72 = 1 Note	–	–	–	
P73	ANI3	Input	PM73 = 1 Note	–	–	–	
P74	ANI4	Input	PM74 = 1 Note	–	–	–	
P75	ANI5	Input	PM75 = 1 Note	–	–	–	
P76	ANI6	Input	PM76 = 1 Note	–	–	–	
P77	ANI7	Input	PM77 = 1 Note	–	–	–	
P78	ANI8	Input	PM78 = 1 Note	–	–	—	
P79	ANI9	Input	PM79 = 1 Note	–	–	–	
P710	ANI10	Input	PM710 = 1 Note	–	–	–	
P711	ANI11	Input	PM711 = 1 Note	–	–	–	
P712	ANI12	Input	PM712 = 1 Note	–	–	–	
P713	ANI13	Input	PM713 = 1 Note	–	–	–	
P714	ANI14	Input	PM714 = 1 Note	–	–	–	
P715	ANI15	Input	PM715 = 1 Note	–	–	–	

Note Set PM7n to 1 to use the alternate function of P7n (ANIn).

Caution If the control mode is specified by using the PMC6 register when the PFC6n bit (n = 0 to 8) of the PFC6 register is the default value (0), the output becomes undefined.
For this reason, first set the PFC6n bit of the PFC6 register and then set the PMC6n bit to 1 to set the control mode.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.
2. INTxn = INTFn, INTRn

Table 4-32 Register Settings to Use Port Pin as Alternate-Function Pins (5/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P80	RXDA3	Input	Setting not required	PMC80 = 1	–	–	**Note 2**
	INTP14	Input	Setting not required	PMC80 = 1	–	–	**Note,** INTx80 (INTx8)
P81	TXDA3	Output	Setting not required	PMC81 = 1	–	–	
P90	KR6	Input	Setting not required	PMC90 = 1	PFC90 = 1	PFCE90 = 0	
	TXDA1	Output	Setting not required	PMC90 = 1	PFC90 = 0	PFCE90 = 1	
P91	KR7 Note 1	Input	Setting not required	PMC91 = 1	PFC91 = 1	PFCE91 = 0	
					PFC91 = 0	PFCE91 = 1	
	RXDA1	Input	Setting not required	PMC91 = 1	PFC91 = 0	PFCE91 = 1	
P92	TIQ11	Input	Setting not required	PMC92 = 1	PFC92 = 1	PFCE92 = 0	
	TOQ11	Output	Setting not required	PMC92 = 1	PFC92 = 0	PFCE92 = 1	
P93	TIQ12	Input	Setting not required	PMC93 = 1	PFC93 = 1	PFCE93 = 0	
	TOQ12	Output	Setting not required	PMC93 = 1	PFC93 = 0	PFCE93 = 1	
P94	TIQ13	Input	Setting not required	PMC94 = 1	PFC94 = 1	PFCE94 = 0	
	TOQ13	Output	Setting not required	PMC94 = 1	PFC94 = 0	PFCE94 = 1	
P95	TIQ10	Input	Setting not required	PMC95 = 1	PFC95 = 1	PFCE95 = 0	
	TOQ10	Output	Setting not required	PMC95 = 1	PFC95 = 0	PFCE95 = 1	
P96	TIP21	Input	Setting not required	PMC96 = 1	PFC96 = 0	PFCE96 = 1	
	TOP21	Output	Setting not required	PMC96 = 1	PFC96 = 1	PFCE96 = 1	
P97	SIB1	Input	Setting not required	PMC97 = 1	PFC97 = 1	PFCE97 = 0	
	TIP20	Input	Setting not required	PMC97 = 1	PFC97 = 0	PFCE97 = 1	
	TOP20	Output	Setting not required	PMC97 = 1	PFC97 = 1	PFCE97 = 1	
P98	SOB1	Output	Setting not required	PMC98 = 1	PFC98 = 1	–	
P99	$\overline{\text{SCKB1}}$	I/O	Setting not required	PMC99 = 1	PFC99 = 1	–	
P910	SIB2	Input	Setting not required	PMC910 = 1	PFC910 = 1	–	
P911	SOB2	Output	Setting not required	PMC911 = 1	PFC911 = 1	–	
P912	$\overline{\text{SCKB2}}$	I/O	Setting not required	PMC912 = 1	PFC912 = 1	–	
P913	INTP4	Input	Setting not required	PMC913 = 1	PFC913 = 1	PFCE913 = 0	INTx913 (INTx9H)
	PCL	Output	Setting not required	PMC913 = 1	PFC913 = 0	PFCE913 = 1	
P914	INTP5	Input	Setting not required	PMC914 = 1	PFC914 = 1	–	INTx914 (INTx9H)
P915	INTP6	Input	Setting not required	PMC915 = 1	PFC915 = 1	–	INTx915 (INTx9H)

Note 1. The KR7 pin and the RXDA1 pin are using combined. Invalidate the key return detection of the KR7 pin when you use the terminal as the RXDA1 pin ("0" is set to the KRM7 bit of the KRM register.). Moreover, it is recommended to set it to PFC91 bit = 1, PFCE91 bit =0 when using it as the KR7 pin.

2 The INTP14 pin functions alternately as the RXDA3 pin. To use this pin as the RXDA3 pin, invalidate the edge detection function of the alternate-function INTP14 pin (by clearing the INTF80 bit of the INTF8 register to 0 and the INTR80 bit of the INTR8 register to 0). To use this pin as the INTP14 pin, stop the reception operation of UARTA3 (by clearing the UA3RXE bit of the UA3CTL0 register to 0).

Caution If the control mode is specified by using the PMC9 register when the PFC9n bit of the PFC9 register and the PFCE9n bit of the PFCE9 register are the default values (0), the output becomes undefined. For this reason, first set the PFC9n bit of the PFC9 register and the PFCE9n bit of the PFCE9 register, and then set the PMC9n bit to 1 to set the control mode.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

2. INTxn = INTFn, INTRn

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (6/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
P120	ANI16	Input	PM120 = 1[Note]	–	–	–	
P121	ANI17	Input	PM121 = 1[Note]	–	–	–	
P122	ANI18	Input	PM122 = 1[Note]	–	–	–	
P123	ANI19	Input	PM123 = 1[Note]	–	–	–	
P124	ANI20	Input	PM124 = 1[Note]	–	–	–	
P125	ANI21	Input	PM125 = 1[Note]	–	–	–	
P126	ANI22	Input	PM126 = 1[Note]	–	–	–	
P127	ANI23	Input	PM127 = 1[Note]	–	–	–	
PCM0	$\overline{\text{WAIT}}$	Input	Setting not required	PMCCM0 = 1	–	–	
PCM1	CLKOUT	Output	Setting not required	PMCCM1 = 1	–	–	
PCM2	$\overline{\text{HLDAK}}$	Output	Setting not required	PMCCM2 = 1	–	–	
PCM3	$\overline{\text{HLDRQ}}$	Input	Setting not required	PMCCM3 = 1	–	–	
PCS0	$\overline{\text{CS0}}$	Output	Setting not required	PMCCS0 = 1	–	–	
PCS1	$\overline{\text{CS1}}$	Output	Setting not required	PMCCS1 = 1	–	–	
PCS2	$\overline{\text{CS2}}$	Output	Setting not required	PMCCS2 = 1	–	–	
PCS3	$\overline{\text{CS3}}$	Output	Setting not required	PMCCS3 = 1	–	–	
PCT0	$\overline{\text{WR0}}$	Output	Setting not required	PMCCT0 = 1	–	–	
PCT1	$\overline{\text{WR1}}$	Output	Setting not required	PMCCT1 = 1	–	–	
PCT4	$\overline{\text{RD}}$	Output	Setting not required	PMCCT4 = 1	–	–	
PCT6	ASTB	Output	Setting not required	PMCCT6 = 1	–	–	

Note Set PM12n to 1 to use the alternate function of P12n (ANIn).

Remark The port register (Pn) does not have to be set when the alternate function is used.

Table 4-32 Register Settings to Use Port Pins as Alternate-Function Pins (7/7)

Pin Name	Alternate-Function Pin		PMn Register	PMCn Register	PFCm Register	PFCEm Register	Other Bits (Register)
	Name	I/O					
PDL0	AD0	I/O	Setting not required	PMCDL0 = 1	–	–	
PDL1	AD1	I/O	Setting not required	PMCDL1 = 1	–	–	
PDL2	AD2	I/O	Setting not required	PMCDL2 = 1	–	–	
PDL3	AD3	I/O	Setting not required	PMCDL3 = 1	–	–	
PDL4	AD4	I/O	Setting not required	PMCDL4 = 1	–	–	
PDL5	AD5	I/O	Setting not required	PMCDL5 = 1	–	–	
	FLMD1	Input	Setting not required	Setting not required	–	–	**Note**
PDL6	AD6	I/O	Setting not required	PMCDL6 = 1	–	–	
PDL7	AD7	I/O	Setting not required	PMCDL7 = 1	–	–	
PDL8	AD8	I/O	Setting not required	PMCDL8 = 1	–	–	
PDL9	AD9	I/O	Setting not required	PMCDL9 = 1	–	–	
PDL10	AD10	I/O	Setting not required	PMCDL10 = 1	–	–	
PDL11	AD11	I/O	Setting not required	PMCDL11 = 1	–	–	
PDL12	AD12	I/O	Setting not required	PMCDL12 = 1	–	–	
PDL13	AD13	I/O	Setting not required	PMCDL13 = 1	–	–	
PDL14	AD14	I/O	Setting not required	PMCDL14 = 1	–	–	
PDL15	AD15	I/O	Setting not required	PMCDL15 = 1	–	–	

Note The FLMD1 pin does not have to be manipulated by using a port control register because it is used in the flash programming mode. For details, refer to **CHAPTER 25 FLASH MEMORY**.

Remark The port register (Pn) does not have to be set when the alternate function is used.

4.3.18 Operation of Port Function

The operation of a port differs depending on setting of the input or output mode, as follows.

(1) Writing to I/O port

(a) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch are output from the pin. Once data has been written to the output latch, it is retained until new data is written to the output latch.

(b) In input mode

A value can be written to the output latch by using a transfer instruction. Because the output buffer is off, however, the status of the pin remains unchanged.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

Caution Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. If a port has a mixture of input and output pins, therefore, the contents of the output latch of a pin set in the input mode become undefined, even if the pin is not subject to manipulation.

(2) Reading from I/O port

(a) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch are not changed.

(b) In input mode

The status of the pin can be read by using a transfer instruction. The contents of the output latch are not changed.

(3) Operation of I/O port

(a) In output mode

An operation is performed on the contents of the output latch and the result is written to the output latch. The contents of the output latch are output from the pin.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

(b) In input mode

The contents of the output latch become undefined. Because the output buffer is off, however, the status of the pin remains unchanged.

Caution Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. If a port has a mixture of input and output pins, therefore, the contents of the output latch of a pin set in the input mode become undefined, even if the pin is not subject to manipulation.

4.4 Cautions

4.4.1 Cautions on setting port pins

(1) The general-purpose port function and several peripheral function I/O pin share a pin. To switch between the general-purpose port (port mode) and the peripheral function I/O pin (alternate-function mode), set by the PMCn register. In regards to this register setting sequence, note with caution the following.

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

To switch from the port mode to alternate-function mode in the following order.

<1> Set the PFn register[Note]: N-ch open-drain setting
<2> Set the PFCn and PFCEn registers: Alternate-function selection
<3> Set the corresponding bit of the PMCn register to 1: Switch to alternate-function mode

If the PMCn register is set first, note with caution that, at that moment or depending on the change of the pin states in accordance with the setting of the PFn, PFCn, and PFCEn registers, unexpected operations may occur.

Note No-ch open-drain output pin only

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

- **Pn register read: Read the port output latch value (when PMn.PMnm bit = 0), or read the pin states (PMn.PMnm bit = 1).**
- **. Pn register write: Write to the port output latch**

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

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

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

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

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

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the value of the output latch of an input port that is not subject to manipulation may be written in addition to the targeted bit. Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

4.4.3 Cautions on on-chip debug pins

The $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins are on-chip debug pins (these pins are available only in the flash memory versions).

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

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

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

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

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

Handle the P05 pin with the utmost care.

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

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

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

CHAPTER 5 BUS CONTROL FUNCTION

The V850ES/FJ2 is provided with an external bus interface function by which memories such as ROM and RAM, or I/O, can be externally connected. V850ES/FE2, V850ES/FF2, V850ES/FG2 do not embed this interface.

5.1 Features

- ○ Output 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 per memory block
  - External wait function using WAIT pin
- ○ Idle state insertion function
- ○ Bus hold function
- ○ External devices can be connected using alternate-function port pins
- ○ Fixed to little-endian format
- ○ Misaligned access possible
- ○ Chip select function (4 spaces)

5.2 Bus Control Pins

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

Table 5-1. Bus Control Pins (Multiplexed Bus)

Bus Control Pin	Alternate-Function Pin	I/O	Function
AD0 to AD15	PDL0 to PDL15	I/O	Address/data bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	PCS0 to PCS3	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 on-chip peripheral I/O is accessed

The following list shows pin statuses when internal ROM, internal RAM, or on-chip peripheral I/O is accessed.

Access Destination	Address Bus	Data Bus	Control Signal
Internal ROM	Undefined	Hi-Z	Inactive
Internal RAM	Undefined	Hi-Z	Inactive
On-chip peripheral I/O	**Note**	Hi-Z	Inactive

Note When an on-chip peripheral I/O is accessed, the address of the on-chip peripheral I/O being accessed is output via the address bus.

5.2.2 Pin status in each operation mode

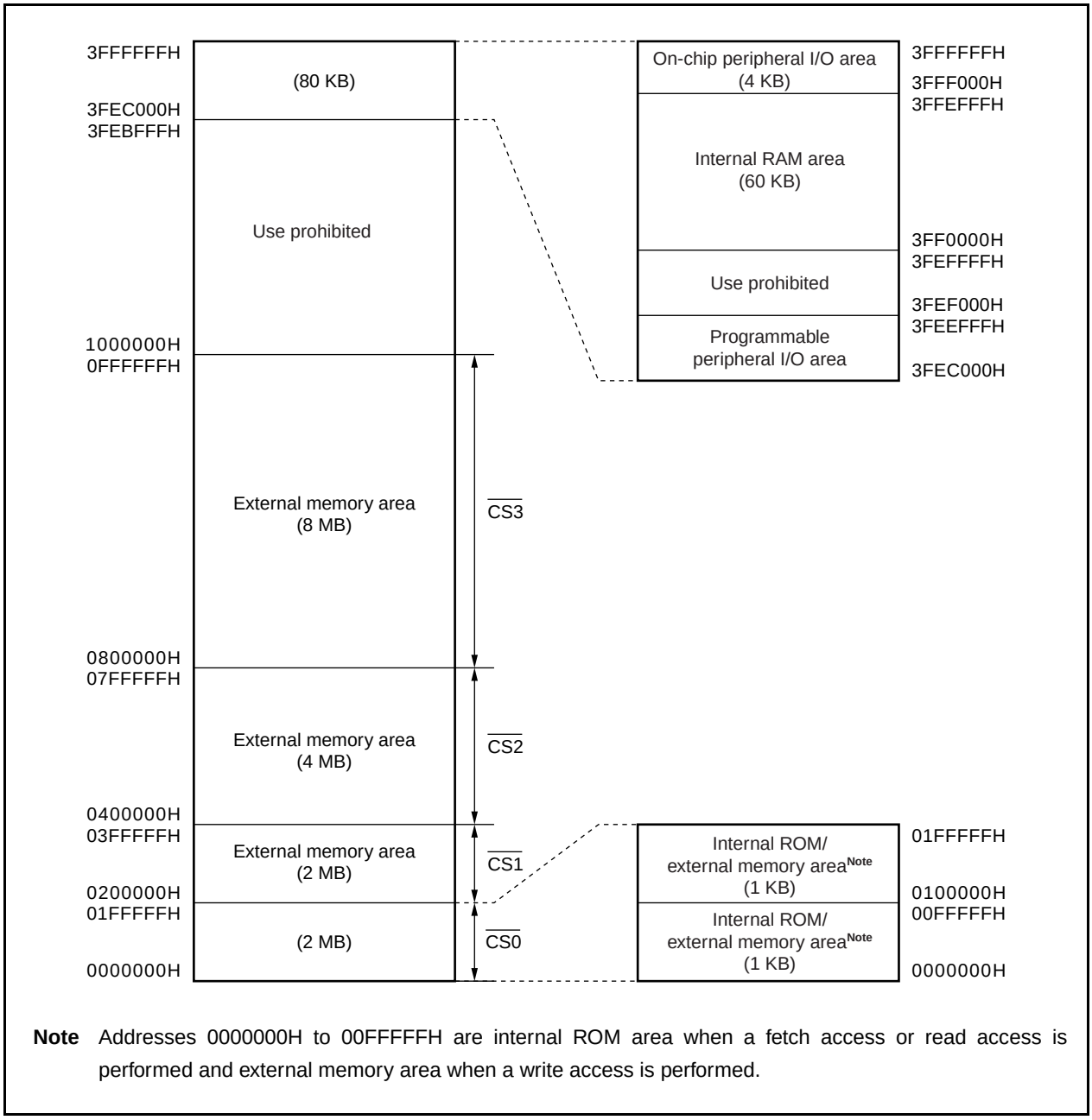
For the pin status of the V850ES/FJ2 in each operation mode, see **2.2 Pin Status**.

5.3 Memory Block Function

5.3.1 Memory space

The 64 MB memory space is divided into memory blocks of (lower) 2 MB, 2 MB, 4 MB, and 8 MB. The programmable wait function and bus cycle operation mode for each of these blocks can be independently controlled in one-block units.

Figure 5-1. Data Memory Map



Note Addresses 0000000H to 00FFFFFH are internal ROM area when a fetch access or read access is performed and external memory area when a write access is performed.

5.3.2 Chip select function

Of the 64 MB address space, the lower 16 MB (0000000H to 0FFFFFFH) include four chip select functions, $\overline{CS0}$ to $\overline{CS3}$. The areas that can be selected by $\overline{CS0}$ to $\overline{CS3}$ are fixed as shown in Table 5-2.

However, since the V850ES/FJ2 has sixteen address pins (PDL0/AD0 to PDL15/AD15); 64 KB addresses can be selected linearly.

Table 5-2. Area Selected by Chip Select Function

Pin Name	Area
$\overline{CS0}$	0000000H to 01FFFFFH (2 MB)
$\overline{CS1}$	0200000H to 03FFFFFH (2 MB)
$\overline{CS2}$	0400000H to 07FFFFFH (4 MB)
$\overline{CS3}$	0800000H to 0FFFFFFH (8 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.

Table 5-3. Number of Clocks for Access

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]	3 + n
Instruction fetch (branch)	2	1	3 + n
Operand data access	3	1	3 + n

Note 2 if a conflict with a data access occurs.

Remark Unit: Clocks/access

5.4.2 Bus size setting function

The bus size of each external memory area selected by $\overline{CS0}$ to $\overline{CS3}$ can be set (to 8 bits or 16 bits) by using the bus size configuration (BSC) register.

The external memory area (01000000H to 0FFFFFFH) of the V850ES/FJ2 is selected by $\overline{CS0}$ to $\overline{CS3}$.

(1) Bus size configuration (BSC) register

The BSC 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.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1

7	6	5	4	3	2	1	0
0	BS30	0	BS20	0	BS10	0	BS00

$\overline{CSn}$ signal: $\overline{CS3}$ $\overline{CS2}$ $\overline{CS1}$ $\overline{CS0}$

BSn0	Data bus size of $\overline{CSn}$ space (n = 0 to 3)
0	8 bits
1	16 bits

Caution Be sure to set bits 14, 12, 10, and 8 to 1, and clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to 0.

5.4.3 Access according to bus size

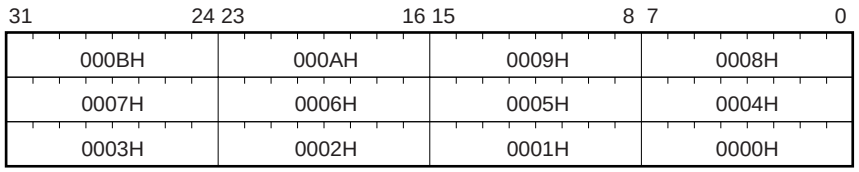
The V850ES/FJ2 accesses the on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The bus size is as follows.

- The bus size of the on-chip peripheral I/O is fixed to 16 bits.
- The bus size of the external memory is selectable from 8 bits or 16 bits (by using the BSC register).

The operation when each of the above is accessed is described below. All data is accessed starting from the lower side.

The V850ES/FJ2 supports only the little-endian format.

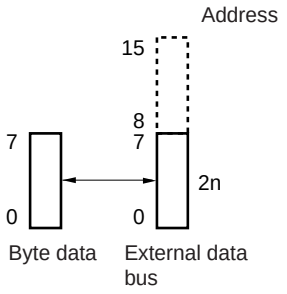
Figure 5-2. Little-Endian Address in Word



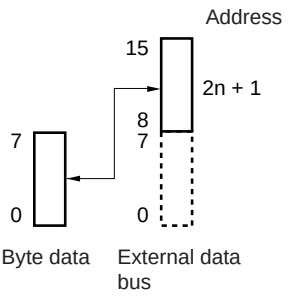
(1) Byte access (8 bits)

(a) 16-bit data bus width

<1> Access to even address (2n)

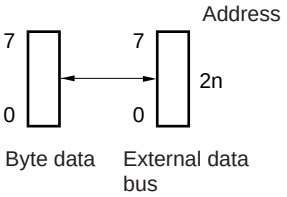


<2> Access to odd address (2n + 1)

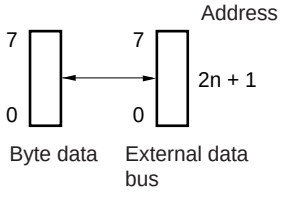


(b) 8-bit data bus width

<1> Access to even address (2n)



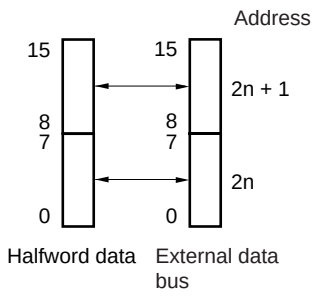
<2> Access to odd address (2n + 1)



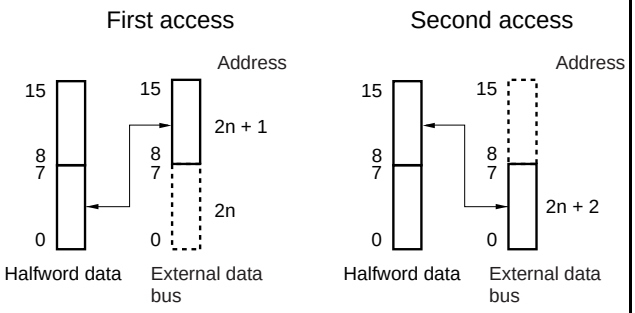
(2) Halfword access (16 bits)

(a) With 16-bit data bus width

<1> Access to even address (2n)

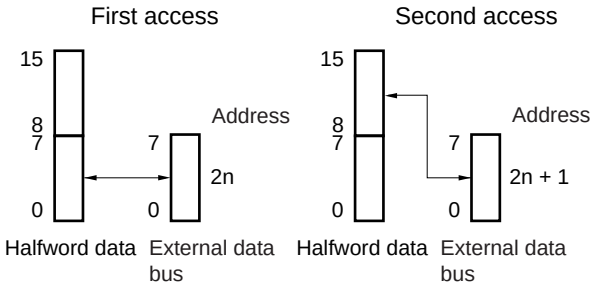


<2> Access to odd address (2n + 1)

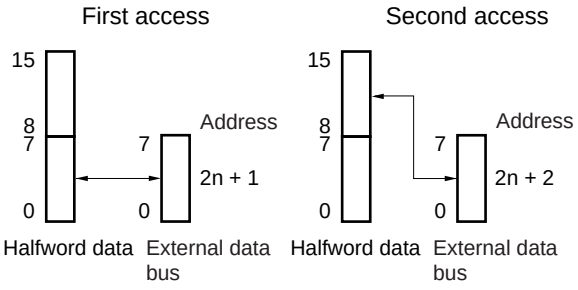


(b) 8-bit data bus width

<1> Access to even address (2n)



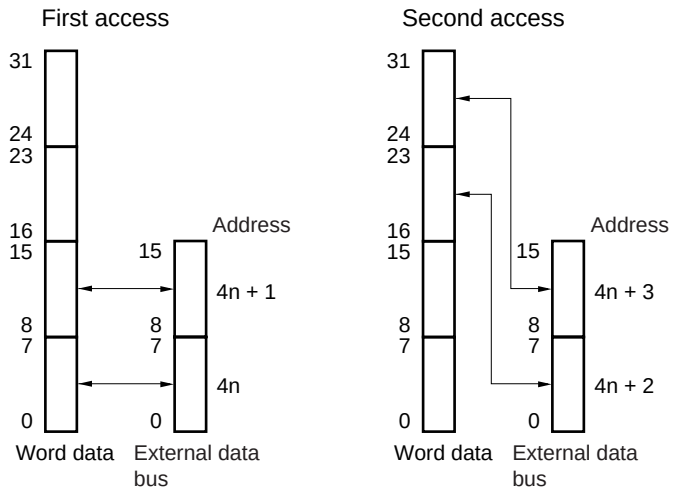
<2> Access to odd address (2n + 1)



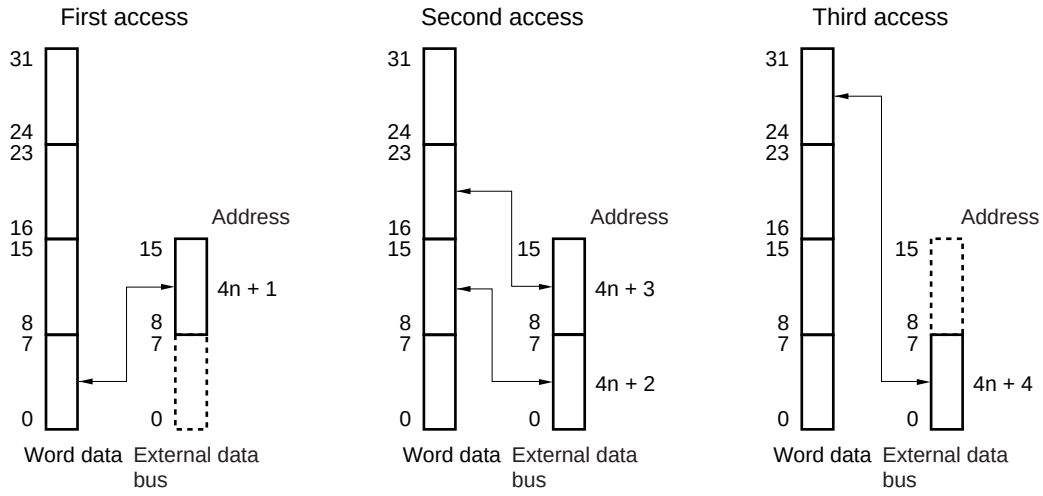
(3) Word access (32 bits)

(a) 16-bit data bus width (1/2)

<1> Access to address (4n)

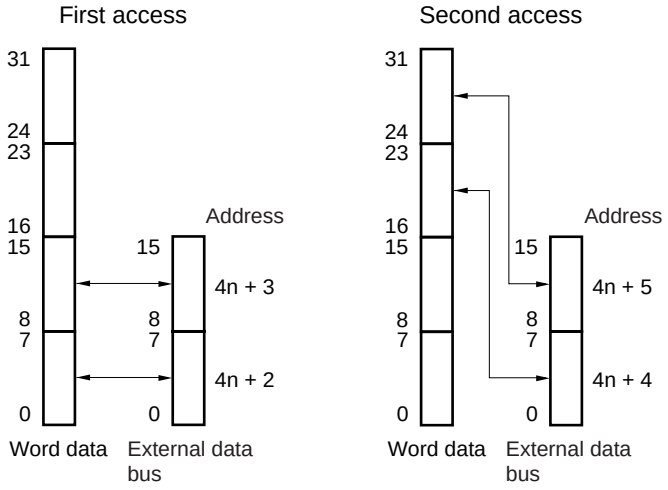


<2> Access to address (4n + 1)

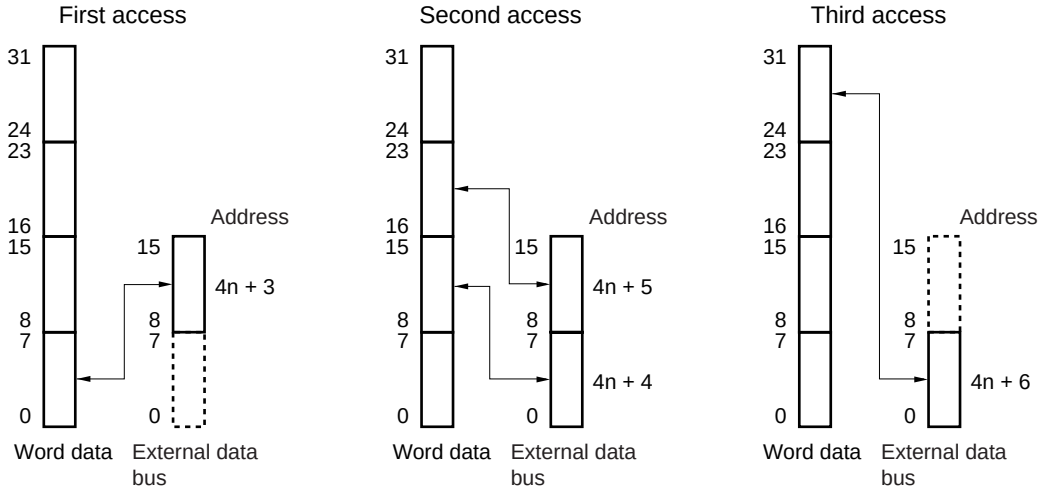


(a) 16-bit data bus width (2/2)

<3> Access to address (4n + 2)

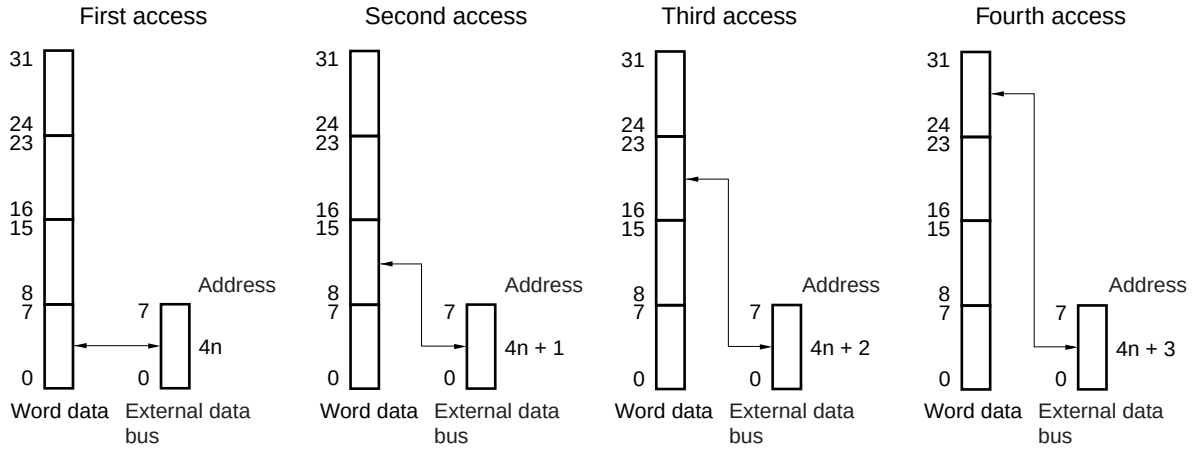


<4> Access to address (4n + 3)

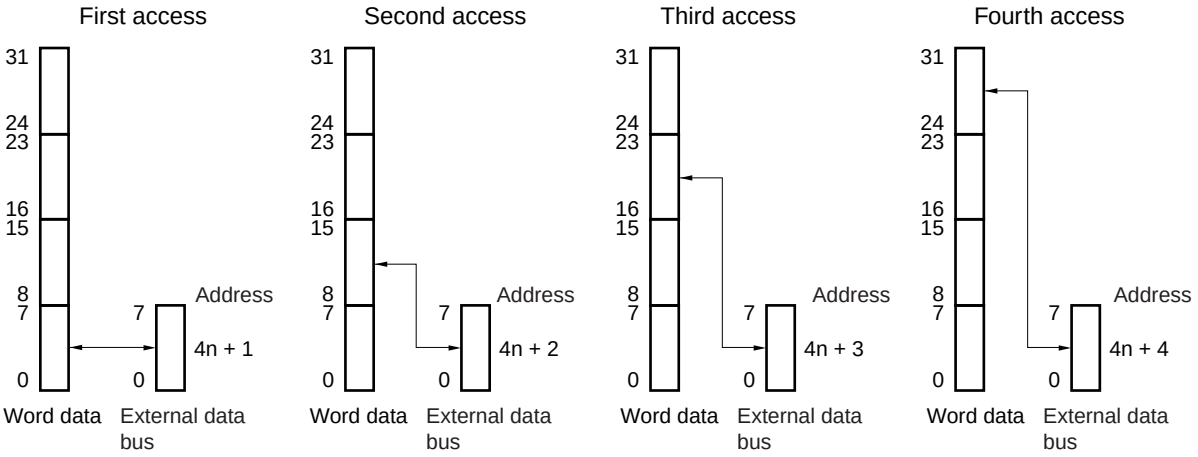


(b) 8-bit data bus width (1/2)

<1> Access to address (4n)

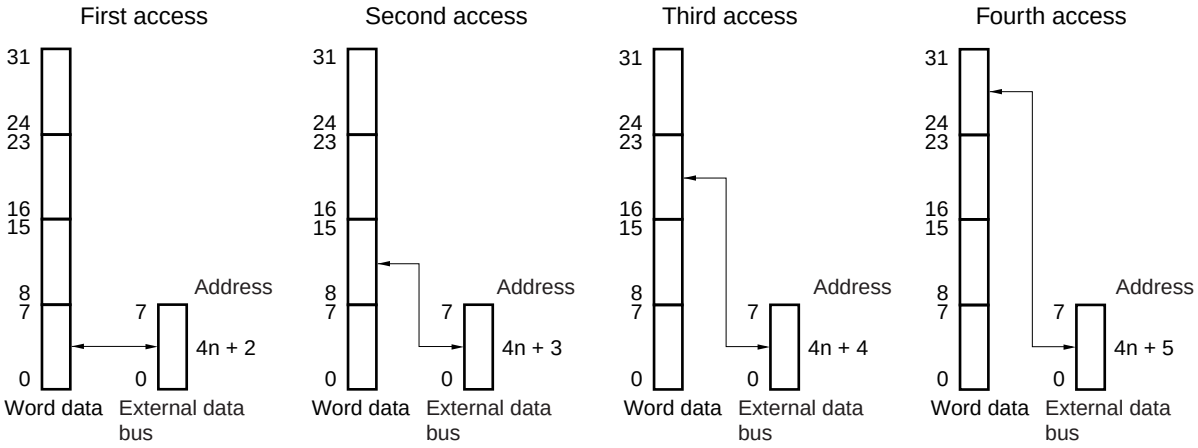


<2> Access to address (4n + 1)

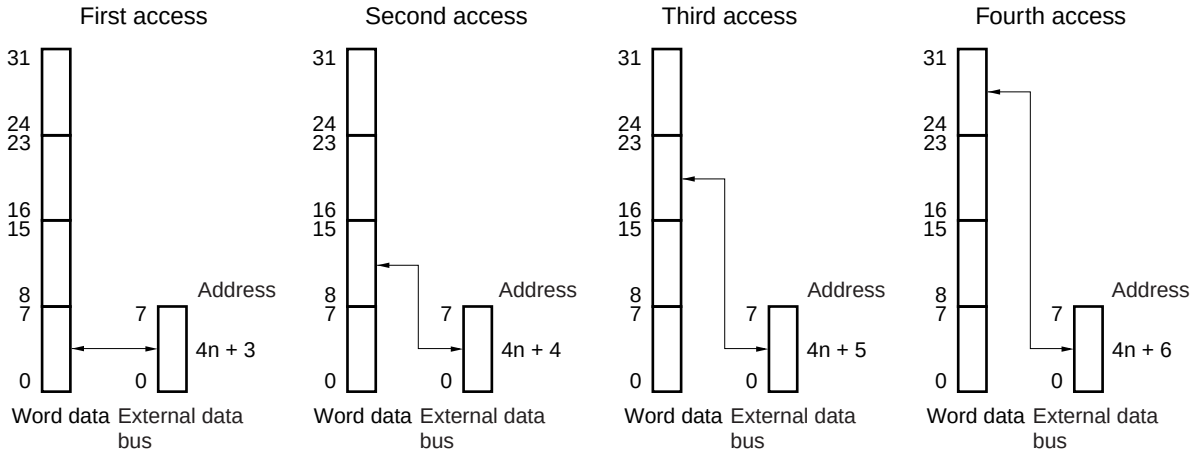


(b) 8-bit data bus width (2/2)

<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 memory block space.

The number of wait states can be programmed for each chip select area ($\overline{CS0}$ to $\overline{CS3}$) 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.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
$\overline{CSn}$ signal			$\overline{CS3}$				$\overline{CS2}$	

	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 $\overline{CSn}$ space (n = 0 to 3)
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 clear bits 15, 11, 7, and 3 to 0.

5.5.2 External wait function

To synchronize an extremely slow external memory, I/O device, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

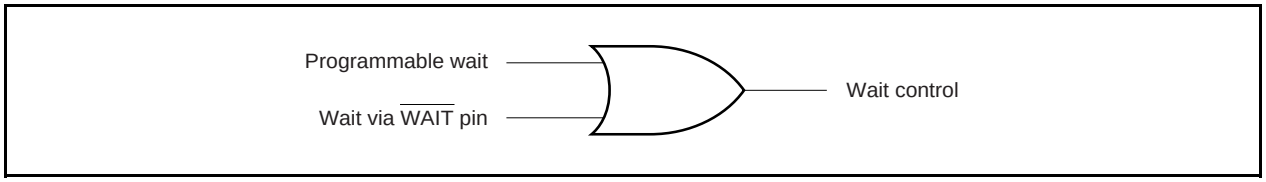
Access to each area of the internal ROM, internal RAM, and on-chip peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.

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. 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

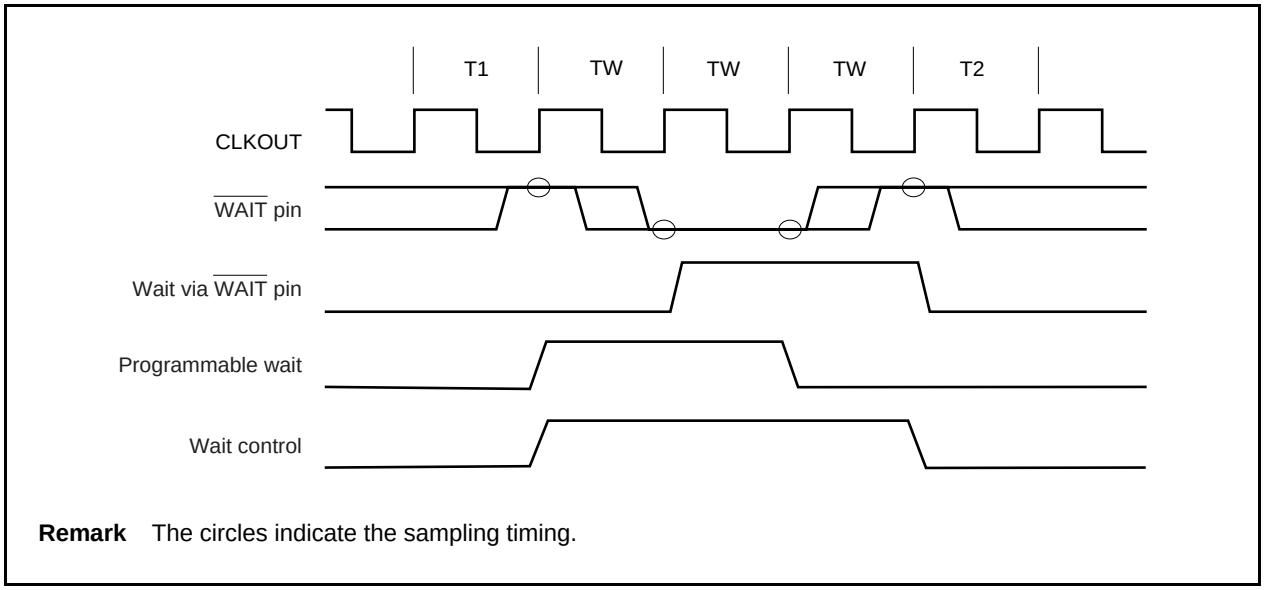
Wait cycles are inserted as the result of an OR operation between the wait cycles specified by the set value of the programmable wait and the wait cycles controlled by the $\overline{\text{WAIT}}$ pin.

Figure 5-3. Wait Control



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 5-4. Inserting Wait Example



Remark The circles indicate the sampling timing.

5.5.4 Programmable address wait function

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the address wait control register (AWC). Address wait insertion is set for each chip select area ($\overline{CS0}$ to $\overline{CS3}$).

If an address setup wait is inserted, it seems that the high-clock period of the T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of the T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

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

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
	AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
$\overline{CSn}$ signal		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$

AHWn	Specification of insertion of address hold wait in $\overline{CSn}$ space (n = 0 to 3)
0	Not inserted
1	Inserted

ASWn	Specification of insertion of address setup wait in $\overline{CSn}$ space (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15 to 8 to 1.

5.6 Idle State Insertion Function

To facilitate interfacing with low-speed memories, one idle state (TI) can be inserted after the T3 state in the bus cycle that is executed for each space selected by the chip select function. By inserting an idle state, the data output float delay time of the memory can be secured during read access (an idle state cannot be inserted during write access).

Whether the idle state is to be inserted can be programmed by using the bus cycle control (BCC) register.

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control (BCC) register

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

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area 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.

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0

	7	6	5	4	3	2	1	0
	BC31	0	BC21	0	BC11	0	BC01	0
$\overline{\text{CSn}}$ signal	$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

BCn1	Specification of insertion of idle state (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15, 13, 11, and 9 to 1, and clear bits 14, 12, 10, 8, 6, 4, 2, and 0 to 0.

5.7 Bus Hold Function

5.7.1 Functional outline

The $\overline{\text{HLDAK}}$ and $\overline{\text{HLDRQ}}$ functions are valid if the PCM2 and PCM3 pins are set in the control mode.

When the $\overline{\text{HLDRQ}}$ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus goes into a high-impedance state and is released (bus hold status). If the request for bus mastership is cleared and the $\overline{\text{HLDRQ}}$ pin is deasserted (high level), driving these pins is started again.

During the bus hold period, execution of the program in the internal ROM and internal RAM is continued until a peripheral I/O register or the external memory is accessed.

The bus hold status is indicated by assertion of the $\overline{\text{HLDAK}}$ pin (low level). The bus hold function enables the configuration of multi-processor type systems in which two or more bus masters exist.

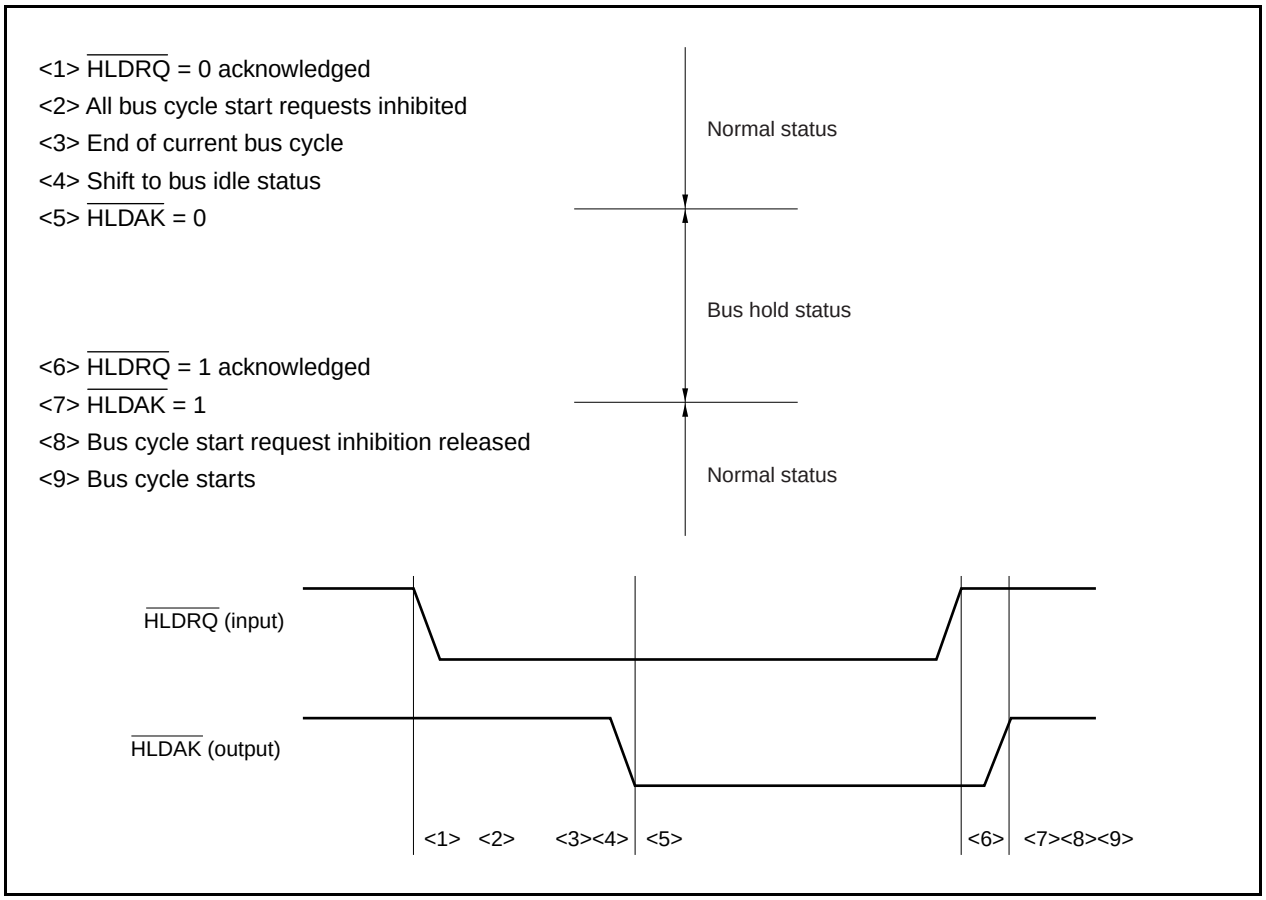
Note that the bus hold request is not acknowledged during a multiple-access cycle initiated by the bus sizing function or a bit manipulation instruction.

Status	Data Bus Width	Access Type	Timing at Which Bus Hold Request Is Not Acknowledged
CPU bus lock	16 bits	Word access to even address	Between first and second access
		Word access to odd address	Between first and second access
			Between second and third access
		Halfword access to odd address	Between first and second access
	8 bits	Word access	Between first and second access
			Between second and third access
			Between third and fourth access
		Halfword access	Between first and second access
Read-modify-write access of bit manipulation instruction	–	–	Between read access and write access

5.7.2 Bus hold procedure

The bus hold status transition procedure is shown in Figure 5-5.

Figure 5-5. Bus Hold Status Transition



5.7.3 Operation in power save mode

Because the internal system clock is stopped in the software STOP, IDLE1, IDLE2 and Sub IDLE modes, the bus hold status is not entered even if the HLDRQ pin is asserted.

In the HALT mode, the HLDAK pin is asserted as soon as the HLDRQ pin has been asserted, and the bus hold status is entered. When the HLDRQ pin is later deasserted, the 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 ↕ Low	Bus hold	External device
	DMA transfer	DMAC
	Operand data access	CPU
	Instruction fetch (branch)	CPU
	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 on-chip 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 V850ES/FJ2 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

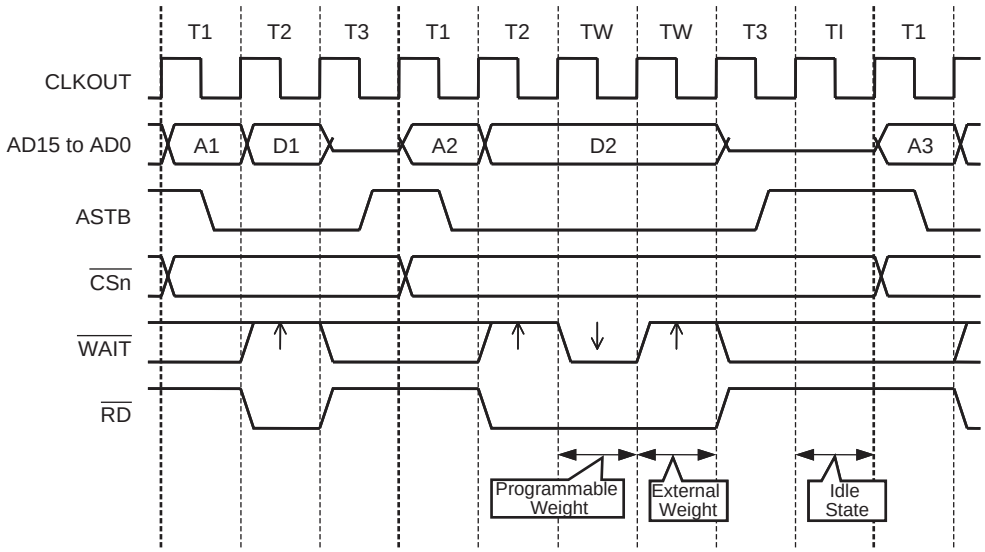
(a) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(b) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

5.10 Bus Timing

5.10.1 Multiplexed bus

Figure 5-6. Basic Bus Cycle



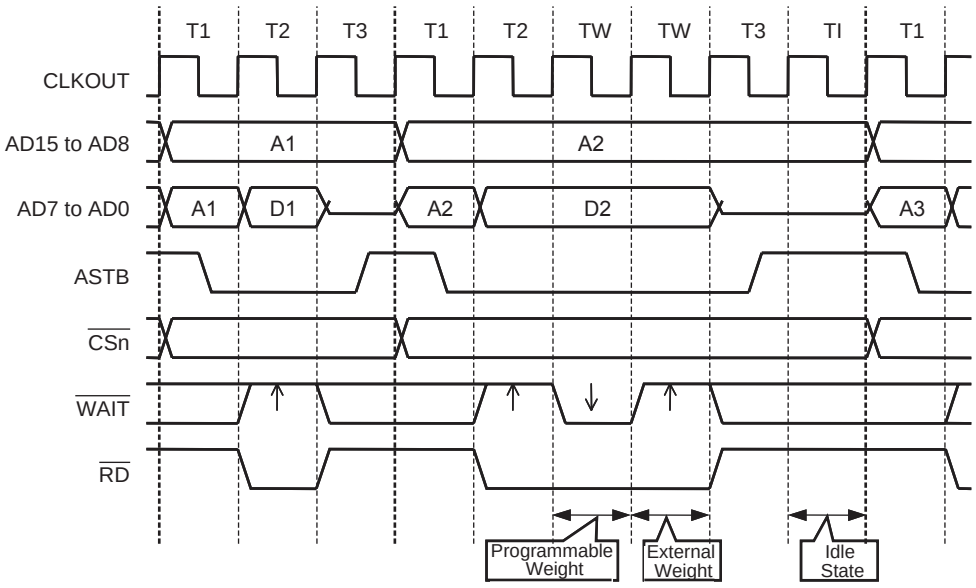
Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. $\overline{CSn}$ (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, $\overline{CSn}$ is always high level.

Remark ↑↓: Sampling clock

At the time of 8bit access	Odd number address	Even number address
AD15-AD8	Data	–
AD7-AD0	–	Data

Figure 5-7. When Wait State (1 Wait) Is Inserted

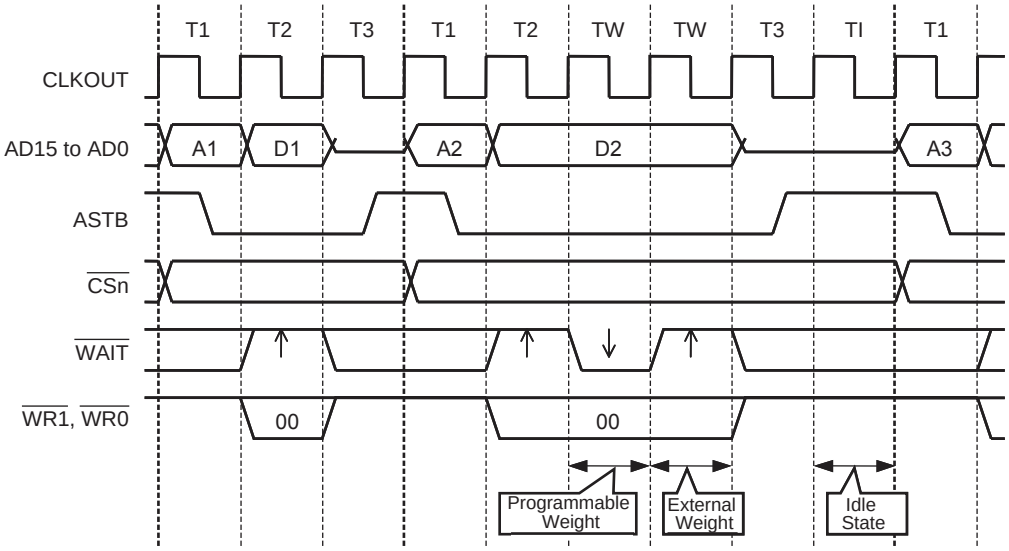


Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. $\overline{CSn}$ (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, $\overline{CSn}$ is always high level.

Remark ↑↓: Sampling clock

Figure 5-8. When Idle State Is Inserted



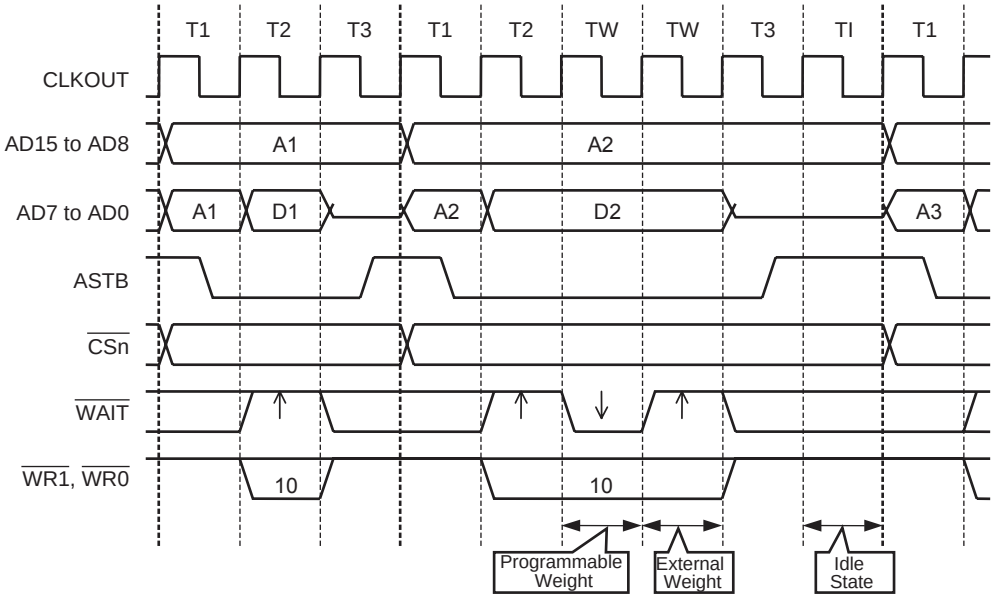
Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. CSn (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, CSn is always high level.

Remark. ↑↓: Sampling clock

At the time of 8bit access	Odd number address	Even number address
AD15-AD8	Data	Uncertain
AD7-AD0	Uncertain	Data
$\overline{WRn}$ (n = 1 or 0)	01	10

Figure 5-9. When Wait State (1 Wait) and Idle State Are Inserted

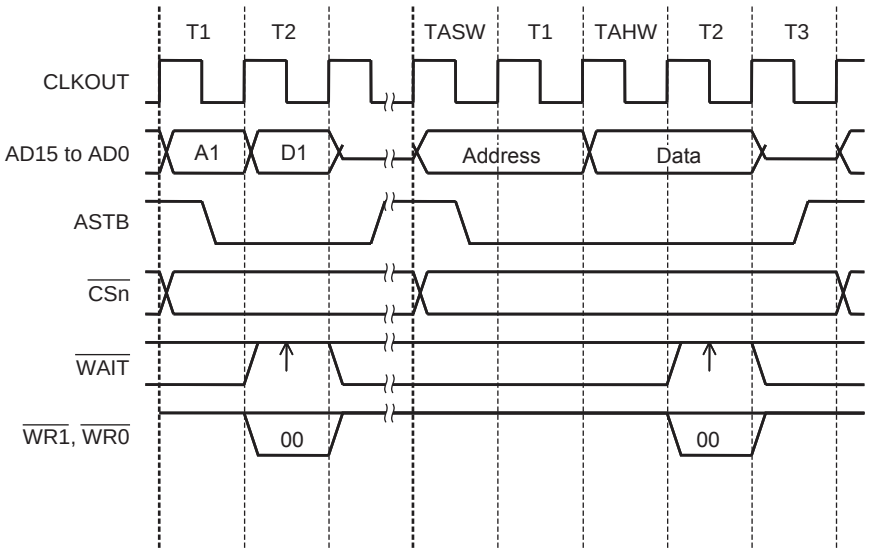


Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. $\overline{CSn}$ (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, $\overline{CSn}$ is always high level.

Remark. ↑↓: Sampling clock

Figure 5-10. When Address Wait State Is Inserted

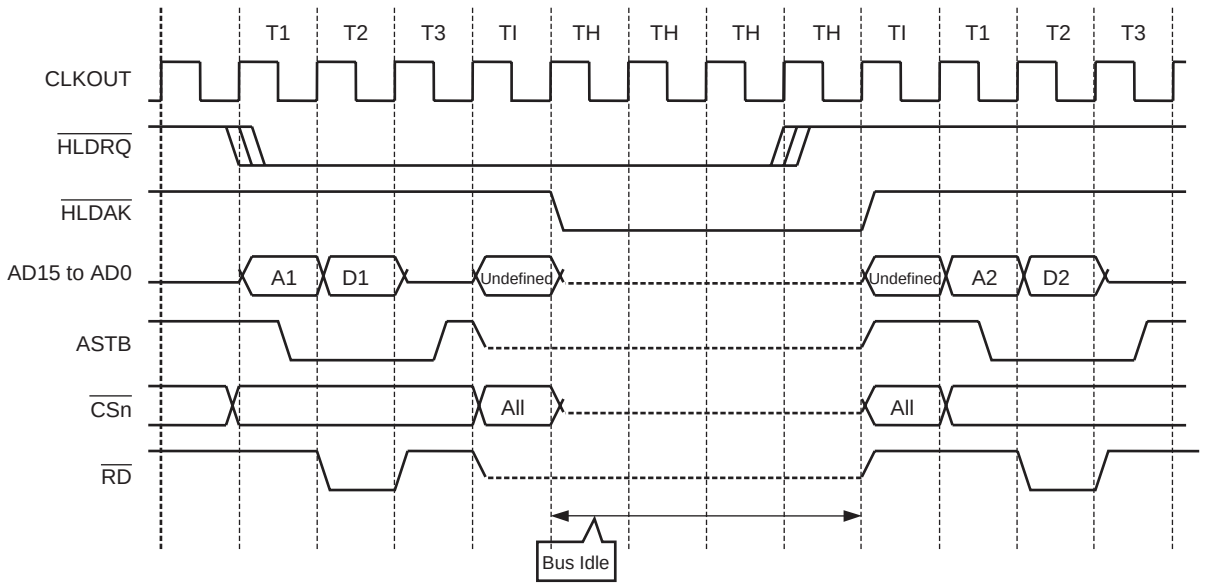


Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. $\overline{CSn}$ (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, $\overline{CSn}$ is always high level.

Remark. ↑: Sampling clock

Figure 5-11. Basic Bus Cycle



Notes 1. AD0 to AD7 hold the address output when odd address byte data is accessed.
AD8 to AD15 hold the address output when even address byte data is accessed.

2. $\overline{WR0}$ and $\overline{WR1}$ 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. $\overline{CSn}$ (n = 3 to 0) becomes low level, as shown above, when the corresponding CSn area is accessed. Otherwise, $\overline{CSn}$ is always high level.

4. Idle state (TI) that does not depend on the setting value of BCC register.

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Overview

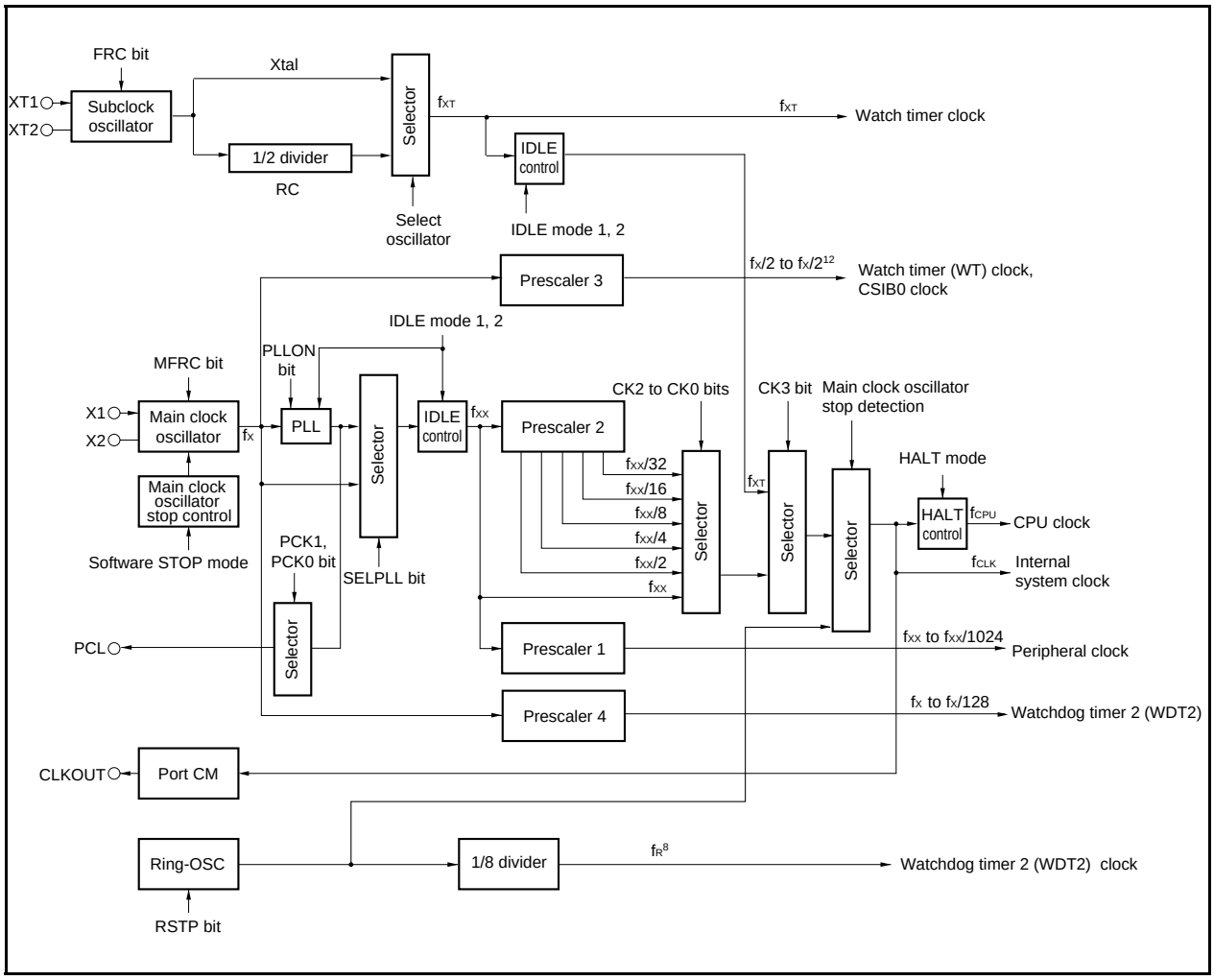
The following clock generation functions are available.

- Main clock oscillator
  - In clock-through mode
    f_X = 4 to 5 MHz (f_{XX} = 4 to 5 MHz)
  - In PLL (Phase Locked Loop) mode
    f_X = 4 to 5 MHz (f_{XX} = 16 to 20 MHz)
- Subclock oscillator (sub-resonator)
  - 32.768 kHz
  - 20 kHz (RCR = 390 kΩ, C = 47 pF)
- Multiply (×4) function via PLL (Phase Locked Loop)
  - Clock-through mode/PLL mode selectable
- Ring OSC
  - f_R = 200 kHz (TYP.)
- Internal system clock generation
  - 7 steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})
- Peripheral clock generation
- Clock output function
- Programmable clock output (PCL) function

Remarks: 1. f_X : main clock oscillation frequency
2. f_{XX} : Main clock frequency
3. f_R : Internal oscillator clock frequency

6.2 Configuration

Figure 6-1. Clock Generator



(1) Main clock oscillator

The main clock oscillator oscillates the following frequencies (fX).

- In clock-through mode
  fX = 4 to 5 MHz (internal fXX = 4 to 5 MHz)
- In PLL mode
  fX = 4 to 5 MHz (internal fXX = 16 to 20 MHz)

(2) Subclock oscillator

The subclock oscillator oscillates a frequency (fXT) of 32.768 kHz or 20 kHz.

(3) Main clock oscillator stop control

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

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

(4) Ring-OSC

Outputs a frequency (f_R) = 200 kHz (TYP.)

(5) Prescaler 1

This prescaler generates the clock (fxx to fxx/1,024) to be supplied to on-chip peripheral functions. Peripheral functions are as follows.

TMP0-TMP3, TMQ0-TMQ2, TMM0, CSIB0-CSIB2, UARTA0-UARTA3, ADC, WDT2.

(6) Prescaler 2

This circuit divides the CPU clock (f_{CPU}) and main clock (f_{XX}).

The clock generated by prescaler 2 (f_{XX} to $f_{XX}/32$) is supplied to the selector that generates the internal system clock (f_{CLK}).

f_{CLK} is the clock supplied to the INTC, ROM, and RAM blocks, and can be output from the CLKOUT pin.

(7) Prescaler 3

This circuit divides the clock generated by the main clock oscillator (f_X) to a specific frequency (32.768 kHz) and supplies that clock to the watch timer block.

For details, see **CHAPTER 10 WATCH TIMER FUNCTIONS**.

(8) Prescaler 4

This prescaler generates the clock (f_X to $f_X/1048$) to be supplied to on-chip peripheral functions such as the only WDT2.

(9) PLL

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

It operates in two modes: clock-through mode in which f_X is output as is, and PLL mode in which a multiplied clock is output. These modes can be selected by using the SELPLL bit of the PLL control register (PLLCTL).

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

6.3 Control Registers

(1) Processor clock control register (PCC)

The PCC register is a special register. Data can be written to this register in combination of specific sequences (see **3.4.9 Special registers**).

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

Reset input sets this register to 03H.

(1/2)

After reset: 03H R/W Address: FFFFF828H

	7	6	5	4	3	2	1	0
PCC	FRC	MCK	MFRC	CLS[Note]	CK3	CK2	CK1	CK0

FRC	Use of subclock on-chip feedback resistor
0	Used
1	Not used

MCK	Operation of main clock
0	Enable oscillation
1	Oscillation stopped

- Even if the MCK bit is set to 1 while the system is operating with the main clock as the CPU clock, the operation of the main clock does not stop. It stops after the CPU clock has been changed to the subclock.
- When the main clock is stopped and the device is operating on the subclock, clear the MCK bit to 0 and wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

MFRC	Use of main clock on-chip feedback resistor
0	Used
1	Not used

CLS	Status of CPU clock (f_{CPU})
0	Main clock operation
1	Subclock operation

Note The CLS bit is a read-only bit.

(2/2)

CK3	CK2	CK1	CK0	Clock selection (f_{CLK}/f_{CPU})
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	×	Setting prohibited
1	×	×	×	f_{XT}

Cautions 1. Do not change the CPU clock (by using the CK3 to CK0 bits) while CLKOUT is being output.

2. Use a bit manipulation instruction to manipulate the CK3 (0→1 or 1→0) bit. When using an 8-bit manipulation instruction, do not change the set values of the CK2 to CK0 bits.

Remark ×: don't care

(a) Example of setting main clock operation → subclock operation

<1> CK3 bit ← 1: Use a bit manipulation instruction. Do not change the CK2 to CK0 bits.

<2> Subclock operation: It takes up to the following number of instructions after the CK3 bit is set until the subclock operation is started.
1/subclock frequency (f_{XT})
Therefore, read the CLS bit to check if the subclock operation has started.

<3> MCK bit ← 1: Set the MCK bit to 1 only when stopping the main clock.

Cautions 1. When stopping the main clock, stop the PLL.

2. If the following conditions are not satisfied, change the CK2 to CK0 bits so that the conditions are satisfied, then change to the subclock operation mode. Setting clock by CK2-CK0 (fxx-fxx/32) > Subclock (f_{XT}) x4

(b) Example of setting subclock operation → main clock operation

<1> MCK bit ← 0: Main clock oscillation starts.

<2> Insert wait cycles by program and wait until the oscillation of the main clock has stabilized.

<3> CK3 bit ← 0: Use a bit manipulation instruction. Do not change the CK2 to CK0 bits.

<4> Main clock operation: it takes up to the following time after the CK3 bit is set until the main clock operation specified by the CK2 to CK0 bits is started.
Max.: (1/subclock frequency)
Therefore, read the CLS bit to check if the main clock operation has started.

(2) CPU operation clock status register (CCLS)

The CCLS register indicates the CPU operating clock status.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R Address: FFFFF82EH

	7	6	5	4	3	2	1	0
CCLS	0	0	0	0	0	0	0	CCLSF

CCLSF	CPU operating clock status
0	Operates on main clock (f_X) or subclock (f_{XT})
1	Operates on Internal oscillator (f_R)

Caution If WDT2 overflows before counting the oscillation stabilization time ends after a reset or STOP mode release, it is judged as abnormal oscillation of f_X (main clock) and the CPU operates on the Ring-OSC clock.

(3) Ring-OSC mode register (RCM)

The RCM register is an 8-bit register that sets the operation mode of Ring-OSC.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF80CH

	7	6	5	4	3	2	1	0
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	Operation/stop of Ring-OSC
0	Ring-OSC operating
1	Ring-OSC stopped

Caution 1. Ring-OSC can be stopped by setting the RSTOP bit of the RCM register to 1 only when "Ring-OSC stopped" is selected by the option function.

Caution 2. If RSTOP bit is set (1), the internal oscillator is oscillated by .CCLSF bit is set (1) (WDT overflow is generated in the oscillation stabilization time).Then, the RSTOP bit remains being set (1).

(4) Oscillation stabilization time select register (OSTS)

The OSTS register selects the oscillation stabilization time following reset or release of the STOP mode.

See **11.3 (1) Oscillation stabilization time select register (OSTS)**.

6.4 Operation

6.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 6-1. Operation Status of Each Clock

	PCC Register								
	CLS Bit = 0, MCK Bit = 0					CLS Bit = 1, MCK Bit = 0		CLS Bit = 1, MCK Bit = 1	
	<1>	<2>	<3>	<4>	<5>	<6>	<7>	<6>	<7>
Main clock oscillator (f_X)	×	○	○	○	×	○	○	×	×
Subclock oscillator (f_{XT})	○	○	○	○	○	○	○	○	○
CPU clock (f_{CPU})	×	×	×	×	×	○	×	○	×
Internal system clock (f_{CLK})	×	×	○	×	×	○	×	○	×
Peripheral clock (f_{XX} to f_{XX}/1,024)	×	×	○	×	×	○	×	×	×
WT clock (main)	×	×	○	○	×	○	○	×	×
WT clock (sub)	○	○	○	○	○	○	○	○	○
WDT2 clock (ring)	×	○	○	○	○	○	○	○	○
WDT2 clock (main)	×	×	○	○	×	○	○	×	×
Main clock oscillator (fxx)	×	×	○	×	×	○	×	×	×
PLL clock (f_{PLL})	×	Note1	○	Note 2	×	○	○	×	×

Notes:1. The stable clock is supplied from beginning operation after the time of 172 passes and through the lock-up time.

2. The operation enable at IDLE1 mode. Stopped at IDLE2 mode

Remark CLS bit: Bit 4 of the processor clock control register (PCC)
MCK bit: Bit 6 of the PCC register
○: Operable
×: Stopped
<1>: $\overline{\text{RESET}}$ pin input
<2>: During oscillation stabilization time count
<3>: HALT mode
<4>: IDLE1, IDLE2 mode
<5>: Software STOP mode
<6>: Subclock operation mode
<7>: Sub-IDLE mode

6.4.2 Clock output function

The clock output function is used to output the internal system clock (f_{CLK}) from the CLKOUT pin.

The internal system clock (f_{CLK}) is selected by using the CK3 to CK0 bits of the processor clock control register (PCC).

The CLKOUT pin functions alternately as the PCM1 pin and functions as a clock output pin if so specified by the control register of port CM.

The status of the CLKOUT pin is the same as the internal system clock in Table 6-1 and the pin can output the clock when it is in the operable status. It outputs a low level in the stopped status. However, the alternate-function pin (PCM1: input mode) is selected in <1> and <2> after the $\overline{\text{RESET}}$ signal has been input. Consequently, the CLKOUT pin goes into a high-impedance state.

6.5 PLL Function

6.5.1 Overview

The PLL function is used to output the operating clock of the CPU and peripheral macro at a frequency 4 times higher than the oscillation frequency, and select the clock-through mode.

When PLL function is used: Input clock = 4 to 5 MHz (output: 16 to 20 MHz)
Clock-through mode: Input clock = 4 to 5 MHz (output: 4 to 5 MHz)

6.5.2 Control registers

(1) PLL control register (PLLCTL)

The PLLCTL register is an 8-bit register that controls the PLL function.

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

Reset input sets this register to 01H.

After reset: 01H R/W Address: FFFFF82CH

	7	6	5	4	3	2	1	0
PLLCTL	0	0	0	0	0	0	SELPLL	PLLON

SELPLL	CPU operation clock selection
0	Clock-through mode
1	PLL mode

PLLON	Control of PLL operation/stop
0	PLL stopped
1	PLL operating (After PLL operation starts, a lockup time is required for frequency stabilization)

Cautions 1. The SELPLL bit can be set to 1 only when the PLL clock frequency is stabilized. If not (unlocked), "0" is written to the SELOLL bit if data is written to it.

2. When the PLLON bit is cleared to 0, the SELPLL bit is automatically cleared to 0 (clock-through mode).

(2) Lock register (LOCKR)

Phase lock occurs at a given frequency following power application or immediately after the software STOP mode is released, and the time required for stabilization is the lockup time (frequency stabilization time). This time until stabilization is called the lockup status, and the stabilized state is called the locked status.

The lock register (LOCKR) includes a LOCK bit that reflects the PLL frequency stabilization status.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Remark: At the lockup time (frequency stabilization time) LOCKR is set (1)

After reset: 00H R Address: FFFFF824H

	7	6	5	4	3	2	1	0
LOCKR	0	0	0	0	0	0	0	LOCK

LOCK	PLL lock status check
0	Locked status
1	Unlocked status

Caution The LOCK register does not reflect the lock status of the PLL in real time. The set/reset conditions are as follows.

[Set conditions]

- In IDLE2 or upon system reset[Note]
- In software STOP mode
- Upon setting of PLL stop (clearing of PLLON bit of PLLCTL register to 0)
- Upon stopping main clock and using CPU with subclock (setting of CK3 bit of PCC register to 1 and setting of MCK bit of same register to 1)

Note This register is set to 01H by reset and cleared to 00H after the reset has been released and the oscillation stabilization time has elapsed.

[Reset conditions]

- Upon overflow of oscillation stabilization time following reset release (OSTS register default time)
- Upon oscillation stabilization timer overflow (time set by OSTS register) following software STOP mode release, when the software STOP mode was set in the PLL operating status
- Upon PLL lockup timer overflow (time set by PLLS register) when the PLLON bit of the PLLCTL register is changed from 0 to 1
- Upon oscillation stabilization timer overflow (time set by OSTS register) following software IDLE2 mode release, when the software IDLE2 mode was set in the PLL operating status

(3) PLL lockup time specification register (PLLS)

The PLLS register is an 8-bit register used to select the PLL lockup time when the PLLON bit of the PLLCTL register is changed from 0 to 1.

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

Reset input sets this register to 03H.

After reset: 03H R/W Address: FFFFF6C1H

	7	6	5	4	3	2	1	0
PLLS	0	0	0	0	0	0	PLLS1	PLLS0

PLLS1	PLLS0	Selection of PLL lockup time
0	0	Setting prohibited
0	1	Setting prohibited
1	0	$2^{12}/f_X$
1	1	$2^{13}/f_X$ (default value)

Caution Set so that the lockup time is 800 μs or longer.

(4) Programmable clock mode register (PCLM)

The PCLM register is an 8-bit register used to control the PCL output.

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

After reset: 00H R/W Address: FFFFF82FH

	7	6	5	4	3	2	1	0
PCLM	0	0	0	PCLE	0	0	PCK1	PCK0

PCLE	Selection of PCL output operation
0	PCL output disabled (fixed to low level)
1	PCL output enabled

Caution Set the port-related control registers (PM, PMC, PFC, PFCE, etc.) first, and then set PCLE to 1.

PCK1	PCK0	Selection of PLL output clock
0	0	$f_{XX}/2$
0	1	$f_{XX}/4$
1	0	$f_{XX}/8$
1	1	$f_{XX}/16$

Caution Set PCLE to 1 only during PLL operation. To stop the PLL, clear PCLE to 0.

6.5.3 Usage

(1) To use PLL

- After the RESET signal has been released, the PLL operates (PLLON bit = 1), but because the default mode is the clock-through mode (SELPLL bit = 0), select the PLL mode (SELPLL bit = 1).
- To operate the PLL from the stopped status, set the PLLON bit to 1, and then set the SELPLL bit to 1 after the LOCKR.LOCK bit = 0 (the lockup time can be counted by setting the lockup time to the PLLS register and monitoring the LOCK flag of the LOCKR register).
- To stop the PLL, first select the clock-through mode (SELPLL bit = 0), wait for 8 clocks or more, and then stop the PLL (PLLON bit = 0)

When shifting to the IDLE2 or STOP mode while remaining in the PLL operation mode, set the OSTS register as follows.

- Software STOP mode: Oscillation stabilization time > PLL lockup time (800 μs (min.))
- IDLE2 mode: Setup time > PLL lockup time (800 μs (min.))

When shifting to the IDLE1 mode, the PLL does not stop.Stop the PLL if necessary.

(2) When PLL is not used

- The clock-through mode (SELPLL bit = 0) is selected after the $\overline{\text{RESET}}$ signal has been released, but the PLL is operating (PLLON bit = 1) and must therefore be stopped (PLLON bit = 0).

CHAPTER 7 16-BIT TIMER/EVENT COUNTER P

The V850ES/FE2, V850ES/FF2, V850ES/FG2, V850ES/FJ2 include 16-bit timer/event counter P (TMP0 to TMP3).

7.1 Features

Timer P (TMP) is a 16-bit timer/event counter that can be used in various ways.
TMP can perform the following operations.

- PWM output
- Interval timer
- External event counter (operation disabled when clock is stopped)
- One-shot pulse output
- Pulse width measurement function
- Timer synchronized operation function
- Free-running function
- External trigger pulse output function

7.2 Functional Outline

- Capture trigger input signal × 2
- External trigger input signal × 1
- Clock selection × 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 consists of the following hardware.

Table 7-1. Configuration of TMP0 to TMP3

Item	Configuration
Timer register	16-bit counter
Registers	TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter read buffer register (TPnCNT) CCR0, CCR1 buffer registers
Timer inputs	2 (TIPn0[Note 1], TIPn1)
Timer outputs	2 (TOPn0, TOPn1)
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option register 0 (TPnOPT0) Selector operation control registers 0, 1 (SELCNT0, SELCNT1[Note 2]) TIPnm pin noise elimination control register (PnmNFC)

Notes 1. TIPn0 functions alternately as a capture trigger input signal, external trigger input signal, and external event count input signal.

2. SELCNT1 is incorporated only in the μPD70F3239.

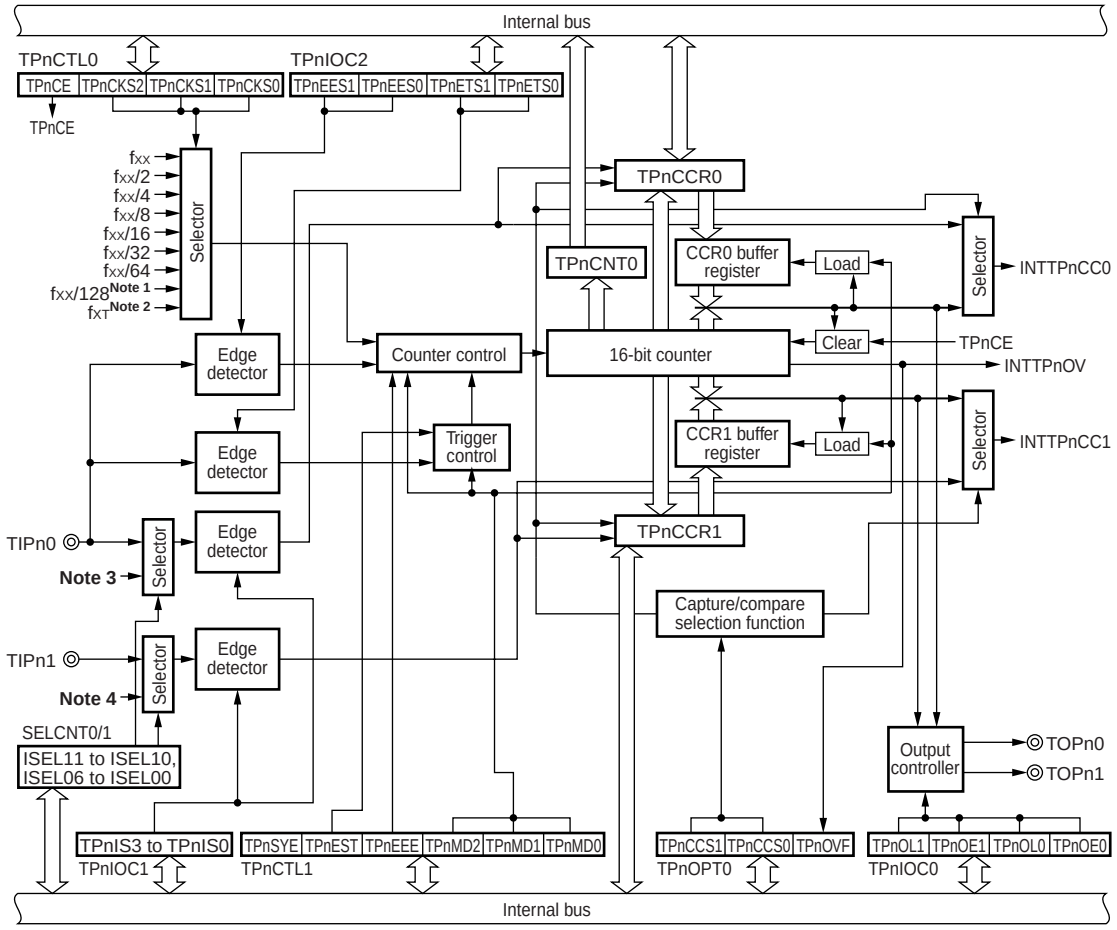
Remark n = 0 to 3, m = 0, 1

The pins of TMP function alternately as port pins. For how to set the alternate function, refer to the description of the registers in **CHAPTER 4 PORT FUNCTIONS**.

Table 7-2. TMP Pin List

Pin Name	Alternate-Function Pin	I/O	Function
TIP00	P32/ASCKA0/TOP00/TOP01	Input	External event/clock input (TMP0)
TIP01	P33/TOP01/CTXD0		
TIP10	P34/TOP10/CRXD0		External event/clock input (TMP1)
TIP11	P35/TOP11		
TIP20	P97/SIB1/TOP20		External event/clock input (TMP2)
TIP21	P96/TOP21		
TIP30	P01/TOP30		External event/clock input (TMP3)
TIP31	P00/TOP31		
TOP00	P32/ASCKA0/TIP00/TOP01	Output	Timer output (TMP0)
TOP01	P32/ASCKA0/TIP00/TOP00		
	P33/TIP01/CTXD0		
TOP10	P34/TIP10/CRXD0		Timer output (TMP1)
TOP11	P35/TIP11		
TOP20	P97/SIB1/TIP20		Timer output (TMP2)
TOP21	P96/TIP21		
TOP30	P01/TIP30		Timer output (TMP3)
TOP31	P00/TIP31		

Figure 7-1. Block Diagram of Timer P



Notes 1. TMP0, TMP2

2. TMP1, TMP3

3. TSOUT signal of CAN0 block (TMP0)
RXDA0 pin (TMP1)
TSOUT signal of CAN2 block (TMP2)
RXDA2 pin (TMP3)
Refer to **7.4 (7) Selector operation control register 0 (SELCNT0)** and **7.4 (8) Selector operation control register 1 (SELCNT1)**.

4. INTTM0EQ0 interrupt of TMM block or TSOUT signal of CAN1 block (TMP0)
RXDA1 pin (TMP1)
TSOUT signal of CAN3 block (TMP2)
RXDA3 pin (TMP3)
Refer to **7.7 (1) Selector operation control register 0 (SELCNT0)** and **7.7 (2) Selector operation control register 1 (SELCNT1)**.

Remark n = 0 to 3

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

The TPnCCR0 register is a 16-bit register that has a capture function and a compare function.

Only in the free-running mode, this register is used as a capture register or a compare register, this behavior can be specified by using the TPnCCS0 bit of the TPnOPT0 register.

In the pulse width measurement mode, this register works only as a capture 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 TPnCCR0 register functions as a compare register.

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

Reset input clears this register to 0000H.

Caution At the time of subclock operated and main clock stopped, the access to TPnCCR0 is prohibited. For details, refer to 3. 4. 10（2）

After reset: 0000H R/W 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
TPnCCR0																

(n = 0 to 3)

- When used as compare register

  TPnCCR0 can be rewritten when TPnCE = 1.

TMP Operation Mode	Method of Writing TPnCCR0 Register
PWM output mode or external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse output mode, or interval timer mode	Anytime 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 TPnCCR0 on detection of the edge of the capture trigger (TIPn0) input.

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

The TPnCCR1 register is a 16-bit register that has a capture function and a compare function.

Only in the free-running mode, this register is used as a capture register or a compare register, this behavior can be specified by using the TPnCCS0 bit of the TPnOPT0 register.

In the pulse width measurement mode, this register works only as a capture register.

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

Reset input clears this register to 0000H.

Caution At the time of subclock operation and main clock stopped, the access to TPnCCR1 is prohibited. For details, refer to 3. 4. 10（2）

After reset: 0000H R/W 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
TPnCCR1																

(n = 0 to 3)

- When used as compare register

  TPnCCR1 can be rewritten when TPnCE = 1.

TMP Operation Mode	Method of Writing TPnCCR1 Register
PWM output mode or external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse output mode, or interval timer mode	Anytime 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 TPnCCR1 on detection of the edge of the capture trigger (TIPn1) input.

(3) TMPn counter read buffer register (TPnCNT)

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

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

Reset input sets this register to FFFFH.

Although the hardware status is FFFFH when TPnCE = 0, 0000H is read from this register.

The counter value of the 16-bit counter is read when TPnCE = 1.

Caution At subclock operated and at main clock stopped, the acsess to TPnCNT register is not enable. For details, refer to 3. 4. 10（2）

After reset: FFFFH R 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
TPnCNT																

(n = 0 to 3)

7.4 Control Registers

(1) TMPn control register 0 (TPnCTL0)

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

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

Reset input clears this register to 00H.

TPnCTL0 register cannot be rewritten in operating. But only TPnCE bit can always be rewritten.

(1/2)

After reset: 00H R/W Address: TP0CTL0: FFFFF590H, TP1CTL0: FFFFF5A0H, TP2CTL0: FFFFF5B0H, TP3CTL0: FFFFF5C0H

	7	6	5	4	3	2	1	0
TPnCTL0 (n = 0 to 3)	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	Control of operation of timer Pn
0	Disable internal operating clock operation (asynchronously reset TMPn).
1	Enable internal operating clock operation.

The TPnCE bit controls the internal operating clock and asynchronously resets TMPn. When this bit is cleared to 0, the internal operating clock of TMPn is stopped (fixed to the low level), and TMPn is asynchronously reset.

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	Selection of internal count clock	
			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_{XT}

Caution Set the TPnCKS2 to TPnCKS0 bits when TPnCE = 0. When the TPnCE bit setting is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set at the same time.

Remark f_{XX}: Main system clock frequency
f_{XT}: XT1 input clock frequency

(2/2)

Resolution and maximum number of counts

Internal count clock	Resolution [μs]		Maximum count time [ms]	
	f_{XX} = 16 MHz	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 20 MHz
f_{XX}	0.0625	0.050	4.10	3.28
$f_{XX}/2$	0.125	0.100	8.19	6.55
$f_{XX}/4$	0.250	0.200	16.38	13.11
$f_{XX}/8$	0.500	0.400	32.77	26.21
$f_{XX}/16$	1.000	0.800	65.54	52.43
$f_{XX}/32$	2.000	1.600	131.11	104.86
$f_{XX}/64$	4.000	3.200	262.14	209.72
$f_{XX}/128$	8.000	6.400	524.29	419.43

Internal count clock	Resolution [μs]	Maximum count time [ms]
	f_{XT} = 32.768 kHz	f_{XT} = 32.768 kHz
f_{XT}	30.52	2000.00

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the operation of timer P.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0CTL1: FFFFF591H, TP1CTL1: FFFFF5A1H, TP2CTL1: FFFFF5B1H, TP3CTL1: FFFFF5C1H

	7	6	5	4	3	2	1	0
TPnCTL1	TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

(n = 0 to 3)

TPnSYE	Tuned operation mode enable control
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer P can operate in synchronization with a master timer. For the tuned operation mode, refer to **7.6 Timer Synchronized Operation Function**. **Caution Be sure to clear the TP0SYE and TP2SYE bits to 0.**

Master timer	Slave timer	
TMP0	TMP1	–
TMP2	TMP3	TMQ0
TMQ1	TMQ2	–

TPnEST	Software trigger control
0	No operation
1	In one-shot pulse mode: One-shot pulse software trigger In external trigger pulse output mode: Pulse output software trigger
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.	

TPnEEE	Selection of count clock
0	Internal clock (clock selected by TPnCKS2 to TPnCKS0 bits)
1	External event count input (edge of input to TIPn0)
The valid edge is specified by the TPnEES1 and TPnEES0 bits when TPnEEE = 1 (External event count input: TIPn0).	

(2/2)

TPnMD2	TPnMD1	TPnMD0	Selection of timer mode
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse mode
1	0	0	PWM mode
1	0	1	Free-running mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Cautions

1. **Set the TPnEEE and TPnMD2 to TPnMD0 bits when TPnCE = 0 (the same value can be written when TPnCE = 1). If these bits are rewritten when TPnCE = 1, the operation cannot be guaranteed. If these bits are rewritten by mistake, clear TPnCE to 0 and then set them again.**
2. **The external event count input is selected regardless of the value of the TPnEEE bit at an external event count mode.**
3. **Set "0" to bit 3 and 4.**

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer outputs (TOPn0 and TOPn1).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC0: FFFFF592H, TP1IOC0: FFFFF5A2H, TP2IOC0: FFFFF5B2H, TP3IOC0: FFFFF5C2H

	7	6	5	4	3	2	1	0
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

(n = 0 to 3)

TPnOLm	Setting of TOPnm output level (m = 0, 1)
0	Normal output
1	Inverted output

TPnOEm	Setting of TOPnm output (m = 0, 1)
0	Disable timer output (TOPnm pin outputs low level when TPnOLm = 0, and high level when TPnOLm = 1).
1	Enable timer output (TOPnm pin outputs pulses).

Cautions 1. Rewrite the TPnOL1, TPnOE1, TPnOL0 and TPnOE0 bits when TPnCE = 0 (the same value can be written when TPnCE = 1). If these bits are rewritten by mistake, clear TPnCE to 0 and then set them again.

2. To enable the timer output, be sure to set the corresponding alternate-function pins TPnIS3 to TPnIS0 of the TPnIOC1 register to "Detect no edge" and invalidate the capture operation. Then set the corresponding alternate-function port to output mode.

3. In the state of TPnCE bit = 0 and TPnOEm bit = 0, the output level of TOPnm pin changes even when the TPnOLm bit is operated.

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge of the external input signals (TIPn0 and TIPn1).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC1: FFFFF593H, TP1IOC1: FFFFF5A3H, TP2IOC1: FFFFF5B3H, TP3IOC1: FFFFF5C3H

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0 to 3)

TPnIS3	TPnIS2	Setting of valid edge of capture input (TIPn1)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TPnIS1	TPnIS0	Setting of valid edge of capture input (TIPn0)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

Cautions 1. Rewrite the TPnIS3 to TPnIS0 bits when TPnCE0 = 0 (the same value can be written when TPnCE = 1). If these bits are rewritten by mistake, clear TPnCE to 0 and then set them again.

2. The TPnIS3 to TPnIS0 bits are valid only in the free-running mode and pulse width measurement mode. A capture operation is not performed in any other mode.

3. If used as the capture input, be sure to set the corresponding alternate-function pins TPnOE1 and TPnOE0 of the TPnIOC0 register to "Disable timer output" and set the capture input valid edge. Then set the corresponding alternate-function port to input mode

4. Set it without the edge detection of the TIPn0 capture input (TPnIS1 and 0 bit = 00b) when using it in the external event count mode (TPnEEE bit = 1 of TPnCTL1).

(5) TMPn I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIPn0) and external trigger input signal (TIPn0).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC2: FFFFF594H, TP1IOC2: FFFFF5A4H, TP2IOC2: FFFFF5B4H, TP3IOC2: FFFFF5C4H

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

(n = 0 to 3)

TPnEES1	TPnEES0	Setting of valid edge of external event count input (TIP00)
0	0	Detect no edge (external event count is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TPnETS1	TPnETS0	Setting of valid edge of external trigger input (TIP00)
0	0	Detect no edge (external trigger is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

Cautions 1. Rewrite the TPnEES1, TPnEES0, TPnETS1and TPnETS0 bits when TPnCE = 0 (the same value can be written when TPnCE = 1). If these bits are rewritten by mistake, clear TPnCE to 0 and then set them again.

2. The TPnEES1 and TPnEES0 bits are valid when TPnEEE = 1 or when the external event count mode is set (TPnMD2 to TPnMD of TIPnCTL1 register = 001).

3. TPnETS1and TPnETS0 bits are valid when the external trigger pulse output mode (TPnMD2-0=010b of TPnCTL register) and the one-shot pulse output mode (TPnMD2-0=011b of TPnCTL1 register) is set.

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register that selects a capture or compare operation, and detects an overflow.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0OPT0: FFFFF595H, TP1OPT0: FFFFF5A5H, TP2OPT0: FFFFF5B5H, TP3OPT0: FFFFF5C5H

	7	6	5	4	3	2	1	0
TPnOPT0 (n = 0 to 3)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCSm	Selection of capture or compare operation of TPnCCRm register (m = 0, 1)
0	Compare register
1	Capture register
The set value of the TPnCCSm bit is valid only in the free-running mode.	

TPnOVF	Detection of overflow of timer P
Set (1)	Overflow occurred
Reset (0)	0 written to TPnOVF bit or TPnCE = 0

- The TPnOVF bit is set when the 16-bit counter overflows from FFFFH to 0000H in the free-running mode and pulse width measurement mode.
- As soon as the TPnOVF bit has been set to 1, an interrupt request signal (INTTPnOV) is generated. The INTTPnOV signal is not generated in any mode other than the free-running mode and pulse width measurement mode.
- The TPnOVF bit is not cleared even if the TPnOVF bit and TPnOPT0 register are read when TPnOVF = 1.
- The TPnOVF bit can be read and written, but 1 cannot be written to the TPnOVF bit. Writing 1 to this bit does not affect the operation of timer P.

Caution Rewrite the TPnCCS1 and TPnCCS0 bits when TPnCE0 = 0 (the same value can be written when TPnCE = 1). If these bits are rewritten by mistake, clear TPnCE to 0 and then set them again.

(7) 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 for noise elimination.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: 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)

	7	6	5	4	3	2	1	0
PnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFSTS	Setting of number of times of sampling by digital noise filter
0	3 times
1	2 times

NFC2	NFC1	NFC0	Sampling clock	
			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	$f_{XX}/16$	$f_{XX}/8$
1	0	0	$f_{XX}/32$	$f_{XX}/16$
1	0	1	$f_{XX}/64$	f_{XT}
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 performs the following operations.

Operation	TPnEST (Software Trigger Bit)	TIPn0 (External Trigger Input)	Capture/Compare Selection	Compare Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode Note 1	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode Note 2	Valid	Valid	Compare only	Reload
One-shot pulse output mode Note 2	Valid	Valid	Compare only	Anytime write
PWM mode	Invalid	Invalid	Compare only	Reload
Free-running mode	Invalid	Invalid	Capture/compare selectable	Anytime write
Pulse width measurement mode Note 2	Invalid	Invalid	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).

Remark n = 0 to 3

7.5.1 Anytime write and reload

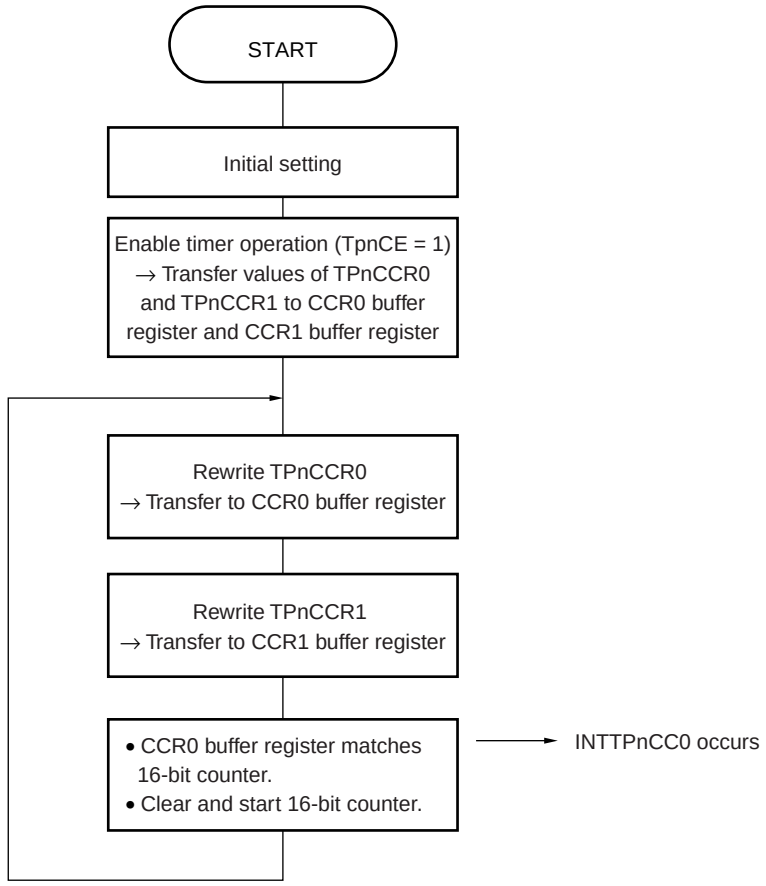
Timer P allows rewriting of the TPnCCR0 and TPnCCR1 registers while the timer is operating (TPnCE = 1). These registers are written differently (anytime write or reload) depending on the mode.

(1) Anytime write

When data is written to the TPnCCRm register during timer operation, it is transferred at any time to the CCRm buffer register and is compared with the value of the 16-bit counter.

Remark n = 0 to 3
m = 0, 1

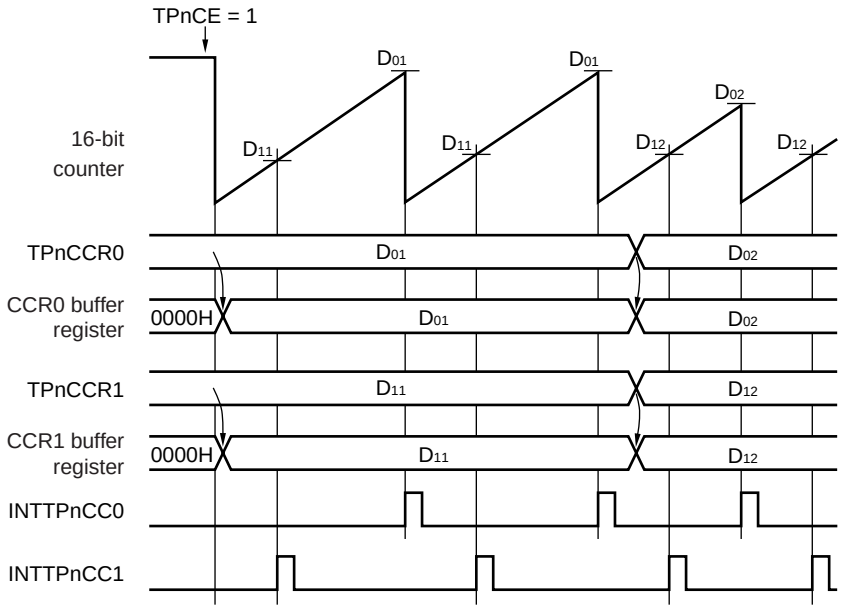
Figure 7-2. Flowchart of Basic Operation of Anytime Write



Remarks 1. This is an example in the interval timer mode.

2. n = 0 to 3

Figure 7-3. Timing Chart of Anytime Write



Remarks 1. D_{01}, D_{02}: Set value of TPnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TPnCCR1 register (0000H to FFFFH)

2. This is an example in the interval timer mode.

3. n = 0 to 3

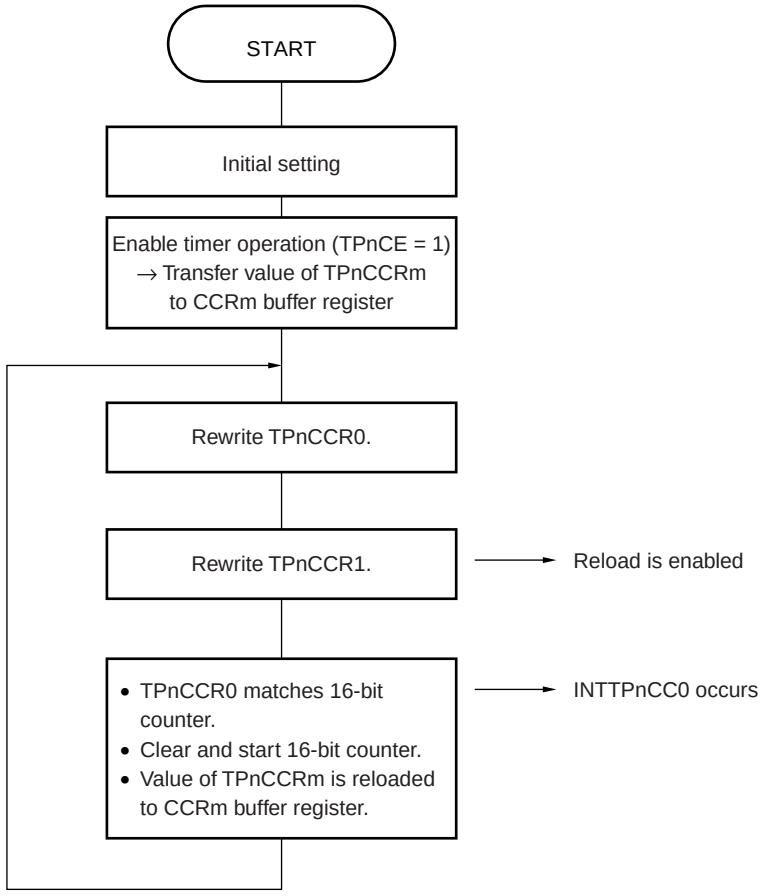
(2) Reload

When data is written to the TPnCCR0 and TPnCCR1 registers during timer operation, it is compared with the value of the 16-bit counter transferred to the CCRm buffer register after reserved until the written value becoming a specific state. 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 7-4. Flowchart of Basic Operation of Reload

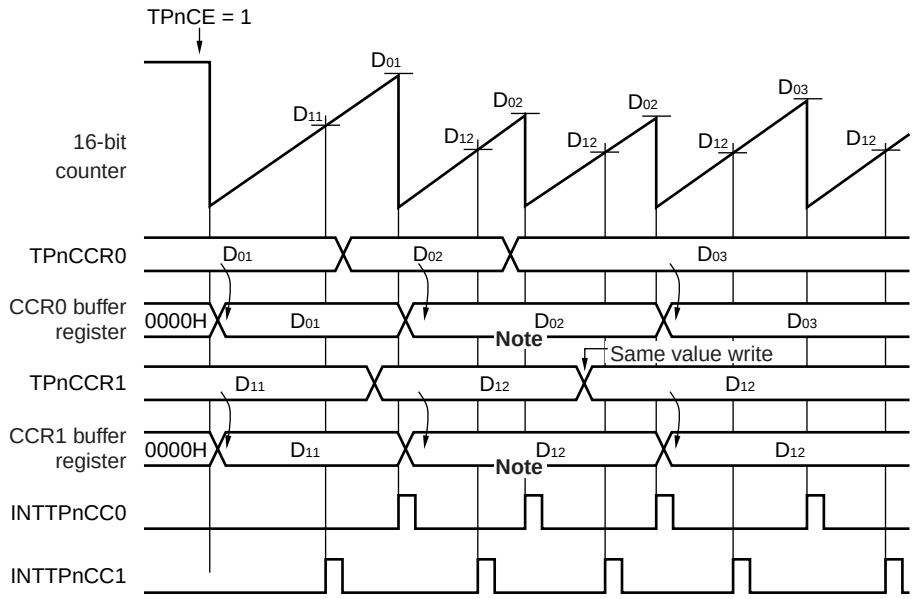


Caution Writing the TPnCCR1 register includes an operation to enable reload. Therefore, rewrite the TPnCCR1 register after rewriting the TPnCCR0 register.

Remarks 1. This is an example in the PWM mode.

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

Figure 7-5. Timing Chart of Reload



Note The value is not reloaded because the TPnCCR1 register is not written.

Remarks 1. D_{01}, D_{02}, D_{03}: Set value of TPnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TPnCCR1 register (0000H to FFFFH)

2. This is an example in the PWM mode.

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 when the set value of the TPnCCR0 register matches the value of the 16-bit counter, and the 16-bit counter is cleared. Rewriting the TPnCCR0 register is enabled when TPnCE = 1. When a value is set to the TPnCCRm register, it is transferred to the CCRm buffer register by means of anytime write, and is compared with the value of the 16 bit counter.

The 16-bit counter is not cleared by using the TPnCCR1 register.

However, the set value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter. As a result, an interrupt request (INTTPnCC1) is generated.

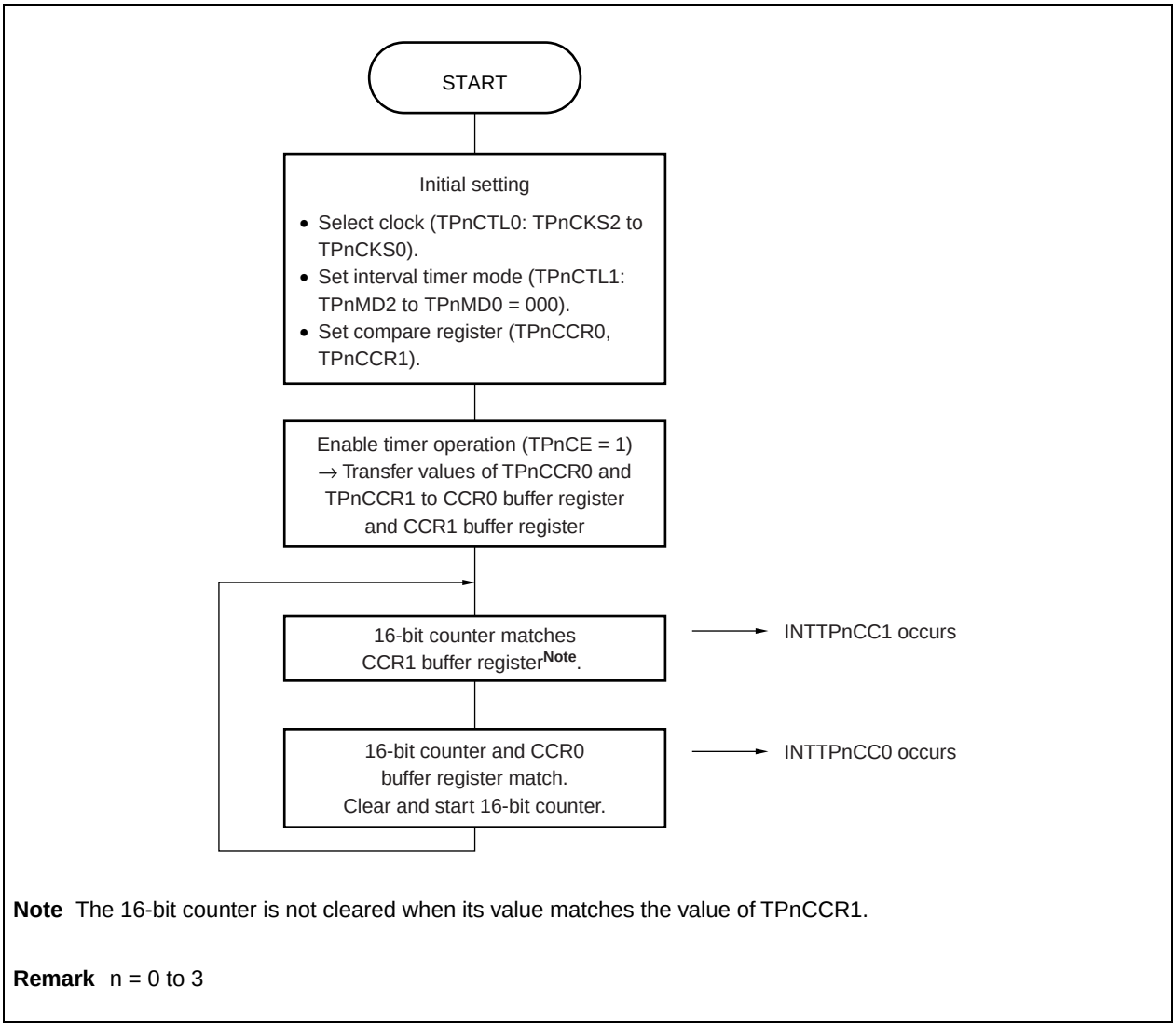
The value can also be output from the TOPnm pin by setting the TPnOEm bit to 1.

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

Remarks 1. Refer to 7.5.1 Anytime write and reload about write operation of TPnCCR0, TPnCCR1 during timer operation (TPnCE = 1).

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

Figure 7-6. Flowchart of Basic Operation in Interval Timer Mode

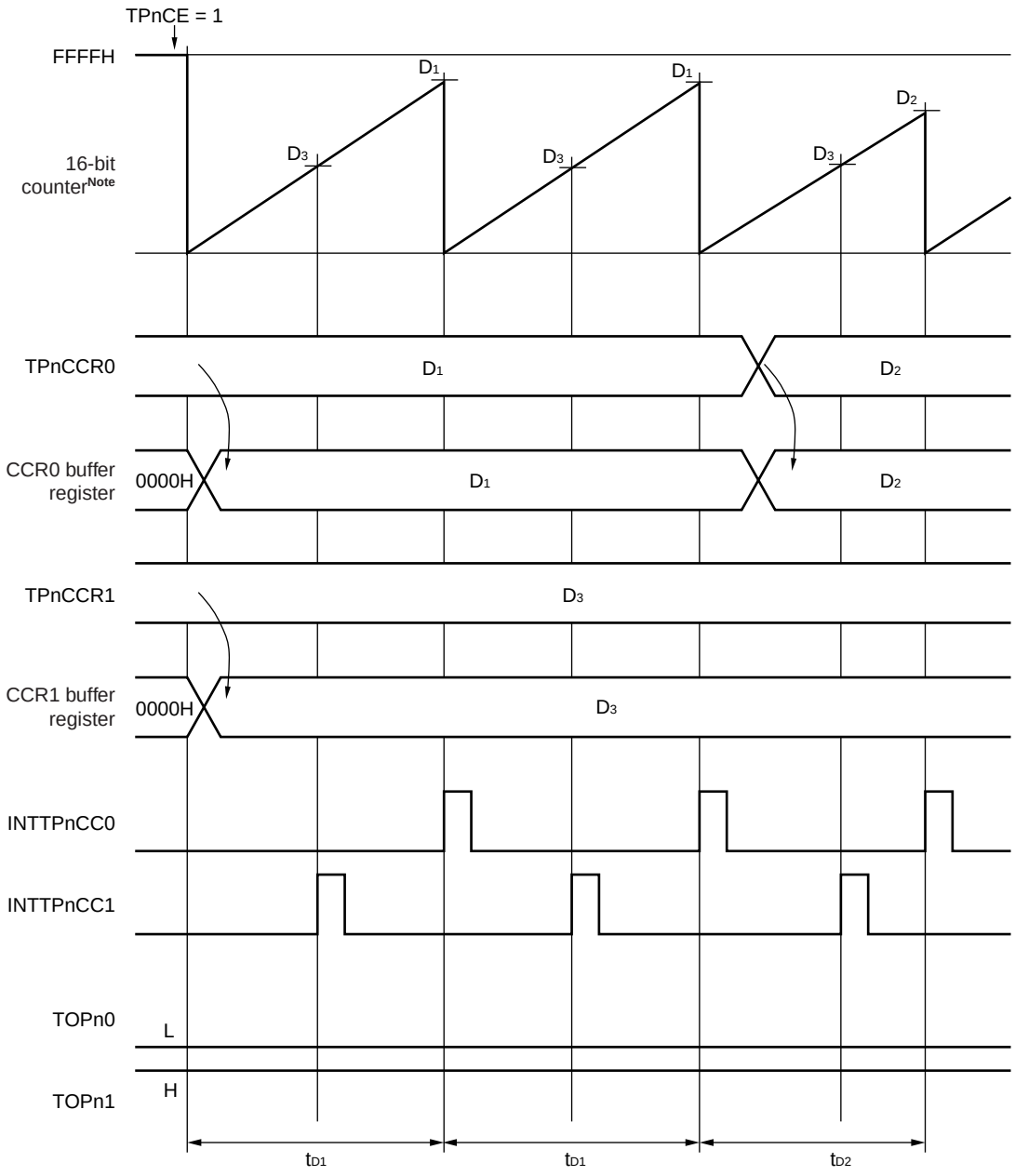


Note The 16-bit counter is not cleared when its value matches the value of TPnCCR1.

Remark n = 0 to 3

Figure 7-7. Timing of Basic Operation in Interval Timer Mode (1/2)

(a) When $D_1 > D_2 > D_3$, 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.

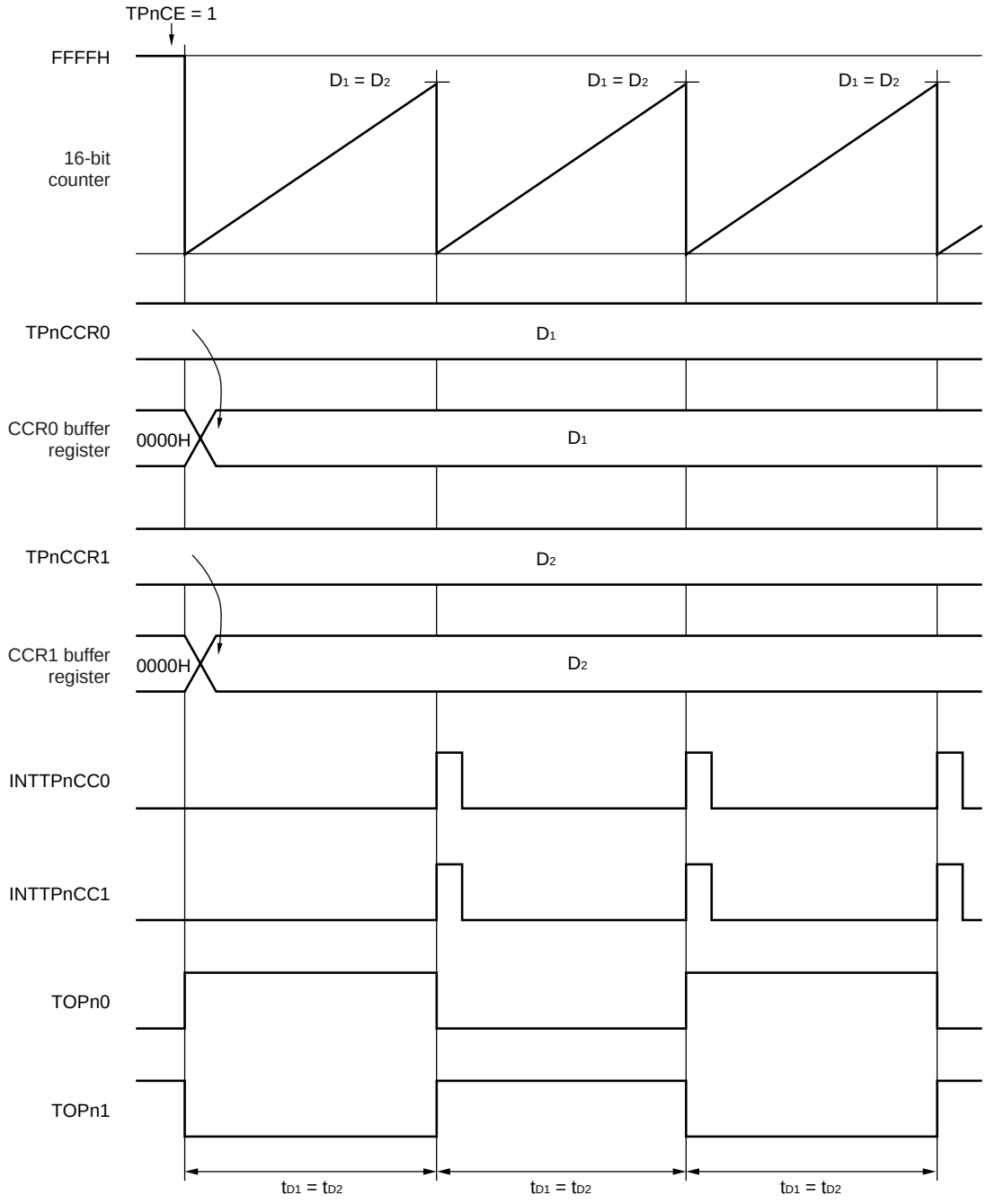
Remarks 1. D_1, D_2: Set value of TPnCCR0 register (0000H to FFFFH)
D_3: Set value of TPnCCR1 register (0000H to FFFFH)

2. Interval time (t_{Dn}) = (Dn + 1) × (Count clock cycle)

3. n = 0 to 3

Figure 7-7. Timing of Basic Operation in Interval Timer Mode (2/2)

(b) When $D_1 = D_2$, TPnCCR0 and TPnCCR1 are not rewritten, and TOPn0 and TOPn1 are output (TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)



Remarks 1. D_1: Set value of TPnCCR0 register (0000H to FFFFH)
D_2: Set 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 count 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 TPnCCRm register, it is transferred to the CCR0 buffer register, and is compared with the value of the 16-bit counter.

The 16-bit counter cannot be cleared by using the TPnCCR1 register.

However, the set value of the TPnCCR1 register is transferred to the CCR1 buffer register and is compared with the value of the 16-bit counter. As a result, an interrupt request (INTTPnCC1) is generated.

By setting the TPnOE1 bit to 1, a signal can be output from the TOPn1 pin.

Rewriting the TPnCCR0 register is enabled when TPnCE = 1. When the TPnCCR1 register is not used, it is recommended to set TPnCCR1 to FFFFH.

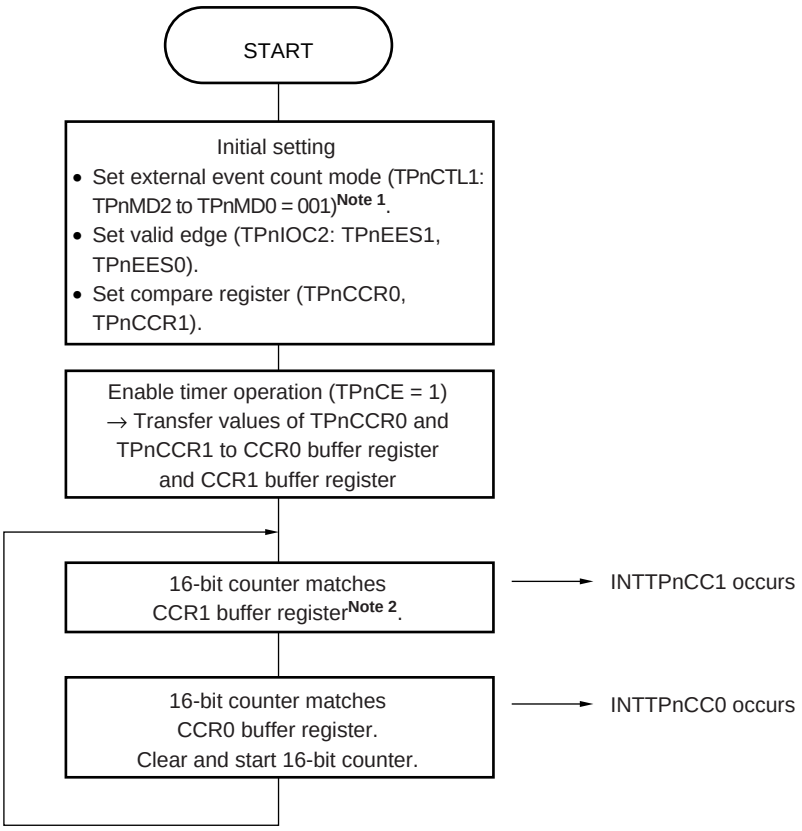
Remarks 1. Refer to 7.5.1 Anytime write and reload about write operation of TPnCCR0, TPnCCR1 during timer operation (TPnCE = 1).

2. n = 0 to 3

Caution 1. TOPn0 pin output in an external event count mode cannot be used. Set to TPnEEE = 1by interval timer mode (TPnMD2 to 0 = 000b) when TOPn0 pin output in an external event count mode is used.

2. In external event count mode, when TPnCCRm register value is set to 0000H the interrupt occurs after the overflow of the timer (FFFFH to 0000H)

Figure 7-8. Flowchart of Basic Operation in External Event Count Mode



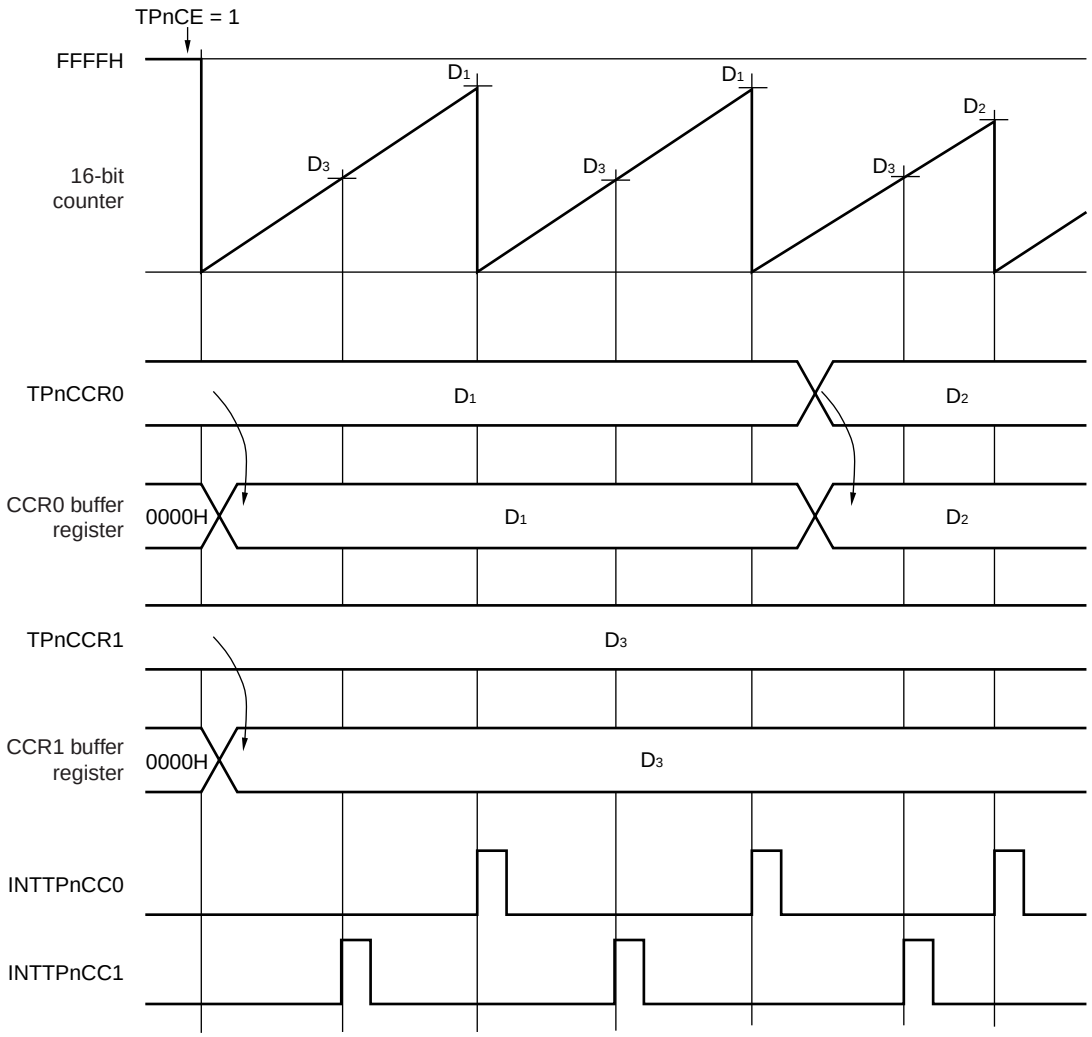
Notes 1. Selecting the TPnEEE bit has no effect.

2. The 16-bit counter is not cleared when it matches the CCR1 buffer register.

Remark n = 0 to 3

Figure 7-9. Timing of Basic Operation in External Event Count Mode (1/2)

(a) When $D_1 > D_2 > D_3$, only TPnCCR0 register value is rewritten, and TOPn1 is not output (TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 1)



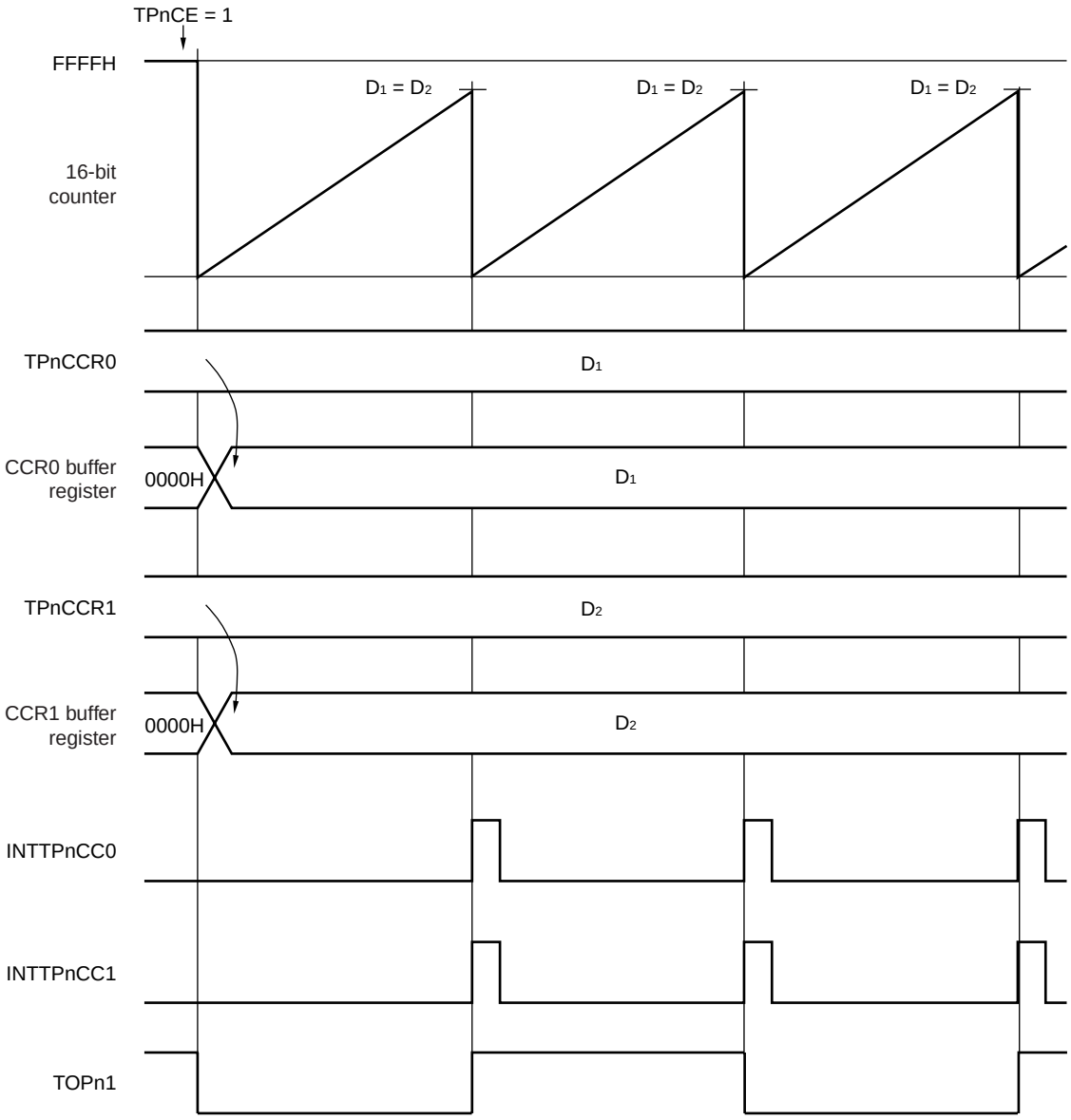
Remarks 1. D_1, D_2: Set value of TPnCCR0 register (0000H to FFFFH)
D_3: Set value of TPnCCR1 register (0000H to FFFFH)

2. Whenever times of (Set value of TPnCCRm register +1) are detected, the compare match interrupt is generated (m = 0, 1)

3. n = 0 to 3

Figure 7-9. Timing of Basic Operation in External Event Count Mode (2/2)

(b) When $D_1 = D_2$, TPnCCR0 and TPnCCR1 are not rewritten, and TOPn1 is output (TPnOE0 = 0, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)



Remarks 1. D_1: Set value of TPnCCR0 register (0000H to FFFFH)

D_2: Set value of TPnCCR1 register (0000H to FFFFH)

2. Whenever times of (Set value of TPnCCRm register +1) are detected, the compare match interrupt is generated (m = 0, 1)

3. n = 0 to 3

7.5.4 External trigger pulse output mode (TPnMD2 to TPnMD0 = 010)

When TPnCE = 1 in the external trigger pulse output mode, the 16-bit counter keeps at FFFFH and waits for input of an external trigger (input of TIPn0 pin or set of TPnEST bit). When the counter detects the trigger pulse 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).

In case of the software trigger mode, pulse of half cycle setting by TPnCCR0 register is outputted from TOPn0 terminal pin.

Rewriting the TPnCCR0 and TPnCCR1 registers is possible when TPnCE = 1.

To stop timer P, clear TPnCE to o. If the edge of the external trigger (input of TIPn0 pin or set TPnEST bit) is detected more than once in the external trigger pulse output mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. Then, TOPn0, TOPn1 terminal pin is initialized at the same time.

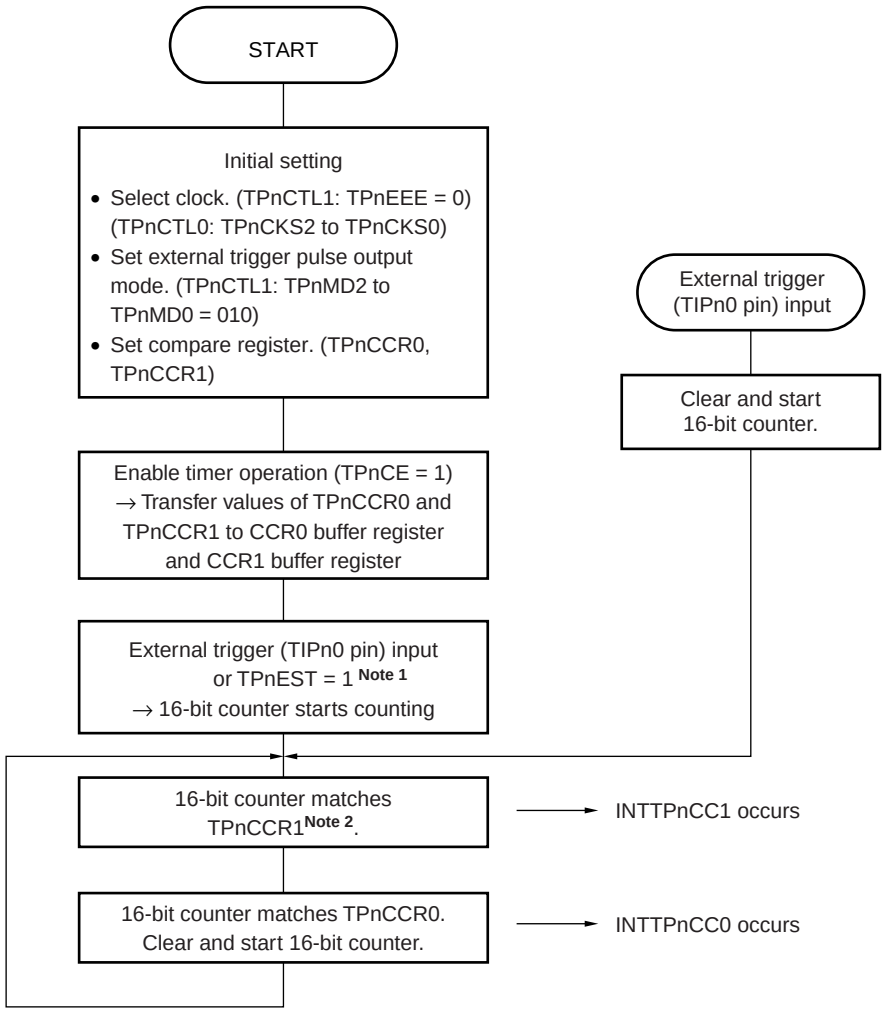
Caution 1. In the external trigger pulse output mode, select the internal clock (TPnEEE of TPnCTL1 register = 0) as the count clock.

2. In the external trigger pulse output mode, TPnCCR0 and TPnCCR1 registers are fixed as compare register. Therefore, capture function can not to use.

Remarks 1. For the reload operation when TPnCCR0 and TPnCCR1 are rewritten during timer operation, refer to **7.5.1 (2) Reload.**

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

Figure 7-10. Flowchart of Basic Operation in External Trigger Pulse Output Mode



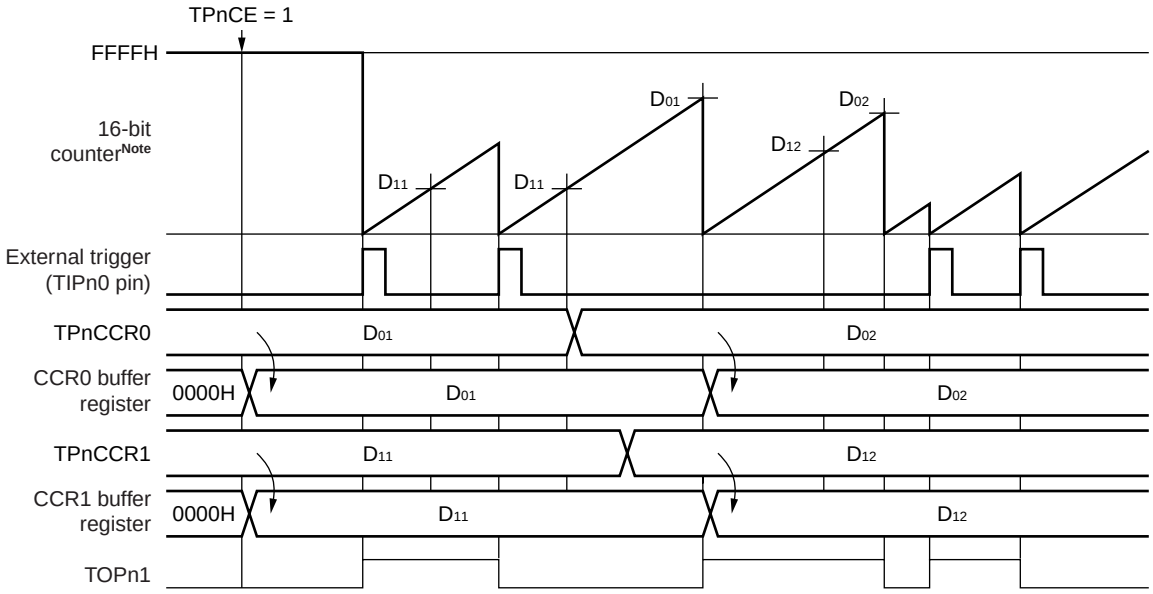
Notes: **1.** TPnEST bit of TPnCTL1 register can be written during timer operation (TPnCE = 1)

2. The 16-bit counter is not cleared when it matches the CCR1 buffer register.

Remark n = 0 to 3

Figure 7-11. Timing of Basic Operation in External Trigger Pulse Output Mode

(TPnOE0 = 0, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)



Note The 16-bit counter is not cleared when it matches the CCR1 buffer register.

Remarks 1. D_{01}, D_{02}: Set value of TPnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (Set value of TPnCCR1 register) / (Set value of TP0CCR0 register +1)
Cycle TOPn1 output = (Set value of TPnCCR0 +1) × (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. The waveform of the one-shot pulse is output from the TOPn1 pin. The TOPn0 pin produces an active level output during counting by timer counter. Active level is set by TPnCL0 register.

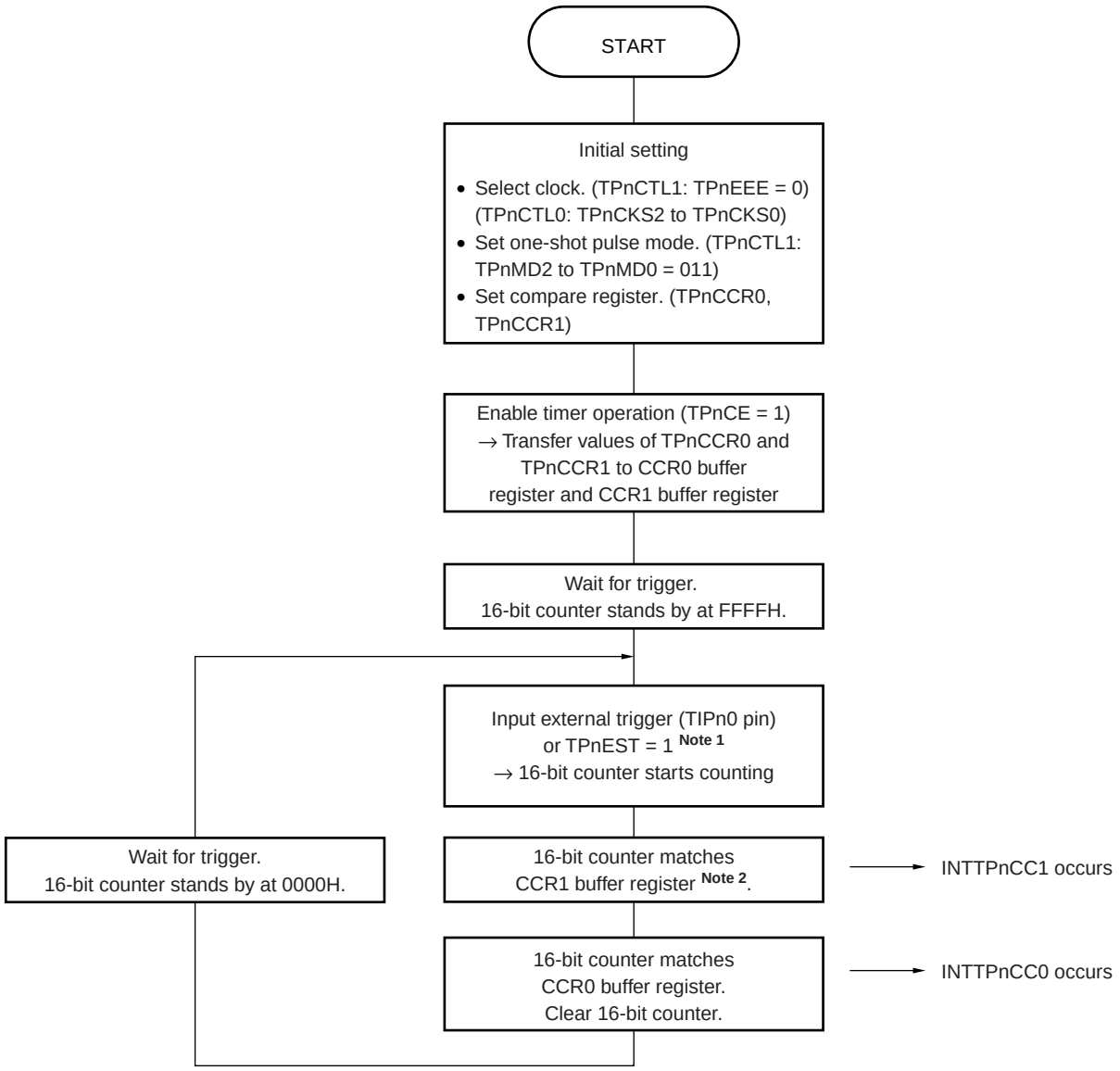
Cautions:

1. **Select the internal clock (TPnEEE of the TPnCTL1 register = 0) as the count clock in the one-shot pulse mode.**
2. **In the one-shot pulse mode, TPnCCR0 and TPnCCR1 registers are fixed as compare register. Therefore, capture function can not to use.**
3. **In the one-shot pulse mode, when setting value of TPnCCR1 register is bigger than setting value of TPnCCR0 register, on-shot pulse is not outputted.**

Remarks

1. In the one-shot pulse mode, TPnCCR0 and TPnCCR1 are rewritten during timer operation (TPnCE=1). During timer operation (TPnCE=1), for anytime write operation when rewriting of TPnCCR0 and TPnCCR1, refer to **7.5.1 (1) Anytime write.**
2. n = 0 to 3

Figure 7-12. Flowchart of Basic Operation in One-Shot Pulse Mode



Notes: **1.** Only TPnEST bit of TPnCTL1 register can be written during timer operation (TPnCE = 1).

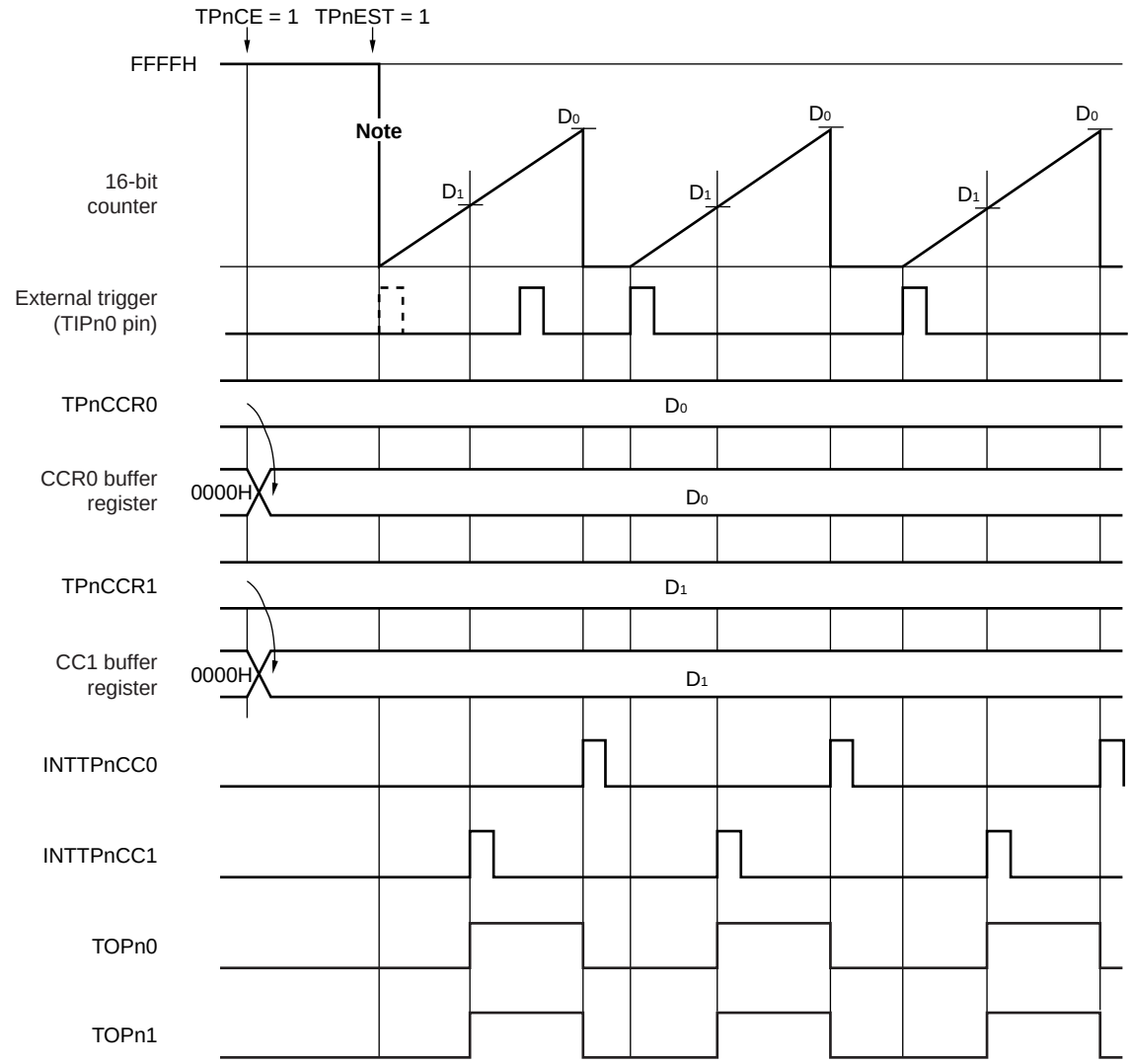
2. The 16-bit counter is not cleared when it matches the CCR1 buffer register.

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 3

Figure 7-13. Timing of Basic Operation in One-Shot Pulse Mode

(TPnOE0 = 0, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Note The 16-bit counter starts counting up either when TPnEST = 1 or when TIPn0 is input.

Remarks 1. D_0: Set value of TPnCCR0 register (0000H to FFFFH)
D_1: Set value of TPnCCR1 register (0000H to FFFFH)

2. n = 0 to 3

3. 3. The active level term of TOPn1 pin output is: (Set value of TPnCCR0 - Set value of TPnCCR1 +1) × Count clock cycle
Time of output delay = (Set value of TPnCCR1register) × Count clock cycle

7.5.6 PWM mode (TPnMD2 to TPnMD0 = 100)

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.

To stop timer P, clear TPnCE to 0. The waveform of PWM is output from the TOPn1 pin. The TOPn0 pin produces a pulse of half the PWM cycle.

The TPnCCR0 and TPnCCR1 registers cannot be used as capture registers.

Remark n = 0 to 3

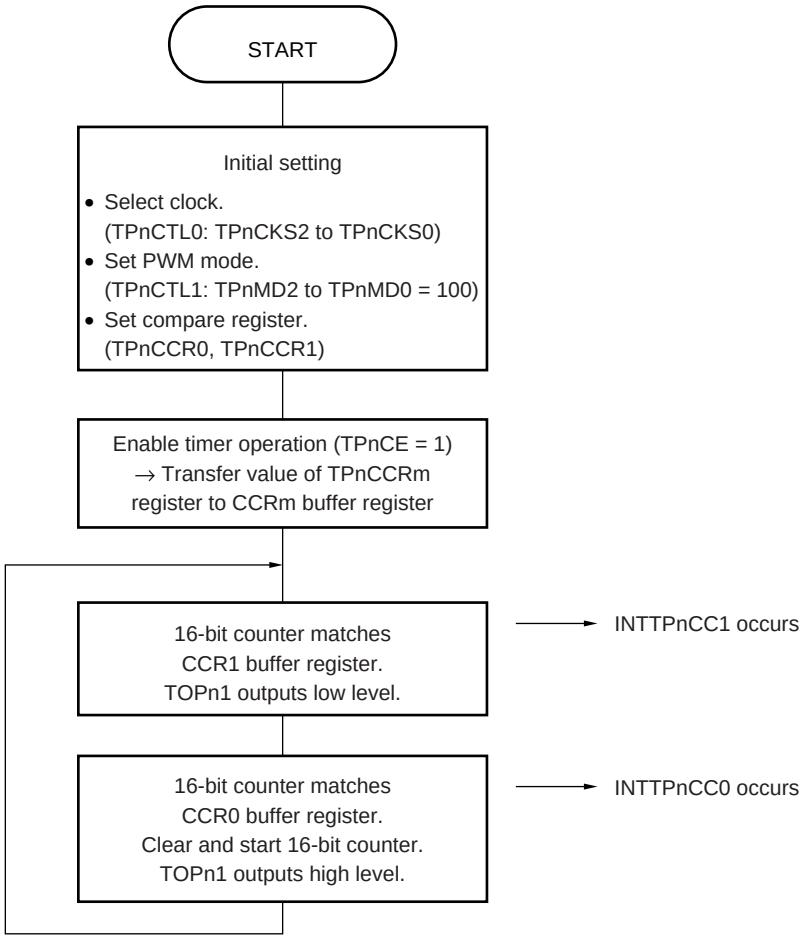
For the reload operation when TPnCCR0 and TPnCCR1 are rewritten during timer operation (TPnCE1=1), refer to **7.5.1 (2) Reload**.

Caution In the PWM mode, TPnCCR0 and TPnCCR1 registers are fixed as compare register. Therefore, capture function can not to use.

(1) Operation flowchart of PWM mode

Figure 7-14. Flowchart of Basic Operation in PWM Mode (1/2)

(a) When values of TPnCCR0 and TPnCCR1 registers are not rewritten during timer operation



Remark n = 0 to 3
m = 0, 1

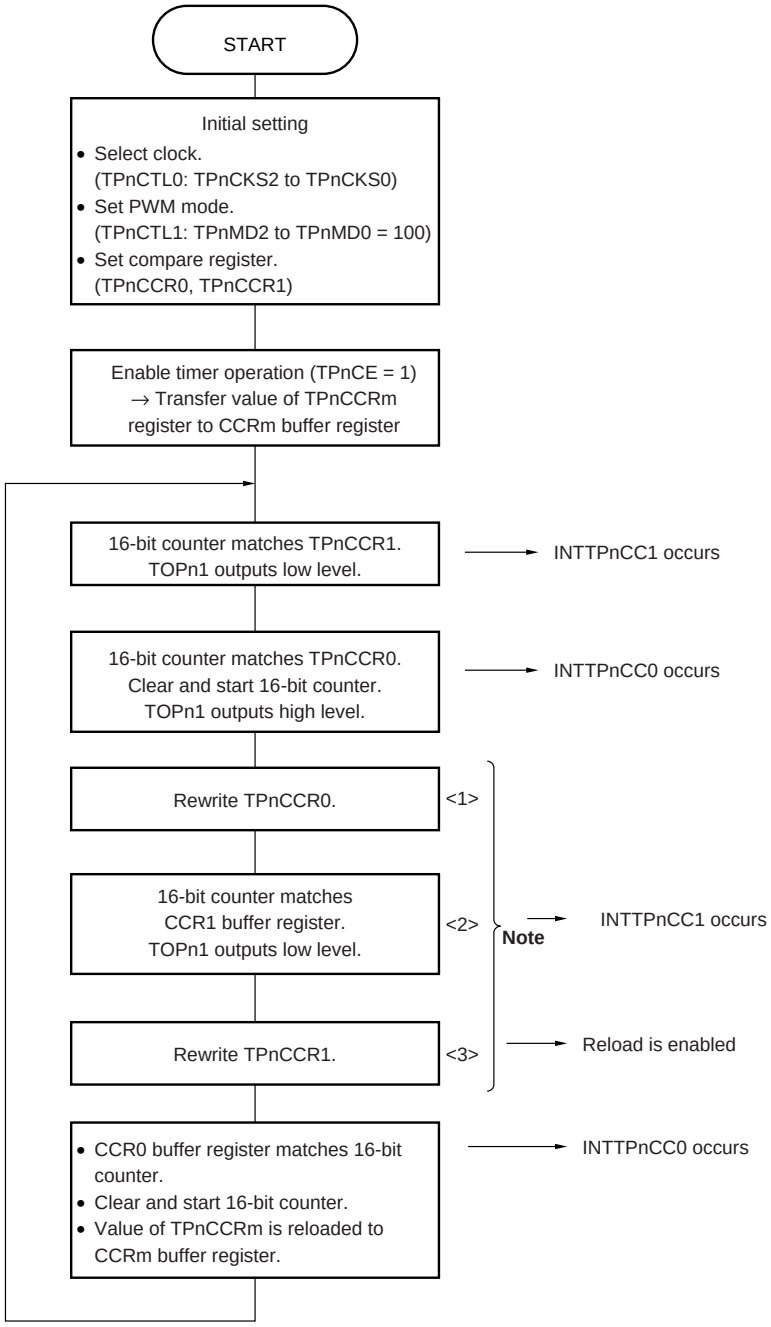
(2) Operation flowchart of PWM mode

(a) Change of pulse width during operation

When change of PWM waveform during operation, please write to TPnCCR1 register at last. After write to TPnCCR1 register, when write to TPnCCR0 register again, please rewrite after detection of INTTPnCC1 signal.

Figure 7-14. Flowchart of Basic Operation in PWM Mode (2/2)

(b) When values of TPnCCR0 and TPnCCR1 registers are rewritten during timer operation



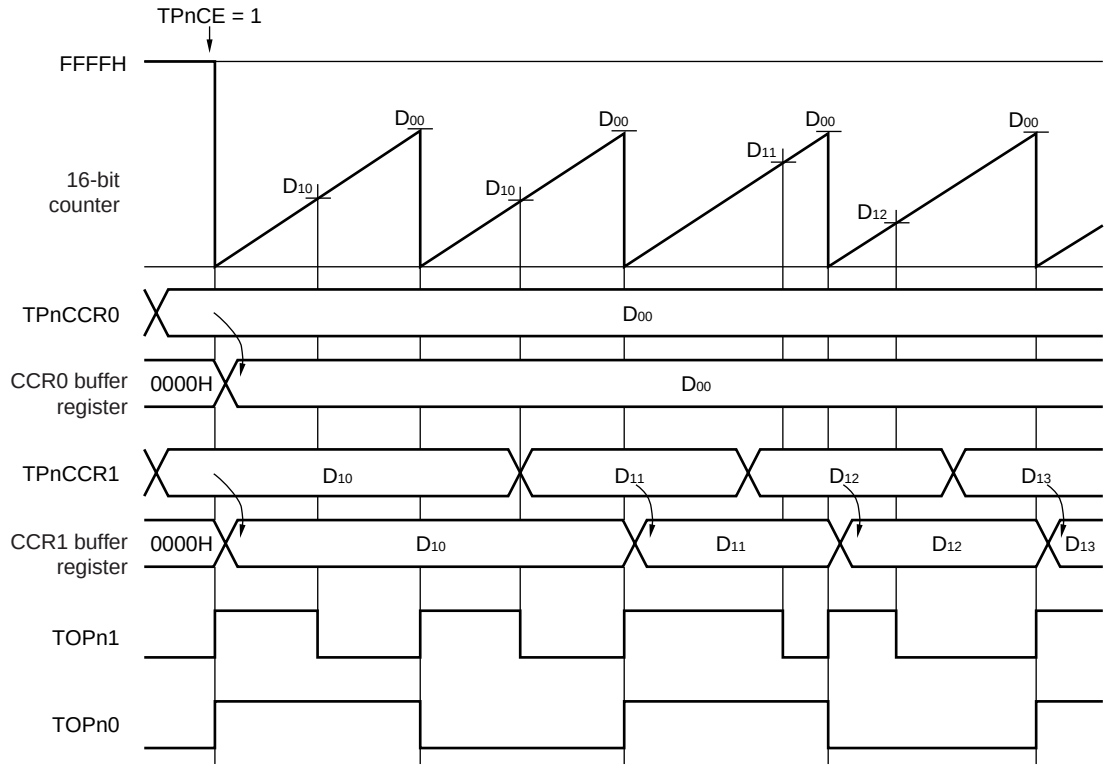
Note The timing of <2> may differ depending on the rewrite timing of <1> and <3> and the value of TPnCCR1, but make sure that <3> comes after <1>.

Remark n = 0 to 3
m = 0, 1

Figure 7-15. Timing of Basic Operation in PWM Mode (1/2)

(a) When rewriting value of TPnCCR1

(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks 1. D_{00}: Set value of TPnCCR0 register (0000H to FFFFH)

D_{10}, D_{11}, D_{12}, D_{13}: Set value of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (set value of TPnCCR1 register +1)/(Set value of TP0CCR0 register)

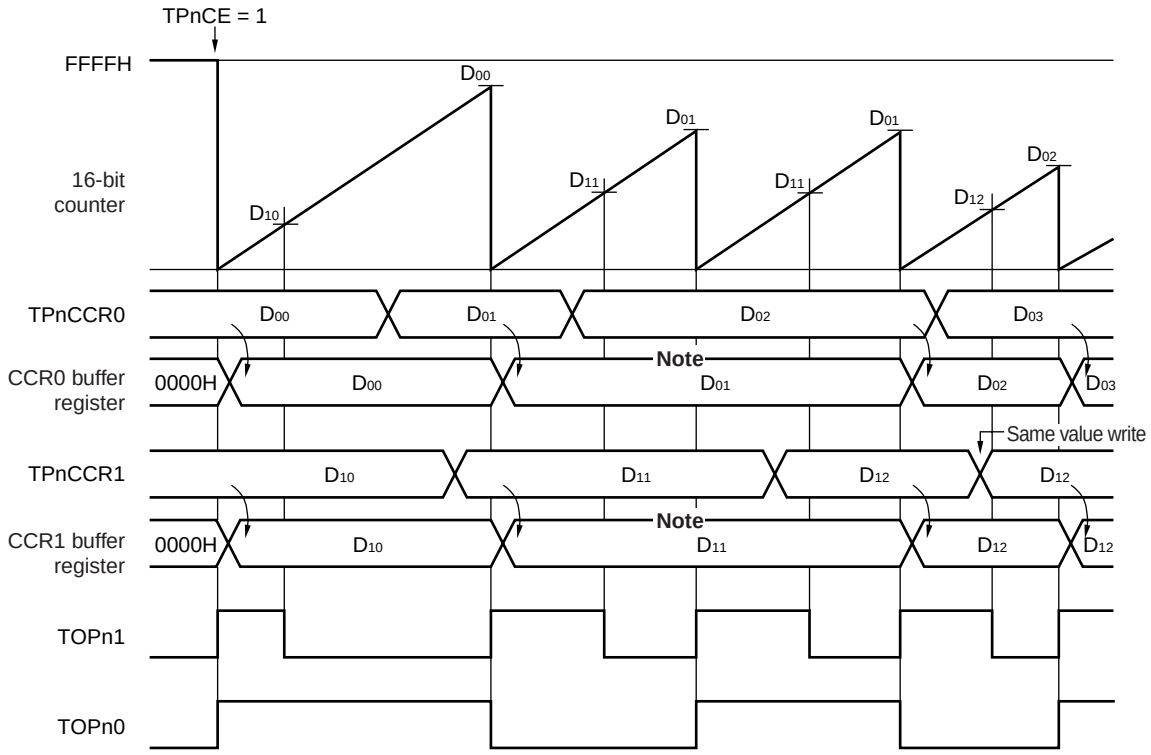
Cycle of TOPn1 output = (Set value of TPnCCR0 register +1)×(Count clock cycle)

Toggle width of TOPn0 output = (Set value of TPnCCR0 register + 1) × (Count clock period)

3. n = 0 to 3

Figure 7-15. Timing of Basic Operation in PWM Mode (2/2)

(b) When rewriting values of TPnCCR0 and TPnCCR1
(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Note No value is reloaded because the TPnCCR1 register is not rewritten.

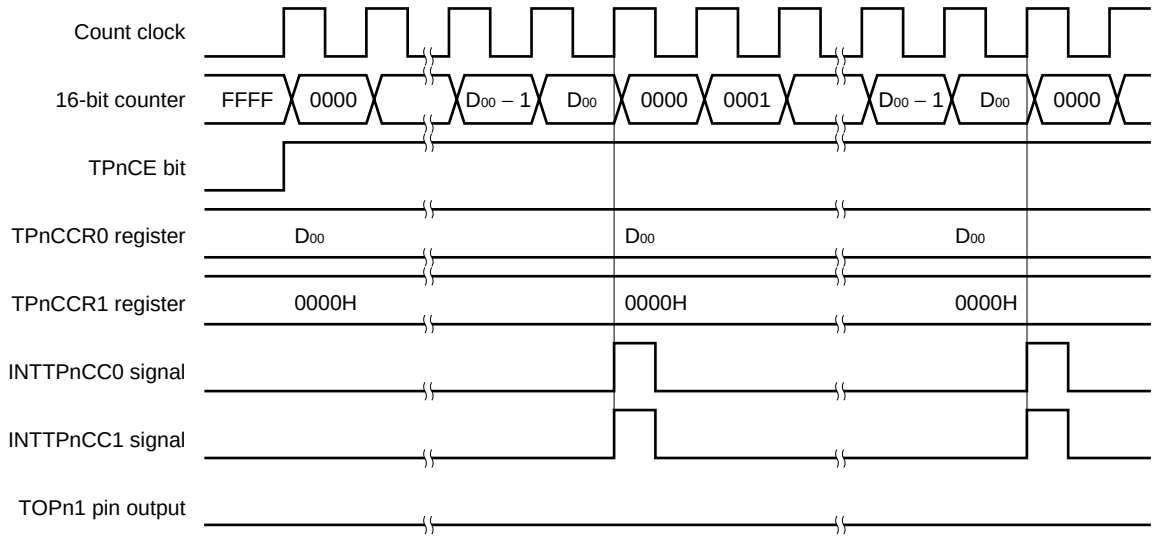
Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Set value of TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Set value of TPnCCR1 register (0000H to FFFFH)

2. Duty of TOPn1 output = (set value of TPnCCR1 register +1)/(Set value of TP0CCR0 register)
Cycle of TOPn1 output = (Set value of TPnCCR0 register +1)×(Count clock cycle)
Toggle width of TOPn0 output = (Set value of TPnCCR0 register + 1) × (Count clock cycle)

3. n = 0 to 3

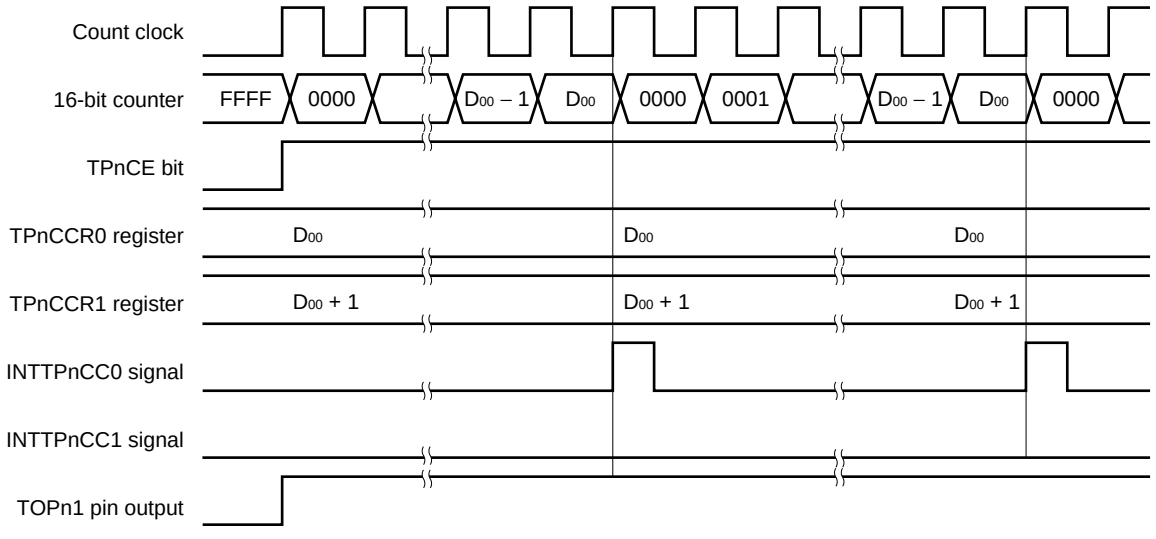
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TPnCCR1 register to 0000H. If the set value of the TPnCCR0 register is FFFFH, the INTTPnCC1 signal is generated periodically.



Remark n = 0 to 8

To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 8

7.5.7 Free-running mode (TPnMD2 to TPnMD0 = 101)

In the free-running mode, the 16-bit counter free-runs, and the bit that selects the capture or compare register function can be select by the setting of the TPnCCS1 and TPnCCS0 bits.

Setting of the TPnCCS1 and TPnCCS0 bits of the TPnOPT0 register is valid only in the free-running mode.

TPnCCS1	Operation
0	TPnCCR1 register is used as compare register.
1	TPnCCR1 register is used as capture register.

TPnCCS0	Operation
0	TPnCCR0 register is used as compare register.
1	TPnCCR0 register is used as capture register.

- When TPnCCR1 register is used as compare register

  When the value of the 16-bit counter matches the value of the CCR0 buffer register in the free-running mode, an interrupt is generated.

  TPnCCR1 register is enabled for write operation when TPnCE=1. Any data is set to TPnCCR1 register by anytime write, data is translated to CCR1 buffer register, and data become comparison value with value of the 16 bit counter.

  If timer output (TOPn0) is enabled, TOPn0 produces a toggle output when the value of the 16-bit counter matches the value of the CCR0 buffer register.

- When TPnCCR1 register is used as capture register

  The value of the 16-bit counter is stored in the TPnCCR1 register when the edge of the TIPn1 pin is detected.

- When TPnCCR0 register is used as compare register

  When the value of the 16-bit counter matches the value of the CCR1 buffer register in the free-running mode, an interrupt is generated.

  TPnCCR0 register is enabled for write operation when TPnCE=1. Any data is set to TPnCCR0 register by anytime write, data is translated to CCR1 buffer register, and data become comparison value with value of the 16 bit counter.

  If timer output (TOPn0) is enabled, TOPn0 produces a toggle output when the value of the 16-bit counter matches the value of the CCR0 buffer register.

- When TPnCCR0 register is used as capture register

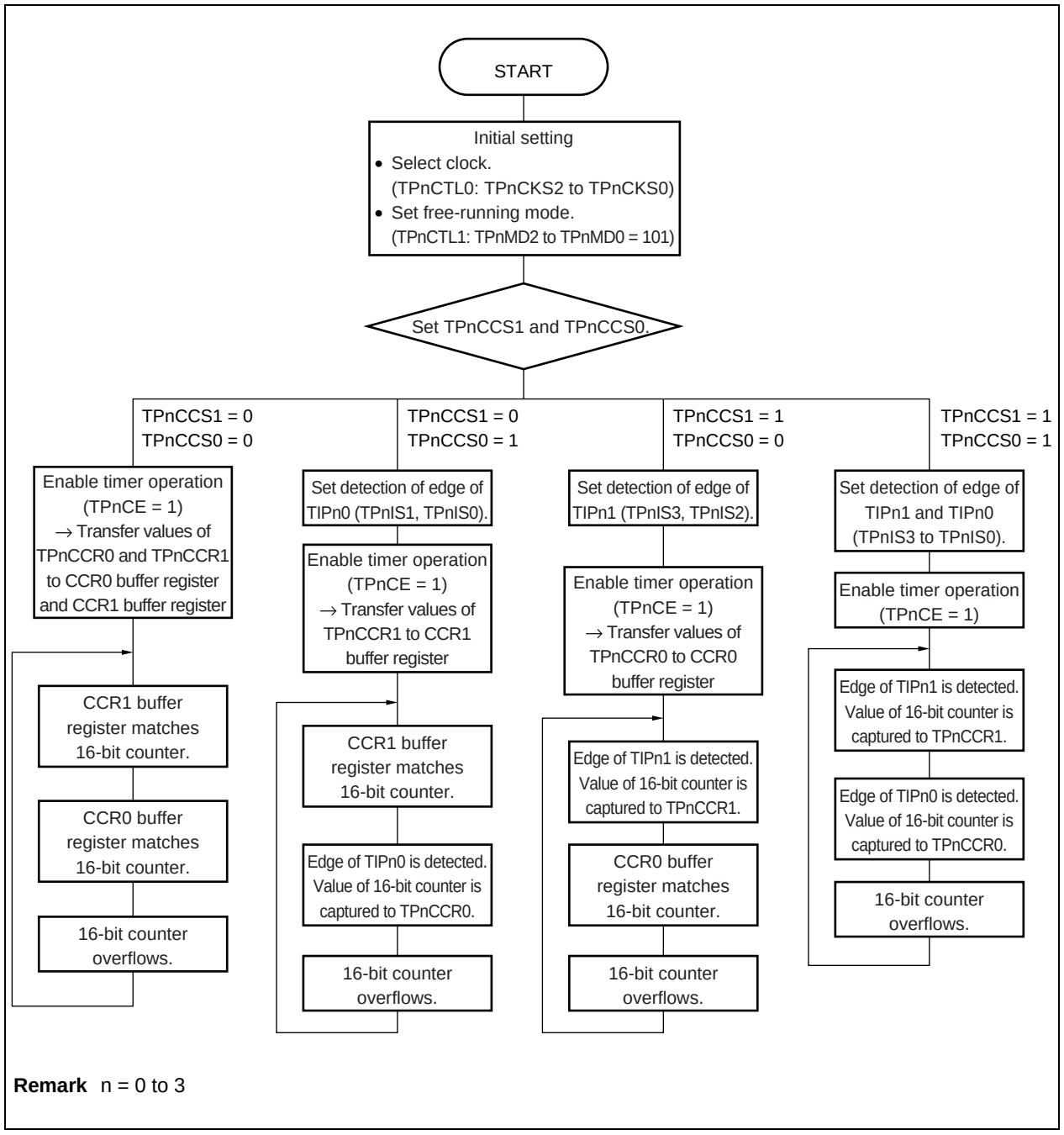
  The value of the 16-bit counter is stored in the TPnCCR0 register when the edge of the TIPn0 pin is detected.

Caution: External event count input as count clock (TPnCTRL.TPnEEE=1), TPnCCR0 register can not to use as capture register.

Remark: Using TPnCCR0 and TPnCCR1 register as a compare register, the written operation at timer operation (TPnCE = 1) refer to 7. 5. 1 (1) Anytime write.

Caution: At free running mode, count clear operation is not used by compare register matches.

Figure 7-16. Flowchart of Basic Operation in Free-Running Mode



Remark n = 0 to 3

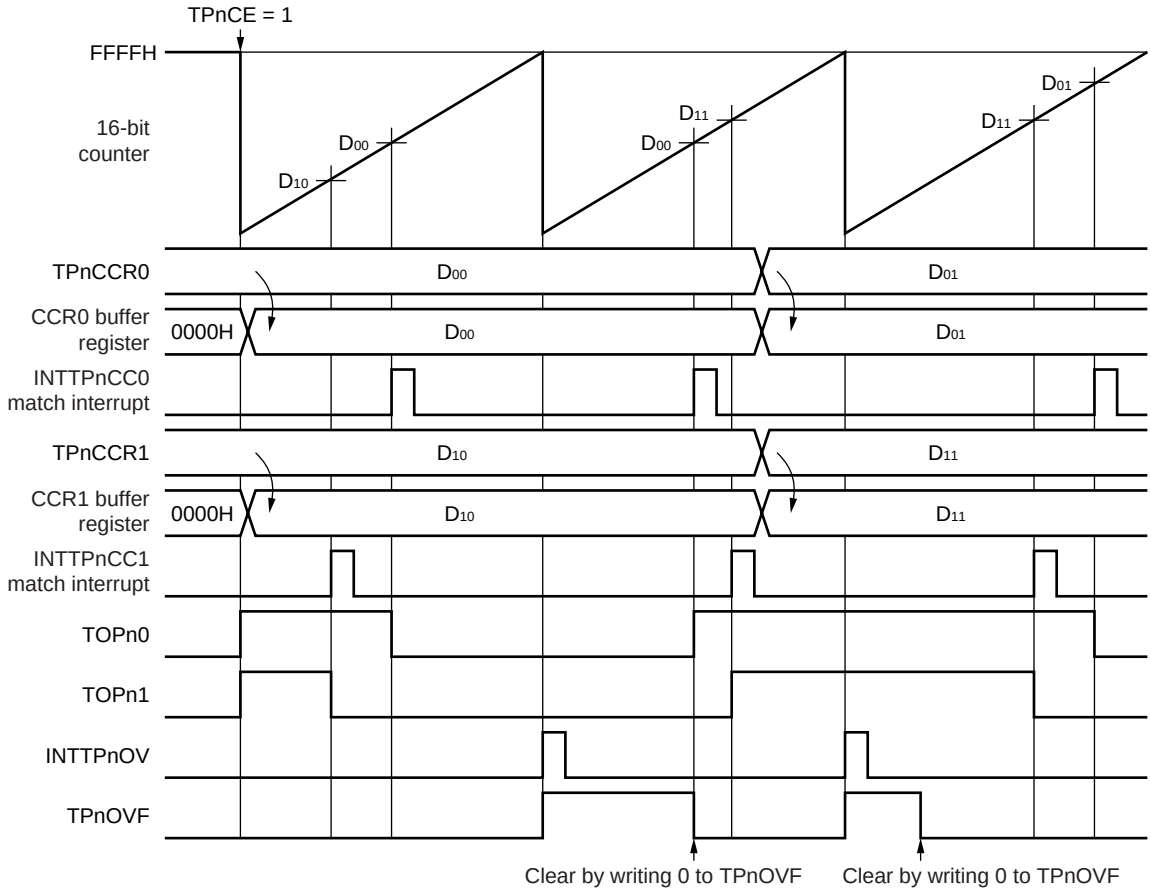
(1) When TPnCCS1 = 0 and TPnCCS0 = 0 (compare function)

When TPnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH, and continues counting up in the free-running mode until TPnCE is cleared to 0. If a value is written to the TPnCCR0 and TPnCCR1 registers in this mode, it is transferred to the CCR0 and CCR1 buffer registers (anytime write). Even if a one-shot pulse trigger is input in this mode, a one-shot pulse is not generated. If TPnOEm is set to 1, TOPnm produces a toggle output when the value of the 16-bit counter matches the value of the CCRm buffer register.

Remark n = 0 to 3
m = 0, 1

Figure 7-17. Timing of Basic Operation in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 0)

(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks 1. D_{00}, D_{01}: Set value of TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Set value of TPnCCR1 register (0000H to FFFFH)

2. Toggle width of TOPn0 output = (Set value of TPnCCR0 register) × (Count clock cycle)
Toggle width of TOPn1 output = (Set value of TPnCCR1 register)

3. TOPnm output goes high when counting is started.

4. n = 0 to 3
m = 0, 1

(2) When TPnCCS1 = 1 and TPnCCS0 = 1 (capture function)

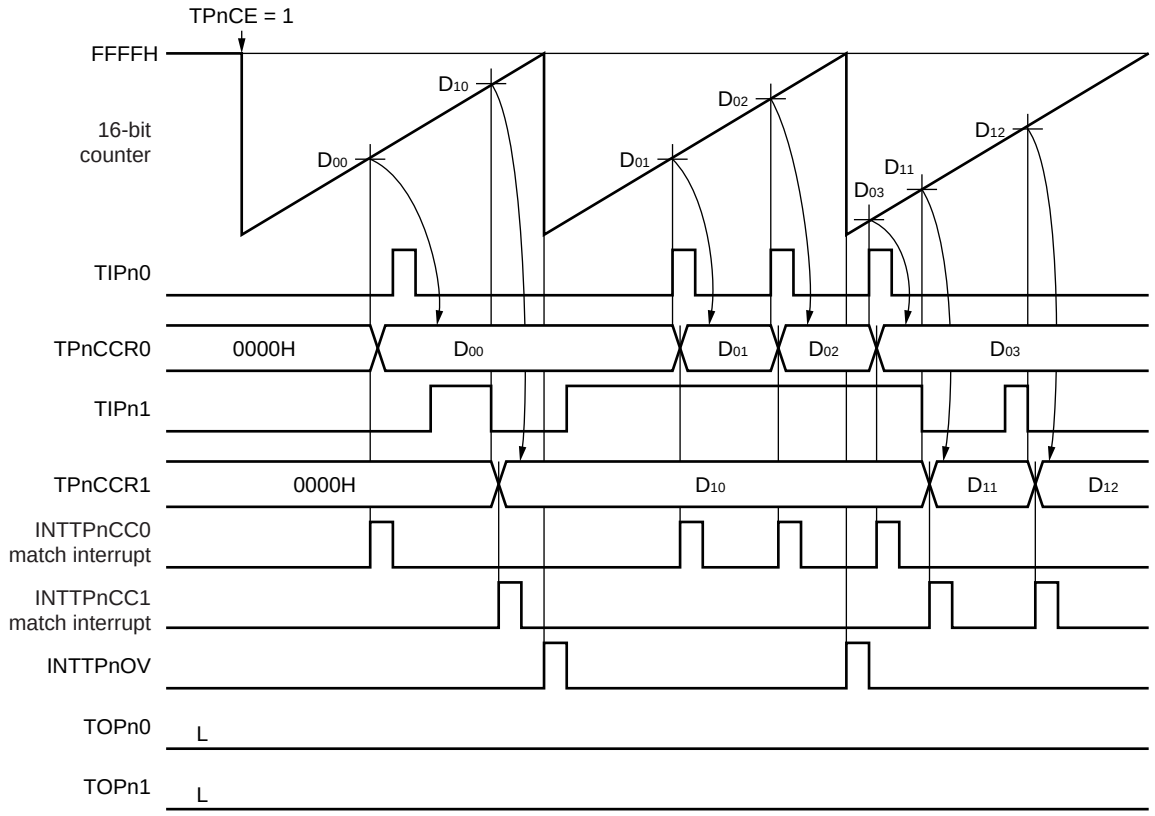
When TPnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH, and continues counting up in the free-running mode until TPnCE is cleared to 0. The value captured by the capture trigger is written to the TPnCCR0 and TPnCCR1 registers.

Capturing close to an overflow (FFFFH) is judged using the overflow flag (TPnOVF).

However, if the interval of the capture trigger is such that the overflow occurs twice (two or more cycles of free-running); the TPnOVF flag cannot be used for judgment.

Figure 7-18. Timing of Basic Operation in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 1)

(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}: Value captured to TPnCCR1 register (0000H to FFFFH)

2. TIPn0: Rising edge is detected (TPnIS1, TPnIS0 = 01).
TIPn1: Falling edge is detected (TPnIS3, TPnIS2 = 10).

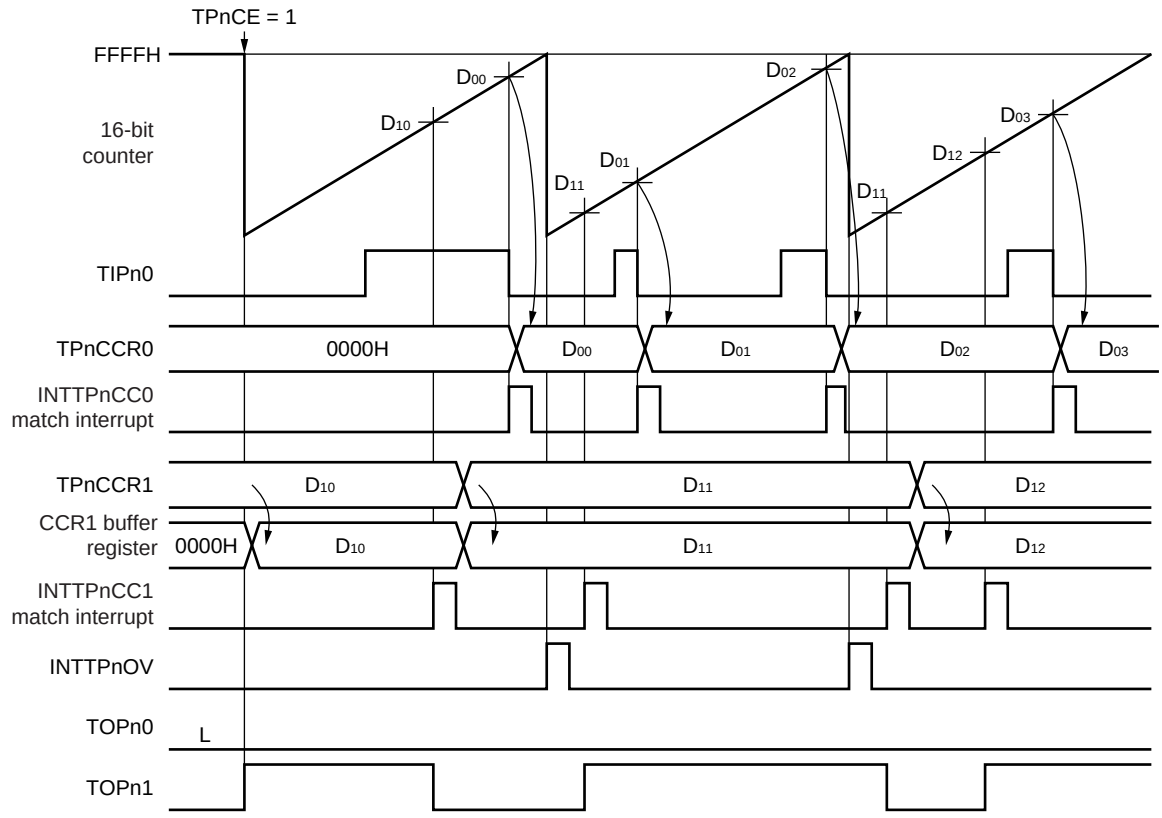
3. n = 0 to 3

(3) When TPnCCS1 = 0 and TPnCCS0 = 1

When TPnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH, and continues counting up in the free-running mode until TPnCE is cleared to 0. The TPnCCR1 register is used as a compare register. As an interval function, an interrupt signal is output when the value of the 16-bit counter matches the set value of the TPnCCR1 register. If TPnOE1 is set to 1, TOPn1 produces a toggle output when the value of the 16-bit counter matches the set value of the TPnCCR1 register.

Figure 7-19. Timing of Basic Operation in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 1)

(TPnOE0 = 1, TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 0)



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}: Value captured to TPnCCR1 register (0000H to FFFFH)

2. TIPn0: Falling edge is detected (TPnIS1, TPnIS0 = 10).

3. n = 0 to 3

(4) 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 generated.

After generation of the overflow interrupt (INTTPnOV), be sure to check if the overflow flag (TPnOVF) is set to 1.

The overflow flag is cleared by the CPU by writing 0 to it.

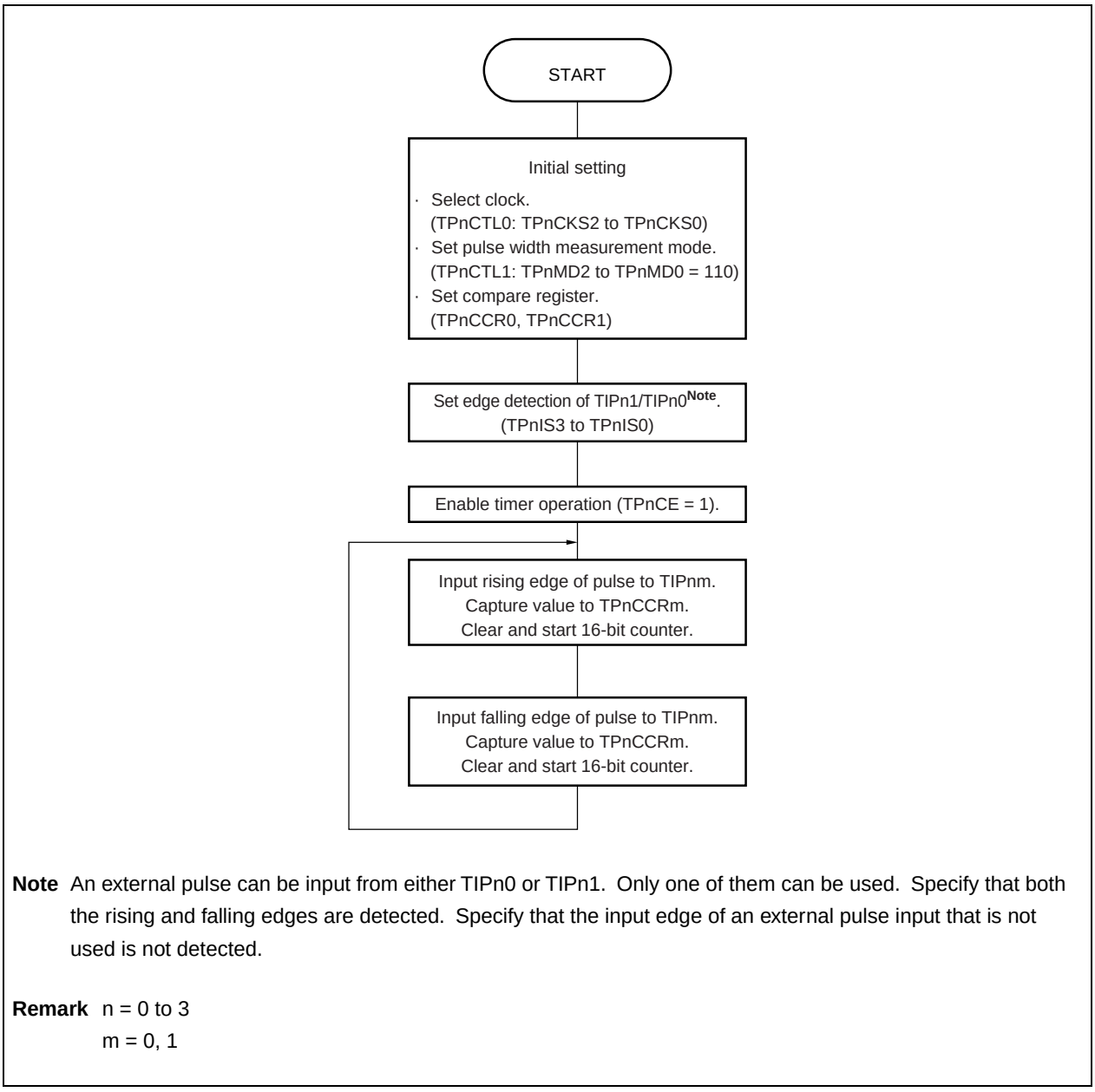
7.5.8 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110)

In the pulse width measurement mode, free-running counting is performed. The value of the 16-bit counter is captured to capture register 0 (TPnCCR0) when both the rising and falling edges of the TIPn0 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. For measurement a pulse width that causes overflow to occur twice or more, please count number with overflow interrupt, etc. When the edge of the TIPn1 pin is detected, the value of the 16-bit counter is stored in capture register 1 (TPnCCR1), and the 16-bit counter is cleared.

Caution In the pulse width measurement mode, select the internal clock (TPnEEE of the TPnCTL1 register = 0) as the count clock.

Figure 7-20. Flowchart of Basic Operation in Pulse Width Measurement Mode

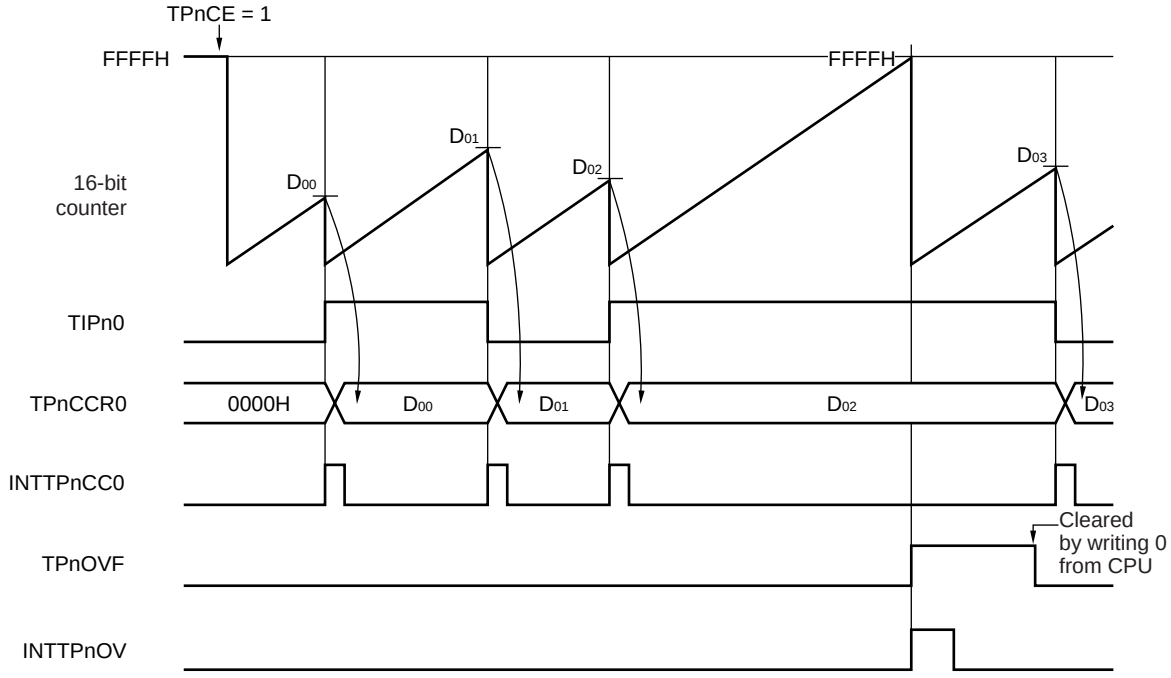


Note An external pulse can be input from either TIPn0 or TIPn1. Only one of them can be used. Specify that both the rising and falling edges are detected. Specify that the input edge of an external pulse input that is not used is not detected.

Remark n = 0 to 3
m = 0, 1

Figure 7-21. Timing of Basic Operation in Pulse Width Measurement Mode

(TPnOE0 = 0, TPnOE1 = 0, TPnOL0 = 0, TPnOL1 = 0)



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TPnCCR0 register (0000H to FFFFH)

2. TIPn0: Both the rising and falling edges are detected (TPnIS1, TPnIS0 = 11).

3. n = 0 to 34.

4. Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIPnm pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTPnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TPnOVF bit set (1) count + Captured value) × Count clock cycle

7.6 Timer Synchronized 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.

Table 7-3. Tuned Operation Mode of Timers

Master Timer	Slave Timer	
TMP0	TMP1	–
TMP2	TMP3	TMQ0
TMQ1	TMQ2	–

Cautions 1. The tuned operation mode is enabled or disabled by the TPmSYE bit of the TPmCTL1 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.

<1> Set the TPmSYE bit of the TPmCTL1 register and the TQnSYE bit of the TQnCTL1 register of the slave timer to enable the tuned operation.
Set the TPmMD2 to TPmMD0 bits of the TPmCTL1 register and TQnMD2 to TQnMD0 bits of the TQnCTL1 register of the slave timer to the free-running mode.

<2> 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.

<3> Set the compare register value of the master and slave timers.

<4> Set the TPmCE bit of the TPmCTL0 register and the TQnCE bit of the TQnCTL0 register of the slave timer to enable operation on the internal operating clock.

<5> 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.

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

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	PPG	←	Toggle	PWM	N/A	←
		TOP11	PPG	←	PWM	←	N/A	←
Ch1	TMP2 (master)	TOP20	PPG	←	Toggle	PWM	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	←
	TMQ2 (slave)	TOQ20	PPG	←	Toggle	←	Toggle	Triangular wave PWM
		TOQ21 to TOQ23	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 2)

Figure 7-22. Tuned Operation Image (TMP2, TMP3, TMQ0)

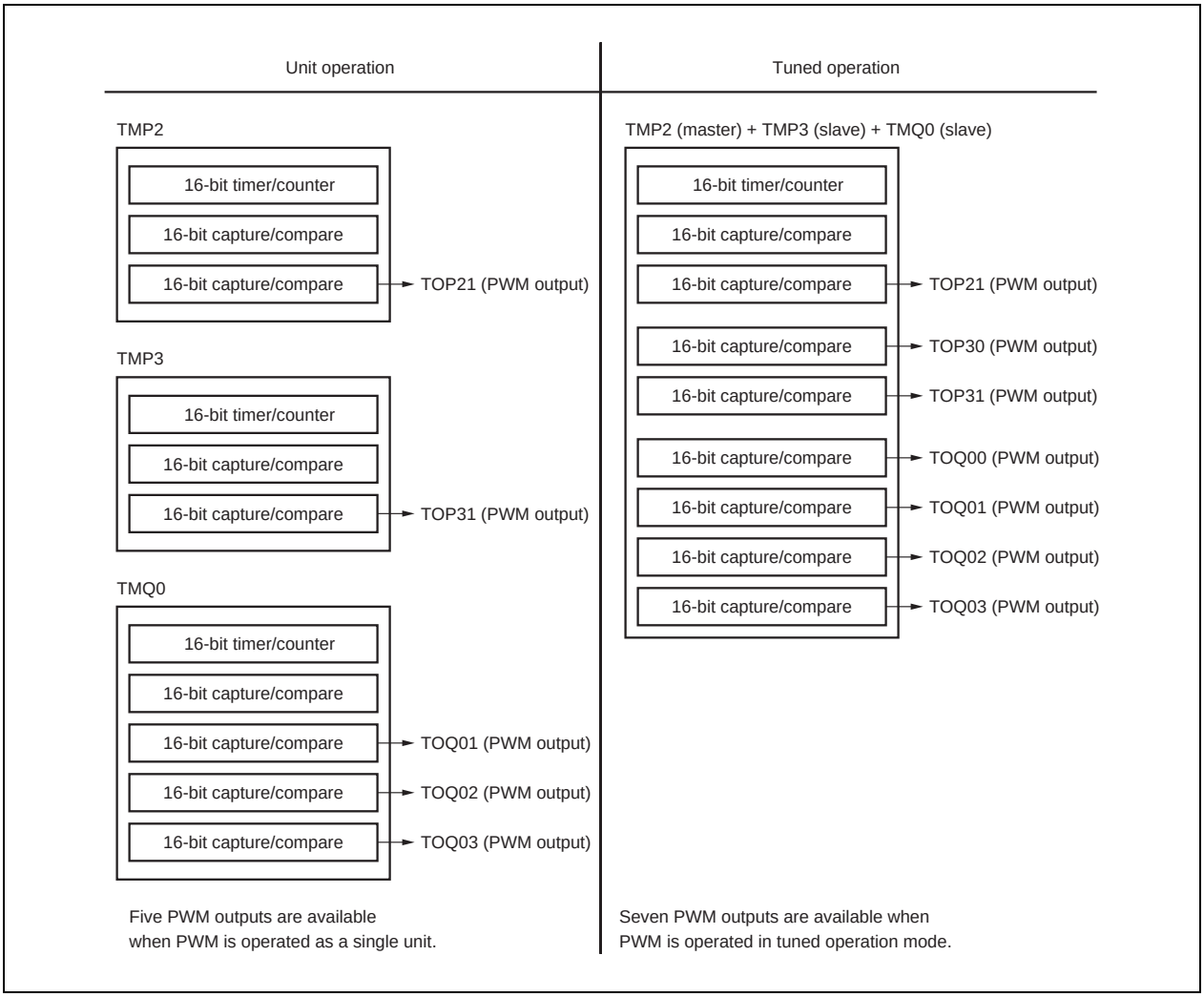
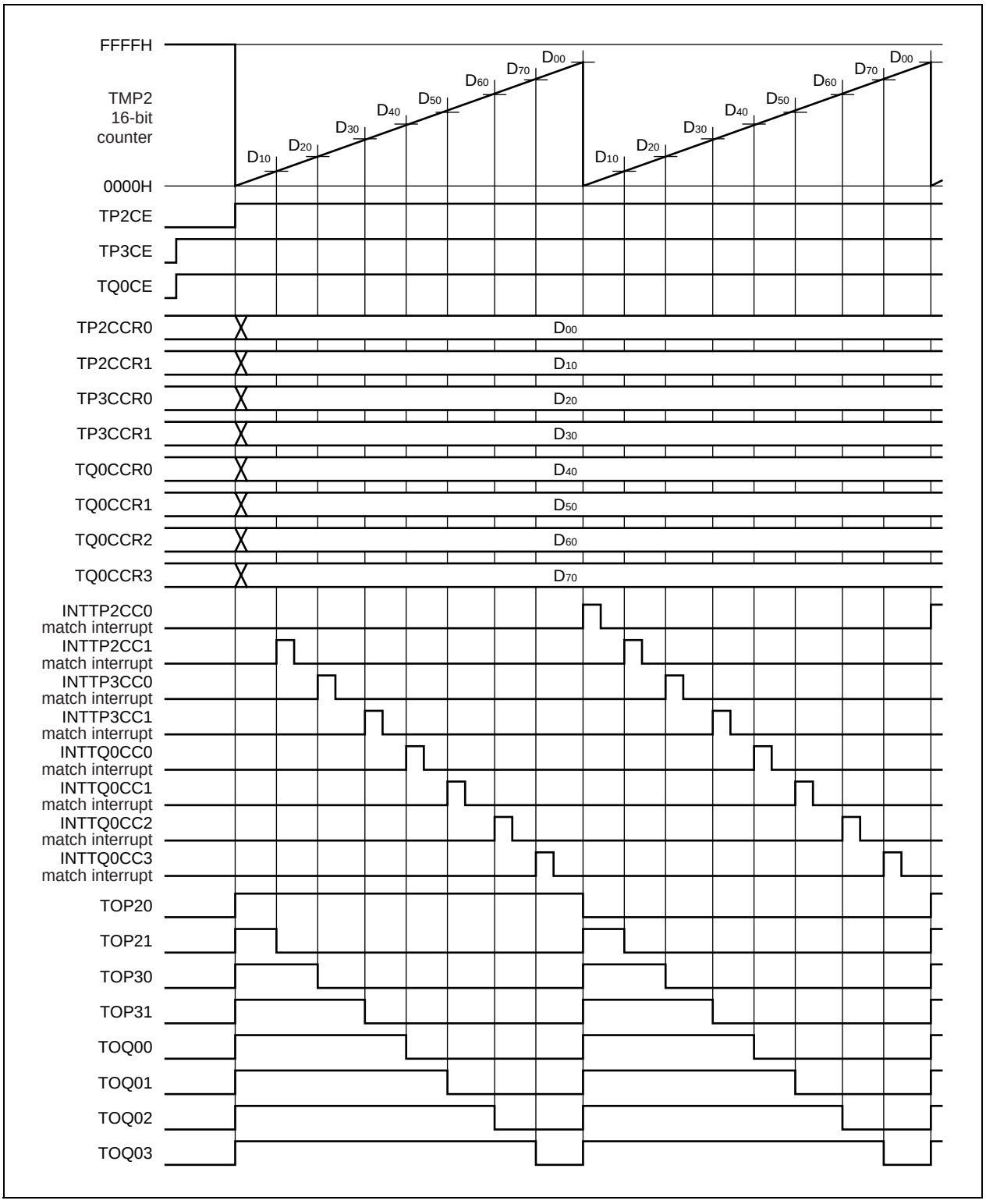


Figure 7-23. Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)



7.7 Selector Function

In the V850ES/Fx2, the TIP input/RXDA input and the TIP input/TSOUT signal can be used to select the capture trigger input of TMP.

By using this function, the following is possible.

- The TIP00 and TIP01 input signals can be selected from the port/timer alternate-function pins (TIP00 and TIP01 pins) and the TSOUT signal of the CAN controller.
  → If the TSOUT signal of CAN0 or CAN1 is selected, the time stamp function of the CAN controller can be used.
- The TIP10 and TIP11 input signals of TMP1 can be selected from the port/timer alternate-function pins (TIP10 and TIP11 pins) and the UARTA reception alternate-function pins (RXDA0 and RXDA1). The TIP30 and TIP31 input signals of TMP3 can be selected from a port/timer alternate-function pin (TIP30 and TIP31 pins) and the UARTA reception alternate function pin (RXDA2 and RXDA3).
  → When the RXDA0, RXDA1, RXDA2 or RXDA3 signal of UART0, UART1, UART2 or UART3 is selected, the LIN reception transfer rate and baud rate error of UARTA can be calculated.

Cautions 1. When using the selector function, set the capture trigger input of TMP before connecting the timer.

2. When setting the selector function, first disable the peripheral I/O to be connected (TMP/UARTA or TMP/CAN controller).

The capture input for the selector function is specified by the following register.

(1) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TMPn.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF308H

(i) V850ES/FE2, V850ES/FF2, V850ES/FG2

	7	6	5	4	3	2	1	0
SELCNT0	0	0	0	ISEL04	ISEL03	ISEL02	0	ISEL00

(ii) V850ES/FJ2: µPD70F3237

	7	6	5	4	3	2	1	0
SELCNT0	0	0	ISEL05	ISEL04	ISEL03	ISEL02	ISEL01	ISEL00

(iii) V850ES/FJ2: µPD70F3238, µPD70F3239

	7	6	5	4	3	2	1	0
SELCNT0	0	ISEL06	ISEL05	ISEL04	ISEL03	ISEL02	ISEL01	ISEL00

ISEL06	Selection of TIP31 input signal (TMP3)
0	TIP31 pin input
1	RXDA3 pin input

ISEL05	Selection of TIP30 input signal (TMP3)
0	TIP30 pin input
1	RXDA2 pin input

ISEL04	Selection of TIP11 input signal (TMP1)
0	TIP11 pin input
1	RXDA1 pin input

ISEL03	Selection of TIP10 input signal (TMP1)
0	TIP10 pin input
1	RXDA0 pin input

ISEL02[Note]	Selection of TIP01 input signal (TMP0)
0	Signal selected by ISEL01 bit
1	INTTM0EQ0 interrupt of TMM0

ISEL01	Selection of TIP01 input signal (TMP0)
0	TIP01 pin input
1	TSOUT signal of CAN1

ISEL00	Selection of TIP00 input signal (TMP0)
0	TIP00 pin input
1	TSOUT signal of CAN0

Note Use the INTTM0EQ0 interrupt signal as the TIP01 input signal in the following range.
TMM operation clock cycle ≥ TMP operation clock cycle × 4

Cautions 1. To set the ISEL06 to ISEL00 bits to 1, set the corresponding pin in the capture input mode.
2. Set TMP0 and CAN0 after prohibiting operating when you set the ISEL00 bit.
Set TMP0 and CAN1 after prohibiting operating when you set the ISEL01 bit.
Set TMP0 and TMM0 after prohibiting operating when you set the ISEL02 bit.
Set TMP1 and UARTA0 after prohibiting operating when you set the ISEL03 bit.
Set TMP1 and UARTA1 after prohibiting operating when you set the ISEL04 bit.
Set TMP3 and UARTA2 after prohibiting operating when you set the ISEL05 bit.
Set TMP3 and UARTA3 after prohibiting operating when you set the ISEL06 bit.

(2) Selector operation control register 1 (SELCNT1)

The SELCNT1 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.

This register is incorporated only in the μPD70F3238 and μPD70F3239.

After reset: 00H R/W Address: FFFFF30AH

(i) V850ES/FJ2: μPD70F3238, μPD70F3239

	7	6	5	4	3	2	1	0
SELCNT1	0	0	0	0	0	0	ISEL11	ISEL10

ISEL11[Note]	Selection of TIP21 input signal (TMP2)
0	TIP21 pin input
1	TSOUT signal of CAN3

ISEL10[Note]	Selection of TIP20 input signal (TMP2)
0	TIP20 pin input
1	TSOUT signal of CAN2

Note The μPD70F3237 does not have the CAN3 and CAN2 functions. Fix the ISEL11 and ISEL10 bits of these products to 0.

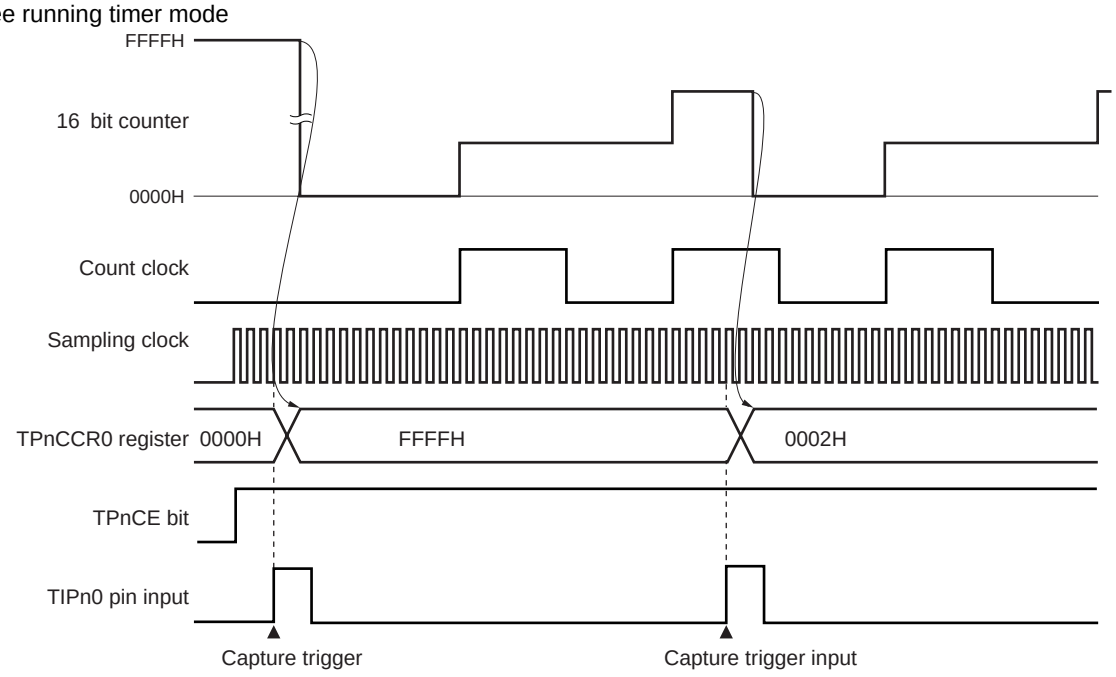
Cautions 1. To set the ISEL11 and ISEL10 bits to 1, set the corresponding pin in the capture input mode.
2. Set TMP2 and CAN2 after prohibiting operating when you set the ISEL10 bit. Set TMP2 and CAN3 after prohibiting operating when you set the ISEL11 bit.

7. 8 Cautions

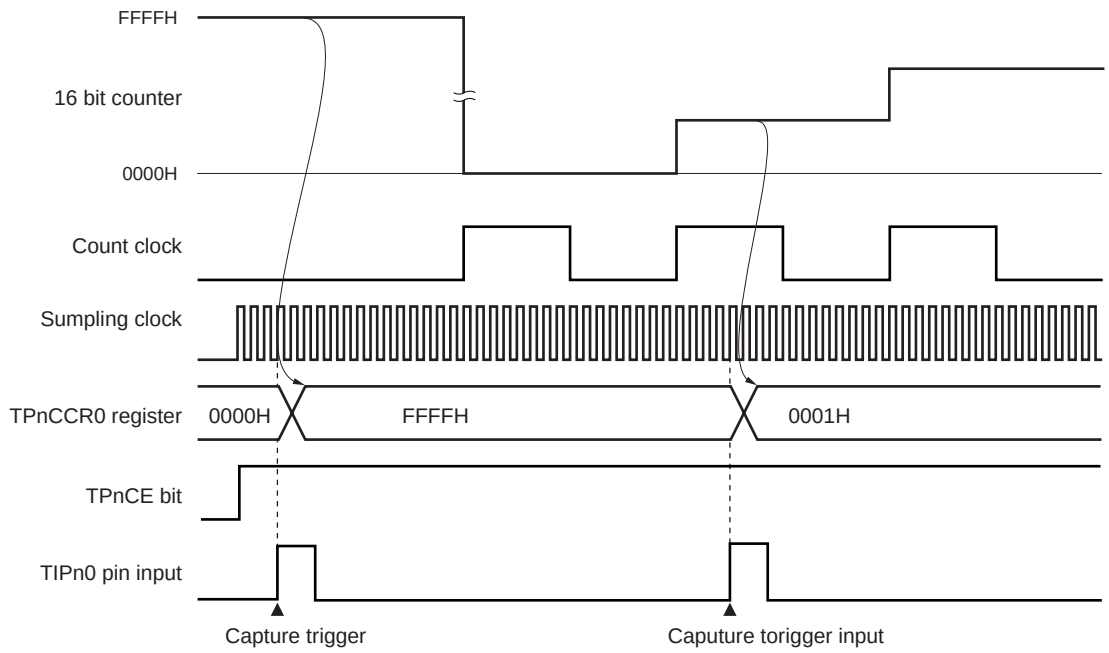
(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TPnCCR0 and TPnCCR1 registers if the capture trigger is input until operation start of count clock after the TPnCE bit is set to 1.

(a)Free running timer mode



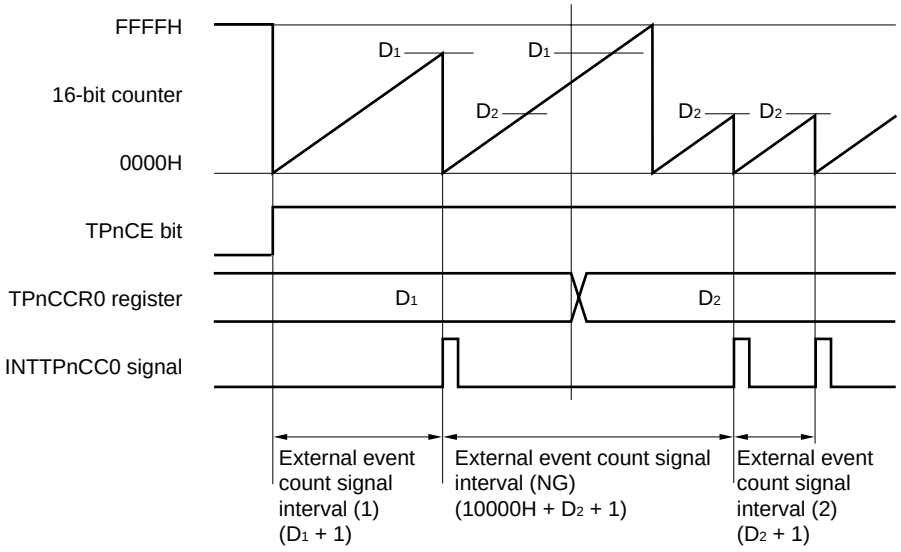
(b)Pulse mode



(2) Notes on rewriting the TPnCCR0 register

To change the value of the TPnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remark n = 0 to 3

If the value of the TPnCCR0 register is changed from D1 to D2 while the count value is greater than D2 but less than D1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D2.

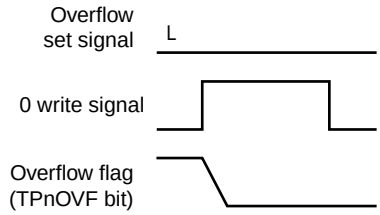
Because the count value has already exceeded D2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D2, the INTTPnCC0 signal is generated.

Therefore, the INTTPnCC0 signal may not be generated at the valid edge count of “(D1 + 1) times” or “(D2 + 1) times” originally expected, but may be generated at the valid edge count of “(10000H + D2 + 1) times”.

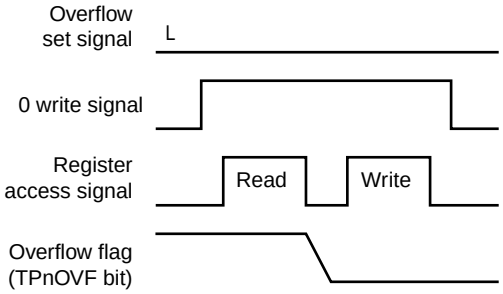
(3) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TPnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TPnOPT0 register. To accurately detect an overflow, read the TPnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

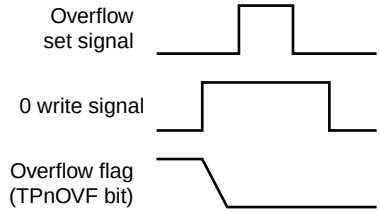
(i) Operation to write 0 (without conflict with setting)



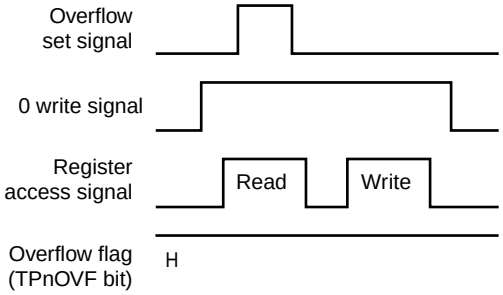
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



Remark n = 0 to 3

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

CHAPTER 8 16-BIT TIMER/EVENT COUNTER Q

The V850ES/FE2, V850ES/FF2, V850ES/FG2, and V850ES/FJ2 include 16-bit timer/event counter Q. The number of channels of timer Q (TMQ) differs depending on the product.

Table 8-1. Number of Channels of Timer Q

Product	Number of Channels
V850ES/FE2	1 (TMQ0)
V850ES/FF2	
V850ES/FG2	2 (TMQ0, TMQ1)
V850ES/FJ2	3 (TMQ0, TMQ1, TIMQ2)

8.1 Features

Timer Q (TMQ) is a 16-bit timer/event counter that can be used in various ways.
TMQ can perform the following operations.

- PWM output
- Interval timer
- External event counter (operation disabled when clock is stopped)
- One-shot pulse output
- Pulse width measurement function
- Triangular wave PWM output
- Timer synchronized operation function
- External trigger pulse output function
- Free-running function

8.2 Functional Outline

- Capture trigger input signal × 4
- External trigger input signal × 1
- Clock selection × 8
- External event count input × 1
- Readable counter × 1
- Capture/compare reload register × 4
- Capture/compare match interrupt × 4
- Timer output (TOQn0 to TOQn3) × 4

Remark n = 0 (V850ES/FE2, V850ES/FF2)
n = 0,1 (V850ES/FG2)
n = 0 to 2 (V850ES/FJ2)

This chapter explains the case where n = 0 to 2.

8.3 Configuration

TMQ consists of the following hardware.

Table 8-2. Configuration of TMQ0 to TMQ2

Item	Configuration
Timer register	16-bit counter
Registers	TMQn capture/compare registers 0 to 3 (TQnCCR0 to TQnCCR3) TMQn counter read buffer register (TQnCNT) CCR0 buffer register to CCR3 buffer register
Timer inputs	4 (TIQn0[Note] to TIQn3)
Timer outputs	2 (TOQn0 to TOQn3)
Control registers	TMQn timer control registers 0, 1 (TQnCTL0, TQnCTL1) TMQn timer dedicated I/O control registers 0 to 2 (TQnIOC0 to TQnIOC2) TMQn timer option register 0 (TQnOPT0) TIQnm pin noise elimination control register (QnmNFC)

Note TIQn0 functions alternately as a capture trigger input signal, external trigger input signal, and external event count input signal.

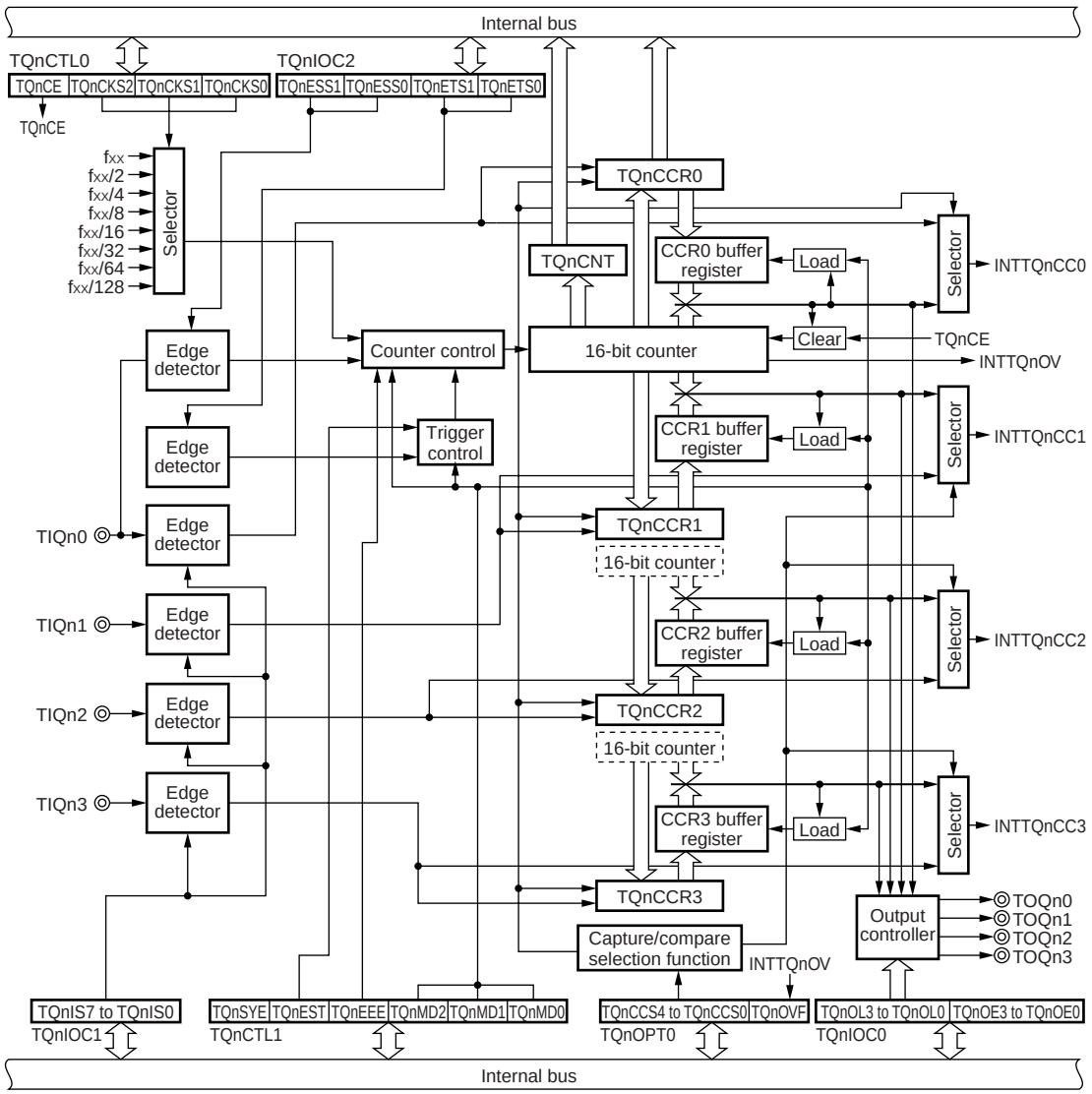
Remark n = 0 to 2, m = 0 to 3

The pins of TMQ function alternately as port pins. For how to set the alternate function, refer to the description of the registers in **CHAPTER 4 PORT FUNCTIONS**.

Table 8-3. TMQ Pin List

Pin Name	Alternate-Function Pin	I/O	Function
TIQ00	P53/KR3/TOQ00/DDO	Input	External event/clock input (TMQ0)
TIQ01	P50/KR0/TOQ01		
TIQ02	P51/KR1/TOQ02		
TIQ03	P52/KR2/TOQ03/DDI		
TIQ10	P95/TOQ10		External event/clock input (TMQ1)
TIQ11	P92/TOQ11		
TIQ12	P93/TOQ12		
TIQ13	P94/TOQ13		
TIQ20	P610/TOQ20		External event/clock input (TMQ2)
TIQ21	P611/TOQ21		
TIQ22	P612/TOQ22		
TIQ23	P613/TOQ23		
TOQ00	P53/KR3/TIQ00/DDO	Output	Timer output (TMQ0)
TOQ01	P50/KR0/TIQ01		
TOQ02	P51/KR1/TIQ02		
TOQ03	P52/KR2/TIQ03/DDI		
TOQ10	P95/TIQ10		Timer output (TMQ1)
TOQ11	P92/TIQ11		
TOQ12	P93/TIQ12		
TOQ13	P94/TIQ13		
TOQ20	P610/TIQ20		Timer output (TMQ2)
TOQ21	P611/TIQ21		
TOQ22	P612/TIQ22		
TOQ23	P613/TIQ23		

Figure 8-1. Block Diagram of Timer Q



Remark n = 0 to 2

(1) TMQn capture/compare register 0 (TQnCCR0)

The TQnCCR0 register is a 16-bit register that has a capture function and a compare function.

A capture register or a compare register behavior can be set by the setting of TQnCCS0 bit only in the free-running mode.

In the pulse width measurement mode, this register functions only as a capture 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.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TQ0CCR0: FFFFF546H, TQ1CCR0: FFFFF616H, TQ2CCR0: FFFFF626H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR0																

(n = 0 to 2)

- When used as compare register

  TQnCCR0 can be rewritten when TQnCE = 1.

  Each operation mode and capture/compare register functions and the method of writing the compare register are as follows.

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	Anytime write
Pulse width measurement mode	Cannot be written because used only as capture register

- When used as capture register

  The count value is stored in TQnCCR0 on detection of the edge of the capture trigger (TIQn0) input.

Caution: At subclock operated and at main clock stopped, the access to TQnCCR0 register is prohibited. For details, refer to 3. 4. 10（2）

(2) TMQn capture/compare register 1 (TQnCCR1)

The TQnCCR1 register is a 16-bit register that has a capture function and a compare function.

A capture register or a compare register behavior can be specified by setting the TQnCCS1 bit of the TQnOPT0 register only in the free-running mode.

In the pulse width measurement mode, this register functions only as a capture register.

In all the modes other than the free-running mode and pulse width measurement mode, this register functions as a compare register.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TQ0CCR1: FFFFF548H, TQ1CCR1: FFFFF618H, TQ2CCR1: FFFFF628H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR1																

(n = 0 to 2)

- When used as compare register

  TQnCCR1 can be rewritten when TQnCE = 1.

  Each operation mode and capture/compare register functions and the method of writing the compare register are as follows.

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	Anytime write
Pulse width measurement mode	Cannot be written because used only as capture register

- When used as capture register

  The count value is stored in TQnCCR1 on detection of the edge of the capture trigger (TIQn1) input.

Caution: At subclock operated and at main clock stopped, the access to TQnCCR1 register is prohibited. For details, refer to 3. 4. 10（2）

(3) TMQn capture/compare register 2 (TQnCCR2)

The TQnCCR2 register is a 16-bit register that has a capture function and a compare function.

A capture register or a compare register behavior can be specified by setting the TQnCCS2 bit of the TQnOPT0 register only in the free-running mode.

In the pulse width measurement mode, this register functions only as a capture 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.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TQ0CCR2: FFFFF54AH, TQ1CCR2: FFFFF61AH, TQ2CCR2: FFFFF62AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR2																

(n = 0 to 2)

- When used as compare register

  TQnCCR2 can be rewritten when TQnCE = 1.

  Each operation mode and capture/compare register functions and the method of writing the compare register are as follows.

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	Anytime write
Pulse width measurement mode	Cannot be written because used only as capture register

- When used as capture register

  The count value is stored in TQnCCR2 on detection of the edge of the capture trigger (TIQn2) input.

Caution: At subclock operated and at main clock stopped, the access to TQnCCR1 register is prohibited. For details, refer to 3. 4. 10（2）

(4) TMQn capture/compare register 3 (TQnCCR3)

The TQnCCR3 register is a 16-bit register that has a capture function and a compare function.

A capture register or a compare register can be specified by setting the TQnCCS3 bit of the TQnOPT0 register only in the free-running mode.

In the pulse width measurement mode, this register functions only as a capture 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.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TQ0CCR3: FFFFF54CH, TQ1CCR3: FFFFF61CH, TQ2CCR3: FFFFF62CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR3																

(n = 0 to 2)

- When used as compare register

  TQnCCR3 can be rewritten when TQnCE = 1.

  Each operation mode and capture/compare register functions and the method of writing the compare register are as follows.

TMQ Operation Mode	Method of Writing TQnCCR3 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	Anytime write
Pulse width measurement mode	Cannot be written because used only as capture register

- When used as capture register

  The count value is stored in TQnCCR3 on detection of the edge of the capture trigger (TIQn3) input.

Caution: At subclock operated and at main clock stopped, the access to TQnCCR1 register is prohibited. For details, refer to 3. 4. 10（2）

(5) TMQn counter read buffer register (TQnCNT)

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

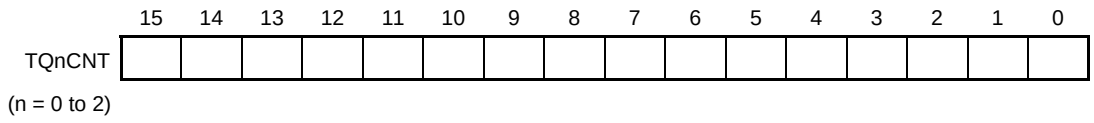
This register is read-only, in 16-bit units.

Reset input clears this register to FFFFH.

When TQnCE bit = 0, the TQnCNT register is 000H. At this time if TQnCNT register is read, the value of 16-bit counter is not read and 0000H is read as it is.

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

After reset: FFFFH R Address: TQ0CNT: FFFFF54EH, TQ1CNT: FFFFF61EH,
TQ2CNT: FFFFF62EH



Caution: At subclock operated and at main clock stopped, the access to TQnCCR1 register is prohibited. For details, refer to 3. 4. 10（2）

8.4 Control Registers

(1) TMQn control register 0 (TQnCTL0)

The TQnCTL0 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.

Reset input clears this register of default value to 00H.

Rewriting the TQnCTL0 register is prohibited while operating (TQnCE = 1). However, only the TQnCE bit can be rewritten at any time.

(1/2)

After reset: 00H R/W Address: TQ0CTL0: FFFFF540H, TQ1CTL0: FFFFF610H, TQ2CTL0: FFFFF620H

	7	6	5	4	3	2	1	0
TQnCTL0 (n = 0 to 2)	TQnCE	0	0	0	0	TQnCKS2	TQnCKS1	TQnCKS0

TQnCE	Control of operation of timer Qn
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	Selection of internal count clock
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

Cautions: 1. Set the TQnCKS2 to TQnCKS0 bits when TQnCE = 0. When the TQnCE bit setting is changed from 0 to 1, the TQnCKS2 to TQnCKS0 bits can be set at the same time.

2. Be sure to clear bit 3 to bit 6 to 0.

Remark f_{XX}: Main system clock frequency

Resolution and Maximum Count time

Internal count clock	Resolution [μs]		Maximum Count Time [ms]	
	f_{XX} = 16 MHz	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 20 MHz
f_{XX}	0.0625	0.050	4.10	3.28
$f_{XX}/2$	0.125	0.100	8.19	6.55
$f_{XX}/4$	0.250	0.200	16.38	13.11
$f_{XX}/8$	0.500	0.400	32.77	26.21
$f_{XX}/16$	1.000	0.800	65.54	52.43
$f_{XX}/32$	2.000	1.600	131.11	104.86
$f_{XX}/64$	4.000	3.200	262.14	209.72
$f_{XX}/128$	8.000	6.400	524.29	419.43

(2) TMQn timer 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.

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: TQ0CTL1: FFFFF541H, TQ1CTL1: FFFFF611H, TQ2CTL1: FFFFF621H

	7	6	5	4	3	2	1	0
TQnCTL1	TQnSYE	TQnEST	TQnEEE	0	0	TQnMD2	TQnMD1	TQnMD0

(n = 0 to 2)

TQnSYE	Tuned operation mode enable
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer Q can operate in synchronization with a master timer. Master timer: TMP2 — Slave timer: TMP3, TMQ0 Master timer: TMQ1 — Slave timer: TMQ2, – For the tuned operation mode, refer to **8.6 Timer Synchronized Operation Function**.

TQnEST	Software trigger control
0	No operation
1	In one-shot pulse mode: One-shot pulse software trigger In external trigger pulse output mode: Pulse output software trigger
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.	

TQnEEE	Selection of count clock
0	Internal clock (clock selected by TQnCKS2 to TQnCKS0 bits)
1	External event count input (edge of input to TIQn0)
The valid edge is specified by the TQnEES1 and TQnEES0 bits when TQnEEE = 1 (External event count input: TIQn0).	

(2/2)

TQnMD2	TQnMD1	TQnMD0	Selection of timer mode
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse 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

Cautions 1. Set the TQnEEE and TQnMD2 to TQnMD0 bits when TQnCE = 0 (the same value can be written when TQnCE = 1). If these bits are rewritten when TQnCE = 1, the operation cannot be guaranteed. If these bits are rewritten by mistake, clear TQnCE to 0 and then set them again.

2. The external event count input is selected regardless of the value of the TQnEEE bit at an external event count mode.

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

(3) TMQn timer dedicated I/O control register 0 (TQnIOC0)

The TQnIOC0 register is an 8-bit register that controls the timer outputs.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TQ0IOC0: FFFFF542H, TQ1IOC0: FFFFF612H, TQ2IOC0: FFFFF622H

	7	6	5	4	3	2	1	0
TQnIOC0	TQnOL3	TQnOE3	TQnOL2	TQnOE2	TQnOL1	TQnOE1	TQnOL0	TQnOE0

(n = 0 to 2)

TQnOLm	Setting of TOQnm output level (m = 0 to 3)
0	Normal output
1	Inverted output

TQnOEm	Setting of TOQnm output (m = 0 to 3)
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 the TQnOL1, TQnOE1, TQnOL0, and TQnOE0 bits when TQnCE = 0 (the same value can be written when TQnCE = 1). If these bits are rewritten by mistake, clear TQnCE to 0 and then set them 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.

3. The output level of the TOQnm pin changes into the state of TQnCE=0 and TQnOEm=0 if the TQnOLm bit is operated when the pin is assumed to be a control output mode.

(4) TMQn timer 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.

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: TQ0IOC1: FFFFF543H, TQ1IOC1: FFFFF613H, TQ2IOC1: FFFFF623H

	7	6	5	4	3	2	1	0
TQnIOC1	TQnIS7	TQnIS6	TQnIS5	TQnIS4	TQnIS3	TQnIS2	TQnIS1	TQnIS0

(n = 0 to 2)

TQnIS7	TQnIS6	Setting of valid edge of capture input (TIQn3)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TQnIS5	TQnIS4	Setting of valid edge of capture input (TIQn2)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TQnIS3	TQnIS2	Setting of valid edge of capture input (TIQn1)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TQnIS1	TQnIS0	Setting of valid edge of capture input (TIQn0)
0	0	Detect no edge (capture operation is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

Remark: Refer to the next page for the cautions.

(2/2)

Cautions 1. Rewrite the TQnIS7 to TQnIS0 bits when TQnCE0 = 0 (the same value can be written when TQnCE = 1). If these bits are rewritten by mistake, clear TQnCE to 0 and then set them 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.

3. To use the capture input, be sure to set the corresponding alternate-function pins TQnOE3 to TQnOE0 of the TQnIOC register to "Timer output prohibit" and be sure to set variable edge of capture input. Then, set the corresponding alternate-function port to input mode.

4. To use the external event count mode (TQnCTL1.TQ0EEE bit=1), be sure to set TIQn0 capture input to "No edge detection" (TQnIS1, 0 bit=00b).

(5) TMQn timer 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.

Reset input clears this register to 00H.

After reset: 00H R/W Address: TQ0IOC2: FFFFF544H, TQ1IOC2: FFFFF614H, TQ2IOC2: FFFFF624H

	7	6	5	4	3	2	1	0
TQnIOC2	0	0	0	0	TQnEES1	TQnEES0	TQnETS1	TQnETS0

(n = 0 to 2)

TQnEES1	TQnEES0	Setting of valid edge of external event count input (TIQn0)
0	0	Detect no edge (external event count is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

TQnETS1	TQnETS0	Setting of valid edge of external trigger input (TIQn0)
0	0	Detect no edge (external trigger is invalid).
0	1	Detect rising edge.
1	0	Detect falling edge.
1	1	Detect both the edges.

Cautions 1. Rewrite the TQnEES1, TQnEES0, TQnETS1 and TQnETS0 bits when TQnCE = 0 (the same value can be written when TQnCE = 1). If these bits are rewritten by mistake, clear TQnCE to 0 and then set them 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).

3. When setting of the external trigger pulse output mode (TQnCTL1.TQnMD2-0=010b) or the one-shot pulse output mode (TQnCTL1.TQnMD2=011b) only, TQnETS1, TQnETS0 bits are variable.

(6) TMQn timer 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 or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: TQ0OPT0: FFFFF545H, TQ1OPT0: FFFFF615H, TQ2OPT0: FFFFF625H

TQnOPT0 (n = 0 to 2)

7	6	5	4	3	2	1	0
TQnCCS3	TQnCCS2	TQnCCS1	TQnCCS0	0	0	TQnCUF	TQnOVF

TQnCCSm	Selection of capture or compare operation of TQnCCRm register (m = 0 to 3)
0	Compare register
1	Capture register
The set value of the TQnCCSm bit is valid only in the free-running mode.	

TQnCUF	Timer Q down count flag
0	TMQn counting up
1	TMQn counting down
TQnUCF bit is valid in the triangular wave PWM mode. This is read-only; a value written to this flag is invalid.	

TQnOVF	Detection of overflow of timer Q
Set (1)	Overflow occurred
Reset (0)	0 written to TQnOVF bit or TQnCE = 0

- The TQnOVF bit is set when the 16-bit counter overflows from FFFFH to 0000H in the free-running mode and 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 if the TQnOVF bit and TQnOPT0 register are read when TQnOVF = 1.
- The TQnOVF bit can be read and written, but 1 cannot be written to the TQnOVF bit. Writing 1 to this bit does not affect the operation of timer Q.

Cautions 1. Rewrite the TQnCCS1 and TQnCCS0 bits when TQnCE0 = 0 (the same value can be written when TQnCE = 1). If these bits are rewritten by mistake, clear TQnCE to 0 and then set them again.

2. Be sure to clear bits 2 and 3 to 0.

(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 for noise elimination.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: 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)
Q20NFC: FFFFFB70H (TIQ20 pin)
Q21NFC: FFFFFB74H (TIQ21 pin)
Q22NFC: FFFFFB78H (TIQ22 pin)
Q23NFC: FFFFFB7CH (TIQ23 pin)

	7	6	5	4	3	2	1	0
QnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFSTS	Setting of number of times of sampling by digital noise filter
0	3 times
1	2 times

NFC2	NFC1	NFC0	Sampling clock
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 (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 to 2)
m: Number of input pins (0 to 3)

8.5 Operation

Timer Q performs the following operations.

Operation	TQnEST (Software Trigger Bit)	TIQn0 (External Trigger Input)	Capture/Compare Selection	Compare Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Valid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Reload
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM mode	Invalid	Invalid	Compare only	Reload
Free-running mode	Invalid	Invalid	Capture/compare selectable	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable
Triangular wave PWM mode	Invalid	Invalid	Compare only	Reload

Notes 1. To use the external event counter input function, specify that the input edge of the TIQn0 pin is not detected (by clearing the TQnIS1 and TQnIS0 bits of the TQnIOC1 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 TQnEEE bit of the TQnCTL1 register to 0).

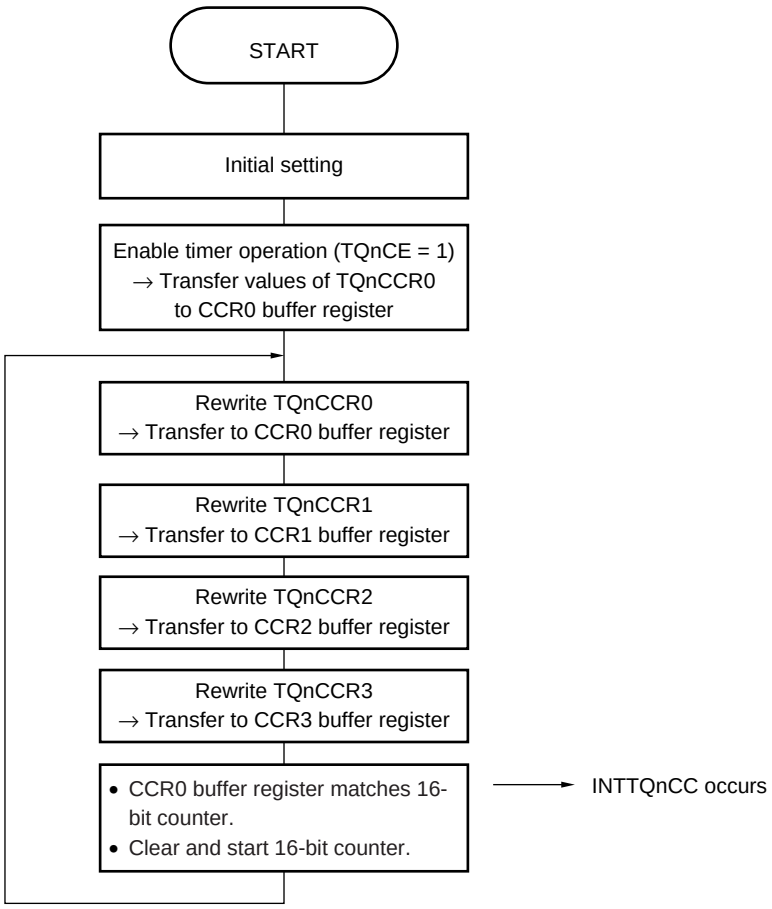
8.5.1 Anytime write and reload

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 to CCR3 buffer register and is compared with the value of the 16-bit counter.

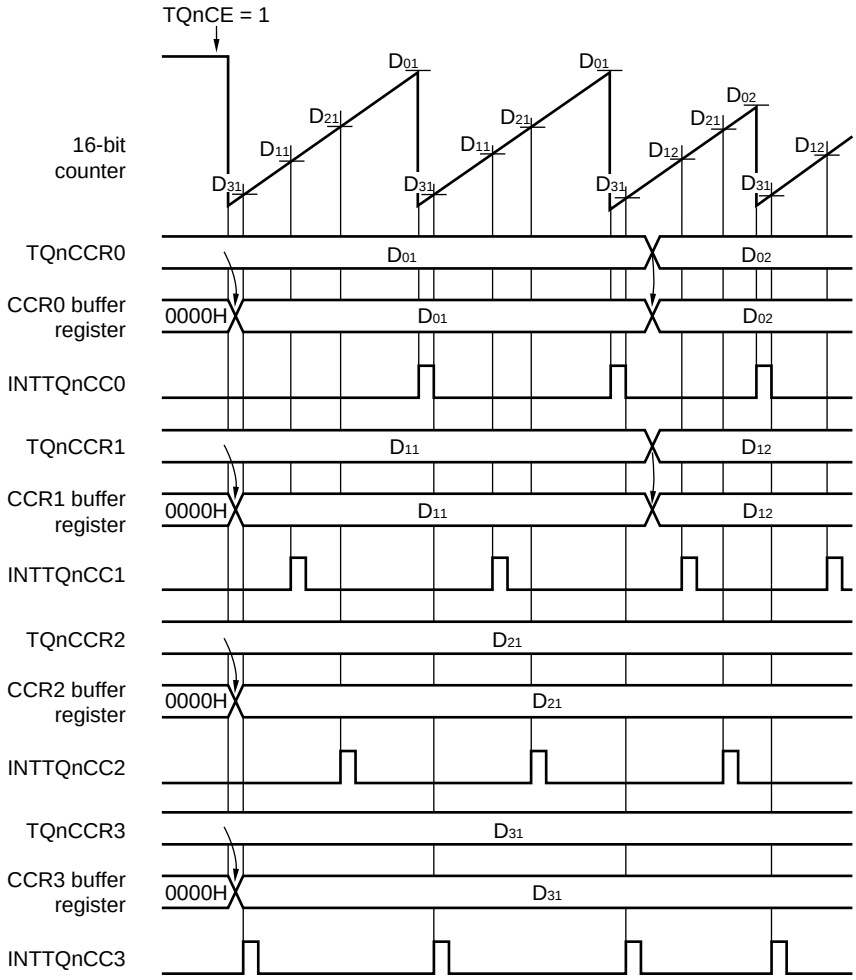
Figure 8-2. Flowchart of Basic Operation of Anytime Write



Remarks 1. This is an example in the interval timer mode.

2. n = 0 to 2

Figure 8-3. Timing Chart of Anytime Write



Remarks 1. D_{01}, D_{02}: Set value of TQnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)
D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)
D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

2. This is an example in the interval timer mode.

3. n = 0 to 2

(2) Reload

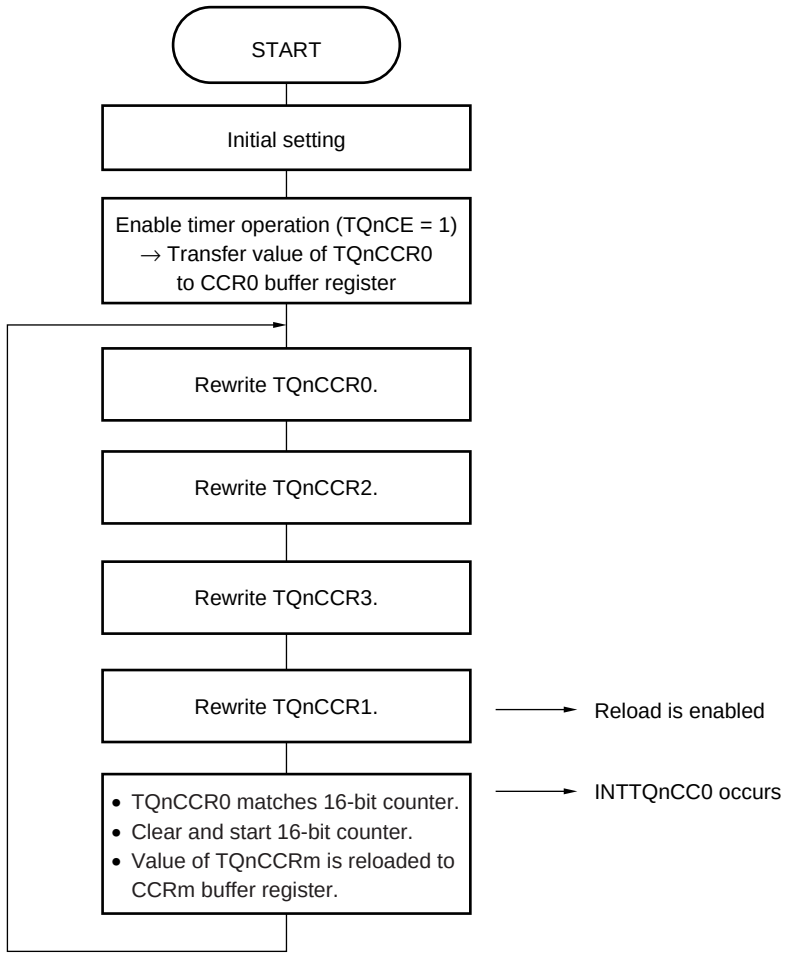
When data is written to the TQnCCRm register during timer operation, it is compared with the value of the 16-bit counter via the CCRm buffer register. The value of the TQnCCRm register can be rewritten when TQnCE = 1. So that the set values of the TQnCCRm register is compared with the value of the 16-bit counter (the set values are reloaded to the CCRm buffer register), the value of the TQnCCR0, TQnCCR2 and TQnCCR3 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 CCR0 buffer register.

When the value of the CCR0 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.

Therefore, when rewriting value of TQnCCR0, TQnCCR2 or TQnCCR3 registers, be sure to write TQnCCR1 register to same value (Set same value of TQnCCR1 register).

Figure 8-4. Flowchart of Basic Operation of Reload

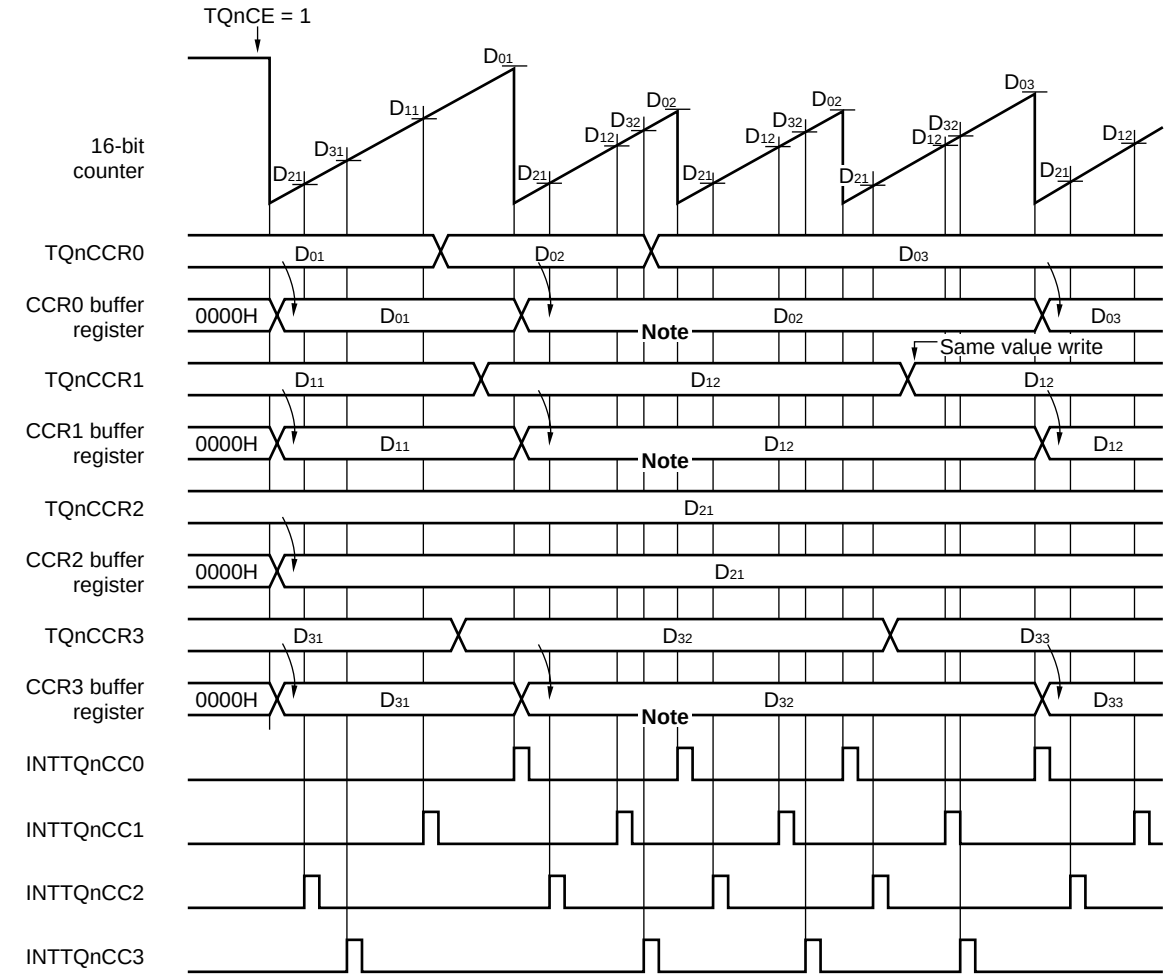


Caution Writing the TQnCCR1 register includes an operation to enable reload. Therefore, when rewriting one of TQnCCR0, TQnCCR2 or TQnCCR3 registers, TQnCCR1 register needs to write same value enable the next reload. Then rewrite the TQnCCR1 register after rewriting other TQnCCR registers.

Remarks 1. This is an example in the PWM mode.

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

Figure 8-5. Timing Chart of Reload



Note The value is not reloaded because the TQnCCR1 register is not written.

Remarks 1. D_{01}, D_{02}, D_{03}: Set value of TQnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)
D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)
D_{31}, D_{32}, D_{33}: Set value of TQnCCR3 register (0000H to FFFFH)

2. This is an example in the PWM mode.

3. n = 0 to 2

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, it is transferred to the CCRm buffer register by means of anytime write, and is compared with the value of the 16-bit counter.

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.

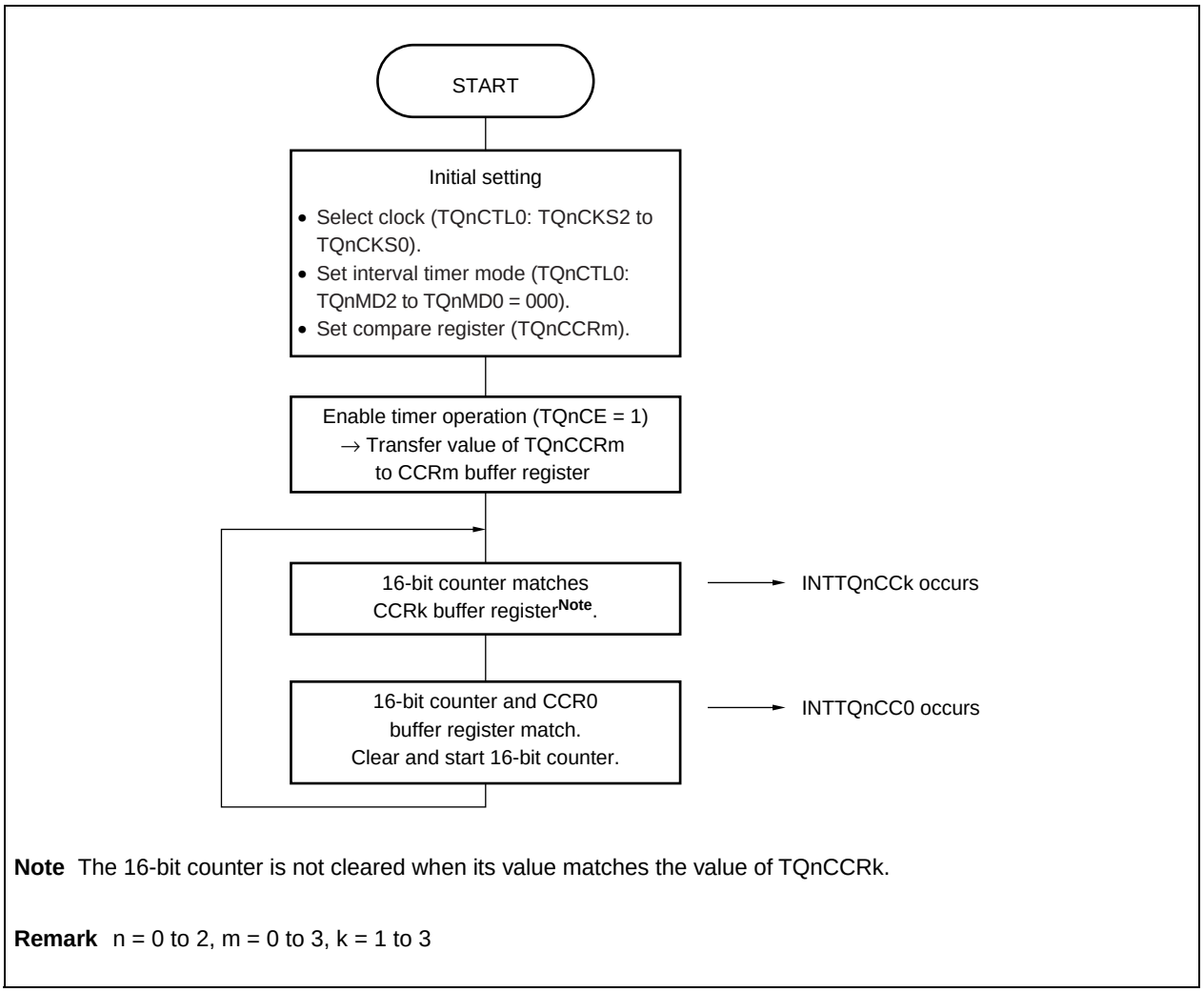
The value can also be output from the TOQnm pin by setting the TQnOEm bit to 1.

When the TQnCCRk register is not used, it is recommended to set the TQnCCRk register to FFFFH.

Remarks 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE=1), refer to **8. 5. 1 Anytime write**.

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

Figure 8-6. Flowchart of Basic Operation in Interval Timer Mode



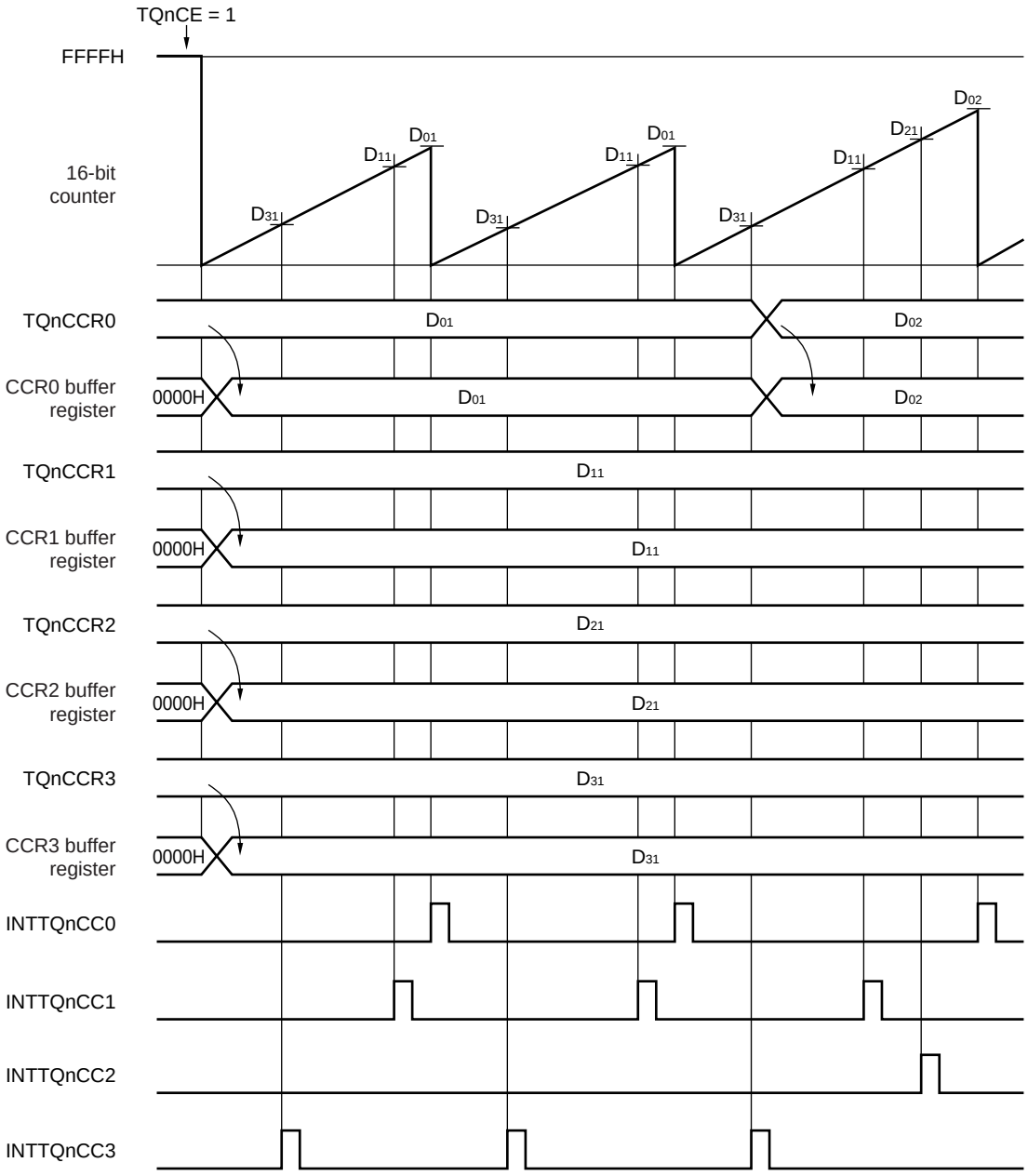
Note The 16-bit counter is not cleared when its value matches the value of TQnCCRk.

Remark n = 0 to 2, m = 0 to 3, k = 1 to 3

Figure 8-7. Timing of Basic Operation in Interval Timer Mode (1/2)

(a) When only TQnCCR0 register value is rewritten and TOQnm is not output

(TQnOEm = 0, TQnOLm = 0)



Remarks 1. D_{01}, D_{02}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

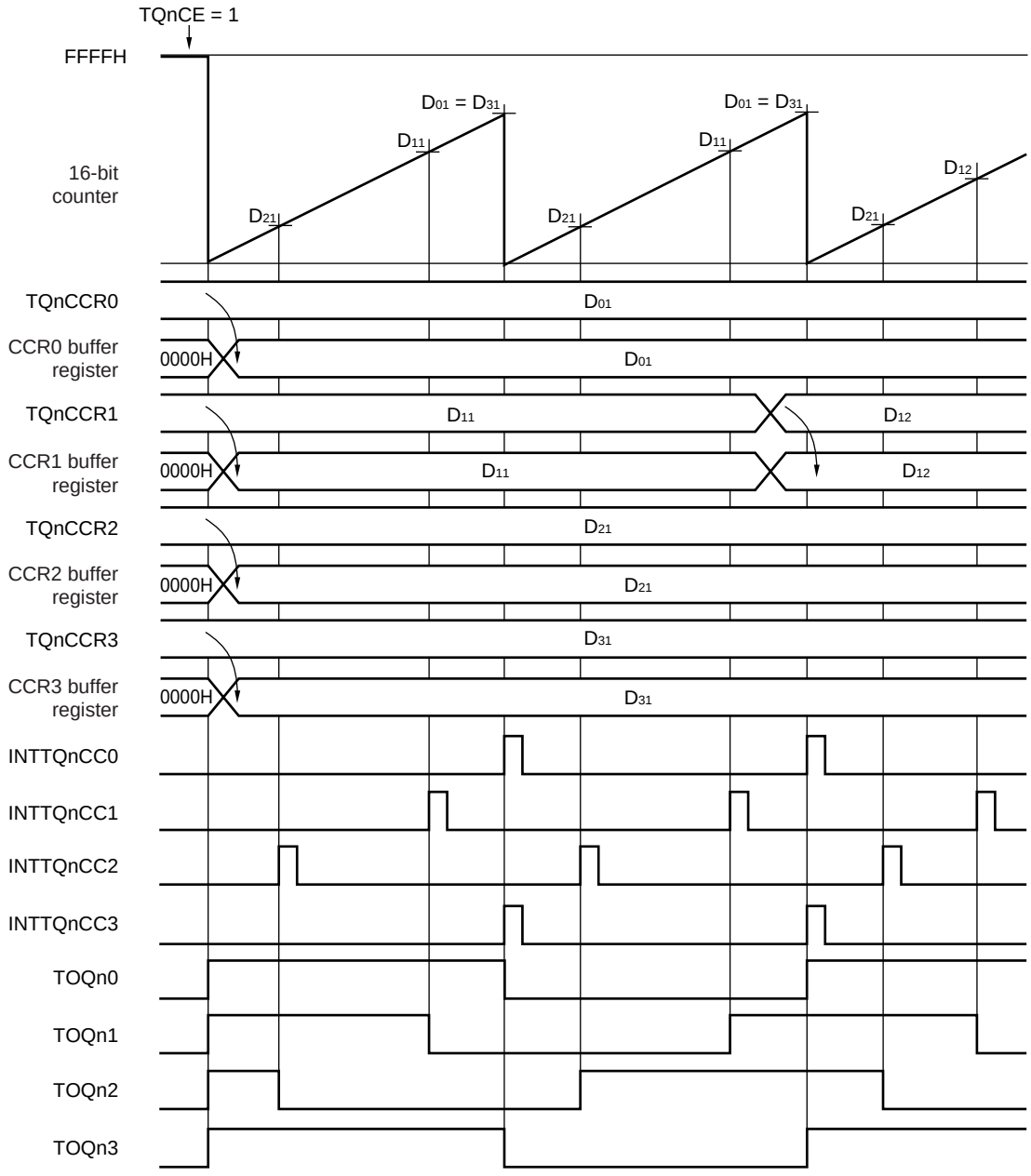
2. Interval time = (D0n + 1) × (Count clock cycle)

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

Figure 8-7. Timing of Basic Operation in Interval Timer Mode (2/2)

(b) When $D_{01} = D_{31}$, only TQnCCR1 register value is rewritten, and TOQnm is output

(TQnOEm = 1, TQnOLm = 0)



Remarks 1. D_{01}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

2. Interval time = (D0n + 1) × (Count clock cycle)

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

8.5.3 External event count 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, it is transferred to the CCRm buffer register by means of anytime write, and is compared with the value of the 16-bit counter.

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 TQnOEk bit to 1, a signal can be output from the TOQnk pin.

TOQn pin can not to use. When the TQnCCRk register is not used, it is recommended to set TQnCCRk to FFFFH.

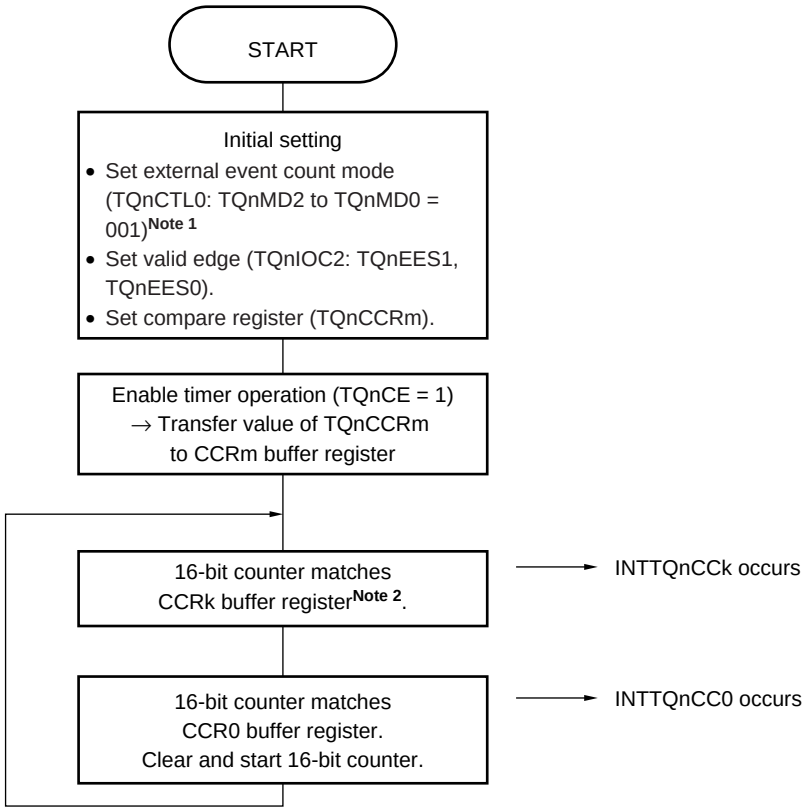
Remarks 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE=1), refer to **8. 5. 1 Anytime write**.

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

Caution 1. TOQn0 pin output in an external event count mode cannot be used. Set to TQnEEE = 1by interval timer mode (TQnMD2 to 0 = 000b) when TOQn0 pin output in an external event count mode is used.

2. In external event count mode, when TQnCCRm register value is set to 0000H the interrupt occurs after the overflow of the timer (FFFFH to 0000H)

Figure 8-8. Flowchart of Basic Operation in External Event Count Mode



Notes 1. Selecting the TQnEEE bit has no effect.

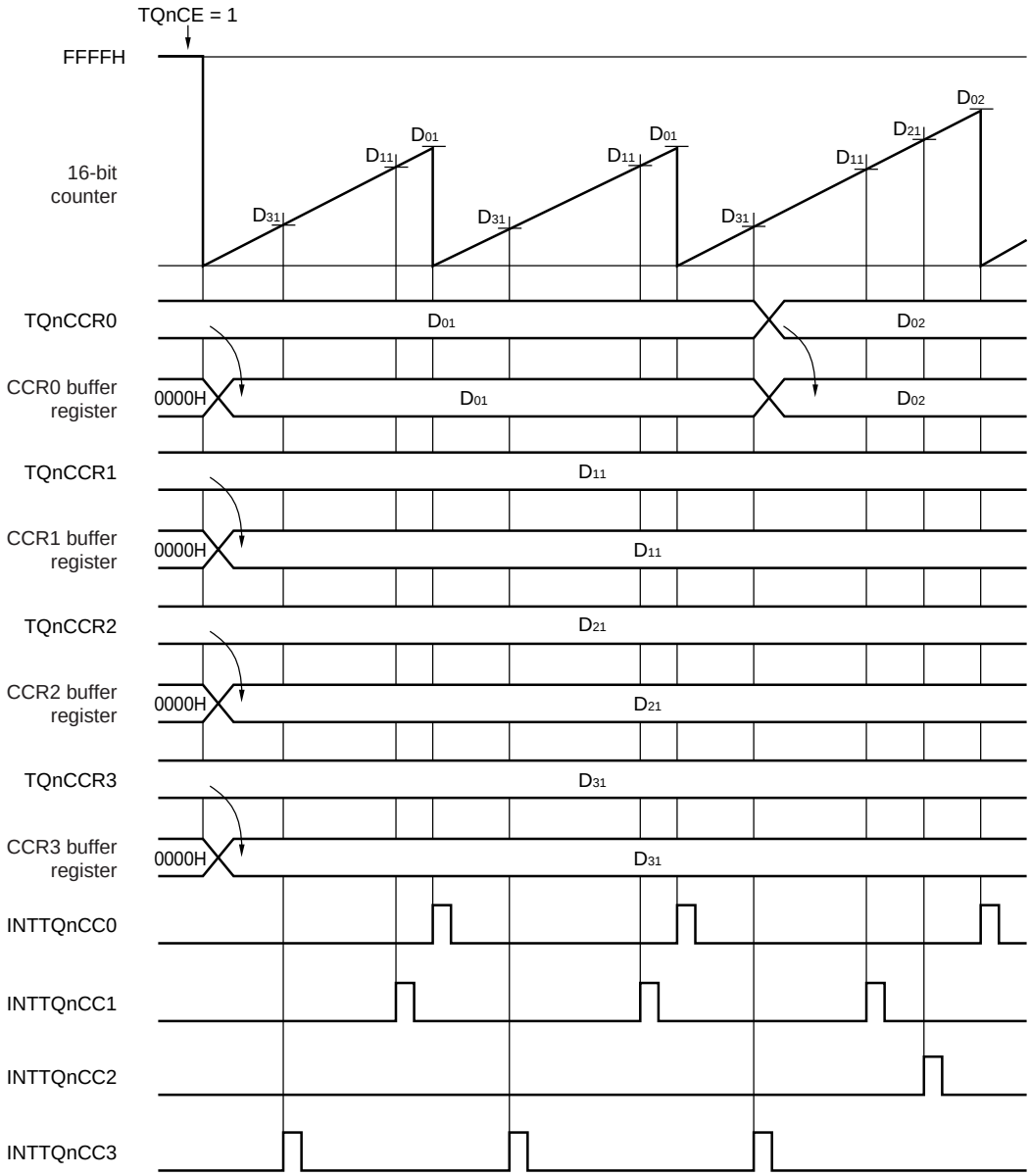
2. The 16-bit counter is not cleared when it matches the CCRk buffer register.

Remark n = 0 to 2, m = 0 to 3, k = 1 to 3

Figure 8-9. Timing of Basic Operation in External Event Count Mode (1/2)

(a) When only TQnCCR0 register value is rewritten and TOQnm is not output

(TQnOEm = 0, TQnOLm = 0)



Remarks 1. D_{01}, D_{02}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

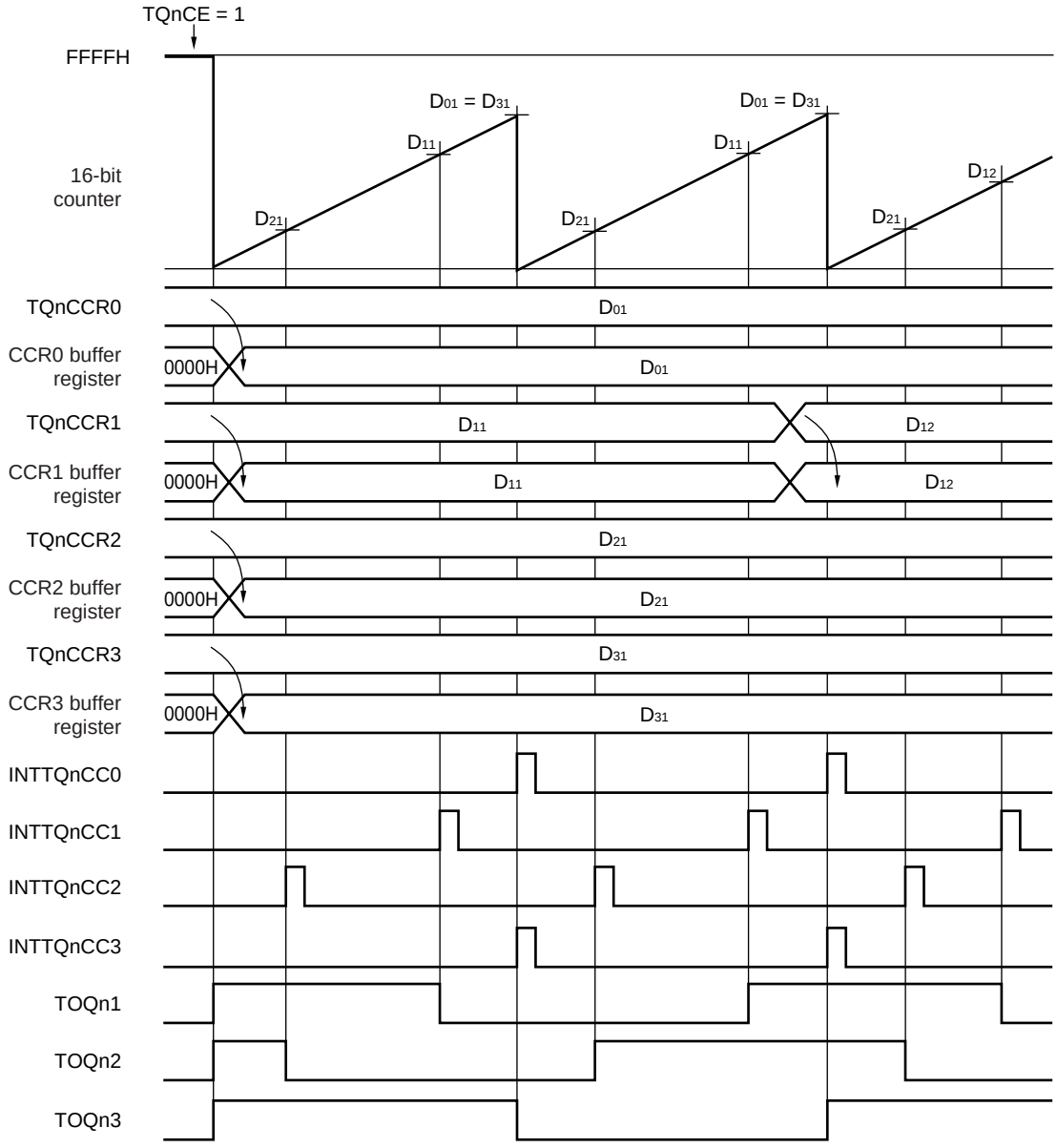
2. A compare match interrupt is generated each time (the value set to TQnCCRm register +1) is detected.

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

Figure 8-9. Timing of Basic Operation in External Event Count Mode (2/2)

(b) When $D_{01} = D_{31}$, only TQnCCR1 register is rewritten, and TOQnm is output

(TQnOE0 = 0, TQnOEk = 1, TQnOL0 = 0, TQnOLk = 0)



Remarks 1. D_{01}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

2. A compare match interrupt is generated each time (the value set to TQnCCRm register +1) is detected.

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

8.5.4 External trigger pulse output mode (TQnMD2 to TQnMD0 = 010)

When TQnCE = 1 in the external trigger pulse output 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.

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 output mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. At the same time, TOQn0 pin is initialized. To realize the same function as the external trigger pulse output mode by using a software trigger instead of the external trigger input (TIQn0 pin input) (software trigger pulse output 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.

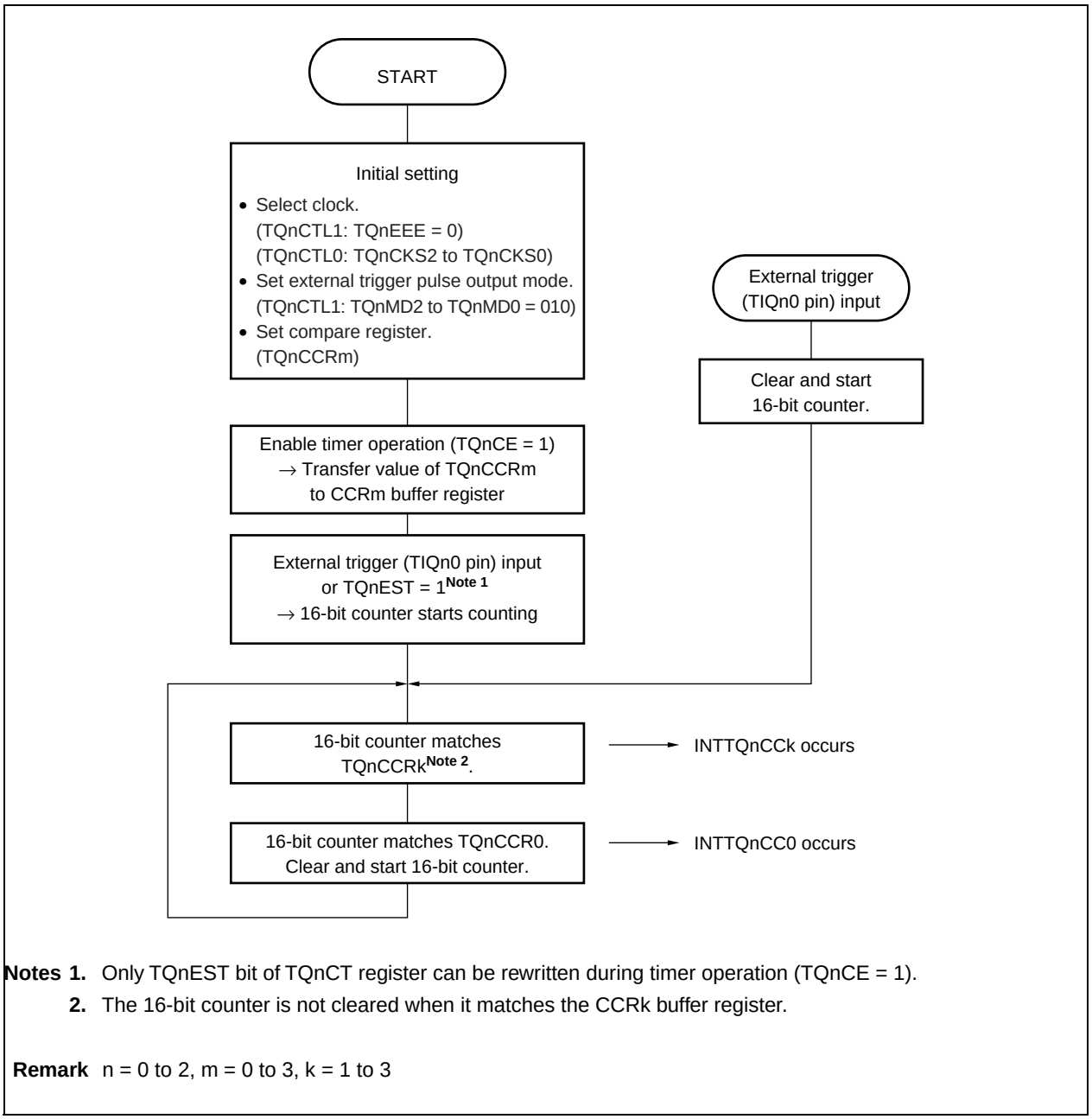
In the external trigger pulse output mode, the capture function of the TQnCCRm register cannot be used because this register can be used only as a compare register.

Caution In the external trigger pulse output mode, select the internal clock (TQnEEE of TQnCTL1 register = 0) as the count clock.

Remarks 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE=1), refer to **8.5.1 (2) Reload.**

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

Figure 8-10. Flowchart of Basic Operation in External Trigger Pulse Output Mode

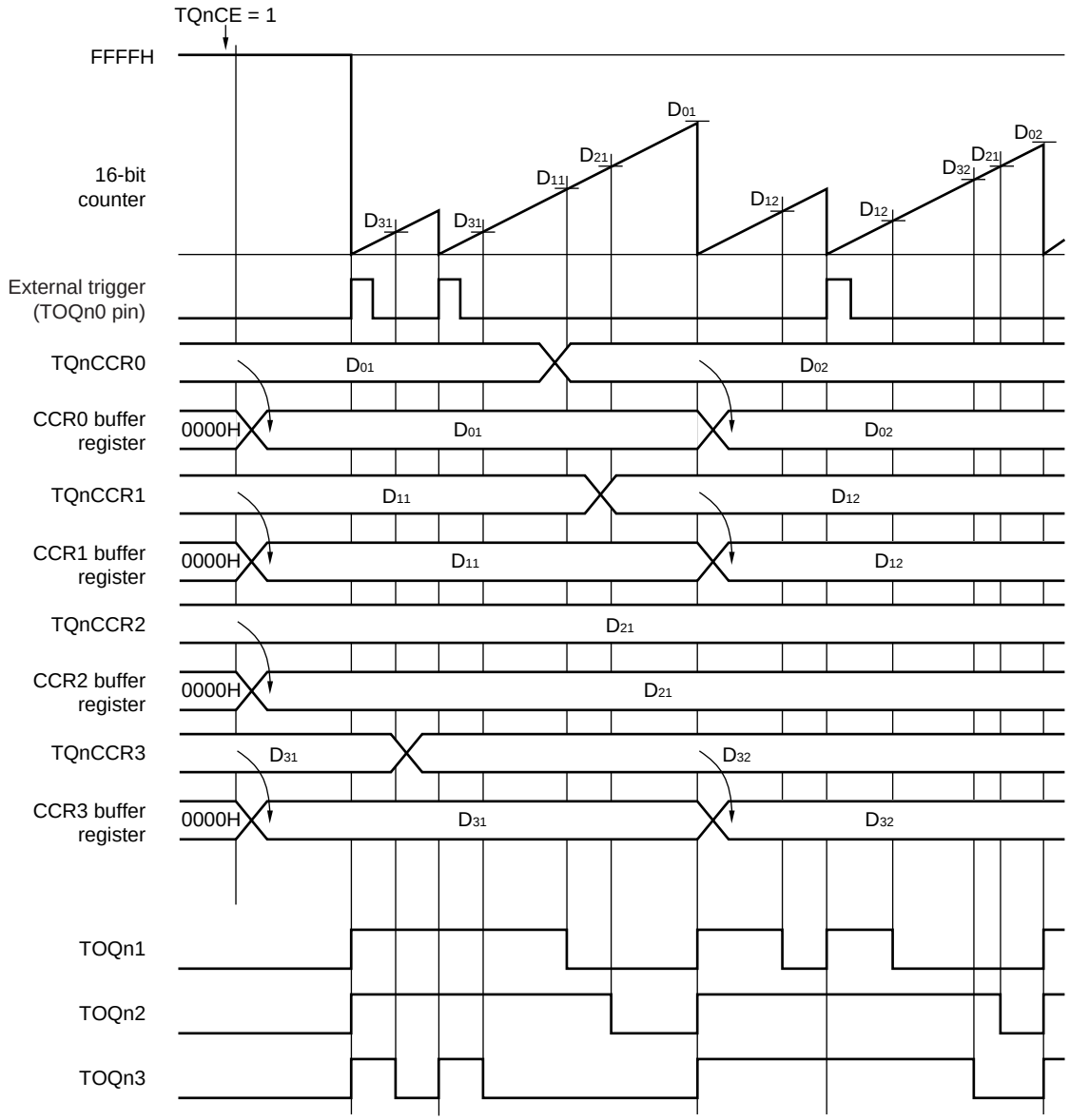


Notes 1. Only TQnEST bit of TQnCT register can be rewritten during timer operation (TQnCE = 1).

2. The 16-bit counter is not cleared when it matches the CCRk buffer register.

Remark n = 0 to 2, m = 0 to 3, k = 1 to 3

Figure 8-11. Timing of Basic Operation in External Trigger Pulse Output Mode
(TQnOE0 = 0, TQnOEk = 1, TQnOL0 = 0, TQnOLk = 0)



Remarks 1. D_{01}, D_{02}: Set value of TQnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)
D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)
D_{31}, D_{32}: Set value 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 +1) × (Count clock cycle)

3. n = 0 to 2, 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 inputted, 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. If the value is set to the TQnCCRm register, it is transferred to the CCRm buffer register by anytime write and it becomes an object of comparison value with 16 bit counter value. The waveform of the one-shot pulse is output from the TOQnk pin. The TOQnm pin produces an active level until counting by timer counter. Active level is set by TQnOL0 bit.

Cautions 1. Select the internal clock (TQnEEE of the TQnCTL1 register = 0) as the count clock in the one-shot pulse mode.

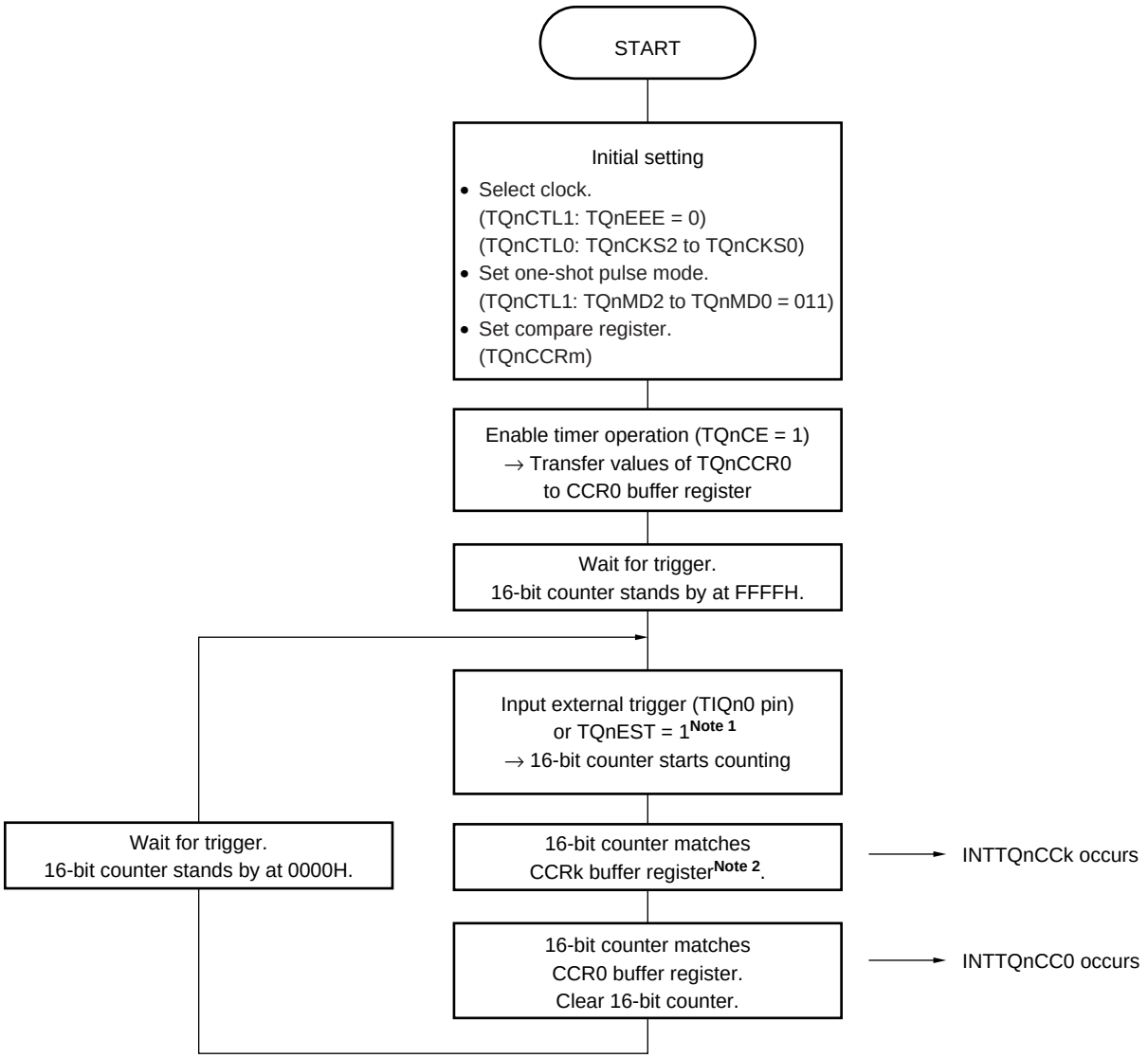
2. In the one-shot pulse mode, the TQnCCRm register is used only as a compare register. It cannot be used as a capture register.

3. When the set value of TQnCCRk is larger than the set value of TQnCCR0 in the one-shot pulse mode, in the one-shot pulse is not output.

Remarks 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE=1), refer to 8. 5. 1 Anytime write.

2. n = 0 to 2, m = 0 to 3, k = 1 to 3.

Figure 8-12. Flowchart of Basic Operation in One-Shot Pulse Mode



Notes 1. Only TQnEST bit of TQnCT register can be rewritten during timer operation (TQnCE = 1).

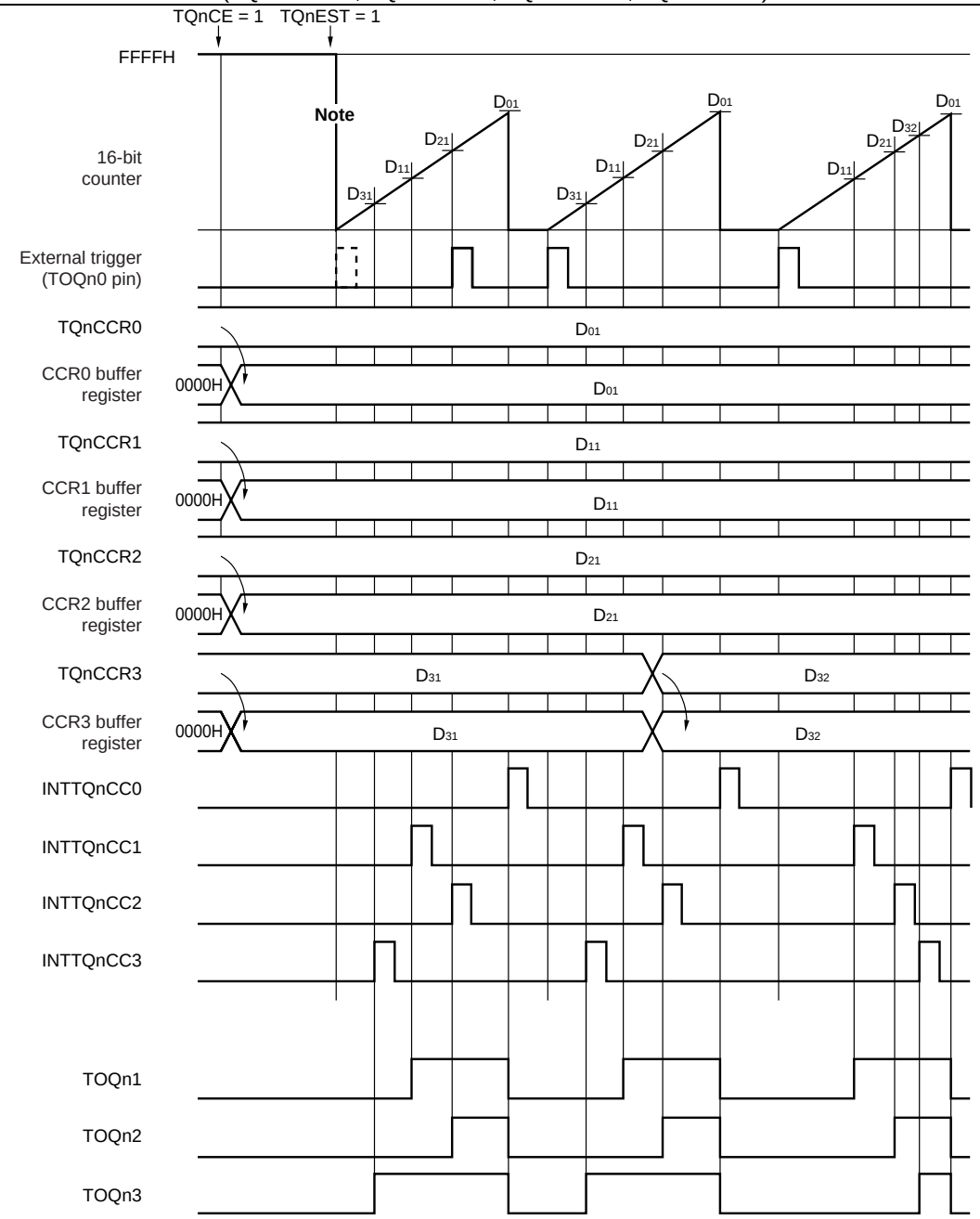
2. The 16-bit counter is not cleared when it matches the CCRk buffer register.

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 2, m = 0 to 3, k = 1 to 3

Figure 8-13. Timing of Basic Operation in One-Shot Pulse Mode

(TQnOE0 = 0, TQnOEk = 1, TQnOL0 = 0, TQnOLk = 0)



Note The 16-bit counter starts counting up either when TQnEST = 1 or when an external trigger (TOQn0 pin) is input.

Remarks 1. D_{01}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}, D_{32}: Set value of TQnCCR3 register (0000H to FFFFH)

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

3. Output delay time = (Set value of TQnCCRk register) × (Count clock cycle)

Active level width = (Set value of TQnCCR0 register - Set value of TQnCCRk register +1) × (Count clock cycle)

8.5.6 PWM mode (TQnMD2 to TQnMD0 = 100)

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.

To stop timer Q, clear TQnCE to 0. The waveform of PWM is output from the TOQnk pin. The TOQn0 pin produces a half pulse output of PWM cycle when the 16-bit counter matches the TQnCCR0 register.

Remarks 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE = 1), refer to **8.5.1 (2) Reload**.

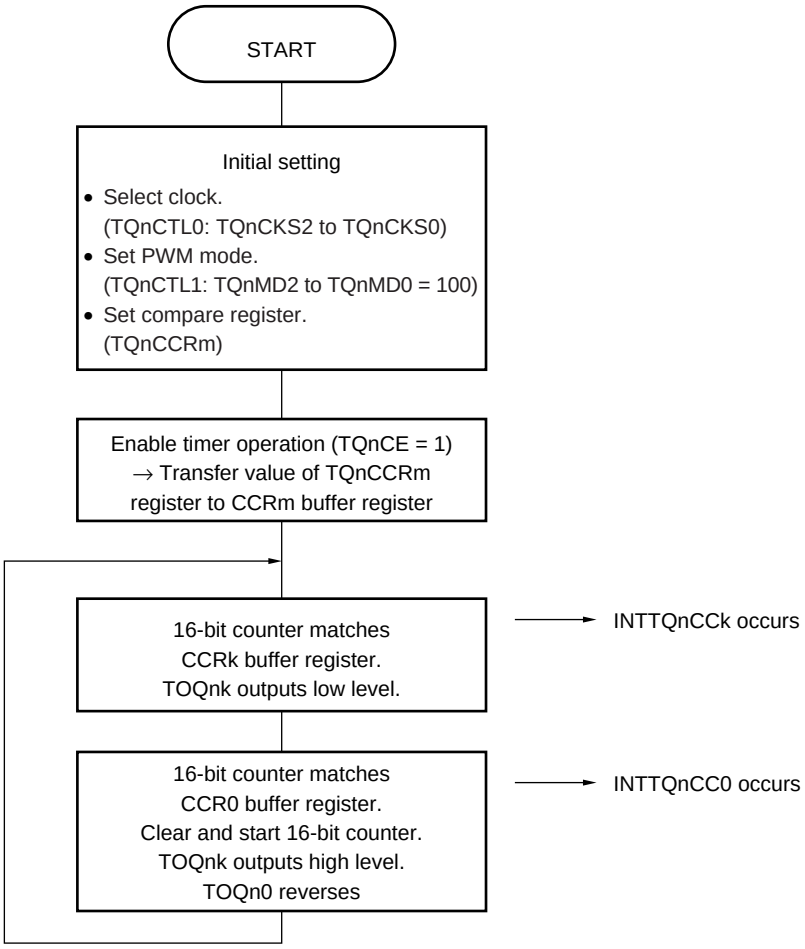
2. n = 0 to 2, m = 0 to 3, k = 1 to 3

Caution: In the PWM mode, the TQnCCRm register is used only as a compare register. It cannot be used as a capture register.

(1) Operation flow of PWM mode

Figure 8-14. Flowchart of Basic Operation in PWM Mode (1/2)

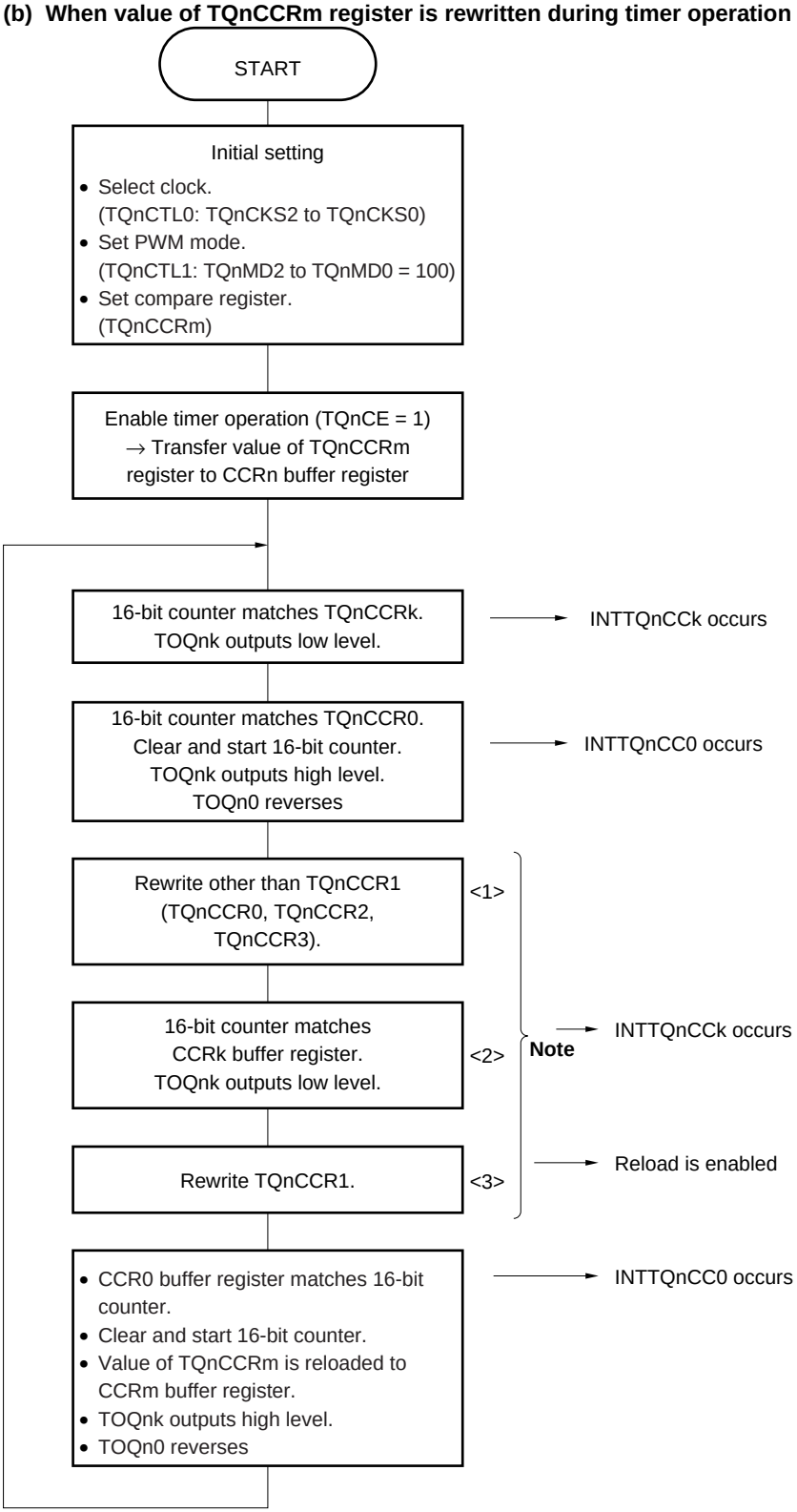
(a) When values of TQnCCRm register is not rewritten during timer operation



Remark n = 0 to 2, m = 0 to 3, k = 1 to 3

Figure 8-14. Flowchart of Basic Operation in PWM Mode (2/2)

(b) When value of TQnCCRm register is rewritten during timer operation



Note The timing of <2> may differ depending on the rewrite timing of <1> and <3> and the value of TQnCCRk, but make sure that <3> comes after <1>.

Remark n = 0 to 2, m = 0 to 3, k = 1 to 3

(2) PWM output mode operation timing

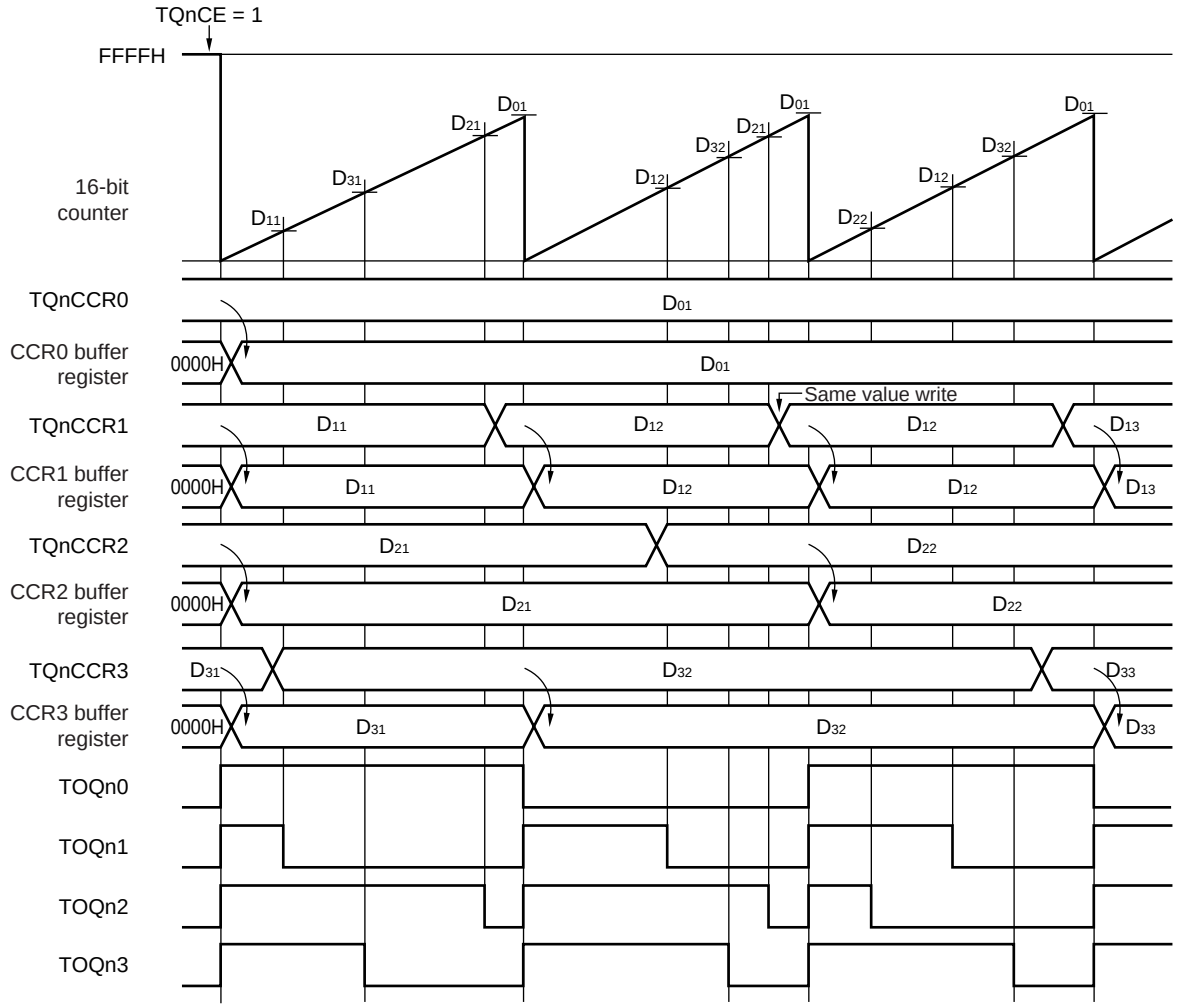
(a) Change of pulse width during operation

When change of PWM waveform during operation, please write to TQ0CCR1 register at last. After write to TQ0CCR1 register, when write to TPnCCR0 register again, please rewrite after detection of INTTQ0CC1 signal.

Figure 8-15. Timing of Basic Operation in PWM Mode (1/2)

(a) When rewriting values of TQnCCR1 to TQnCCR3 registers

(TQnOE0 = 1, TQnOEk = 1, TQnOL0 = 0, TQnOLk = 0)



Remarks 1. D_{01}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{11}, D_{12}, D_{13}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{21}, D_{22}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{31}, D_{32}, D_{33}: Set value of TQnCCR3 register (0000H to FFFFH)

2. Duty of TOQnk output = (Set value of TQnCCRk register) / (Set value of TQnCCR0 register + 1)

Cycle of TOQnk output = (Set value of TQnCCR0 register) × (Count clock cycle)

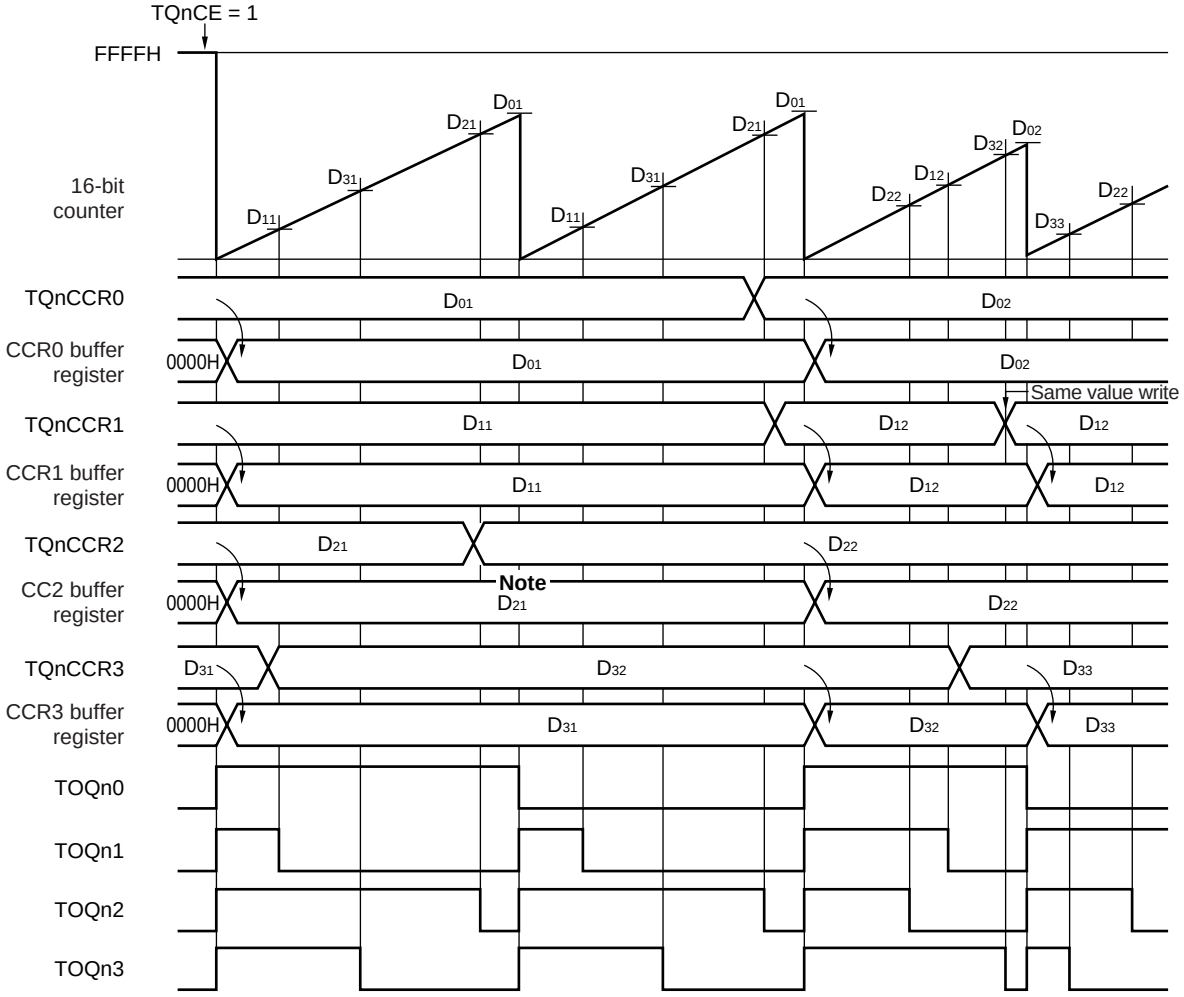
Toggle width of TOQn0 output = (Set value of TQnCCR0 register + 1) × (Count clock cycle)

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

Figure 8-15. Timing of Basic Operation in PWM Mode (2/2)

(b) When rewriting values of TQnCCR0 to TQnCCR3 registers

(TQnOE0 = 1, TQnOEk = 1, TQnOL0 = 0, TQnOLk = 0)



Note No value is reloaded because the TQnCCR1 register is not rewritten.

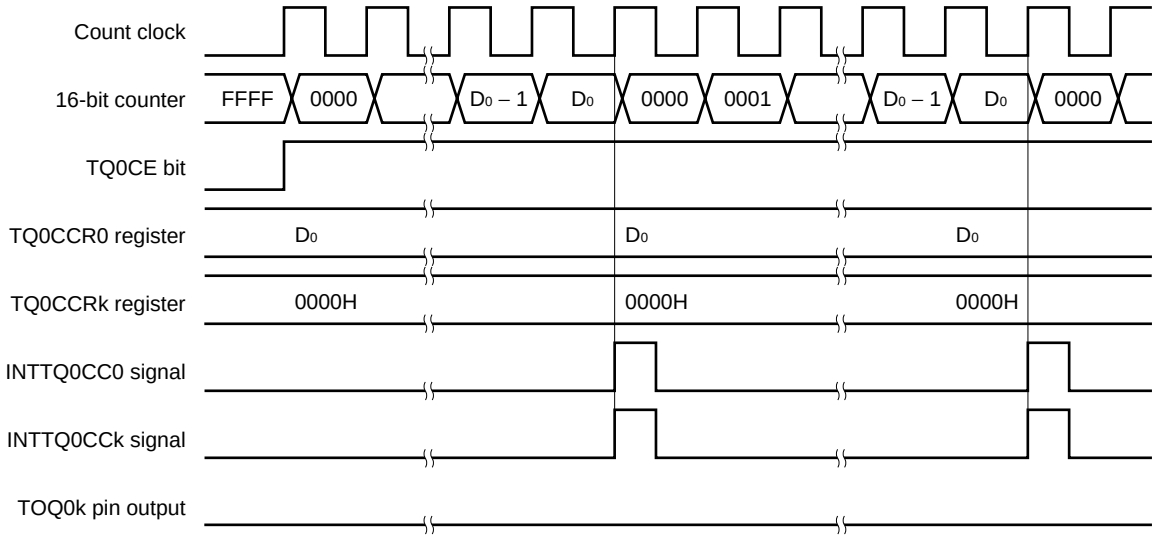
Remarks 1. D_{01}, D_{02}: Set value of TQnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Set value of TQnCCR1 register (0000H to FFFFH)
D_{21}, D_{22}: Set value of TQnCCR2 register (0000H to FFFFH)
D_{31}, D_{32}, D_{33}: Set value of TQnCCR3 register (0000H to FFFFH)

2. Duty of TOQnk output = (Set value of TQnCCRk register) / (Set value of TQnCCR0 register + 1)
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 2, k = 1 to 3

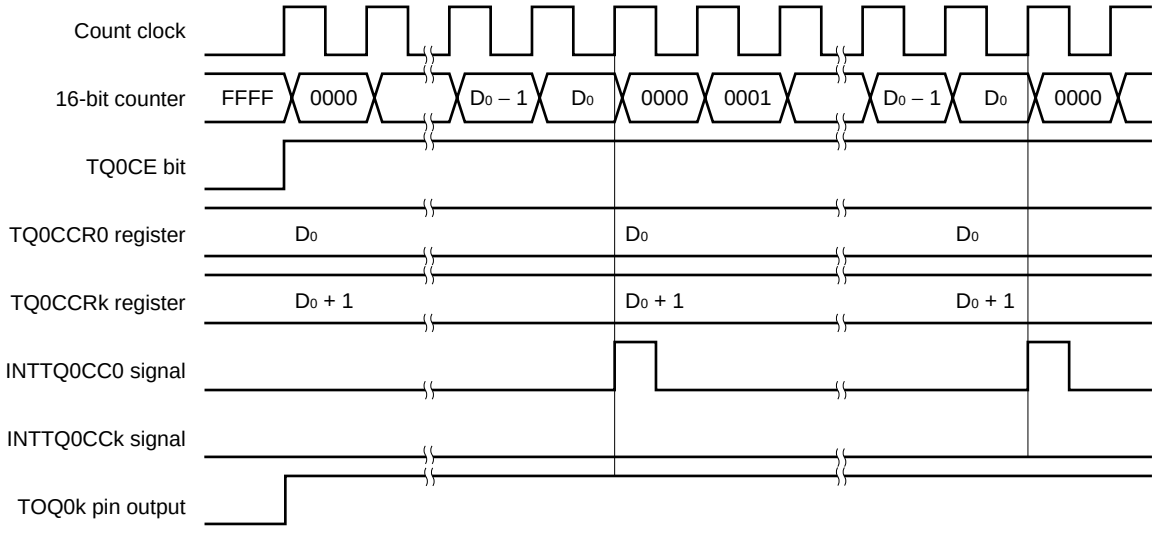
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TQ0CCRk register to 0000H. If the set value of the TQ0CCR0 register is FFFFH, the INTTQ0CCk signal is generated periodically.



Remark k = 1 to 3

To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRk register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.



Remark k = 1 to 3

8.5.7 Free-running mode (TQnMD2 to TQnMD0 = 101)

In the free-running mode, the 16-bit counter free-runs, and the bit that selects the capture or compare register function can be by the setting of the TQnCCS3 to TQnCCS0 bits, so that an interval function and a capture function can be realized.

Setting of the TQnCCS3 to TQnCCS0 bits of the TQnOPT0 register is valid only in the free-running mode.

Caution: In the free-running mode, counter clear can be operated by matched compare register.

TQnCCSm	Operation
0	TQnCCRm register is used as compare register.
1	TQnCCRm register is used 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.

  TQnCCRm register is enabled for write operation when TPnCE=1. Any data is set to TPnCCR1 register by anytime write, data is translated to CCRm buffer register, and data become comparison value with value of the 16 bit counter.

  Caution: External event count input as count clock (TQnCTL.TQnEEE=1), TQnCCR0 register can not to use as capture register.

  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.

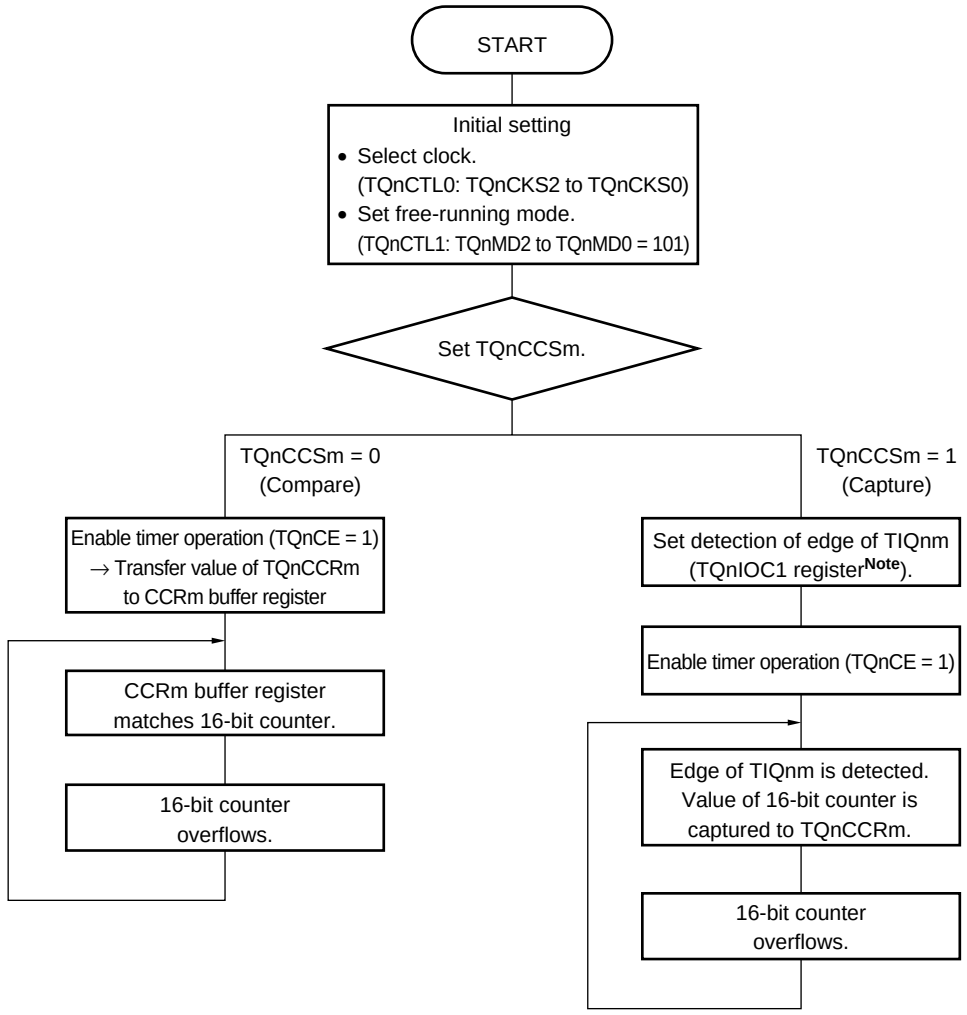
- When TQnCCRm register is used as capture register

  The value of the 16-bit counter is stored in the TQnCCRm register when the edge of the TIQnm pin is detected.

Remarks 1. For the rewritten TQnCCR0 to TQnCCR3 during timer operation, refer to **8.5.1 (1) Anytime write**.

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

Figure 8-16. Flowchart of Basic Operation in Free-Running Mode



Note TQnCCR0 edge detection: TQnIS1, TQnIS0 bits
TQnCCR1 edge detection: TQnIS3, TQnIS2 bits
TQnCCR2 edge detection: TQnIS5, TQnIS4 bits
TQnCCR3 edge detection: TQnIS7, TQnIS6 bits

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

(1) When TQnCCSm = 0 (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 a one-shot pulse trigger is input in this mode, a 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 TQnCCSm = 1 (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).

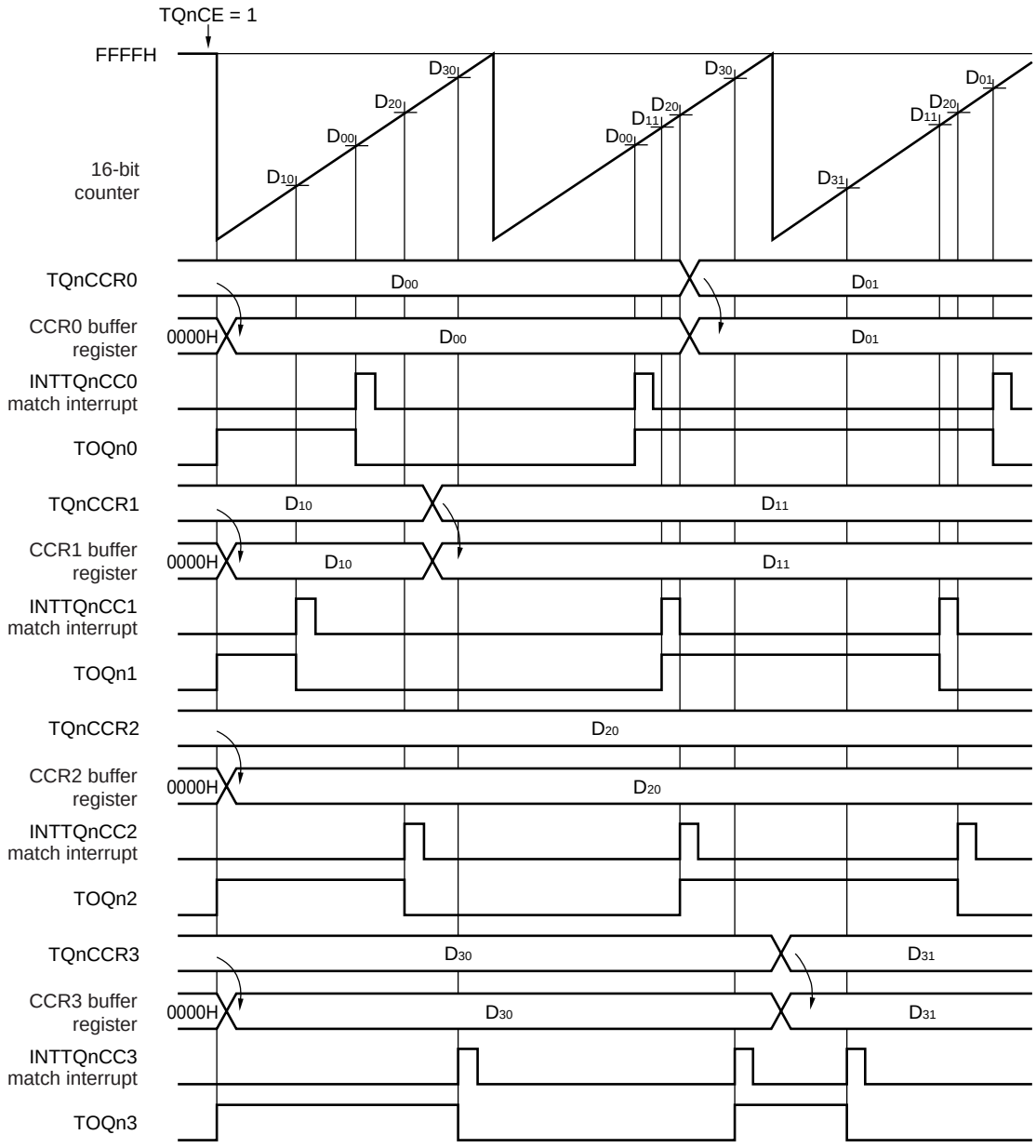
However, if the interval of the capture trigger is such that the overflow occurs two times (two periods of more of free-running), the TQnOVF flag cannot be used for judgment.

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

Figure 8-17. Timing of Basic Operation in Free-Running Mode (1/4)

(a) TQnCCS3 = 0, TQnCCS2 = 0, TQnCCS1 = 0, TQnCCS0 = 0

(TQnOEm = 1, TQnOLm = 0)



Remarks 1. D_{00}, D_{01}: Set value of TQnCCR0 register (0000H to FFFFH)

D_{10}, D_{11}: Set value of TQnCCR1 register (0000H to FFFFH)

D_{20}: Set value of TQnCCR2 register (0000H to FFFFH)

D_{30}, D_{31}: Set value of TQnCCR3 register (0000H to FFFFH)

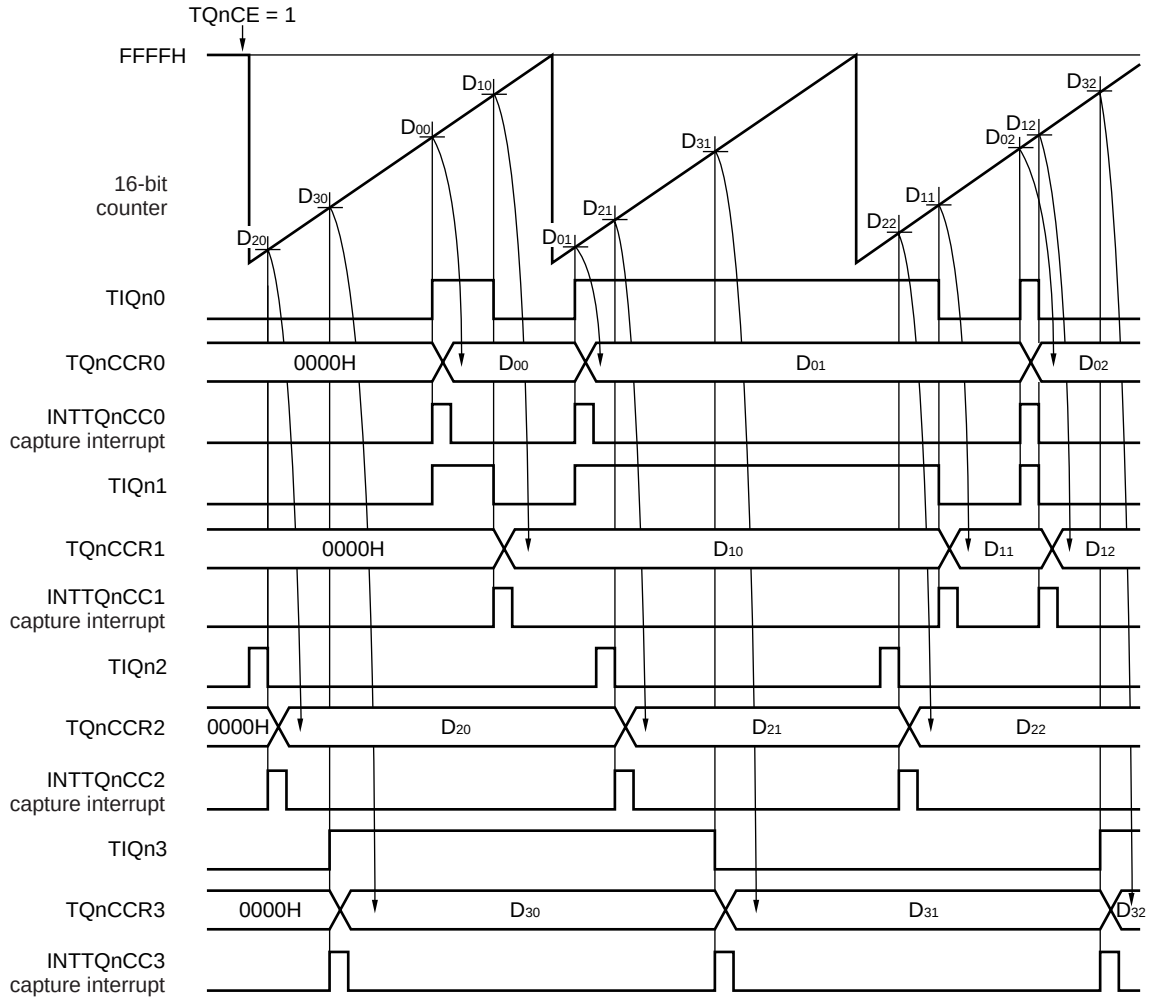
2. TOQnm output goes high when counting is started.

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

Figure 8-17. Timing of Basic Operation in Free-Running Mode (2/4)

(b) TQnCCS3 = 1, TQnCCS2 = 1, TQnCCS1 = 1, TQnCCS0 = 1

(TQnOEm = 0, TQnOLm = 0)



Remarks 1. D_{00}, D_{01}, D_{02}: Value captured to TQnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}: Value captured to TQnCCR1 register (0000H to FFFFH)
D_{20}, D_{21}, D_{22}: Value captured to TQnCCR2 register (0000H to FFFFH)
D_{30}, D_{31}, D_{32}: Value captured to TQnCCR3 register (0000H to FFFFH)

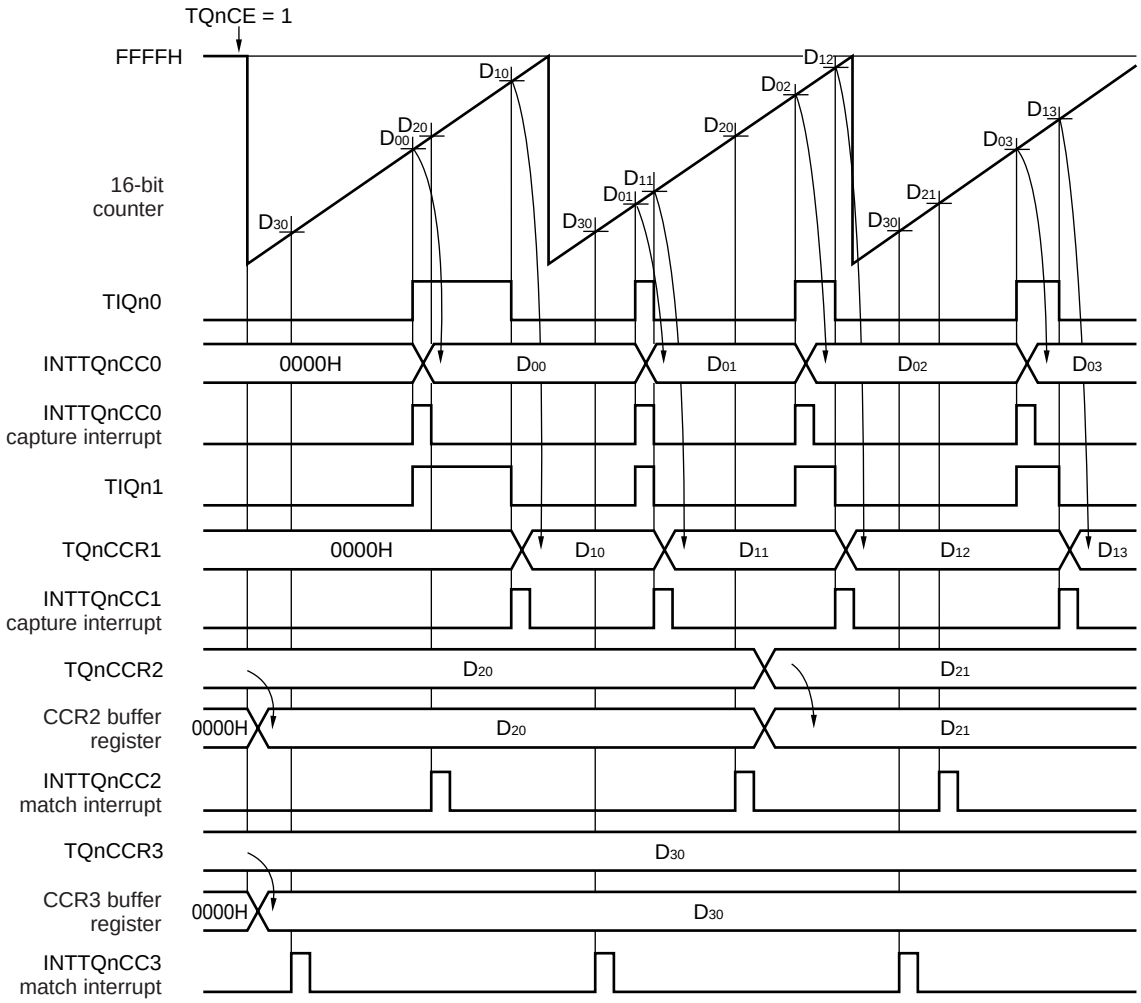
2. TIQn0: Detection of rising edge (TQnIS1, TQnIS0 = 01) is set.
TIQn1: Detection of falling edge (TQnIS3, TQnIS2 = 10) is set.
TIQn2: Detection of falling edge (TQnIS5, TQnIS4 = 10) is set.
TIQn3: Detection of both rising and falling edges (TQnIS7, TQnIS6 = 11) is set.

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

Figure 8-17. Timing of Basic Operation in Free-Running Mode (3/4)

(c) TQnCCS3 = 0, TQnCCS2 = 0, TQnCCS1 = 1, TQnCCS0 = 1

(TQnOEm = 0, TQnOLm = 0)



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TQnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Value captured to TQnCCR1 register (0000H to FFFFH)
D_{20}, D_{21}: Value captured to TQnCCR2 register (0000H to FFFFH)
D_{30}: Value captured to TQnCCR3 register (0000H to FFFFH)

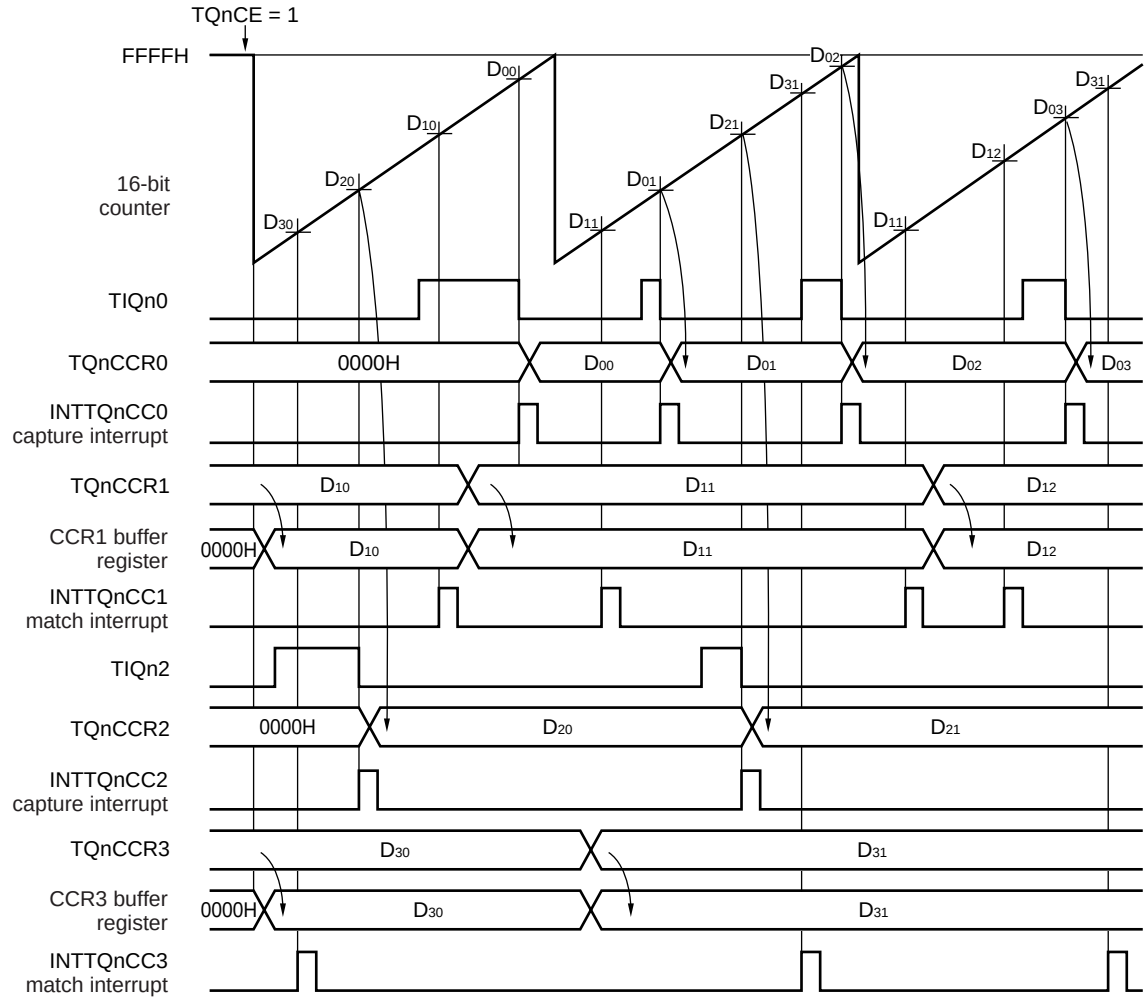
2. TIQn0: Detection of rising edge (TQnIS1, TQnIS0 = 01) is set.
TIQn1: Detection of falling edge (TQnIS3, TQnIS2 = 10) is set.

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

Figure 8-17. Timing of Basic Operation in Free-Running Mode (4/4)

(d) TQnCCS3 = 0, TQnCCS2 = 1, TQnCCS1 = 0, TQnCCS0 = 1

(TQnOEm = 0, TQnOLm = 0)



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TQnCCR0 register (0000H to FFFFH)

D_{10}, D_{11}, D_{12}: Value captured to TQnCCR1 register (0000H to FFFFH)

D_{20}, D_{21}: Value captured to TQnCCR2 register (0000H to FFFFH)

D_{30}, D_{31}: Value captured to TQnCCR3 register (0000H to FFFFH)

2. TIQn0: Detection of rising edge (TQnIS1, TQnIS0 = 10) is set.

TIQn2: Detection of falling edge (TQnIS5, TQnIS4 = 10) is set.

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

(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 generated.

The overflow flag is cleared by the CPU by writing 0 to it.

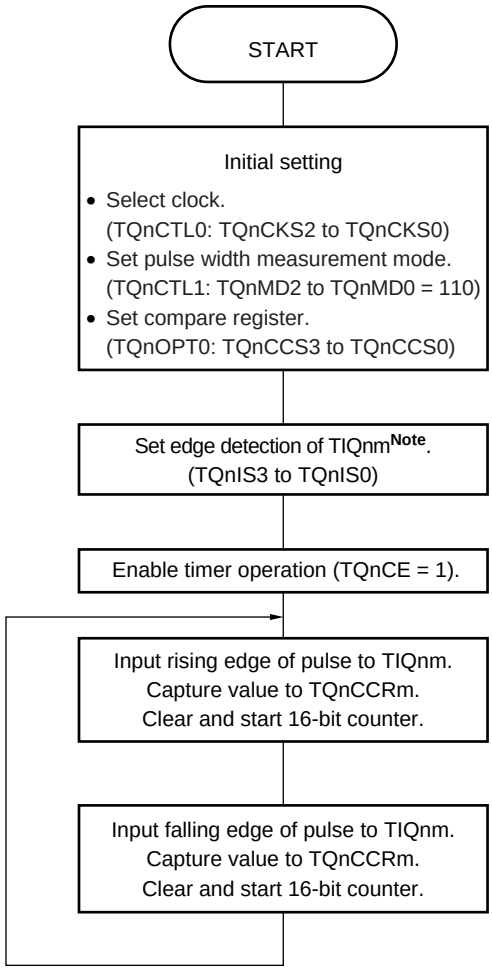
8.5.8 Pulse width measurement mode (TQnMD2 to TQnMD0 = 110)

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. For measurement a pulse width that causes overflow to occur twice or more, please count the overflow number with the overflow interrupt.

Caution In the pulse width measurement mode, select the internal clock (TQnEEE of the TQnCTL1 register = 0) as the count clock.

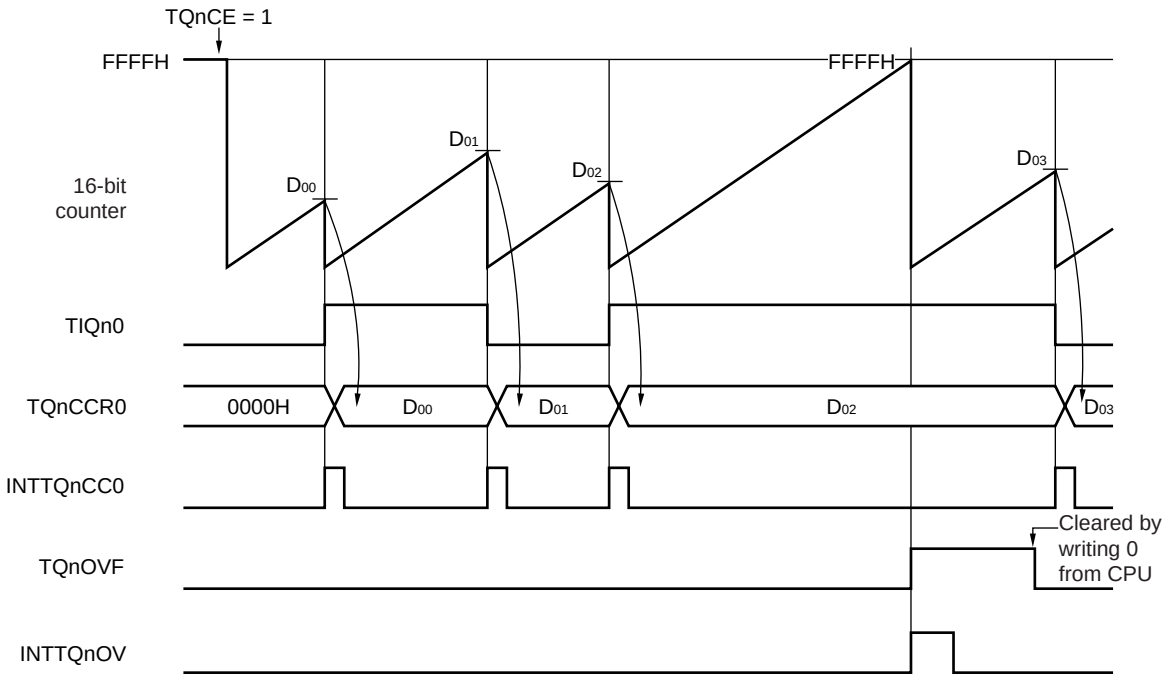
Figure 8-18 Flowchart of Basic Operation in Pulse Width Measurement Mode



Note An external pulse can be input from any of TIQn0 to TIQn3. 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 2, m = 0 to 3

Figure 8-19. Timing of Basic Operation in Pulse Width Measurement Mode



Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Value captured to TQnCCR0 register (0000H to FFFFH)

2. TIQn0: Both the rising and falling edges are detected.

3. n = 0 to 2

4. Pulse width = Captured value × Count clock cycle

If the valid edge is not input even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTQnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TQnOPT0.TQnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TQnOVF bit set (1) count + Captured value) × Count clock cycle

8.5.9 Triangular wave PWM mode (TQnMD2 to TQnMD0 = 111)

In the triangular wave 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 four registers and operating the timer, triangular wave PWM with a variable cycle is output.

The value of the TQnCCRm register can be rewritten when TQnCE = 1.

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.

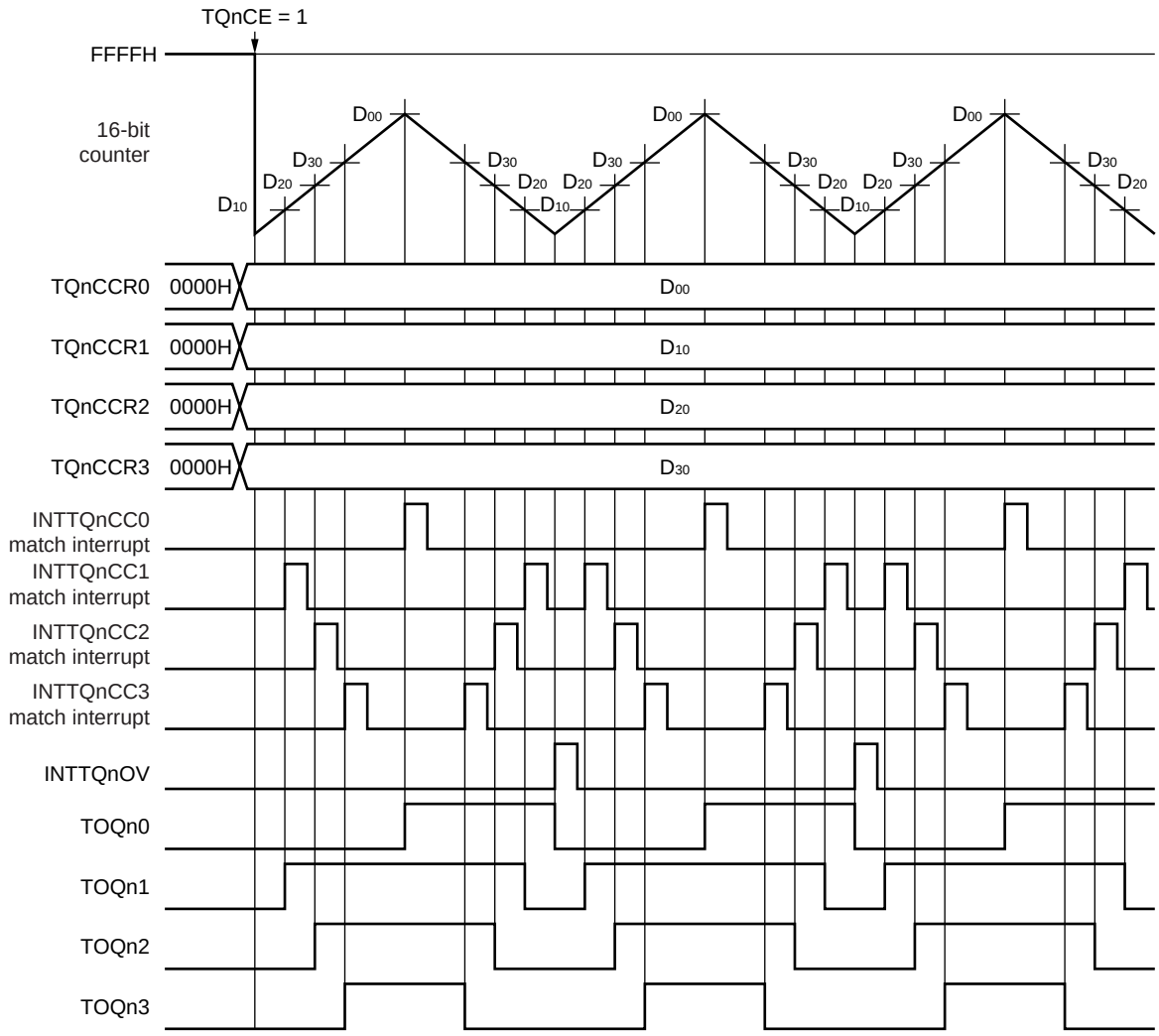
To stop timer Q, clear TQnCE to 0. The waveform of PWM is output from the TOQnk pin. The TOQn0 pin produces a toggle output when the value of the 16-bit counter matches the value of the TQnCCR0 register and when the counter underflows.

Remarks: 1. For the rewriting TQnCCR0 to TQnCCR3 during timer operation (TQnCE=1), refer to **8. 5. 1 (2) Reload**.

2. n = 0 to 2, m = 0 to 3, k = 1 to 3

Caution: In the PWM mode, the TQnCCRm register is used only as a compare register. It cannot be used as a capture register.

Figure 8-20. Timing of Basic Operation in Triangular Wave PWM Mode



Remark n = 0 to 2

8.6 Timer Synchronized 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 8-4.

Table 8-4. Tuned Operation Mode of Timers

Master Timer	Slave Timer	
TMP0	TMP1	–
TMP2	TMP3	TMQ0
TMQ1	TMQ2	–

Cautions 1. The tuned operation mode is enabled or disabled by the TPmSYE bit of the TPmCTL1 register and TQnSYE bit of the TQnCTL1 register. For TMQ2, either or both TMQ3 and TMQ0 can be specified as slaves.

2. Set the tuned operation mode using the following procedure.

<1> Set the TPmSYE bit of the TPmCTL1 register and the TQnSYE bit of the TQnCTL1 register of the slave timer to enable the tuned operation.
Set the TPmMD2 to TPmMD0 bits of the TPmCTL1 register and TQnMD2 to TQnMD0 bits of the TQnCTL1 register of the slave timer to the free-running mode

<2> 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.

<3> Set the compare register value of the master and slave timers.

<4> Set the TPmCE bit of the TPmCTL0 register and the TQnCE bit of the TQnCTL0 register of the slave timer to enable operation on the internal operating clock.

<5> 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.

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

Tables 8-5 and 8-6 show the timer modes that can be used in the tuned operation mode (√: Settable, ×: Not settable).

Table 8-5. Timer Modes Usable in Tuned Operation Mode

Master Timer	Free-Running Mode	PWM Mode	Triangular Wave PWM Mode
TMP0	√	√	×
TMP2	√	√	×
TMQ1	√	√	√

Table 8-6. Timer Output Functions (1/2)

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	PPG	←	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	←

Table 8-7. Timer Output Functions (2/2)

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
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	←
	TMQ2 (slave)	TOQ20	PPG	←	Toggle	PWM	Toggle	Triangular wave PWM
		TOQ21 to TOQ23	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 2)

Figure 8-21. Tuned Operation Image (TMP2, TMP3, TMQ0)

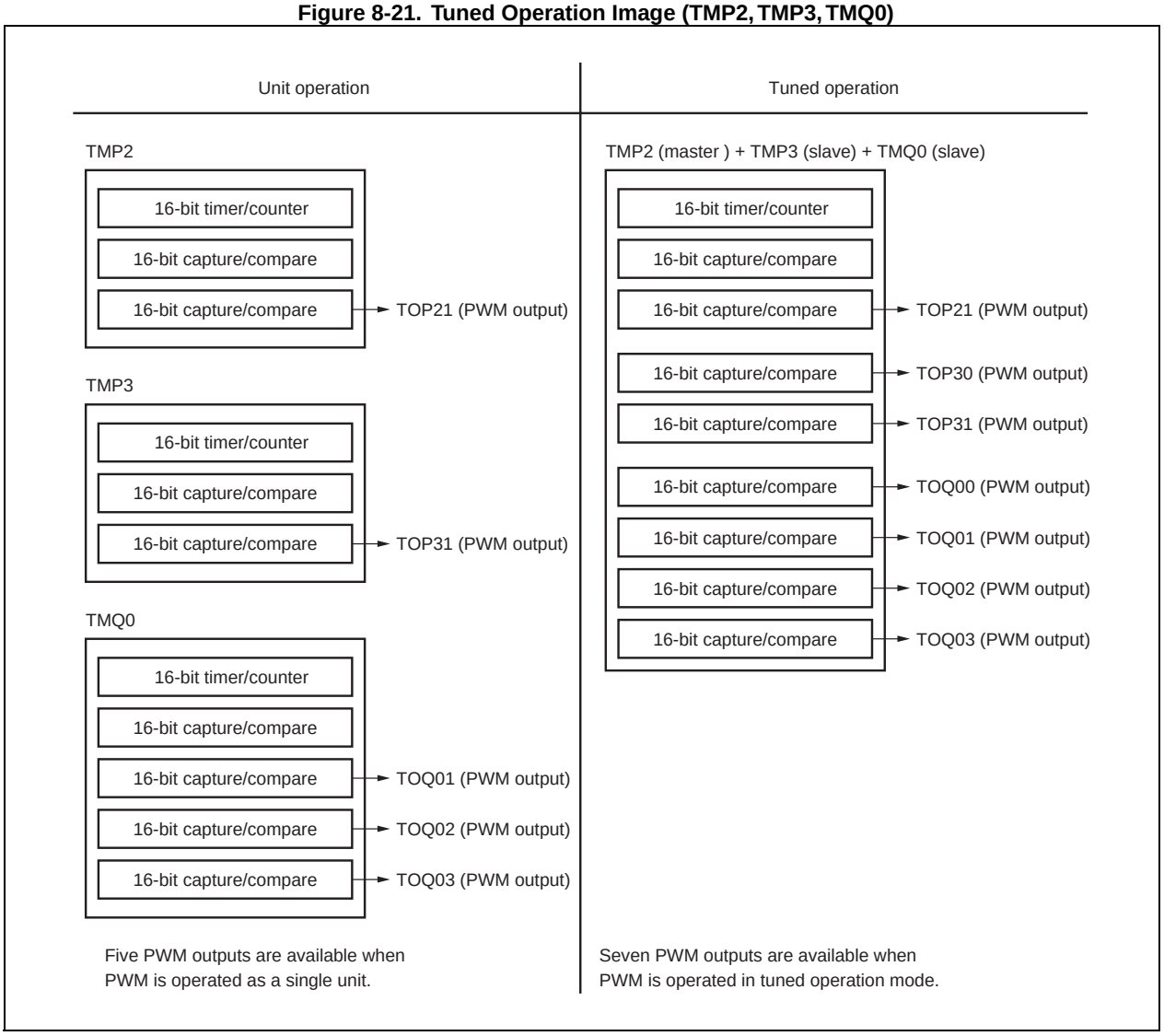
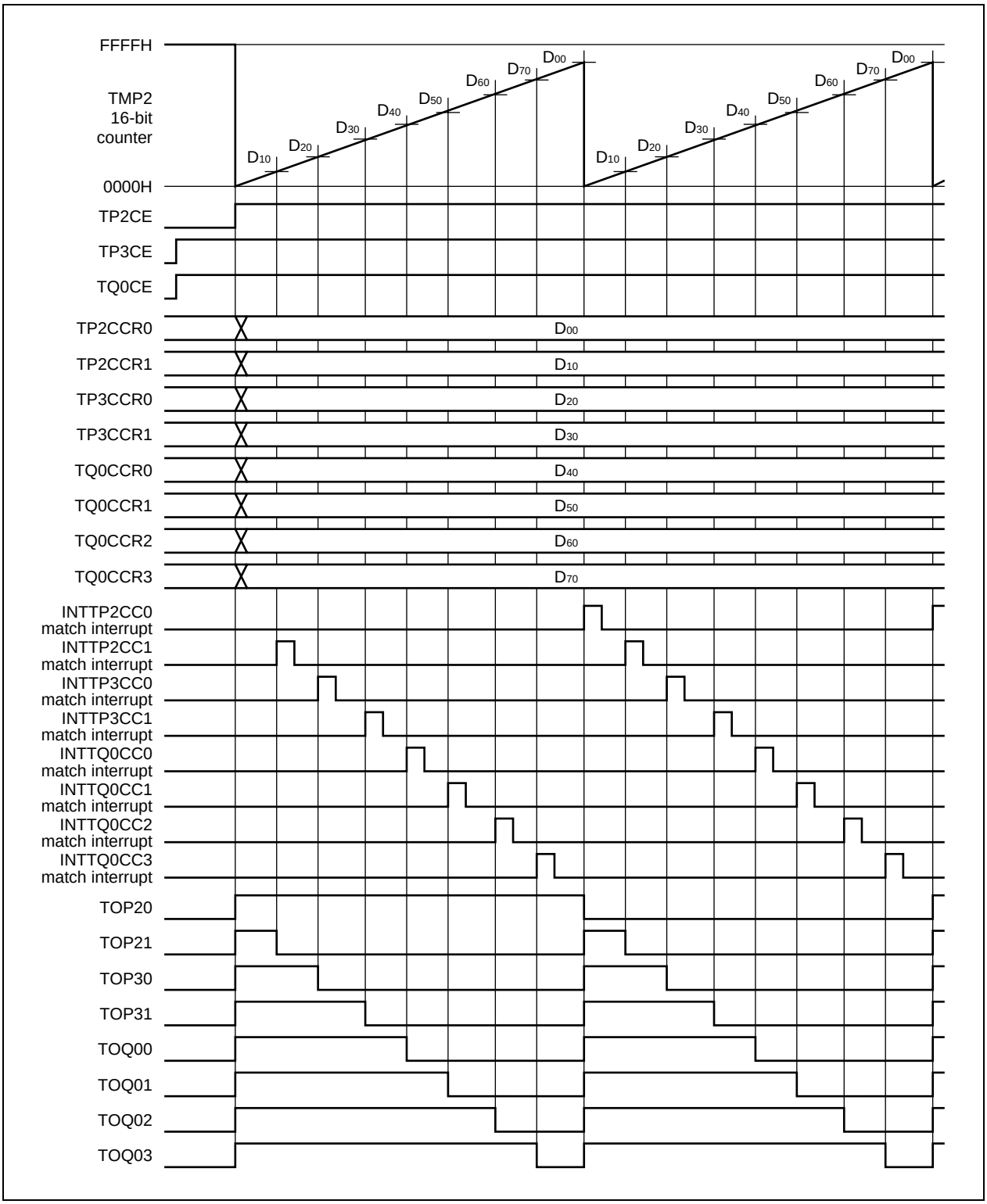


Figure 8-22. Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)

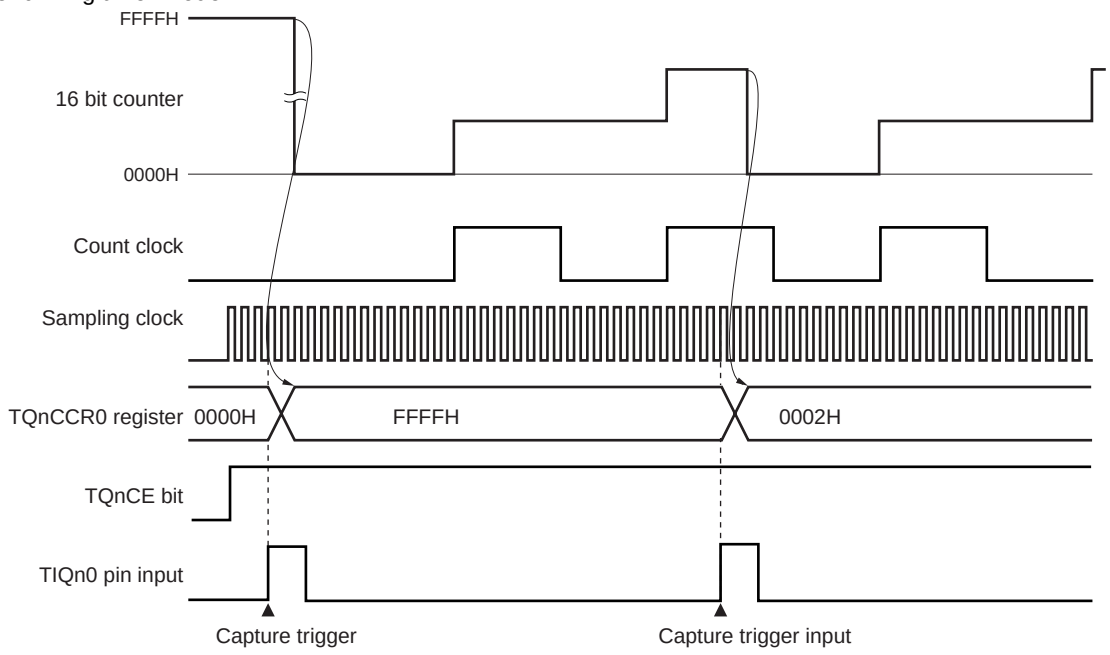


8.7 Cautions

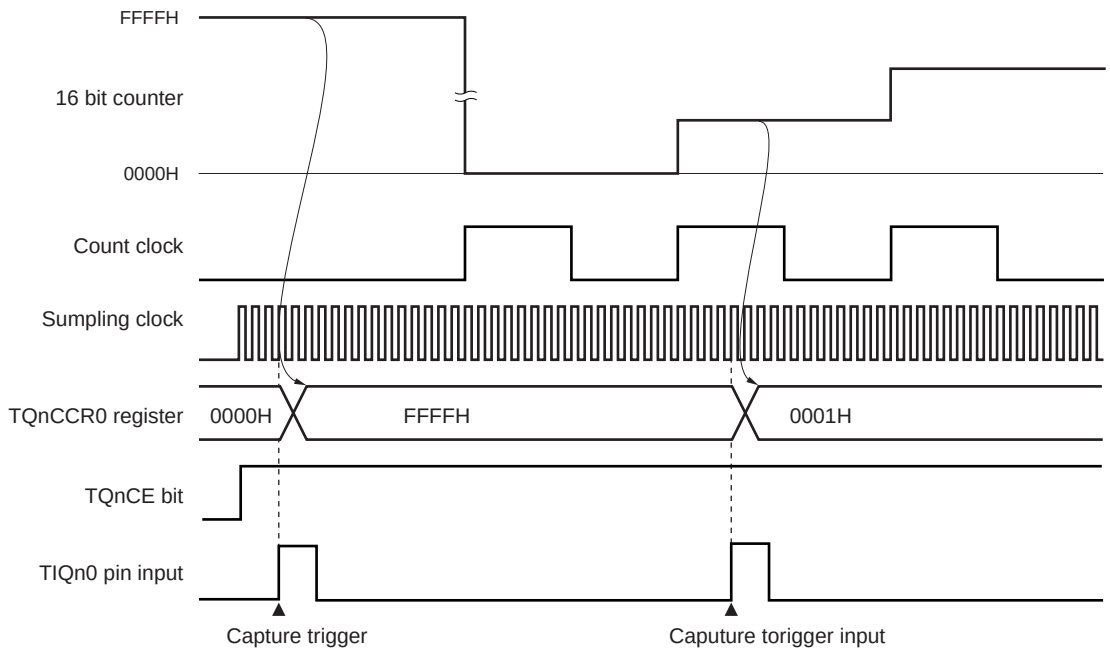
(1) Capture operation

When the capture signal occurs before the count clock is available and the selected count clock is slower than the internal sampling signal, the value in the capture register is FFFFH (instead of 0000H).

(a)Free running timer mode



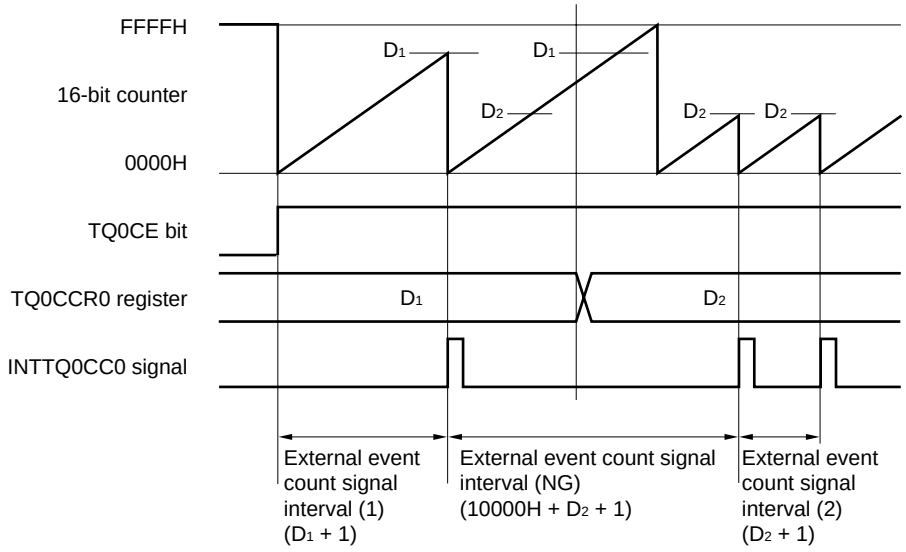
(b)Pulse mode



(2) Notes on rewriting the TQ0CCR0 register

To change the value of the TQ0CCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



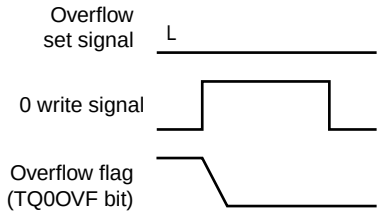
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTQ0CC0 signal is generated. Therefore, the INTTQ0CC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2+ 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

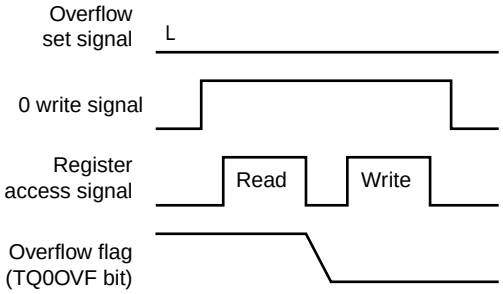
(3) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TQ0OVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TQ0OPT0 register. To accurately detect an overflow, read the TQ0OVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

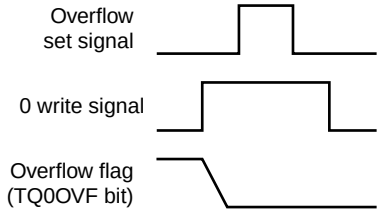
(i) Operation to write 0 (without conflict with setting)



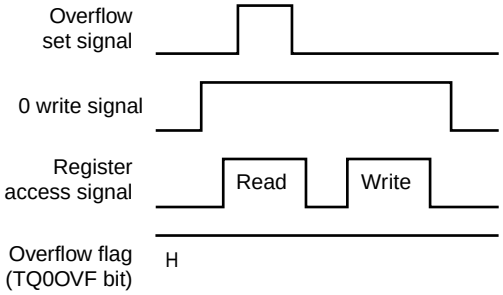
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

CHAPTER 9 16-BIT INTERVAL TIMER M

The V850ES/Fx2 include a 16-bit interval timer M (TMM0).

Table 9-1. Number of Channels of Timer M

Product	Number of Channels
V850ES/FE2	1 channel (TMM)
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	

9.1 Features

Timer M (TMM) 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
- Clock selection × 8
- Simple counter × 1
  (The simple counter is a counter that does not use a counter read buffer. This counter cannot be read during timer count operation.)
- Simple compare × 1
  (The simple compare register is a register that does not use a compare write buffer. No data can be written to this compare register during timer count operation.)
- Compare match interrupt × 1

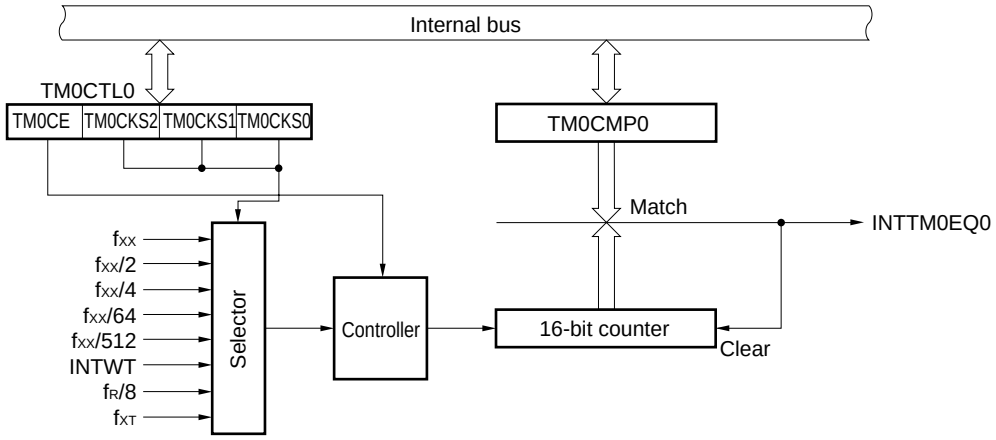
9.2 Configuration

TMM consists of the following hardware.

Table 9-2. Configuration of TMM

Item	Configuration
Timer register	16-bit counter
Register	TMM compare register 0 (TM0CMP0)
Control register	TMM0 control register (TM0CTL0)

Figure 9-1. Block Diagram of Timer M



Remark f_{XX}: Main clock frequency
f_R: Internal oscillation clock frequency
f_{XT}: Subclock frequency
INTWT: Watch timer interrupt request signal

(1) 16-bit counter

This is a 16-bit counter that counts the internal clock.

The 16-bit counter cannot be read or written.

(2) TMM0 compare register 0 (TM0CMP0)

The TM0CMP0 register is a 16-bit compare register.

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

Reset input clears this register to 0000H.

The same value can always be written to the TM0CMP0 register by software.

After reset: 0000H R/W Address: TM0CMP0: FFFFF694H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0CMP0																

Caution: Rewriting the TM0CMP0 register is prohibited while the timer is working (TM0CE = 1). But The same value can be rewritten.

9.3 Control Register

(1) TMM0 control register 0 (TM0CTL0)

The TM0CTL0 register is an 8-bit register that controls the operation of TMM.

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

Reset input clears this register to 00H.

Rewriting the TM0CTL0 register is prohibited while the timer is working. Only the TM0CE bit can always be rewritten.

(1/2)

After reset: 00H R/W Address: TM0CTL0: FFFFF690H

	7	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	Control of operation of timer M0
0	Disable internal operating clock operation (asynchronously reset TMM0).
1	Enable internal operating clock operation.

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.

TM0CKS2	TM0CKS1	TM0CKS0	Selection of internal count clock
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/64$
1	0	0	$f_{XX}/512$
1	0	1	INTWT
1	1	0	$f_R/8$
1	1	1	f_{XT}

Cautions: 1. Set TM0CKS2 to TM0CKS0 bits at TM0CE = 0. When the TM0CE bit is set from 0 to 1, the TM0CKS2 to TM0CKS0 bits can be set at the same time.

2. Set bit 6-3 to 0.

Remark f_{XX}: Main system clock frequency

f_R: Ring-OSC clock frequency

f_{XT}: Subclock frequency

(2/2)

Resolution and maximum number of counts

Internal count clock	Resolution [μs]		Maximum count time [ms]	
	f_{XX} = 16 MHz	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 20 MHz
f_{XX}	0.0625	0.050	4.10	3.28
$f_{XX}/2$	0.125	0.100	8.19	6.55
$f_{XX}/4$	0.250	0.200	16.38	13.11
$f_{XX}/64$	4.000	3.200	262.14	209.72
$f_{XX}/512$	32.000	25.600	2097.15	1677.72

Internal count clock	Resolution [μs]			Maximum count time [ms]		
	f_R = 100 kHz (Min.)	f_R = 200 kHz (Typ.)	f_R = 400 kHz (Max.)	f_R = 100 kHz (Min.)	f_R = 200 kHz (Typ.)	f_R = 400 kHz (Max.)
$f_R/8$	80.0	40.0	20.0	5242.88	2621.44	1310.72

Internal count clock	Resolution [μs]	Maximum count time [ms]
	f_{XT} = 32.768 kHz	f_{XT} = 32.768 kHz
f_{XT}	30.52	2000.00

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.

Figure 9-2. Basic Timing of Operation in Interval Timer Mode

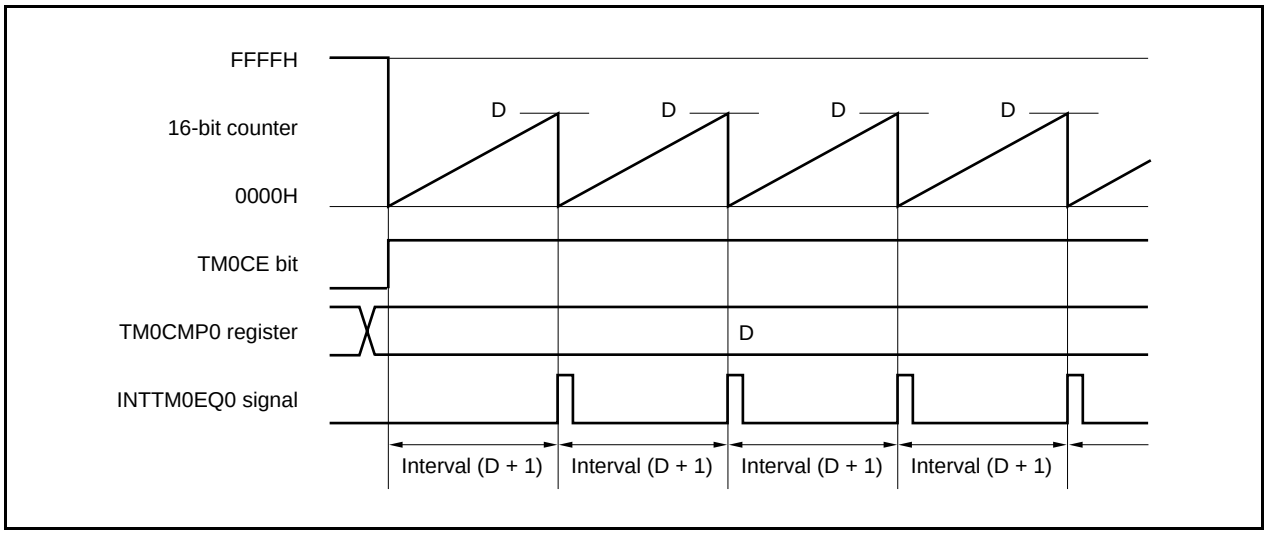
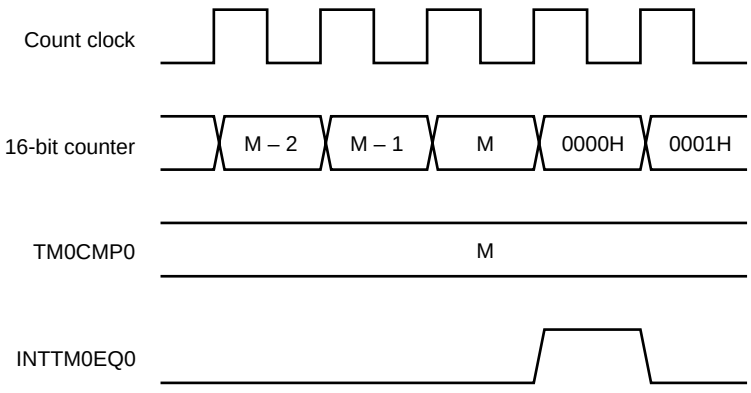


Figure 9-3. Timing of Operation in Interval Timer Mode



Caution To set M clocks as the interval period, set the TM0CMP0 register to M – 1.

When FFFFH is set to the TM0CMP0 register, timer M performs an operation similar to that in the free-running mode.

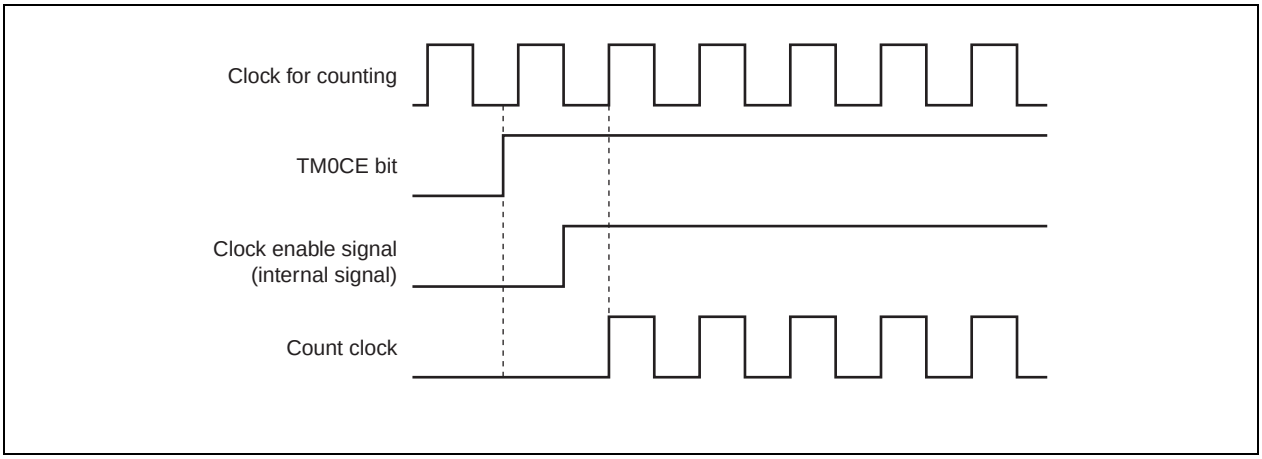
9. 5 Cautions

(1) Clock generator and clock enable timing

It takes the 16-bit counter up to the following time to start counting after the TM0CTL0.TM0CE bit is set to 1, depending on the count clock selected.

Selected Count Clock	Maximum Time Before Counting Start
f_{XX}	$2/f_{XX}$
$f_{XX}/2$	$6/f_{XX}$
$f_{XX}/4$	$24/f_{XX}$
$f_{XX}/64$	$128/f_{XX}$
$f_{XX}/512$	$1024/f_{XX}$
INTWT	Second rising edge of INTWT signal
$f_R/8$	$16/f_R$
f_{XT}	$2/f_{XT}$

Figure 9-4. Count Operation Start Timing



(2) Rewriting the TM0CMP0 and TM0CTL0 registers is prohibited while TMM0 is operating.

If these registers are rewritten while the TM0CE bit is 1, the operation cannot be guaranteed.

If they are rewritten by mistake, clear the TM0CTL0.TM0CE bit to 0, and re-set the registers.

CHAPTER 10 WATCH TIMER FUNCTIONS

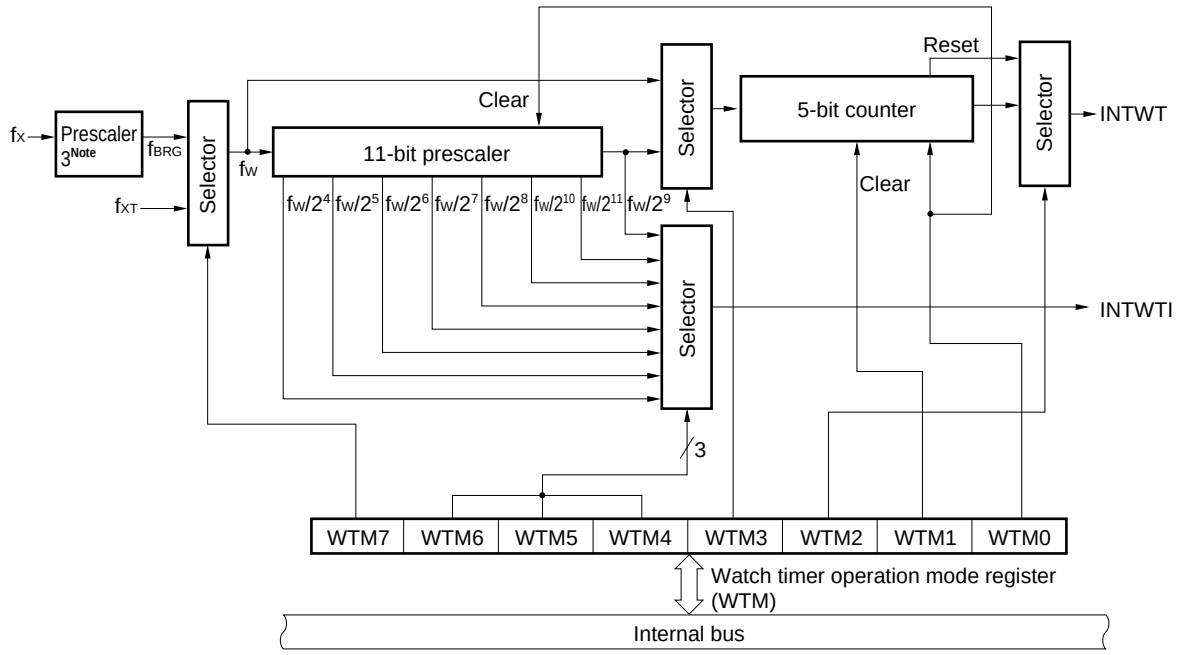
10.1 Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and interval timer functions can be used at the same time.

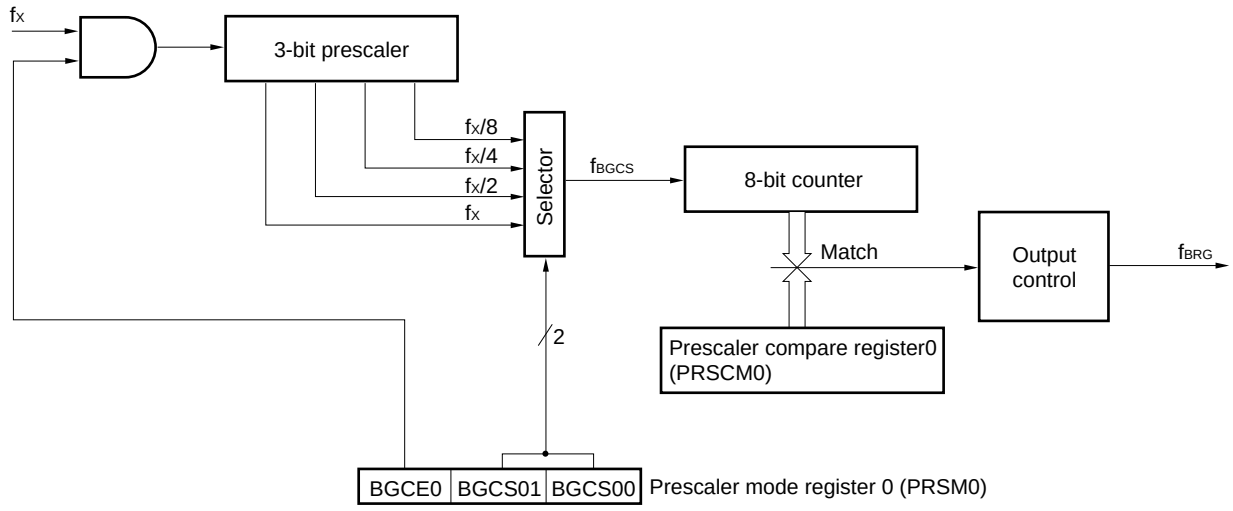
Figure 10-1. Block Diagram of Watch Timer



Note For details of prescaler 3, see **Figure 10-2 Block Diagram of Prescaler 3**.

Remark f_{BRG}: Prescaler 3 output frequency
f_X: Main clock oscillation frequency
f_{XT}: Subclock frequency
f_W: Watch timer clock frequency
INTWT: Watch timer interrupt
INTWTI: Interval timer interrupt

Figure 10-2. Block Diagram of Prescaler 3



Remark f_{BGCS}: Prescaler 3 count clock frequency
f_{BRG}: Prescaler 3 output frequency
f_X: Oscillation frequency

(1) Watch timer

The watch timer generates an interrupt request (INTWT) at time intervals of 0.5 or 0.25 seconds by using the subclock (f_{XT} = 32.768 kHz).

Caution When using a clock obtained by dividing the main clock as the watch timer count clock, set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain a divided clock frequency of 32.768 kHz.
If 32.768 kHz cannot be generated, correction using software is necessary to realize the watch function.

(2) Interval timer

The watch timer generates an interrupt request (INTWTI) at time intervals specified in advance.

Table 10-1. Interval Time of Interval Timer

Interval Time	Operation at f_W = 32.768 kHz
$2^4 \times 1/f_W$	488 μs
$2^5 \times 1/f_W$	977 μs
$2^6 \times 1/f_W$	1.95 ms
$2^7 \times 1/f_W$	3.91 ms
$2^8 \times 1/f_W$	7.81 ms
$2^9 \times 1/f_W$	15.6 ms
$2^{10} \times 1/f_W$	31.2 ms
$2^{11} \times 1/f_W$	62.5 ms

Remark f_W: Watch timer clock frequency

10.2 Configuration

The watch timer consists of the following hardware.

Table 10-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer operation mode register (WTM)

10.3 Control Registers

The watch timer operation mode register (WTM) controls the watch timer. Before operating the watch timer, set the count clock and the interval time.

(1) Watch timer operation mode register (WTM)

The WTM register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the set time of the watch flag.

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

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF680H

	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	Selection of watch timer interrupt time
0	0	0	0	$2^4/f_W$ (488 μs: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{XT}$)
0	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{XT}$)
0	0	1	1	$2^7/f_W$ (3.91 ms: $f_W = f_{XT}$)
0	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{XT}$)
0	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{XT}$)
0	1	1	0	$2^{10}/f_W$ (31.3 ms: $f_W = f_{XT}$)
0	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{XT}$)
1	0	0	0	$2^4/f_W$ (488 μs: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{BRG}$)
1	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{BRG}$)
1	0	1	1	$2^7/f_W$ (3.90 ms: $f_W = f_{BRG}$)
1	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{BRG}$)
1	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{BRG}$)
1	1	1	0	$2^{10}/f_W$ (31.2 ms: $f_W = f_{BRG}$)
1	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{BRG}$)

Remarks 1. f_W: Watch timer clock frequency

f_{XT}: Subclock frequency

f_{BRG}: Prescaler 3 output frequency

2. Values in parentheses apply to operation with f_W = 32.768 kHz

(2/2)

WTM7	WTM3	WTM2	Selection of set time of watch flag
0	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{XT}$)
0	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{XT}$)
0	1	0	$2^5/f_W$ (977 μs: $f_W = f_{XT}$)
0	1	1	$2^4/f_W$ (488 μs: $f_W = f_{XT}$)
1	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{BRG}$)
1	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{BRG}$)
1	1	0	$2^5/f_W$ (977 μs: $f_W = f_{BRG}$)
1	1	1	$2^4/f_W$ (488 μs: $f_W = f_{BRG}$)

WTM1	Control of 5-bit counter operation
0	Clears after operation stops
1	Starts

WTM0	Watch timer operation enable
0	Stops operation (clears both prescaler and 5-bit counter)
1	Enables operation

Caution Rewrite the WTM2 to WTM7 bits while both the WTM0 and WTM1 bits are 0.

Remarks 1. f_W: Watch timer clock frequency
f_{XT}: Subclock frequency
f_{BRG}: Prescaler 3 output frequency

2. Values in parentheses apply to operation with f_W = 32.768 kHz

10.4 Operation

10.4.1 Operation as watch timer

The watch timer generates an interrupt request at fixed time intervals. The watch timer operates using time intervals of 0.5 or 0.25 seconds with the subclock (32.768 kHz).

The count operation starts when the WTM1 and WTM0 bits of the WTM register are set to 11. When the WTM0 bit is cleared to 0, the 11-bit prescaler and 5-bit counter are cleared and the count operation stops.

The time of the watch timer can be adjusted by clearing the WTM1 bit to 0 and then the 5-bit counter. At this time, an error of up to 15.6 ms may occur.

The interval timer may be cleared by clearing the WTM0 bit to 0. However, because the 5-bit counter is cleared at the same time, an error of up to 0.5 seconds may occur when the watch timer overflows (INTWT).

10.4.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt at intervals specified by a preset count value.

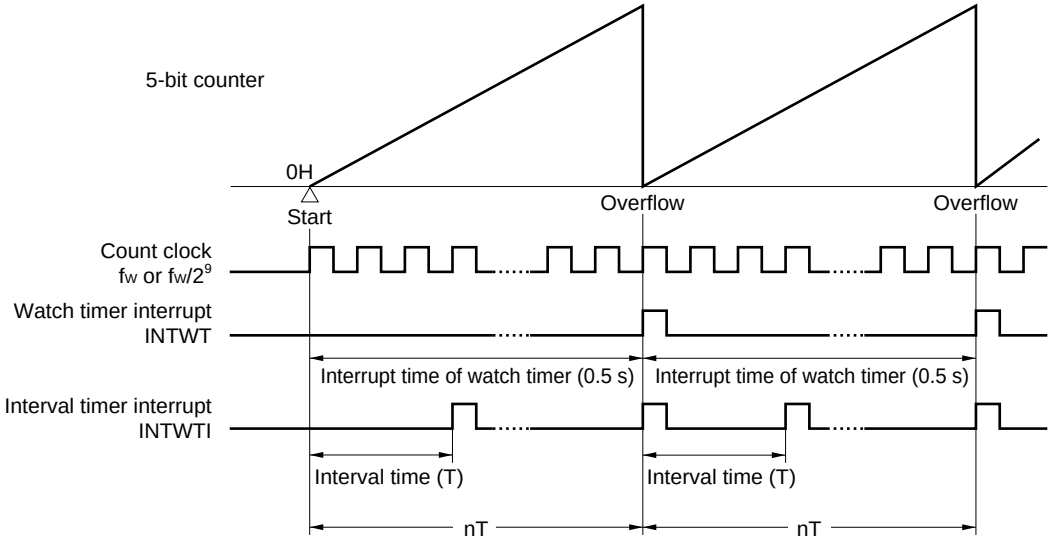
The interval time can be selected by the WTM4 to WTM7 bits of the WTM register.

Table 10-3. Interval Time of Interval Timer

WTM7	WTM6	WTM5	WTM4	Interval Time	
0	0	0	0	$2^4 \times 1/fw$	488 μs (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	0	1	$2^5 \times 1/fw$	977 μs (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
1	0	0	0	$2^4 \times 1/fw$	488 μs (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	0	1	$2^5 \times 1/fw$	977 μs (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)

Remark f_W: Watch timer clock frequency
f_{XT}: Subclock frequency
f_{BRG}: Prescaler 3 output frequency

Figure 10-3. Operation Timing of Watch Timer/Interval Timer



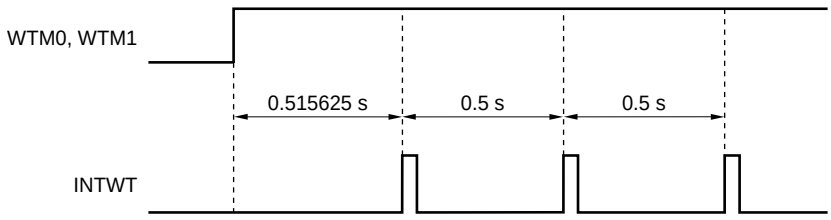
Remark fw: Watch timer clock frequency
Values in parentheses apply to operation with count clock fw = 32.768 kHz.
n: Number of interval timer operations

10.4.3 Cautions

The following time is required before the first watch timer interrupt request signal (INTWT) is generated after operation is enabled (WTM1 and WTM0 bits of WTM register = 1).

Figure 10-4. Example of Generation of Watch Timer Interrupt Request Signal (INTWT) (When Interrupt Period = 0.5 s)

It takes 0.515625 seconds for the first INTWT signal to be generated ($2^9 \times 1/32768 = 0.015625$ s longer). The INTWT signal is then generated every 0.5 seconds.



10.5 Prescaler 3

Prescaler 3 has the following function.

- Generation of watch timer count clock (source clock: main oscillation clock)

10.5.1 Control registers

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls the generation of the watch timer count clock.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Prescaler output
0	Disabled (fixed to 0)
1	Enabled

BGCS01	BGCS00	Selection of prescaler 3 clock (f_{BGCS})		
			f_X = 4 MHz	f_X = 5 MHz
0	0	f_X	250 ns	200 ns
0	1	$f_X/2$	500 ns	400 ns
1	0	$f_X/4$	1 μs	800 ns
1	1	$f_X/8$	2 μs	1.6 μs

Cautions 1. Do not change the values of the BGCS00 and BGCS01 bits during watch timer operation.

2. Set the PRSM0 register before setting the BGCE0 bit to 1.

3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare register.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

Cautions 1. Do not rewrite the PRSCM0 register during watch timer operation.

2. Set the PRSCM0 register before setting the BGCE0 bit of the PRSM0 register to 1.

3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

10.5.2 Generation of watch timer count clock

The clock input to the watch timer (f_{BRG}) can be corrected to approximate 32.768 kHz.

The relationship between the main clock (f_X), prescaler 3 clock selection bit BGCSn setting value (m), PRSCM0 register setting value (N) and output clock (f_{BRG}) is as follows.

$$f_{BRG} = \frac{f_X}{2^m \times N \times 2}$$

Example: When f_X = 4.00 MHz, m = 0 (BGCS01 bit = BGCS00 bit = 0), and N = 3DH

f_{BRG} = 32.787 kHz

Remark f_{BRG}: Watch timer count clock

N: PRSCM0 register setting value (1 to FFH)

In the case of a PRSCM0 register setting value of 00H, N = 256

m: BGCSn bit setting value (0 to 3)

n = 00, 01

CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2

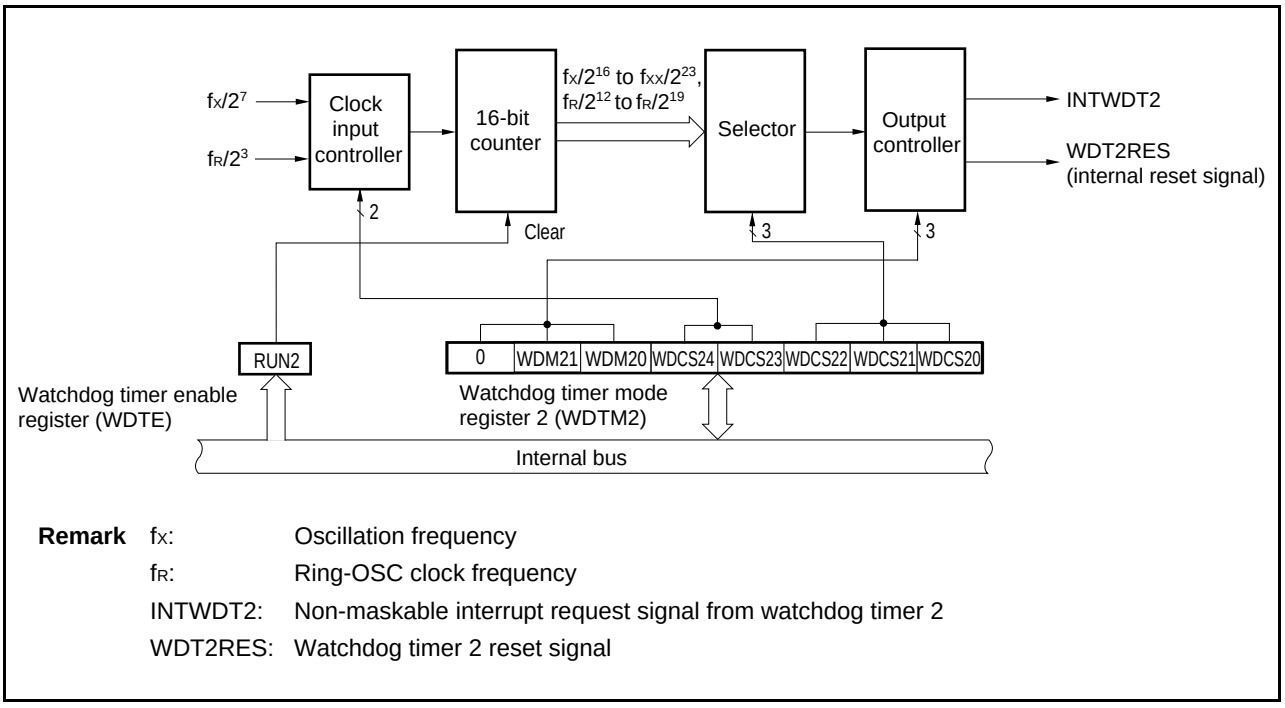
11.1 Functions

Watchdog timer 2 has the following functions.

- Default-start watchdog timer
  → 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]
- Input selectable from main clock and Ring-OSC as the source clock

Note Restoring using the RETI instruction following non-maskable interrupt servicing due to a non-maskable interrupt request signal (INTWDT2) is not possible. Therefore, following completion of interrupt servicing, perform a system reset.

Figure 11-1. Block Diagram of Watchdog Timer 2



Remark f_X: Oscillation frequency
f_R: Ring-OSC clock frequency
INTWDT2: Non-maskable interrupt request signal from watchdog timer 2
WDT2RES: Watchdog timer 2 reset signal

11.2 Configuration

Watchdog timer 2 consists of the following hardware.

Table 11-1. 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)

11.3 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 can be read or written in 8-bit units.

Reset input sets this register to 06H.

After reset: 06H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time[Note]
0	0	0	$2^{10}/f_X$
0	0	1	$2^{11}/f_X$
0	1	0	$2^{12}/f_X$
0	1	1	$2^{13}/f_X$
1	0	0	$2^{14}/f_X$
1	0	1	$2^{15}/f_X$
1	1	0	$2^{16}/f_X$
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.

(2) Watchdog timer mode register 2 (WDTM2)

This register is a special register. This register can be written only by a specific sequence.

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

This register can be read or written in 8-bit units. This register can be read any number of times, but it can be written only once following reset release.

Reset input sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	Selection of operation mode of watchdog timer 2
0	0	Stops operation
0	1	Non-maskable interrupt request mode (generation of INTWDT2 signal)
1	–	Reset mode (generation of WDT2RES signal)

Cautions 1. For details of the WDCS20 to WDCS24 bits, see Table 11-2 Watchdog Timer 2 Clock Selection.

2. 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.

3. 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 11-2. 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	$2^{17}/f_R$	1,310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$ (default)	5,242.9 ms	2,621.47 ms	1,310.7 ms
						f_X = 4 MHz		f_X = 5 MHz
0	1	0	0	0	$2^{16}/f_X$	16.4 ms		13.1 ms
0	1	0	0	1	$2^{17}f_X$	32.8 ms		26.2 ms
0	1	0	1	0	$2^{18}/f_X$	65.5 ms		52.4 ms
0	1	0	1	1	$2^{19}/f_X$	131.1 ms		104.9 ms
0	1	1	0	0	$2^{20}/f_X$	262.1 ms		209.7 ms
0	1	1	0	1	$2^{21}/f_X$	524.3 ms		419.4 ms
0	1	1	1	0	$2^{22}/f_X$	1,048.6 ms		838.9 ms
0	1	1	1	1	$2^{23}/f_X$	2,097.2 ms		1,677.7 ms
1	1	1	1	1	Stop			

(3) Watchdog timer enable register (WDTE)

The counter of watchdog timer 2 is cleared and counting restarted by writing “ACH” to the WDTE register.

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

Reset input sets this register to 9AH.

After reset: 9AH R/W Address: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE	RUN2							

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 $\overline{\text{RESET}}$ 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.

2. When a 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”).

CHAPTER 12 A/D CONVERTER

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2. The description focus on the V850ES/FJ2

12.1 Overview

This product features an A/D converter. The number of channels varies depending on the product as shown below.

Product Name	Number of Channels
V850ES/FE2	10
V850ES/FF2	12
V850ES/FG2	16
V850ES/FJ2	24

12.2 Functions

The A/D converter converts analog input signals into digital values, has a resolution of 10 bits, and can handle up to 24 analog input signal channels (ANI0 to ANIn).

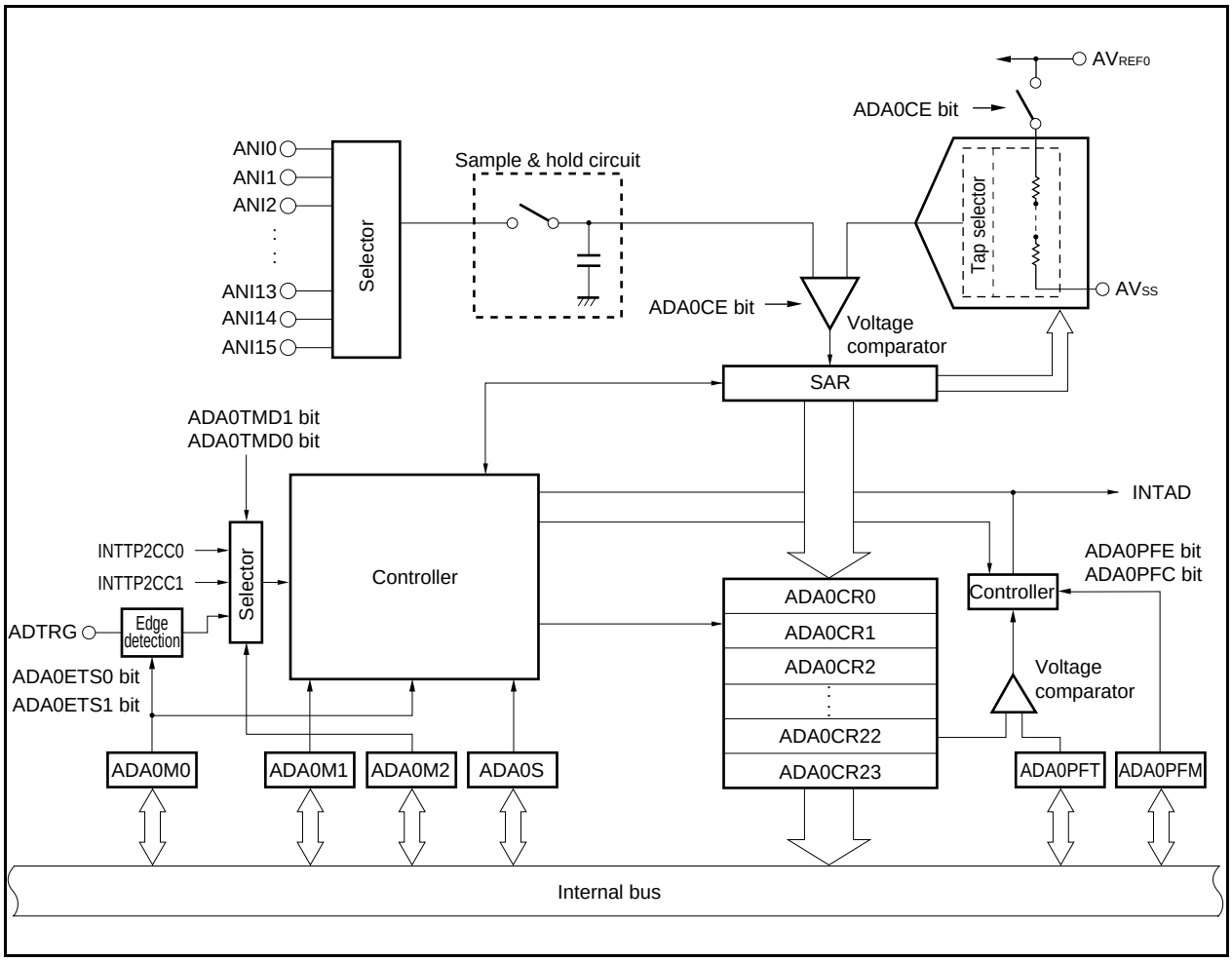
Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

The A/D converter has the following features.

- 10-bit resolution
- 10, 12, 16 or 24 channels
- Successive approximation method
- Operating voltage: AV_{REF0} = 4.0 to 5.5 V
- Analog input voltage: 0 V to AV_{REF0}
- The following functions are provided as operation modes.
  - Continuous select mode
  - Continuous scan mode
  - One-shot scan mode
- The following functions are provided as trigger modes.
  - Software trigger mode
  - External trigger mode (external, 1)
  - Timer trigger mode
- Power-fail monitor function (conversion result compare function)

The block diagram of the A/D converter is shown below.

Figure 12-1. Block Diagram of A/D Converter



12.3 Configuration

The A/D converter includes the following hardware.

Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog inputs	10 channels for V850ES/FE2 (ANI0 to ANI9 pins) 12 channels for V850ES/FF2 (ANI0 to ANI11 pins) 16 channels for V850ES/FG2 (ANI0 to ANI15 pins) 24 channels for V850ES/FJ2 (ANI0 to ANI23 pins)
Registers	Successive approximation register (SAR) A/D conversion result registers 0 to 9 for V850ES/FE2 (ADA0CR0 to ADA0CR9) 0 to 11 for V850ES/FF2 (ADA0CR0 to ADA0CR11) 0 to 15 for V850ES/FG2 (ADA0CR0 to ADA0CR15) 0 to 23 for V850ES/FJ2 (ADA0CR0 to ADA0CR23) A/D conversion result registers high where only higher 8 bits can be read 0H to 9H for V850ES/FE2 (ADA0CR0H to ADA0CR9H) 0H to 11H for V850ES/FF2 (ADA0CR0H to ADA0CR11H) 0H to 15H for V850ES/FG2 (ADA0CR0H to ADA0CR15H) 0H to 23H for V850ES/FJ2 (ADA0CR0H to ADA0CR23H)
Control registers	A/D converter mode registers 0 to 2 (ADA0M0 to ADA0M2) A/D converter channel specification register 0 (ADA0S) Power-fail compare mode register (ADA0PFM) Power-fail compare threshold value register (ADA0PFT)

(1) Successive approximation register (SAR)

The SAR register compares the voltage value of the analog input signal with the voltage tap (compare voltage) value from the D/A converter, and holds the comparison result starting from the most significant bit (MSB).

When the comparison result has been held down to the least significant bit (LSB) (i.e., when A/D conversion is complete), the contents of the SAR register are transferred to the ADA0CRn register.

Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit also holds the sampled analog input signal voltage during A/D conversion.

(3) Voltage comparator

The voltage comparator compares a voltage value that has been sampled and held with the voltage value of the D/A converter.

(4) D/A converter

This D/A converter is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(5) ANIn pins

These are analog input pins for the A/D converter channels and are used to input analog signals to be converted into digital signals. Pins other than the one selected as the analog input by the ADA0S register can be used as input port pins.

Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

Cautions 1. Make sure that the voltages input to the ANI0 to ANI23 pins do not exceed the rated values. In particular if a voltage of AV_{REF0} or higher is input to a channel, the conversion value of that channel becomes undefined, and the conversion values of the other channels may also be affected.

2. The analog input pins (ANI0 to ANI23) function alternately as input port pins (P70 to P79, P710 to P715, P120 to P127). If any of ANI0 to ANI23 is selected and A/D converted, do not execute an input instruction to ports 7 and 12 during conversion. If executed, the conversion resolution may be degraded.

(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 ANI23 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.

12.4 Control Registers

The A/D converter is controlled by the following registers.

- A/D converter mode registers 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D converter channel specification register 0 (ADA0S)
- Power-fail compare mode register (ADA0PFM)

The following registers are also used.

- A/D conversion result register n (ADA0CRn)
- A/D conversion result register nH (ADA0CRnH)
- Power-fail compare threshold value register (ADA0PFT)

Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

(1) A/D converter mode register 0 (ADA0M0)

The ADA0M0 register is an 8-bit register that specifies the operation mode and controls conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF200H

	7	6	5	4	3	2	1	0
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D conversion control
0	Stops conversion
1	Enables conversion

ADA0MD1	ADA0MD0	Specification of A/D converter operation mode
0	0	Continuous select mode
0	1	Continuous scan mode
1	1	One-shot scan mode
Other than above		Setting prohibited

ADA0ETS1	ADA0ETS0	Specification of external trigger (ADTRG pin) input valid edge
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Detection of both rising and falling edges

ADA0TMD	Trigger mode specification
0	Software trigger mode
1	External trigger mode/timer trigger mode

ADA0EF	A/D converter status display
0	A/D conversion stopped
1	A/D conversion in progress

Cautions 1. If bit 0 is written, this is ignored.

2. Changing the ADA0FR2 to ADA0FR0 bits of the ADA0M1 register during conversion (ADA0CE0 bit = 1) is prohibited.

3. When not using the A/D converter, stop the operation by setting the ADA0CE bit to 0 to reduce the current consumption.

4. The resolution of the first input terminal immediately after A/D conversion can be decreased. For details, refer to 12. 6. (7) AVREF0 pin.

5. When the subclock is operating and the main clock is stopped, accessing the ADA0M0 register is disabled. For details, see 3.4.10 (2).

(2) A/D converter mode register 1 (ADA0M1)

The ADA0M1 register is an 8-bit register that controls the conversion time specification.

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

Reset input clears this bit to 00H.

After reset: 00H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	ADA0HS0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0

Caution Be sure to clear bits 5 and 4 to 0.

Remark For A/D conversion time setting examples, see **Table 12-2**.

Table 12-2. Conversion Mode Setting Example

ADA0HS		ADA0FR3 to ADA0FR0				A/D Conversion Time	A/D Sampling Time	A/D Conversion time including sample time			A/D Stabilization Time[Note]
1	0	3	2	1	0			f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 4 MHz	
1	x	0	0	0	0	31/f_{XX}	8/f_{XX}	Setting prohibited	Setting prohibited	7.75 μs	16/f_{XX}
		0	0	0	1	62/f_{XX}	16/f_{XX}	3.10 μs	3.88 μs	15.50 μs	31/f_{XX}
		0	0	1	0	93/f_{XX}	24/f_{XX}	4.65 μs	5.81 μs	Setting prohibited	47/f_{XX}
		0	0	1	1	124/f_{XX}	32/f_{XX}	6.20 μs	7.75 μs	Setting prohibited	50/f_{XX}
		0	1	0	0	155/f_{XX}	40/f_{XX}	7.75 μs	9.69 μs	Setting prohibited	50/f_{XX}
		0	1	0	1	186/f_{XX}	48/f_{XX}	9.30 μs	11.63 μs	Setting prohibited	50/f_{XX}
		0	1	1	0	217/f_{XX}	56/f_{XX}	10.85 μs	13.56 μs	Setting prohibited	50/f_{XX}
		0	1	1	1	248/f_{XX}	64/f_{XX}	12.40 μs	15.50 μs	Setting prohibited	50/f_{XX}
		1	0	0	0	279/f_{XX}	72/f_{XX}	13.95 μs	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	0	1	310/f_{XX}	80/f_{XX}	15.50 μs	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	1	0	341/f_{XX}	88/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	1	1	372/f_{XX}	96/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	0	0	403/f_{XX}	104/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	0	1	434/f_{XX}	112/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	1	0	465/f_{XX}	120/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	1	1	496/f_{XX}	128/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}

Note When the ADA0CE bit of the ADA0M0 register is changed from 0 to 1 to secure the A/D converter stabilization time, the first A/D conversion starts after one of the above clock values is input.

(3) A/D converter mode register (ADA0M2)

The ADA0M2 register specifies the hardware trigger mode.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	Specification of hardware trigger mode
0	0	External trigger mode (when ADTRG pin valid edge detected)
0	1	Timer trigger mode 0 (when INTTP2CC0 interrupt request generated)
1	0	Timer trigger mode 1 (when INTTP2CC1 interrupt request generated)
1	1	Setting prohibited

Caution Be sure to clear bits 7 to 2 to 0.

(4) A/D converter channel specification register 0 (ADA0S)

The ADA0S register specifies the pin that inputs the analog voltage to be converted into a digital signal.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF202H

	7	6	5	4	3	2	1	0
ADA0S	0	0	0	ADA0S4	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S4	ADA0S3	ADA0S2	ADA0S1	ADA0S0	Select mode	Scan mode
0	0	0	0	0	ANI0	ANI0
0	0	0	0	1	ANI1	ANI0, ANI1
0	0	0	1	0	ANI2	ANI0 to ANI2
0	0	0	1	1	ANI3	ANI0 to ANI3
0	0	1	0	0	ANI4	ANI0 to ANI4
0	0	1	0	1	ANI5	ANI0 to ANI5
0	0	1	1	0	ANI6	ANI0 to ANI6
0	0	1	1	1	ANI7	ANI0 to ANI7
0	1	0	0	0	ANI8	ANI0 to ANI8
0	1	0	0	1	ANI9	ANI0 to ANI9
0	1	0	1	0	ANI10	ANI0 to ANI10
0	1	0	1	1	ANI11	ANI0 to ANI11
0	1	1	0	0	ANI12	ANI0 to ANI12
0	1	1	0	1	ANI13	ANI0 to ANI13
0	1	1	1	0	ANI14	ANI0 to ANI14
0	1	1	1	1	ANI15	ANI0 to ANI15
1	0	0	0	0	ANI16	ANI0 to ANI16
1	0	0	0	1	ANI17	ANI0 to ANI17
1	0	0	1	0	ANI18	ANI0 to ANI18
1	0	0	1	1	ANI19	ANI0 to ANI19
1	0	1	0	0	ANI20	ANI0 to ANI20
1	0	1	0	1	ANI21	ANI0 to ANI21
1	0	1	1	0	ANI22	ANI0 to ANI22
1	0	1	1	1	ANI23	ANI0 to ANI23
Other than above					Setting prohibited	

Remark ANI0 to ANI9 (V850ES/FE2)
ANI0 to ANI12 (V850ES/FF2)
ANI0 to ANI15 (V850ES/FG2)
ANI0 to ANI23 (V850ES/FJ2)

(5) A/D conversion result registers n, nH (ADA0CRn, ADA0CRnH)

The ADA0CRn register is a 16-bit register that stores the A/D conversion result. ADA0CRn consist of n registers. The ADA0CRn and ADA0CRnH registers are read-only, in 16-bit or 8-bit units. However, specify the ADA0CRn register for 16-bit access and the ADA0CRnH register for 8-bit access. The 10 bits of the conversion result are read from the higher 10 bits of the ADA0CRn register, and 0 is read from the lower 6 bits. The higher 8 bits of the conversion result are read from the ADA0CRnH register.

After reset: 00H R Address: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H,
ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H
ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH
ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH
ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H
ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H
ADA0CR12 FFFFF228H, ADA0CR13 FFFFF22AH
ADA0CR14 FFFFF22CH, ADA0CR15 FFFFF22EH
ADA0CR16 FFFFF230H, ADA0CR17 FFFFF232H
ADA0CR18 FFFFF234H, ADA0CR19 FFFFF236H
ADA0CR20 FFFFF238H, ADA0CR21 FFFFF23AH
ADA0CR22 FFFFF23CH, ADA0CR23 FFFFF23EH

	15	14	13	12	11	10	9	8
ADA0CRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

7	6	5	4	3	2	1	0
AD1	AD0	0	0	0	0	0	0

After reset: 00H R Address: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H
ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H
ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH
ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH
ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H
ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H
ADA0CR12H FFFFF229H, ADA0CR13H FFFFF22BH
ADA0CR14H FFFFF22DH, ADA0CR15H FFFFF22FH
ADA0CR16H FFFFF231H, ADA0CR17H FFFFF233H
ADA0CR18H FFFFF235H, ADA0CR19H FFFFF237H
ADA0CR20H FFFFF239H, ADA0CR21H FFFFF23BH
ADA0CR22H FFFFF23DH, ADA0CR23H FFFFF23FH

	7	6	5	4	3	2	1	0
ADA0CRnH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

Cautions 1. A write operation to the ADA0M0 and ADA0S registers may cause the contents of the ADA0CRn register to become undefined. After the conversion, read the conversion result before writing to the ADA0M0 and ADA0S registers. Correct conversion results may not be read if a sequence other than the above is used.

2. When the subclock is operating and the main clock is stopped, accessing the ADA0CRn and ADA0CRnH registers is disabled. For details, see 3.4.10 (2).

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI11) and the A/D conversion result (of A/D conversion result register n (ADA0CRn)) is as follows.

$$ADA0CR = INT\left(\frac{V_{IN}}{AV_{REF0}} \times 1{,}024 + 0.5\right)$$

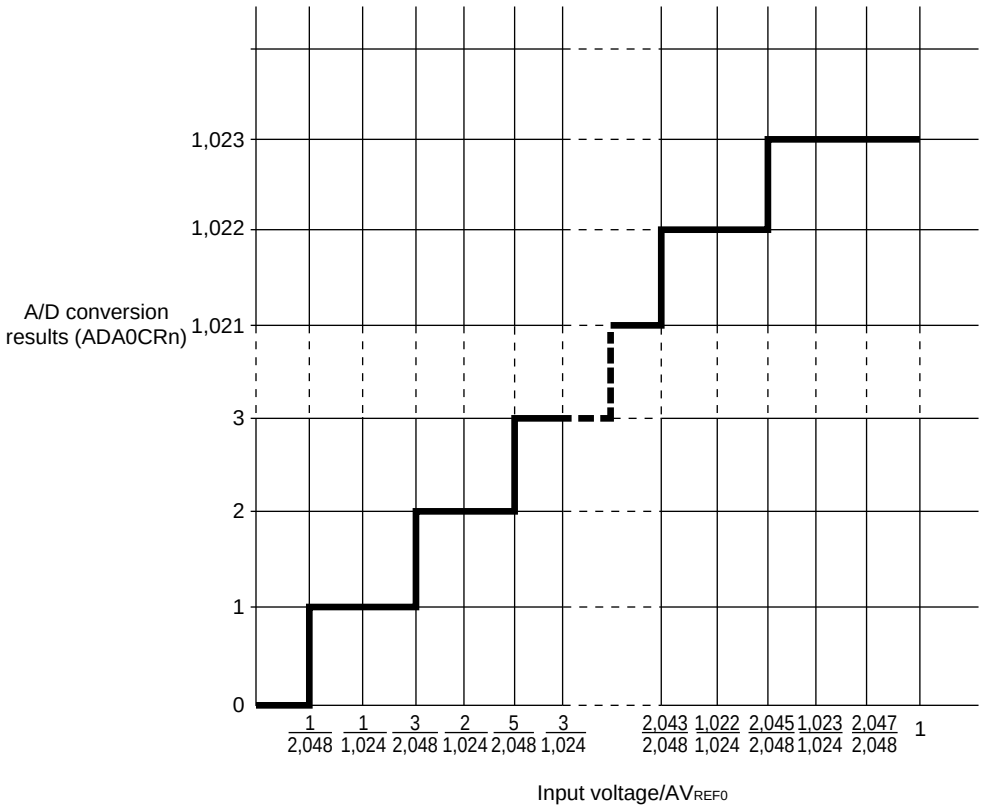
Or,

$$(ADA0CR - 0.5) \times \frac{AV_{REF0}}{1{,}024} \leq V_{IN} < (ADA0CR + 0.5) \times \frac{AV_{REF0}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{REF0}: AV_{REF0} pin voltage
ADA0CR: Value of A/D conversion result register n (ADA0CRn)

Figure 12-2 shows the relationship between the analog input voltage and the A/D conversion results.

Figure 12-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0 to 11

(6) Power-fail compare mode register (ADA0PFM)

The ADA0PFM register is an 8-bit register that sets the power-fail compare mode.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF204H

	7	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	Selection of power-fail compare enable/disable
0	Power-fail compare disabled
1	Power-fail compare enabled

ADA0PFC	Selection of power-fail compare mode
0	Generates an interrupt request signal (INTAD) when ADA0CRn ≥ ADA0PFT
1	Generates an interrupt request signal (INTAD) when ADA0CRn < ADA0PFT

Cautions 1. In the select mode, the 8-bit data set to the ADA0PFT register is compared with the value of the ADA0CRnH register specified by the ADA0S register. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register and the INTAD signal is generated. If it does not match, however, the interrupt signal is not generated.

2. In the scan mode, the 8-bit data set to the ADA0PFT register is compared with the contents of the ADA0CR0H register. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register and the INTAD signal is generated. If it does not match, however, the INTAD signal is not generated. Regardless of the comparison result, the scan operation is continued and the conversion result is stored in the ADA0CRn register until the scan operation is completed. However, the INTAD signal is not generated after the scan operation has been completed.

(7) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets a threshold value that is compared with the value of A/D conversion result register nH (ADA0CRnH). The 8-bit data set to the ADA0PFT register is compared with the higher 8 bits of the A/D conversion result register (ADA0CRnH).

The ADA0PFT register sets the compare value in the power-fail compare mode.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF205H

	7	6	5	4	3	2	1	0
ADA0PFT								

12.5 Operation

12.5.1 Basic operation

<1> Set the operation mode, trigger mode, and conversion time for executing A/D conversion by using the ADA0M0, ADA0M1, ADA0M2, and ADA0S registers. When the ADA0CE bit of the ADA0M0 register is set, conversion is started in the software trigger mode and the A/D converter waits for a trigger in the external or timer trigger mode.

<2> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When the sample & hold circuit samples the input channel for a specific time, it enters the hold status, and holds the input analog voltage until A/D conversion is complete.

<4> Set bit 9 of the successive approximation register (SAR). The voltage of the D/A converter is (1/2) AVREF0.

<5> The voltage difference between the voltage of the D/A converter and the analog input voltage is compared by the voltage comparator. If the analog input voltage is higher than (1/2) AV_{REF0}, the MSB of the SAR register remains set. If it is lower than (1/2) AV_{REF0}, the MSB is reset.

<6> Next, bit 8 of the SAR register is automatically set and the next comparison is started. Depending on the value of bit 9, to which a result has been already set, <6>, the voltage of the D/A converter is selected as follows.

- Bit 9 = 1: (3/4) AV_{REF0}
- Bit 9 = 0: (1/4) AV_{REF0}

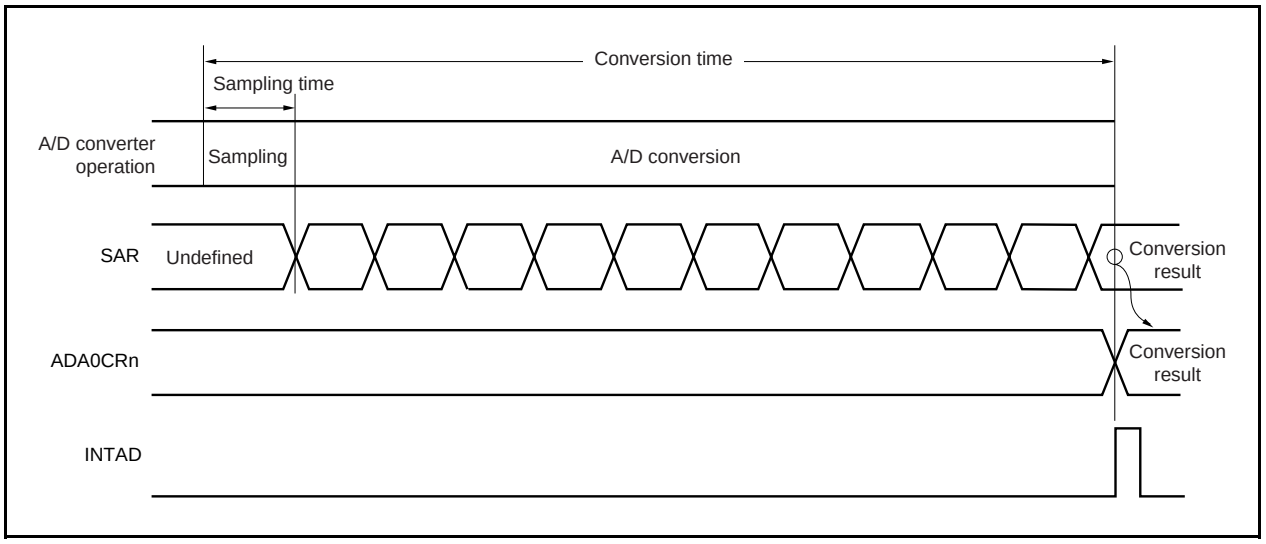
This voltage of the D/A converter and the analog input voltage are compared and, depending on the result, bit 8 is manipulated as follows.

- Analog input voltage ≧ Voltage of the D/A converter
- Analog input voltage ≦ Voltage of the D/A converter

<7> This comparison is continued to bit 0 of the SAR register.

<8> When comparison of the 10 bits is complete, the valid digital result is stored in the SAR register, 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 12-3. A/D Converter Basic Operation



12.5.2 Trigger mode

The timing of starting the conversion operation is specified by setting a trigger mode. The trigger mode includes a software trigger mode and hardware trigger modes. The hardware trigger modes include timer trigger modes 0 and 1, and external trigger mode. The ADA0TMD bit of the ADA0M0 register is used to set the trigger mode. The hardware trigger modes are set by the ADA0TMD1 and ADA0TMD0 bits of the ADA0M2 register.

(1) Software trigger mode

When the ADA0CE bit of the ADA0M0 register is set to 1, the signal of the analog input pin (ANI0 to ANI23 pin) specified by the ADA0S register is converted. When conversion is complete, the result is stored in the ADA0CRn register. At the same time, the A/D conversion end interrupt request signal (INTAD) is generated.

If the operation mode specified by the ADA0MD1 and ADA0MD0 bits of the ADA0M0 register is the continuous select/scan mode, the next conversion is started, unless the ADA0CE bit is cleared to 0 after completion of the first conversion.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress).

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted and started again from the beginning.

(2) External trigger mode

In this mode, converting the signal of the analog input pin (ANI0 to ANI23) specified by the ADA0S register is started when an external trigger is input (to the ADTRG pin). Which edge of the external trigger is to be detected (i.e., the rising edge, falling edge, or both rising and falling edges) can be specified by using the ADA0ETS1 and ATA0ETS0 bits of the ADA0M0 register. When the ADA0CE bit of the ADA0M0 register set to 1, the A/D converter waits for the trigger, and starts conversion after the external trigger has been input.

When conversion is completed, the result of conversion is stored in the ADA0CRn register. At the same time, the A/D conversion end interrupt request signal (INTAD) is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped). If the valid trigger is input during the conversion operation, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during the conversion operation, the conversion is not aborted, and the A/D converter waits for the trigger again.

(3) Timer trigger mode

In this mode, converting the signal of the analog input pin (ANI0 to ANI23) specified by the ADA0S register is started by the compare match interrupt request signal (INTTP2CC0 or INTTP2CC1) of the capture/compare register connected to the timer. The timer compare match interrupt request signal (INTTP2CC0 or INTTP2CC1) is selected by the ADA0TMD1 and ADA0TMD0 bits of the ADA0M2 register, and conversion is started at the rising edge of the specified compare match interrupt request signal. When the ADA0CE bit of the ADA0M0 register is set to 1, the A/D converter waits for a trigger, and starts conversion when the compare match interrupt signal of the timer is input.

When conversion is completed, the result of the conversion is stored in the ADA0CRn register. At the same time, the A/D conversion end interrupt request signal (INTAD) is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped). If the valid trigger is input during the conversion operation, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is stopped and the A/D converter waits for the trigger again.

12.5.3 Operation mode

Three operation modes are available as the modes in which to set the ANI0 to ANI23 pins: continuous select mode continuous scan mode and one-shot scan mode.

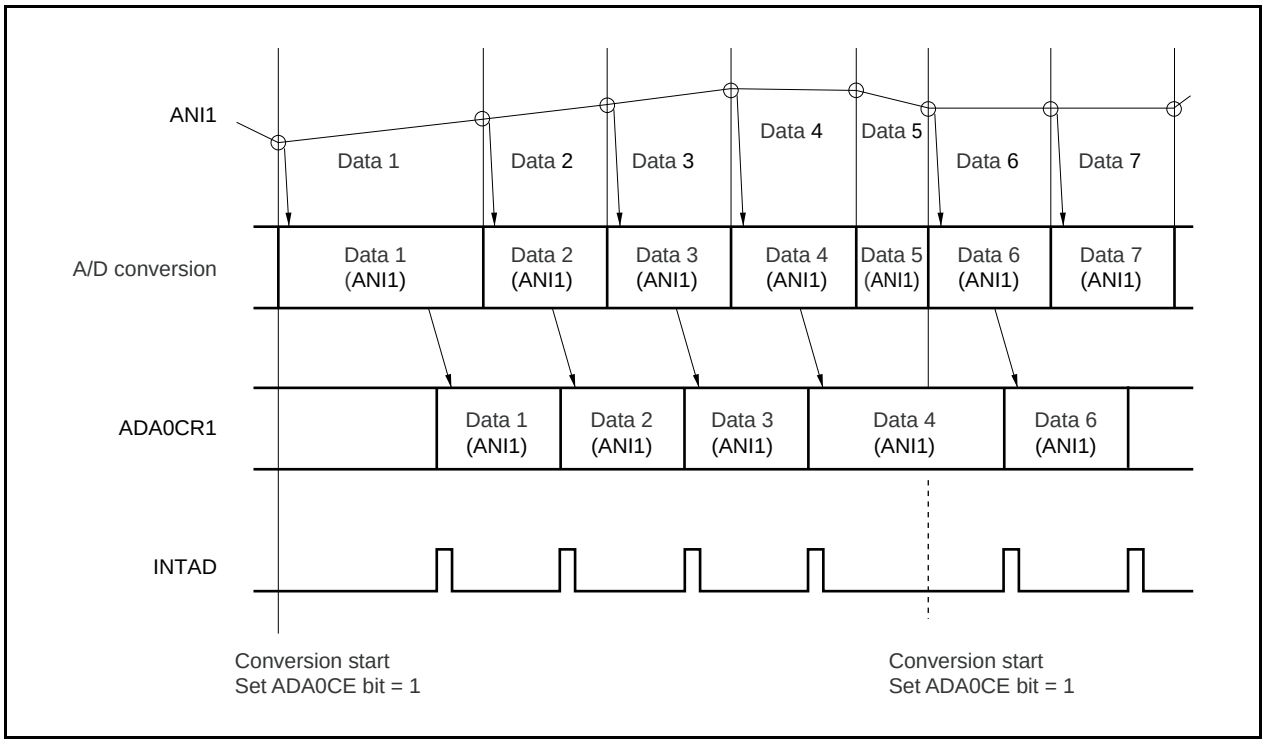
The operation mode is selected by the ADA0MD1 and ADA0MD0 bits of the ADA0M0 register.

(1) Continuous select mode

In this mode, the voltage of one analog input pin selected by the ADA0S register is continuously converted into a digital value.

The conversion result is stored in the ADA0CRn register corresponding to the analog input pin. In this mode, an analog input pin corresponds to an ADA0CRn register on a one-to-one basis. Each time A/D conversion is completed, the A/D conversion end interrupt request signal (INTAD) is generated. After completion of conversion, the next conversion is started, unless the ADA0CE bit of the ADA0M0 register is cleared to 0 (n = 0 to 23).

Figure 12-4. Timing Example of Continuous Select Mode Operation (ADA0S = 01H)



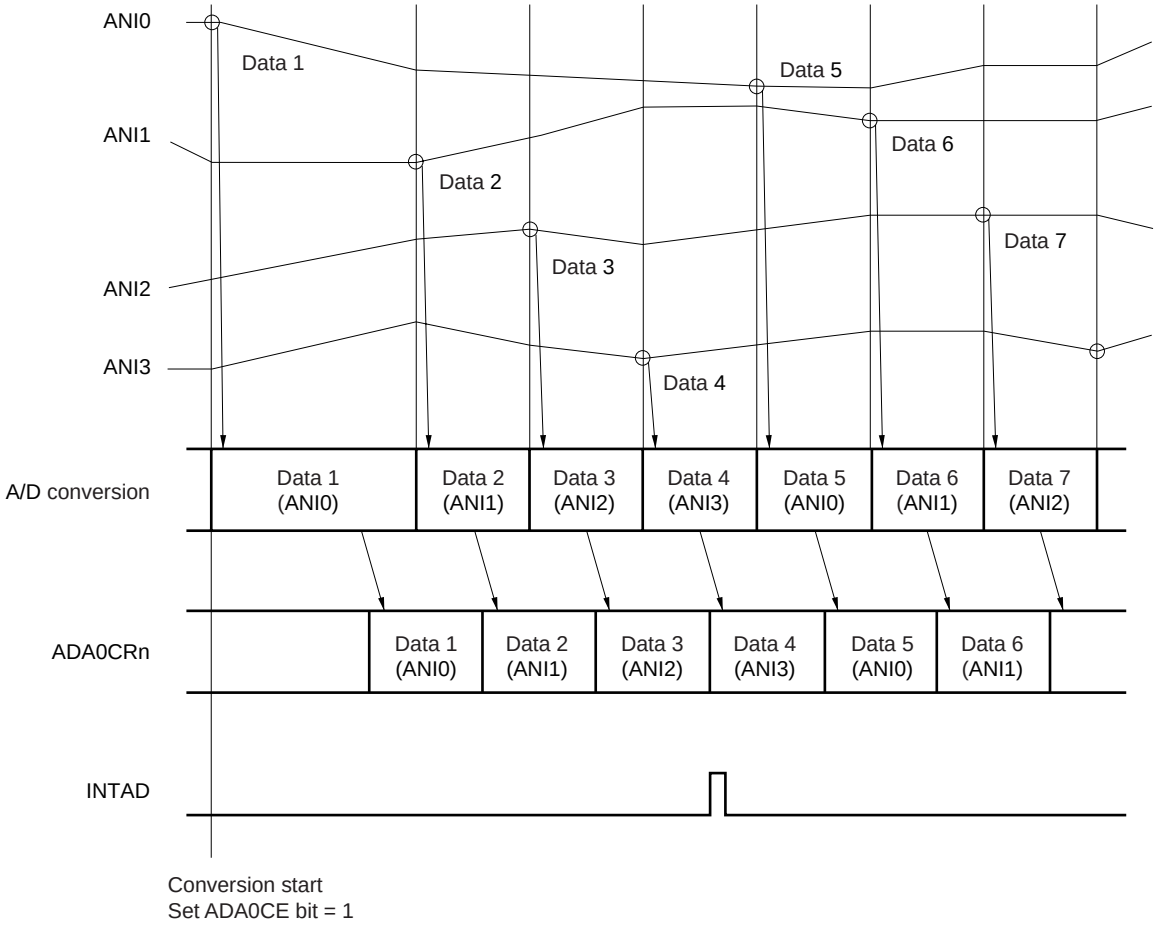
(2) Continuous scan mode

In this mode, analog input pins are sequentially selected, from the ANI0 pin to the pin specified by the ADA0S register, and their values are converted into digital values.

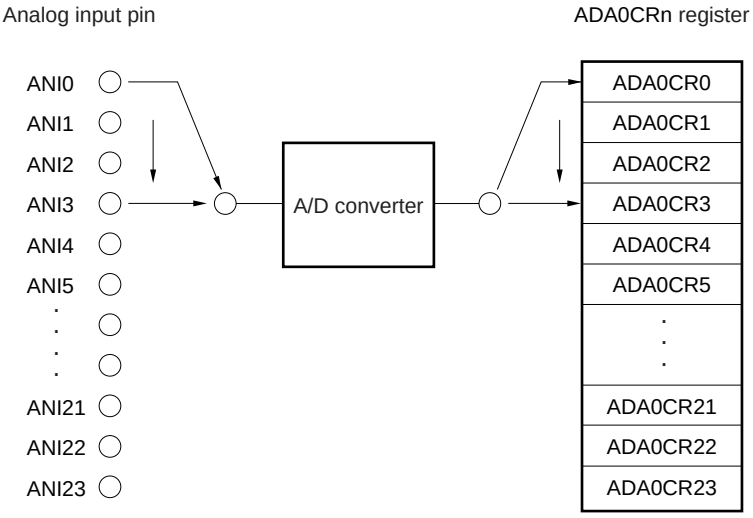
The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the A/D conversion end interrupt request signal (INTAD) is generated, and A/D conversion is started again from the ANI0 pin, unless the ADA0CE bit of the ADA0M0 register is cleared to 0 (n = 0 to 23).

Figure 12-5. Timing Example of Continuous Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram

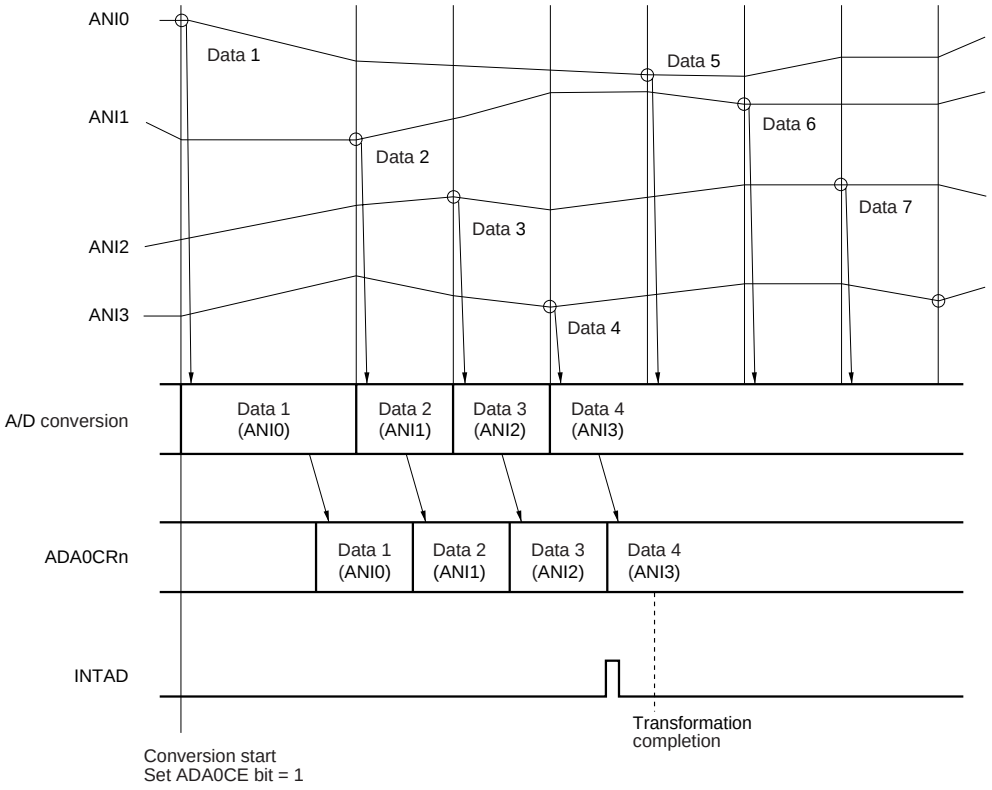


(3) One-shot scan mode

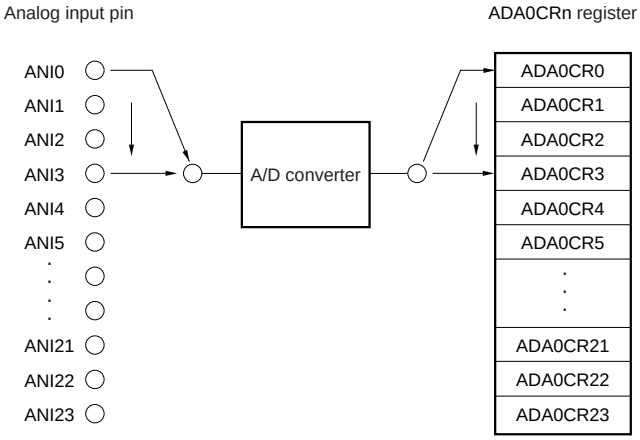
In this mode, analog input pins are sequentially selected, from the ANI0 pin to the pin specified by the ADA0S register, and their values are converted into digital values. The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the A/D conversion end interrupt request signal (INTAD) is generated, and A/D conversion is stopped.

Figure 12-6. Timing Example of One-shot Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



12.5.4 Power-fail compare mode

The A/D conversion end interrupt request signal (INTAD) can be controlled as follows by the ADA0PFM and ADA0PFT registers.

- When the ADA0PFE bit = 0, the INTAD signal is generated each time conversion is completed (normal use of the A/D converter).
- When the ADA0PFE bit = 1 and when the ADA0PFC bit = 0, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if ADA0CR0H ≥ ADA0PFT.
- When the ADA0PFE bit = 1 and when the ADA0PFC bit = 1, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if ADA0CR0H < ADA0PFT.

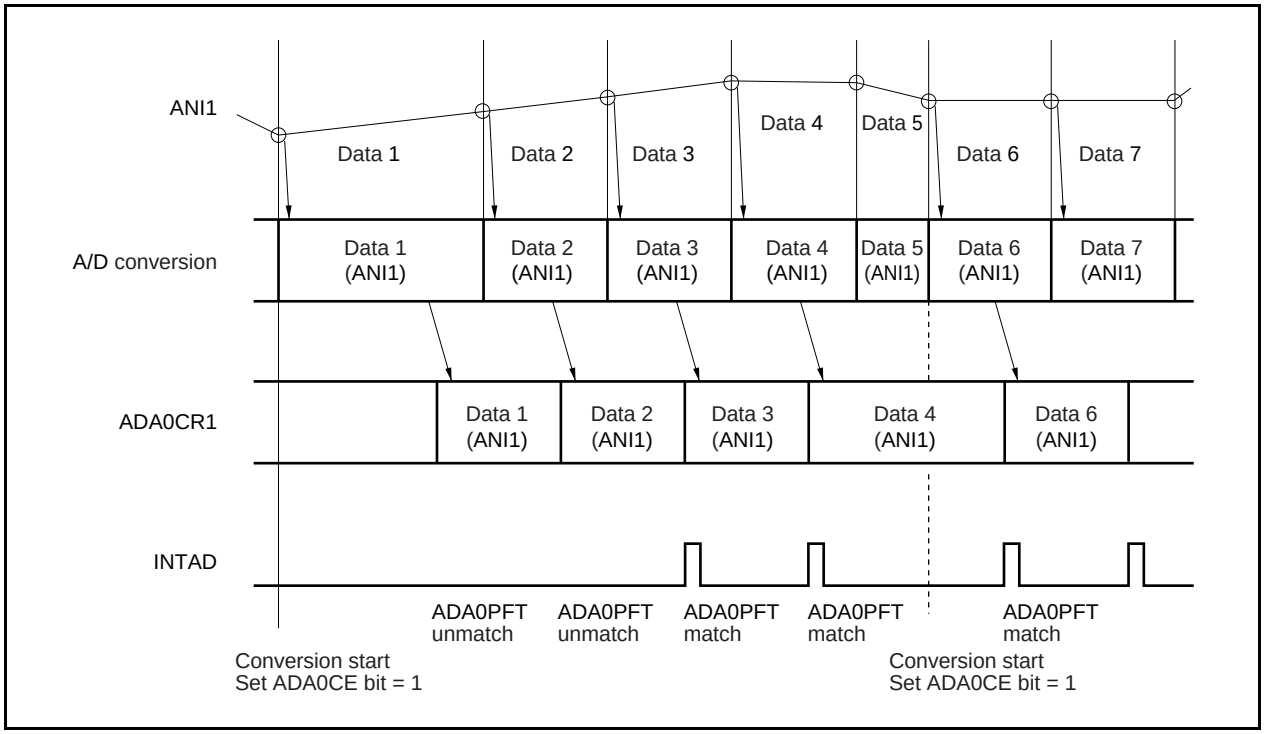
Remark n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

In the power-fail compare mode, two modes are available as modes in which to set the ANI0 to ANI23 pins: continuous select mode and continuous scan mode.

(1) Continuous select mode

In this mode, the result of converting the voltage of the analog input pin specified by the ADA0S register is compared with the set value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CRn register, and the INTAD signal is not generated. After completion of the first conversion, the next conversion is started, unless the ADA0CE bit of the ADA0M0 register is cleared to 0 (n = 0 to 23).

Figure 12-7. Timing Example of Continuous Select Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 01H)



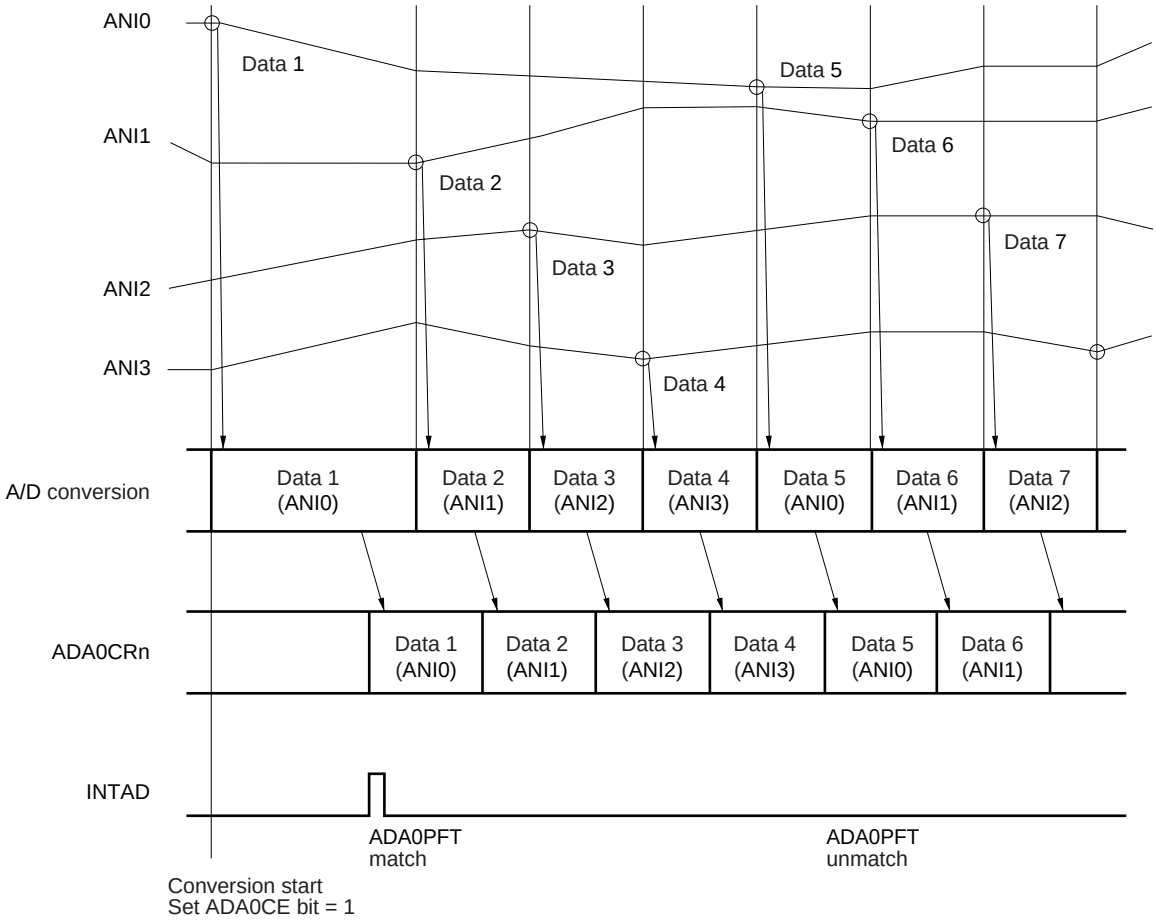
(2) Continuous scan mode

In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANI0 pin to the pin specified by the ADA0S register are stored, and the set value of the ADA0CR0H register of channel 0 is compared with the value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit of the ADA0PFM register, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is not generated.

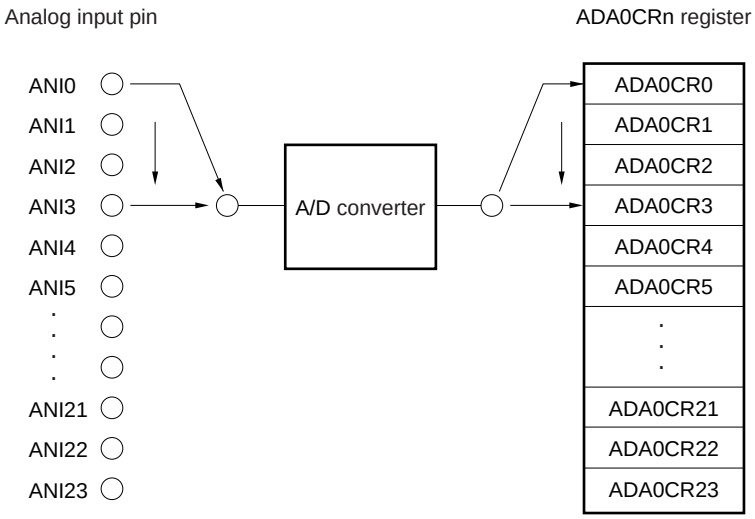
After the result of the first conversion has been stored in the ADA0CR0 register, the results of sequentially converting the voltages on the analog input pins up to the pin specified by the ADA0S register are continuously stored. After completion of conversion, the next conversion is started from the ANI0 pin again, unless the ADA0CE bit of the ADA0M0 register is cleared to 0.

Figure 12-8. Timing Example of Continuous Scan Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



★ **(3) One-shot scan mode**

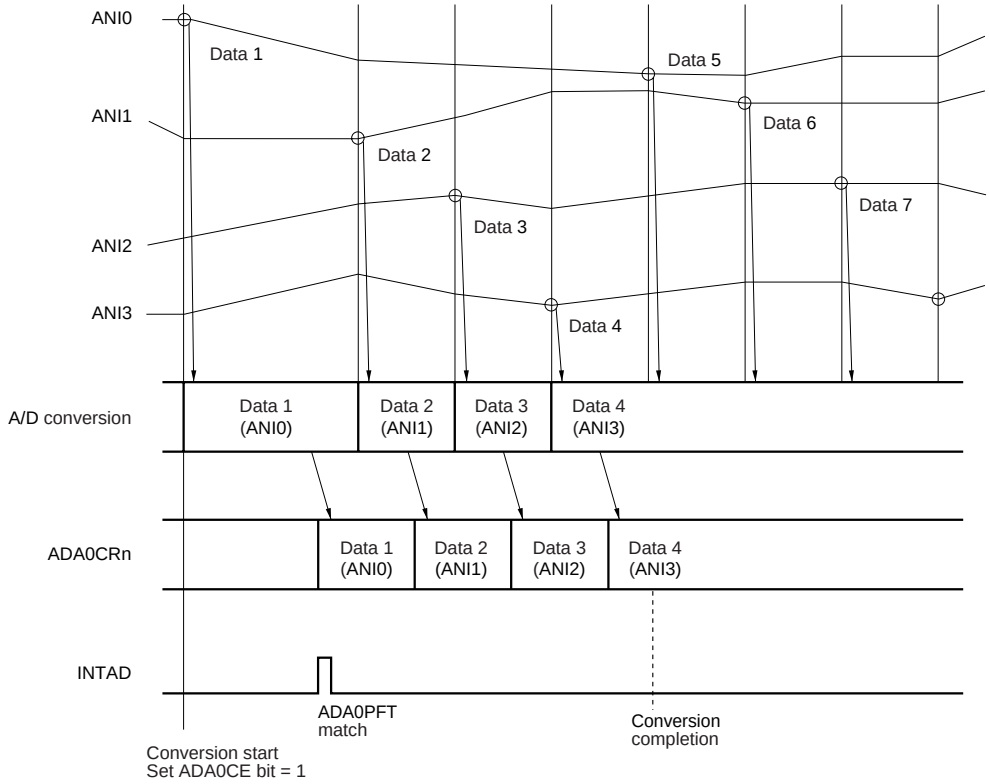
In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANI0 pin to the pin specified by the ADA0S register are stored, and the set value of the ADA0CR0H register of channel 0 is compared with the value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit of the ADA0PFM register, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is not generated.

After the result of the first conversion has been stored in the ADA0CR0 register, the results of sequentially converting the voltages on the analog input pins up to the pin specified by the ADA0S register are continuously stored.

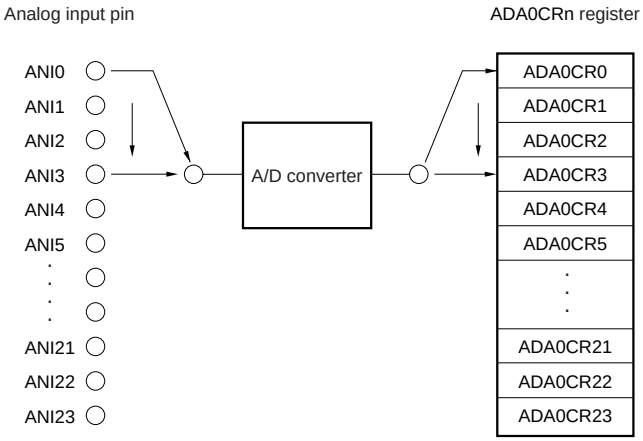
After completion of conversion, A/D conversion is stopped. The 1st conversion result after A/D conversion has to be ignored, because it is not good.

Figure 12-9. Timing Example of One-shot Scan Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



12.6 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the ADA0CE bit of the ADA0M0 register to 0.

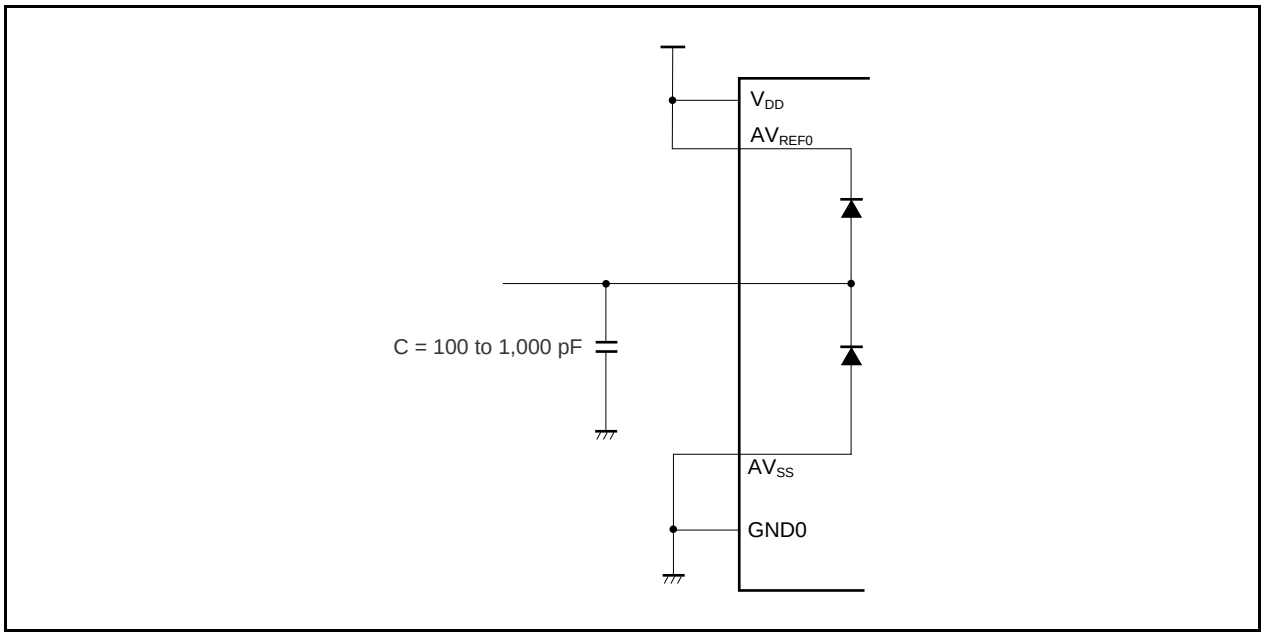
(2) Input range of ANI0 to ANI23 pins

Input the voltage within the specified range to the ANI0 to ANI23 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 ANI23 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 12-10 is recommended.

Figure 12-10. Processing of Analog Input Pin



(4) Alternate I/O

The analog input pins (ANI0 to ANI23) function alternately as port pins. When selecting one of the ANI0 to ANI23 pins to execute A/D conversion, do not execute an instruction to read an input port or write to an output port during conversion as the conversion resolution may drop.

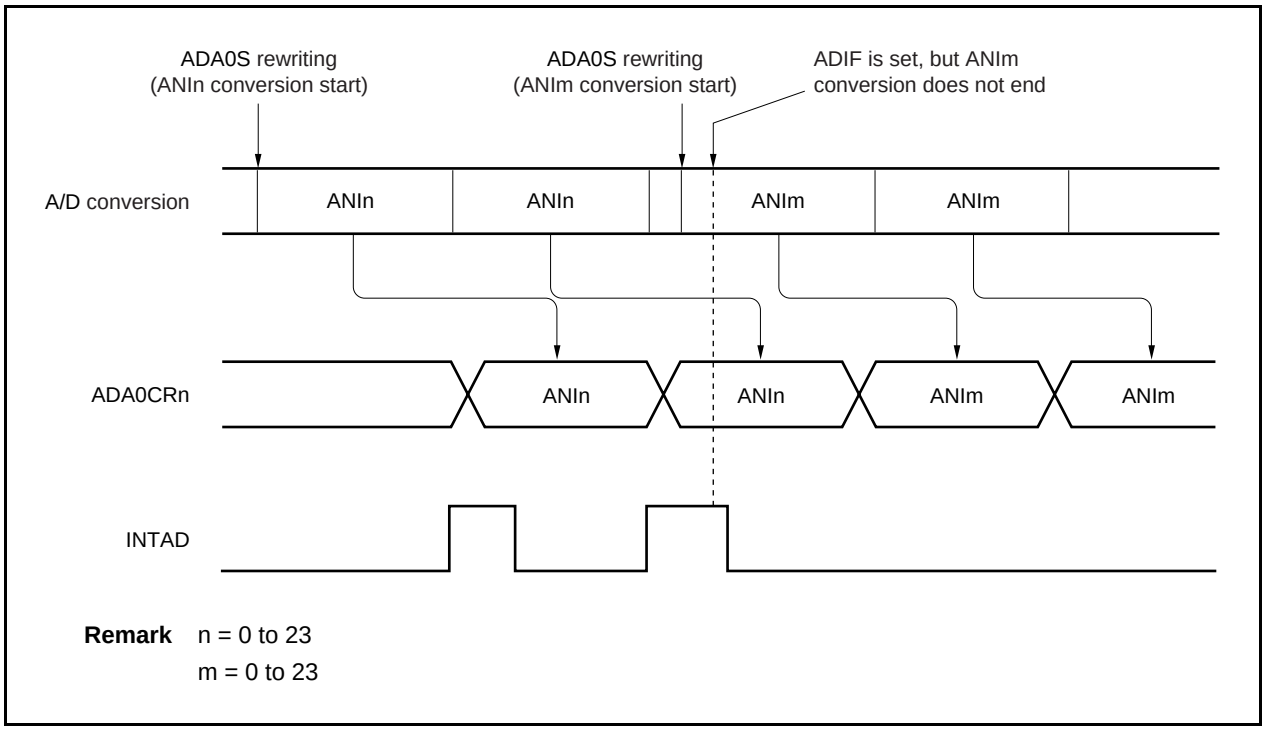
Also the conversion resolution may drop at the pins set as output port pins during A/D conversion if the output current fluctuates due to the effect of the external circuit connected to the port pins.

If a digital pulse is applied to a pin adjacent to the pin whose input signal is being converted, the A/D conversion value may not be as expected due to the influence of coupling noise. Therefore, do not apply a pulse to a pin adjacent to the pin undergoing A/D conversion.

(5) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the contents of the ADA0S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the conversion end interrupt request flag may be set immediately before the ADA0S register is rewritten. If the ADIF flag is read immediately after the ADA0S register is rewritten, the ADIF flag may be set even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear the ADIF flag before resuming conversion.

Figure 12-11. Generation Timing of A/D Conversion End Interrupt Request

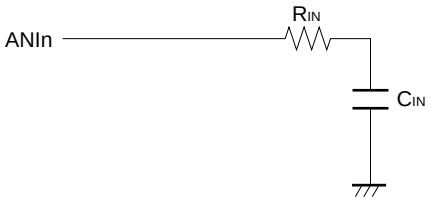


Remark n = 0 to 23
m = 0 to 23

(6) Internal equivalent circuit

The following shows the equivalent circuit of the analog input block.

Figure 13-14. Internal Equivalent Circuit of ANIn Pin



Product Name	R_{IN}	C_{IN}
V850ES/FE2, FF2, FG2	5.9 kΩ	7.0 pF
V850ES/FJ2	6.0 kΩ	8.3 pF

Remarks 1. The above values are reference values.

2. n = 0 to 9 (V850ES/FE2)
n = 0 to 11 (V850ES/FF2)
n = 0 to 15 (V850ES/FG2)
n = 0 to 23 (V850ES/FJ2)

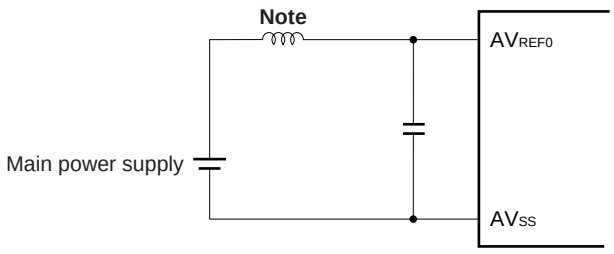
(7) AV_{REF0} pin

(a) The AV_{REF0} pin is used as the power supply pin of the A/D converter and also supplies power to the alternate-function ports. In an application where a backup power supply is used, be sure to supply the same voltage as V_{DD} to the AV_{REF0} pin as shown in Figure 12-10.

(b) The AV_{REF0} pin is also used as the reference voltage pin of the A/D converter. If the source supplying power to the AV_{REF0} pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion operation enable bit ADA0CE has been set to 1). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AV_{REF0} and AV_{SS} pins to suppress the reference voltage fluctuation as shown in Figure 12-12.

(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 12-12. AV_{REF0} Pin Processing Example



Note Parasitic inductance

(8) Reading ADA0CRn register

When the ADA0M0 to ADA0M2 or ADA0S register is written, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the ADA0M0 to ADA0M2 and ADA0S registers. The correct conversion result may not be read at a timing different from the above.

★ **(9) Standby mode**

Because the A/D converter stops operating in the STOP mode, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the STOP mode is released, but the A/D conversion results after the STOP mode is released are invalid. When using the A/D converter after the STOP mode is released, before setting the STOP mode or releasing the STOP mode, clear the ADA0M0.ADA0CE bit to 0 then set the ADA0CE bit to 1 after releasing the STOP mode.

In the IDLE1, IDLE2, or subclock operation mode, operation continues. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE bit to 0. In the IDLE1 and IDLE2 modes, since the analog input voltage value cannot be retained, the A/D conversion results after the IDLE1 and IDLE2 modes are released are invalid. The results of conversions before the IDLE1 and IDLE2 modes were set are valid.

(10) 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) A/D conversion result hysteresis characteristics**

The successive comparison type A/D converter holds the analog input voltage in the internal sample & holds capacitor and then performs A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Thus, even if the conversion is performed at the same potential, the result may vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Thus, even if the conversion is performed at the same potential, the result may vary.

12.7 How to Read A/D Converter Characteristics Table

This section describes the terms related to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be recognized, i.e., the ratio of an analog input voltage to 1 bit of digital output is called 1 LSB (least significant bit). The ratio of 1 LSB to the full scale is expressed as %FSR (full-scale range). %FSR is the ratio of a range of convertible analog input voltages expressed as a percentage, and can be expressed as follows, independently of the resolution.

1%FSR = (Maximum value of convertible analog input voltage – Minimum value of convertible analog input voltage)/100
= (AV_{REF0} – 0)/100
= AV_{REF0}/100

When the resolution is 10 bits, 1 LSB is as follows:

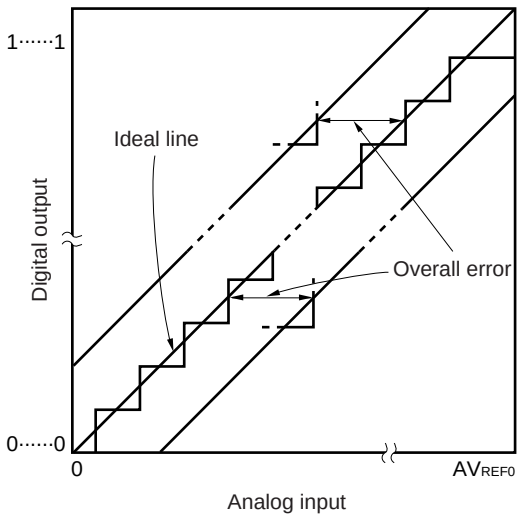
1 LSB = $1/2^{10}$ = 1/1,024
= 0.098%FSR

The accuracy is determined by the overall error, independently of the resolution.

(2) Overall error

This is the maximum value of the difference between an actually measured value and a theoretical value.
It is a total of zero-scale error, full-scale error, linearity error, and a combination of these errors.
The overall error in the characteristics table does not include the quantization error.

Figure 12-13. Overall Error

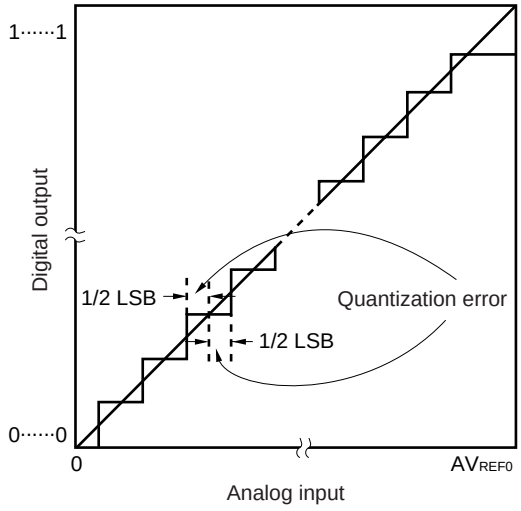


(3) Quantization error

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.

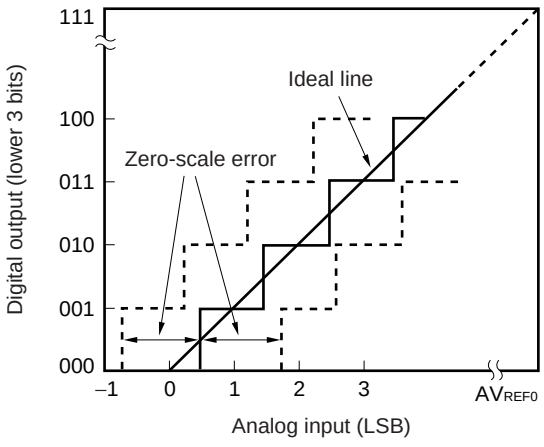
Figure 12-14. Quantization Error



(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).

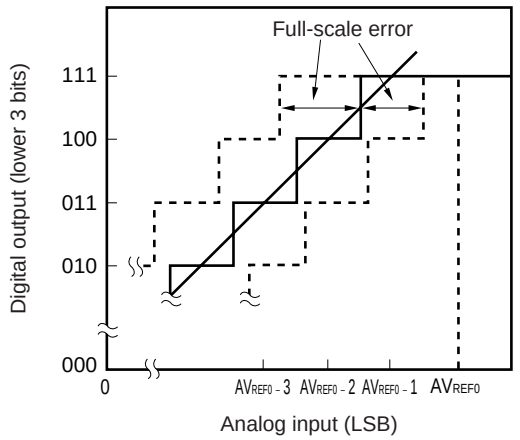
Figure 12-15. Zero-Scale Error



(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).

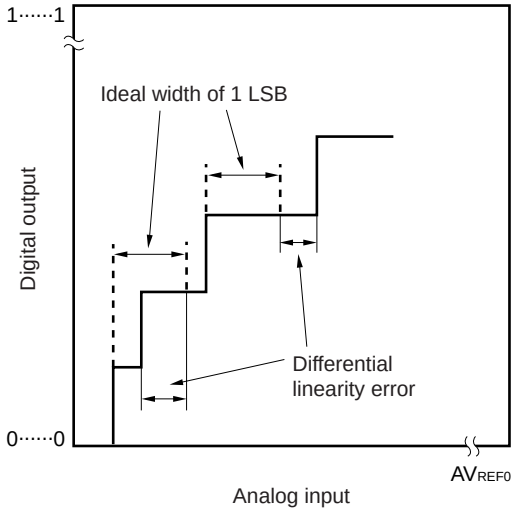
Figure 12-16. Full-Scale Error



(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.

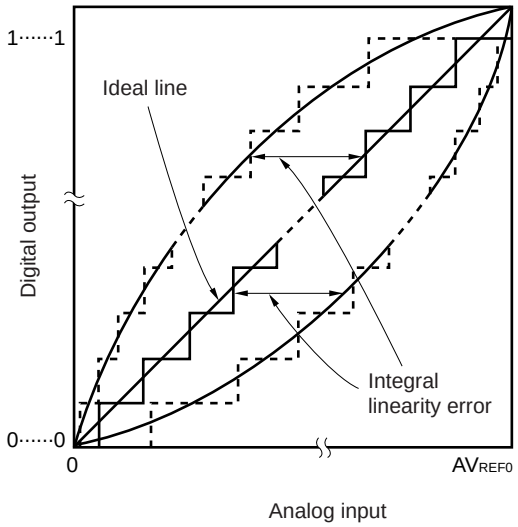
Figure 12-17. Differential Linearity Error



(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 12-18. Integral Linearity Error



(8) Conversion time

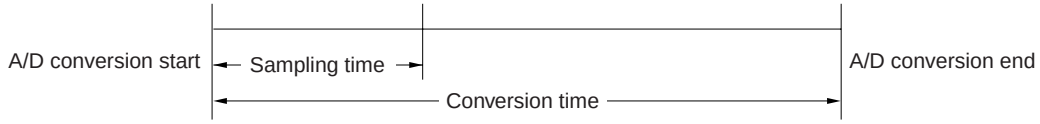
This is the time required to obtain a digital output after an analog input voltage has been assigned.

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 12-19. Sampling Time



CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

The V850ES/Fx2 includes asynchronous serial interface A (UARTA).

The number of channels differs depending on the product. Table 13-1 shows the number of channels of each product.

Table 13-1. Number of Channels of Asynchronous Serial Interface A

Product Name (Part Number)		Number of Channels
V850ES/FE2		2 (UARTA0 to UARTA1)
V850ES/FF2		
V850ES/FG2		3 (UARTA0 to UARTA2)
V850ES/FJ2	μPD70F3237	3 (UARTA0 to UARTA2)
	μPD70F3238 μPD70F3239	4 (UARTA0 to UARTA3)

13.1 Features

- Transfer rate: 300 bps to 312.5 kbps (using internal system clock of 20 MHz and dedicated baud rate generator)
- 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: 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
- 13 to 20 bits selectable for SBF (Sync Break Field) transmission in LIN (Local Interconnect Network) communication format
- 11 or more bits recognizable for SBF reception in LIN communication format
- SBF reception flag

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

13.2 Configuration

UARTA consists of the following hard ware

Table 13-2. configuration of UARTA0 to UARTA2

Item	Configuration
Register	UARTAn reception shift register UARTAn reception data register (UAnRX) UARTAn transmit shift register UARTAn transmit data register (UAnTX)
Reception data input	μPD70F3237: 3 (RXDAn) μPD70F3239: 4
Transmit data output	μPD70F3237: 3 (TXDAn) μPD70F3239: 4
Baud rate[Note] clock input	1 (ASCKA0)
Control register	UARTAn control register (UAnCTL0 to UAnCTL3) UARTAn option control register (UAnOPT0) UARTAn status register (UAnSTR)

Note IN the baud rate clock input which is supported only for UARTA0

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

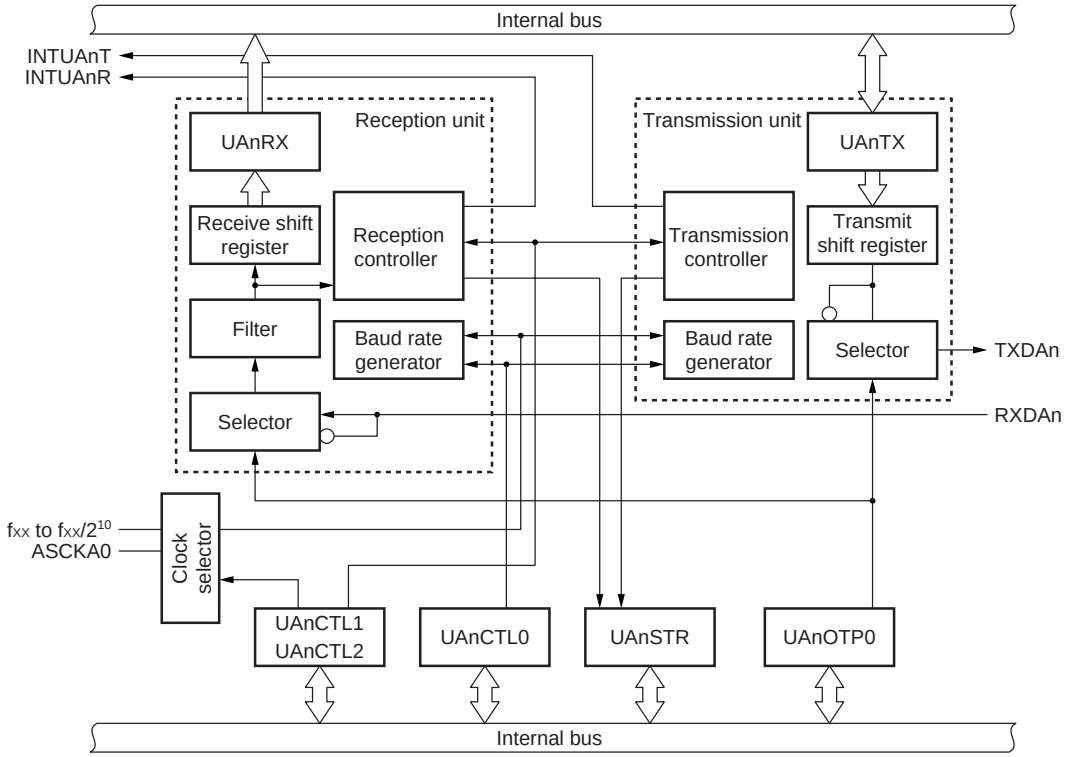
The pins of asynchronous serial interface A (UARTA) function alternately as port pins. For how to select the alternate functions, refer to the description of registers in **CHAPTER 4 PORT FUNCTIONS**.

Table 13-3. 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	P91/KR7		Serial receive data input (UARTA1)
RXDA2	P39/INTP8		Serial receive data input (UARTA2)
RXDA3	P80/INTP14		Serial receive data input (UARTA3)
TXDA0	P30	Output	Serial transmit data output (UARTA0)
TXDA1	P90/KR6		Serial transmit data output (UARTA1)
TXDA2	P38		Serial transmit data output (UARTA2)
TXDA3	P81		Serial transmit data output (UARTA3)
ASCKA0	P32/TIP00/TOP00	Input	Baud rate clock input of UARTA0

Remark The number of channels differs depending on the product.

Figure 13-1. Block Diagram of Asynchronous Serial Interface A



Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

Note: UARTA0 only.

13.2.1 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 A.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the input clock of the asynchronous serial interface A.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register that controls the baud rate of the asynchronous serial interface A.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls serial transfer by the asynchronous serial interface A.

(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.

13.3 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.

After reset: 10H R/W Address: UA0CTL0: FFFFFA00H, UA1CTL0: FFFFFA10H, UA2CTL0: FFFFFA20H, UA3CTL0: FFFFFA30H

	7	6	5	4	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

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.	

UAnTXE	Transmission operation enable
0	Stop transmission operation.
1	Enable transmission operation.
When the UAnTDL bit is cleared to 0, then the UAnTXE bit is cleared to 0, the output of the TXDAn pin is fixed to the high level. When the UAnTDL bit is set to 1, then the UAnTXE bit is set to 0, the output of the TXDAn pin is fixed to the low level. 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.	

(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. ・ Set the UA0DIR bits to "1" to execute transmission/reception in LIN format.	

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.
- Clear the UAnPS1 and UAnPS0 bits to “00” to execute transmission/reception in LIN format.

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. ・ Set the UA0CL bits to "1" to execute transmission/reception in LIN format.	

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 **13.5.9 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 or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: UA0CTL1: FFFFFA01H, UA1CTL1: FFFFFA11H, UA2CTL1: FFFFFA21H, UA3CTL1: FFFFFA31H

	7	6	5	4	3	2	1	0
UAnCTL1	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

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[Note] (ASCKA0 pin)
Other than above				Setting prohibited

Note The ASCKA0 pin can be used only when UARTA0 is used. Setting this bit is prohibited when UARTA1 to UARTA3 are used.

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.

After reset: FFH R/W Address: UA0CTL2: FFFFFA02H, UA1CTL2: FFFFFA12H, UA2CTL2: FFFFFA22H, UA3CTL2: FFFFFA32H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

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	$f_{XCLK}/255$

Remarks: 1. f_{XCLK} is the frequency of the base clock selected by the UAnCTL1 register.

2. Refer to Table 13.6 about setting samples of fxclk.

3. ×: Don't care

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.

(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.

After reset: 14H R/W Address: UA0OPT0: FFFFFA03H, UA1OPT0: FFFFFA13H, UA2OPT0: FFFFFA23H, UA3OPT0: FFFFFA33H

	7	6	5	4	3	2	1	0
UAnOPT0	UAnSFR	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

UAnSFR	SBF reception flag
0	When UAnCTL0 register's UAnPWR bit = UAnRXE bit = 0. Or, on normal completion of SBF reception
1	SBF reception in progress

- This bit indicates that SBF (Sync Brake Field) is received in LIN communication.
- In case of an SBF reception error, the UAnSRF bit is hold to 1, and then SBF reception is started again.
- The UAnSFR bit can only be read.

UAnSRT	SBF reception trigger
0	—
1	SBF reception trigger

- This is the reception trigger bit of SBF in LIN communication. It is always 0 when read. To receive SBF, set the UAnSRT bit to 1 to enable SBF reception.
- Set the UAnPWR bit and UAnRXE bit of the UAnCTL0 register to 1 and then set the UAnSRT bit.

UAnSTT	SBF transmission trigger
0	—
1	SBF transmission trigger

- This is the transmission trigger bit of SBF in LIN communication. It is always 0 when read.
- Set the UAnPWR bit and UAnTXE bit of the UAnCTL0 register to 1 and then set the UAnSTT bit.

(2/2)

UAnSLS2	UAnSLS1	UAnSLS0	SBF length selection
1	0	1	Outputs 13 bits (reset value).
1	1	0	Outputs 14 bits.
1	1	1	Outputs 15 bits.
0	0	0	Outputs 16 bits.
0	0	1	Outputs 17 bits.
0	1	0	Outputs 18 bits.
0	1	1	Outputs 19 bits.
1	0	0	Outputs 20 bits.
This bit can be set when the UAnPWR bit of the UAnCTL0 register = 0 or when the UAnTXE bit of the UAnCTL0 register = 0.			

UAnTDL	Transmit data level 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 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 **13.5.9 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

After reset: 00H R/W Address: UA0STR: FFFFFA04H, UA1STR: FFFFFA14H, UA2STR: FFFFFA24H, UA3STR: FFFFFA34H

	7	6	5	4	3	2	1	0
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

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.	

UAnFE	Framing error flag
0	• When UAnPWR bit of UAnCTL0 register = 0 or when UAnRXE bit of UAnCTL0 register = 0 • When 0 is written
1	When a stop bit is not detected on reception
• 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.	

UAnOVE	Overrun error flag
0	• When UAnPWR bit of UAnCTL0 register = 0 or when UAnRXE bit of UAnCTL0 register = 0 • When 0 is written
1	When receive data is set to the UAnRX register and the next reception operation is completed before that data is read
• 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.	

(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.

After reset: FFH R Address: UA0RX: FFFFFA06H, UA1RX: FFFFFA16H, UA2RX: FFFFFA26H, UA3RX: FFFFFA36H

	7	6	5	4	3	0	1	2
UAnRX								

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

(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 or 1-bit units.

Reset input sets this register to FFH.

After reset: FFH R/W Address: UA0TX: FFFFFA07H, UA1TX: FFFFFA17H, UA2TX: FFFFFA27H, UA3TX: FFFFFA37H

	7	6	5	4	3	2	1	0
UAnTX								

n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, µPD70F3237)
n = 0 to 3 (µPD70F3238, µPD70F3239)

13.4 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 13-4. 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.

A reception error interrupt can also be generated in this interrupt request signal if a reception error occurs.

Moreover, 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.

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

13.5 Operation

13.5.1 Data format

Full-duplex serial data is transmitted or received.

The transmit/receive data is in the format shown in Figure 13-2, consisting of a start bit, character bits, a parity bit, and 1 or 2 stop bits.

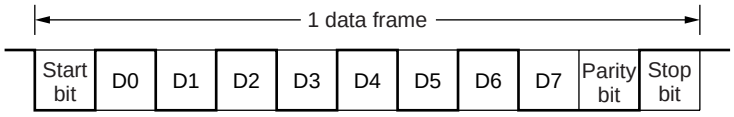
The character bit length in one data frame, parity, stop bit length, and whether data is transferred with the MSB or LSB first, are specified by the UAnCTL0 register.

The UAnTDL bit of the UAnOPT0 register is used to specify whether the signal output from the TXDAn pin is inverted or not.

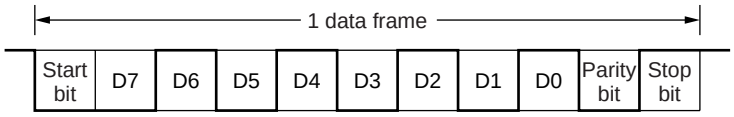
- Start bit … 1 bit
- Character bit … 7 or 8 bits
- Parity bit … Even parity, odd parity, 0 parity, or no parity
- Stop bit … 1 or 2 bits

Figure 13-2. Format of Transmit/Receive Data of UARTA

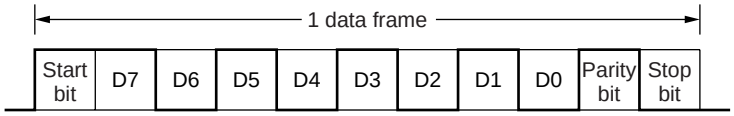
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



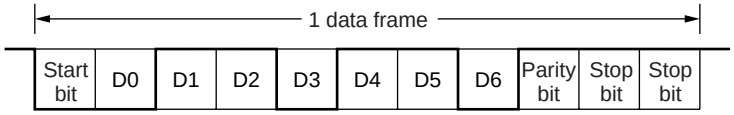
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



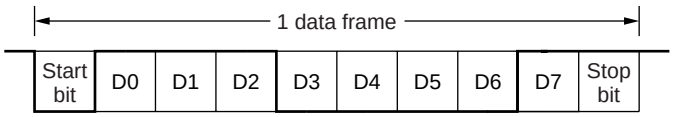
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inverted



(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



13.5.2 SBF transmission/reception format

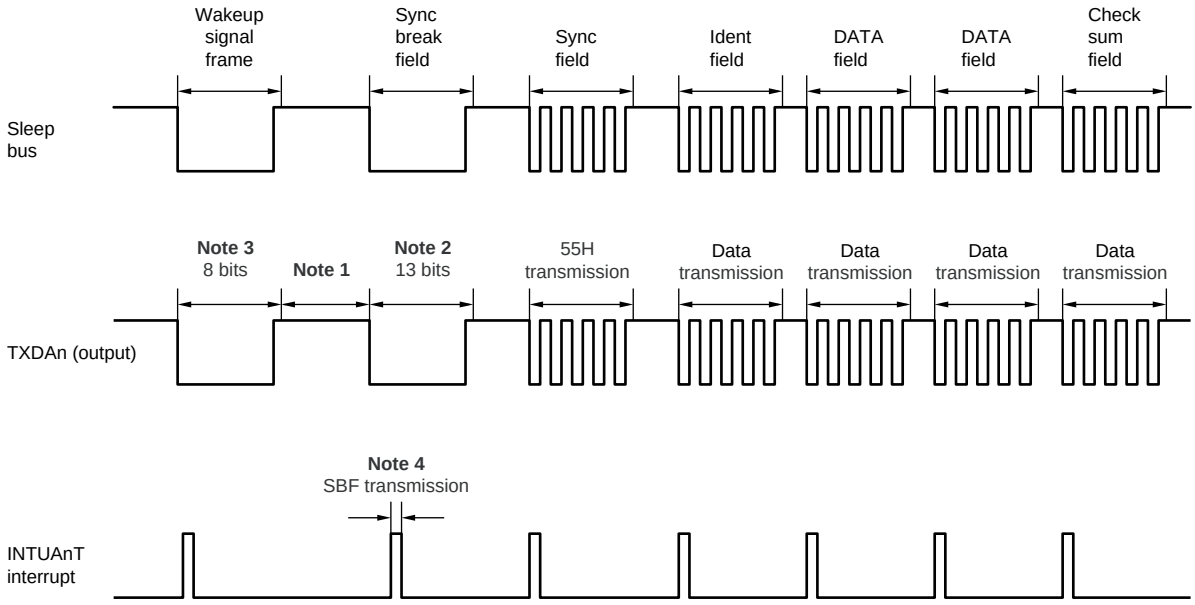
The V850ES/Fx2 has an SBF (Sync Break Field) transmission/reception control function as a LIN (Local Interconnect Network) function.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network. LIN communication is single-master communication, and up to 15 slaves can be connected to the LIN master via the LIN network. Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figure 13-3. Outline of Transmission Operation of LIN



Notes 1. The interval between each field is controlled by software.

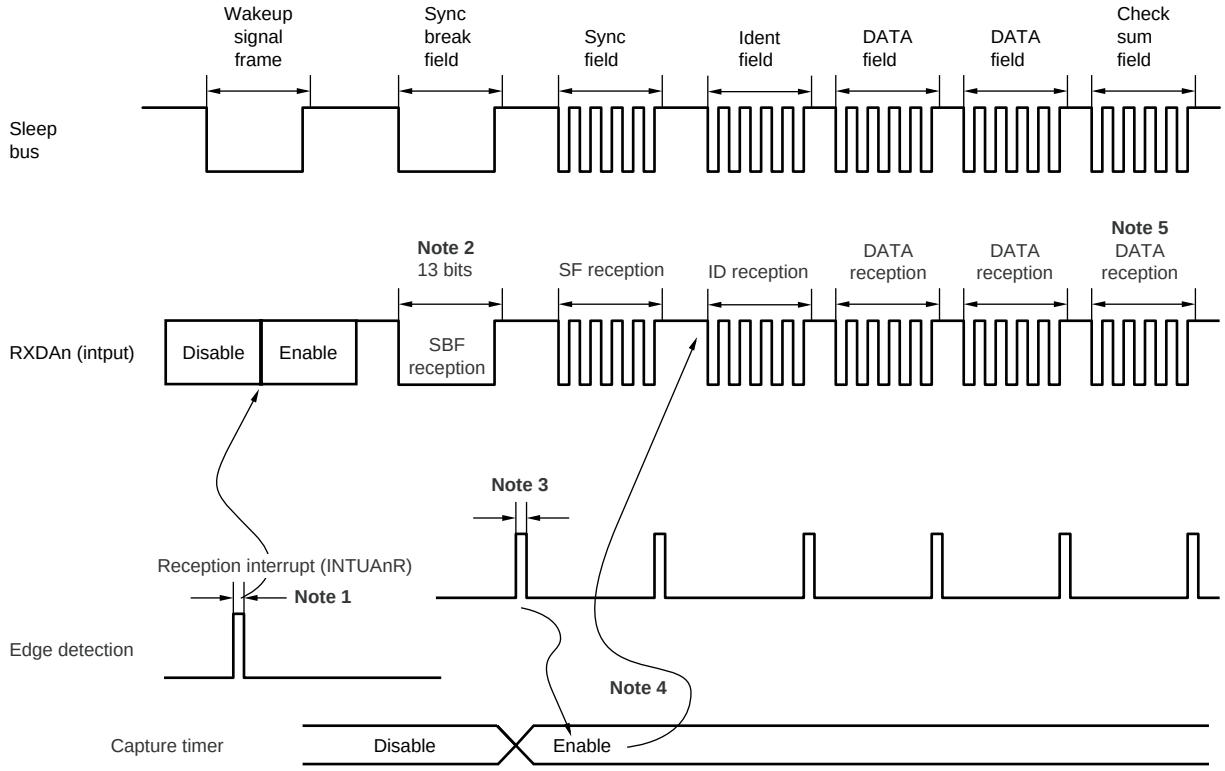
2. SBF is output by hardware. The output width is the bit length specified by the UAnSBL2 to UAnSBL0 bits of the UAnOPT0 register. If the output width must be adjusted more finely, the UAnBRST7 to UAnBRS0 bits of the UAnCTLn register can be used.

3. The wakeup signal frame is substituted by 80H transfer in the 8-bit mode.

4. A transmission enable interrupt request signal (INTUAnT) is output each time transmission is started. The INTUAnT signal is also output when SBF transmission is started.

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

★ **Figure 13-4. Outline of Reception Operation of LIN**



Notes 1. The wakeup signal is detected by the edge detector of the pin, and enables UARTAn and places it in the SBF reception mode.

2. Reception is performed until the STOP bit is detected. When SBF reception of 11 bits or more is detected, it is assumed that normal SBF reception has been completed, and the interrupt signal is output. If SBF reception of less than 11 bits is detected, it is assumed that an SBF reception error has occurred. No interrupt signal is output and UARTAn returns to the SBF reception mode.

3. When SBF reception is completed normally, the interrupt signal is output. The SBF reception complete interrupt enables a timer. Error detection by the UAnOVE, UAnPE, and UAnFE bits of the UAnSTR register is suppressed. Consequently, neither error detection processing of UART communication nor data transfer from the UARTAn receive shift register to UAnRX register is executed. The UARTAn receive shift register holds the default value FFH.

4. The RXDAn pin is connected to TI (capture input) of the timer and the transfer rate and baud rate error are calculated. After SF reception, UARTAn is no longer enabled. The value with the baud rate error corrected is set to the UAnCTL2 register to enable reception.

5. The checksum field is distinguished by software. After CSF reception, UARTAn is initialized and the SBF mode is set again by software.

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

13.5.3 SBF transmission

Transmission is enabled when the UAnPWR bit and UAnTXE bit of the UAnCTL0 register are set to 1, and SBF transmission is started by setting the SBF transmission trigger (UAnSTT bit of the UAnOPT0 register) to 1.

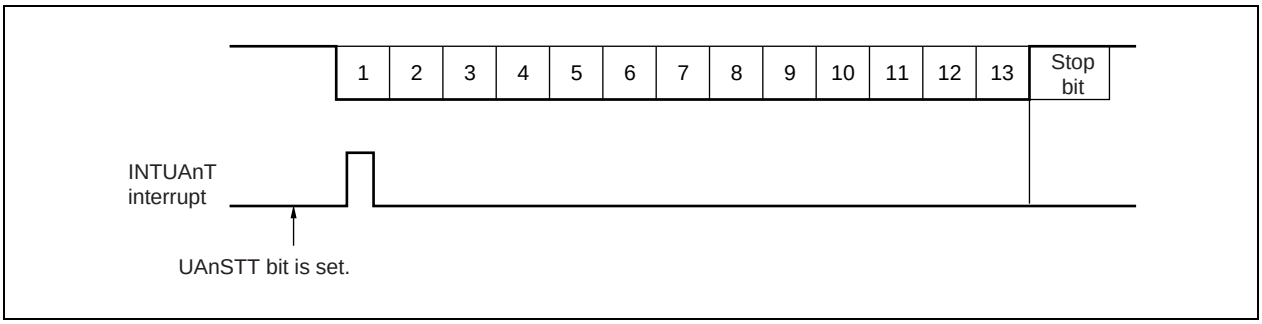
After that, a 13-bit to 20-bit low level, as specified by the UAnSLS2 to UAnSLS0 bits of the UAnOPT0 register, is output. A transmission enable interrupt request signal (INTUAnT) is generated when SBF transmission is started. After SBF transmission is completed, the UAnSTT bit is automatically cleared, and the UART transmission mode is restored.

The transmission operation is stopped until the data to be transmitted next is written to the UAnTX register or the SBF transmission trigger (UAnSTT bit) is set.

Cautions 1. This macro becomes error when SBF is transmitted with data reception because it doesn't assume the thing that SBF is transmitted while receiving data.

2. Set (1) neither SBF reception trigger bit （UAnSRT） nor SBF transmission trigger bit (UAnSTT) while receiving SBF (UAnSRF = 1).

Figure 13-5. SBF Transmission



13.5.4 SBF 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.

When the SBF reception trigger (UAnSRT bit of the UAnOPT0 register) is set to 1, UARTA waits for SBF reception.

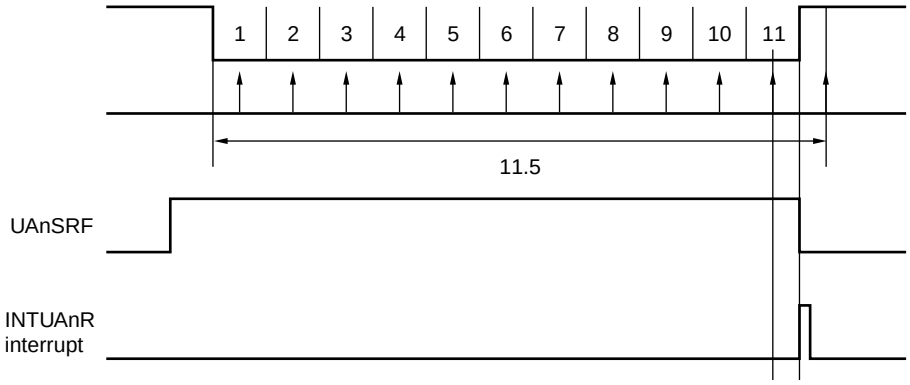
In the SBF reception waiting status, the RXDAn pin is monitored and the start bit is detected, in the same manner as in the reception wait status of UART.

When the start bit is detected, reception is started, and the internal counter counts up at the selected baud rate.

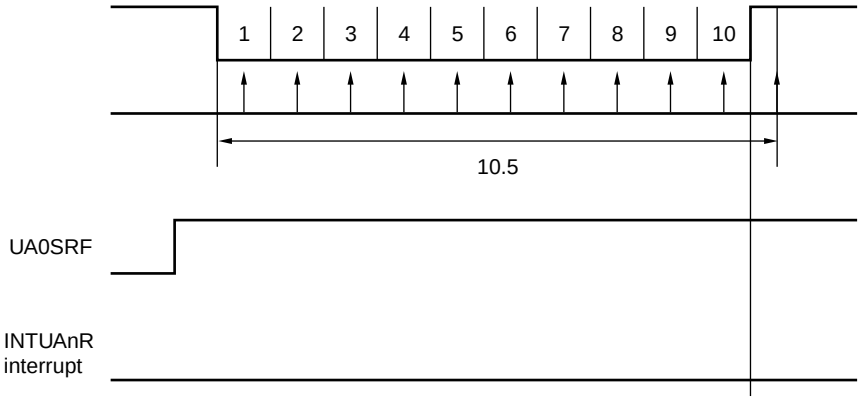
When the stop bit is received, a reception complete interrupt request signal (INTUAnR) is generated as normal processing, if the width of SBF is 11 bits or longer. The UAnSRF bit of the UAnOPT0 register is automatically cleared, and SBF reception is completed. Error detection by the UAnOVE, UAnPE, and UAnFE bits of the UAnSTR register is suppressed, and error detection processing of UART communication is not performed. Moreover, data is not transferred from the UARTAn receive shift register to the UAnRX register, and the UAnRX register holds the default value FFH. If the width of SBF is 10 bits or less, the interrupt does not occur, reception is completed, and the SBF reception mode is restored again, as error processing. At this time, the UAnSRF bit is not cleared.

Figure 13-6. SBF Reception

(a) Normal SBF reception (STOP bit is detected when SBF width exceeds 10.5 bits)



(b) SBF reception error (STOP bit is detected when SBF width is 10.5 bits or less)



13.5.5 UART transmission

When the UAnPWR bit of the UAnCTL0 register is set to 1, the TXDAn pin outputs a high level.

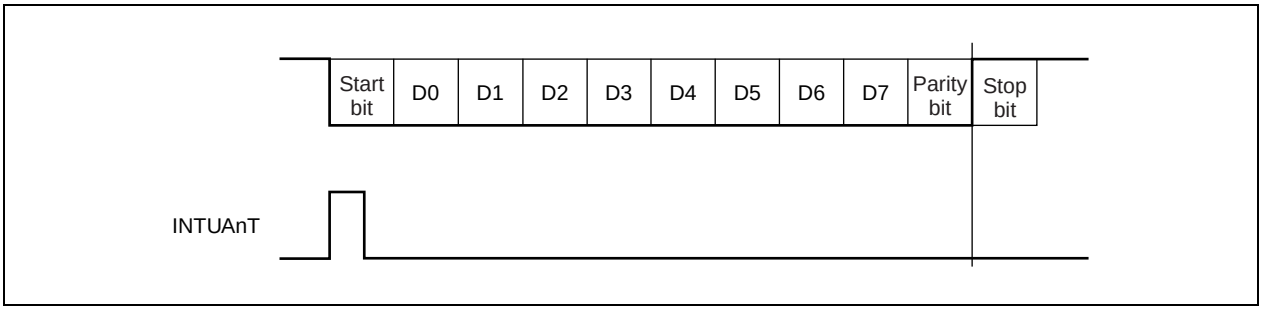
If the UAnTXE bit of the UAnCTL0 register is subsequently set to 1, transmission is enabled. Transmission is started by writing transmit data to the UAnTX register. A start bit, parity bit, and stop bit are automatically appended to the transmit data.

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 13-7. UART Transmission



13.5.6 Procedure of continuous transmission

With UARTAn, 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.
The communication rate from the stop bit to the following start bit expands more than usually for two clocks of the operation clock at a continuous transmission.

Figure 13-8. Processing Flow of Continuous Transfer

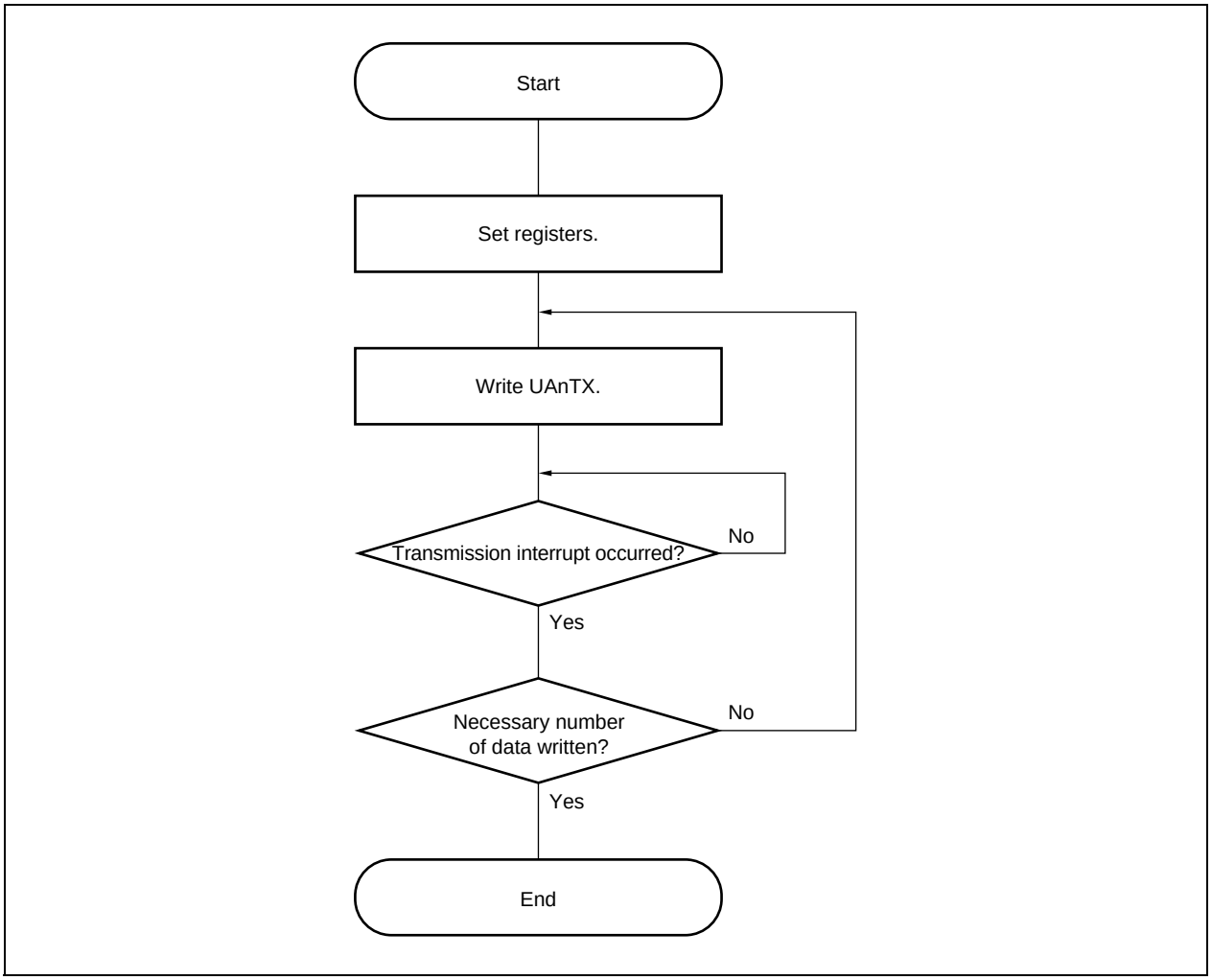
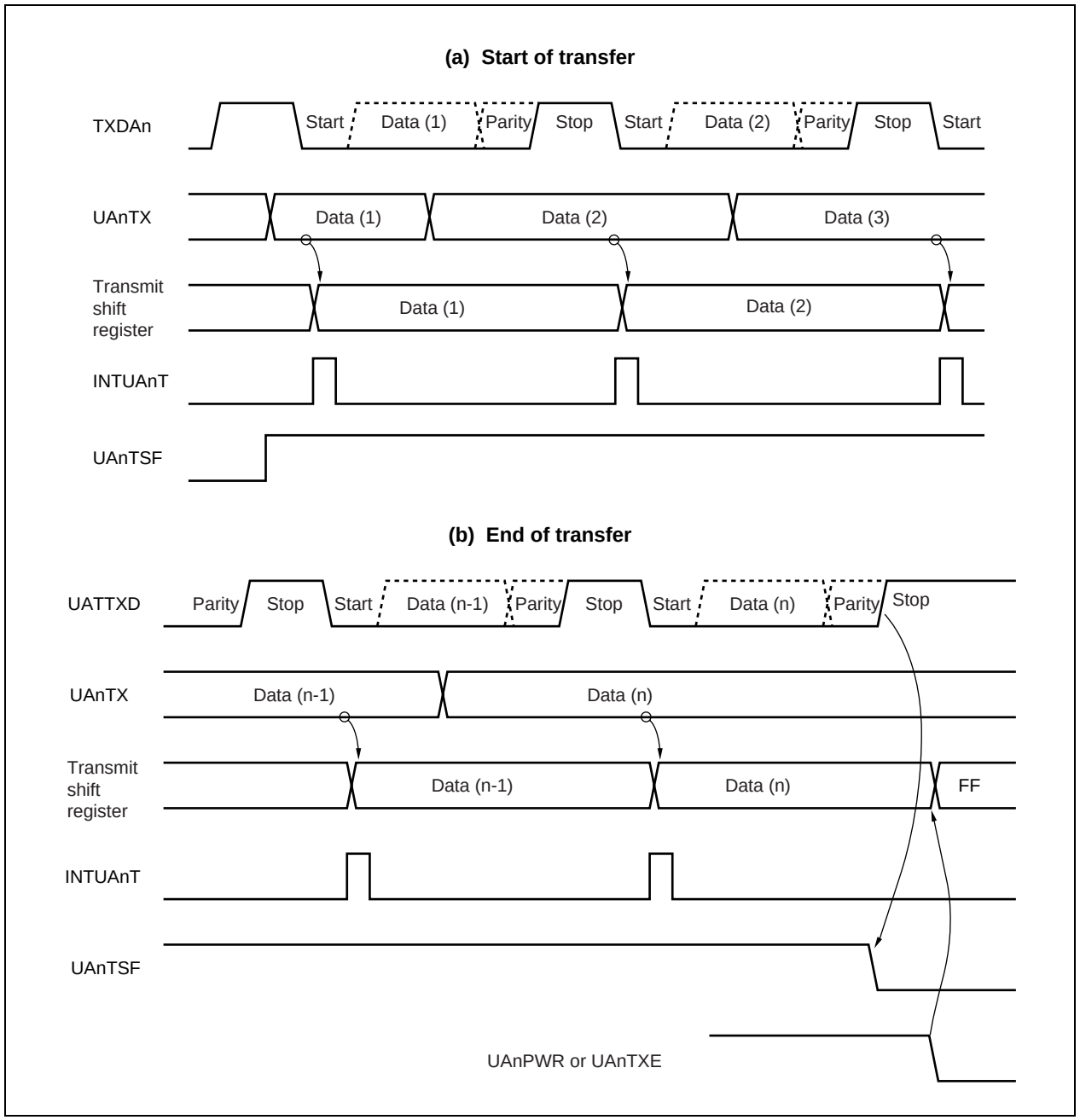


Figure 13-9. Timing of Continuous Transmission Operation



13.5.7 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.

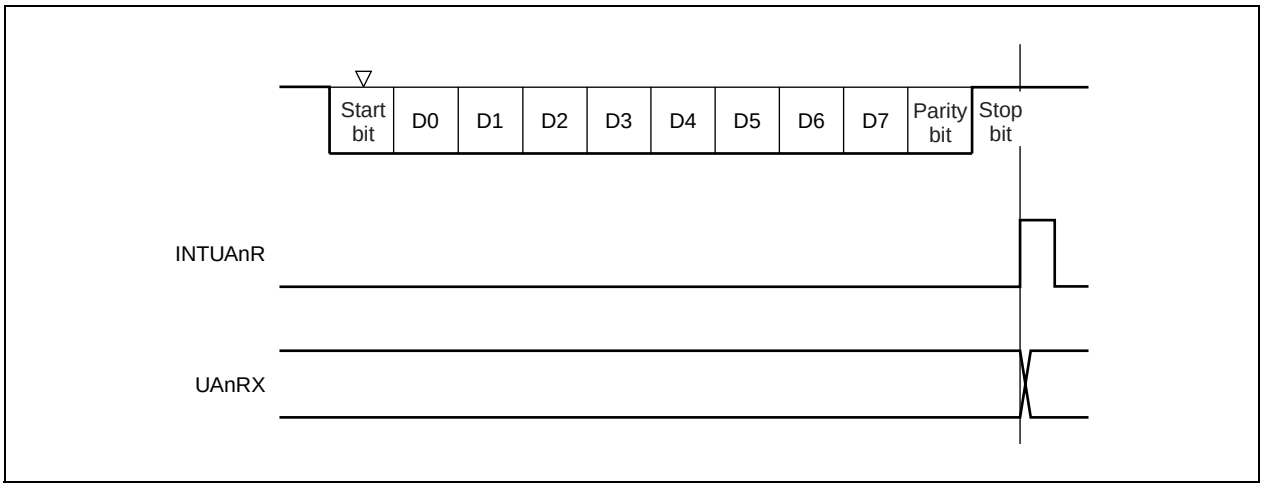
To recognize the start bit, a two-stage detection routine is used.

First the falling edge of the RXDAn pin is detected and sampling is started. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. 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 13-10. 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 request signal (INTUAnR) has been generated, and clear the UAnPWR or UAnRXE bit to 0. If the UAnPWR or UAnRXE bit is cleared to 0 before the NTUAnR signal is generated, the read value of the UAnRX register cannot be guaranteed.

4. If receive completion processing (INTUAnR signal generation) of UARTAn and the UAnPWR bit = 0 or UAnRXE bit = 0 conflict, the INTUAnR signal may be generated in spite of these being no data stored in the UAnRX register. To complete reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIF) of the UAnRIC register, after setting (1) the interrupt mask flag (UAnRMK) of the interrupt control register (UAnRIC) and then set (1) the UAnPWR bit = 0 or UAnRXE bit = 0.

13.5.8 Reception errors

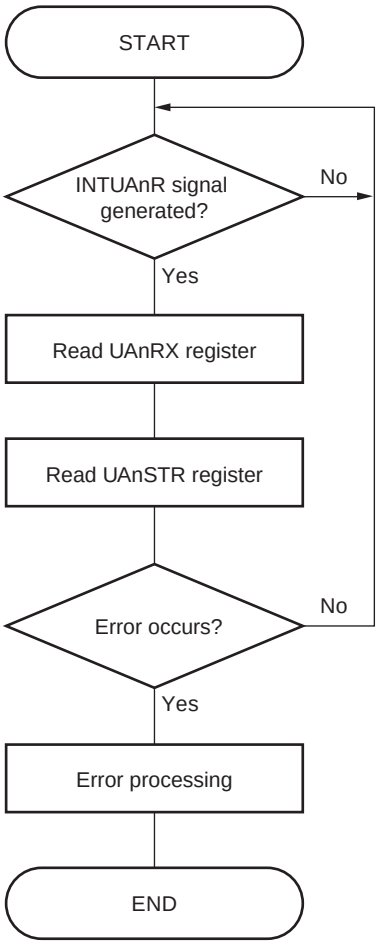
Reception errors are classified into three types: parity errors, framing errors, and overrun errors. As a result of receiving data, an error flag is set in the UAnSTR register, and a reception complete interrupt request signal (INTUAnR) is generated.

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.

- Receive data read flow

Figure 13-11. Receive data read flow



Caution When an INTUAnR signal is generated, the UAnSTR register must be read to check for errors.

- Reception error causes

Table 13-5. 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.

When reception errors occur, perform the following procedures depending upon the kind 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, and then perform error processing.

13.5.9 Types and operation of parity

Caution When using the LIN function, fix the UAnPS1 and UAnPS0 bits of the UAnCTL0 register to “00”.

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.

★ **13.5.10 Noise filter of receive data**

The RXDAn pin is sampled using the UART internal clock (f_{XCLK}).

When the same sampling value is read twice, the match detector output changes and the RXDAn signal is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 13-13**). See **13.6 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 13-12, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Figure 13-12. Noise Filter Circuit

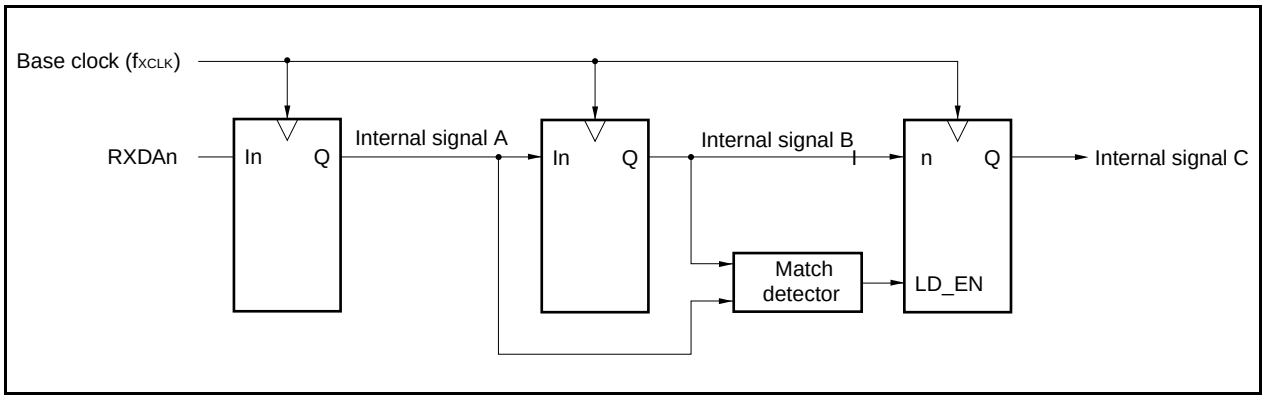
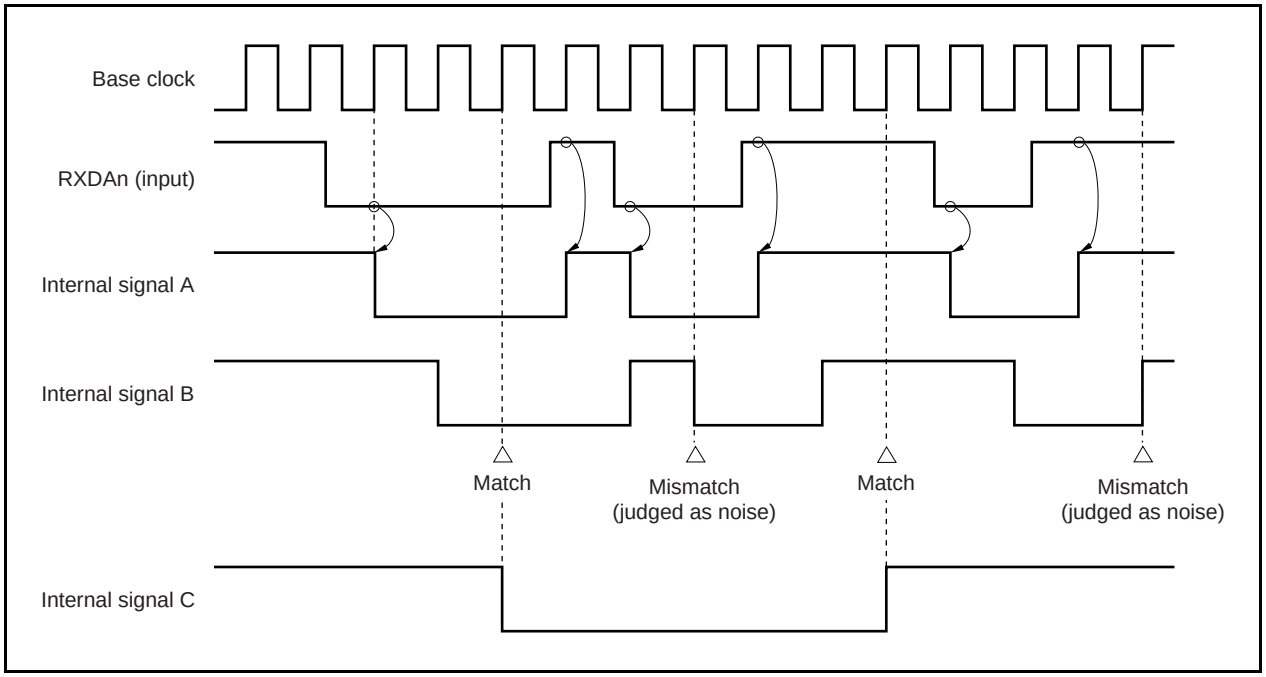


Figure 13-13. Timing of RXDAn Signal Judged as Noise



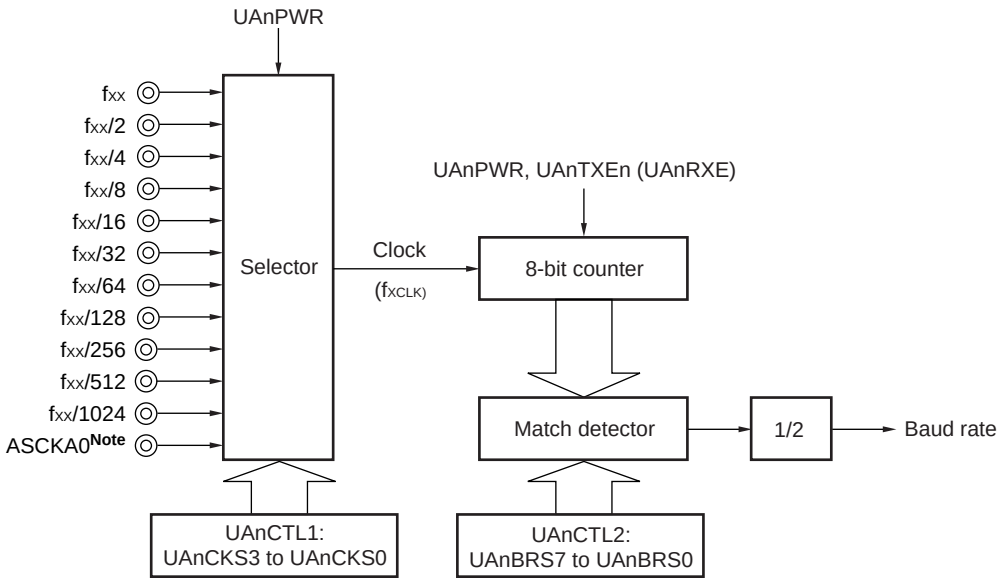
13.6 Dedicated Baud Rate Generator

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 13-14. Configuration of Baud Rate Generator



Note The ASCKA0 pin can be used only for UARTA0. It cannot be used for UARTA1 to UARTA3.

Remarks 1. n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

2. f_{XX}: Internal system clock

3. f_{XCLK} = Base clock frequency

(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}.

(b) Generation of serial clock

A serial clock can be generated in accordance with the setting of the UAnCTL1 and UAnCTL2 registers

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 **13.3 (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 **13.3 (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.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

$$= \left(\frac{f_{XCLK}}{2 \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in (7) Permissible baud rate range for reception.

Example: Frequency of base clock = 20 MHz = 20,000,000 Hz

Set value of UAnBRS7 to UAnBRS0 bits of UAnCTL2 register = 01000001B (k = 65)

Target baud rate = 153,600 bps

Baud rate = 20,000,000/ (2 × 65)

= 153,846 [bps]

Error = (153,846/153,600 – 1) × 100

= 0.160 [%]

(6) Example of baud rate setting

Table 13-6. Baud Rate Generator Set Data

Baud Rate (bps)	f_{XX} = 20 MHz			f_{XX} = 16 MHz			f_{XX} = 10 MHz		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	09H	41H	0.16	0AH	1AH	0.16	08H	41H	0.16
600	08H	41H	0.16	0AH	0DH	0.16	07H	41H	0.16
1200	07H	41H	0.16	09H	0DH	0.16	06H	41H	0.16
2,400	06H	41H	0.16	08H	0DH	0.16	05H	41H	0.16
4,800	05H	41H	0.16	07H	0DH	0.16	04H	41H	0.16
9,600	04H	41H	0.16	06H	0DH	0.16	03H	41H	0.16
19,200	03H	41H	0.16	05H	0DH	0.16	02H	41H	0.16
31,250	01H	A0H	0.00	01H	80H	0.00	00H	A0H	0.00
38,400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76,800	00H	82H	0.16	00H	68H	0.16	00H	41H	0.16
153,600	00H	41H	0.16	00H	34H	0.16	00H	21H	–1.36
312,500	00H	20H	0.00	00H	1AH	–1.54	00H	10H	0.00

Remarks: 1. f_{XX}: Internal system clock
ERR: Baud rate error [%]

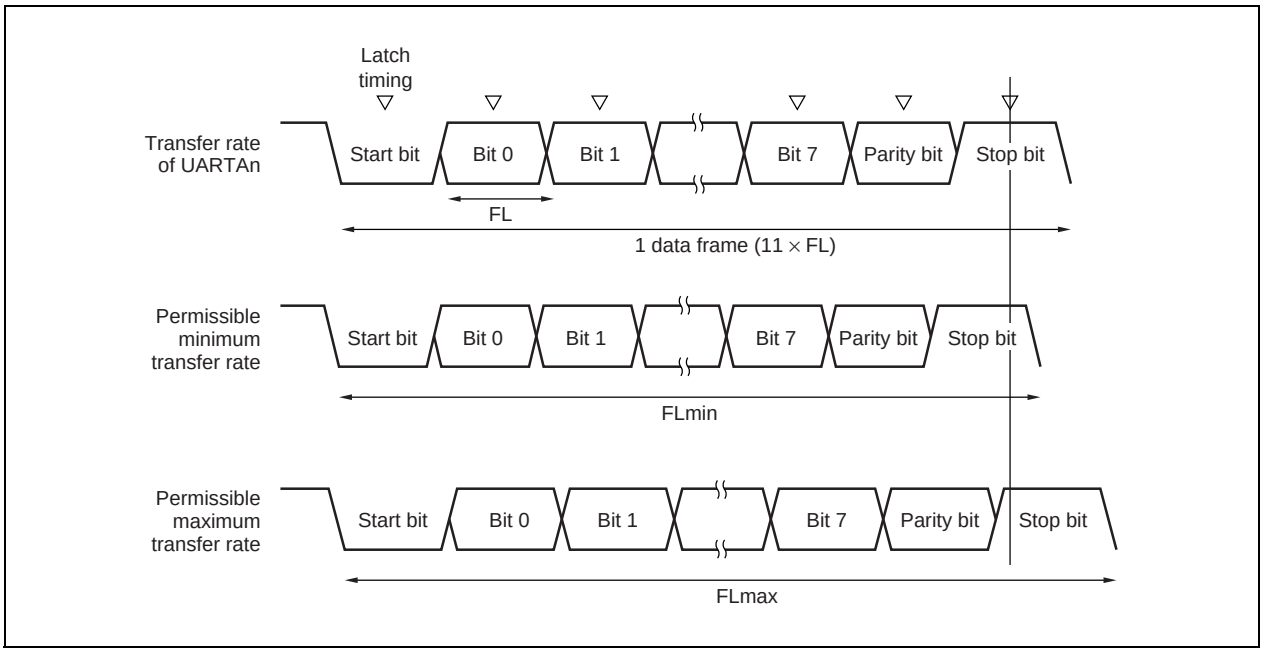
2. n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

(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 13-15. Permissible Baud Rate Range for Reception



Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2)
n = 0 to 2 (V850ES/FG2, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

As shown in Figure 13-15, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: UARTAn baud rate (n = 0 to 3)
k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 3)
FL: 1-bit data length
Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

Obtaining the allowable baud rate error for UARTAn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 13-7. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	–2.43%
8	+3.52%	–3.61%
20	+4.26%	–4.30%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.72%

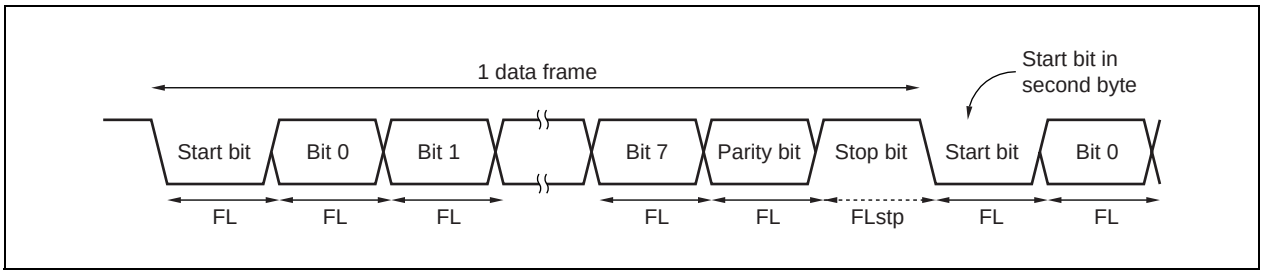
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 3)

(8) 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 13-16. 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.

$$\text{Transfer rate} = 11 \times FL + 2/f_{XCLK}$$

13.7 Cautions

(1) When the clock supply to UARTAn is stopped (for example, in IDLE1, IDLE2, or STOP mode), the operation stops with each register retaining the value it had immediately before the clock supply was stopped. The TXDAn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn, and UAnCTL0.UAnTXEn bits to 000.

(2) The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0).

(3) In UARTAn, the interrupt caused by a communication error does not occur. When performing the transfer of transmit data and receive data using DMA transfer, error processing cannot be performed even if errors (parity, overrun, framing) occur during transfer. Either read the UAnSTR register after DMA transfer has been completed to make sure that there are no errors, or read the UAnSTR register during communication to check for errors.

(4) Start up the UARTAn in the following sequence.
<1> Set the UAnCTL0.UAnPWR bit to 1.
<2> Set the ports.
<3> Set the UAnCTL0.UAnTXE bit to 1, UAnCTL0.UAnRXE bit to 1.

(5) Stop the UARTAn in the following sequence.
<1> Set the UAnCTL0.UAnTXE bit to 0, UAnCTL0.UAnRXE bit to 0.
<2> Set the ports and set the UAnCTL0.UAnPWR bit to 0 (it is not a problem if port setting is not changed).

(6) In transmit mode (UAnCTL0.UAnPWR bit = 1 and UAnCTL0.UAnTXE bit = 1), do not overwrite the same value to the UAnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.

(7) In continuous transmission, the communication rate from the stop bit to the next start bit is extended 2 base clocks more than usual. However, the reception side initializes the timing by detecting the start bit, so the reception result is not affected.

(8) When break command is based and UARTA receives data for on-chip debug mode, over run error is occurred.

CHAPTER 14 3-WIRE SERIAL INTERFACE (CSIB)

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

The V850ES/Fx2 includes a 3-wire serial interface (CSIB).

The number of channels differs depending on the product. Following table shows the number of channels of each product.

Table 14-1. Number of Channels of 3-wire serial interface B

Product Name (Part Number)	Number of Channels
V850ES/FE2	2 (CSIB0 to CSIB1)
V850ES/FF2	
V850ES/FG2	
V850ES/FJ2	3 (CSIB0 to CSIB2)

14.1 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 SOBn: Serial data output
  SIBn: Serial data input
  $\overline{\text{SCKBn}}$: Serial clock I/O
- Transmission mode, reception mode, and transmission/reception mode selectable
- Transfer rate : 8 Mbps - 4.9 kbps (fxx=20 MHz, using internal clock)

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2, V850ES/FG2)
n = 0 to 2 (V850ES/FJ2)

14.2 Configuration

CSIBn consists of the following hardware

Table 14-2. configuration of CSIBn

Item	Configuration
Register	CSIBn reception data register (CBnRX) CSIBn transmit data register (CBnTX)
Reception data input	SIBn
Transmit data output	SOBn
Serial clock I/O	$\overline{\text{SCKBn}}$
Control register	CSIBn control register (CBnCTL0 to CBnCTL2) CSIBn status register (CBnSTR)

Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2, V850ES/FG2)
n = 0 to 2 (V850ES/FJ2)

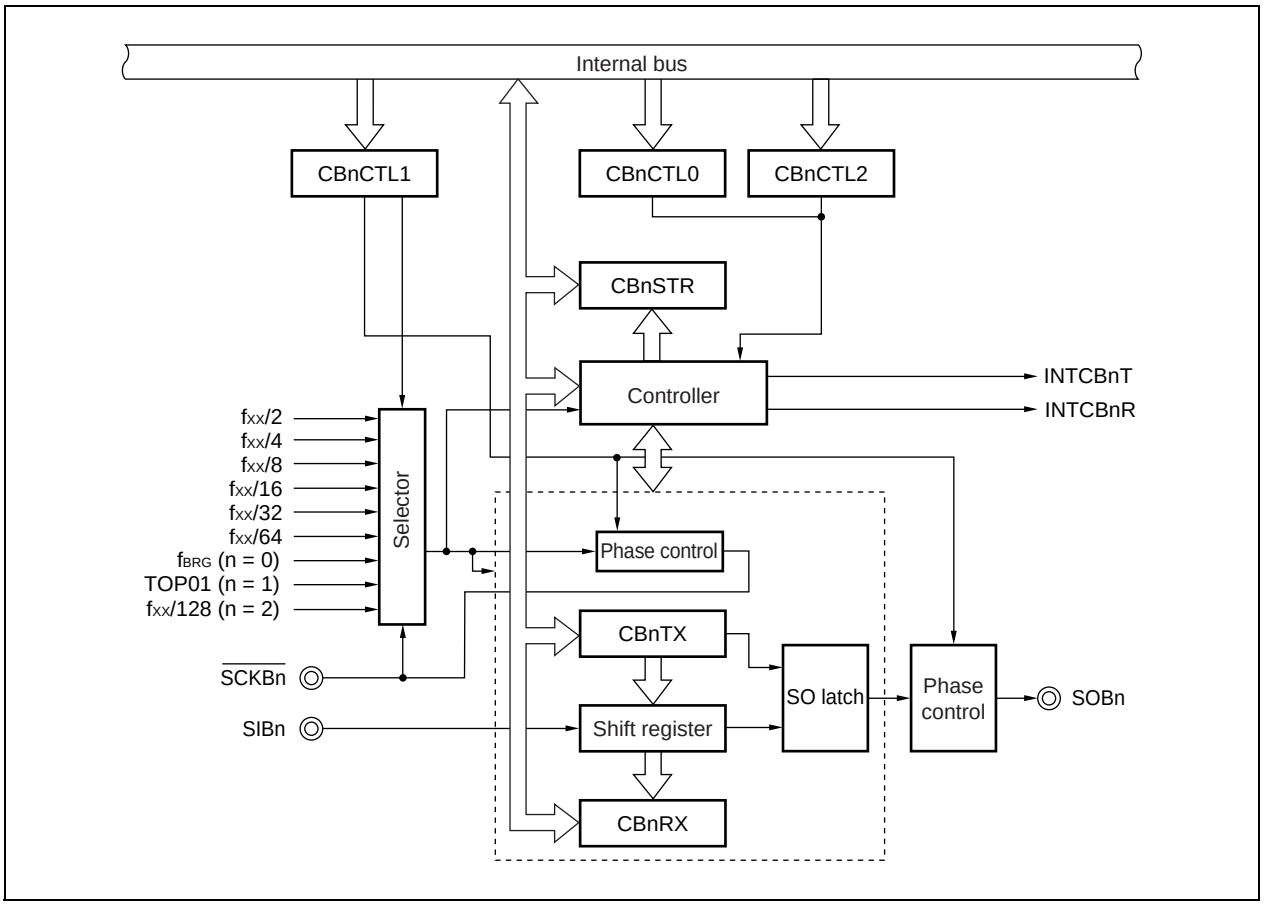
The pins of the 3-wire serial interface (CSIBn) function alternately as port pins. For how to select the alternate function, refer to the descriptions of the registers in **CHAPTER 4 PORT FUNCTIONS**.

Table 14-3. 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/TIP20/TOP20		Serial receive data input (CSIB1)
SIB2	P910		Serial receive data input (CSIB2)
SOB0	P41	Output	Serial transmit data input (CSIB0)
SOB1	P98		Serial transmit data input (CSIB1)
SOB2	P911		Serial transmit data input (CSIB2)
$\overline{\text{SCKB0}}$	P42	I/O	Serial clock I/O (CSIB0)
$\overline{\text{SCKB1}}$	P99		Serial clock I/O (CSIB1)
$\overline{\text{SCKB2}}$	P912		Serial clock I/O (CSIB2)

Remark The number of channels differs depending on the product.

Figure 14-1. Block Diagram of 3-Wire Serial Interface



Remark n = 0 to 1 (V850ES/FE2, V850ES/FF2, V850ES/FG2)
n = 0 to 2 (V850ES/FJ2)

(1) CSIBn receive data register (CBnRX)

The CBnRX register is a 16-bit buffer register that holds receive data.

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

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.

In addition to reset input, the CBnRX register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

After reset: 0000H R Address: CB0RX: FFFFFD04H, CB1RX: FFFFFD14H, CB2RX: FFFFFD24H

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

(n = 0 to 2)

(2) 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.

After reset: 0000H R/W Address: CB0TX: FFFFFD06H, CB1TX: FFFFFD16H, CB2TX: FFFFFD26H

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

(n = 0 to 2)

Remark The communication start conditions are shown below.

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):	Write to CBnTX register
Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):	Write to CBnTX register
Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):	Read from CBnRX register

14.3 Control Registers

(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.

(1/2)

After reset: 01H R/W Address: CB0CTL0: FFFFFD00H, CB1CTL0: FFFFFD10H, CB2CTL0: FFFFFD20H

	7	6	5	4	3	2	1	0
CBnCTL0	CBnPWR	CBnTXE Note1	CBnRXE Note1	CBnDIR Note1	0	0	CBnTMS Note1	CBnSCE

(n = 0 to 2)

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 Note1	Specification of transmission operation stop or enable
0	Stop transmission.
1	Enable transmission.
When the CBxTXE bit is cleared to 0, the serial output pin SOBn is fixed to the low level and communication is stopped.	

CBnRXE Note1	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 Note1	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

CBnTMS Note1	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.	

(2/2)

CBnSCE	Specification of start transfer disable or enable
0	Disable transfer operation.
1	Enable transfer operation.

• In master mode

This bit enables or disables the communication start trigger.

(a) In single transmission or transmission/reception mode, or continuous transmission or continuous transmission/reception mode
A communication operation can be started only by writing data to the CBnTX register when the CBnSCE bit is 1.
Set the CBnSCE bit to 1.

(b) In single reception mode
Clear the CBnSCE bit to 0 before reading the last receive data because reception is started by reading the receive data (CBnRX register)[Note 2] to disable the reception startup.

(c) In continuous reception mode
Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed to disable the reception startup after the last data is received[Note 3].

• In slave mode

This bit enables or disables the communication start trigger.
Set the CBnSCE bit to 1.

[Usage of CBnSCE bit]

• In single reception mode

<1> When reception of the last data is completed by INTCBnR interrupt servicing, clear the CBnSCE bit to 0 before reading the CBnRX register.

<2> After confirming the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception. To continue reception, set the CBnSCE bit to 1 to start up the next reception by dummy-reading the CBnRX register.

• In continuous reception mode

<1> Clear the CBnSCE bit to 0 during the reception of the last data by INTCBnR interrupt servicing.

<2> Read the CBnRX register.

<3> Read the last reception data by reading the CBnRX register after acknowledging the CBnTIR interrupt.

<4> After confirming the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception. To continue reception, set the CBnSCE bit to 1 to wait for the next reception by dummy-reading the CBnRX register.

Notes 1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR can be set to 1 at the same time as these bits are rewritten.

2. If the CBnSCE bit is read while it is 1, the next communication operation is started.

3. The CBnSCE bit is not cleared to 0 one communication clock before the completion of the last data reception, the next communication operation is automatically started.

Caution 1: To forcibly suspend transmission/reception, clear the CBnPWR bit instead of the CBnRXE bit and the CBnTXE bit to 0. At this time, the clock output is stopped.

2: Be sure to clear bits 3 and 2 to 0.

(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.

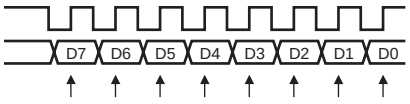
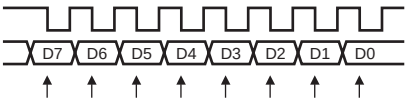
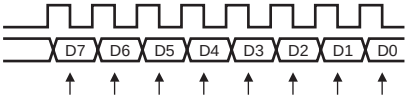
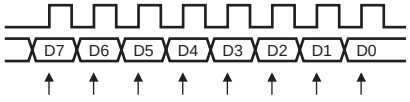
Reset input clears this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when the CBnTXE and CBnRXE bits are 0.

After reset: 00H R/W Address: CB0CTL1: FFFFFD01H, CB1CTL1: FFFFFD11H, CB2CTL1: FFFFFD21H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0 to 2)

CBnCKP	CBnDAP	Specification of transmission/reception timing of data of $\overline{\text{SCKBn}}$
0	0	
0	1	
1	0	
1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Input clock			Mode
			n = 0	n = 1	n = 2	
0	0	0	$f_{XX}/2$			Master mode
0	0	1	$f_{XX}/4$			Master mode
0	1	0	$f_{XX}/8$			Master mode
0	1	1	$f_{XX}/16$			Master mode
1	0	0	$f_{XX}/32$			Master mode
1	0	1	$f_{XX}/64$			Master mode
1	1	0	f_{BRG} Note	TMP0(TOP01)	$f_{XX}/128$	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 **14.8 Prescaler 3**.

(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 or 1-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.

After reset: 00H R/W Address: CB0CTL2: FFFFFD02H, CB1CTL2: FFFFFD12H, CB2CTL2: FFFFFD22H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

(n = 0 to 2)

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.

Remark. ×: don't care

(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 CBnTSF 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.

After reset: 00H R/W Address: CB0STR: FFFFFD03H, CB1STR: FFFFFD13H, CB2STR: FFFFFD23H

	7	6	5	4	3	2	1	0
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0 to 2)

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating

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

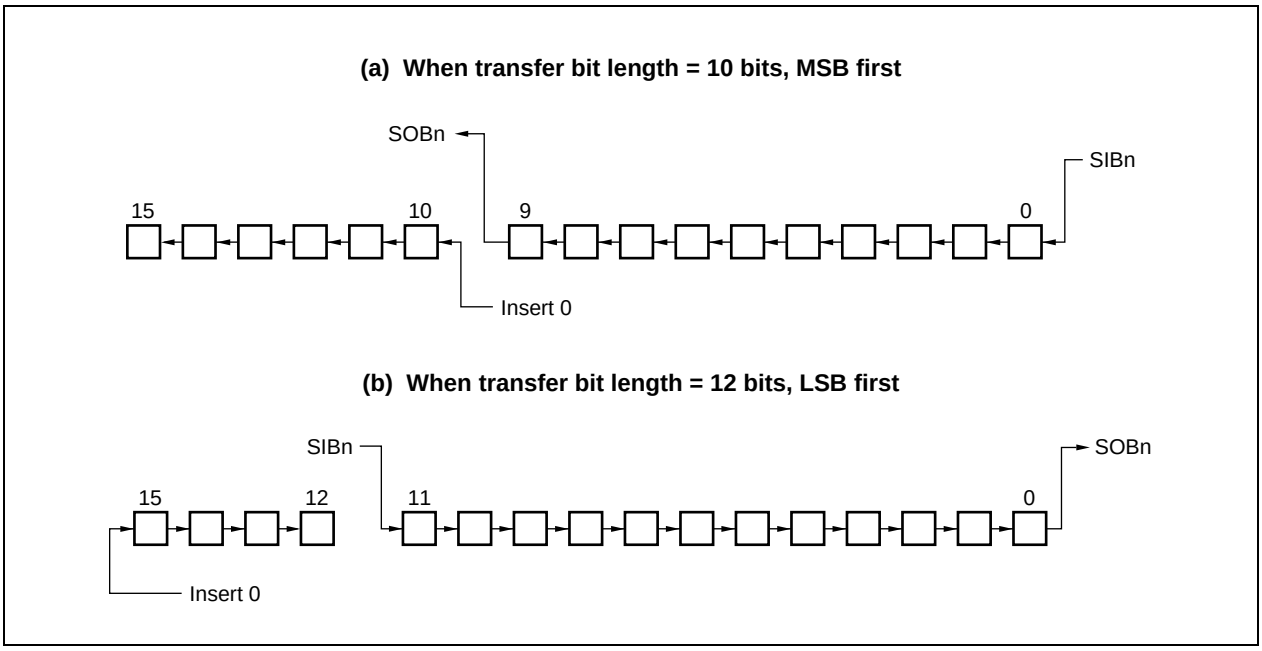
- An overrun error occurs when the next reception completes without reading the value of the receive buffer by CPU, upon completion of the receive operation.
  The CBnOVE flag indicates occurrence of this overrun error.
- The CBnOVE bit is valid also in the single transfer mode. Therefore, when only using transmission, note the following.
  - Do not check the CBnOVE flag.
  - Read this bit even if reading the reception data is not required.
- The OBnOVE flag is cleared when 0 is written to it. It is not set when 1 is written to it.

14.4 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 14-2. Changing Transfer Data Length



14.5 Interrupt Request Signals

CSIBn can generate the following two types of interrupt request signals.

- Reception complete interrupt request signal (INTCBnR)
- Transmission enable interrupt request signal (INTCBnT)

Of these two interrupt request signals, the reception complete interrupt request signal has the higher priority by default, and the priority of the transmission enable interrupt request signal is lower.

Table 14-4. 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.

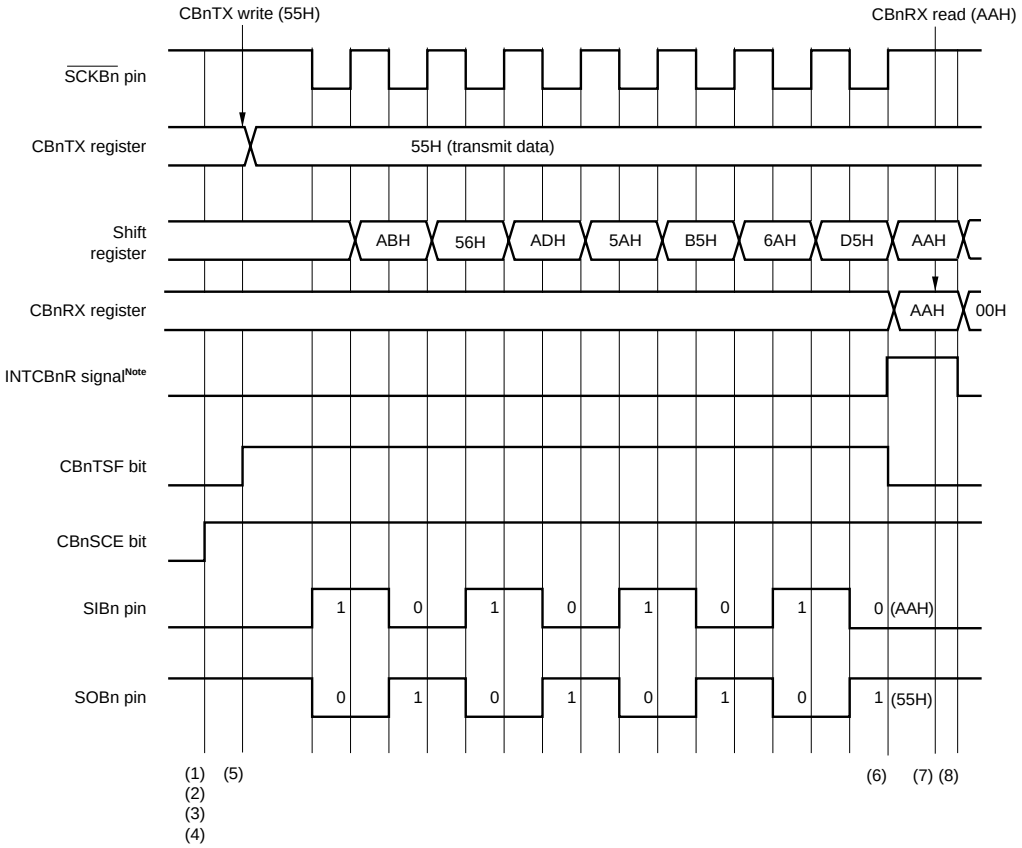
(2) Transmission enable interrupt request signal (INTCBnT)

The transmission enable interrupt request signal is generated when transmit data is transferred from the CBnTX register while transmission enabled.

14.6 Operation

14.6.1 Single transfer mode (master mode, transmission/reception mode)

This section shows a case of MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**, and transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).



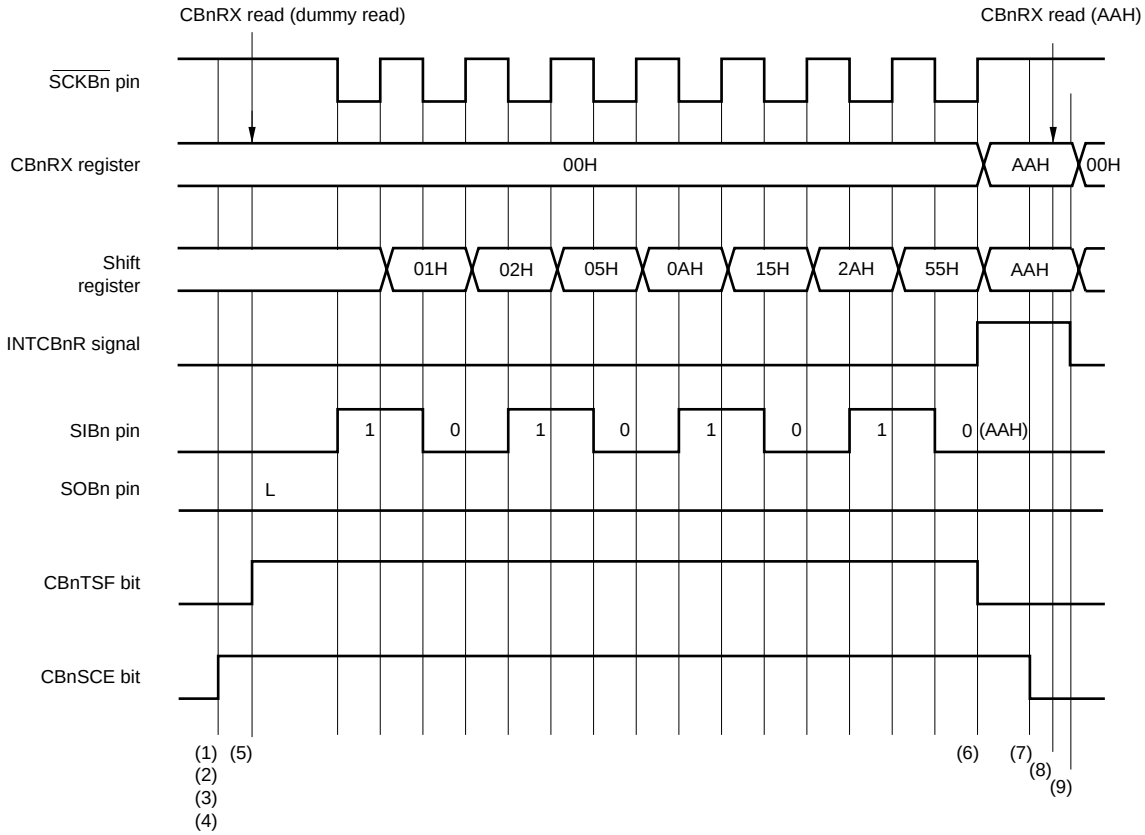
(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Write transfer data to the CBnTX register (transmission start).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) Read the CBnRX register before clearing the CBnPWR bit to 0.
(8) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop operation of CSIBn (end of transmission/reception).

Note In single transmission or single transmission/reception mode, the INTCBnT signal is not generated. When communication is complete, the INTCBnR signal is generated.

Remark The processing of steps (3) and (4) can be set simultaneously.

14.6.2 Single transfer mode (master mode, reception mode)

This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 1 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**, transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).

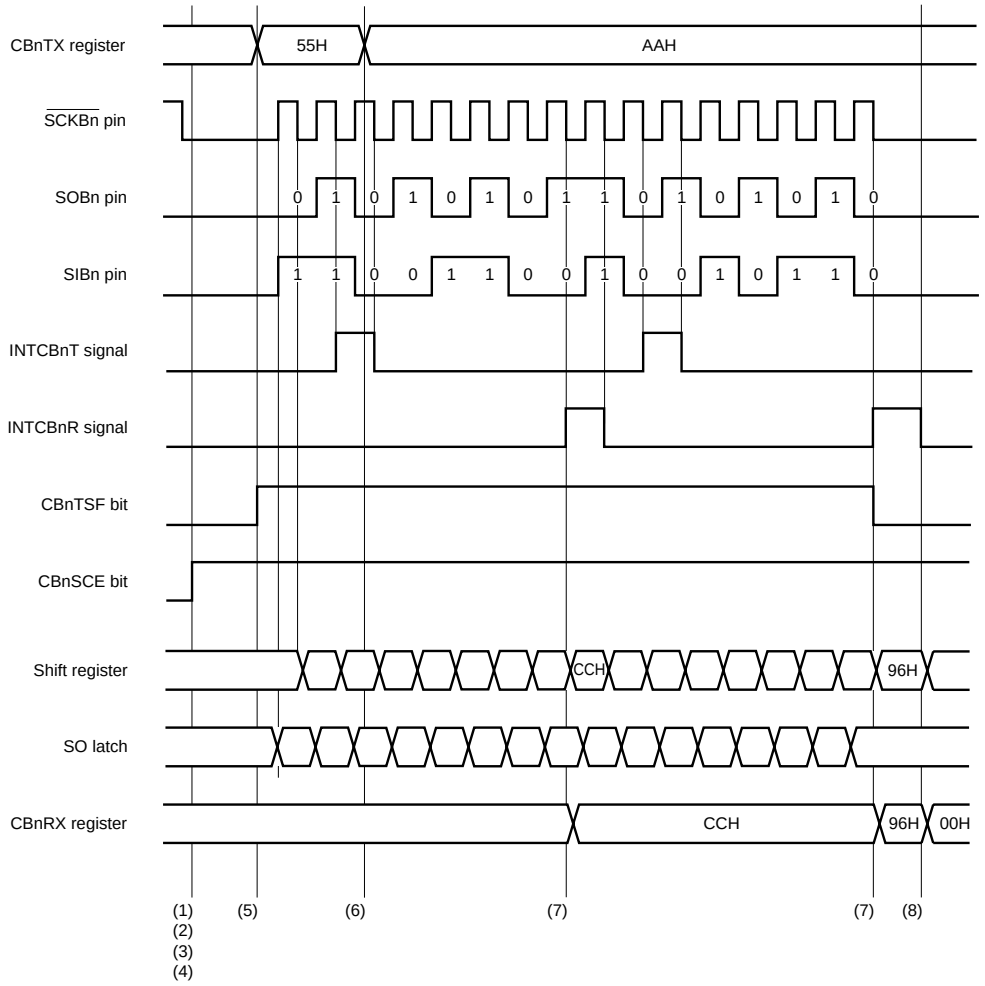


(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) Set the CBnSCE bit to 0 to set the final receive data status.
(8) Read the CBnRX register.
(9) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the CSIBn operation (end of reception).

Remark The processing of steps (3) and (4) can be set simultaneously.

14.6.3 Continuous mode (master mode, transmission/reception mode)

This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 3 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).



(1) Clear the CBnCTL0.CBnPWR bit to 0.

(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.

(3) Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.

(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.

(5) Write transfer data to the CBnTX register (transmission start).

(6) The transmission enable interrupt request signal (INTCBnT) is received and transfer data is written to the CBnTX register.

(7) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.

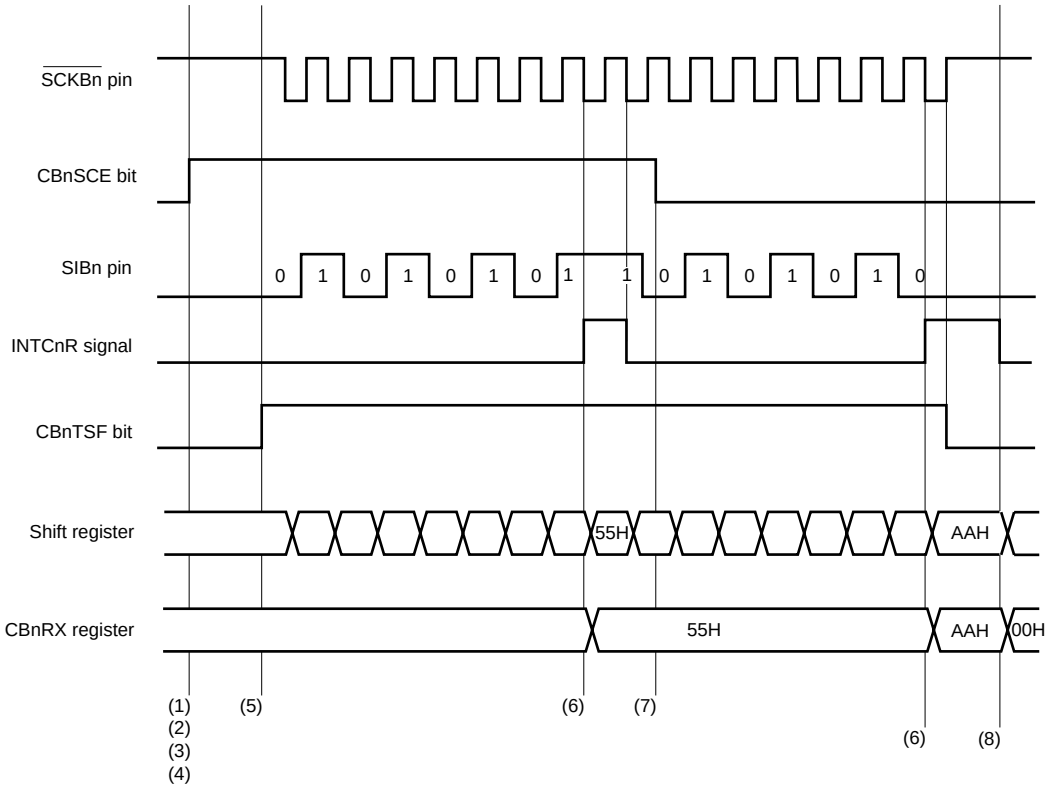
(8) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

In transmission mode or transmission/reception mode, the communication is not started by reading the CBnRX register.

14.6.4 Continuous mode (master mode, reception mode)

This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 2 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).

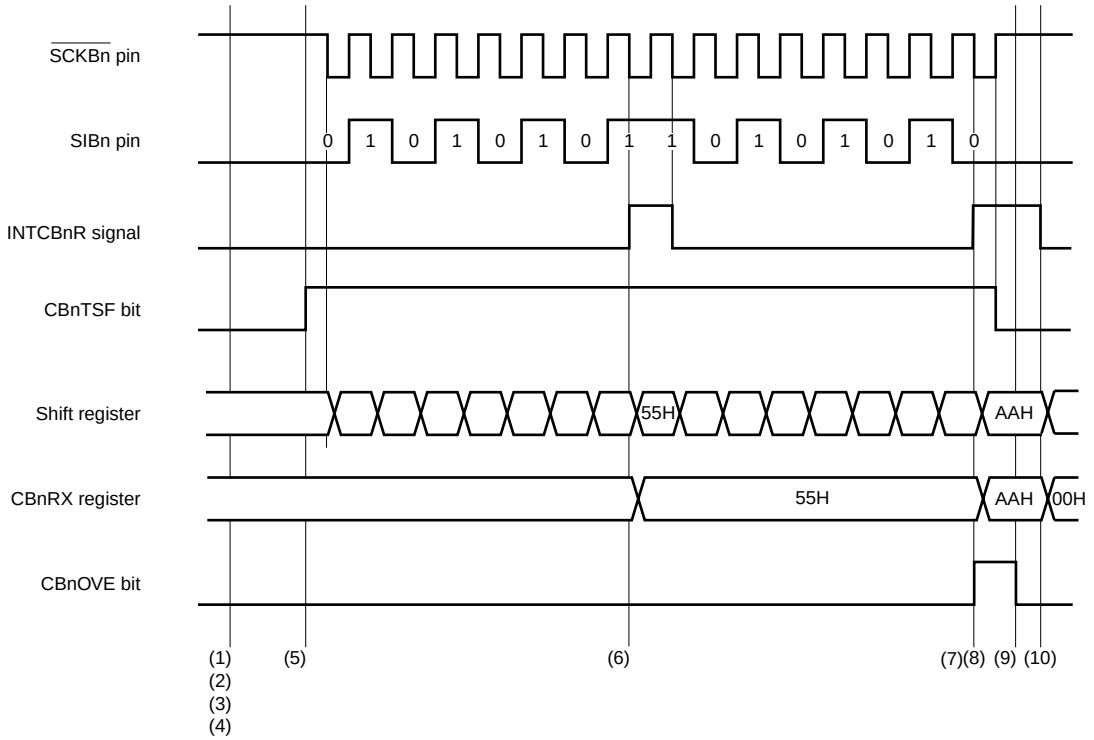


(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE bit to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(7) Set the CBnCTL0.CBnSCE bit = 0 while the last data being received to set the final receive data status.
(8) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

14.6.5 Continuous reception mode (error)

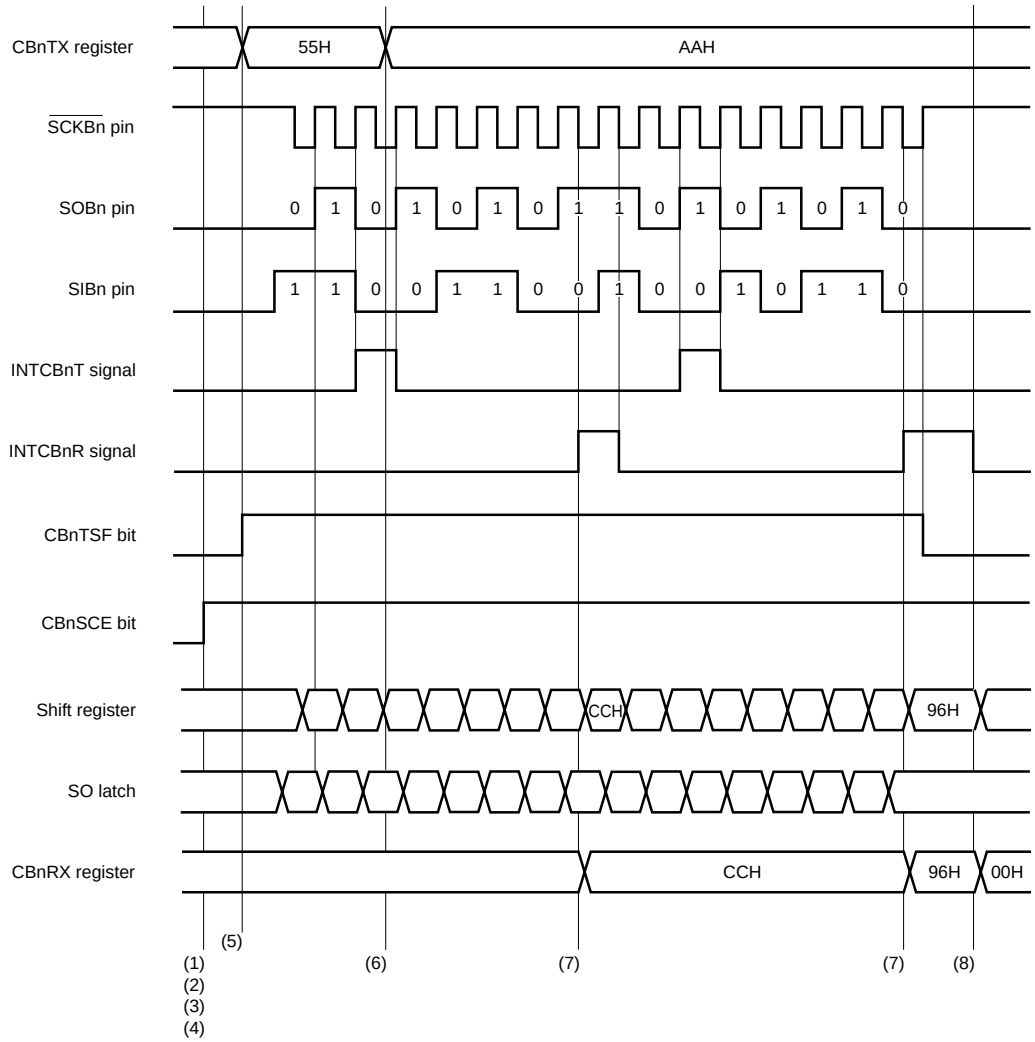
This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 2 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).



(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE bit to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit = 1 to enable CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) If the data could not be read before the end of the next transfer, the CBnSTR.CBnOVE flag is set to 1 upon the end of reception and the INTCBnR signal is output.
(8) Overrun error processing is performed after checking that the CBnOVE bit = 1 in the INTCBnR interrupt servicing.
(9) Clear CBnOVE bit to 0.
(10) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation CSIBn (end of reception).

14.6.6 Continuous mode (slave mode, transmission/reception mode)

This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 2 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CSnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).

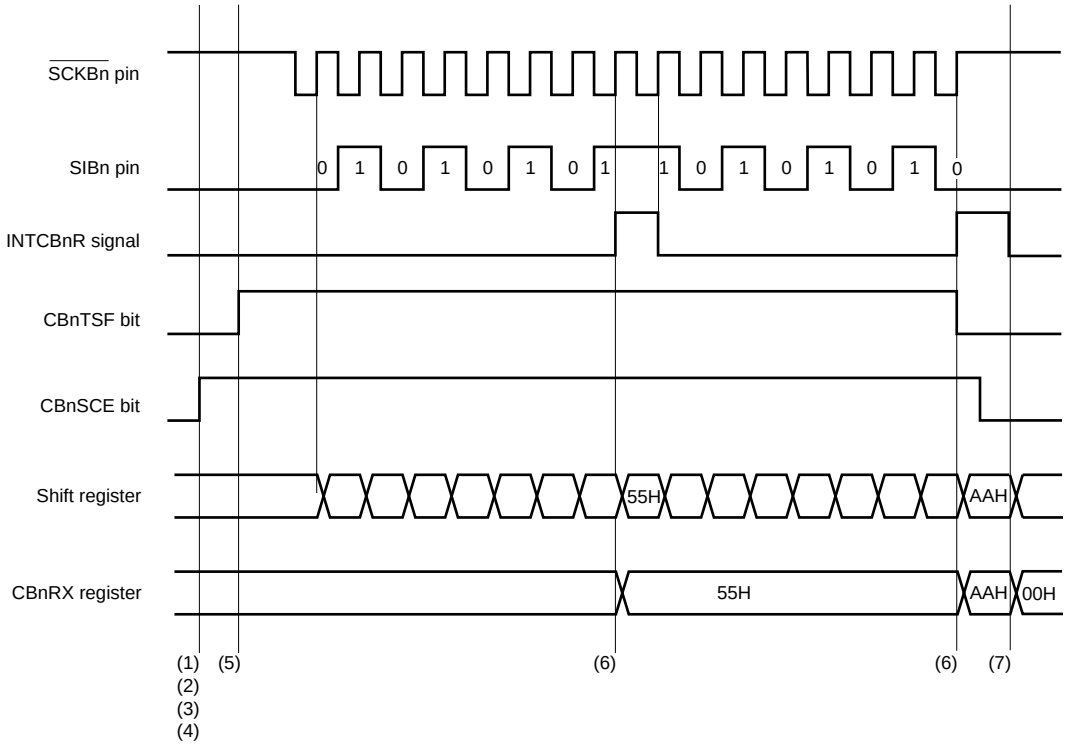


(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnTXE, CBnRXE and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
(4) Set the CBnPWR bit to 1 to enable supply of the CSIBn operation.
(5) Write the transfer data to the CBnTX register.
(6) The transmission enable interrupt request signal (INTCBnT) is received and the transfer data is written to the CBnTX register.
(7) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register.
(8) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

14.6.7 Continuous mode (slave mode, reception mode)

This section shows the case using MSB first (CBnCTL0.CBnDIR bit = 0) and communication type 1 (see **14.3 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0).



(1) Clear the CBnCTL0.CBnPWR bit to 0.

(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.

(3) Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.

(4) Set the CBnPWR bit = 1 to enable CSIBn operation.

(5) Perform a dummy read of the CBnRX register (reception start trigger).

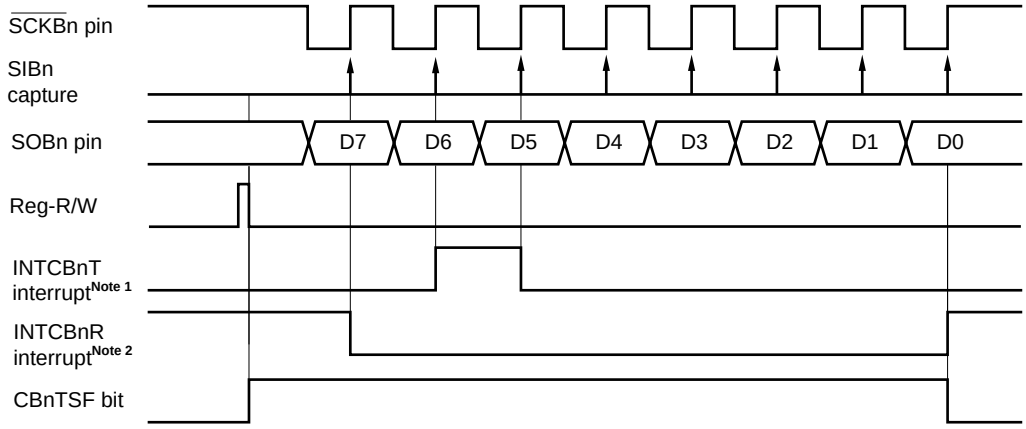
(6) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register. When reading the last data, clear the CBnCTL0.CBnSCE bit to 0 before reading the CBnRX register.

(7) Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

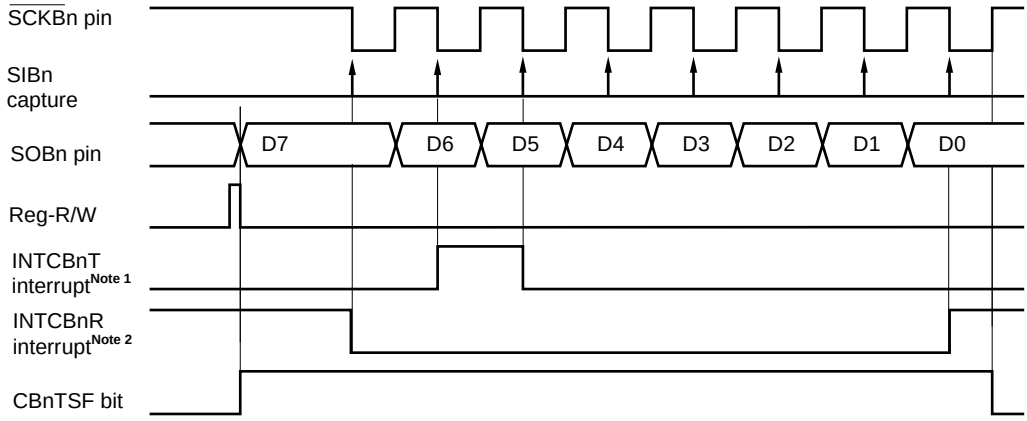
To continue transfer, repeat steps (5) and (6) before (7).

14.6.8 Clock timing

(1) Communication type 1 (CBnCKP = 0, CBnDAP = 0)



(2) Communication type 2 (CBnCKP = 0, CBnDAP = 1)



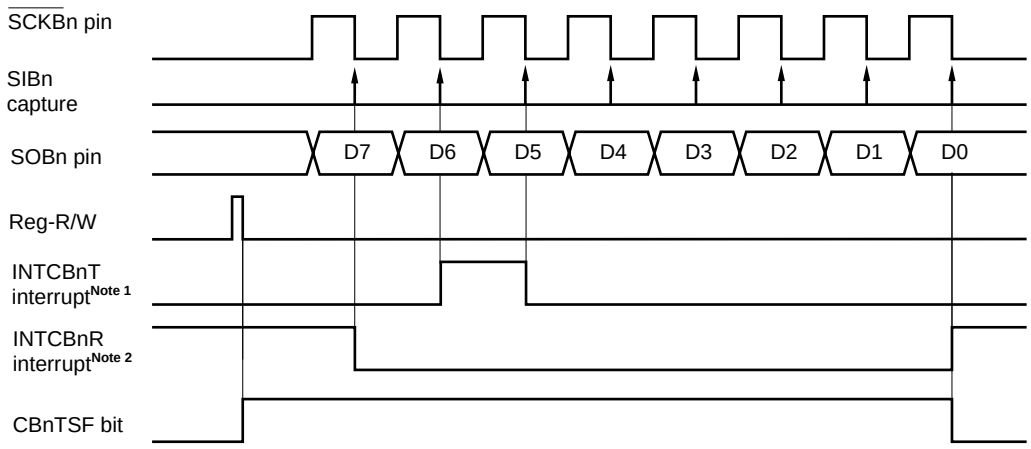
Notes 1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.

2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

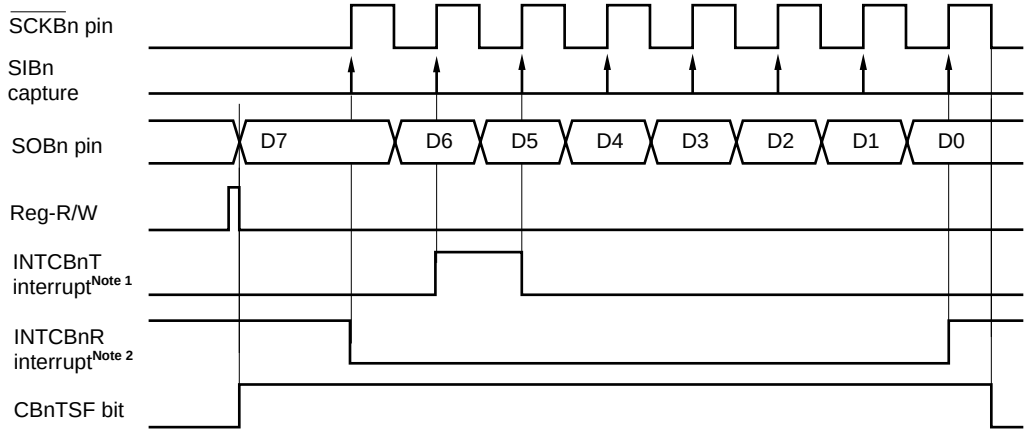
★ **Caution In communication type 2, the CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock after generation of the INTCBnR interrupt request signal.**

(2/2)

(3) Communication type 3 (CBnCKP = 1, CBnDAP = 0)



(4) Communication type 4 (CBnCKP = 1, CBnDAP = 1)



Notes 1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.

2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

★ **Caution In communication type 4, the CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock after generation of the INTCBnR interrupt request signal.**

14.7 Output Pins

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

CBnCKS2	CBnCKS1	CBnCKS0	CBnCKP	$\overline{\text{SCKBn}}$ Pin Output
1	1	1	×	High impedance
Other than above			0	Fixed to high level
			1	Fixed to low level

Remarks 1. The output level of the $\overline{\text{SCKBn}}$ pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

2. n = 0 to 2

3. ×: don't care

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX register value (MSB)
		1	CBnTX register value (LSB)

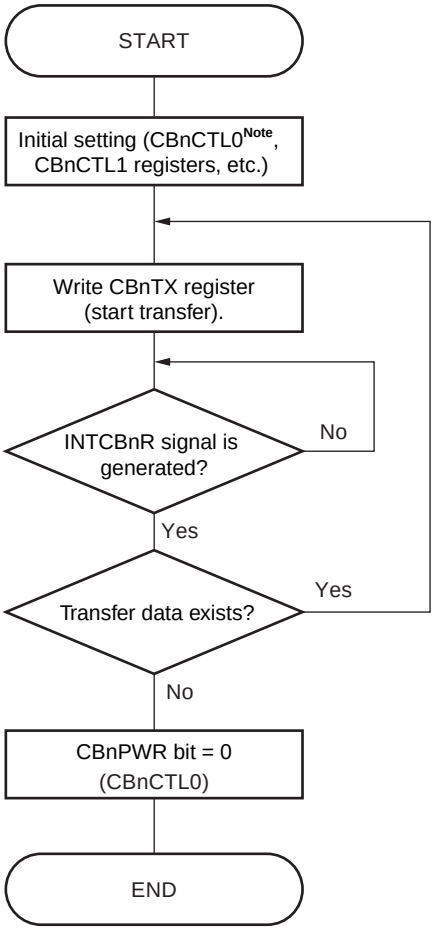
Remarks 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. n = 0 to 2

3. ×: don't care

14.8 Operation Flow

(1) Single transmission

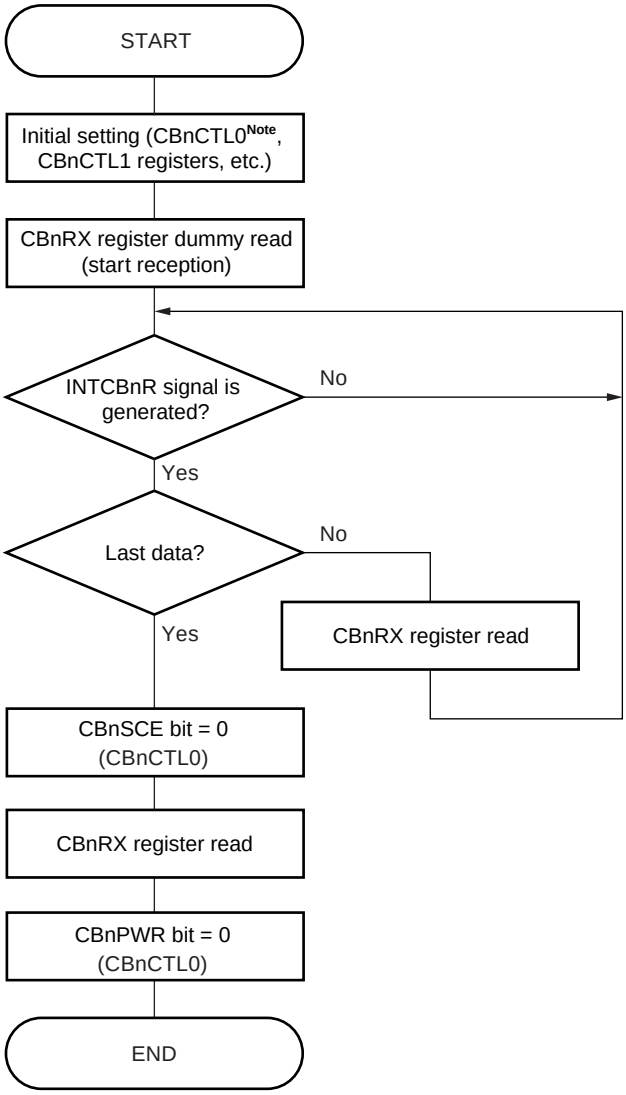


Note Set the CBnSCE bit to 1 in the initial setting.

Caution In the slave mode, data cannot be correctly transmitted if the next transfer clock is input earlier than the CBnTX register is written.

Remark n = 0 to 2

(2) Single reception

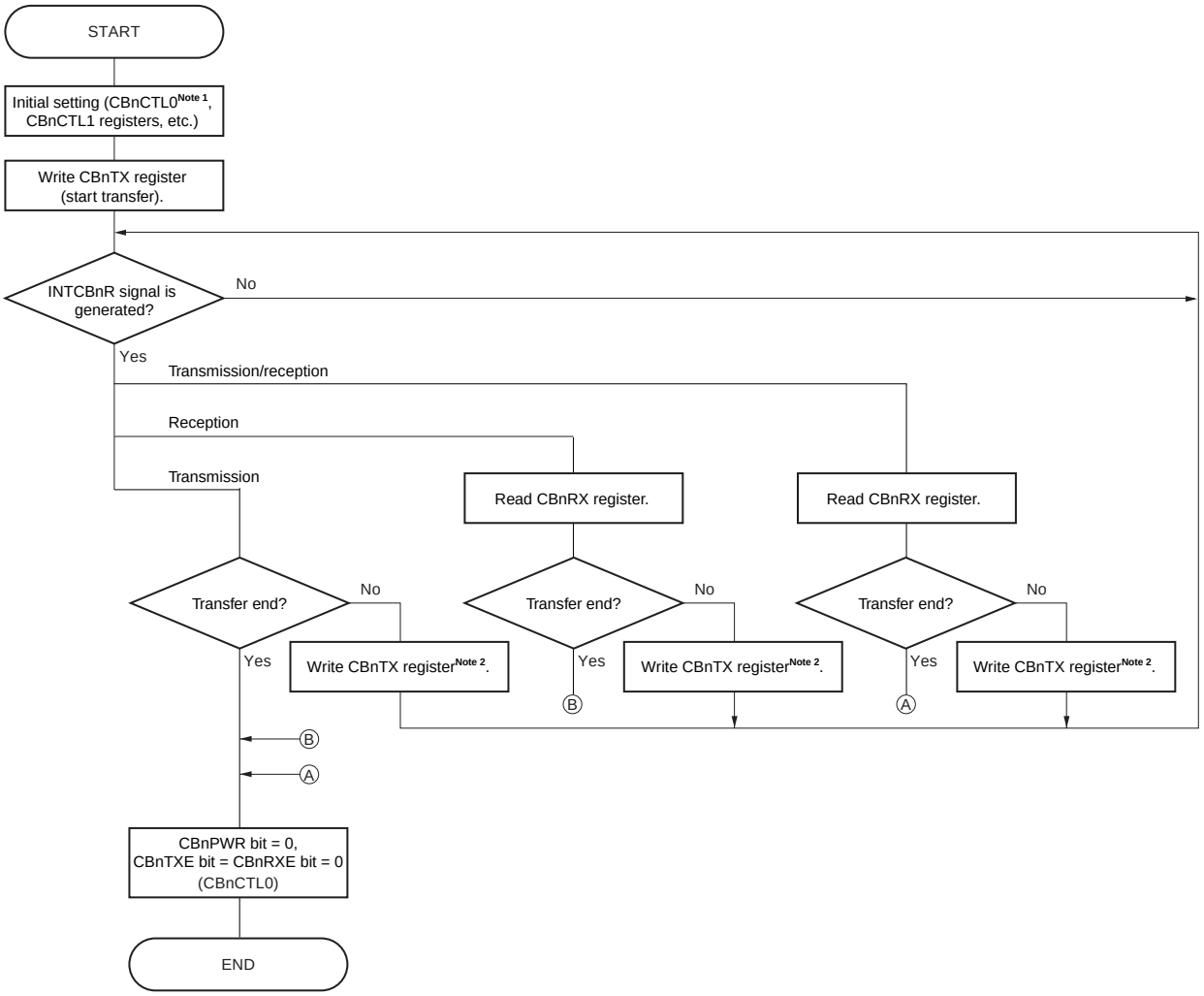


Note Set the CBnSCE bit to 1 in the initial setting.

Caution In the single mode, data cannot be correctly received if the next transfer clock is input earlier than the CBnRX register is read.

Remark n = 0 to 2

(3) Single transmission/reception



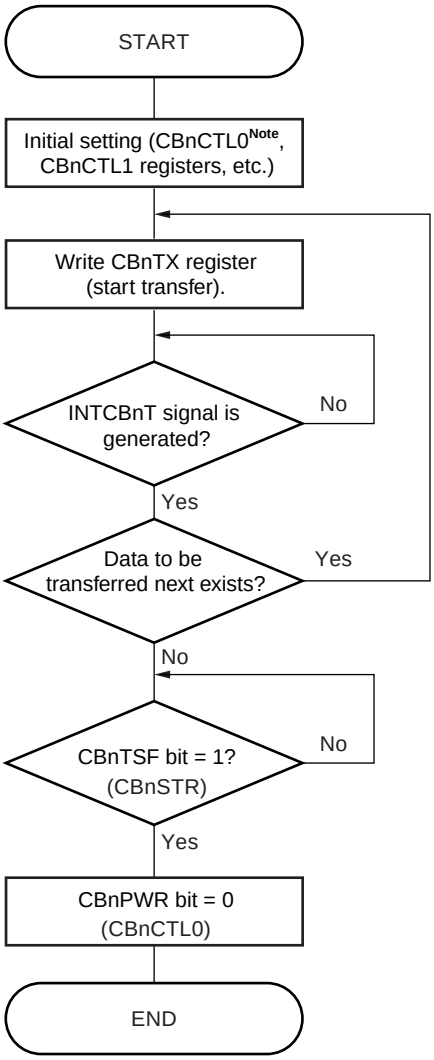
Notes 1. Set the CBnSCE bit to 1 in the initial setting.

2. If the next transfer is reception only, dummy data is written to the CBnTX register.

Caution Even in the single mode, the CBnSTR.CBnOVE flag is set to 1. If only transmission is used in the transmission/reception mode, therefore, checking the CBnOVE flag is not required.

Remark n = 0 to 2

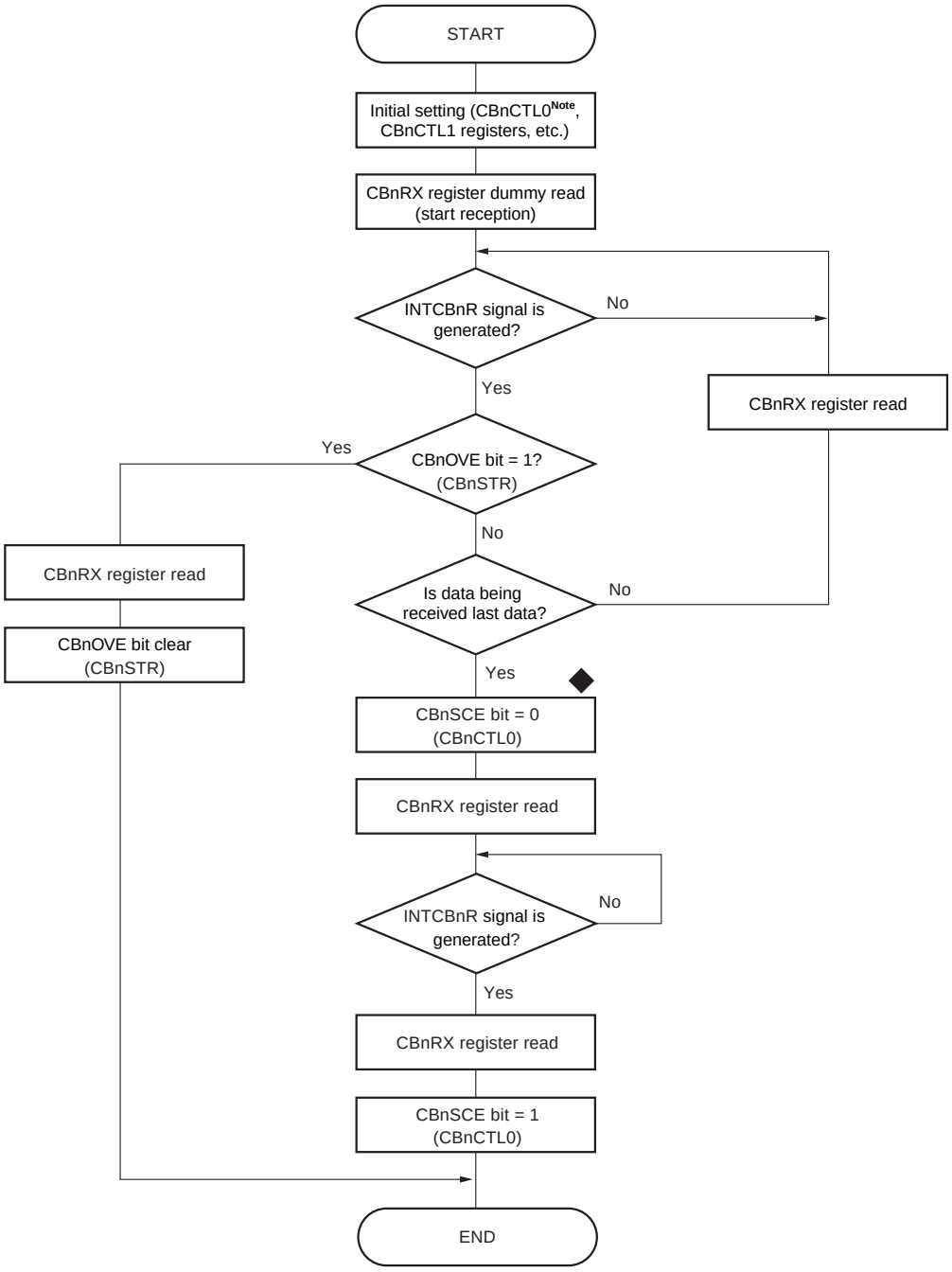
(4) Continuous transmission



Note Set the CBnSCE bit to 1 in the initial setting.

Remark n = 0 to 2

(5) Continuous reception

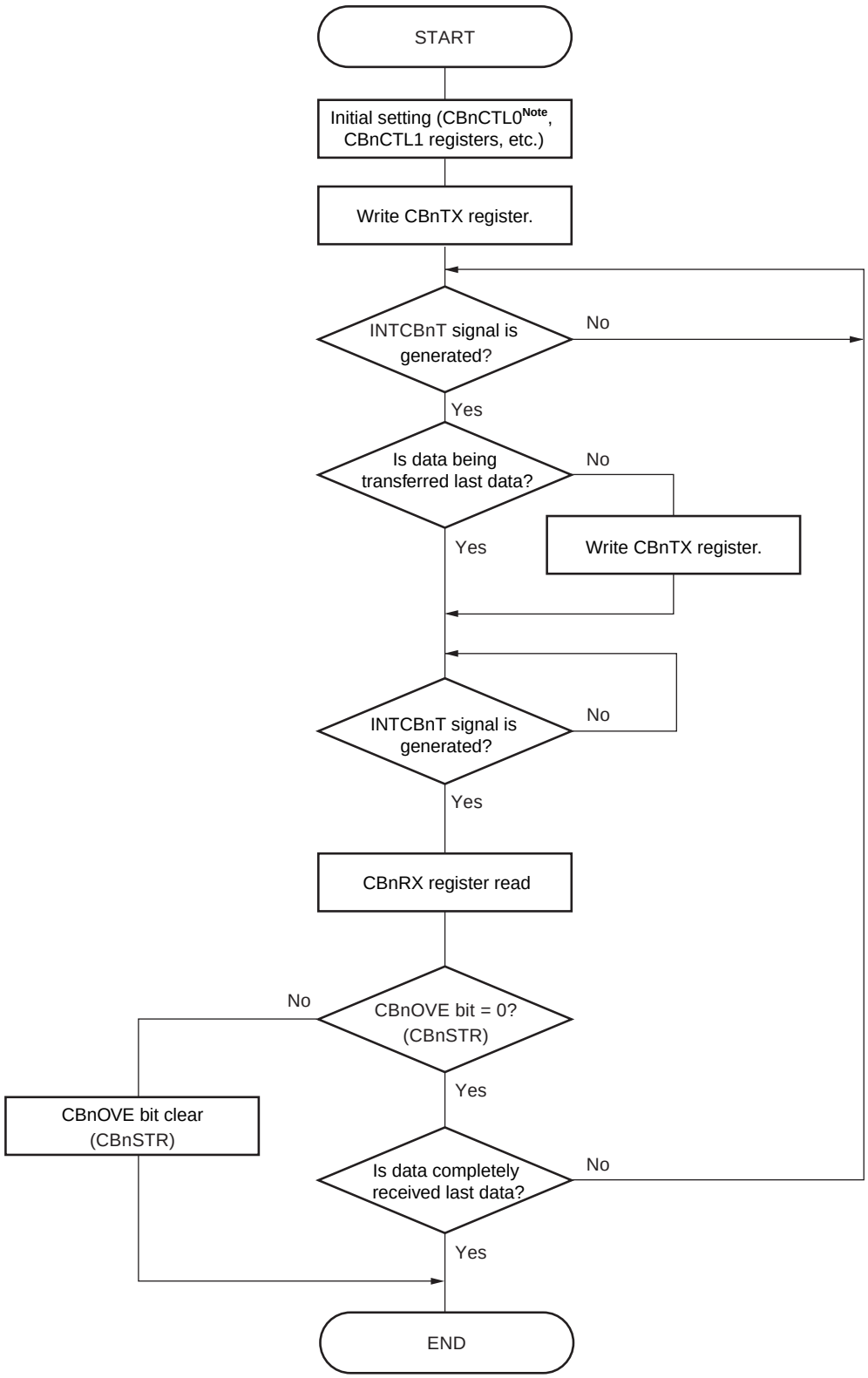


Note Set the CBnSCE bit to 1 in the initial setting

Caution In the master mode, the clock is output without limit when dummy data is read from the CBnRX register. To stop the clock, execute the flow marked ◆ in the above flowchart.
In the slave mode, malfunction due to noise during communication can be prevented by executing the flow marked ◆ in the above flowchart.
Before resuming communication, set the CBnCTL0.CBnSCE bit to 1, and read dummy data from the CBnRX register.

Remark n = 0 to 2

(6) Continuous transmission/reception



Note Set the CBnSCE bit to 1 in the initial setting.

Remark n = 0 to 2

14.9 Prescaler 3

Prescaler 3 has the following function.

- Generation of count clock for watch timer and CSIB0 (source clock: main oscillation clock)

14.9.1 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.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Prescaler output
0	Disabled
1	Enabled

BGCS01	BGCS00	Selection of count clock (f_{BRG})		
			f_X = 4 MHz	f_X = 5 MHz
0	0	f_X	250 ns	200 ns
0	1	$f_X/2$	500 ns	400 ns
1	0	$f_X/4$	1 μs	800 ns
1	1	$f_X/8$	2 μs	1.6 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.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

14.9.2 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 BGCSn (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 = 4.00 MHz, m = 0 (BGCS01 bit = BGCS00 bit = 0), and N = 3DH

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 BGCSn bits (00B to 11B)

n = 00, 01

14.10 Cautions

(1) When transferring transmit data and receive data using DMA transfer, error processing cannot be performed even if an overrun error occurs during serial transfer. Check that the no overrun error has occurred by reading the CBnSTR.CBnOVE bit after DMA transfer has been completed.

(2) In regards to registers that are forbidden from being rewritten during operations (CBnCTL0.CBnPWR bit is 1), if rewriting has been carried out by mistake during operations, set the CBnCTL0.CBnPWR bit to 0 once, then initialize CSIBn.

Registers to which rewriting during operation are prohibited are shown below.

- CBnCTL0 register: CBnTXE, CBnRXE, CBnDIR, CBnTMS bits
- CBnCTL1 register: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 bits
- CBnCTL2 register: CBnCL3 to CBnCL0 bits

★ (3) In communication type 2 or 4 (CBnCTL1.CBnDAP bit = 1), the CBnSTR.CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock after occurrence of a reception complete interrupt (INTCBnR).

In the single transfer mode, writing the next transmit data is ignored during communication (CBnTSF bit = 1), and the next communication is not started. Also if reception-only communication (CBnCTL0.CBnTXE bit = 0, CBnCTL0.CBnRXE bit = 1) is set, the next communication is not started even if the receive data is read during communication (CBnTSF bit = 1).

Therefore, when using the single transfer mode with communication type 2 or 4 (CBnDAP bit = 1), pay particular attention to the following.

- To start the next transmission, confirm that CBnTSF bit = 0 and then write the transmit data to the CBnTX register.
- To perform the next reception continuously when reception-only communication (CBnTXE bit = 0, CBnRXE bit = 1) is set, confirm that CBnTSF bit = 0 and then read the CBnRX register.

Or, use the continuous transfer mode instead of the single transfer mode. Use of the continuous transfer mode is recommended especially for using DMA.

Remark n = 0 to 2

CHAPTER 15 CAN CONTROLLER

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

15.1 Overview

This product features an on-chip n-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.

Product Name (Part Number)		Number of Channels
V850ES/FE2		1
V850ES/FF2		1
V850ES/FG2		2
V850ES/FJ2	μPD70F3237	2
	μPD70F3238 μPD70F3239	4

15.1.1 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 per each channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

15.1.2 Overview of Functions

Table 15-1 presents an overview of the CAN controller functions.

Table 15-1. Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	• 32 message buffers per each channel[Note] • 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)

Note 4 channels max.

15.1.3 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.

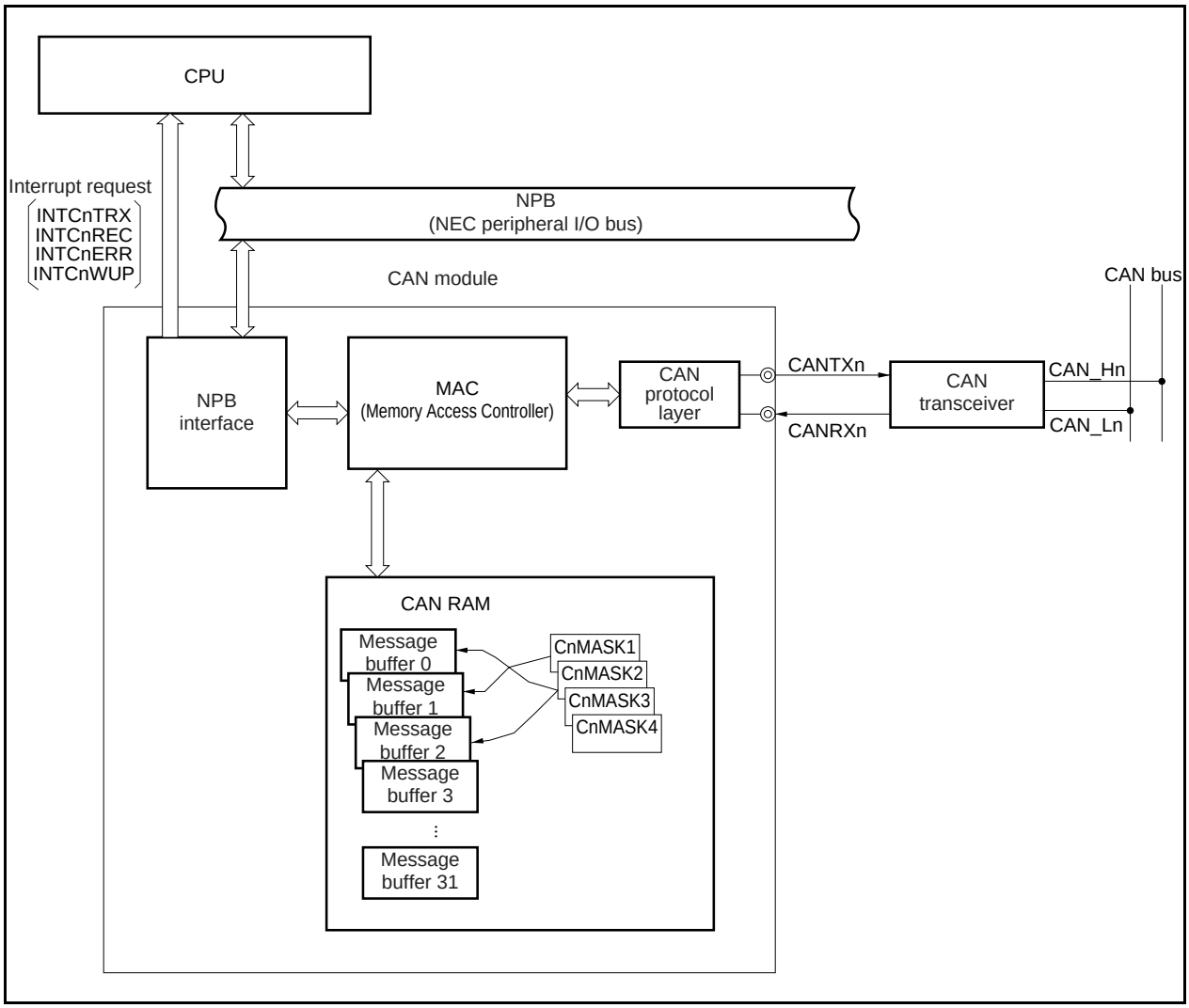
(3) CAN protocol layer

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 15-1. Block Diagram of CAN Module



15.2 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 15-2. Composition of Layers

Higher

Lower

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
Physical layer		Prescription of signal level and bit description

Note CAN controller specification

15.2.1 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.

15.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 15-2. 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.

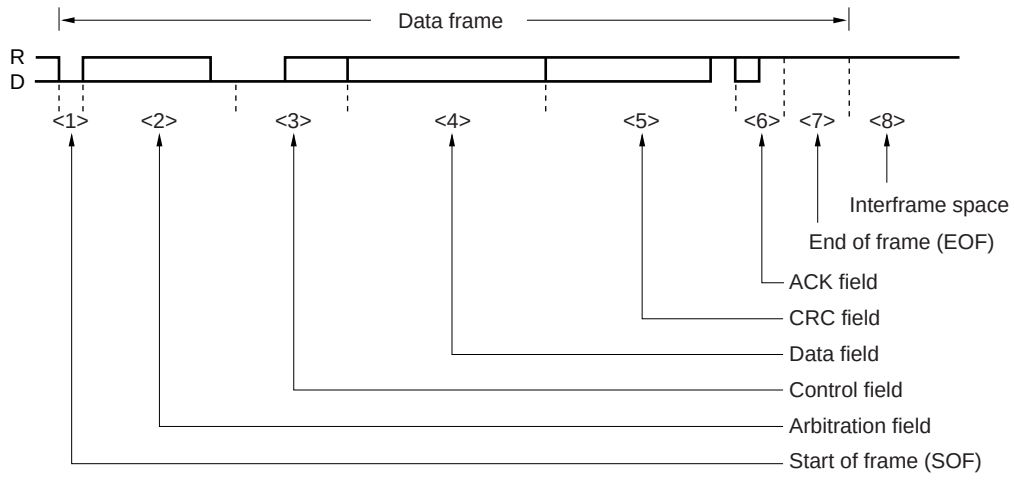
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

15.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 15-3. Data Frame

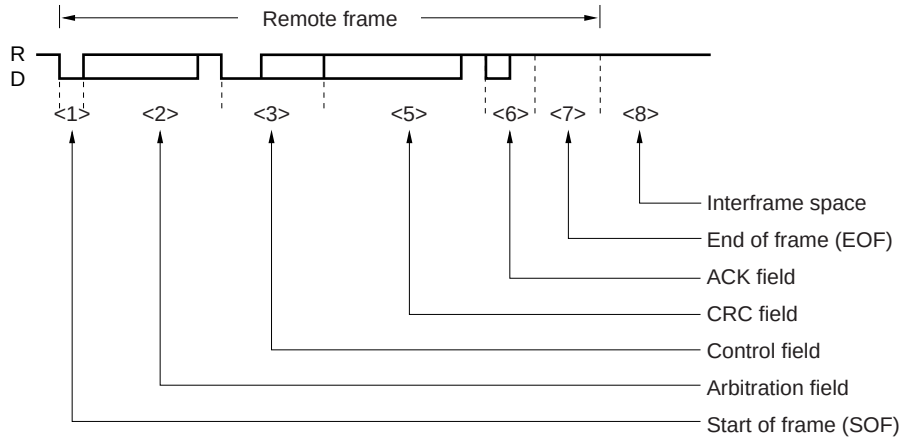


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 15-4. Remote Frame



Remarks 1. The data field is not transferred even if the control field's data length code is not "0000B".

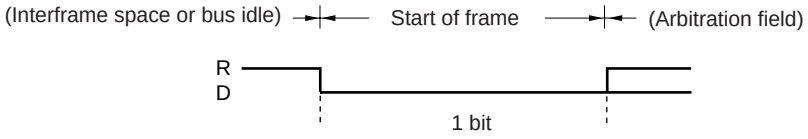
2. D: Dominant = 0
R: Recessive = 1

(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 15-5. Start of Frame (SOF)



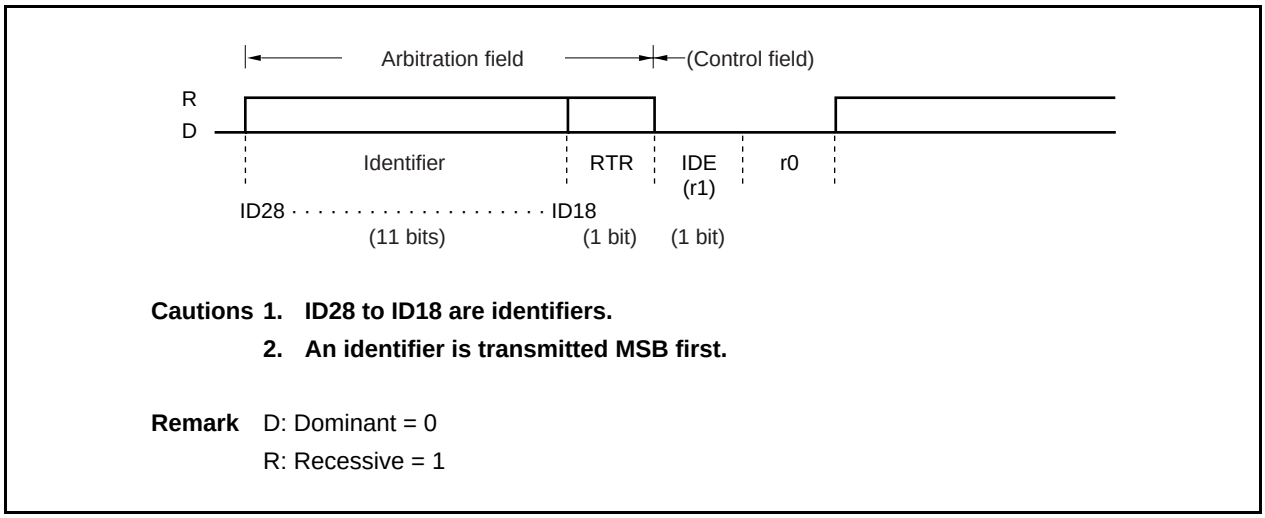
Remark D: Dominant = 0
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

<2> Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 15-6. Arbitration Field (in Standard Format Mode)



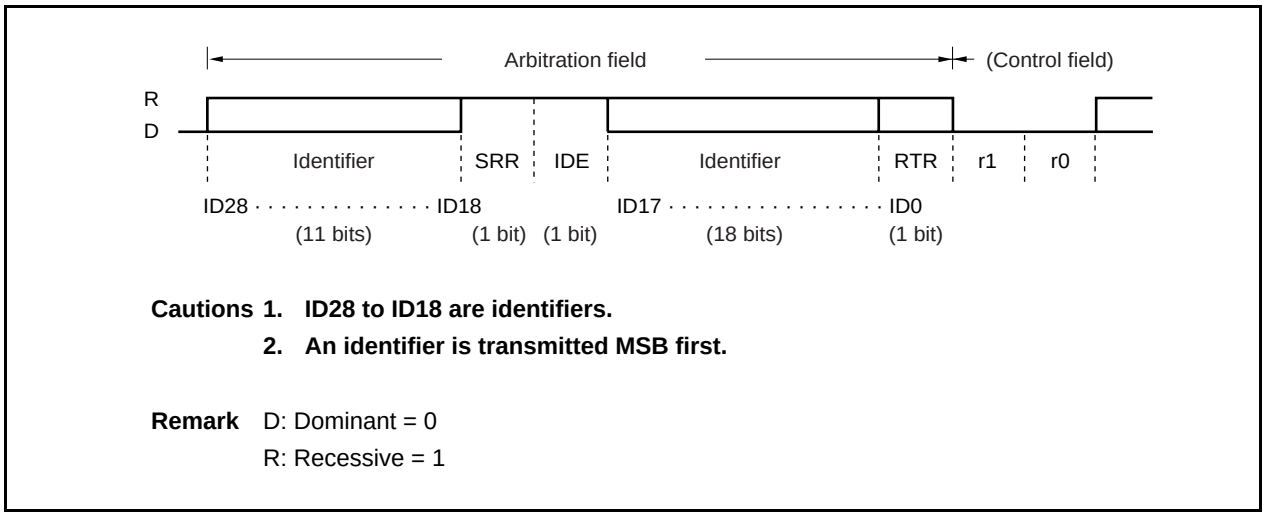
Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Figure 15-7. 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 15-3. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

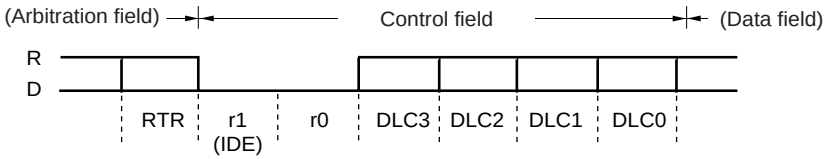
Table 15-4. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

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 15-8. 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 15-5. Data Length Setting

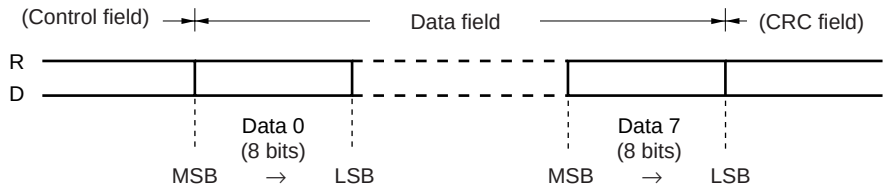
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

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 15-9. Data Field

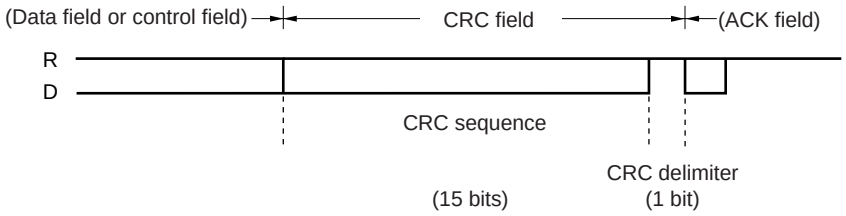


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 15-10. CRC Field



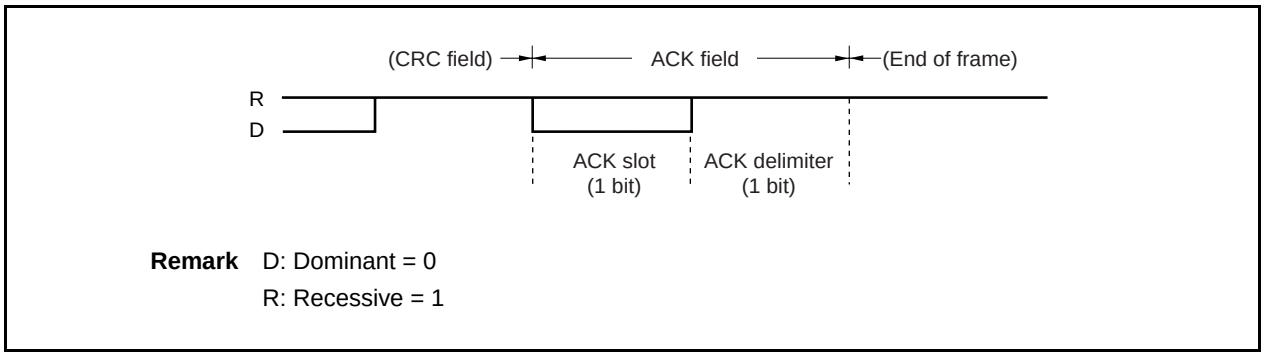
Remark D: Dominant = 0
R: Recessive = 1

- 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 15-11. ACK Field



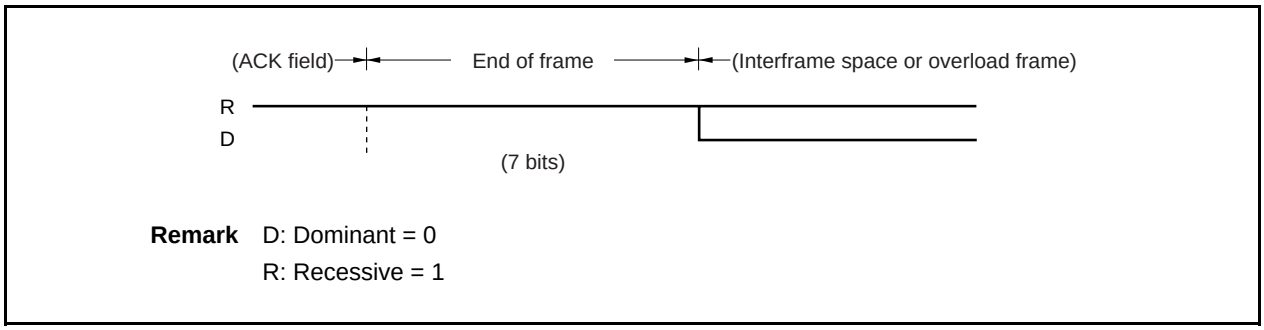
Remark D: Dominant = 0
R: Recessive = 1

- 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 15-12. End of Frame (EOF)



Remark D: Dominant = 0
R: Recessive = 1

<8> Interframe space

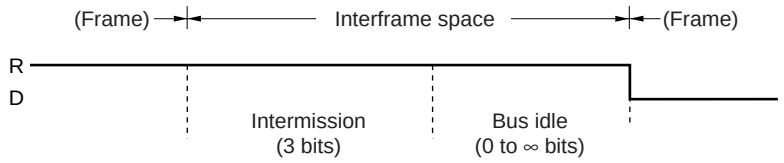
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- 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 15-13. Interframe Space (Error Active Node)



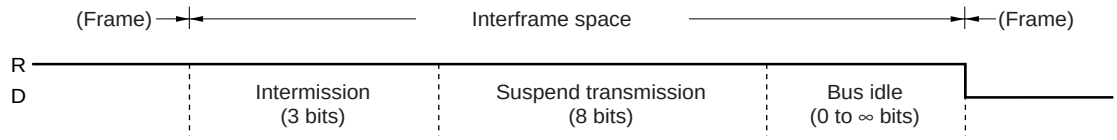
Remarks 1. Bus idle: State in which the bus is not used by any node.

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 15-14. 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

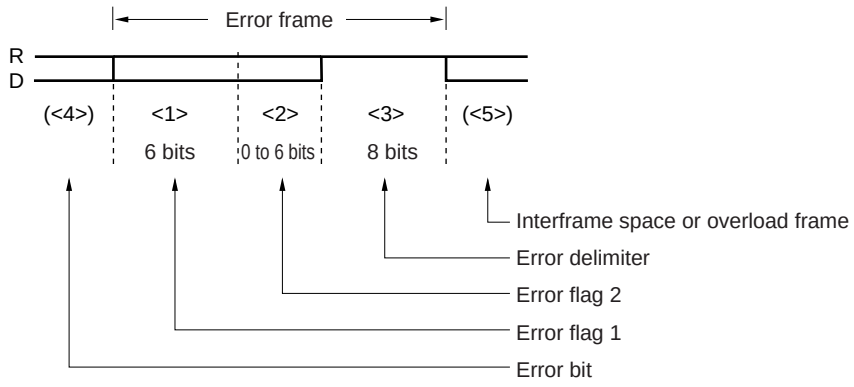
Table 15-6. Operation in Error Status

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.

15.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 15-15. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 15-7. 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 issue 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.

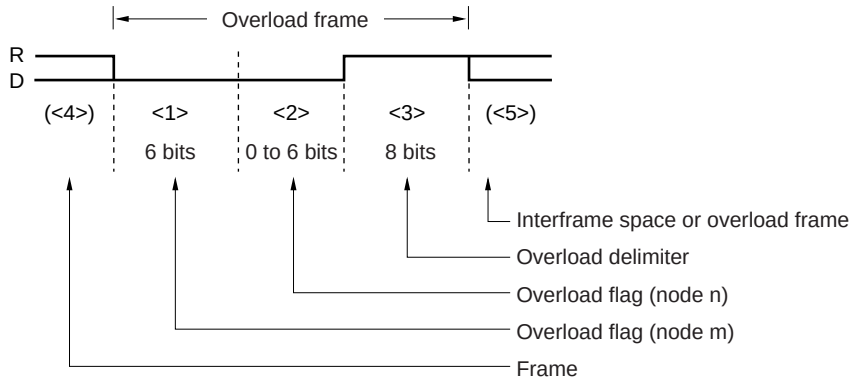
15.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation [Note]
- 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 The CAN is internally fast enough to process all received frames not generating overload frames.

Figure 15-16. Overload Frame



Remark D: Dominant = 0
R: Recessive = 1
Node n ≠ node m

Table 15-8. 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.

15.3 Functions

15.3.1 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 15-9. 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.

15.3.2 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 15-10. 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.

15.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

15.3.4 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.

15.3.5 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).

15.3.6 Error control function

(1) Error types

Table 15-11. 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 15-12. 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 15-13.

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 startup (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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

Table 15-13. 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 Value of the transmission error counter (TEC) is not meaning when BOFF bit is 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, the counter is not incremented and the bus-off state is assumed.

Remark n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

(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 15-14. 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 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 15-17). 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.

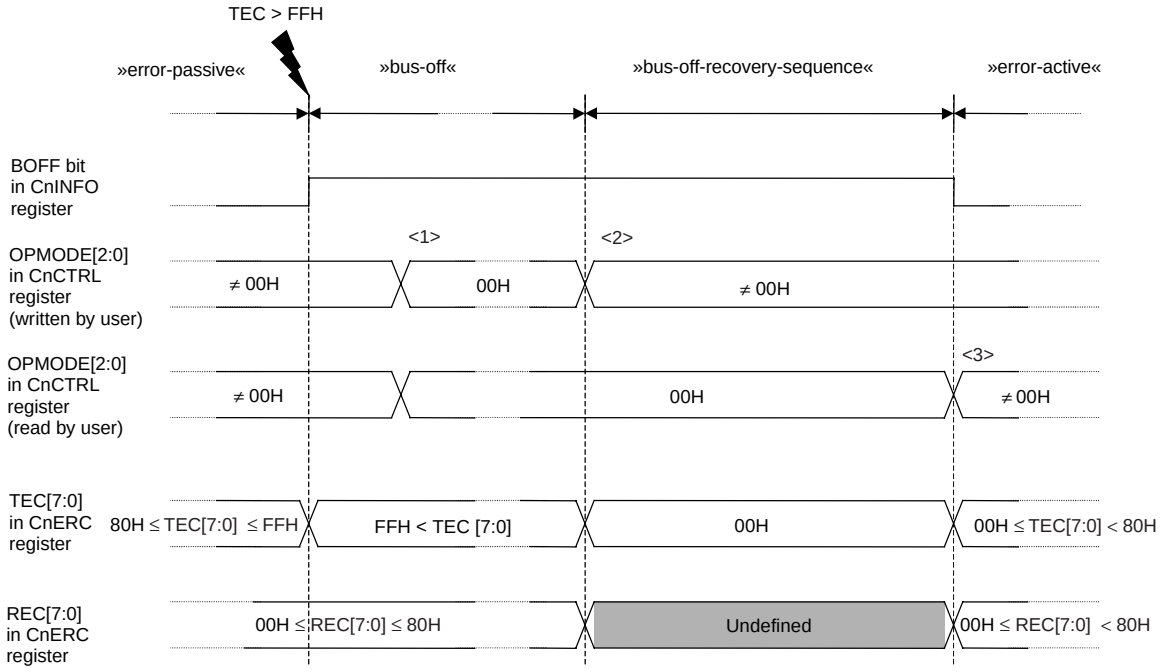
Next, the module requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in Figure 15-17). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the 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 15-17), 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, REC[6:0] 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 enter the initialization mode. In this case, release the module from the CAN sleep or stop mode, and then make a request to place the module in the initialization mode.

Remark n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

Figure 15-17. Recovery from Bus-off State Through Normal Recovery Sequence



Remark n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

(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 **15.3.6 (5) (a) Recovery from bus-off state through normal recovery sequence**.

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 15-53.

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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

(6) 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

15.3.7 Baud rate control function

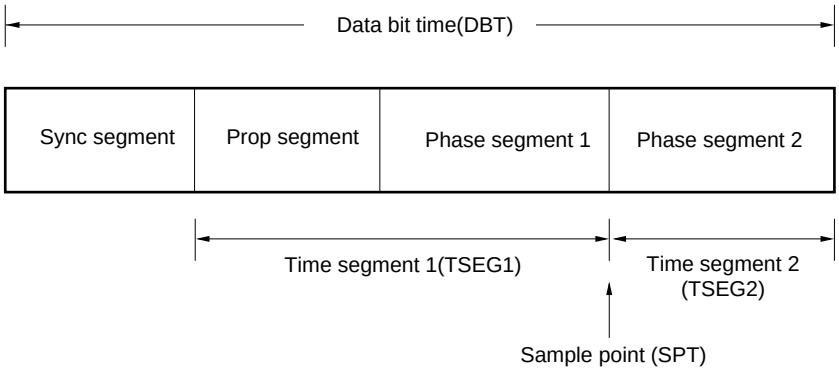
(1) Prescaler

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 **15.6 (12) CAN module bit rate prescaler register (CnBRP)**).

(2) Data bit time (8 to 25 time quanta)

One data bit time is defined as figure 14-8. The CAN controller sets time segment 1, time segment 2, and resynchronization Jump Width (SJW) as the data bit time, as shown in Figure 15-18. 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 15-18. Segment Setting

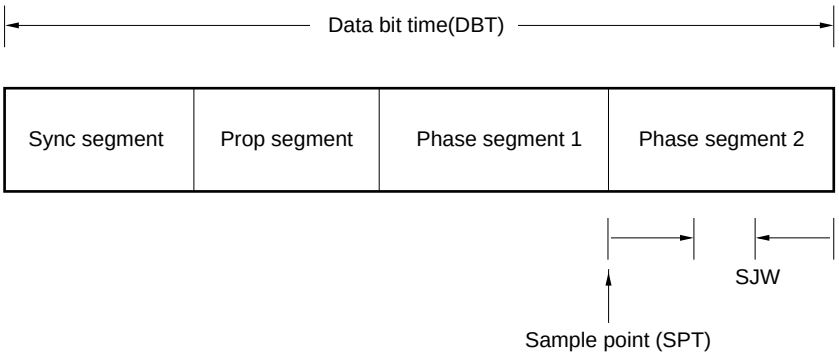


Segment name	Settable range	Notes on setting to conform to CAN specification
Time segment 1 (TSEG1)	2TQ to 16TQ	—
Time segment 2 (TSEG2)	1TQ to 8TQ	IPT[Note] of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length equal to phase segment 1 must be set here. This 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	The length of time segment 1 minus 1TQ or 4 TQ, 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 15-19.

Figure 15-19. 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 1TQ to length of segment 1 or 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

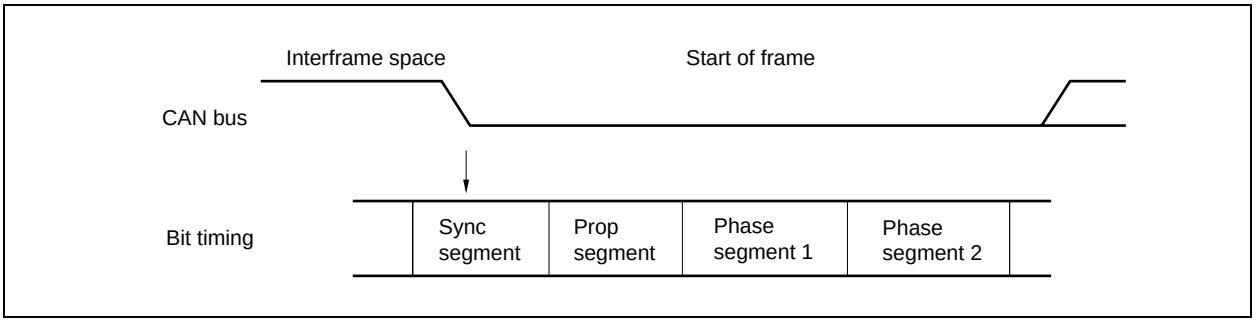
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(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 15-20. Adjusting Synchronization of Data Bit

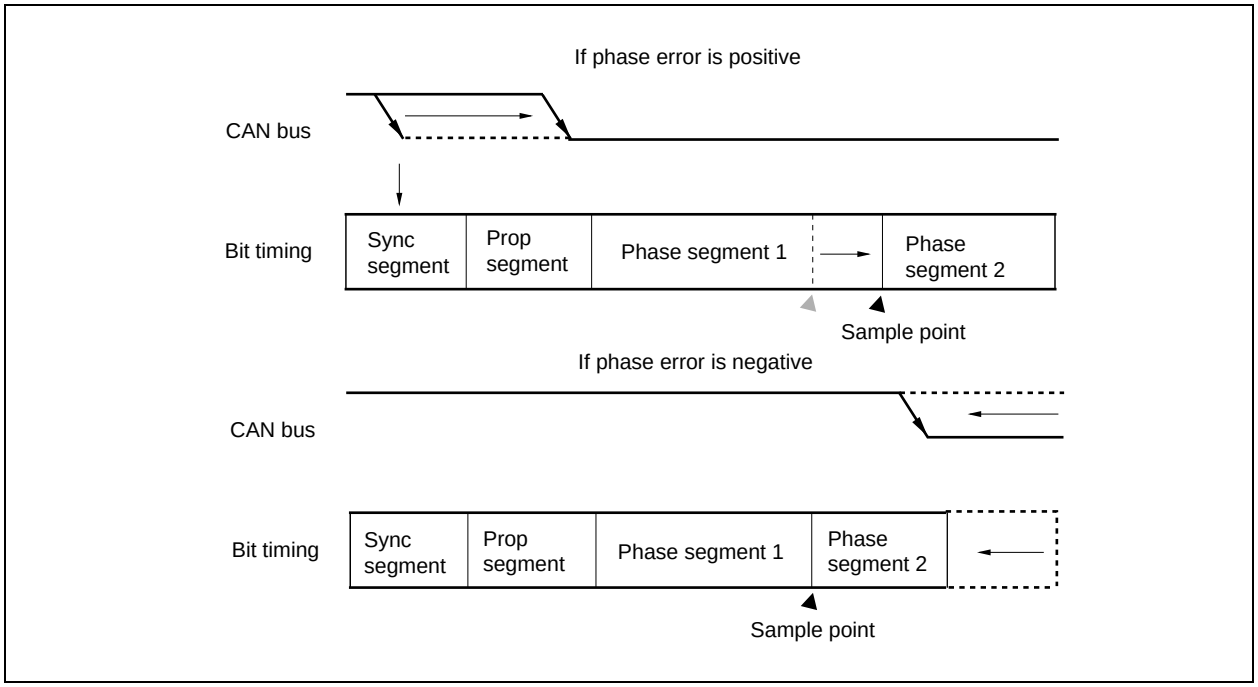


(b) Resynchronization

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.

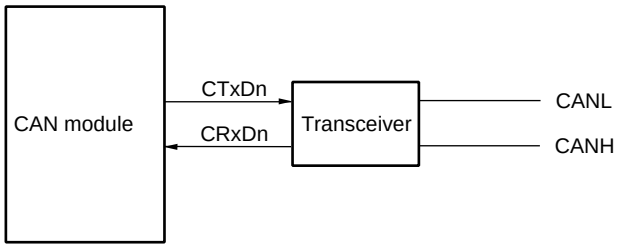
Figure 15-21. Resynchronization



15.4 Connection with Target System

The CAN module has to be connected to the CAN bus using an external transceiver.

Figure 15-22. Connection to CAN Bus



Remark n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

15.5 Internal Registers of CAN controller

15.5.1 CAN controller configuration

Table 15-15. List of CAN Controller Registers

(1/2)

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)

Remark 1. n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

2. CAN global registers are identified by CnGM<register function>.
CAN module registers are identified by Cn<register function>.
Message buffer registers are identified by CnM<register function>.

(2/2)

Item	Register Name
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 1. n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

2. CAN global registers are identified by CnGM<register function>.
CAN module registers are identified by Cn<register function>.
Message buffer registers are identified by CnM<register function>.

15.5.2 Register access type

The peripheral I/O register for the CAN controller is assigned to 03FEC000H - 03FED800H. For details, refer to **3.4.8 Programmable peripheral I/O register**.

Table 15-16. Register Access Type

(1/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC000H	CAN0 global control register	C0GMCTRL	R/W	–	–	√	0000H
03FEC002H	CAN0 global clock select register	C0GMCS	R/W	–	√	–	0FH
03FEC006H	CAN0 global block transmission control register	C0GMABT	R/W	–	–	√	0000H
03FEC008H	CAN0 global 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

(2/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC100H	CAN0 message data byte 01 register 00	C0MDATA0100	R/W			√	Undefined
03FEC100H	CAN0 message data byte 0 register 00	C0MDATA000			√		Undefined
03FEC101H	CAN0 message data byte 1 register 00	C0MDATA100			√		Undefined
03FEC102H	CAN0 message data byte 23 register 00	C0MDATA2300				√	Undefined
03FEC102H	CAN0 message data byte 2 register 00	C0MDATA200			√		Undefined
03FEC103H	CAN0 message data byte 3 register 00	C0MDATA300			√		Undefined
03FEC104H	CAN0 message data byte 45 register 00	C0MDATA4500				√	Undefined
03FEC104H	CAN0 message data byte 4 register 00	C0MDATA400			√		Undefined
03FEC105H	CAN0 message data byte 5 register 00	C0MDATA500			√		Undefined
03FEC106H	CAN0 message data byte 67 register 00	C0MDATA6700				√	Undefined
03FEC106H	CAN0 message data byte 6 register 00	C0MDATA600			√		Undefined
03FEC107H	CAN0 message data byte 7 register 00	C0MDATA700			√		Undefined
03FEC108H	CAN0 message data length code register 00	C0MDLC00			√		0000xxxxB
03FEC109H	CAN0 message configuration register 00	C0MCONF00			√		Undefined
03FEC10AH	CAN0 message ID register 00	C0MIDL00				√	Undefined
03FEC10CH		C0MIDH00				√	Undefined
03FEC10EH	CAN0 message control register 00	C0MCTRL00				√	00x00000 000xx000B
03FEC120H	CAN0 message data byte 01 register 01	C0MDATA0101				√	Undefined
03FEC120H	CAN0 message data byte 0 register 01	C0MDATA001			√		Undefined
03FEC121H	CAN0 message data byte 1 register 01	C0MDATA101			√		Undefined
03FEC122H	CAN0 message data byte 23 register 01	C0MDATA2301				√	Undefined
03FEC122H	CAN0 message data byte 2 register 01	C0MDATA201			√		Undefined
03FEC123H	CAN0 message data byte 3 register 01	C0MDATA301			√		Undefined
03FEC124H	CAN0 message data byte 45 register 01	C0MDATA4501				√	Undefined
03FEC124H	CAN0 message data byte 4 register 01	C0MDATA401			√		Undefined
03FEC125H	CAN0 message data byte 5 register 01	C0MDATA501			√		Undefined
03FEC126H	CAN0 message data byte 67 register 01	C0MDATA6701				√	Undefined
03FEC126H	CAN0 message data byte 6 register 01	C0MDATA601			√		Undefined
03FEC127H	CAN0 message data byte 7 register 01	C0MDATA701			√		Undefined
03FEC128H	CAN0 message data length code register 01	C0MDLC01			√		0000xxxxB
03FEC129H	CAN0 message configuration register 01	C0MCONF01			√		Undefined
03FEC12AH	CAN0 message ID register 01	C0MIDL01				√	Undefined
03FEC12CH		C0MIDH01				√	Undefined
03FEC12EH	CAN0 message control register 01	C0MCTRL01				√	00x00000 000xx000B

(3/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC140H	CAN0 message data byte 01 register 02	C0MDATA0102	R/W			√	Undefined
03FEC140H	CAN0 message data byte 0 register 02	C0MDATA002			√		Undefined
03FEC141H	CAN0 message data byte 1 register 02	C0MDATA102			√		Undefined
03FEC142H	CAN0 message data byte 23 register 02	C0MDATA2302				√	Undefined
03FEC142H	CAN0 message data byte 2 register 02	C0MDATA202			√		Undefined
03FEC143H	CAN0 message data byte 3 register 02	C0MDATA302			√		Undefined
03FEC144H	CAN0 message data byte 45 register 02	C0MDATA4502				√	Undefined
03FEC144H	CAN0 message data byte 4 register 02	C0MDATA402			√		Undefined
03FEC145H	CAN0 message data byte 5 register 02	C0MDATA502			√		Undefined
03FEC146H	CAN0 message data byte 67 register 02	C0MDATA6702				√	Undefined
03FEC146H	CAN0 message data byte 6 register 02	C0MDATA602			√		Undefined
03FEC147H	CAN0 message data byte 7 register 02	C0MDATA702			√		Undefined
03FEC148H	CAN0 message data length code register 02	C0MDLC02			√		0000xxxxB
03FEC149H	CAN0 message configuration register 02	C0MCONF02			√		Undefined
03FEC14AH	CAN0 message ID register 02	C0MIDL02				√	Undefined
03FEC14CH		C0MIDH02				√	Undefined
03FEC14EH	CAN0 message control register 02	C0MCTRL02				√	00x00000 000xx000B
03FEC160H	CAN0 message data byte 01 register 03	C0MDATA0103				√	Undefined
03FEC160H	CAN0 message data byte 0 register 03	C0MDATA003			√		Undefined
03FEC161H	CAN0 message data byte 1 register 03	C0MDATA103			√		Undefined
03FEC162H	CAN0 message data byte 23 register 03	C0MDATA2303				√	Undefined
03FEC162H	CAN0 message data byte 2 register 03	C0MDATA203			√		Undefined
03FEC163H	CAN0 message data byte 3 register 03	C0MDATA303			√		Undefined
03FEC164H	CAN0 message data byte 45 register 03	C0MDATA4503				√	Undefined
03FEC164H	CAN0 message data byte 4 register 03	C0MDATA403			√		Undefined
03FEC165H	CAN0 message data byte 5 register 03	C0MDATA503			√		Undefined
03FEC166H	CAN0 message data byte 67 register 03	C0MDATA6703				√	Undefined
03FEC166H	CAN0 message data byte 6 register 03	C0MDATA603			√		Undefined
03FEC167H	CAN0 message data byte 7 register 03	C0MDATA703			√		Undefined
03FEC168H	CAN0 message data length code register 03	C0MDLC03			√		0000xxxxB
03FEC169H	CAN0 message configuration register 03	C0MCONF03			√		Undefined
03FEC16AH	CAN0 message ID register 03	C0MIDL03				√	Undefined
03FEC16CH		C0MIDH03				√	Undefined
03FEC16EH	CAN0 message control register 03	C0MCTRL03				√	00x00000 000xx000B

(4/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC180H	CAN0 message data byte 01 register 04	C0MDATA0104	R/W			√	Undefined
03FEC180H	CAN0 message data byte 0 register 04	C0MDATA004			√		Undefined
03FEC181H	CAN0 message data byte 1 register 04	C0MDATA104			√		Undefined
03FEC182H	CAN0 message data byte 23 register 04	C0MDATA2304				√	Undefined
03FEC182H	CAN0 message data byte 2 register 04	C0MDATA204			√		Undefined
03FEC183H	CAN0 message data byte 3 register 04	C0MDATA304			√		Undefined
03FEC184H	CAN0 message data byte 45 register 04	C0MDATA4504				√	Undefined
03FEC184H	CAN0 message data byte 4 register 04	C0MDATA404			√		Undefined
03FEC185H	CAN0 message data byte 5 register 04	C0MDATA504			√		Undefined
03FEC186H	CAN0 message data byte 67 register 04	C0MDATA6704				√	Undefined
03FEC186H	CAN0 message data byte 6 register 04	C0MDATA604			√		Undefined
03FEC187H	CAN0 message data byte 7 register 04	C0MDATA704			√		Undefined
03FEC188H	CAN0 message data length code register 04	C0MDLC04			√		0000xxxxB
03FEC189H	CAN0 message configuration register 04	C0MCONF04			√		Undefined
03FEC18AH	CAN0 message ID register 04	C0MIDL04				√	Undefined
03FEC18CH		C0MIDH04				√	Undefined
03FEC18EH	CAN0 message control register 04	C0MCTRL04				√	00x00000 000xx000B
03FEC1A0H	CAN0 message data byte 01 register 05	C0MDATA0105				√	Undefined
03FEC1A0H	CAN0 message data byte 0 register 05	C0MDATA005			√		Undefined
03FEC1A1H	CAN0 message data byte 1 register 05	C0MDATA105			√		Undefined
03FEC1A2H	CAN0 message data byte 23 register 05	C0MDATA2305				√	Undefined
03FEC1A2H	CAN0 message data byte 2 register 05	C0MDATA205			√		Undefined
03FEC1A3H	CAN0 message data byte 3 register 05	C0MDATA305			√		Undefined
03FEC1A4H	CAN0 message data byte 45 register 05	C0MDATA4505				√	Undefined
03FEC1A4H	CAN0 message data byte 4 register 05	C0MDATA405			√		Undefined
03FEC1A5H	CAN0 message data byte 5 register 05	C0MDATA505			√		Undefined
03FEC1A6H	CAN0 message data byte 67 register 05	C0MDATA6705				√	Undefined
03FEC1A6H	CAN0 message data byte 6 register 05	C0MDATA605			√		Undefined
03FEC1A7H	CAN0 message data byte 7 register 05	C0MDATA705			√		Undefined
03FEC1A8H	CAN0 message data length code register 05	C0MDLC05			√		0000xxxxB
03FEC1A9H	CAN0 message configuration register 05	C0MCONF05			√		Undefined
03FEC1AAH	CAN0 message ID register 05	C0MIDL05				√	Undefined
03FEC1ACH		C0MIDH05				√	Undefined
03FEC1AEH	CAN0 message control register 05	C0MCTRL05				√	00x00000 000xx000B

(5/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC1C0H	CAN0 message data byte 01 register 06	C0MDATA0106	R/W			√	Undefined
03FEC1C0H	CAN0 message data byte 0 register 06	C0MDATA006			√		Undefined
03FEC1C1H	CAN0 message data byte 1 register 06	C0MDATA106			√		Undefined
03FEC1C2H	CAN0 message data byte 23 register 06	C0MDATA2306				√	Undefined
03FEC1C2H	CAN0 message data byte 2 register 06	C0MDATA206			√		Undefined
03FEC1C3H	CAN0 message data byte 3 register 06	C0MDATA306			√		Undefined
03FEC1C4H	CAN0 message data byte 45 register 06	C0MDATA4506				√	Undefined
03FEC1C4H	CAN0 message data byte 4 register 06	C0MDATA406			√		Undefined
03FEC1C5H	CAN0 message data byte 5 register 06	C0MDATA506			√		Undefined
03FEC1C6H	CAN0 message data byte 67 register 06	C0MDATA6706				√	Undefined
03FEC1C6H	CAN0 message data byte 6 register 06	C0MDATA606			√		Undefined
03FEC1C7H	CAN0 message data byte 7 register 06	C0MDATA706			√		Undefined
03FEC1C8H	CAN0 message data length code register 06	C0MDLC06			√		0000xxxxB
03FEC1C9H	CAN0 message configuration register 06	C0MCONF06			√		Undefined
03FEC1CAH	CAN0 message ID register 06	C0MIDL06				√	Undefined
03FEC1CCH		C0MIDH06				√	Undefined
03FEC1CEH	CAN0 message control register 06	C0MCTRL06				√	00x00000 000xx000B
03FEC1E0H	CAN0 message data byte 01 register 07	C0MDATA0107				√	Undefined
03FEC1E0H	CAN0 message data byte 0 register 07	C0MDATA007			√		Undefined
03FEC1E1H	CAN0 message data byte 1 register 07	C0MDATA107			√		Undefined
03FEC1E2H	CAN0 message data byte 23 register 07	C0MDATA2307				√	Undefined
03FEC1E2H	CAN0 message data byte 2 register 07	C0MDATA207			√		Undefined
03FEC1E3H	CAN0 message data byte 3 register 07	C0MDATA307			√		Undefined
03FEC1E4H	CAN0 message data byte 45 register 07	C0MDATA4507				√	Undefined
03FEC1E4H	CAN0 message data byte 4 register 07	C0MDATA407			√		Undefined
03FEC1E5H	CAN0 message data byte 5 register 07	C0MDATA507			√		Undefined
03FEC1E6H	CAN0 message data byte 67 register 07	C0MDATA6707				√	Undefined
03FEC1E6H	CAN0 message data byte 6 register 07	C0MDATA607			√		Undefined
03FEC1E7H	CAN0 message data byte 7 register 07	C0MDATA707			√		Undefined
03FEC1E8H	CAN0 message data length code register 07	C0MDLC07			√		0000xxxxB
03FEC1E9H	CAN0 message configuration register 07	C0MCONF07			√		Undefined
03FEC1EAH	CAN0 message ID register 07	C0MIDL07				√	Undefined
03FEC1ECH		C0MIDH07				√	Undefined
03FEC1EEH	CAN0 message control register 07	C0MCTRL07				√	00x00000 000xx000B

(6/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC200H	CAN0 message data byte 01 register 08	C0MDATA0108	R/W			√	Undefined
03FEC200H	CAN0 message data byte 0 register 08	C0MDATA008			√		Undefined
03FEC201H	CAN0 message data byte 1 register 08	C0MDATA108			√		Undefined
03FEC202H	CAN0 message data byte 23 register 08	C0MDATA2308				√	Undefined
03FEC202H	CAN0 message data byte 2 register 08	C0MDATA208			√		Undefined
03FEC203H	CAN0 message data byte 3 register 08	C0MDATA308			√		Undefined
03FEC204H	CAN0 message data byte 45 register 08	C0MDATA4508				√	Undefined
03FEC204H	CAN0 message data byte 4 register 08	C0MDATA408			√		Undefined
03FEC205H	CAN0 message data byte 5 register 08	C0MDATA508			√		Undefined
03FEC206H	CAN0 message data byte 67 register 08	C0MDATA6708				√	Undefined
03FEC206H	CAN0 message data byte 6 register 08	C0MDATA608			√		Undefined
03FEC207H	CAN0 message data byte 7 register 08	C0MDATA708			√		Undefined
03FEC208H	CAN0 message data length code register 08	C0MDLC08			√		0000xxxxB
03FEC209H	CAN0 message configuration register 08	C0MCONF08			√		Undefined
03FEC20AH	CAN0 message ID register 08	C0MIDL08				√	Undefined
03FEC20CH		C0MIDH08				√	Undefined
03FEC20EH	CAN0 message control register 08	C0MCTRL08				√	00x00000 000xx000B
03FEC220H	CAN0 message data byte 01 register 09	C0MDATA0109				√	Undefined
03FEC220H	CAN0 message data byte 0 register 09	C0MDATA009			√		Undefined
03FEC221H	CAN0 message data byte 1 register 09	C0MDATA109			√		Undefined
03FEC222H	CAN0 message data byte 23 register 09	C0MDATA2309				√	Undefined
03FEC222H	CAN0 message data byte 2 register 09	C0MDATA209			√		Undefined
03FEC223H	CAN0 message data byte 3 register 09	C0MDATA309			√		Undefined
03FEC224H	CAN0 message data byte 45 register 09	C0MDATA4509				√	Undefined
03FEC224H	CAN0 message data byte 4 register 09	C0MDATA409			√		Undefined
03FEC225H	CAN0 message data byte 5 register 09	C0MDATA509			√		Undefined
03FEC226H	CAN0 message data byte 67 register 09	C0MDATA6709				√	Undefined
03FEC226H	CAN0 message data byte 6 register 09	C0MDATA609			√		Undefined
03FEC227H	CAN0 message data byte 7 register 09	C0MDATA709			√		Undefined
03FEC228H	CAN0 message data length code register 09	C0MDLC09			√		0000xxxxB
03FEC229H	CAN0 message configuration register 09	C0MCONF09			√		Undefined
03FEC22AH	CAN0 message ID register 09	C0MIDL09				√	Undefined
03FEC22CH		C0MIDH09				√	Undefined
03FEC22EH	CAN0 message control register 09	C0MCTRL09				√	00x00000 000xx000B

(7/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC240H	CAN0 message data byte 01 register 10	C0MDATA0110	R/W			√	Undefined
03FEC240H	CAN0 message data byte 0 register 10	C0MDATA010			√		Undefined
03FEC241H	CAN0 message data byte 1 register 10	C0MDATA110			√		Undefined
03FEC242H	CAN0 message data byte 23 register 10	C0MDATA2310				√	Undefined
03FEC242H	CAN0 message data byte 2 register 10	C0MDATA210			√		Undefined
03FEC243H	CAN0 message data byte 3 register 10	C0MDATA310			√		Undefined
03FEC244H	CAN0 message data byte 45 register 10	C0MDATA4510				√	Undefined
03FEC244H	CAN0 message data byte 4 register 10	C0MDATA410			√		Undefined
03FEC245H	CAN0 message data byte 5 register 10	C0MDATA510			√		Undefined
03FEC246H	CAN0 message data byte 67 register 10	C0MDATA6710				√	Undefined
03FEC246H	CAN0 message data byte 6 register 10	C0MDATA610			√		Undefined
03FEC247H	CAN0 message data byte 7 register 10	C0MDATA710			√		Undefined
03FEC248H	CAN0 message data length code register 10	C0MDLC10			√		0000xxxxB
03FEC249H	CAN0 message configuration register 10	C0MCONF10			√		Undefined
03FEC24AH	CAN0 message ID register 10	C0MIDL10				√	Undefined
03FEC24CH		C0MIDH10				√	Undefined
03FEC24EH	CAN0 message control register 10	C0MCTRL10				√	00x00000 000xx000B
03FEC260H	CAN0 message data byte 01 register 11	C0MDATA0111				√	Undefined
03FEC260H	CAN0 message data byte 0 register 11	C0MDATA011			√		Undefined
03FEC261H	CAN0 message data byte 1 register 11	C0MDATA111			√		Undefined
03FEC262H	CAN0 message data byte 23 register 11	C0MDATA2311				√	Undefined
03FEC262H	CAN0 message data byte 2 register 11	C0MDATA211			√		Undefined
03FEC263H	CAN0 message data byte 3 register 11	C0MDATA311			√		Undefined
03FEC264H	CAN0 message data byte 45 register 11	C0MDATA4511				√	Undefined
03FEC264H	CAN0 message data byte 4 register 11	C0MDATA411			√		Undefined
03FEC265H	CAN0 message data byte 5 register 11	C0MDATA511			√		Undefined
03FEC266H	CAN0 message data byte 67 register 11	C0MDATA6711				√	Undefined
03FEC266H	CAN0 message data byte 6 register 11	C0MDATA611			√		Undefined
03FEC267H	CAN0 message data byte 7 register 11	C0MDATA711			√		Undefined
03FEC268H	CAN0 message data length code register 11	C0MDLC11			√		0000xxxxB
03FEC269H	CAN0 message configuration register 11	C0MCONF11			√		Undefined
03FEC26AH	CAN0 message ID register 11	C0MIDL11				√	Undefined
03FEC26CH		C0MIDH11				√	Undefined
03FEC26EH	CAN0 message control register 11	C0MCTRL11				√	00x00000 000xx000B

(8/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC280H	CAN0 message data byte 01 register 12	C0MDATA0112	R/W			√	Undefined
03FEC280H	CAN0 message data byte 0 register 12	C0MDATA012			√		Undefined
03FEC281H	CAN0 message data byte 1 register 12	C0MDATA112			√		Undefined
03FEC282H	CAN0 message data byte 23 register 12	C0MDATA2312				√	Undefined
03FEC282H	CAN0 message data byte 2 register 12	C0MDATA212			√		Undefined
03FEC283H	CAN0 message data byte 3 register 12	C0MDATA312			√		Undefined
03FEC284H	CAN0 message data byte 45 register 12	C0MDATA4512				√	Undefined
03FEC284H	CAN0 message data byte 4 register 12	C0MDATA412			√		Undefined
03FEC285H	CAN0 message data byte 5 register 12	C0MDATA512			√		Undefined
03FEC286H	CAN0 message data byte 67 register 12	C0MDATA6712				√	Undefined
03FEC286H	CAN0 message data byte 6 register 12	C0MDATA612			√		Undefined
03FEC287H	CAN0 message data byte 7 register 12	C0MDATA712			√		Undefined
03FEC288H	CAN0 message data length code register 12	C0MDLC12			√		0000xxxxB
03FEC289H	CAN0 message configuration register 12	C0MCONF12			√		Undefined
03FEC28AH	CAN0 message ID register 12	C0MIDL12				√	Undefined
03FEC28CH		C0MIDH12				√	Undefined
03FEC28EH	CAN0 message control register 12	C0MCTRL12				√	00x00000 000xx000B
03FEC2A0H	CAN0 message data byte 01 register 13	C0MDATA0113				√	Undefined
03FEC2A0H	CAN0 message data byte 0 register 13	C0MDATA013			√		Undefined
03FEC2A1H	CAN0 message data byte 1 register 13	C0MDATA113			√		Undefined
03FEC2A2H	CAN0 message data byte 23 register 13	C0MDATA2313				√	Undefined
03FEC2A2H	CAN0 message data byte 2 register 13	C0MDATA213			√		Undefined
03FEC2A3H	CAN0 message data byte 3 register 13	C0MDATA313			√		Undefined
03FEC2A4H	CAN0 message data byte 45 register 13	C0MDATA4513				√	Undefined
03FEC2A4H	CAN0 message data byte 4 register 13	C0MDATA413			√		Undefined
03FEC2A5H	CAN0 message data byte 5 register 13	C0MDATA513			√		Undefined
03FEC2A6H	CAN0 message data byte 67 register 13	C0MDATA6713				√	Undefined
03FEC2A6H	CAN0 message data byte 6 register 13	C0MDATA613			√		Undefined
03FEC2A7H	CAN0 message data byte 7 register 13	C0MDATA713			√		Undefined
03FEC2A8H	CAN0 message data length code register 13	C0MDLC13			√		0000xxxxB
03FEC2A9H	CAN0 message configuration register 13	C0MCONF13			√		Undefined
03FEC2AAH	CAN0 message ID register 13	C0MIDL13				√	Undefined
03FEC2ACH		C0MIDH13				√	Undefined
03FEC2AEH	CAN0 message control register 13	C0MCTRL13				√	00x00000 000xx000B

(9/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC2C0H	CAN0 message data byte 01 register 14	C0MDATA0114	R/W			√	Undefined
03FEC2C0H	CAN0 message data byte 0 register 14	C0MDATA014			√		Undefined
03FEC2C1H	CAN0 message data byte 1 register 14	C0MDATA114			√		Undefined
03FEC2C2H	CAN0 message data byte 23 register 14	C0MDATA2314				√	Undefined
03FEC2C2H	CAN0 message data byte 2 register 14	C0MDATA214			√		Undefined
03FEC2C3H	CAN0 message data byte 3 register 14	C0MDATA314			√		Undefined
03FEC2C4H	CAN0 message data byte 45 register 14	C0MDATA4514				√	Undefined
03FEC2C4H	CAN0 message data byte 4 register 14	C0MDATA414			√		Undefined
03FEC2C5H	CAN0 message data byte 5 register 14	C0MDATA514			√		Undefined
03FEC2C6H	CAN0 message data byte 67 register 14	C0MDATA6714				√	Undefined
03FEC2C6H	CAN0 message data byte 6 register 14	C0MDATA614			√		Undefined
03FEC2C7H	CAN0 message data byte 7 register 14	C0MDATA714			√		Undefined
03FEC2C8H	CAN0 message data length code register 14	C0MDLC14			√		0000xxxxB
03FEC2C9H	CAN0 message configuration register 14	C0MCONF14			√		Undefined
03FEC2CAH	CAN0 message ID register 14	C0MIDL14				√	Undefined
03FEC2CCH		C0MIDH14				√	Undefined
03FEC2CEH	CAN0 message control register 14	C0MCTRL14				√	00x00000 000xx000B
03FEC2E0H	CAN0 message data byte 01 register 15	C0MDATA0115				√	Undefined
03FEC2E0H	CAN0 message data byte 0 register 15	C0MDATA015			√		Undefined
03FEC2E1H	CAN0 message data byte 1 register 15	C0MDATA115			√		Undefined
03FEC2E2H	CAN0 message data byte 23 register 15	C0MDATA2315				√	Undefined
03FEC2E2H	CAN0 message data byte 2 register 15	C0MDATA215			√		Undefined
03FEC2E3H	CAN0 message data byte 3 register 15	C0MDATA315			√		Undefined
03FEC2E4H	CAN0 message data byte 45 register 15	C0MDATA4515				√	Undefined
03FEC2E4H	CAN0 message data byte 4 register 15	C0MDATA415			√		Undefined
03FEC2E5H	CAN0 message data byte 5 register 15	C0MDATA515			√		Undefined
03FEC2E6H	CAN0 message data byte 67 register 15	C0MDATA6715				√	Undefined
03FEC2E6H	CAN0 message data byte 6 register 15	C0MDATA615			√		Undefined
03FEC2E7H	CAN0 message data byte 7 register 15	C0MDATA715			√		Undefined
03FEC2E8H	CAN0 message data length code register 15	C0MDLC15			√		0000xxxx
03FEC2E9H	CAN0 message configuration register 15	C0MCONF15			√		Undefined
03FEC2EAH	CAN0 message ID register 15	C0MIDL15				√	Undefined
03FEC2ECH		C0MIDH15				√	Undefined
03FEC2EEH	CAN0 message control register 15	C0MCTRL15				√	00x00000 000xx000B

(10/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC300H	CAN0 message data byte 01 register 16	C0MDATA0116	R/W			√	Undefined
03FEC300H	CAN0 message data byte 0 register 16	C0MDATA016			√		Undefined
03FEC301H	CAN0 message data byte 1 register 16	C0MDATA116			√		Undefined
03FEC302H	CAN0 message data byte 23 register 16	C0MDATA2316				√	Undefined
03FEC302H	CAN0 message data byte 2 register 16	C0MDATA216			√		Undefined
03FEC303H	CAN0 message data byte 3 register 16	C0MDATA316			√		Undefined
03FEC304H	CAN0 message data byte 45 register 16	C0MDATA4516				√	Undefined
03FEC304H	CAN0 message data byte 4 register 16	C0MDATA416			√		Undefined
03FEC305H	CAN0 message data byte 5 register 16	C0MDATA516			√		Undefined
03FEC306H	CAN0 message data byte 67 register 16	C0MDATA6716				√	Undefined
03FEC306H	CAN0 message data byte 6 register 16	C0MDATA616			√		Undefined
03FEC307H	CAN0 message data byte 7 register 16	C0MDATA716			√		Undefined
03FEC308H	CAN0 message data length code register 16	C0MDLC16			√		0000xxxxB
03FEC309H	CAN0 message configuration register 16	C0MCONF16			√		Undefined
03FEC30AH	CAN0 message ID register 16	C0MIDL16				√	Undefined
03FEC30CH		C0MIDH16				√	Undefined
03FEC30EH	CAN0 message control register 16	C0MCTRL16				√	00x00000 000xx000B
03FEC320H	CAN0 message data byte 01 register 17	C0MDATA0117				√	Undefined
03FEC320H	CAN0 message data byte 0 register 17	C0MDATA017			√		Undefined
03FEC321H	CAN0 message data byte 1 register 17	C0MDATA117			√		Undefined
03FEC322H	CAN0 message data byte 23 register 17	C0MDATA2317				√	Undefined
03FEC322H	CAN0 message data byte 2 register 17	C0MDATA217			√		Undefined
03FEC323H	CAN0 message data byte 3 register 17	C0MDATA317			√		Undefined
03FEC324H	CAN0 message data byte 45 register 17	C0MDATA4517				√	Undefined
03FEC324H	CAN0 message data byte 4 register 17	C0MDATA417			√		Undefined
03FEC325H	CAN0 message data byte 5 register 17	C0MDATA517			√		Undefined
03FEC326H	CAN0 message data byte 67 register 17	C0MDATA6717				√	Undefined
03FEC326H	CAN0 message data byte 6 register 17	C0MDATA617			√		Undefined
03FEC327H	CAN0 message data byte 7 register 17	C0MDATA717			√		Undefined
03FEC328H	CAN0 message data length code register 17	C0MDLC17			√		0000xxxxB
03FEC329H	CAN0 message configuration register 17	C0MCONF17			√		Undefined
03FEC32AH	CAN0 message ID register 17	C0MIDL17				√	Undefined
03FEC32CH		C0MIDH17				√	Undefined
03FEC32EH	CAN0 message control register 17	C0MCTRL17				√	00x00000 000xx000B

(11/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC340H	CAN0 message data byte 01 register 18	C0MDATA0118	R/W			√	Undefined
03FEC340H	CAN0 message data byte 0 register 18	C0MDATA018			√		Undefined
03FEC341H	CAN0 message data byte 1 register 18	C0MDATA118			√		Undefined
03FEC342H	CAN0 message data byte 23 register 18	C0MDATA2318				√	Undefined
03FEC342H	CAN0 message data byte 2 register 18	C0MDATA218			√		Undefined
03FEC343H	CAN0 message data byte 3 register 18	C0MDATA318			√		Undefined
03FEC344H	CAN0 message data byte 45 register 18	C0MDATA4518				√	Undefined
03FEC344H	CAN0 message data byte 4 register 18	C0MDATA418			√		Undefined
03FEC345H	CAN0 message data byte 5 register 18	C0MDATA518			√		Undefined
03FEC346H	CAN0 message data byte 67 register 18	C0MDATA6718				√	Undefined
03FEC346H	CAN0 message data byte 6 register 18	C0MDATA618			√		Undefined
03FEC347H	CAN0 message data byte 7 register 18	C0MDATA718			√		Undefined
03FEC348H	CAN0 message data length code register 18	C0MDLC18			√		0000xxxxB
03FEC349H	CAN0 message configuration register 18	C0MCONF18			√		Undefined
03FEC34AH	CAN0 message ID register 18	C0MIDL18				√	Undefined
03FEC34CH		C0MIDH18				√	Undefined
03FEC34EH	CAN0 message control register 18	C0MCTRL18				√	00x00000 000xx000B
03FEC360H	CAN0 message data byte 01 register 19	C0MDATA0119				√	Undefined
03FEC360H	CAN0 message data byte 0 register 19	C0MDATA019			√		Undefined
03FEC361H	CAN0 message data byte 1 register 19	C0MDATA119			√		Undefined
03FEC362H	CAN0 message data byte 23 register 19	C0MDATA2319				√	Undefined
03FEC362H	CAN0 message data byte 2 register 19	C0MDATA219			√		Undefined
03FEC363H	CAN0 message data byte 3 register 19	C0MDATA319			√		Undefined
03FEC364H	CAN0 message data byte 45 register 19	C0MDATA4519				√	Undefined
03FEC364H	CAN0 message data byte 4 register 19	C0MDATA419			√		Undefined
03FEC365H	CAN0 message data byte 5 register 19	C0MDATA519			√		Undefined
03FEC366H	CAN0 message data byte 67 register 19	C0MDATA6719				√	Undefined
03FEC366H	CAN0 message data byte 6 register 19	C0MDATA619			√		Undefined
03FEC367H	CAN0 message data byte 7 register 19	C0MDATA719			√		Undefined
03FEC368H	CAN0 message data length code register 19	C0MDLC19			√		0000xxxxB
03FEC369H	CAN0 message configuration register 19	C0MCONF19			√		Undefined
03FEC36AH	CAN0 message ID register 19	C0MIDL19				√	Undefined
03FEC36CH		C0MIDH19				√	Undefined
03FEC36EH	CAN0 message control register 19	C0MCTRL19				√	00x00000 000xx000B

(12/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC380H	CAN0 message data byte 01 register 20	C0MDATA0120	R/W			√	Undefined
03FEC380H	CAN0 message data byte 0 register 20	C0MDATA020			√		Undefined
03FEC381H	CAN0 message data byte 1 register 20	C0MDATA120			√		Undefined
03FEC382H	CAN0 message data byte 23 register 20	C0MDATA2320				√	Undefined
03FEC382H	CAN0 message data byte 2 register 20	C0MDATA220			√		Undefined
03FEC383H	CAN0 message data byte 3 register 20	C0MDATA320			√		Undefined
03FEC384H	CAN0 message data byte 45 register 20	C0MDATA4520				√	Undefined
03FEC384H	CAN0 message data byte 4 register 20	C0MDATA420			√		Undefined
03FEC385H	CAN0 message data byte 5 register 20	C0MDATA520			√		Undefined
03FEC386H	CAN0 message data byte 67 register 20	C0MDATA6720				√	Undefined
03FEC386H	CAN0 message data byte 6 register 20	C0MDATA620			√		Undefined
03FEC387H	CAN0 message data byte 7 register 20	C0MDATA720			√		Undefined
03FEC388H	CAN0 message data length code register 20	C0MDLC20			√		0000xxxxB
03FEC389H	CAN0 message configuration register 20	C0MCONF20			√		Undefined
03FEC38AH	CAN0 message ID register 20	C0MIDL20				√	Undefined
03FEC38CH		C0MIDH20				√	Undefined
03FEC38EH	CAN0 message control register 20	C0MCTRL20				√	00x00000 000xx000B
03FEC3A0H	CAN0 message data byte 01 register 21	C0MDATA0121				√	Undefined
03FEC3A0H	CAN0 message data byte 0 register 21	C0MDATA021			√		Undefined
03FEC3A1H	CAN0 message data byte 1 register 21	C0MDATA121			√		Undefined
03FEC3A2H	CAN0 message data byte 23 register 21	C0MDATA2321				√	Undefined
03FEC3A2H	CAN0 message data byte 2 register 21	C0MDATA221			√		Undefined
03FEC3A3H	CAN0 message data byte 3 register 21	C0MDATA321			√		Undefined
03FEC3A4H	CAN0 message data byte 45 register 21	C0MDATA4521				√	Undefined
03FEC3A4H	CAN0 message data byte 4 register 21	C0MDATA421			√		Undefined
03FEC3A5H	CAN0 message data byte 5 register 21	C0MDATA521			√		Undefined
03FEC3A6H	CAN0 message data byte 67 register 21	C0MDATA6721				√	Undefined
03FEC3A6H	CAN0 message data byte 6 register 21	C0MDATA621			√		Undefined
03FEC3A7H	CAN0 message data byte 7 register 21	C0MDATA721			√		Undefined
03FEC3A8H	CAN0 message data length code register 21	C0MDLC21			√		0000xxxxB
03FEC3A9H	CAN0 message configuration register 21	C0MCONF21			√		Undefined
03FEC3AAH	CAN0 message ID register 21	C0MIDL21				√	Undefined
03FEC3ACH		C0MIDH21				√	Undefined
03FEC3AEH	CAN0 message control register 21	C0MCTRL21				√	00x00000 000xx000B

(13/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC3C0H	CAN0 message data byte 01 register 22	C0MDATA0122	R/W			√	Undefined
03FEC3C0H	CAN0 message data byte 0 register 22	C0MDATA022			√		Undefined
03FEC3C1H	CAN0 message data byte 1 register 22	C0MDATA122			√		Undefined
03FEC3C2H	CAN0 message data byte 23 register 22	C0MDATA2322				√	Undefined
03FEC3C2H	CAN0 message data byte 2 register 22	C0MDATA222			√		Undefined
03FEC3C3H	CAN0 message data byte 3 register 22	C0MDATA322			√		Undefined
03FEC3C4H	CAN0 message data byte 45 register 22	C0MDATA4522				√	Undefined
03FEC3C4H	CAN0 message data byte 4 register 22	C0MDATA422			√		Undefined
03FEC3C5H	CAN0 message data byte 5 register 22	C0MDATA522			√		Undefined
03FEC3C6H	CAN0 message data byte 67 register 22	C0MDATA6722				√	Undefined
03FEC3C6H	CAN0 message data byte 6 register 22	C0MDATA622			√		Undefined
03FEC3C7H	CAN0 message data byte 7 register 22	C0MDATA722			√		Undefined
03FEC3C8H	CAN0 message data length code register 22	C0MDLC22			√		0000xxxxB
03FEC3C9H	CAN0 message configuration register 22	C0MCONF22			√		Undefined
03FEC3CAH	CAN0 message ID register 22	C0MIDL22				√	Undefined
03FEC3CCH		C0MIDH22				√	Undefined
03FEC3CEH	CAN0 message control register 22	C0MCTRL22				√	00x00000 000xx000B
03FEC3E0H	CAN0 message data byte 01 register 23	C0MDATA0123				√	Undefined
03FEC3E0H	CAN0 message data byte 0 register 23	C0MDATA023			√		Undefined
03FEC3E1H	CAN0 message data byte 1 register 23	C0MDATA123			√		Undefined
03FEC3E2H	CAN0 message data byte 23 register 23	C0MDATA2323				√	Undefined
03FEC3E2H	CAN0 message data byte 2 register 23	C0MDATA223			√		Undefined
03FEC3E3H	CAN0 message data byte 3 register 23	C0MDATA323			√		Undefined
03FEC3E4H	CAN0 message data byte 45 register 23	C0MDATA4523				√	Undefined
03FEC3E4H	CAN0 message data byte 4 register 23	C0MDATA423			√		Undefined
03FEC3E5H	CAN0 message data byte 5 register 23	C0MDATA523			√		Undefined
03FEC3E6H	CAN0 message data byte 67 register 23	C0MDATA6723				√	Undefined
03FEC3E6H	CAN0 message data byte 6 register 23	C0MDATA623			√		Undefined
03FEC3E7H	CAN0 message data byte 7 register 23	C0MDATA723			√		Undefined
03FEC3E8H	CAN0 message data length code register 23	C0MDLC23			√		0000xxxxB
03FEC3E9H	CAN0 message configuration register 23	C0MCONF23			√		Undefined
03FEC3EAH	CAN0 message ID register 23	C0MIDL23				√	Undefined
03FEC3ECH		C0MIDH23				√	Undefined
03FEC3EEH	CAN0 message control register 23	C0MCTRL23				√	00x00000 000xx000B

(14/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC400H	CAN0 message data byte 01 register 24	C0MDATA0124	R/W			√	Undefined
03FEC400H	CAN0 message data byte 0 register 24	C0MDATA024			√		Undefined
03FEC401H	CAN0 message data byte 1 register 24	C0MDATA124			√		Undefined
03FEC402H	CAN0 message data byte 23 register 24	C0MDATA2324				√	Undefined
03FEC402H	CAN0 message data byte 2 register 24	C0MDATA224			√		Undefined
03FEC403H	CAN0 message data byte 3 register 24	C0MDATA324			√		Undefined
03FEC404H	CAN0 message data byte 45 register 24	C0MDATA4524				√	Undefined
03FEC404H	CAN0 message data byte 4 register 24	C0MDATA424			√		Undefined
03FEC405H	CAN0 message data byte 5 register 24	C0MDATA524			√		Undefined
03FEC406H	CAN0 message data byte 67 register 24	C0MDATA6724				√	Undefined
03FEC406H	CAN0 message data byte 6 register 24	C0MDATA624			√		Undefined
03FEC407H	CAN0 message data byte 7 register 24	C0MDATA724			√		Undefined
03FEC408H	CAN0 message data length code register 24	C0MDLC24			√		0000xxxxB
03FEC409H	CAN0 message configuration register 24	C0MCONF24			√		Undefined
03FEC40AH	CAN0 message ID register 24	C0MIDL24				√	Undefined
03FEC40CH		C0MIDH24				√	Undefined
03FEC40EH	CAN0 message control register 24	C0MCTRL24				√	00x00000 000xx000B
03FEC420H	CAN0 message data byte 01 register 25	C0MDATA0125				√	Undefined
03FEC420H	CAN0 message data byte 0 register 25	C0MDATA025			√		Undefined
03FEC421H	CAN0 message data byte 1 register 25	C0MDATA125			√		Undefined
03FEC422H	CAN0 message data byte 23 register 25	C0MDATA2325				√	Undefined
03FEC422H	CAN0 message data byte 2 register 25	C0MDATA225			√		Undefined
03FEC423H	CAN0 message data byte 3 register 25	C0MDATA325			√		Undefined
03FEC424H	CAN0 message data byte 45 register 25	C0MDATA4525				√	Undefined
03FEC424H	CAN0 message data byte 4 register 25	C0MDATA425			√		Undefined
03FEC425H	CAN0 message data byte 5 register 25	C0MDATA525			√		Undefined
03FEC426H	CAN0 message data byte 67 register 25	C0MDATA6725				√	Undefined
03FEC426H	CAN0 message data byte 6 register 25	C0MDATA625			√		Undefined
03FEC427H	CAN0 message data byte 7 register 25	C0MDATA725			√		Undefined
03FEC428H	CAN0 message data length code register 25	C0MDLC25			√		0000xxxxB
03FEC429H	CAN0 message configuration register 25	C0MCONF25			√		Undefined
03FEC42AH	CAN0 message ID register 25	C0MIDL25				√	Undefined
03FEC42CH		C0MIDH25				√	Undefined
03FEC42EH	CAN0 message control register 25	C0MCTRL25				√	00x00000 000xx000B

(15/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC440H	CAN0 message data byte 01 register 26	C0MDATA0126	R/W			√	Undefined
03FEC440H	CAN0 message data byte 0 register 26	C0MDATA026			√		Undefined
03FEC441H	CAN0 message data byte 1 register 26	C0MDATA126			√		Undefined
03FEC442H	CAN0 message data byte 23 register 26	C0MDATA2326				√	Undefined
03FEC442H	CAN0 message data byte 2 register 26	C0MDATA226			√		Undefined
03FEC443H	CAN0 message data byte 3 register 26	C0MDATA326			√		Undefined
03FEC444H	CAN0 message data byte 45 register 26	C0MDATA4526				√	Undefined
03FEC444H	CAN0 message data byte 4 register 26	C0MDATA426			√		Undefined
03FEC445H	CAN0 message data byte 5 register 26	C0MDATA526			√		Undefined
03FEC446H	CAN0 message data byte 67 register 26	C0MDATA6726				√	Undefined
03FEC446H	CAN0 message data byte 6 register 26	C0MDATA626			√		Undefined
03FEC447H	CAN0 message data byte 7 register 26	C0MDATA726			√		Undefined
03FEC448H	CAN0 message data length code register 26	C0MDLC26			√		0000xxxxB
03FEC449H	CAN0 message configuration register 26	C0MCONF26			√		Undefined
03FEC44AH	CAN0 message ID register 26	C0MIDL26				√	Undefined
03FEC44CH		C0MIDH26				√	Undefined
03FEC44EH	CAN0 message control register 26	C0MCTRL26				√	00x00000 000xx000B
03FEC460H	CAN0 message data byte 01 register 27	C0MDATA0127				√	Undefined
03FEC460H	CAN0 message data byte 0 register 27	C0MDATA027			√		Undefined
03FEC461H	CAN0 message data byte 1 register 27	C0MDATA127			√		Undefined
03FEC462H	CAN0 message data byte 23 register 27	C0MDATA2327				√	Undefined
03FEC462H	CAN0 message data byte 2 register 27	C0MDATA227			√		Undefined
03FEC463H	CAN0 message data byte 3 register 27	C0MDATA327			√		Undefined
03FEC464H	CAN0 message data byte 45 register 27	C0MDATA4527				√	Undefined
03FEC464H	CAN0 message data byte 4 register 27	C0MDATA427			√		Undefined
03FEC465H	CAN0 message data byte 5 register 27	C0MDATA527			√		Undefined
03FEC466H	CAN0 message data byte 67 register 27	C0MDATA6727				√	Undefined
03FEC466H	CAN0 message data byte 6 register 27	C0MDATA627			√		Undefined
03FEC467H	CAN0 message data byte 7 register 27	C0MDATA727			√		Undefined
03FEC468H	CAN0 message data length code register 27	C0MDLC27			√		0000xxxxB
03FEC469H	CAN0 message configuration register 27	C0MCONF27			√		Undefined
03FEC46AH	CAN0 message ID register 27	C0MIDL27				√	Undefined
03FEC46CH		C0MIDH27				√	Undefined
03FEC46EH	CAN0 message control register 27	C0MCTRL27				√	00x00000 000xx000B

(16/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC480H	CAN0 message data byte 01 register 28	C0MDATA0128	R/W			√	Undefined
03FEC480H	CAN0 message data byte 0 register 28	C0MDATA028			√		Undefined
03FEC481H	CAN0 message data byte 1 register 28	C0MDATA128			√		Undefined
03FEC482H	CAN0 message data byte 23 register 28	C0MDATA2328				√	Undefined
03FEC482H	CAN0 message data byte 2 register 28	C0MDATA228			√		Undefined
03FEC483H	CAN0 message data byte 3 register 28	C0MDATA328			√		Undefined
03FEC484H	CAN0 message data byte 45 register 28	C0MDATA4528				√	Undefined
03FEC484H	CAN0 message data byte 4 register 28	C0MDATA428			√		Undefined
03FEC485H	CAN0 message data byte 5 register 28	C0MDATA528			√		Undefined
03FEC486H	CAN0 message data byte 67 register 28	C0MDATA6728				√	Undefined
03FEC486H	CAN0 message data byte 6 register 28	C0MDATA628			√		Undefined
03FEC487H	CAN0 message data byte 7 register 28	C0MDATA728			√		Undefined
03FEC488H	CAN0 message data length code register 28	C0MDLC28			√		0000xxxxB
03FEC489H	CAN0 message configuration register 28	C0MCONF28			√		Undefined
03FEC48AH	CAN0 message ID register 28	C0MIDL28				√	Undefined
03FEC48CH		C0MIDH28				√	Undefined
03FEC48EH	CAN0 message control register 28	C0MCTRL28				√	00x00000 000xx000B
03FEC4A0H	CAN0 message data byte 01 register 29	C0MDATA0129				√	Undefined
03FEC4A0H	CAN0 message data byte 0 register 29	C0MDATA029			√		Undefined
03FEC4A1H	CAN0 message data byte 1 register 29	C0MDATA129			√		Undefined
03FEC4A2H	CAN0 message data byte 23 register 29	C0MDATA2329				√	Undefined
03FEC4A2H	CAN0 message data byte 2 register 29	C0MDATA229			√		Undefined
03FEC4A3H	CAN0 message data byte 3 register 29	C0MDATA329			√		Undefined
03FEC4A4H	CAN0 message data byte 45 register 29	C0MDATA4529				√	Undefined
03FEC4A4H	CAN0 message data byte 4 register 29	C0MDATA429			√		Undefined
03FEC4A5H	CAN0 message data byte 5 register 29	C0MDATA529			√		Undefined
03FEC4A6H	CAN0 message data byte 67 register 29	C0MDATA6729				√	Undefined
03FEC4A6H	CAN0 message data byte 6 register 29	C0MDATA629			√		Undefined
03FEC4A7H	CAN0 message data byte 7 register 29	C0MDATA729			√		Undefined
03FEC4A8H	CAN0 message data length code register 29	C0MDLC29			√		0000xxxxB
03FEC4A9H	CAN0 message configuration register 29	C0MCONF29			√		Undefined
03FEC4AAH	CAN0 message ID register 29	C0MIDL29				√	Undefined
03FEC4ACH		C0MIDH29				√	Undefined
03FEC4AEH	CAN0 message control register 29	C0MCTRL29				√	00x00000 000xx000B

(17/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC4C0H	CAN0 message data byte 01 register 30	C0MDATA0130	R/W			√	Undefined
03FEC4C0H	CAN0 message data byte 0 register 30	C0MDATA030			√		Undefined
03FEC4C1H	CAN0 message data byte 1 register 30	C0MDATA130			√		Undefined
03FEC4C2H	CAN0 message data byte 23 register 30	C0MDATA2330				√	Undefined
03FEC4C2H	CAN0 message data byte 2 register 30	C0MDATA230			√		Undefined
03FEC4C3H	CAN0 message data byte 3 register 30	C0MDATA330			√		Undefined
03FEC4C4H	CAN0 message data byte 45 register 30	C0MDATA4530				√	Undefined
03FEC4C4H	CAN0 message data byte 4 register 30	C0MDATA430			√		Undefined
03FEC4C5H	CAN0 message data byte 5 register 30	C0MDATA530			√		Undefined
03FEC4C6H	CAN0 message data byte 67 register 30	C0MDATA6730				√	Undefined
03FEC4C6H	CAN0 message data byte 6 register 30	C0MDATA630			√		Undefined
03FEC4C7H	CAN0 message data byte 7 register 30	C0MDATA730			√		Undefined
03FEC4C8H	CAN0 message data length code register 30	C0MDLC30			√		0000xxxxB
03FEC4C9H	CAN0 message configuration register 30	C0MCONF30			√		Undefined
03FEC4CAH	CAN0 message ID register 30	C0MIDL30				√	Undefined
03FEC4CCH		C0MIDH30				√	Undefined
03FEC4CEH	CAN0 message control register 30	C0MCTRL30				√	00x00000 000xx000B
03FEC4E0H	CAN0 message data byte 01 register 31	C0MDATA0131				√	Undefined
03FEC4E0H	CAN0 message data byte 0 register 31	C0MDATA031			√		Undefined
03FEC4E1H	CAN0 message data byte 1 register 31	C0MDATA131			√		Undefined
03FEC4E2H	CAN0 message data byte 23 register 31	C0MDATA2331				√	Undefined
03FEC4E2H	CAN0 message data byte 2 register 31	C0MDATA231			√		Undefined
03FEC4E3H	CAN0 message data byte 3 register 31	C0MDATA331			√		Undefined
03FEC4E4H	CAN0 message data byte 45 register 31	C0MDATA4531				√	Undefined
03FEC4E4H	CAN0 message data byte 4 register 31	C0MDATA431			√		Undefined
03FEC4E5H	CAN0 message data byte 5 register 31	C0MDATA531			√		Undefined
03FEC4E6H	CAN0 message data byte 67 register 31	C0MDATA6731				√	Undefined
03FEC4E6H	CAN0 message data byte 6 register 31	C0MDATA631			√		Undefined
03FEC4E7H	CAN0 message data byte 7 register 31	C0MDATA731			√		Undefined
03FEC4E8H	CAN0 message data length code register 31	C0MDLC31			√		0000xxxx
03FEC4E9H	CAN0 message configuration register 31	C0MCONF31			√		Undefined
03FEC4EAH	CAN0 message ID register 31	C0MIDL31				√	Undefined
03FEC4ECH		C0MIDH31				√	Undefined
03FEC4EEH	CAN0 message control register 31	C0MCTRL31				√	00x00000 000xx000B

(18/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC600H	CAN1 global control register	C1GMCTRL	R/W	–	–	√	0000H
03FEC602H	CAN1 global clock select register	C1GMCS	R/W	–	√	–	0FH
03FEC606H	CAN1 global block transmission control register	C1GMABT	R/W	–	–	√	0000H
03FEC608H	CAN1 global 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
03FEC666H	CAN1 module time stamp register	C1TS	R/W	–	–	√	0000H

(19/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC700H	CAN1 message data byte 01 register 00	C1MDATA0100	R/W			√	Undefined
03FEC700H	CAN1 message data byte 0 register 00	C1MDATA000			√		Undefined
03FEC701H	CAN1 message data byte 1 register 00	C1MDATA100			√		Undefined
03FEC702H	CAN1 message data byte 23 register 00	C1MDATA2300				√	Undefined
03FEC702H	CAN1 message data byte 2 register 00	C1MDATA200			√		Undefined
03FEC703H	CAN1 message data byte 3 register 00	C1MDATA300			√		Undefined
03FEC704H	CAN1 message data byte 45 register 00	C1MDATA4500				√	Undefined
03FEC704H	CAN1 message data byte 4 register 00	C1MDATA400			√		Undefined
03FEC705H	CAN1 message data byte 5 register 00	C1MDATA500			√		Undefined
03FEC706H	CAN1 message data byte 67 register 00	C1MDATA6700				√	Undefined
03FEC706H	CAN1 message data byte 6 register 00	C1MDATA600			√		Undefined
03FEC707H	CAN1 message data byte 7 register 00	C1MDATA700			√		Undefined
03FEC708H	CAN1 message data length code register 00	C1MDLC00			√		0000xxxxB
03FEC709H	CAN1 message configuration register 00	C1MCONF00			√		Undefined
03FEC70AH	CAN1 message ID register 00	C1MIDL00				√	Undefined
03FEC70CH		C1MIDH00				√	Undefined
03FEC70EH	CAN1 message control register 00	C1MCTRL00				√	00x00000 000xx000B
03FEC720H	CAN1 message data byte 01 register 01	C1MDATA0101				√	Undefined
03FEC720H	CAN1 message data byte 0 register 01	C1MDATA001			√		Undefined
03FEC721H	CAN1 message data byte 1 register 01	C1MDATA101			√		Undefined
03FEC722H	CAN1 message data byte 23 register 01	C1MDATA2301				√	Undefined
03FEC722H	CAN1 message data byte 2 register 01	C1MDATA201			√		Undefined
03FEC723H	CAN1 message data byte 3 register 01	C1MDATA301			√		Undefined
03FEC724H	CAN1 message data byte 45 register 01	C1MDATA4501				√	Undefined
03FEC724H	CAN1 message data byte 4 register 01	C1MDATA401			√		Undefined
03FEC725H	CAN1 message data byte 5 register 01	C1MDATA501			√		Undefined
03FEC726H	CAN1 message data byte 67 register 01	C1MDATA6701				√	Undefined
03FEC726H	CAN1 message data byte 6 register 01	C1MDATA601			√		Undefined
03FEC727H	CAN1 message data byte 7 register 01	C1MDATA701			√		Undefined
03FEC728H	CAN1 message data length code register 01	C1MDLC01			√		0000xxxxB
03FEC729H	CAN1 message configuration register 01	C1MCONF01			√		Undefined
03FEC72AH	CAN1 message ID register 01	C1MIDL01				√	Undefined
03FEC72CH		C1MIDH01				√	Undefined
03FEC72EH	CAN1 message control register 01	C1MCTRL01				√	00x00000 000xx000B

(20/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC740H	CAN1 message data byte 01 register 02	C1MDATA0102	R/W			√	Undefined
03FEC740H	CAN1 message data byte 0 register 02	C1MDATA002			√		Undefined
03FEC741H	CAN1 message data byte 1 register 02	C1MDATA102			√		Undefined
03FEC742H	CAN1 message data byte 23 register 02	C1MDATA2302				√	Undefined
03FEC742H	CAN1 message data byte 2 register 02	C1MDATA202			√		Undefined
03FEC743H	CAN1 message data byte 3 register 02	C1MDATA302			√		Undefined
03FEC744H	CAN1 message data byte 45 register 02	C1MDATA4502				√	Undefined
03FEC744H	CAN1 message data byte 4 register 02	C1MDATA402			√		Undefined
03FEC745H	CAN1 message data byte 5 register 02	C1MDATA502			√		Undefined
03FEC746H	CAN1 message data byte 67 register 02	C1MDATA6702				√	Undefined
03FEC746H	CAN1 message data byte 6 register 02	C1MDATA602			√		Undefined
03FEC747H	CAN1 message data byte 7 register 02	C1MDATA702			√		Undefined
03FEC748H	CAN1 message data length code register 02	C1MDLC02			√		0000xxxxB
03FEC749H	CAN1 message configuration register 02	C1MCONF02			√		Undefined
03FEC74AH	CAN1 message ID register 02	C1MIDL02				√	Undefined
03FEC74CH		C1MIDH02				√	Undefined
03FEC74EH	CAN1 message control register 02	C1MCTRL02				√	00x00000 000xx000B
03FEC760H	CAN1 message data byte 01 register 03	C1MDATA0103				√	Undefined
03FEC760H	CAN1 message data byte 0 register 03	C1MDATA003			√		Undefined
03FEC761H	CAN1 message data byte 1 register 03	C1MDATA103			√		Undefined
03FEC762H	CAN1 message data byte 23 register 03	C1MDATA2303				√	Undefined
03FEC762H	CAN1 message data byte 2 register 03	C1MDATA203			√		Undefined
03FEC763H	CAN1 message data byte 3 register 03	C1MDATA303			√		Undefined
03FEC764H	CAN1 message data byte 45 register 03	C1MDATA4503				√	Undefined
03FEC764H	CAN1 message data byte 4 register 03	C1MDATA403			√		Undefined
03FEC765H	CAN1 message data byte 5 register 03	C1MDATA503			√		Undefined
03FEC766H	CAN1 message data byte 67 register 03	C1MDATA6703				√	Undefined
03FEC766H	CAN1 message data byte 6 register 03	C1MDATA603			√		Undefined
03FEC767H	CAN1 message data byte 7 register 03	C1MDATA703			√		Undefined
03FEC768H	CAN1 message data length code register 03	C1MDLC03			√		0000xxxxB
03FEC769H	CAN1 message configuration register 03	C1MCONF03			√		Undefined
03FEC76AH	CAN1 message ID register 03	C1MIDL03				√	Undefined
03FEC76CH		C1MIDH03				√	Undefined
03FEC76EH	CAN1 message control register 03	C1MCTRL03				√	00x00000 000xx000B

(21/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC780H	CAN1 message data byte 01 register 04	C1MDATA0104	R/W			√	Undefined
03FEC780H	CAN1 message data byte 0 register 04	C1MDATA004			√		Undefined
03FEC781H	CAN1 message data byte 1 register 04	C1MDATA104			√		Undefined
03FEC782H	CAN1 message data byte 23 register 04	C1MDATA2304				√	Undefined
03FEC782H	CAN1 message data byte 2 register 04	C1MDATA204			√		Undefined
03FEC783H	CAN1 message data byte 3 register 04	C1MDATA304			√		Undefined
03FEC784H	CAN1 message data byte 45 register 04	C1MDATA4504				√	Undefined
03FEC784H	CAN1 message data byte 4 register 04	C1MDATA404			√		Undefined
03FEC785H	CAN1 message data byte 5 register 04	C1MDATA504			√		Undefined
03FEC786H	CAN1 message data byte 67 register 04	C1MDATA6704				√	Undefined
03FEC786H	CAN1 message data byte 6 register 04	C1MDATA604			√		Undefined
03FEC787H	CAN1 message data byte 7 register 04	C1MDATA704			√		Undefined
03FEC788H	CAN1 message data length code register 04	C1MDLC04			√		0000xxxxB
03FEC789H	CAN1 message configuration register 04	C1MCONF04			√		Undefined
03FEC78AH	CAN1 message ID register 04	C1MIDL04				√	Undefined
03FEC78CH		C1MIDH04				√	Undefined
03FEC78EH	CAN1 message control register 04	C1MCTRL04				√	00x00000 000xx000B
03FEC7A0H	CAN1 message data byte 01 register 05	C1MDATA0105				√	Undefined
03FEC7A0H	CAN1 message data byte 0 register 05	C1MDATA005			√		Undefined
03FEC7A1H	CAN1 message data byte 1 register 05	C1MDATA105			√		Undefined
03FEC7A2H	CAN1 message data byte 23 register 05	C1MDATA2305				√	Undefined
03FEC7A2H	CAN1 message data byte 2 register 05	C1MDATA205			√		Undefined
03FEC7A3H	CAN1 message data byte 3 register 05	C1MDATA305			√		Undefined
03FEC7A4H	CAN1 message data byte 45 register 05	C1MDATA4505				√	Undefined
03FEC7A4H	CAN1 message data byte 4 register 05	C1MDATA405			√		Undefined
03FEC7A5H	CAN1 message data byte 5 register 05	C1MDATA505			√		Undefined
03FEC7A6H	CAN1 message data byte 67 register 05	C1MDATA6705				√	Undefined
03FEC7A6H	CAN1 message data byte 6 register 05	C1MDATA605			√		Undefined
03FEC7A7H	CAN1 message data byte 7 register 05	C1MDATA705			√		Undefined
03FEC7A8H	CAN1 message data length code register 05	C1MDLC05			√		0000xxxxB
03FEC7A9H	CAN1 message configuration register 05	C1MCONF05			√		Undefined
03FEC7AAH	CAN1 message ID register 05	C1MIDL05				√	Undefined
03FEC7ACH		C1MIDH05				√	Undefined
03FEC7AEH	CAN1 message control register 05	C1MCTRL05				√	00x00000 000xx000B

(22/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC7C0H	CAN1 message data byte 01 register 06	C1MDATA0106	R/W			√	Undefined
03FEC7C0H	CAN1 message data byte 0 register 06	C1MDATA006			√		Undefined
03FEC7C1H	CAN1 message data byte 1 register 06	C1MDATA106			√		Undefined
03FEC7C2H	CAN1 message data byte 23 register 06	C1MDATA2306				√	Undefined
03FEC7C2H	CAN1 message data byte 2 register 06	C1MDATA206			√		Undefined
03FEC7C3H	CAN1 message data byte 3 register 06	C1MDATA306			√		Undefined
03FEC7C4H	CAN1 message data byte 45 register 06	C1MDATA4506				√	Undefined
03FEC7C4H	CAN1 message data byte 4 register 06	C1MDATA406			√		Undefined
03FEC7C5H	CAN1 message data byte 5 register 06	C1MDATA506			√		Undefined
03FEC7C6H	CAN1 message data byte 67 register 06	C1MDATA6706				√	Undefined
03FEC7C6H	CAN1 message data byte 6 register 06	C1MDATA606			√		Undefined
03FEC7C7H	CAN1 message data byte 7 register 06	C1MDATA706			√		Undefined
03FEC7C8H	CAN1 message data length code register 06	C1MDLC06			√		0000xxxxB
03FEC7C9H	CAN1 message configuration register 06	C1MCONF06			√		Undefined
03FEC7CAH	CAN1 message ID register 06	C1MIDL06				√	Undefined
03FEC7CCH		C1MIDH06				√	Undefined
03FEC7CEH	CAN1 message control register 06	C1MCTRL06				√	00x00000 000xx000B
03FEC7E0H	CAN1 message data byte 01 register 07	C1MDATA0107				√	Undefined
03FEC7E0H	CAN1 message data byte 0 register 07	C1MDATA007			√		Undefined
03FEC7E1H	CAN1 message data byte 1 register 07	C1MDATA107			√		Undefined
03FEC7E2H	CAN1 message data byte 23 register 07	C1MDATA2307				√	Undefined
03FEC7E2H	CAN1 message data byte 2 register 07	C1MDATA207			√		Undefined
03FEC7E3H	CAN1 message data byte 3 register 07	C1MDATA307			√		Undefined
03FEC7E4H	CAN1 message data byte 45 register 07	C1MDATA4507				√	Undefined
03FEC7E4H	CAN1 message data byte 4 register 07	C1MDATA407			√		Undefined
03FEC7E5H	CAN1 message data byte 5 register 07	C1MDATA507			√		Undefined
03FEC7E6H	CAN1 message data byte 67 register 07	C1MDATA6707				√	Undefined
03FEC7E6H	CAN1 message data byte 6 register 07	C1MDATA607			√		Undefined
03FEC7E7H	CAN1 message data byte 7 register 07	C1MDATA707			√		Undefined
03FEC7E8H	CAN1 message data length code register 07	C1MDLC07			√		0000xxxxB
03FEC7E9H	CAN1 message configuration register 07	C1MCONF07			√		Undefined
03FEC7EAH	CAN1 message ID register 07	C1MIDL07				√	Undefined
03FEC7ECH		C1MIDH07				√	Undefined
03FEC7EEH	CAN1 message control register 07	C1MCTRL07				√	00x00000 000xx000B

(23/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC800H	CAN1 message data byte 01 register 08	C1MDATA0108	R/W			√	Undefined
03FEC800H	CAN1 message data byte 0 register 08	C1MDATA008			√		Undefined
03FEC801H	CAN1 message data byte 1 register 08	C1MDATA108			√		Undefined
03FEC802H	CAN1 message data byte 23 register 08	C1MDATA2308				√	Undefined
03FEC802H	CAN1 message data byte 2 register 08	C1MDATA208			√		Undefined
03FEC803H	CAN1 message data byte 3 register 08	C1MDATA308			√		Undefined
03FEC804H	CAN1 message data byte 45 register 08	C1MDATA4508				√	Undefined
03FEC804H	CAN1 message data byte 4 register 08	C1MDATA408			√		Undefined
03FEC805H	CAN1 message data byte 5 register 08	C1MDATA508			√		Undefined
03FEC806H	CAN1 message data byte 67 register 08	C1MDATA6708				√	Undefined
03FEC806H	CAN1 message data byte 6 register 08	C1MDATA608			√		Undefined
03FEC807H	CAN1 message data byte 7 register 08	C1MDATA708			√		Undefined
03FEC808H	CAN1 message data length code register 08	C1MDLC08			√		0000xxxxB
03FEC809H	CAN1 message configuration register 08	C1MCONF08			√		Undefined
03FEC80AH	CAN1 message ID register 08	C1MIDL08				√	Undefined
03FEC80CH		C1MIDH08				√	Undefined
03FEC80EH	CAN1 message control register 08	C1MCTRL08				√	00x00000 000xx000B
03FEC820H	CAN1 message data byte 01 register 09	C1MDATA0109				√	Undefined
03FEC820H	CAN1 message data byte 0 register 09	C1MDATA009			√		Undefined
03FEC821H	CAN1 message data byte 1 register 09	C1MDATA109			√		Undefined
03FEC822H	CAN1 message data byte 23 register 09	C1MDATA2309				√	Undefined
03FEC822H	CAN1 message data byte 2 register 09	C1MDATA209			√		Undefined
03FEC823H	CAN1 message data byte 3 register 09	C1MDATA309			√		Undefined
03FEC824H	CAN1 message data byte 45 register 09	C1MDATA4509				√	Undefined
03FEC824H	CAN1 message data byte 4 register 09	C1MDATA409			√		Undefined
03FEC825H	CAN1 message data byte 5 register 09	C1MDATA509			√		Undefined
03FEC826H	CAN1 message data byte 67 register 09	C1MDATA6709				√	Undefined
03FEC826H	CAN1 message data byte 6 register 09	C1MDATA609			√		Undefined
03FEC827H	CAN1 message data byte 7 register 09	C1MDATA709			√		Undefined
03FEC828H	CAN1 message data length code register 09	C1MDLC09			√		0000xxxxB
03FEC829H	CAN1 message configuration register 09	C1MCONF09			√		Undefined
03FEC82AH	CAN1 message ID register 09	C1MIDL09				√	Undefined
03FEC82CH		C1MIDH09				√	Undefined
03FEC82EH	CAN1 message control register 09	C1MCTRL09				√	00x00000 000xx000B

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC840H	CAN1 message data byte 01 register 10	C1MDATA0110	R/W			√	Undefined
03FEC840H	CAN1 message data byte 0 register 10	C1MDATA010			√		Undefined
03FEC841H	CAN1 message data byte 1 register 10	C1MDATA110			√		Undefined
03FEC842H	CAN1 message data byte 23 register 10	C1MDATA2310				√	Undefined
03FEC842H	CAN1 message data byte 2 register 10	C1MDATA210			√		Undefined
03FEC843H	CAN1 message data byte 3 register 10	C1MDATA310			√		Undefined
03FEC844H	CAN1 message data byte 45 register 10	C1MDATA4510				√	Undefined
03FEC844H	CAN1 message data byte 4 register 10	C1MDATA410			√		Undefined
03FEC845H	CAN1 message data byte 5 register 10	C1MDATA510			√		Undefined
03FEC846H	CAN1 message data byte 67 register 10	C1MDATA6710				√	Undefined
03FEC846H	CAN1 message data byte 6 register 10	C1MDATA610			√		Undefined
03FEC847H	CAN1 message data byte 7 register 10	C1MDATA710			√		Undefined
03FEC848H	CAN1 message data length code register 10	C1MDLC10			√		0000xxxxB
03FEC849H	CAN1 message configuration register 10	C1MCONF10			√		Undefined
03FEC84AH	CAN1 message ID register 10	C1MIDL10				√	Undefined
03FEC84CH		C1MIDH10				√	Undefined
03FEC84EH	CAN1 message control register 10	C1MCTRL10				√	00x00000 000xx000B
03FEC860H	CAN1 message data byte 01 register 11	C1MDATA0111				√	Undefined
03FEC860H	CAN1 message data byte 0 register 11	C1MDATA011			√		Undefined
03FEC861H	CAN1 message data byte 1 register 11	C1MDATA111			√		Undefined
03FEC862H	CAN1 message data byte 23 register 11	C1MDATA2311				√	Undefined
03FEC862H	CAN1 message data byte 2 register 11	C1MDATA211			√		Undefined
03FEC863H	CAN1 message data byte 3 register 11	C1MDATA311			√		Undefined
03FEC864H	CAN1 message data byte 45 register 11	C1MDATA4511				√	Undefined
03FEC864H	CAN1 message data byte 4 register 11	C1MDATA411			√		Undefined
03FEC865H	CAN1 message data byte 5 register 11	C1MDATA511			√		Undefined
03FEC866H	CAN1 message data byte 67 register 11	C1MDATA6711				√	Undefined
03FEC866H	CAN1 message data byte 6 register 11	C1MDATA611			√		Undefined
03FEC867H	CAN1 message data byte 7 register 11	C1MDATA711			√		Undefined
03FEC868H	CAN1 message data length code register 11	C1MDLC11			√		0000xxxxB
03FEC869H	CAN1 message configuration register 11	C1MCONF11			√		Undefined
03FEC86AH	CAN1 message ID register 11	C1MIDL11				√	Undefined
03FEC86CH		C1MIDH11				√	Undefined
03FEC86EH	CAN1 message control register 11	C1MCTRL11				√	00x00000 000xx000B

(25/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC880H	CAN1 message data byte 01 register 12	C1MDATA0112	R/W			√	Undefined
03FEC880H	CAN1 message data byte 0 register 12	C1MDATA012			√		Undefined
03FEC881H	CAN1 message data byte 1 register 12	C1MDATA112			√		Undefined
03FEC882H	CAN1 message data byte 23 register 12	C1MDATA2312				√	Undefined
03FEC882H	CAN1 message data byte 2 register 12	C1MDATA212			√		Undefined
03FEC883H	CAN1 message data byte 3 register 12	C1MDATA312			√		Undefined
03FEC884H	CAN1 message data byte 45 register 12	C1MDATA4512				√	Undefined
03FEC884H	CAN1 message data byte 4 register 12	C1MDATA412			√		Undefined
03FEC885H	CAN1 message data byte 5 register 12	C1MDATA512			√		Undefined
03FEC886H	CAN1 message data byte 67 register 12	C1MDATA6712				√	Undefined
03FEC886H	CAN1 message data byte 6 register 12	C1MDATA612			√		Undefined
03FEC887H	CAN1 message data byte 7 register 12	C1MDATA712			√		Undefined
03FEC888H	CAN1 message data length code register 12	C1MDLC12			√		0000xxxxB
03FEC889H	CAN1 message configuration register 12	C1MCONF12			√		Undefined
03FEC88AH	CAN1 message ID register 12	C1MIDL12				√	Undefined
03FEC88CH		C1MIDH12				√	Undefined
03FEC88EH	CAN1 message control register 12	C1MCTRL12				√	00x00000 000xx000B
03FEC8A0H	CAN1 message data byte 01 register 13	C1MDATA0113				√	Undefined
03FEC8A0H	CAN1 message data byte 0 register 13	C1MDATA013			√		Undefined
03FEC8A1H	CAN1 message data byte 1 register 13	C1MDATA113			√		Undefined
03FEC8A2H	CAN1 message data byte 23 register 13	C1MDATA2313				√	Undefined
03FEC8A2H	CAN1 message data byte 2 register 13	C1MDATA213			√		Undefined
03FEC8A3H	CAN1 message data byte 3 register 13	C1MDATA313			√		Undefined
03FEC8A4H	CAN1 message data byte 45 register 13	C1MDATA4513				√	Undefined
03FEC8A4H	CAN1 message data byte 4 register 13	C1MDATA413			√		Undefined
03FEC8A5H	CAN1 message data byte 5 register 13	C1MDATA513			√		Undefined
03FEC8A6H	CAN1 message data byte 67 register 13	C1MDATA6713				√	Undefined
03FEC8A6H	CAN1 message data byte 6 register 13	C1MDATA613			√		Undefined
03FEC8A7H	CAN1 message data byte 7 register 13	C1MDATA713			√		Undefined
03FEC8A8H	CAN1 message data length code register 13	C1MDLC13			√		0000xxxxB
03FEC8A9H	CAN1 message configuration register 13	C1MCONF13			√		Undefined
03FEC8AAH	CAN1 message ID register 13	C1MIDL13				√	Undefined
03FEC8ACH		C1MIDH13				√	Undefined
03FEC8AEH	CAN1 message control register 13	C1MCTRL13				√	00x00000 000xx000B

(26/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC8C0H	CAN1 message data byte 01 register 14	C1MDATA0114	R/W			√	Undefined
03FEC8C0H	CAN1 message data byte 0 register 14	C1MDATA014			√		Undefined
03FEC8C1H	CAN1 message data byte 1 register 14	C1MDATA114			√		Undefined
03FEC8C2H	CAN1 message data byte 23 register 14	C1MDATA2314				√	Undefined
03FEC8C2H	CAN1 message data byte 2 register 14	C1MDATA214			√		Undefined
03FEC8C3H	CAN1 message data byte 3 register 14	C1MDATA314			√		Undefined
03FEC8C4H	CAN1 message data byte 45 register 14	C1MDATA4514				√	Undefined
03FEC8C4H	CAN1 message data byte 4 register 14	C1MDATA414			√		Undefined
03FEC8C5H	CAN1 message data byte 5 register 14	C1MDATA514			√		Undefined
03FEC8C6H	CAN1 message data byte 67 register 14	C1MDATA6714				√	Undefined
03FEC8C6H	CAN1 message data byte 6 register 14	C1MDATA614			√		Undefined
03FEC8C7H	CAN1 message data byte 7 register 14	C1MDATA714			√		Undefined
03FEC8C8H	CAN1 message data length code register 14	C1MDLC14			√		0000xxxxB
03FEC8C9H	CAN1 message configuration register 14	C1MCONF14			√		Undefined
03FEC8CAH	CAN1 message ID register 14	C1MIDL14				√	Undefined
03FEC8CCH		C1MIDH14				√	Undefined
03FEC8CEH	CAN1 message control register 14	C1MCTRL14				√	00x00000 000xx000B
03FEC8E0H	CAN1 message data byte 01 register 15	C1MDATA0115				√	Undefined
03FEC8E0H	CAN1 message data byte 0 register 15	C1MDATA015			√		Undefined
03FEC8E1H	CAN1 message data byte 1 register 15	C1MDATA115			√		Undefined
03FEC8E2H	CAN1 message data byte 23 register 15	C1MDATA2315				√	Undefined
03FEC8E2H	CAN1 message data byte 2 register 15	C1MDATA215			√		Undefined
03FEC8E3H	CAN1 message data byte 3 register 15	C1MDATA315			√		Undefined
03FEC8E4H	CAN1 message data byte 45 register 15	C1MDATA4515				√	Undefined
03FEC8E4H	CAN1 message data byte 4 register 15	C1MDATA415			√		Undefined
03FEC8E5H	CAN1 message data byte 5 register 15	C1MDATA515			√		Undefined
03FEC8E6H	CAN1 message data byte 67 register 15	C1MDATA6715				√	Undefined
03FEC8E6H	CAN1 message data byte 6 register 15	C1MDATA615			√		Undefined
03FEC8E7H	CAN1 message data byte 7 register 15	C1MDATA715			√		Undefined
03FEC8E8H	CAN1 message data length code register 15	C1MDLC15			√		0000xxxxB
03FEC8E9H	CAN1 message configuration register 15	C1MCONF15			√		Undefined
03FEC8EAH	CAN1 message ID register 15	C1MIDL15				√	Undefined
03FEC8ECH		C1MIDH15				√	Undefined
03FEC8EEH	CAN1 message control register 15	C1MCTRL15				√	00x00000 000xx000B

(27/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC900H	CAN1 message data byte 01 register 16	C1MDATA0116	R/W			√	Undefined
03FEC900H	CAN1 message data byte 0 register 16	C1MDATA016			√		Undefined
03FEC901H	CAN1 message data byte 1 register 16	C1MDATA116			√		Undefined
03FEC902H	CAN1 message data byte 23 register 16	C1MDATA2316				√	Undefined
03FEC902H	CAN1 message data byte 2 register 16	C1MDATA216			√		Undefined
03FEC903H	CAN1 message data byte 3 register 16	C1MDATA316			√		Undefined
03FEC904H	CAN1 message data byte 45 register 16	C1MDATA4516				√	Undefined
03FEC904H	CAN1 message data byte 4 register 16	C1MDATA416			√		Undefined
03FEC905H	CAN1 message data byte 5 register 16	C1MDATA516			√		Undefined
03FEC906H	CAN1 message data byte 67 register 16	C1MDATA6716				√	Undefined
03FEC906H	CAN1 message data byte 6 register 16	C1MDATA616			√		Undefined
03FEC907H	CAN1 message data byte 7 register 16	C1MDATA716			√		Undefined
03FEC908H	CAN1 message data length code register 16	C1MDLC16			√		0000xxxxB
03FEC909H	CAN1 message configuration register 16	C1MCONF16			√		Undefined
03FEC90AH	CAN1 message ID register 16	C1MIDL16				√	Undefined
03FEC90CH		C1MIDH16				√	Undefined
03FEC90EH	CAN1 message control register 16	C1MCTRL16				√	00x00000 000xx000B
03FEC920H	CAN1 message data byte 01 register 17	C1MDATA0117				√	Undefined
03FEC920H	CAN1 message data byte 0 register 17	C1MDATA017			√		Undefined
03FEC921H	CAN1 message data byte 1 register 17	C1MDATA117			√		Undefined
03FEC922H	CAN1 message data byte 23 register 17	C1MDATA2317				√	Undefined
03FEC922H	CAN1 message data byte 2 register 17	C1MDATA217			√		Undefined
03FEC923H	CAN1 message data byte 3 register 17	C1MDATA317			√		Undefined
03FEC924H	CAN1 message data byte 45 register 17	C1MDATA4517				√	Undefined
03FEC924H	CAN1 message data byte 4 register 17	C1MDATA417			√		Undefined
03FEC925H	CAN1 message data byte 5 register 17	C1MDATA517			√		Undefined
03FEC926H	CAN1 message data byte 67 register 17	C1MDATA6717				√	Undefined
03FEC926H	CAN1 message data byte 6 register 17	C1MDATA617			√		Undefined
03FEC927H	CAN1 message data byte 7 register 17	C1MDATA717			√		Undefined
03FEC928H	CAN1 message data length code register 17	C1MDLC17			√		0000xxxxB
03FEC929H	CAN1 message configuration register 17	C1MCONF17			√		Undefined
03FEC92AH	CAN1 message ID register 17	C1MIDL17				√	Undefined
03FEC92CH		C1MIDH17				√	Undefined
03FEC92EH	CAN1 message control register 17	C1MCTRL17				√	00x00000 000xx000B

(28/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC940H	CAN1 message data byte 01 register 18	C1MDATA0118	R/W			√	Undefined
03FEC940H	CAN1 message data byte 0 register 18	C1MDATA018			√		Undefined
03FEC941H	CAN1 message data byte 1 register 18	C1MDATA118			√		Undefined
03FEC942H	CAN1 message data byte 23 register 18	C1MDATA2318				√	Undefined
03FEC942H	CAN1 message data byte 2 register 18	C1MDATA218			√		Undefined
03FEC943H	CAN1 message data byte 3 register 18	C1MDATA318			√		Undefined
03FEC944H	CAN1 message data byte 45 register 18	C1MDATA4518				√	Undefined
03FEC944H	CAN1 message data byte 4 register 18	C1MDATA418			√		Undefined
03FEC945H	CAN1 message data byte 5 register 18	C1MDATA518			√		Undefined
03FEC946H	CAN1 message data byte 67 register 18	C1MDATA6718				√	Undefined
03FEC946H	CAN1 message data byte 6 register 18	C1MDATA618			√		Undefined
03FEC947H	CAN1 message data byte 7 register 18	C1MDATA718			√		Undefined
03FEC948H	CAN1 message data length code register 18	C1MDLC18			√		0000xxxxB
03FEC949H	CAN1 message configuration register 18	C1MCONF18			√		Undefined
03FEC94AH	CAN1 message ID register 18	C1MIDL18				√	Undefined
03FEC94CH		C1MIDH18				√	Undefined
03FEC94EH	CAN1 message control register 18	C1MCTRL18				√	00x00000 000xx000B
03FEC960H	CAN1 message data byte 01 register 19	C1MDATA0119				√	Undefined
03FEC960H	CAN1 message data byte 0 register 19	C1MDATA019			√		Undefined
03FEC961H	CAN1 message data byte 1 register 19	C1MDATA119			√		Undefined
03FEC962H	CAN1 message data byte 23 register 19	C1MDATA2319				√	Undefined
03FEC962H	CAN1 message data byte 2 register 19	C1MDATA219			√		Undefined
03FEC963H	CAN1 message data byte 3 register 19	C1MDATA319			√		Undefined
03FEC964H	CAN1 message data byte 45 register 19	C1MDATA4519				√	Undefined
03FEC964H	CAN1 message data byte 4 register 19	C1MDATA419			√		Undefined
03FEC965H	CAN1 message data byte 5 register 19	C1MDATA519			√		Undefined
03FEC966H	CAN1 message data byte 67 register 19	C1MDATA6719				√	Undefined
03FEC966H	CAN1 message data byte 6 register 19	C1MDATA619			√		Undefined
03FEC967H	CAN1 message data byte 7 register 19	C1MDATA719			√		Undefined
03FEC968H	CAN1 message data length code register 19	C1MDLC19			√		0000xxxxB
03FEC969H	CAN1 message configuration register 19	C1MCONF19			√		Undefined
03FEC96AH	CAN1 message ID register 19	C1MIDL19				√	Undefined
03FEC96CH		C1MIDH19				√	Undefined
03FEC96EH	CAN1 message control register 19	C1MCTRL19				√	00x00000 000xx000B

(29/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC980H	CAN1 message data byte 01 register 20	C1MDATA0120	R/W			√	Undefined
03FEC980H	CAN1 message data byte 0 register 20	C1MDATA020			√		Undefined
03FEC981H	CAN1 message data byte 1 register 20	C1MDATA120			√		Undefined
03FEC982H	CAN1 message data byte 23 register 20	C1MDATA2320				√	Undefined
03FEC982H	CAN1 message data byte 2 register 20	C1MDATA220			√		Undefined
03FEC983H	CAN1 message data byte 3 register 20	C1MDATA320			√		Undefined
03FEC984H	CAN1 message data byte 45 register 20	C1MDATA4520				√	Undefined
03FEC984H	CAN1 message data byte 4 register 20	C1MDATA420			√		Undefined
03FEC985H	CAN1 message data byte 5 register 20	C1MDATA520			√		Undefined
03FEC986H	CAN1 message data byte 67 register 20	C1MDATA6720				√	Undefined
03FEC986H	CAN1 message data byte 6 register 20	C1MDATA620			√		Undefined
03FEC987H	CAN1 message data byte 7 register 20	C1MDATA720			√		Undefined
03FEC988H	CAN1 message data length code register 20	C1MDLC20			√		0000xxxxB
03FEC989H	CAN1 message configuration register 20	C1MCONF20			√		Undefined
03FEC98AH	CAN1 message ID register 20	C1MIDL20				√	Undefined
03FEC98CH		C1MIDH20				√	Undefined
03FEC98EH	CAN1 message control register 20	C1MCTRL20				√	00x00000 000xx000B
03FEC9A0H	CAN1 message data byte 01 register 21	C1MDATA0121				√	Undefined
03FEC9A0H	CAN1 message data byte 0 register 21	C1MDATA021			√		Undefined
03FEC9A1H	CAN1 message data byte 1 register 21	C1MDATA121			√		Undefined
03FEC9A2H	CAN1 message data byte 23 register 21	C1MDATA2321				√	Undefined
03FEC9A2H	CAN1 message data byte 2 register 21	C1MDATA221			√		Undefined
03FEC9A3H	CAN1 message data byte 3 register 21	C1MDATA321			√		Undefined
03FEC9A4H	CAN1 message data byte 45 register 21	C1MDATA4521				√	Undefined
03FEC9A4H	CAN1 message data byte 4 register 21	C1MDATA421			√		Undefined
03FEC9A5H	CAN1 message data byte 5 register 21	C1MDATA521			√		Undefined
03FEC9A6H	CAN1 message data byte 67 register 21	C1MDATA6721				√	Undefined
03FEC9A6H	CAN1 message data byte 6 register 21	C1MDATA621			√		Undefined
03FEC9A7H	CAN1 message data byte 7 register 21	C1MDATA721			√		Undefined
03FEC9A8H	CAN1 message data length code register 21	C1MDLC21			√		0000xxxxB
03FEC9A9H	CAN1 message configuration register 21	C1MCONF21			√		Undefined
03FEC9AAH	CAN1 message ID register 21	C1MIDL21				√	Undefined
03FEC9ACH		C1MIDH21				√	Undefined
03FEC9AEH	CAN1 message control register 21	C1MCTRL21				√	00x00000 000xx000B

(30/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FEC9C0H	CAN1 message data byte 01 register 22	C1MDATA0122	R/W			√	Undefined
03FEC9C0H	CAN1 message data byte 0 register 22	C1MDATA022			√		Undefined
03FEC9C1H	CAN1 message data byte 1 register 22	C1MDATA122			√		Undefined
03FEC9C2H	CAN1 message data byte 23 register 22	C1MDATA2322				√	Undefined
03FEC9C2H	CAN1 message data byte 2 register 22	C1MDATA222			√		Undefined
03FEC9C3H	CAN1 message data byte 3 register 22	C1MDATA322			√		Undefined
03FEC9C4H	CAN1 message data byte 45 register 22	C1MDATA4522				√	Undefined
03FEC9C4H	CAN1 message data byte 4 register 22	C1MDATA422			√		Undefined
03FEC9C5H	CAN1 message data byte 5 register 22	C1MDATA522			√		Undefined
03FEC9C6H	CAN1 message data byte 67 register 22	C1MDATA6722				√	Undefined
03FEC9C6H	CAN1 message data byte 6 register 22	C1MDATA622			√		Undefined
03FEC9C7H	CAN1 message data byte 7 register 22	C1MDATA722			√		Undefined
03FEC9C8H	CAN1 message data length code register 22	C1MDLC22			√		0000xxxxB
03FEC9C9H	CAN1 message configuration register 22	C1MCONF22			√		Undefined
03FEC9CAH	CAN1 message ID register 22	C1MIDL22				√	Undefined
03FEC9CCH		C1MIDH22				√	Undefined
03FEC9CEH	CAN1 message control register 22	C1MCTRL22				√	00x00000 000xx000B
03FEC9E0H	CAN1 message data byte 01 register 23	C1MDATA0123				√	Undefined
03FEC9E0H	CAN1 message data byte 0 register 23	C1MDATA023			√		Undefined
03FEC9E1H	CAN1 message data byte 1 register 23	C1MDATA123			√		Undefined
03FEC9E2H	CAN1 message data byte 23 register 23	C1MDATA2323				√	Undefined
03FEC9E2H	CAN1 message data byte 2 register 23	C1MDATA223			√		Undefined
03FEC9E3H	CAN1 message data byte 3 register 23	C1MDATA323			√		Undefined
03FEC9E4H	CAN1 message data byte 45 register 23	C1MDATA4523				√	Undefined
03FEC9E4H	CAN1 message data byte 4 register 23	C1MDATA423			√		Undefined
03FEC9E5H	CAN1 message data byte 5 register 23	C1MDATA523			√		Undefined
03FEC9E6H	CAN1 message data byte 67 register 23	C1MDATA6723				√	Undefined
03FEC9E6H	CAN1 message data byte 6 register 23	C1MDATA623			√		Undefined
03FEC9E7H	CAN1 message data byte 7 register 23	C1MDATA723			√		Undefined
03FEC9E8H	CAN1 message data length code register 23	C1MDLC23			√		0000xxxxB
03FEC9E9H	CAN1 message configuration register 23	C1MCONF23			√		Undefined
03FEC9EAH	CAN1 message ID register 23	C1MIDL23				√	Undefined
03FEC9ECH		C1MIDH23				√	Undefined
03FEC9EEH	CAN1 message control register 23	C1MCTRL23				√	00x00000 000xx000B

(31/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECA00H	CAN1 message data byte 01 register 24	C1MDATA0124	R/W			√	Undefined
03FECA00H	CAN1 message data byte 0 register 24	C1MDATA024			√		Undefined
03FECA01H	CAN1 message data byte 1 register 24	C1MDATA124			√		Undefined
03FECA02H	CAN1 message data byte 23 register 24	C1MDATA2324				√	Undefined
03FECA02H	CAN1 message data byte 2 register 24	C1MDATA224			√		Undefined
03FECA03H	CAN1 message data byte 3 register 24	C1MDATA324			√		Undefined
03FECA04H	CAN1 message data byte 45 register 24	C1MDATA4524				√	Undefined
03FECA04H	CAN1 message data byte 4 register 24	C1MDATA424			√		Undefined
03FECA05H	CAN1 message data byte 5 register 24	C1MDATA524			√		Undefined
03FECA06H	CAN1 message data byte 67 register 24	C1MDATA6724				√	Undefined
03FECA06H	CAN1 message data byte 6 register 24	C1MDATA624			√		Undefined
03FECA07H	CAN1 message data byte 7 register 24	C1MDATA724			√		Undefined
03FECA08H	CAN1 message data length code register 24	C1MDLC24			√		0000xxxxB
03FECA09H	CAN1 message configuration register 24	C1MCONF24			√		Undefined
03FECA0AH	CAN1 message ID register 24	C1MIDL24				√	Undefined
03FECA0CH		C1MIDH24				√	Undefined
03FECA0EH	CAN1 message control register 24	C1MCTRL24				√	00x00000 000xx000B
03FECA20H	CAN1 message data byte 01 register 25	C1MDATA0125				√	Undefined
03FECA20H	CAN1 message data byte 0 register 25	C1MDATA025			√		Undefined
03FECA21H	CAN1 message data byte 1 register 25	C1MDATA125			√		Undefined
03FECA22H	CAN1 message data byte 23 register 25	C1MDATA2325				√	Undefined
03FECA22H	CAN1 message data byte 2 register 25	C1MDATA225			√		Undefined
03FECA23H	CAN1 message data byte 3 register 25	C1MDATA325			√		Undefined
03FECA24H	CAN1 message data byte 45 register 25	C1MDATA4525				√	Undefined
03FECA24H	CAN1 message data byte 4 register 25	C1MDATA425			√		Undefined
03FECA25H	CAN1 message data byte 5 register 25	C1MDATA525			√		Undefined
03FECA26H	CAN1 message data byte 67 register 25	C1MDATA6725				√	Undefined
03FECA26H	CAN1 message data byte 6 register 25	C1MDATA625			√		Undefined
03FECA27H	CAN1 message data byte 7 register 25	C1MDATA725			√		Undefined
03FECA28H	CAN1 message data length code register 25	C1MDLC25			√		0000xxxxB
03FECA29H	CAN1 message configuration register 25	C1MCONF25			√		Undefined
03FECA2AH	CAN1 message ID register 25	C1MIDL25				√	Undefined
03FECA2CH		C1MIDH25				√	Undefined
03FECA2EH	CAN1 message control register 25	C1MCTRL25				√	00x00000 000xx000B

(32/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECA40H	CAN1 message data byte 01 register 26	C1MDATA0126	R/W			√	Undefined
03FECA40H	CAN1 message data byte 0 register 26	C1MDATA026			√		Undefined
03FECA41H	CAN1 message data byte 1 register 26	C1MDATA126			√		Undefined
03FECA42H	CAN1 message data byte 23 register 26	C1MDATA2326				√	Undefined
03FECA42H	CAN1 message data byte 2 register 26	C1MDATA226			√		Undefined
03FECA43H	CAN1 message data byte 3 register 26	C1MDATA326			√		Undefined
03FECA44H	CAN1 message data byte 45 register 26	C1MDATA4526				√	Undefined
03FECA44H	CAN1 message data byte 4 register 26	C1MDATA426			√		Undefined
03FECA45H	CAN1 message data byte 5 register 26	C1MDATA526			√		Undefined
03FECA46H	CAN1 message data byte 67 register 26	C1MDATA6726				√	Undefined
03FECA46H	CAN1 message data byte 6 register 26	C1MDATA626			√		Undefined
03FECA47H	CAN1 message data byte 7 register 26	C1MDATA726			√		Undefined
03FECA48H	CAN1 message data length code register 26	C1MDLC26			√		0000xxxxB
03FECA49H	CAN1 message configuration register 26	C1MCONF26			√		Undefined
03FECA4AH	CAN1 message ID register 26	C1MIDL26				√	Undefined
03FECA4CH		C1MIDH26				√	Undefined
03FECA4EH	CAN1 message control register 26	C1MCTRL26				√	00x00000 000xx000B
03FECA60H	CAN1 message data byte 01 register 27	C1MDATA0127				√	Undefined
03FECA60H	CAN1 message data byte 0 register 27	C1MDATA027			√		Undefined
03FECA61H	CAN1 message data byte 1 register 27	C1MDATA127			√		Undefined
03FECA62H	CAN1 message data byte 23 register 27	C1MDATA2327				√	Undefined
03FECA62H	CAN1 message data byte 2 register 27	C1MDATA227			√		Undefined
03FECA63H	CAN1 message data byte 3 register 27	C1MDATA327			√		Undefined
03FECA64H	CAN1 message data byte 45 register 27	C1MDATA4527				√	Undefined
03FECA64H	CAN1 message data byte 4 register 27	C1MDATA427			√		Undefined
03FECA65H	CAN1 message data byte 5 register 27	C1MDATA527			√		Undefined
03FECA66H	CAN1 message data byte 67 register 27	C1MDATA6727				√	Undefined
03FECA66H	CAN1 message data byte 6 register 27	C1MDATA627			√		Undefined
03FECA67H	CAN1 message data byte 7 register 27	C1MDATA727			√		Undefined
03FECA68H	CAN1 message data length code register 27	C1MDLC27			√		0000xxxxB
03FECA69H	CAN1 message configuration register 27	C1MCONF27			√		Undefined
03FECA6AH	CAN1 message ID register 27	C1MIDL27				√	Undefined
03FECA6CH		C1MIDH27				√	Undefined
03FECA6EH	CAN1 message control register 27	C1MCTRL27				√	00x00000 000xx000B

(33/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECA80H	CAN1 message data byte 01 register 28	C1MDATA0128	R/W			√	Undefined
03FECA80H	CAN1 message data byte 0 register 28	C1MDATA028			√		Undefined
03FECA81H	CAN1 message data byte 1 register 28	C1MDATA128			√		Undefined
03FECA82H	CAN1 message data byte 23 register 28	C1MDATA2328				√	Undefined
03FECA82H	CAN1 message data byte 2 register 28	C1MDATA228			√		Undefined
03FECA83H	CAN1 message data byte 3 register 28	C1MDATA328			√		Undefined
03FECA84H	CAN1 message data byte 45 register 28	C1MDATA4528				√	Undefined
03FECA84H	CAN1 message data byte 4 register 28	C1MDATA428			√		Undefined
03FECA85H	CAN1 message data byte 5 register 28	C1MDATA528			√		Undefined
03FECA86H	CAN1 message data byte 67 register 28	C1MDATA6728				√	Undefined
03FECA86H	CAN1 message data byte 6 register 28	C1MDATA628			√		Undefined
03FECA87H	CAN1 message data byte 7 register 28	C1MDATA728			√		Undefined
03FECA88H	CAN1 message data length code register 28	C1MDLC28			√		0000xxxxB
03FECA89H	CAN1 message configuration register 28	C1MCONF28			√		Undefined
03FECA8AH	CAN1 message ID register 28	C1MIDL28				√	Undefined
03FECA8CH		C1MIDH28				√	Undefined
03FECA8EH	CAN1 message control register 28	C1MCTRL28				√	00x00000 000xx000B
03FECAA0H	CAN1 message data byte 01 register 29	C1MDATA0129				√	Undefined
03FECAA0H	CAN1 message data byte 0 register 29	C1MDATA029			√		Undefined
03FECAA1H	CAN1 message data byte 1 register 29	C1MDATA129			√		Undefined
03FECAA2H	CAN1 message data byte 23 register 29	C1MDATA2329				√	Undefined
03FECAA2H	CAN1 message data byte 2 register 29	C1MDATA229			√		Undefined
03FECAA3H	CAN1 message data byte 3 register 29	C1MDATA329			√		Undefined
03FECAA4H	CAN1 message data byte 45 register 29	C1MDATA4529				√	Undefined
03FECAA4H	CAN1 message data byte 4 register 29	C1MDATA429			√		Undefined
03FECAA5H	CAN1 message data byte 5 register 29	C1MDATA529			√		Undefined
03FECAA6H	CAN1 message data byte 67 register 29	C1MDATA6729				√	Undefined
03FECAA6H	CAN1 message data byte 6 register 29	C1MDATA629			√		Undefined
03FECAA7H	CAN1 message data byte 7 register 29	C1MDATA729			√		Undefined
03FECAA8H	CAN1 message data length code register 29	C1MDLC29			√		0000xxxxB
03FECAA9H	CAN1 message configuration register 29	C1MCONF29			√		Undefined
03FECAAAH	CAN1 message ID register 29	C1MIDL29				√	Undefined
03FECAACH		C1MIDH29				√	Undefined
03FECAAEH	CAN1 message control register 29	C1MCTRL29				√	00x00000 000xx000B

(34/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECAC0H	CAN1 message data byte 01 register 30	C1MDATA0130	R/W			√	Undefined
03FECAC0H	CAN1 message data byte 0 register 30	C1MDATA030			√		Undefined
03FECAC1H	CAN1 message data byte 1 register 30	C1MDATA130			√		Undefined
03FECAC2H	CAN1 message data byte 23 register 30	C1MDATA2330				√	Undefined
03FECAC2H	CAN1 message data byte 2 register 30	C1MDATA230			√		Undefined
03FECAC3H	CAN1 message data byte 3 register 30	C1MDATA330			√		Undefined
03FECAC4H	CAN1 message data byte 45 register 30	C1MDATA4530				√	Undefined
03FECAC4H	CAN1 message data byte 4 register 30	C1MDATA430			√		Undefined
03FECAC5H	CAN1 message data byte 5 register 30	C1MDATA530			√		Undefined
03FECAC6H	CAN1 message data byte 67 register 30	C1MDATA6730				√	Undefined
03FECAC6H	CAN1 message data byte 6 register 30	C1MDATA630			√		Undefined
03FECAC7H	CAN1 message data byte 7 register 30	C1MDATA730			√		Undefined
03FECAC8H	CAN1 message data length code register 30	C1MDLC30			√		0000xxxxB
03FECAC9H	CAN1 message configuration register 30	C1MCONF30			√		Undefined
03FECACAH	CAN1 message ID register 30	C1MIDL30				√	Undefined
03FECACCH		C1MIDH30				√	Undefined
03FECACEH	CAN1 message control register 30	C1MCTRL30				√	00x00000 000xx000B
03FECAE0H	CAN1 message data byte 01 register 31	C1MDATA0131				√	Undefined
03FECAE0H	CAN1 message data byte 0 register 31	C1MDATA031			√		Undefined
03FECAE1H	CAN1 message data byte 1 register 31	C1MDATA131			√		Undefined
03FECAE2H	CAN1 message data byte 23 register 31	C1MDATA2331				√	Undefined
03FECAE2H	CAN1 message data byte 2 register 31	C1MDATA231			√		Undefined
03FECAE3H	CAN1 message data byte 3 register 31	C1MDATA331			√		Undefined
03FECAE4H	CAN1 message data byte 45 register 31	C1MDATA4531				√	Undefined
03FECAE4H	CAN1 message data byte 4 register 31	C1MDATA431			√		Undefined
03FECAE5H	CAN1 message data byte 5 register 31	C1MDATA531			√		Undefined
03FECAE6H	CAN1 message data byte 67 register 31	C1MDATA6731				√	Undefined
03FECAE6H	CAN1 message data byte 6 register 31	C1MDATA631			√		Undefined
03FECAE7H	CAN1 message data byte 7 register 31	C1MDATA731			√		Undefined
03FECAE8H	CAN1 message data length code register 31	C1MDLC31			√		0000xxxx
03FECAE9H	CAN1 message configuration register 31	C1MCONF31			√		Undefined
03FECAEAH	CAN1 message ID register 31	C1MIDL31				√	Undefined
03FECAECH		C1MIDH31				√	Undefined
03FECAEEH	CAN1 message control register 31	C1MCTRL31				√	00x00000 000xx000B

(35/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECC00H	CAN2 global control register	C2GMCTRL	R/W	–	–	√	0000H
03FECC02H	CAN2 global clock select register	C2GMCS	R/W	–	√	–	0FH
03FECC06H	CAN2 global block transmission control register	C2GMABT	R/W	–	–	√	0000H
03FECC08H	CAN2 global block transmission delay setting register	C2GMABTD	R/W	–	√	–	00H
03FECC40H	CAN2 module mask 1 register	C2MASK1L	R/W	–	–	√	Undefined
03FECC42H		C2MASK1H					
03FECC44H	CAN2 module mask 2 register	C2MASK2L	R/W	–	–	√	Undefined
03FECC46H		C2MASK2H					
03FECC48H	CAN2 module mask 3 register	C2MASK3L	R/W	–	–	√	Undefined
03FECC4AH		C2MASK3H					
03FECC4CH	CAN2 module mask 4 register	C2MASK4L	R/W	–	–	√	Undefined
03FECC4EH		C2MASK4H					
03FECC50H	CAN2 module control register	C2CTRL	R/W	–	–	√	0000H
03FECC52H	CAN2 module last error information register	C2LEC	R/W	–	√	–	00H
03FECC53H	CAN2 module information register	C2INFO	R	–	√	–	00H
03FECC54H	CAN2 module error counter register	C2ERC	R	–	–	√	0000H
03FECC56H	CAN2 module interrupt enable register	C2IE	R/W	–	–	√	0000H
03FECC58H	CAN2 module interrupt status register	C2INTS	R/W	–	–	√	0000H
03FECC5AH	CAN2 module bit rate prescaler register	C2BRP	R/W	–	√	–	FFH
03FECC5CH	CAN2 module bit rate register	C2BTR	R/W	–	–	√	370FH
03FECC5EH	CAN2 module last in-pointer register	C2LIPT	R	–	√	–	Undefined
03FECC60H	CAN2 module receive history list register	C2RGPT	R/W	–	–	√	xx02H
03FECC62H	CAN2 module last out-pointer register	C2LOPT	R	–	√	–	Undefined
03FECC64H	CAN2 module transmit history list register	C2TGPT	R/W	–	–	√	xx02H
03FECC66H	CAN2 module time stamp register	C2TS	R/W	–	–	√	0000H

(36/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECD00H	CAN2 message data byte 01 register 00	C2MDATA0100	R/W			√	Undefined
03FECD00H	CAN2 message data byte 0 register 00	C2MDATA000			√		Undefined
03FECD01H	CAN2 message data byte 1 register 00	C2MDATA100			√		Undefined
03FECD02H	CAN2 message data byte 23 register 00	C2MDATA2300				√	Undefined
03FECD02H	CAN2 message data byte 2 register 00	C2MDATA200			√		Undefined
03FECD03H	CAN2 message data byte 3 register 00	C2MDATA300			√		Undefined
03FECD04H	CAN2 message data byte 45 register 00	C2MDATA4500				√	Undefined
03FECD04H	CAN2 message data byte 4 register 00	C2MDATA400			√		Undefined
03FECD05H	CAN2 message data byte 5 register 00	C2MDATA500			√		Undefined
03FECD06H	CAN2 message data byte 67 register 00	C2MDATA6700				√	Undefined
03FECD06H	CAN2 message data byte 6 register 00	C2MDATA600			√		Undefined
03FECD07H	CAN2 message data byte 7 register 00	C2MDATA700			√		Undefined
03FECD08H	CAN2 message data length code register 00	C2MDLC00			√		0000xxxxB
03FECD09H	CAN2 message configuration register 00	C2MCONF00			√		Undefined
03FECD0AH	CAN2 message ID register 00	C2MIDL00				√	Undefined
03FECD0CH		C2MIDH00				√	Undefined
03FECD0EH	CAN2 message control register 00	C2MCTRL00				√	00x00000 000xx000B
03FECD20H	CAN2 message data byte 01 register 01	C2MDATA0101				√	Undefined
03FECD20H	CAN2 message data byte 0 register 01	C2MDATA001			√		Undefined
03FECD21H	CAN2 message data byte 1 register 01	C2MDATA101			√		Undefined
03FECD22H	CAN2 message data byte 23 register 01	C2MDATA2301				√	Undefined
03FECD22H	CAN2 message data byte 2 register 01	C2MDATA201			√		Undefined
03FECD23H	CAN2 message data byte 3 register 01	C2MDATA301			√		Undefined
03FECD24H	CAN2 message data byte 45 register 01	C2MDATA4501				√	Undefined
03FECD24H	CAN2 message data byte 4 register 01	C2MDATA401			√		Undefined
03FECD25H	CAN2 message data byte 5 register 01	C2MDATA501			√		Undefined
03FECD26H	CAN2 message data byte 67 register 01	C2MDATA6701				√	Undefined
03FECD26H	CAN2 message data byte 6 register 01	C2MDATA601			√		Undefined
03FECD27H	CAN2 message data byte 7 register 01	C2MDATA701			√		Undefined
03FECD28H	CAN2 message data length code register 01	C2MDLC01			√		0000xxxxB
03FECD29H	CAN2 message register 01	C2MCONF01			√		Undefined
03FECD2AH	CAN2 message ID register 01	C2MIDL01				√	Undefined
03FECD2CH		C2MIDH01				√	Undefined
03FECD2EH	CAN2 message control register 01	C2MCTRL01				√	00x00000 000xx000B

(37/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECD40H	CAN2 message data byte 01 register 02	C2MDATA0102	R/W			√	Undefined
03FECD40H	CAN2 message data byte 0 register 02	C2MDATA002			√		Undefined
03FECD41H	CAN2 message data byte 1 register 02	C2MDATA102			√		Undefined
03FECD42H	CAN2 message data byte 23 register 02	C2MDATA2302				√	Undefined
03FECD42H	CAN2 message data byte 2 register 02	C2MDATA202			√		Undefined
03FECD43H	CAN2 message data byte 3 register 02	C2MDATA302			√		Undefined
03FECD44H	CAN2 message data byte 45 register 02	C2MDATA4502				√	Undefined
03FECD44H	CAN2 message data byte 4 register 02	C2MDATA402			√		Undefined
03FECD45H	CAN2 message data byte 5 register 02	C2MDATA502			√		Undefined
03FECD46H	CAN2 message data byte 67 register 02	C2MDATA6702				√	Undefined
03FECD46H	CAN2 message data byte 6 register 02	C2MDATA602			√		Undefined
03FECD47H	CAN2 message data byte 7 register 02	C2MDATA702			√		Undefined
03FECD48H	CAN2 message data length code register 02	C2MDLC02			√		0000xxxxB
03FECD49H	CAN2 message configuration register 02	C2MCONF02			√		Undefined
03FECD4AH	CAN2 message ID register 02	C2MIDL02				√	Undefined
03FECD4CH		C2MIDH02				√	Undefined
03FECD4EH	CAN2 message control register 02	C2MCTRL02				√	00x00000 000xx000B
03FECD60H	CAN2 message data byte 01 register 03	C2MDATA0103				√	Undefined
03FECD60H	CAN2 message data byte 0 register 03	C2MDATA003			√		Undefined
03FECD61H	CAN2 message data byte 1 register 03	C2MDATA103			√		Undefined
03FECD62H	CAN2 message data byte 23 register 03	C2MDATA2303				√	Undefined
03FECD62H	CAN2 message data byte 2 register 03	C2MDATA203			√		Undefined
03FECD63H	CAN2 message data byte 3 register 03	C2MDATA303			√		Undefined
03FECD64H	CAN2 message data byte 45 register 03	C2MDATA4503				√	Undefined
03FECD64H	CAN2 message data byte 4 register 03	C2MDATA403			√		Undefined
03FECD65H	CAN2 message data byte 5 register 03	C2MDATA503			√		Undefined
03FECD66H	CAN2 message data byte 67 register 03	C2MDATA6703				√	Undefined
03FECD66H	CAN2 message data byte 6 register 03	C2MDATA603			√		Undefined
03FECD67H	CAN2 message data byte 7 register 03	C2MDATA703			√		Undefined
03FECD68H	CAN2 message data length code register 03	C2MDLC03			√		0000xxxxB
03FECD69H	CAN2 message configuration register 03	C2MCONF03			√		Undefined
03FECD6AH	CAN2 message ID register 03	C2MIDL03				√	Undefined
03FECD6CH		C2MIDH03				√	Undefined
03FECD6EH	CAN2 message control register 03	C2MCTRL03				√	00x00000 000xx000B

(38/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECD80H	CAN2 message data byte 01 register 04	C2MDATA0104	R/W			√	Undefined
03FECD80H	CAN2 message data byte 0 register 04	C2MDATA004			√		Undefined
03FECD81H	CAN2 message data byte 1 register 04	C2MDATA104			√		Undefined
03FECD82H	CAN2 message data byte 23 register 04	C2MDATA2304				√	Undefined
03FECD82H	CAN2 message data byte 2 register 04	C2MDATA204			√		Undefined
03FECD83H	CAN2 message data byte 3 register 04	C2MDATA304			√		Undefined
03FECD84H	CAN2 message data byte 45 register 04	C2MDATA4504				√	Undefined
03FECD84H	CAN2 message data byte 4 register 04	C2MDATA404			√		Undefined
03FECD85H	CAN2 message data byte 5 register 04	C2MDATA504			√		Undefined
03FECD86H	CAN2 message data byte 67 register 04	C2MDATA6704				√	Undefined
03FECD86H	CAN2 message data byte 6 register 04	C2MDATA604			√		Undefined
03FECD87H	CAN2 message data byte 7 register 04	C2MDATA704			√		Undefined
03FECD88H	CAN2 message data length code register 04	C2MDLC04			√		0000xxxxB
03FECD89H	CAN2 message configuration register 04	C2MCONF04			√		Undefined
03FECD8AH	CAN2 message ID register 04	C2MIDL04				√	Undefined
03FECD8CH		C2MIDH04				√	Undefined
03FECD8EH	CAN2 message control register 04	C2MCTRL04				√	00x00000 000xx000B
03FECDA0H	CAN2 message data byte 01 register 05	C2MDATA0105				√	Undefined
03FECDA0H	CAN2 message data byte 0 register 05	C2MDATA005			√		Undefined
03FECDA1H	CAN2 message data byte 1 register 05	C2MDATA105			√		Undefined
03FECDA2H	CAN2 message data byte 23 register 05	C2MDATA2305				√	Undefined
03FECDA2H	CAN2 message data byte 2 register 05	C2MDATA205			√		Undefined
03FECDA3H	CAN2 message data byte 3 register 05	C2MDATA305			√		Undefined
03FECDA4H	CAN2 message data byte 45 register 05	C2MDATA4505				√	Undefined
03FECDA4H	CAN2 message data byte 4 register 05	C2MDATA405			√		Undefined
03FECDA5H	CAN2 message data byte 5 register 05	C2MDATA505			√		Undefined
03FECDA6H	CAN2 message data byte 67 register 05	C2MDATA6705				√	Undefined
03FECDA6H	CAN2 message data byte 6 register 05	C2MDATA605			√		Undefined
03FECDA7H	CAN2 message data byte 7 register 05	C2MDATA705			√		Undefined
03FECDA8H	CAN2 message data length code register 05	C2MDLC05			√		0000xxxxB
03FECDA9H	CAN2 message configuration register 05	C2MCONF05			√		Undefined
03FECDAAH	CAN2 message ID register 05	C2MIDL05				√	Undefined
03FECDACH		C2MIDH05				√	Undefined
03FECDAEH	CAN2 message control register 05	C2MCTRL05				√	00x00000 000xx000B

(39/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECDC0H	CAN2 message data byte 01 register 06	C2MDATA0106	R/W			√	Undefined
03FECDC0H	CAN2 message data byte 0 register 06	C2MDATA006			√		Undefined
03FECDC1H	CAN2 message data byte 1 register 06	C2MDATA106			√		Undefined
03FECDC2H	CAN2 message data byte 23 register 06	C2MDATA2306				√	Undefined
03FECDC2H	CAN2 message data byte 2 register 06	C2MDATA206			√		Undefined
03FECDC3H	CAN2 message data byte 3 register 06	C2MDATA306			√		Undefined
03FECDC4H	CAN2 message data byte 45 register 06	C2MDATA4506				√	Undefined
03FECDC4H	CAN2 message data byte 4 register 06	C2MDATA406			√		Undefined
03FECDC5H	CAN2 message data byte 5 register 06	C2MDATA506			√		Undefined
03FECDC6H	CAN2 message data byte 67 register 06	C2MDATA6706				√	Undefined
03FECDC6H	CAN2 message data byte 6 register 06	C2MDATA606			√		Undefined
03FECDC7H	CAN2 message data byte 7 register 06	C2MDATA706			√		Undefined
03FECDC8H	CAN2 message data length code register 06	C2MDLC06			√		0000xxxxB
03FECDC9H	CAN2 message configuration register 06	C2MCONF06			√		Undefined
03FECDCAH	CAN2 message ID register 06	C2MIDL06				√	Undefined
03FECDCCH		C2MIDH06				√	Undefined
03FECDCEH	CAN2 message control register 06	C2MCTRL06				√	00x00000 000xx000B
03FECDE0H	CAN2 message data byte 01 register 07	C2MDATA0107				√	Undefined
03FECDE0H	CAN2 message data byte 0 register 07	C2MDATA007			√		Undefined
03FECDE1H	CAN2 message data byte 1 register 07	C2MDATA107			√		Undefined
03FECDE2H	CAN2 message data byte 23 register 07	C2MDATA2307				√	Undefined
03FECDE2H	CAN2 message data byte 2 register 07	C2MDATA207			√		Undefined
03FECDE3H	CAN2 message data byte 3 register 07	C2MDATA307			√		Undefined
03FECDE4H	CAN2 message data byte 45 register 07	C2MDATA4507				√	Undefined
03FECDE4H	CAN2 message data byte 4 register 07	C2MDATA407			√		Undefined
03FECDE5H	CAN2 message data byte 5 register 07	C2MDATA507			√		Undefined
03FECDE6H	CAN2 message data byte 67 register 07	C2MDATA6707				√	Undefined
03FECDE6H	CAN2 message data byte 6 register 07	C2MDATA607			√		Undefined
03FECDE7H	CAN2 message data byte 7 register 07	C2MDATA707			√		Undefined
03FECDE8H	CAN2 message data length code register 07	C2MDLC07			√		0000xxxxB
03FECDE9H	CAN2 message configuration register 07	C2MCONF07			√		Undefined
03FECDEAH	CAN2 message ID register 07	C2MIDL07				√	Undefined
03FECDECH		C2MIDH07				√	Undefined
03FECDEEH	CAN2 message control register 07	C2MCTRL07				√	00x00000 000xx000B

(40/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECE00H	CAN2 message data byte 01 register 08	C2MDATA0108	R/W			√	Undefined
03FECE00H	CAN2 message data byte 0 register 08	C2MDATA008			√		Undefined
03FECE01H	CAN2 message data byte 1 register 08	C2MDATA108			√		Undefined
03FECE02H	CAN2 message data byte 23 register 08	C2MDATA2308				√	Undefined
03FECE02H	CAN2 message data byte 2 register 08	C2MDATA208			√		Undefined
03FECE03H	CAN2 message data byte 3 register 08	C2MDATA308			√		Undefined
03FECE04H	CAN2 message data byte 45 register 08	C2MDATA4508				√	Undefined
03FECE04H	CAN2 message data byte 4 register 08	C2MDATA408			√		Undefined
03FECE05H	CAN2 message data byte 5 register 08	C2MDATA508			√		Undefined
03FECE06H	CAN2 message data byte 67 register 08	C2MDATA6708				√	Undefined
03FECE06H	CAN2 message data byte 6 register 08	C2MDATA608			√		Undefined
03FECE07H	CAN2 message data byte 7 register 08	C2MDATA708			√		Undefined
03FECE08H	CAN2 message data length code register 08	C2MDLC08			√		0000xxxxB
03FECE09H	CAN2 message configuration register 08	C2MCONF08			√		Undefined
03FECE0AH	CAN2 message ID register 08	C2MIDL08				√	Undefined
03FECE0CH		C2MIDH08				√	Undefined
03FECE0EH	CAN2 message control register 08	C2MCTRL08				√	00x00000 000xx000B
03FECE20H	CAN2 message data byte 01 register 09	C2MDATA0109				√	Undefined
03FECE20H	CAN2 message data byte 0 register 09	C2MDATA009			√		Undefined
03FECE21H	CAN2 message data byte 1 register 09	C2MDATA109			√		Undefined
03FECE22H	CAN2 message data byte 23 register 09	C2MDATA2309				√	Undefined
03FECE22H	CAN2 message data byte 2 register 09	C2MDATA209			√		Undefined
03FECE23H	CAN2 message data byte 3 register 09	C2MDATA309			√		Undefined
03FECE24H	CAN2 message data byte 45 register 09	C2MDATA4509				√	Undefined
03FECE24H	CAN2 message data byte 4 register 09	C2MDATA409			√		Undefined
03FECE25H	CAN2 message data byte 5 register 09	C2MDATA509			√		Undefined
03FECE26H	CAN2 message data byte 67 register 09	C2MDATA6709				√	Undefined
03FECE26H	CAN2 message data byte 6 register 09	C2MDATA609			√		Undefined
03FECE27H	CAN2 message data byte 7 register 09	C2MDATA709			√		Undefined
03FECE28H	CAN2 message data length code register 09	C2MDLC09			√		0000xxxxB
03FECE29H	CAN2 message configuration register 09	C2MCONF09			√		Undefined
03FECE2AH	CAN2 message ID register 09	C2MIDL09				√	Undefined
03FECE2CH		C2MIDH09				√	Undefined
03FECE2EH	CAN2 message control register 09	C2MCTRL09				√	00x00000 000xx000B

(41/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECE40H	CAN2 message data byte 01 register 10	C2MDATA0110	R/W			√	Undefined
03FECE40H	CAN2 message data byte 0 register 10	C2MDATA010			√		Undefined
03FECE41H	CAN2 message data byte 1 register 10	C2MDATA110			√		Undefined
03FECE42H	CAN2 message data byte 23 register 10	C2MDATA2310				√	Undefined
03FECE42H	CAN2 message data byte 2 register 10	C2MDATA210			√		Undefined
03FECE43H	CAN2 message data byte 3 register 10	C2MDATA310			√		Undefined
03FECE44H	CAN2 message data byte 45 register 10	C2MDATA4510				√	Undefined
03FECE44H	CAN2 message data byte 4 register 10	C2MDATA410			√		Undefined
03FECE45H	CAN2 message data byte 5 register 10	C2MDATA510			√		Undefined
03FECE46H	CAN2 message data byte 67 register 10	C2MDATA6710				√	Undefined
03FECE46H	CAN2 message data byte 6 register 10	C2MDATA610			√		Undefined
03FECE47H	CAN2 message data byte 7 register 10	C2MDATA710			√		Undefined
03FECE48H	CAN2 message data length code register 10	C2MDLC10			√		0000xxxxB
03FECE49H	CAN2 message configuration register 10	C2MCONF10			√		Undefined
03FECE4AH	CAN2 message ID register 10	C2MIDL10				√	Undefined
03FECE4CH		C2MIDH10				√	Undefined
03FECE4EH	CAN2 message control register 10	C2MCTRL10				√	00x00000 000xx000B
03FECE60H	CAN2 message data byte 01 register 11	C2MDATA0111				√	Undefined
03FECE60H	CAN2 message data byte 0 register 11	C2MDATA011			√		Undefined
03FECE61H	CAN2 message data byte 1 register 11	C2MDATA111			√		Undefined
03FECE62H	CAN2 message data byte 23 register 11	C2MDATA2311				√	Undefined
03FECE62H	CAN2 message data byte 2 register 11	C2MDATA211			√		Undefined
03FECE63H	CAN2 message data byte 3 register 11	C2MDATA311			√		Undefined
03FECE64H	CAN2 message data byte 45 register 11	C2MDATA4511				√	Undefined
03FECE64H	CAN2 message data byte 4 register 11	C2MDATA411			√		Undefined
03FECE65H	CAN2 message data byte 5 register 11	C2MDATA511			√		Undefined
03FECE66H	CAN2 message data byte 67 register 11	C2MDATA6711				√	Undefined
03FECE66H	CAN2 message data byte 6 register 11	C2MDATA611			√		Undefined
03FECE67H	CAN2 message data byte 7 register 11	C2MDATA711			√		Undefined
03FECE68H	CAN2 message data length code register 11	C2MDLC11			√		0000xxxxB
03FECE69H	CAN2 message configuration register 11	C2MCONF11			√		Undefined
03FECE6AH	CAN2 message ID register 11	C2MIDL11				√	Undefined
03FECE6CH		C2MIDH11				√	Undefined
03FECE6EH	CAN2 message control register 11	C2MCTRL11				√	00x00000 000xx000B

(42/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECE80H	CAN2 message data byte 01 register 12	C2MDATA0112	R/W			√	Undefined
03FECE80H	CAN2 message data byte 0 register 12	C2MDATA012			√		Undefined
03FECE81H	CAN2 message data byte 1 register 12	C2MDATA112			√		Undefined
03FECE82H	CAN2 message data byte 23 register 12	C2MDATA2312				√	Undefined
03FECE82H	CAN2 message data byte 2 register 12	C2MDATA212			√		Undefined
03FECE83H	CAN2 message data byte 3 register 12	C2MDATA312			√		Undefined
03FECE84H	CAN2 message data byte 45 register 12	C2MDATA4512				√	Undefined
03FECE84H	CAN2 message data byte 4 register 12	C2MDATA412			√		Undefined
03FECE85H	CAN2 message data byte 5 register 12	C2MDATA512			√		Undefined
03FECE86H	CAN2 message data byte 67 register 12	C2MDATA6712				√	Undefined
03FECE86H	CAN2 message data byte 6 register 12	C2MDATA612			√		Undefined
03FECE87H	CAN2 message data byte 7 register 12	C2MDATA712			√		Undefined
03FECE88H	CAN2 message data length code register 12	C2MDLC12			√		0000xxxxB
03FECE89H	CAN2 message configuration register 12	C2MCONF12			√		Undefined
03FECE8AH	CAN2 message ID register 12	C2MIDL12				√	Undefined
03FECE8CH		C2MIDH12				√	Undefined
03FECE8EH	CAN2 message control register 12	C2MCTRL12				√	00x00000 000xx000B
03FECEA0H	CAN2 message data byte 01 register 13	C2MDATA0113				√	Undefined
03FECEA0H	CAN2 message data byte 0 register 13	C2MDATA013			√		Undefined
03FECEA1H	CAN2 message data byte 1 register 13	C2MDATA113			√		Undefined
03FECEA2H	CAN2 message data byte 23 register 13	C2MDATA2313				√	Undefined
03FECEA2H	CAN2 message data byte 2 register 13	C2MDATA213			√		Undefined
03FECEA3H	CAN2 message data byte 3 register 13	C2MDATA313			√		Undefined
03FECEA4H	CAN2 message data byte 45 register 13	C2MDATA4513				√	Undefined
03FECEA4H	CAN2 message data byte 4 register 13	C2MDATA413			√		Undefined
03FECEA5H	CAN2 message data byte 5 register 13	C2MDATA513			√		Undefined
03FECEA6H	CAN2 message data byte 67 register 13	C2MDATA6713				√	Undefined
03FECEA6H	CAN2 message data byte 6 register 13	C2MDATA613			√		Undefined
03FECEA7H	CAN2 message data byte 7 register 13	C2MDATA713			√		Undefined
03FECEE8H	CAN2 message data length code register 13	C2MDLC13			√		0000xxxxB
03FECEA9H	CAN2 message configuration register 13	C2MCONF13			√		Undefined
03FECEAAH	CAN2 message ID register 13	C2MIDL13				√	Undefined
03FECEACH		C2MIDH13				√	Undefined
03FECEAEH	CAN2 message control register 13	C2MCTRL13				√	00x00000 000xx000B

(43/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECEC0H	CAN2 message data byte 01 register 14	C2MDATA0114	R/W			√	Undefined
03FECEC0H	CAN2 message data byte 0 register 14	C2MDATA014			√		Undefined
03FECEC1H	CAN2 message data byte 1 register 14	C2MDATA114			√		Undefined
03FECEC2H	CAN2 message data byte 23 register 14	C2MDATA2314				√	Undefined
03FECEC2H	CAN2 message data byte 2 register 14	C2MDATA214			√		Undefined
03FECEC3H	CAN2 message data byte 3 register 14	C2MDATA314			√		Undefined
03FECEC4H	CAN2 message data byte 45 register 14	C2MDATA4514				√	Undefined
03FECEC4H	CAN2 message data byte 4 register 14	C2MDATA414			√		Undefined
03FECEC5H	CAN2 message data byte 5 register 14	C2MDATA514			√		Undefined
03FECEC6H	CAN2 message data byte 67 register 14	C2MDATA6714				√	Undefined
03FECEC6H	CAN2 message data byte 6 register 14	C2MDATA614			√		Undefined
03FECEC7H	CAN2 message data byte 7 register 14	C2MDATA714			√		Undefined
03FECEC8H	CAN2 message data length code register 14	C2MDLC14			√		0000xxxxB
03FECEC9H	CAN2 message configuration register 14	C2MCONF14			√		Undefined
03FECECAH	CAN2 message ID register 14	C2MIDL14				√	Undefined
03FECECCH		C2MIDH14				√	Undefined
03FECECEH	CAN2 message control register 14	C2MCTRL14				√	00x00000 000xx000B
03FECEE0H	CAN2 message data byte 01 register 15	C2MDATA0115				√	Undefined
03FECEE0H	CAN2 message data byte 0 register 15	C2MDATA015			√		Undefined
03FECEE1H	CAN2 message data byte 1 register 15	C2MDATA115			√		Undefined
03FECEE2H	CAN2 message data byte 23 register 15	C2MDATA2315				√	Undefined
03FECEE2H	CAN2 message data byte 2 register 15	C2MDATA215			√		Undefined
03FECEE3H	CAN2 message data byte 3 register 15	C2MDATA315			√		Undefined
03FECEE4H	CAN2 message data byte 45 register 15	C2MDATA4515				√	Undefined
03FECEE4H	CAN2 message data byte 4 register 15	C2MDATA415			√		Undefined
03FECEE5H	CAN2 message data byte 5 register 15	C2MDATA515			√		Undefined
03FECEE6H	CAN2 message data byte 67 register 15	C2MDATA6715				√	Undefined
03FECEE6H	CAN2 message data byte 6 register 15	C2MDATA615			√		Undefined
03FECEE7H	CAN2 message data byte 7 register 15	C2MDATA715			√		Undefined
03FECEE8H	CAN2 message data length code register 15	C2MDLC15			√		0000xxxxB
03FECEE9H	CAN2 message configuration register 15	C2MCONF15			√		Undefined
03FECEEAH	CAN2 message ID register 15	C2MIDL15				√	Undefined
03FECEECH		C2MIDH15				√	Undefined
03FECEEEH	CAN2 message control register 15	C2MCTRL15				√	00x00000 000xx000B

(44/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECF00H	CAN2 message data byte 01 register 16	C2MDATA0116	R/W			√	Undefined
03FECF00H	CAN2 message data byte 0 register 16	C2MDATA016			√		Undefined
03FECF01H	CAN2 message data byte 1 register 16	C2MDATA116			√		Undefined
03FECF02H	CAN2 message data byte 23 register 16	C2MDATA2316				√	Undefined
03FECF02H	CAN2 message data byte 2 register 16	C2MDATA216			√		Undefined
03FECF03H	CAN2 message data byte 3 register 16	C2MDATA316			√		Undefined
03FECF04H	CAN2 message data byte 45 register 16	C2MDATA4516				√	Undefined
03FECF04H	CAN2 message data byte 4 register 16	C2MDATA416			√		Undefined
03FECF05H	CAN2 message data byte 5 register 16	C2MDATA516			√		Undefined
03FECF06H	CAN2 message data byte 67 register 16	C2MDATA6716				√	Undefined
03FECF06H	CAN2 message data byte 6 register 16	C2MDATA616			√		Undefined
03FECF07H	CAN2 message data byte 7 register 16	C2MDATA716			√		Undefined
03FECF08H	CAN2 message data length code register 16	C2MDLC16			√		0000xxxxB
03FECF09H	CAN2 message configuration register 16	C2MCONF16			√		Undefined
03FECF0AH	CAN2 message ID register 16	C2MIDL16				√	Undefined
03FECF0CH		C2MIDH16				√	Undefined
03FECF0EH	CAN2 message control register 16	C2MCTRL16				√	00x00000 000xx000B
03FECF20H	CAN2 message data byte 01 register 17	C2MDATA0117				√	Undefined
03FECF20H	CAN2 message data byte 0 register 17	C2MDATA017			√		Undefined
03FECF21H	CAN2 message data byte 1 register 17	C2MDATA117			√		Undefined
03FECF22H	CAN2 message data byte 23 register 17	C2MDATA2317				√	Undefined
03FECF22H	CAN2 message data byte 2 register 17	C2MDATA217			√		Undefined
03FECF23H	CAN2 message data byte 3 register 17	C2MDATA317			√		Undefined
03FECF24H	CAN2 message data byte 45 register 17	C2MDATA4517				√	Undefined
03FECF24H	CAN2 message data byte 4 register 17	C2MDATA417			√		Undefined
03FECF25H	CAN2 message data byte 5 register 17	C2MDATA517			√		Undefined
03FECF26H	CAN2 message data byte 67 register 17	C2MDATA6717				√	Undefined
03FECF26H	CAN2 message data byte 6 register 17	C2MDATA617			√		Undefined
03FECF27H	CAN2 message data byte 7 register 17	C2MDATA717			√		Undefined
03FECF28H	CAN2 message data length code register 17	C2MDLC17			√		0000xxxxB
03FECF29H	CAN2 message configuration register 17	C2MCONF17			√		Undefined
03FECF2AH	CAN2 message ID register 17	C2MIDL17				√	Undefined
03FECF2CH		C2MIDH17				√	Undefined
03FECF2EH	CAN2 message control register 17	C2MCTRL17				√	00x00000 000xx000B

(45/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECF40H	CAN2 message data byte 01 register 18	C2MDATA0118	R/W			√	Undefined
03FECF40H	CAN2 message data byte 0 register 18	C2MDATA018			√		Undefined
03FECF41H	CAN2 message data byte 1 register 18	C2MDATA118			√		Undefined
03FECF42H	CAN2 message data byte 23 register 18	C2MDATA2318				√	Undefined
03FECF42H	CAN2 message data byte 2 register 18	C2MDATA218			√		Undefined
03FECF43H	CAN2 message data byte 3 register 18	C2MDATA318			√		Undefined
03FECF44H	CAN2 message data byte 45 register 18	C2MDATA4518				√	Undefined
03FECF44H	CAN2 message data byte 4 register 18	C2MDATA418			√		Undefined
03FECF45H	CAN2 message data byte 5 register 18	C2MDATA518			√		Undefined
03FECF46H	CAN2 message data byte 67 register 18	C2MDATA6718				√	Undefined
03FECF46H	CAN2 message data byte 6 register 18	C2MDATA618			√		Undefined
03FECF47H	CAN2 message data byte 7 register 18	C2MDATA718			√		Undefined
03FECF48H	CAN2 message data length code register 18	C2MDLC18			√		0000xxxxB
03FECF49H	CAN2 message configuration register 18	C2MCONF18			√		Undefined
03FECF4AH	CAN2 message ID register 18	C2MIDL18				√	Undefined
03FECF4CH		C2MIDH18				√	Undefined
03FECF4EH	CAN2 message control register 18	C2MCTRL18				√	00x00000 000xx000B
03FECF60H	CAN2 message data byte 01 register 19	C2MDATA0119				√	Undefined
03FECF60H	CAN2 message data byte 0 register 19	C2MDATA019			√		Undefined
03FECF61H	CAN2 message data byte 1 register 19	C2MDATA119			√		Undefined
03FECF62H	CAN2 message data byte 23 register 19	C2MDATA2319				√	Undefined
03FECF62H	CAN2 message data byte 2 register 19	C2MDATA219			√		Undefined
03FECF63H	CAN2 message data byte 3 register 19	C2MDATA319			√		Undefined
03FECF64H	CAN2 message data byte 45 register 19	C2MDATA4519				√	Undefined
03FECF64H	CAN2 message data byte 4 register 19	C2MDATA419			√		Undefined
03FECF65H	CAN2 message data byte 5 register 19	C2MDATA519			√		Undefined
03FECF66H	CAN2 message data byte 67 register 19	C2MDATA6719				√	Undefined
03FECF66H	CAN2 message data byte 6 register 19	C2MDATA619			√		Undefined
03FECF67H	CAN2 message data byte 7 register 19	C2MDATA719			√		Undefined
03FECF68H	CAN2 message data length code register 19	C2MDLC19			√		0000xxxxB
03FECF69H	CAN2 message configuration register 19	C2MCONF19			√		Undefined
03FECF6AH	CAN2 message ID register 19	C2MIDL19				√	Undefined
03FECF6CH		C2MIDH19				√	Undefined
03FECF6EH	CAN2 message control register 19	C2MCTRL19				√	00x00000 000xx000B

(46/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECF80H	CAN2 message data byte 01 register 20	C2MDATA0120	R/W			√	Undefined
03FECF80H	CAN2 message data byte 0 register 20	C2MDATA020			√		Undefined
03FECF81H	CAN2 message data byte 1 register 20	C2MDATA120			√		Undefined
03FECF82H	CAN2 message data byte 23 register 20	C2MDATA2320				√	Undefined
03FECF82H	CAN2 message data byte 2 register 20	C2MDATA220			√		Undefined
03FECF83H	CAN2 message data byte 3 register 20	C2MDATA320			√		Undefined
03FECF84H	CAN2 message data byte 45 register 20	C2MDATA4520				√	Undefined
03FECF84H	CAN2 message data byte 4 register 20	C2MDATA420			√		Undefined
03FECF85H	CAN2 message data byte 5 register 20	C2MDATA520			√		Undefined
03FECF86H	CAN2 message data byte 67 register 20	C2MDATA6720				√	Undefined
03FECF86H	CAN2 message data byte 6 register 20	C2MDATA620			√		Undefined
03FECF87H	CAN2 message data byte 7 register 20	C2MDATA720			√		Undefined
03FECF88H	CAN2 message data length code register 20	C2MDLC20			√		0000xxxxB
03FECF89H	CAN2 message configuration register 20	C2MCONF20			√		Undefined
03FECF8AH	CAN2 message ID register 20	C2MIDL20				√	Undefined
03FECF8CH		C2MIDH20				√	Undefined
03FECF8EH	CAN2 message control register 20	C2MCTRL20				√	00x00000 000xx000B
03FECFA0H	CAN2 message data byte 01 register 21	C2MDATA0121				√	Undefined
03FECFA0H	CAN2 message data byte 0 register 21	C2MDATA021			√		Undefined
03FECFA1H	CAN2 message data byte 1 register 21	C2MDATA121			√		Undefined
03FECFA2H	CAN2 message data byte 23 register 21	C2MDATA2321				√	Undefined
03FECFA2H	CAN2 message data byte 2 register 21	C2MDATA221			√		Undefined
03FECFA3H	CAN2 message data byte 3 register 21	C2MDATA321			√		Undefined
03FECFA4H	CAN2 message data byte 45 register 21	C2MDATA4521				√	Undefined
03FECFA4H	CAN2 message data byte 4 register 21	C2MDATA421			√		Undefined
03FECFA5H	CAN2 message data byte 5 register 21	C2MDATA521			√		Undefined
03FECFA6H	CAN2 message data byte 67 register 21	C2MDATA6721				√	Undefined
03FECFA6H	CAN2 message data byte 6 register 21	C2MDATA621			√		Undefined
03FECFA7H	CAN2 message data byte 7 register 21	C2MDATA721			√		Undefined
03FECFA8H	CAN2 message data length code register 21	C2MDLC21			√		0000xxxxB
03FECFA9H	CAN2 message configuration register 21	C2MCONF21			√		Undefined
03FECFAAH	CAN2 message ID register 21	C2MIDL21				√	Undefined
03FECFACH		C2MIDH21				√	Undefined
03FECFAEH	CAN2 message control register 21	C2MCTRL21				√	00x00000 000xx000B

(47/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FECFC0H	CAN2 message data byte 01 register 22	C2MDATA0122	R/W			√	Undefined
03FECFC0H	CAN2 message data byte 0 register 22	C2MDATA022			√		Undefined
03FECFC1H	CAN2 message data byte 1 register 22	C2MDATA122			√		Undefined
03FECFC2H	CAN2 message data byte 23 register 22	C2MDATA2322				√	Undefined
03FECFC2H	CAN2 message data byte 2 register 22	C2MDATA222			√		Undefined
03FECFC3H	CAN2 message data byte 3 register 22	C2MDATA322			√		Undefined
03FECFC4H	CAN2 message data byte 45 register 22	C2MDATA4522				√	Undefined
03FECFC4H	CAN2 message data byte 4 register 22	C2MDATA422			√		Undefined
03FECFC5H	CAN2 message data byte 5 register 22	C2MDATA522			√		Undefined
03FECFC6H	CAN2 message data byte 67 register 22	C2MDATA6722				√	Undefined
03FECFC6H	CAN2 message data byte 6 register 22	C2MDATA622			√		Undefined
03FECFC7H	CAN2 message data byte 7 register 22	C2MDATA722			√		Undefined
03FECFC8H	CAN2 message data length code register 22	C2MDLC22			√		0000xxxxB
03FECFC9H	CAN2 message configuration register 22	C2MCONF22			√		Undefined
03FECFCAH	CAN2 message ID register 22	C2MIDL22				√	Undefined
03FECFCCH		C2MIDH22				√	Undefined
03FECFCEH	CAN2 message control register 22	C2MCTRL22				√	00x00000 000xx000B
03FECFE0H	CAN2 message data byte 01 register 23	C2MDATA0123				√	Undefined
03FECFE0H	CAN2 message data byte 0 register 23	C2MDATA023			√		Undefined
03FECFE1H	CAN2 message data byte 1 register 23	C2MDATA123			√		Undefined
03FECFE2H	CAN2 message data byte 23 register 23	C2MDATA2323				√	Undefined
03FECFE2H	CAN2 message data byte 2 register 23	C2MDATA223			√		Undefined
03FECFE3H	CAN2 message data byte 3 register 23	C2MDATA323			√		Undefined
03FECFE4H	CAN2 message data byte 45 register 23	C2MDATA4523				√	Undefined
03FECFE4H	CAN2 message data byte 4 register 23	C2MDATA423			√		Undefined
03FECFE5H	CAN2 message data byte 5 register 23	C2MDATA523			√		Undefined
03FECFE6H	CAN2 message data byte 67 register 23	C2MDATA6723				√	Undefined
03FECFE6H	CAN2 message data byte 6 register 23	C2MDATA623			√		Undefined
03FECFE7H	CAN2 message data byte 7 register 23	C2MDATA723			√		Undefined
03FECFE8H	CAN2 message data length code register 23	C2MDLC23			√		0000xxxxB
03FECFE9H	CAN2 message configuration register 23	C2MCONF23			√		Undefined
03FECFEAH	CAN2 message ID register 23	C2MIDL23				√	Undefined
03FECFECH		C2MIDH23				√	Undefined
03FECFEEH	CAN2 message control register 23	C2MCTRL23				√	00x00000 000xx000B

(48/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED000H	CAN2 message data byte 01 register 24	C2MDATA0124	R/W			√	Undefined
03FED000H	CAN2 message data byte 0 register 24	C2MDATA024			√		Undefined
03FED001H	CAN2 message data byte 1 register 24	C2MDATA124			√		Undefined
03FED002H	CAN2 message data byte 23 register 24	C2MDATA2324				√	Undefined
03FED002H	CAN2 message data byte 2 register 24	C2MDATA224			√		Undefined
03FED003H	CAN2 message data byte 3 register 24	C2MDATA324			√		Undefined
03FED004H	CAN2 message data byte 45 register 24	C2MDATA4524				√	Undefined
03FED004H	CAN2 message data byte 4 register 24	C2MDATA424			√		Undefined
03FED005H	CAN2 message data byte 5 register 24	C2MDATA524			√		Undefined
03FED006H	CAN2 message data byte 67 register 24	C2MDATA6724				√	Undefined
03FED006H	CAN2 message data byte 6 register 24	C2MDATA624			√		Undefined
03FED007H	CAN2 message data byte 7 register 24	C2MDATA724			√		Undefined
03FED008H	CAN2 message data length code register 24	C2MDLC24			√		0000xxxxB
03FED009H	CAN2 message configuration register 24	C2MCONF24			√		Undefined
03FED00AH	CAN2 message ID register 24	C2MIDL24				√	Undefined
03FED00CH		C2MIDH24				√	Undefined
03FED00EH	CAN2 message control register 24	C2MCTRL24				√	00x00000 000xx000B
03FED020H	CAN2 message data byte 01 register 25	C2MDATA0125				√	Undefined
03FED020H	CAN2 message data byte 0 register 25	C2MDATA025			√		Undefined
03FED021H	CAN2 message data byte 1 register 25	C2MDATA125			√		Undefined
03FED022H	CAN2 message data byte 23 register 25	C2MDATA2325				√	Undefined
03FED022H	CAN2 message data byte 2 register 25	C2MDATA225			√		Undefined
03FED023H	CAN2 message data byte 3 register 25	C2MDATA325			√		Undefined
03FED024H	CAN2 message data byte 45 register 25	C2MDATA4525				√	Undefined
03FED024H	CAN2 message data byte 4 register 25	C2MDATA425			√		Undefined
03FED025H	CAN2 message data byte 5 register 25	C2MDATA525			√		Undefined
03FED026H	CAN2 message data byte 67 register 25	C2MDATA6725				√	Undefined
03FED026H	CAN2 message data byte 6 register 25	C2MDATA625			√		Undefined
03FED027H	CAN2 message data byte 7 register 25	C2MDATA725			√		Undefined
03FED028H	CAN2 message data length code register 25	C2MDLC25			√		0000xxxxB
03FED029H	CAN2 message configuration register 25	C2MCONF25			√		Undefined
03FED02AH	CAN2 message ID register 25	C2MIDL25				√	Undefined
03FED02CH		C2MIDH25				√	Undefined
03FED02EH	CAN2 message control register 25	C2MCTRL25				√	00x00000 000xx000B

(49/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED040H	CAN2 message data byte 01 register 26	C2MDATA0126	R/W			√	Undefined
03FED040H	CAN2 message data byte 0 register 26	C2MDATA026			√		Undefined
03FED041H	CAN2 message data byte 1 register 26	C2MDATA126			√		Undefined
03FED042H	CAN2 message data byte 23 register 26	C2MDATA2326				√	Undefined
03FED042H	CAN2 message data byte 2 register 26	C2MDATA226			√		Undefined
03FED043H	CAN2 message data byte 3 register 26	C2MDATA326			√		Undefined
03FED044H	CAN2 message data byte 45 register 26	C2MDATA4526				√	Undefined
03FED044H	CAN2 message data byte 4 register 26	C2MDATA426			√		Undefined
03FED045H	CAN2 message data byte 5 register 26	C2MDATA526			√		Undefined
03FED046H	CAN2 message data byte 67 register 26	C2MDATA6726				√	Undefined
03FED046H	CAN2 message data byte 6 register 26	C2MDATA626			√		Undefined
03FED047H	CAN2 message data byte 7 register 26	C2MDATA726			√		Undefined
03FED048H	CAN2 message data length code register 26	C2MDLC26			√		0000xxxxB
03FED049H	CAN2 message configuration register 26	C2MCONF26			√		Undefined
03FED04AH	CAN2 message ID register 26	C2MIDL26				√	Undefined
03FED04CH		C2MIDH26				√	Undefined
03FED04EH	CAN2 message control register 26	C2MCTRL26				√	00x00000 000xx000B
03FED060H	CAN2 message data byte 01 register 27	C2MDATA0127				√	Undefined
03FED060H	CAN2 message data byte 0 register 27	C2MDATA027			√		Undefined
03FED061H	CAN2 message data byte 1 register 27	C2MDATA127			√		Undefined
03FED062H	CAN2 message data byte 23 register 27	C2MDATA2327				√	Undefined
03FED062H	CAN2 message data byte 2 register 27	C2MDATA227			√		Undefined
03FED063H	CAN2 message data byte 3 register 27	C2MDATA327			√		Undefined
03FED064H	CAN2 message data byte 45 register 27	C2MDATA4527				√	Undefined
03FED064H	CAN2 message data byte 4 register 27	C2MDATA427			√		Undefined
03FED065H	CAN2 message data byte 5 register 27	C2MDATA527			√		Undefined
03FED066H	CAN2 message data byte 67 register 27	C2MDATA6727				√	Undefined
03FED066H	CAN2 message data byte 6 register 27	C2MDATA627			√		Undefined
03FED067H	CAN2 message data byte 7 register 27	C2MDATA727			√		Undefined
03FED068H	CAN2 message data length code register 27	C2MDLC27			√		0000xxxxB
03FED069H	CAN2 message configuration register 27	C2MCONF27			√		Undefined
03FED06AH	CAN2 message ID register 27	C2MIDL27				√	Undefined
03FED06CH		C2MIDH27				√	Undefined
03FED06EH	CAN2 message control register 27	C2MCTRL27				√	00x00000 000xx000B

(50/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED080H	CAN2 message data byte 01 register 28	C2MDATA0128	R/W			√	Undefined
03FED080H	CAN2 message data byte 0 register 28	C2MDATA028			√		Undefined
03FED081H	CAN2 message data byte 1 register 28	C2MDATA128			√		Undefined
03FED082H	CAN2 message data byte 23 register 28	C2MDATA2328				√	Undefined
03FED082H	CAN2 message data byte 2 register 28	C2MDATA228			√		Undefined
03FED083H	CAN2 message data byte 3 register 28	C2MDATA328			√		Undefined
03FED084H	CAN2 message data byte 45 register 28	C2MDATA4528				√	Undefined
03FED084H	CAN2 message data byte 4 register 28	C2MDATA428			√		Undefined
03FED085H	CAN2 message data byte 5 register 28	C2MDATA528			√		Undefined
03FED086H	CAN2 message data byte 67 register 28	C2MDATA6728				√	Undefined
03FED086H	CAN2 message data byte 6 register 28	C2MDATA628			√		Undefined
03FED087H	CAN2 message data byte 7 register 28	C2MDATA728			√		Undefined
03FED088H	CAN2 message data length code register 28	C2MDLC28			√		0000xxxxB
03FED089H	CAN2 message configuration register 28	C2MCONF28			√		Undefined
03FED08AH	CAN2 message ID register 28	C2MIDL28				√	Undefined
03FED08CH		C2MIDH28				√	Undefined
03FED08EH	CAN2 message control register 28	C2MCTRL28				√	00x00000 000xx000B
03FED0A0H	CAN2 message data byte 01 register 29	C2MDATA0129				√	Undefined
03FED0A0H	CAN2 message data byte 0 register 29	C2MDATA029			√		Undefined
03FED0A1H	CAN2 message data byte 1 register 29	C2MDATA129			√		Undefined
03FED0A2H	CAN2 message data byte 23 register 29	C2MDATA2329				√	Undefined
03FED0A2H	CAN2 message data byte 2 register 29	C2MDATA229			√		Undefined
03FED0A3H	CAN2 message data byte 3 register 29	C2MDATA329			√		Undefined
03FED0A4H	CAN2 message data byte 45 register 29	C2MDATA4529				√	Undefined
03FED0A4H	CAN2 message data byte 4 register 29	C2MDATA429			√		Undefined
03FED0A5H	CAN2 message data byte 5 register 29	C2MDATA529			√		Undefined
03FED0A6H	CAN2 message data byte 67 register 29	C2MDATA6729				√	Undefined
03FED0A6H	CAN2 message data byte 6 register 29	C2MDATA629			√		Undefined
03FED0A7H	CAN2 message data byte 7 register 29	C2MDATA729			√		Undefined
03FED0A8H	CAN2 message data length code register 29	C2MDLC29			√		0000xxxxB
03FED0A9H	CAN2 message configuration register 29	C2MCONF29			√		Undefined
03FED0AAH	CAN2 message ID register 29	C2MIDL29				√	Undefined
03FED0ACH		C2MIDH29				√	Undefined
03FED0AEH	CAN2 message control register 29	C2MCTRL29				√	00x00000 000xx000B

(51/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED0C0H	CAN2 message data byte 01 register 30	C2MDATA0130	R/W			√	Undefined
03FED0C0H	CAN2 message data byte 0 register 30	C2MDATA030			√		Undefined
03FED0C1H	CAN2 message data byte 1 register 30	C2MDATA130			√		Undefined
03FED0C2H	CAN2 message data byte 23 register 30	C2MDATA2330				√	Undefined
03FED0C2H	CAN2 message data byte 2 register 30	C2MDATA230			√		Undefined
03FED0C3H	CAN2 message data byte 3 register 30	C2MDATA330			√		Undefined
03FED0C4H	CAN2 message data byte 45 register 30	C2MDATA4530				√	Undefined
03FED0C4H	CAN2 message data byte 4 register 30	C2MDATA430			√		Undefined
03FED0C5H	CAN2 message data byte 5 register 30	C2MDATA530			√		Undefined
03FED0C6H	CAN2 message data byte 67 register 30	C2MDATA6730				√	Undefined
03FED0C6H	CAN2 message data byte 6 register 30	C2MDATA630			√		Undefined
03FED0C7H	CAN2 message data byte 7 register 30	C2MDATA730			√		Undefined
03FED0C8H	CAN2 message data length code register 30	C2MDLC30			√		0000xxxxB
03FED0C9H	CAN2 message configuration register 30	C2MCONF30			√		Undefined
03FED0CAH	CAN2 message ID register 30	C2MIDL30				√	Undefined
03FED0CCH		C2MIDH30				√	Undefined
03FED0CEH	CAN2 message control register 30	C2MCTRL30				√	00x00000 000xx000B
03FED0E0H	CAN2 message data byte 01 register 31	C2MDATA0131				√	Undefined
03FED0E0H	CAN2 message data byte 0 register 31	C2MDATA031			√		Undefined
03FED0E1H	CAN2 message data byte 1 register 31	C2MDATA131			√		Undefined
03FED0E2H	CAN2 message data byte 23 register 31	C2MDATA2331				√	Undefined
03FED0E2H	CAN2 message data byte 2 register 31	C2MDATA231			√		Undefined
03FED0E3H	CAN2 message data byte 3 register 31	C2MDATA331			√		Undefined
03FED0E4H	CAN2 message data byte 45 register 31	C2MDATA4531				√	Undefined
03FED0E4H	CAN2 message data byte 4 register 31	C2MDATA431			√		Undefined
03FED0E5H	CAN2 message data byte 5 register 31	C2MDATA531			√		Undefined
03FED0E6H	CAN2 message data byte 67 register 31	C2MDATA6731				√	Undefined
03FED0E6H	CAN2 message data byte 6 register 31	C2MDATA631			√		Undefined
03FED0E7H	CAN2 message data byte 7 register 31	C2MDATA731			√		Undefined
03FED0E8H	CAN2 message data length code register 31	C2MDLC31			√		0000xxxx
03FED0E9H	CAN2 message configuration register 31	C2MCONF31			√		Undefined
03FED0EAH	CAN2 message ID register 31	C2MIDL31				√	Undefined
03FED0ECH		C2MIDH31				√	Undefined
03FED0EEH	CAN2 message control register 31	C2MCTRL31				√	00x00000 000xx000B

(52/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED200H	CAN3 global control register	C3GMCTRL	R/W	–	–	√	0000H
03FED202H	CAN3 global clock select register	C3GMCS	R/W	–	√	–	0FH
03FED206H	CAN3 global block transmission control register	C3GMABT	R/W	–	–	√	0000H
03FED208H	CAN3 global block transmission delay setting register	C3GMABTD	R/W	–	√	–	00H
03FED240H	CAN3 module mask 1 register	C3MASK1L	R/W	–	–	√	Undefined
03FED242H		C3MASK1H					
03FED244H	CAN3 module mask 2 register	C3MASK2L	R/W	–	–	√	Undefined
03FED246H		C3MASK2H					
03FED248H	CAN3 module mask 3 register	C3MASK3L	R/W	–	–	√	Undefined
03FED24AH		C3MASK3H					
03FED24CH	CAN3 module mask 4 register	C3MASK4L	R/W	–	–	√	Undefined
03FED24EH		C3MASK4H					
03FED250H	CAN3 module control register	C3CTRL	R/W	–	–	√	0000H
03FED252H	CAN3 module last error information register	C3LEC	R/W	–	√	–	00H
03FED253H	CAN3 module information register	C3INFO	R	–	√	–	00H
03FED254H	CAN3 module error counter register	C3ERC	R	–	–	√	0000H
03FED256H	CAN3 module interrupt enable register	C3IE	R/W	–	–	√	0000H
03FED258H	CAN3 module interrupt status register	C3INTS	R/W	–	–	√	0000H
03FED25AH	CAN3 module bit rate prescaler register	C3BRP	R/W	–	√	–	FFH
03FED25CH	CAN3 module bit rate register	C3BTR	R/W	–	–	√	370FH
03FED25EH	CAN3 module last in-pointer register	C3LIPT	R	–	√	–	Undefined
03FED260H	CAN3 module receive history list register	C3RGPT	R/W	–	–	√	xx02H
03FED262H	CAN3 module last out-pointer register	C3LOPT	R	–	√	–	Undefined
03FED264H	CAN3 module transmit history list register	C3TGPT	R/W	–	–	√	xx02H
03FED266H	CAN3 module time stamp register	C3TS	R/W	–	–	√	0000H

(53/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED300H	CAN3 message data byte 01 register 00	C3MDATA0100	R/W			√	Undefined
03FED300H	CAN3 message data byte 0 register 00	C3MDATA000			√		Undefined
03FED301H	CAN3 message data byte 1 register 00	C3MDATA100			√		Undefined
03FED302H	CAN3 message data byte 23 register 00	C3MDATA2300				√	Undefined
03FED302H	CAN3 message data byte 2 register 00	C3MDATA200			√		Undefined
03FED303H	CAN3 message data byte 3 register 00	C3MDATA300			√		Undefined
03FED304H	CAN3 message data byte 45 register 00	C3MDATA4500				√	Undefined
03FED304H	CAN3 message data byte 4 register 00	C3MDATA400			√		Undefined
03FED305H	CAN3 message data byte 5 register 00	C3MDATA500			√		Undefined
03FED306H	CAN3 message data byte 67 register 00	C3MDATA6700				√	Undefined
03FED306H	CAN3 message data byte 6 register 00	C3MDATA600			√		Undefined
03FED307H	CAN3 message data byte 7 register 00	C3MDATA700			√		Undefined
03FED308H	CAN3 message data length code register 00	C3MDLC00			√		0000xxxxB
03FED309H	CAN3 message configuration register 00	C3MCONF00			√		Undefined
03FED30AH	CAN3 message ID register 00	C3MIDL00				√	Undefined
03FED30CH		C3MIDH00				√	Undefined
03FED30EH	CAN3 message control register 00	C3MCTRL00				√	00x00000 000xx000B
03FED320H	CAN3 message data byte 01 register 01	C3MDATA0101				√	Undefined
03FED320H	CAN3 message data byte 0 register 01	C3MDATA001			√		Undefined
03FED321H	CAN3 message data byte 1 register 01	C3MDATA101			√		Undefined
03FED322H	CAN3 message data byte 23 register 01	C3MDATA2301				√	Undefined
03FED322H	CAN3 message data byte 2 register 01	C3MDATA201			√		Undefined
03FED323H	CAN3 message data byte 3 register 01	C3MDATA301			√		Undefined
03FED324H	CAN3 message data byte 45 register 01	C3MDATA4501				√	Undefined
03FED324H	CAN3 message data byte 4 register 01	C3MDATA401			√		Undefined
03FED325H	CAN3 message data byte 5 register 01	C3MDATA501			√		Undefined
03FED326H	CAN3 message data byte 67 register 01	C3MDATA6701				√	Undefined
03FED326H	CAN3 message data byte 6 register 01	C3MDATA601			√		Undefined
03FED327H	CAN3 message data byte 7 register 01	C3MDATA701			√		Undefined
03FED328H	CAN3 message data length code register 01	C3MDLC01			√		0000xxxxB
03FED329H	CAN3 message configuration register 01	C3MCONF01			√		Undefined
03FED32AH	CAN3 message ID register 01	C3MIDL01				√	Undefined
03FED32CH		C3MIDH01				√	Undefined
03FED32EH	CAN3 message control register 01	C3MCTRL01				√	00x00000 000xx000B

(54/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED340H	CAN3 message data byte 01 register 02	C3MDATA0102	R/W			√	Undefined
03FED340H	CAN3 message data byte 0 register 02	C3MDATA002			√		Undefined
03FED341H	CAN3 message data byte 1 register 02	C3MDATA102			√		Undefined
03FED342H	CAN3 message data byte 23 register 02	C3MDATA2302				√	Undefined
03FED342H	CAN3 message data byte 2 register 02	C3MDATA202			√		Undefined
03FED343H	CAN3 message data byte 3 register 02	C3MDATA302			√		Undefined
03FED344H	CAN3 message data byte 45 register 02	C3MDATA4502				√	Undefined
03FED344H	CAN3 message data byte 4 register 02	C3MDATA402			√		Undefined
03FED345H	CAN3 message data byte 5 register 02	C3MDATA502			√		Undefined
03FED346H	CAN3 message data byte 67 register 02	C3MDATA6702				√	Undefined
03FED346H	CAN3 message data byte 6 register 02	C3MDATA602			√		Undefined
03FED347H	CAN3 message data byte 7 register 02	C3MDATA702			√		Undefined
03FED348H	CAN3 message data length code register 02	C3MDLC02			√		0000xxxxB
03FED349H	CAN3 message configuration register 02	C3MCONF02			√		Undefined
03FED34AH	CAN3 message ID register 02	C3MIDL02				√	Undefined
03FED34CH		C3MIDH02				√	Undefined
03FED34EH	CAN3 message control register 02	C3MCTRL02				√	00x00000 000xx000B
03FED360H	CAN3 message data byte 01 register 03	C3MDATA0103				√	Undefined
03FED360H	CAN3 message data byte 0 register 03	C3MDATA003			√		Undefined
03FED361H	CAN3 message data byte 1 register 03	C3MDATA103			√		Undefined
03FED362H	CAN3 message data byte 23 register 03	C3MDATA2303				√	Undefined
03FED362H	CAN3 message data byte 2 register 03	C3MDATA203			√		Undefined
03FED363H	CAN3 message data byte 3 register 03	C3MDATA303			√		Undefined
03FED364H	CAN3 message data byte 45 register 03	C3MDATA4503				√	Undefined
03FED364H	CAN3 message data byte 4 register 03	C3MDATA403			√		Undefined
03FED365H	CAN3 message data byte 5 register 03	C3MDATA503			√		Undefined
03FED366H	CAN3 message data byte 67 register 03	C3MDATA6703				√	Undefined
03FED366H	CAN3 message data byte 6 register 03	C3MDATA603			√		Undefined
03FED367H	CAN3 message data byte 7 register 03	C3MDATA703			√		Undefined
03FED368H	CAN3 message data length code register 03	C3MDLC03			√		0000xxxxB
03FED369H	CAN3 message configuration register 03	C3MCONF03			√		Undefined
03FED36AH	CAN3 message ID register 03	C3MIDL03				√	Undefined
03FED36CH		C3MIDH03				√	Undefined
03FED36EH	CAN3 message control register 03	C3MCTRL03				√	00x00000 000xx000B

(55/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED380H	CAN3 message data byte 01 register 04	C3MDATA0104	R/W			√	Undefined
03FED380H	CAN3 message data byte 0 register 04	C3MDATA004			√		Undefined
03FED381H	CAN3 message data byte 1 register 04	C3MDATA104			√		Undefined
03FED382H	CAN3 message data byte 23 register 04	C3MDATA2304				√	Undefined
03FED382H	CAN3 message data byte 2 register 04	C3MDATA204			√		Undefined
03FED383H	CAN3 message data byte 3 register 04	C3MDATA304			√		Undefined
03FED384H	CAN3 message data byte 45 register 04	C3MDATA4504				√	Undefined
03FED384H	CAN3 message data byte 4 register 04	C3MDATA404			√		Undefined
03FED385H	CAN3 message data byte 5 register 04	C3MDATA504			√		Undefined
03FED386H	CAN3 message data byte 67 register 04	C3MDATA6704				√	Undefined
03FED386H	CAN3 message data byte 6 register 04	C3MDATA604			√		Undefined
03FED387H	CAN3 message data byte 7 register 04	C3MDATA704			√		Undefined
03FED388H	CAN3 message data length code register 04	C3MDLC04			√		0000xxxxB
03FED389H	CAN3 message configuration register 04	C3MCONF04			√		Undefined
03FED38AH	CAN3 message ID register 04	C3MIDL04				√	Undefined
03FED38CH		C3MIDH04				√	Undefined
03FED38EH	CAN3 message control register 04	C3MCTRL04				√	00x00000 000xx000B
03FED3A0H	CAN3 message data byte 01 register 05	C3MDATA0105				√	Undefined
03FED3A0H	CAN3 message data byte 0 register 05	C3MDATA005			√		Undefined
03FED3A1H	CAN3 message data byte 1 register 05	C3MDATA105			√		Undefined
03FED3A2H	CAN3 message data byte 23 register 05	C3MDATA2305				√	Undefined
03FED3A2H	CAN3 message data byte 2 register 05	C3MDATA205			√		Undefined
03FED3A3H	CAN3 message data byte 3 register 05	C3MDATA305			√		Undefined
03FED3A4H	CAN3 message data byte 45 register 05	C3MDATA4505				√	Undefined
03FED3A4H	CAN3 message data byte 4 register 05	C3MDATA405			√		Undefined
03FED3A5H	CAN3 message data byte 5 register 05	C3MDATA505			√		Undefined
03FED3A6H	CAN3 message data byte 67 register 05	C3MDATA6705				√	Undefined
03FED3A6H	CAN3 message data byte 6 register 05	C3MDATA605			√		Undefined
03FED3A7H	CAN3 message data byte 7 register 05	C3MDATA705			√		Undefined
03FED3A8H	CAN3 message data length code register 05	C3MDLC05			√		0000xxxxB
03FED3A9H	CAN3 message configuration register 05	C3MCONF05			√		Undefined
03FED3AAH	CAN3 message ID register 05	C3MIDL05				√	Undefined
03FED3ACH		C3MIDH05				√	Undefined
03FED3AEH	CAN3 message control register 05	C3MCTRL05				√	00x00000 000xx000B

(56/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED3C0H	CAN3 message data byte 01 register 06	C3MDATA0106	R/W			√	Undefined
03FED3C0H	CAN3 message data byte 0 register 06	C3MDATA006			√		Undefined
03FED3C1H	CAN3 message data byte 1 register 06	C3MDATA106			√		Undefined
03FED3C2H	CAN3 message data byte 23 register 06	C3MDATA2306				√	Undefined
03FED3C2H	CAN3 message data byte 2 register 06	C3MDATA206			√		Undefined
03FED3C3H	CAN3 message data byte 3 register 06	C3MDATA306			√		Undefined
03FED3C4H	CAN3 message data byte 45 register 06	C3MDATA4506				√	Undefined
03FED3C4H	CAN3 message data byte 4 register 06	C3MDATA406			√		Undefined
03FED3C5H	CAN3 message data byte 5 register 06	C3MDATA506			√		Undefined
03FED3C6H	CAN3 message data byte 67 register 06	C3MDATA6706				√	Undefined
03FED3C6H	CAN3 message data byte 6 register 06	C3MDATA606			√		Undefined
03FED3C7H	CAN3 message data byte 7 register 06	C3MDATA706			√		Undefined
03FED3C8H	CAN3 message data length code register 06	C3MDLC06			√		0000xxxxB
03FED3C9H	CAN3 message configuration register 06	C3MCONF06			√		Undefined
03FED3CAH	CAN3 message ID register 06	C3MIDL06				√	Undefined
03FED3CCH		C3MIDH06				√	Undefined
03FED3CEH	CAN3 message control register 06	C3MCTRL06				√	00x00000 000xx000B
03FED3E0H	CAN3 message data byte 01 register 07	C3MDATA0107				√	Undefined
03FED3E0H	CAN3 message data byte 0 register 07	C3MDATA007			√		Undefined
03FED3E1H	CAN3 message data byte 1 register 07	C3MDATA107			√		Undefined
03FED3E2H	CAN3 message data byte 23 register 07	C3MDATA2307				√	Undefined
03FED3E2H	CAN3 message data byte 2 register 07	C3MDATA207			√		Undefined
03FED3E3H	CAN3 message data byte 3 register 07	C3MDATA307			√		Undefined
03FED3E4H	CAN3 message data byte 45 register 07	C3MDATA4507				√	Undefined
03FED3E4H	CAN3 message data byte 4 register 07	C3MDATA407			√		Undefined
03FED3E5H	CAN3 message data byte 5 register 07	C3MDATA507			√		Undefined
03FED3E6H	CAN3 message data byte 67 register 07	C3MDATA6707				√	Undefined
03FED3E6H	CAN3 message data byte 6 register 07	C3MDATA607			√		Undefined
03FED3E7H	CAN3 message data byte 7 register 07	C3MDATA707			√		Undefined
03FED3E8H	CAN3 message data length code register 07	C3MDLC07			√		0000xxxxB
03FED3E9H	CAN3 message configuration register 07	C3MCONF07			√		Undefined
03FED3EAH	CAN3 message ID register 07	C3MIDL07				√	Undefined
03FED3ECH		C3MIDH07				√	Undefined
03FED3EEH	CAN3 message control register 07	C3MCTRL07				√	00x00000 000xx000B

(57/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED400H	CAN3 message data byte 01 register 08	C3MDATA0108	R/W			√	Undefined
03FED400H	CAN3 message data byte 0 register 08	C3MDATA008			√		Undefined
03FED401H	CAN3 message data byte 1 register 08	C3MDATA108			√		Undefined
03FED402H	CAN3 message data byte 23 register 08	C3MDATA2308				√	Undefined
03FED402H	CAN3 message data byte 2 register 08	C3MDATA208			√		Undefined
03FED403H	CAN3 message data byte 3 register 08	C3MDATA308			√		Undefined
03FED404H	CAN3 message data byte 45 register 08	C3MDATA4508				√	Undefined
03FED404H	CAN3 message data byte 4 register 08	C3MDATA408			√		Undefined
03FED405H	CAN3 message data byte 5 register 08	C3MDATA508			√		Undefined
03FED406H	CAN3 message data byte 67 register 08	C3MDATA6708				√	Undefined
03FED406H	CAN3 message data byte 6 register 08	C3MDATA608			√		Undefined
03FED407H	CAN3 message data byte 7 register 08	C3MDATA708			√		Undefined
03FED408H	CAN3 message data length code register 08	C3MDLC08			√		0000xxxxB
03FED409H	CAN3 message configuration register 08	C3MCONF08			√		Undefined
03FED40AH	CAN3 message ID register 08	C3MIDL08				√	Undefined
03FED40CH		C3MIDH08				√	Undefined
03FED40EH	CAN3 message control register 08	C3MCTRL08				√	00x00000 000xx000B
03FED420H	CAN3 message data byte 01 register 09	C3MDATA0109				√	Undefined
03FED420H	CAN3 message data byte 0 register 09	C3MDATA009			√		Undefined
03FED421H	CAN3 message data byte 1 register 09	C3MDATA109			√		Undefined
03FED422H	CAN3 message data byte 23 register 09	C3MDATA2309				√	Undefined
03FED422H	CAN3 message data byte 2 register 09	C3MDATA209			√		Undefined
03FED423H	CAN3 message data byte 3 register 09	C3MDATA309			√		Undefined
03FED424H	CAN3 message data byte 45 register 09	C3MDATA4509				√	Undefined
03FED424H	CAN3 message data byte 4 register 09	C3MDATA409			√		Undefined
03FED425H	CAN3 message data byte 5 register 09	C3MDATA509			√		Undefined
03FED426H	CAN3 message data byte 67 register 09	C3MDATA6709				√	Undefined
03FED426H	CAN3 message data byte 6 register 09	C3MDATA609			√		Undefined
03FED427H	CAN3 message data byte 7 register 09	C3MDATA709			√		Undefined
03FED428H	CAN3 message data length code register 09	C3MDLC09			√		0000xxxxB
03FED429H	CAN3 message configuration register 09	C3MCONF09			√		Undefined
03FED42AH	CAN3 message ID register 09	C3MIDL09				√	Undefined
03FED42CH		C3MIDH09				√	Undefined
03FED42EH	CAN3 message control register 09	C3MCTRL09				√	00x00000 000xx000B

(58/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED440H	CAN3 message data byte 01 register 10	C3MDATA0110	R/W			√	Undefined
03FED440H	CAN3 message data byte 0 register 10	C3MDATA010			√		Undefined
03FED441H	CAN3 message data byte 1 register 10	C3MDATA110			√		Undefined
03FED442H	CAN3 message data byte 23 register 10	C3MDATA2310				√	Undefined
03FED442H	CAN3 message data byte 2 register 10	C3MDATA210			√		Undefined
03FED443H	CAN3 message data byte 3 register 10	C3MDATA310			√		Undefined
03FED444H	CAN3 message data byte 45 register 10	C3MDATA4510				√	Undefined
03FED444H	CAN3 message data byte 4 register 10	C3MDATA410			√		Undefined
03FED445H	CAN3 message data byte 5 register 10	C3MDATA510			√		Undefined
03FED446H	CAN3 message data byte 67 register 10	C3MDATA6710				√	Undefined
03FED446H	CAN3 message data byte 6 register 10	C3MDATA610			√		Undefined
03FED447H	CAN3 message data byte 7 register 10	C3MDATA710			√		Undefined
03FED448H	CAN3 message data length code register 10	C3MDLC10			√		0000xxxxB
03FED449H	CAN3 message configuration register 10	C3MCONF10			√		Undefined
03FED44AH	CAN3 message ID register 10	C3MIDL10				√	Undefined
03FED44CH		C3MIDH10				√	Undefined
03FED44EH	CAN3 message control register 10	C3MCTRL10				√	00x00000 000xx000B
03FED460H	CAN3 message data byte 01 register 11	C3MDATA0111				√	Undefined
03FED460H	CAN3 message data byte 0 register 11	C3MDATA011			√		Undefined
03FED461H	CAN3 message data byte 1 register 11	C3MDATA111			√		Undefined
03FED462H	CAN3 message data byte 23 register 11	C3MDATA2311				√	Undefined
03FED462H	CAN3 message data byte 2 register 11	C3MDATA211			√		Undefined
03FED463H	CAN3 message data byte 3 register 11	C3MDATA311			√		Undefined
03FED464H	CAN3 message data byte 45 register 11	C3MDATA4511				√	Undefined
03FED464H	CAN3 message data byte 4 register 11	C3MDATA411			√		Undefined
03FED465H	CAN3 message data byte 5 register 11	C3MDATA511			√		Undefined
03FED466H	CAN3 message data byte 67 register 11	C3MDATA6711				√	Undefined
03FED466H	CAN3 message data byte 6 register 11	C3MDATA611			√		Undefined
03FED467H	CAN3 message data byte 7 register 11	C3MDATA711			√		Undefined
03FED468H	CAN3 message data length code register 11	C3MDLC11			√		0000xxxxB
03FED469H	CAN3 message configuration register 11	C3MCONF11			√		Undefined
03FED46AH	CAN3 message ID register 11	C3MIDL11				√	Undefined
03FED46CH		C3MIDH11				√	Undefined
03FED46EH	CAN3 message control register 11	C3MCTRL11				√	00x00000 000xx000B

(59/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED480H	CAN3 message data byte 01 register 12	C3MDATA0112	R/W			√	Undefined
03FED480H	CAN3 message data byte 0 register 12	C3MDATA012			√		Undefined
03FED481H	CAN3 message data byte 1 register 12	C3MDATA112			√		Undefined
03FED482H	CAN3 message data byte 23 register 12	C3MDATA2312				√	Undefined
03FED482H	CAN3 message data byte 2 register 12	C3MDATA212			√		Undefined
03FED483H	CAN3 message data byte 3 register 12	C3MDATA312			√		Undefined
03FED484H	CAN3 message data byte 45 register 12	C3MDATA4512				√	Undefined
03FED484H	CAN3 message data byte 4 register 12	C3MDATA412			√		Undefined
03FED485H	CAN3 message data byte 5 register 12	C3MDATA512			√		Undefined
03FED486H	CAN3 message data byte 67 register 12	C3MDATA6712				√	Undefined
03FED486H	CAN3 message data byte 6 register 12	C3MDATA612			√		Undefined
03FED487H	CAN3 message data byte 7 register 12	C3MDATA712			√		Undefined
03FED488H	CAN3 message data length code register 12	C3MDLC12			√		0000xxxxB
03FED489H	CAN3 message configuration register 12	C3MCONF12			√		Undefined
03FED48AH	CAN3 message ID register 12	C3MIDL12				√	Undefined
03FED48CH		C3MIDH12				√	Undefined
03FED48EH	CAN3 message control register 12	C3MCTRL12				√	00x00000 000xx000B
03FED4A0H	CAN3 message data byte 01 register 13	C3MDATA0113				√	Undefined
03FED4A0H	CAN3 message data byte 0 register 13	C3MDATA013			√		Undefined
03FED4A1H	CAN3 message data byte 1 register 13	C3MDATA113			√		Undefined
03FED4A2H	CAN3 message data byte 23 register 13	C3MDATA2313				√	Undefined
03FED4A2H	CAN3 message data byte 2 register 13	C3MDATA213			√		Undefined
03FED4A3H	CAN3 message data byte 3 register 13	C3MDATA313			√		Undefined
03FED4A4H	CAN3 message data byte 45 register 13	C3MDATA4513				√	Undefined
03FED4A4H	CAN3 message data byte 4 register 13	C3MDATA413			√		Undefined
03FED4A5H	CAN3 message data byte 5 register 13	C3MDATA513			√		Undefined
03FED4A6H	CAN3 message data byte 67 register 13	C3MDATA6713				√	Undefined
03FED4A6H	CAN3 message data byte 6 register 13	C3MDATA613			√		Undefined
03FED4A7H	CAN3 message data byte 7 register 13	C3MDATA713			√		Undefined
03FED4A8H	CAN3 message data length code register 13	C3MDLC13			√		0000xxxxB
03FED4A9H	CAN3 message configuration register 13	C3MCONF13			√		Undefined
03FED4AAH	CAN3 message ID register 13	C3MIDL13				√	Undefined
03FED4ACH		C3MIDH13				√	Undefined
03FED4AEH	CAN3 message control register 13	C3MCTRL13				√	00x00000 000xx000B

(60/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED4C0H	CAN3 message data byte 01 register 14	C3MDATA0114	R/W			√	Undefined
03FED4C0H	CAN3 message data byte 0 register 14	C3MDATA014			√		Undefined
03FED4C1H	CAN3 message data byte 1 register 14	C3MDATA114			√		Undefined
03FED4C2H	CAN3 message data byte 23 register 14	C3MDATA2314				√	Undefined
03FED4C2H	CAN3 message data byte 2 register 14	C3MDATA214			√		Undefined
03FED4C3H	CAN3 message data byte 3 register 14	C3MDATA314			√		Undefined
03FED4C4H	CAN3 message data byte 45 register 14	C3MDATA4514				√	Undefined
03FED4C4H	CAN3 message data byte 4 register 14	C3MDATA414			√		Undefined
03FED4C5H	CAN3 message data byte 5 register 14	C3MDATA514			√		Undefined
03FED4C6H	CAN3 message data byte 67 register 14	C3MDATA6714				√	Undefined
03FED4C6H	CAN3 message data byte 6 register 14	C3MDATA614			√		Undefined
03FED4C7H	CAN3 message data byte 7 register 14	C3MDATA714			√		Undefined
03FED4C8H	CAN3 message data length code register 14	C3MDLC14			√		0000xxxxB
03FED4C9H	CAN3 message configuration register 14	C3MCONF14			√		Undefined
03FED4CAH	CAN3 message ID register 14	C3MIDL14				√	Undefined
03FED4CCH		C3MIDH14				√	Undefined
03FED4CEH	CAN3 message control register 14	C3MCTRL14				√	00x00000 000xx000B
03FED4E0H	CAN3 message data byte 01 register 15	C3MDATA0115				√	Undefined
03FED4E0H	CAN3 message data byte 0 register 15	C3MDATA015			√		Undefined
03FED4E1H	CAN3 message data byte 1 register 15	C3MDATA115			√		Undefined
03FED4E2H	CAN3 message data byte 23 register 15	C3MDATA2315				√	Undefined
03FED4E2H	CAN3 message data byte 2 register 15	C3MDATA215			√		Undefined
03FED4E3H	CAN3 message data byte 3 register 15	C3MDATA315			√		Undefined
03FED4E4H	CAN3 message data byte 45 register 15	C3MDATA4515				√	Undefined
03FED4E4H	CAN3 message data byte 4 register 15	C3MDATA415			√		Undefined
03FED4E5H	CAN3 message data byte 5 register 15	C3MDATA515			√		Undefined
03FED4E6H	CAN3 message data byte 67 register 15	C3MDATA6715				√	Undefined
03FED4E6H	CAN3 message data byte 6 register 15	C3MDATA615			√		Undefined
03FED4E7H	CAN3 message data byte 7 register 15	C3MDATA715			√		Undefined
03FED4E8H	CAN3 message data length code register 15	C3MDLC15			√		0000xxxxB
03FED4E9H	CAN3 message configuration register 15	C3MCONF15			√		Undefined
03FED4EAH	CAN3 message ID register 15	C3MIDL15				√	Undefined
03FED4ECH		C3MIDH15				√	Undefined
03FED4EEH	CAN3 message control register 15	C3MCTRL15				√	00x00000 000xx000B

(61/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED500H	CAN3 message data byte 01 register 16	C3MDATA0116	R/W			√	Undefined
03FED500H	CAN3 message data byte 0 register 16	C3MDATA016			√		Undefined
03FED501H	CAN3 message data byte 1 register 16	C3MDATA116			√		Undefined
03FED502H	CAN3 message data byte 23 register 16	C3MDATA2316				√	Undefined
03FED502H	CAN3 message data byte 2 register 16	C3MDATA216			√		Undefined
03FED503H	CAN3 message data byte 3 register 16	C3MDATA316			√		Undefined
03FED504H	CAN3 message data byte 45 register 16	C3MDATA4516				√	Undefined
03FED504H	CAN3 message data byte 4 register 16	C3MDATA416			√		Undefined
03FED505H	CAN3 message data byte 5 register 16	C3MDATA516			√		Undefined
03FED506H	CAN3 message data byte 67 register 16	C3MDATA6716				√	Undefined
03FED506H	CAN3 message data byte 6 register 16	C3MDATA616			√		Undefined
03FED507H	CAN3 message data byte 7 register 16	C3MDATA716			√		Undefined
03FED508H	CAN3 message data length code register 16	C3MDLC16			√		0000xxxxB
03FED509H	CAN3 message configuration register 16	C3MCONF16			√		Undefined
03FED50AH	CAN3 message ID register 16	C3MIDL16				√	Undefined
03FED50CH		C3MIDH16				√	Undefined
03FED50EH	CAN3 message control register 16	C3MCTRL16				√	00x00000 000xx000B
03FED520H	CAN3 message data byte 01 register 17	C3MDATA0117				√	Undefined
03FED520H	CAN3 message data byte 0 register 17	C3MDATA017			√		Undefined
03FED521H	CAN3 message data byte 1 register 17	C3MDATA117			√		Undefined
03FED522H	CAN3 message data byte 23 register 17	C3MDATA2317				√	Undefined
03FED522H	CAN3 message data byte 2 register 17	C3MDATA217			√		Undefined
03FED523H	CAN3 message data byte 3 register 17	C3MDATA317			√		Undefined
03FED524H	CAN3 message data byte 45 register 17	C3MDATA4517				√	Undefined
03FED524H	CAN3 message data byte 4 register 17	C3MDATA417			√		Undefined
03FED525H	CAN3 message data byte 5 register 17	C3MDATA517			√		Undefined
03FED526H	CAN3 message data byte 67 register 17	C3MDATA6717				√	Undefined
03FED526H	CAN3 message data byte 6 register 17	C3MDATA617			√		Undefined
03FED527H	CAN3 message data byte 7 register 17	C3MDATA717			√		Undefined
03FED528H	CAN3 message data length code register 17	C3MDLC17			√		0000xxxxB
03FED529H	CAN3 message configuration register 17	C3MCONF17			√		Undefined
03FED52AH	CAN3 message ID register 17	C3MIDL17				√	Undefined
03FED52CH		C3MIDH17				√	Undefined
03FED52EH	CAN3 message control register 17	C3MCTRL17				√	00x00000 000xx000B

(62/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED540H	CAN3 message data byte 01 register 18	C3MDATA0118	R/W			√	Undefined
03FED540H	CAN3 message data byte 0 register 18	C3MDATA018			√		Undefined
03FED541H	CAN3 message data byte 1 register 18	C3MDATA118			√		Undefined
03FED542H	CAN3 message data byte 23 register 18	C3MDATA2318				√	Undefined
03FED542H	CAN3 message data byte 2 register 18	C3MDATA218			√		Undefined
03FED543H	CAN3 message data byte 3 register 18	C3MDATA318			√		Undefined
03FED544H	CAN3 message data byte 45 register 18	C3MDATA4518				√	Undefined
03FED544H	CAN3 message data byte 4 register 18	C3MDATA418			√		Undefined
03FED545H	CAN3 message data byte 5 register 18	C3MDATA518			√		Undefined
03FED546H	CAN3 message data byte 67 register 18	C3MDATA6718				√	Undefined
03FED546H	CAN3 message data byte 6 register 18	C3MDATA618			√		Undefined
03FED547H	CAN3 message data byte 7 register 18	C3MDATA718			√		Undefined
03FED548H	CAN3 message data length code register 18	C3MDLC18			√		0000xxxxB
03FED549H	CAN3 message configuration register 18	C3MCONF18			√		Undefined
03FED54AH	CAN3 message ID register 18	C3MIDL18				√	Undefined
03FED54CH		C3MIDH18				√	Undefined
03FED54EH	CAN3 message control register 18	C3MCTRL18				√	00x00000 000xx000B
03FED560H	CAN3 message data byte 01 register 19	C3MDATA0119				√	Undefined
03FED560H	CAN3 message data byte 0 register 19	C3MDATA019			√		Undefined
03FED561H	CAN3 message data byte 1 register 19	C3MDATA119			√		Undefined
03FED562H	CAN3 message data byte 23 register 19	C3MDATA2319				√	Undefined
03FED562H	CAN3 message data byte 2 register 19	C3MDATA219			√		Undefined
03FED563H	CAN3 message data byte 3 register 19	C3MDATA319			√		Undefined
03FED564H	CAN3 message data byte 45 register 19	C3MDATA4519				√	Undefined
03FED564H	CAN3 message data byte 4 register 19	C3MDATA419			√		Undefined
03FED565H	CAN3 message data byte 5 register 19	C3MDATA519			√		Undefined
03FED566H	CAN3 message data byte 67 register 19	C3MDATA6719				√	Undefined
03FED566H	CAN3 message data byte 6 register 19	C3MDATA619			√		Undefined
03FED567H	CAN3 message data byte 7 register 19	C3MDATA719			√		Undefined
03FED568H	CAN3 message data length code register 19	C3MDLC19			√		0000xxxxB
03FED569H	CAN3 message configuration register 19	C3MCONF19			√		Undefined
03FED56AH	CAN3 message ID register 19	C3MIDL19				√	Undefined
03FED56CH		C3MIDH19				√	Undefined
03FED56EH	CAN3 message control register 19	C3MCTRL19				√	00x00000 000xx000B

(63/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED580H	CAN3 message data byte 01 register 20	C3MDATA0120	R/W			√	Undefined
03FED580H	CAN3 message data byte 0 register 20	C3MDATA020			√		Undefined
03FED581H	CAN3 message data byte 1 register 20	C3MDATA120			√		Undefined
03FED582H	CAN3 message data byte 23 register 20	C3MDATA2320				√	Undefined
03FED582H	CAN3 message data byte 2 register 20	C3MDATA220			√		Undefined
03FED583H	CAN3 message data byte 3 register 20	C3MDATA320			√		Undefined
03FED584H	CAN3 message data byte 45 register 20	C3MDATA4520				√	Undefined
03FED584H	CAN3 message data byte 4 register 20	C3MDATA420			√		Undefined
03FED585H	CAN3 message data byte 5 register 20	C3MDATA520			√		Undefined
03FED586H	CAN3 message data byte 67 register 20	C3MDATA6720				√	Undefined
03FED586H	CAN3 message data byte 6 register 20	C3MDATA620			√		Undefined
03FED587H	CAN3 message data byte 7 register 20	C3MDATA720			√		Undefined
03FED588H	CAN3 message data length code register 20	C3MDLC20			√		0000xxxxB
03FED589H	CAN3 message configuration register 20	C3MCONF20			√		Undefined
03FED58AH	CAN3 message ID register 20	C3MIDL20				√	Undefined
03FED58CH		C3MIDH20				√	Undefined
03FED58EH	CAN3 message control register 20	C3MCTRL20				√	00x00000 000xx000B
03FED5A0H	CAN3 message data byte 01 register 21	C3MDATA0121				√	Undefined
03FED5A0H	CAN3 message data byte 0 register 21	C3MDATA021			√		Undefined
03FED5A1H	CAN3 message data byte 1 register 21	C3MDATA121			√		Undefined
03FED5A2H	CAN3 message data byte 23 register 21	C3MDATA2321				√	Undefined
03FED5A2H	CAN3 message data byte 2 register 21	C3MDATA221			√		Undefined
03FED5A3H	CAN3 message data byte 3 register 21	C3MDATA321			√		Undefined
03FED5A4H	CAN3 message data byte 45 register 21	C3MDATA4521				√	Undefined
03FED5A4H	CAN3 message data byte 4 register 21	C3MDATA421			√		Undefined
03FED5A5H	CAN3 message data byte 5 register 21	C3MDATA521			√		Undefined
03FED5A6H	CAN3 message data byte 67 register 21	C3MDATA6721				√	Undefined
03FED5A6H	CAN3 message data byte 6 register 21	C3MDATA621			√		Undefined
03FED5A7H	CAN3 message data byte 7 register 21	C3MDATA721			√		Undefined
03FED5A8H	CAN3 message data length code register 21	C3MDLC21			√		0000xxxxB
03FED5A9H	CAN3 message configuration register 21	C3MCONF21			√		Undefined
03FED5AAH	CAN3 message ID register 21	C3MIDL21				√	Undefined
03FED5ACH		C3MIDH21				√	Undefined
03FED5AEH	CAN3 message control register 21	C3MCTRL21				√	00x00000 000xx000B

(64/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED5C0H	CAN3 message data byte 01 register 22	C3MDATA0122	R/W			√	Undefined
03FED5C0H	CAN3 message data byte 0 register 22	C3MDATA022			√		Undefined
03FED5C1H	CAN3 message data byte 1 register 22	C3MDATA122			√		Undefined
03FED5C2H	CAN3 message data byte 23 register 22	C3MDATA2322				√	Undefined
03FED5C2H	CAN3 message data byte 2 register 22	C3MDATA222			√		Undefined
03FED5C3H	CAN3 message data byte 3 register 22	C3MDATA322			√		Undefined
03FED5C4H	CAN3 message data byte 45 register 22	C3MDATA4522				√	Undefined
03FED5C4H	CAN3 message data byte 4 register 22	C3MDATA422			√		Undefined
03FED5C5H	CAN3 message data byte 5 register 22	C3MDATA522			√		Undefined
03FED5C6H	CAN3 message data byte 67 register 22	C3MDATA6722				√	Undefined
03FED5C6H	CAN3 message data byte 6 register 22	C3MDATA622			√		Undefined
03FED5C7H	CAN3 message data byte 7 register 22	C3MDATA722			√		Undefined
03FED5C8H	CAN3 message data length code register 22	C3MDLC22			√		0000xxxxB
03FED5C9H	CAN3 message configuration register 22	C3MCONF22			√		Undefined
03FED5CAH	CAN3 message ID register 22	C3MIDL22				√	Undefined
03FED5CCH		C3MIDH22				√	Undefined
03FED5CEH	CAN3 message control register 22	C3MCTRL22				√	00x00000 000xx000B
03FED5E0H	CAN3 message data byte 01 register 23	C3MDATA0123				√	Undefined
03FED5E0H	CAN3 message data byte 0 register 23	C3MDATA023			√		Undefined
03FED5E1H	CAN3 message data byte 1 register 23	C3MDATA123			√		Undefined
03FED5E2H	CAN3 message data byte 23 register 23	C3MDATA2323				√	Undefined
03FED5E2H	CAN3 message data byte 2 register 23	C3MDATA223			√		Undefined
03FED5E3H	CAN3 message data byte 3 register 23	C3MDATA323			√		Undefined
03FED5E4H	CAN3 message data byte 45 register 23	C3MDATA4523				√	Undefined
03FED5E4H	CAN3 message data byte 4 register 23	C3MDATA423			√		Undefined
03FED5E5H	CAN3 message data byte 5 register 23	C3MDATA523			√		Undefined
03FED5E6H	CAN3 message data byte 67 register 23	C3MDATA6723				√	Undefined
03FED5E6H	CAN3 message data byte 6 register 23	C3MDATA623			√		Undefined
03FED5E7H	CAN3 message data byte 7 register 23	C3MDATA723			√		Undefined
03FED5E8H	CAN3 message data length code register 23	C3MDLC23			√		0000xxxxB
03FED5E9H	CAN3 message configuration register 23	C3MCONF23			√		Undefined
03FED5EAH	CAN3 message ID register 23	C3MIDL23				√	Undefined
03FED5ECH		C3MIDH23				√	Undefined
03FED5EEH	CAN3 message control register 23	C3MCTRL23				√	00x00000 000xx000B

(65/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED600H	CAN3 message data byte 01 register 24	C3MDATA0124	R/W			√	Undefined
03FED600H	CAN3 message data byte 0 register 24	C3MDATA024			√		Undefined
03FED601H	CAN3 message data byte 1 register 24	C3MDATA124			√		Undefined
03FED602H	CAN3 message data byte 23 register 24	C3MDATA2324				√	Undefined
03FED602H	CAN3 message data byte 2 register 24	C3MDATA224			√		Undefined
03FED603H	CAN3 message data byte 3 register 24	C3MDATA324			√		Undefined
03FED604H	CAN3 message data byte 45 register 24	C3MDATA4524				√	Undefined
03FED604H	CAN3 message data byte 4 register 24	C3MDATA424			√		Undefined
03FED605H	CAN3 message data byte 5 register 24	C3MDATA524			√		Undefined
03FED606H	CAN3 message data byte 67 register 24	C3MDATA6724				√	Undefined
03FED606H	CAN3 message data byte 6 register 24	C3MDATA624			√		Undefined
03FED607H	CAN3 message data byte 7 register 24	C3MDATA724			√		Undefined
03FED608H	CAN3 message data length code register 24	C3MDLC24			√		0000xxxxB
03FED609H	CAN3 message configuration register 24	C3MCONF24			√		Undefined
03FED60AH	CAN3 message ID register 24	C3MIDL24				√	Undefined
03FED60CH		C3MIDH24				√	Undefined
03FED60EH	CAN3 message control register 24	C3MCTRL24				√	00x00000 000xx000B
03FED620H	CAN3 message data byte 01 register 25	C3MDATA0125				√	Undefined
03FED620H	CAN3 message data byte 0 register 25	C3MDATA025			√		Undefined
03FED621H	CAN3 message data byte 1 register 25	C3MDATA125			√		Undefined
03FED622H	CAN3 message data byte 23 register 25	C3MDATA2325				√	Undefined
03FED622H	CAN3 message data byte 2 register 25	C3MDATA225			√		Undefined
03FED623H	CAN3 message data byte 3 register 25	C3MDATA325			√		Undefined
03FED624H	CAN3 message data byte 45 register 25	C3MDATA4525				√	Undefined
03FED624H	CAN3 message data byte 4 register 25	C3MDATA425			√		Undefined
03FED625H	CAN3 message data byte 5 register 25	C3MDATA525			√		Undefined
03FED626H	CAN3 message data byte 67 register 25	C3MDATA6725				√	Undefined
03FED626H	CAN3 message data byte 6 register 25	C3MDATA625			√		Undefined
03FED627H	CAN3 message data byte 7 register 25	C3MDATA725			√		Undefined
03FED628H	CAN3 message data length code register 25	C3MDLC25			√		0000xxxxB
03FED629H	CAN3 message configuration register 25	C3MCONF25			√		Undefined
03FED62AH	CAN3 message ID register 25	C3MIDL25				√	Undefined
03FED62CH		C3MIDH25				√	Undefined
03FED62EH	CAN3 message control register 25	C3MCTRL25				√	00x00000 000xx000B

(66/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED640H	CAN3 message data byte 01 register 26	C3MDATA0126	R/W			√	Undefined
03FED640H	CAN3 message data byte 0 register 26	C3MDATA026			√		Undefined
03FED641H	CAN3 message data byte 1 register 26	C3MDATA126			√		Undefined
03FED642H	CAN3 message data byte 23 register 26	C3MDATA2326				√	Undefined
03FED642H	CAN3 message data byte 2 register 26	C3MDATA226			√		Undefined
03FED643H	CAN3 message data byte 3 register 26	C3MDATA326			√		Undefined
03FED644H	CAN3 message data byte 45 register 26	C3MDATA4526				√	Undefined
03FED644H	CAN3 message data byte 4 register 26	C3MDATA426			√		Undefined
03FED645H	CAN3 message data byte 5 register 26	C3MDATA526			√		Undefined
03FED646H	CAN3 message data byte 67 register 26	C3MDATA6726				√	Undefined
03FED646H	CAN3 message data byte 6 register 26	C3MDATA626			√		Undefined
03FED647H	CAN3 message data byte 7 register 26	C3MDATA726			√		Undefined
03FED648H	CAN3 message data length code register 26	C3MDLC26			√		0000xxxxB
03FED649H	CAN3 message configuration register 26	C3MCONF26			√		Undefined
03FED64AH	CAN3 message ID register 26	C3MIDL26				√	Undefined
03FED64CH		C3MIDH26				√	Undefined
03FED64EH	CAN3 message control register 26	C3MCTRL26				√	00x00000 000xx000B
03FED660H	CAN3 message data byte 01 register 27	C3MDATA0127				√	Undefined
03FED660H	CAN3 message data byte 0 register 27	C3MDATA027			√		Undefined
03FED661H	CAN3 message data byte 1 register 27	C3MDATA127			√		Undefined
03FED662H	CAN3 message data byte 23 register 27	C3MDATA2327				√	Undefined
03FED662H	CAN3 message data byte 2 register 27	C3MDATA227			√		Undefined
03FED663H	CAN3 message data byte 3 register 27	C3MDATA327			√		Undefined
03FED664H	CAN3 message data byte 45 register 27	C3MDATA4527				√	Undefined
03FED664H	CAN3 message data byte 4 register 27	C3MDATA427			√		Undefined
03FED665H	CAN3 message data byte 5 register 27	C3MDATA527			√		Undefined
03FED666H	CAN3 message data byte 67 register 27	C3MDATA6727				√	Undefined
03FED666H	CAN3 message data byte 6 register 27	C3MDATA627			√		Undefined
03FED667H	CAN3 message data byte 7 register 27	C3MDATA727			√		Undefined
03FED668H	CAN3 message data length code register 27	C3MDLC27			√		0000xxxxB
03FED669H	CAN3 message configuration register 27	C3MCONF27			√		Undefined
03FED66AH	CAN3 message ID register 27	C3MIDL27				√	Undefined
03FED66CH		C3MIDH27				√	Undefined
03FED66EH	CAN3 message control register 27	C3MCTRL27				√	00x00000 000xx000B

(67/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED680H	CAN3 message data byte 01 register 28	C3MDATA0128	R/W			√	Undefined
03FED680H	CAN3 message data byte 0 register 28	C3MDATA028			√		Undefined
03FED681H	CAN3 message data byte 1 register 28	C3MDATA128			√		Undefined
03FED682H	CAN3 message data byte 23 register 28	C3MDATA2328				√	Undefined
03FED682H	CAN3 message data byte 2 register 28	C3MDATA228			√		Undefined
03FED683H	CAN3 message data byte 3 register 28	C3MDATA328			√		Undefined
03FED684H	CAN3 message data byte 45 register 28	C3MDATA4528				√	Undefined
03FED684H	CAN3 message data byte 4 register 28	C3MDATA428			√		Undefined
03FED685H	CAN3 message data byte 5 register 28	C3MDATA528			√		Undefined
03FED686H	CAN3 message data byte 67 register 28	C3MDATA6728				√	Undefined
03FED686H	CAN3 message data byte 6 register 28	C3MDATA628			√		Undefined
03FED687H	CAN3 message data byte 7 register 28	C3MDATA728			√		Undefined
03FED688H	CAN3 message data length code register 28	C3MDLC28			√		0000xxxxB
03FED689H	CAN3 message configuration register 28	C3MCONF28			√		Undefined
03FED68AH	CAN3 message ID register 28	C3MIDL28				√	Undefined
03FED68CH		C3MIDH28				√	Undefined
03FED68EH	CAN3 message control register 28	C3MCTRL28				√	00x00000 000xx000B
03FED6A0H	CAN3 message data byte 01 register 29	C3MDATA0129				√	Undefined
03FED6A0H	CAN3 message data byte 0 register 29	C3MDATA029			√		Undefined
03FED6A1H	CAN3 message data byte 1 register 29	C3MDATA129			√		Undefined
03FED6A2H	CAN3 message data byte 23 register 29	C3MDATA2329				√	Undefined
03FED6A2H	CAN3 message data byte 2 register 29	C3MDATA229			√		Undefined
03FED6A3H	CAN3 message data byte 3 register 29	C3MDATA329			√		Undefined
03FED6A4H	CAN3 message data byte 45 register 29	C3MDATA4529				√	Undefined
03FED6A4H	CAN3 message data byte 4 register 29	C3MDATA429			√		Undefined
03FED6A5H	CAN3 message data byte 5 register 29	C3MDATA529			√		Undefined
03FED6A6H	CAN3 message data byte 67 register 29	C3MDATA6729				√	Undefined
03FED6A6H	CAN3 message data byte 6 register 29	C3MDATA629			√		Undefined
03FED6A7H	CAN3 message data byte 7 register 29	C3MDATA729			√		Undefined
03FED6A8H	CAN3 message data length code register 29	C3MDLC29			√		0000xxxxB
03FED6A9H	CAN3 message configuration register 29	C3MCONF29			√		Undefined
03FED6AAH	CAN3 message ID register 29	C3MIDL29				√	Undefined
03FED6ACH		C3MIDH29				√	Undefined
03FED6AEH	CAN3 message control register 29	C3MCTRL29				√	00x00000 000xx000B

(68/68)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
				1	8	16	
03FED6C0H	CAN3 message data byte 01 register 30	C3MDATA0130	R/W			√	Undefined
03FED6C0H	CAN3 message data byte 0 register 30	C3MDATA030			√		Undefined
03FED6C1H	CAN3 message data byte 1 register 30	C3MDATA130			√		Undefined
03FED6C2H	CAN3 message data byte 23 register 30	C3MDATA2330				√	Undefined
03FED6C2H	CAN3 message data byte 2 register 30	C3MDATA230			√		Undefined
03FED6C3H	CAN3 message data byte 3 register 30	C3MDATA330			√		Undefined
03FED6C4H	CAN3 message data byte 45 register 30	C3MDATA4530				√	Undefined
03FED6C4H	CAN3 message data byte 4 register 30	C3MDATA430			√		Undefined
03FED6C5H	CAN3 message data byte 5 register 30	C3MDATA530			√		Undefined
03FED6C6H	CAN3 message data byte 67 register 30	C3MDATA6730				√	Undefined
03FED6C6H	CAN3 message data byte 6 register 30	C3MDATA630			√		Undefined
03FED6C7H	CAN3 message data byte 7 register 30	C3MDATA730			√		Undefined
03FED6C8H	CAN3 message data length code register 30	C3MDLC30			√		0000xxxxB
03FED6C9H	CAN3 message configuration register 30	C3MCONF30			√		Undefined
03FED6CAH	CAN3 message ID register 30	C3MIDL30				√	Undefined
03FED6CCH		C3MIDH30				√	Undefined
03FED6CEH	CAN3 message control register 30	C3MCTRL30				√	00x00000 000xx000B
03FED6E0H	CAN3 message data byte 01 register 31	C3MDATA0131				√	Undefined
03FED6E0H	CAN3 message data byte 0 register 31	C3MDATA031			√		Undefined
03FED6E1H	CAN3 message data byte 1 register 31	C3MDATA131			√		Undefined
03FED6E2H	CAN3 message data byte 23 register 31	C3MDATA2331				√	Undefined
03FED6E2H	CAN3 message data byte 2 register 31	C3MDATA231			√		Undefined
03FED6E3H	CAN3 message data byte 3 register 31	C3MDATA331			√		Undefined
03FED6E4H	CAN3 message data byte 45 register 31	C3MDATA4531				√	Undefined
03FED6E4H	CAN3 message data byte 4 register 31	C3MDATA431			√		Undefined
03FED6E5H	CAN3 message data byte 5 register 31	C3MDATA531			√		Undefined
03FED6E6H	CAN3 message data byte 67 register 31	C3MDATA6731				√	Undefined
03FED6E6H	CAN3 message data byte 6 register 31	C3MDATA631			√		Undefined
03FED6E7H	CAN3 message data byte 7 register 31	C3MDATA731			√		Undefined
03FED6E8H	CAN3 message data length code register 31	C3MDLC31			√		0000xxxx
03FED6E9H	CAN3 message configuration register 31	C3MCONF31			√		Undefined
03FED6EAH	CAN3 message ID register 31	C3MIDL31				√	Undefined
03FED6ECH		C3MIDH31				√	Undefined
03FED6EEH	CAN3 message control register 31	C3MCTRL31				√	00x00000 000xx000B

15.5.3 Register bit configuration

Table 15-17. 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 (μPD70F3231, μPD70F3232, μPD70F3233)

n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)

n = 0 to 3 (μPD70F3238, μPD70F3239)

Table 15-18. 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	CM4ID[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

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)

n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)

n = 0 to 3 (μPD70F3238, μPD70F3239)

(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
03FExx5AH	CnBRP	TQPRS[7:0]							
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 (μPD70F3231, μPD70F3232, μPD70F3233)

n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)

n = 0 to 3 (μPD70F3238, μPD70F3239)

Table 15-19. 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 (μPD70F3231, μPD70F3232, μPD70F3233)

n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)

n = 0 to 3 (μPD70F3238, μPD70F3239)

m = 0 to 31

15.6 Control Registers

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

(1) CAN global control register (CnGMCTRL)

The CnGMCTRL register is used to control the operation of the CAN module.

(1/2)

After reset: 0000H R/W Address: C0GMCTRL 03FFEC000H, C1GMCTRL 03FEC600H
C2GMCTRL 03FECC00H, C3GMCTRL 03FED200H

(a) Read

	15	14	13	12	11	10	9	8
CnGMCTRL	MBON	0	0	0	0	0	0	0
	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
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.

(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 only in the initialization mode.

(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.

After reset: 0FH R/W Address: C0GMCS 03FEC002H, C1GMCS 03FEC602H
C2GMCS 03FECC02H, C3GMCS 03FED202H

	7	6	5	4	3	2	1	0
CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

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.

(1/2)

After reset: 0000H R/W Address: C0GMABT 03FEC006H, C1GMABT 03FEC606H
C2GMABT 03FECC06H, C3GMABT 03FED206H

(a) Read

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

(a) Write

	15	14	13	12	11	10	9	8
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 (00H).

(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.

Caution Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.

(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.

After reset: 00H R/W Address: C0GMABTD 03FEC008H, C1GMABTD 03FEC608H
C2GMABTD 03FECC08H, C3GMABTD 03FED208H

	7	6	5	4	3	2	1	0
CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

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.

(1/2)

- CANn module mask 1 register (CnMASK1L, CnMASK1H)

After reset: Undefined R/W Address: C0MASK1L 03FEC040H, C1MASK1L 03FEC640H
C2MASK1L 03FECC40H, C3MASK1L 03FED240H
C0MASK1H 03FEC042H, C1MASK1H 03FEC642H
C2MASK1H 03FECC42H, C3MASK1H 03FED242H

	15	14	13	12	11	10	9	8
CnMASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	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)

After reset: Undefined R/W Address: C0MASK2L 03FEC044H, C1MASK2L 03FEC644H
C2MASK2L 03FECC44H, C3MASK2L 03FED244H
C0MASK2H 03FEC046H, C1MASK2H 03FEC646H
C2MASK2H 03FECC46H, C3MASK2H 03FED246H

	15	14	13	12	11	10	9	8
CnMASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	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

(2/2)

- CANn module mask 3 register (CnMASK3L, CnMASK3H)

After reset: Undefined R/W Address: C0MASK3L 03FEC048H, C1MASK3L 03FEC648H
C2MASK3L 03FECC48H, C3MASK3L 03FED248H
C0MASK3H 03FEC04AH, C1MASK3H 03FEC64AH
C2MASK3H 03FECC4AH, C3MASK3H 03FED24AH

	15	14	13	12	11	10	9	8
CnMASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	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

- CANn module mask 4 register (CnMASK4L, CnMASK4H)

After reset: Undefined R/W Address: C0MASK4L 03FEC04CH, C1MASK4L 03FEC64CH
C2MASK4L 03FECC4CH, C3MASK4L 03FED24CH
C0MASK4H 03FEC04EH, C1MASK4H 03FEC64EH
C2MASK4H 03FECC4EH, C3MASK4H 03FED24EH

	15	14	13	12	11	10	9	8
CnMASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	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.

(1/4)

After reset: 0000H R/W Address: C0CTRL 03FEC050H, C1CTRL 03FEC650H
C2CTRL 03FECC50H, C3CTRL 03FED250H

(a) Read

	15	14	13	12	11	10	9	8
CnCTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE 1	PSMODE 0	OPMODE 2	OPMODE 1	OPMODE 0

(a) Write

	15	14	13	12	11	10	9	8
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.

Remark
- 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
- 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

(2/4)

TSTAT	Transmission status bit
0	Transmission is stopped.
1	Transmission is in progress.

Remark
- 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
- 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. The CCERC bit is read-only in the CAN sleep mode or CAN stop mode.

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.

Remarks 1. The AL bit is valid only in the single-shot mode.

2. The AL bit is read-only in the CAN sleep mode or CAN stop mode.

(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. 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.

(4/4)

(b) Write

Set CCERC	Setting of CCERC bit
1	CCERC bit is set to 1.
Other than above	CCERC bit is not changed.

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.

After reset: 00H R/W Address: C0LEC 03FEC052H, C1LEC 03FEC652H
C2LEC 03FECC52H, C3LEC 03FED252H

	7	6	5	4	3	2	1	0
CnLEC	0	0	0	0	0	LEC2	LEC1	LEC0

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.

After reset: 00H R Address: C0INFO 03FEC053H, C1INFO 03FEC653H
C2INFO 03FECC53H, C3INFO 03FED253H

	7	6	5	4	3	2	1	0
CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

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.

After reset: 0000H R Address: C0ERC 03FEC054H, C1ERC 03FEC654H
C2ERC 03FECC54H, C3ERC 03FED254H

CnERC	15	14	13	12	11	10	9	8
	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	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.

(1/2)

After reset: 0000H R/W Address: C0IE 03FEC056H, C1IE 03FEC656H
C2IE 03FECC56H, C3IE 03FED256H

(a) Read

	15	14	13	12	11	10	9	8
CnIE	0	0	0	0	0	0	0	0
	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
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.

(2/2)

(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.

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.

After reset: 0000H R/W Address: C0INTS 03FEC058H, C1INTS 03FEC658H
C2INTS 03FECC58H, C3INTS 03FED258H

(a) Read

	15	14	13	12	11	10	9	8
CnINTS	0	0	0	0	0	0	0	0
	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
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	Wakeup 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.

(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)

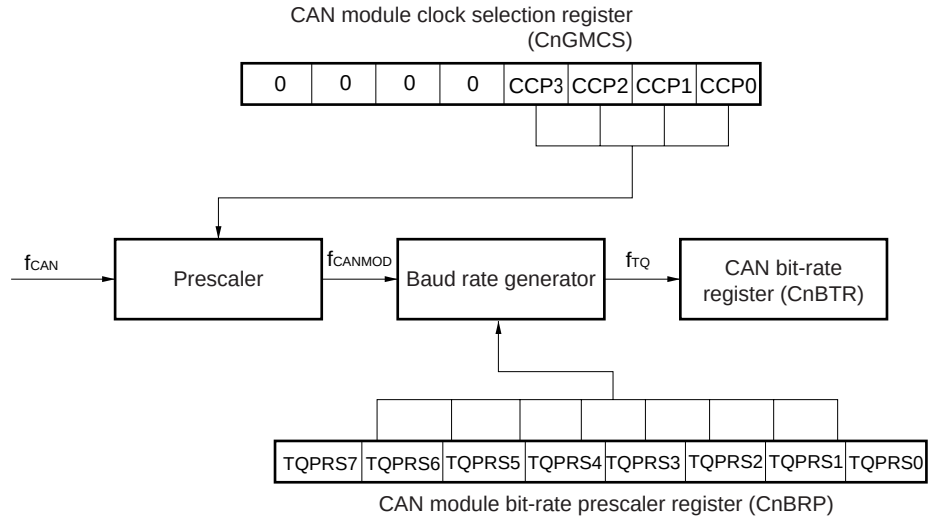
The CnBRP register is used to select the CAN protocol layer base clock (f_{TQ}). The communication baud rate is set to the CnBTR register.

After reset: FFH R/W Address: C0BRP 03FEC05AH, C1BRP 03FEC65AH
C2BRP 03FECC5AH, C3BRP 03FED25AH

	7	6	5	4	3	2	1	0
CnBRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

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 15-23. CAN Module Clock



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)

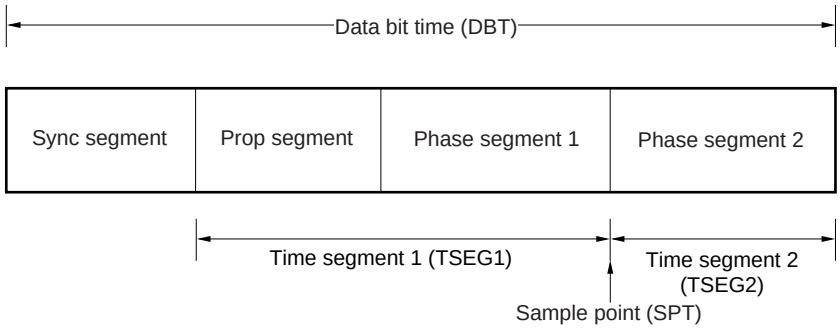
The CnBTR register is used to control the data bit time of the communication baud rate.

(1/2)

After reset: 370FH R/W Address: C0BTR 03FEC05CH, C1BTR 03FEC65CH
C2BTR 03FECC5CH, C3BTR 03FED25CH

	15	14	13	12	11	10	9	8
CnBTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

Figure 15-24. Data Bit Time



(2/2)

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)

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.

After reset: Undefined R Address: C0LIPT 03FEC05EH, C1LIPT 03FEC65EH
C2LIPT 03FECC5EH, C3LIPT 03FED25EH

	7	6	5	4	3	2	1	0
CnLIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

★

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.

(1/2)

After reset: xx02H R/W Address: C0RGPT 03FEC060H, C1RGPT 03FEC660H
C2RGPT 03FECC60H, C3RGPT 03FED260H

(a) Read

	15	14	13	12	11	10	9	8
CnRGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	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
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	All the message buffer numbers that are recorded are preserved except the message buffer number recorded last[Note 2]. The number of the message buffer in which a new data frame or remote frame has been received and stored is recorded to the receive history list, by overwriting the message buffer number that was recorded last (the receive history list does not have a vacant element).

Notes 1. The read value of RGPT0 to 7 is invalid when RHPM = 1.

2. If no new data frame or remote frame is received and stored in a message buffer after the ROVF bit has been set, the message buffer number last recorded to the receive history list is preserved.

(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.

After reset: Undefined R Address: C0LOPT 03FEC062H, C1LOPT 03FEC662H
C2LOPT 03FECC62H, C3LOPT 03FED262H

	7	6	5	4	3	2	1	0
CnLOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

★

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.

(1/2)

After reset: xx02H R/W Address: C0TGPT 03FEC064H, C1TGPT 03FEC664H
C2TGPT 03FECC64H, C3TGPT 03FED264H

(a) Read

	15	14	13	12	11	10	9	8
CnTGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

	15	14	13	12	11	10	9	8
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	All the message buffer numbers that are recorded are preserved except the message buffer number recorded last[Note 2]. The number of the message buffer to which a new data frame or remote frame has been transmitted is recorded to the transmit history list, by overwriting the message buffer number that was recorded last (the transmit history list does not have a vacant element).

Notes 1. The read value of TGPT0 to TGPT7 is invalid when THPM = 1.3.

2. If no new data frame or remote frame is transmitted after the TOVF bit has been set, the message buffer number last recorded to the transmit history list is preserved.

Remark Transmission from message buffers 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

(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.

(1/2)

After reset: 0000H R/W Address: C0TS 03FEC066H, C1TS 03FEC666H
C2TS 03FECC66H, C3TS 03FED266H

(a) Read

	15	14	13	12	11	10	9	8
CnTS	0	0	0	0	0	0	0	0
	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
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.

(2/2)

(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 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.

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.

(1/2)

After reset: Undefined R/W Address: See **Table 15-16**.

CnMDATA01m

15	14	13	12	11	10	9	8
MDATA01 15	MDATA01 14	MDATA01 13	MDATA01 12	MDATA01 11	MDATA01 10	MDATA01 9	MDATA01 8
7	6	5	4	3	2	1	0
MDATA01 7	MDATA01 6	MDATA01 5	MDATA01 4	MDATA01 3	MDATA01 2	MDATA01 1	MDATA01 0

CnMDATA0m

7	6	5	4	3	2	1	0
MDATA0 7	MDATA0 6	MDATA0 5	MDATA0 4	MDATA0 3	MDATA0 2	MDATA0 1	MDATA0 0

CnMDATA1m

7	6	5	4	3	2	1	0
MDATA1 7	MDATA1 6	MDATA1 5	MDATA1 4	MDATA1 3	MDATA1 2	MDATA1 1	MDATA1 0

CnMDATA23m

15	14	13	12	11	10	9	8
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

CnMDATA2m

7	6	5	4	3	2	1	0
MDATA2 7	MDATA2 6	MDATA2 5	MDATA2 4	MDATA2 3	MDATA2 2	MDATA2 1	MDATA2 0

CnMDATA3m

7	6	5	4	3	2	1	0
MDATA3 7	MDATA3 6	MDATA3 5	MDATA3 4	MDATA3 3	MDATA3 2	MDATA3 1	MDATA3 0

(2/2)

CnMDATA45m

15	14	13	12	11	10	9	8
MDATA45 15	MDATA45 14	MDATA45 13	MDATA45 12	MDATA45 11	MDATA45 10	MDATA45 9	MDATA45 8
7	6	5	4	3	2	1	0
MDATA45 7	MDATA45 6	MDATA45 5	MDATA45 4	MDATA45 3	MDATA45 2	MDATA45 1	MDATA45 0

CnMDATA4m

7	6	5	4	3	2	1	0
MDATA4 7	MDATA4 6	MDATA4 5	MDATA4 4	MDATA4 3	MDATA4 2	MDATA4 1	MDATA4 0

CnMDATA5m

7	6	5	4	3	2	1	0
MDATA5 7	MDATA5 6	MDATA5 5	MDATA5 4	MDATA5 3	MDATA5 2	MDATA5 1	MDATA5 0

CnMDATA67m

15	14	13	12	11	10	9	8
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

CnMDATA6m

7	6	5	4	3	2	1	0
MDATA6 7	MDATA6 6	MDATA6 5	MDATA6 4	MDATA6 3	MDATA6 2	MDATA6 1	MDATA6 0

CnMDATA7m

7	6	5	4	3	2	1	0
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.

After reset: 0000xxxxB R/W Address: See **Table 15-16**.

	7	6	5	4	3	2	1	0
CnMDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

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.

(1/2)

After reset: Undefined R/W Address: See **Table 15-16**.

	7	6	5	4	3	2	1	0
CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

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

(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).

After reset: Undefined R/W Address: See **Table 15-16**.

	15	14	13	12	11	10	9	8
CnMIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
	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.

(1/2)

After reset: 00x000000 00000000B R/W Address: See **Table 15-16**.

(a) Read

	15	14	13	12	11	10	9	8
CnMCTRLm	0	0	MUC	0	0	0	0	0
	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
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.

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.

(2/2)

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.

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.

(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.

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.

15.7 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
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 15-25 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 15-26). Figure 15-25 shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 15-25. Example of Bit Setting/Clearing Operations

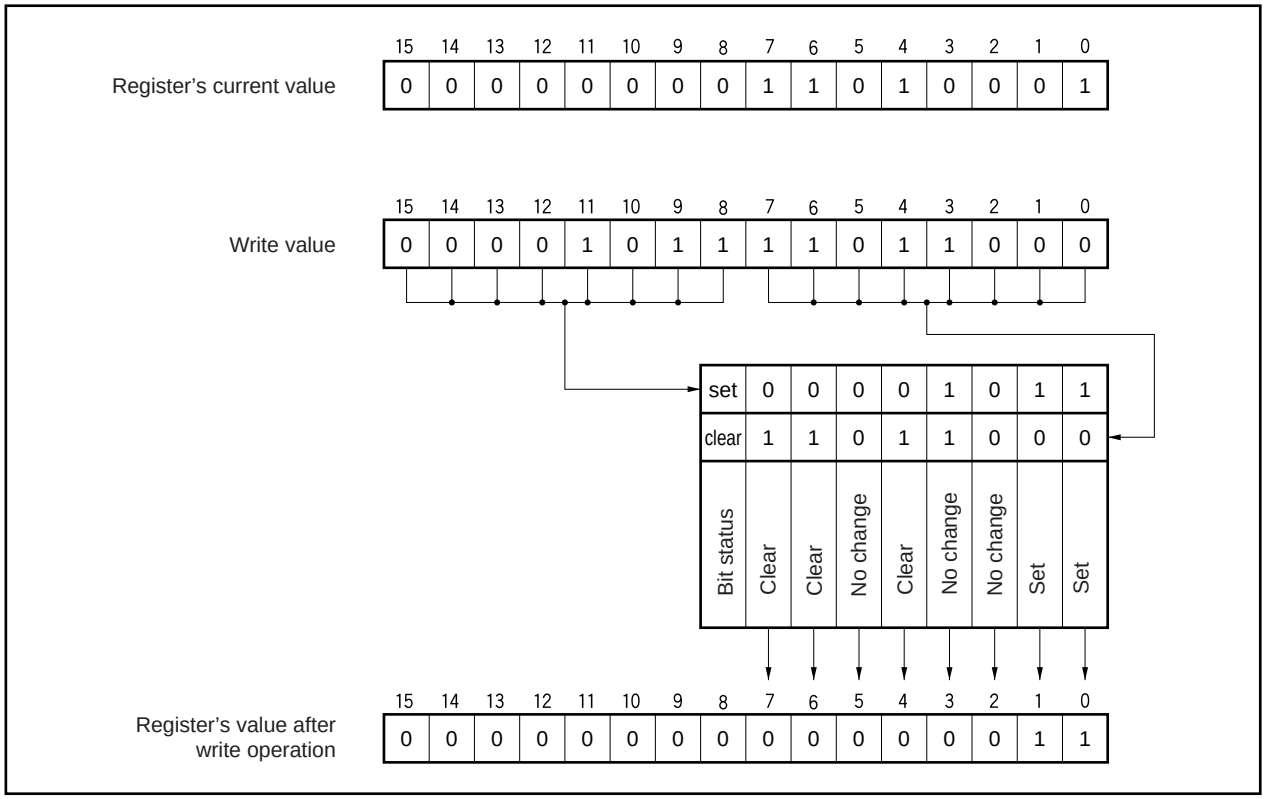


Figure 15-26. Bit Status After Bit Setting/Clearing Operations

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

15.8 CAN Controller Initialization

15.8.1 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 **15.16 Operation of CAN Controller.**

15.8.2 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.8.3 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.

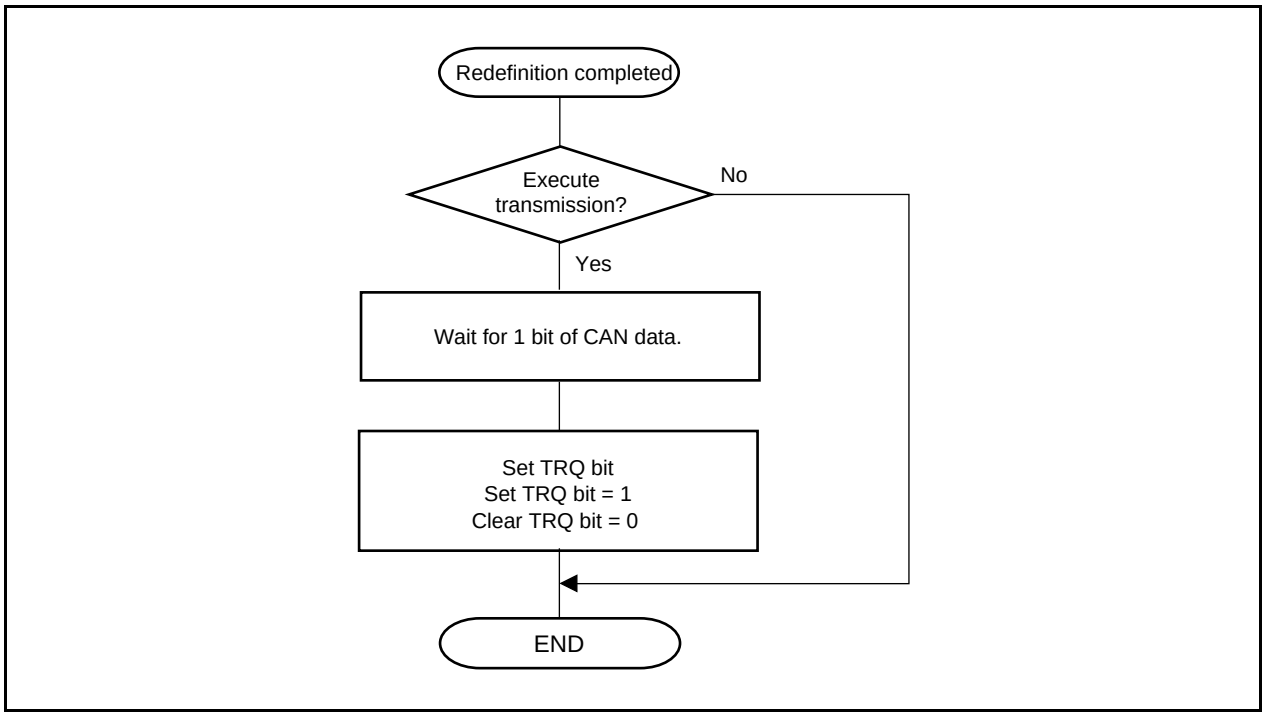
(2) To redefine message buffer during reception

Perform redefinition as shown in Figure 15-38.

(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 **15.10.4 (1) Transmission abort in normal operation mode** and **15.10.4 (2) Transmission abort in normal operation mode with automatic block transmission (ABT))**. 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 15-27. 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 Figure 15-38 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.

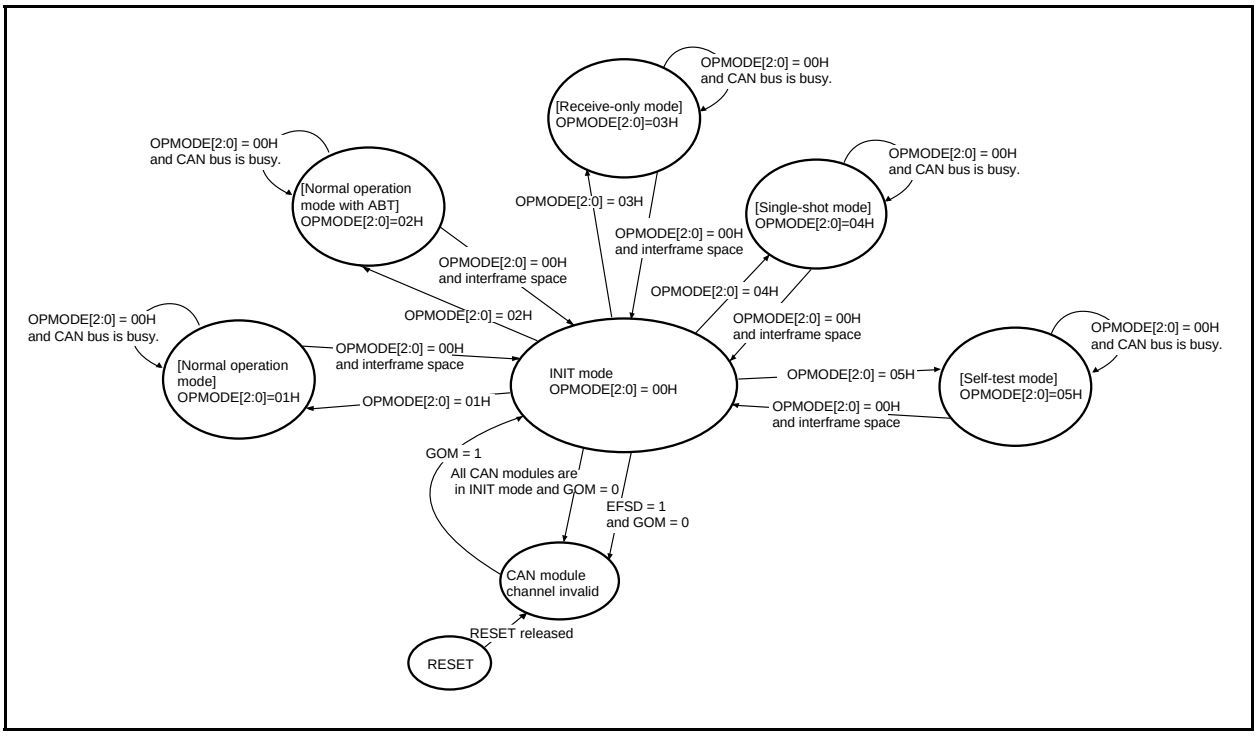
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 15-27 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

15.8.4 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 15-28. Transition to Operation Modes



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 15-36**).

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.8.5 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.9 Message Reception

15.9.1 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 message is always stored in the unmasked receive message buffer even if this unmasked receive buffer has already received a message earlier.

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

15.9.2 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.

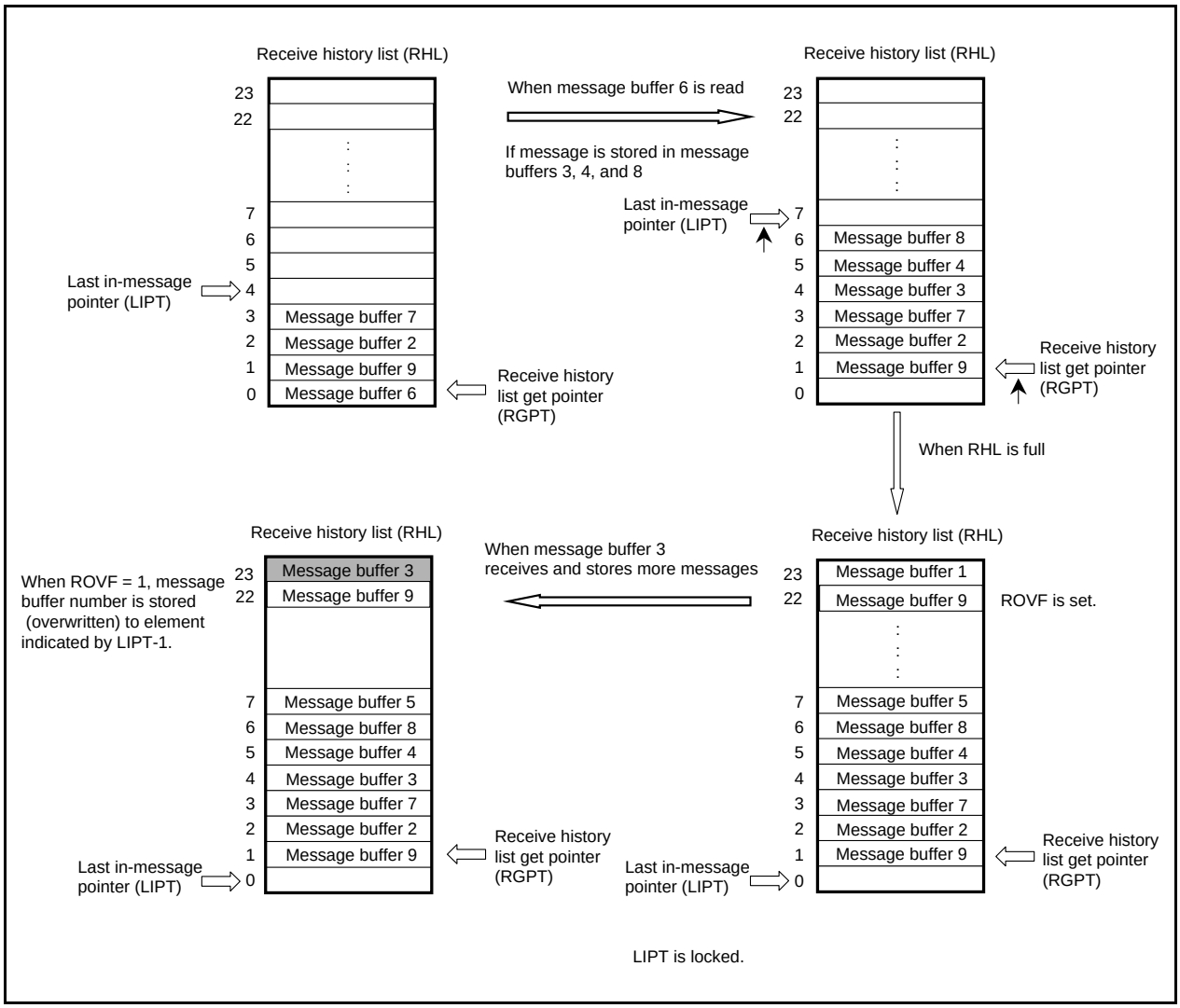
The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CnRGPT register by software, the number of a message buffer that has received and stored a data frame or 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.

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

Figure 15-29. Receive History List



15.9.3 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

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

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.

15.9.4 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.

5. The priority between MBRBs is mentioned in the table of 15.9.1 Message Reception.

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.9.5 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.10 Message Transmission

15.10.1 Message transmission

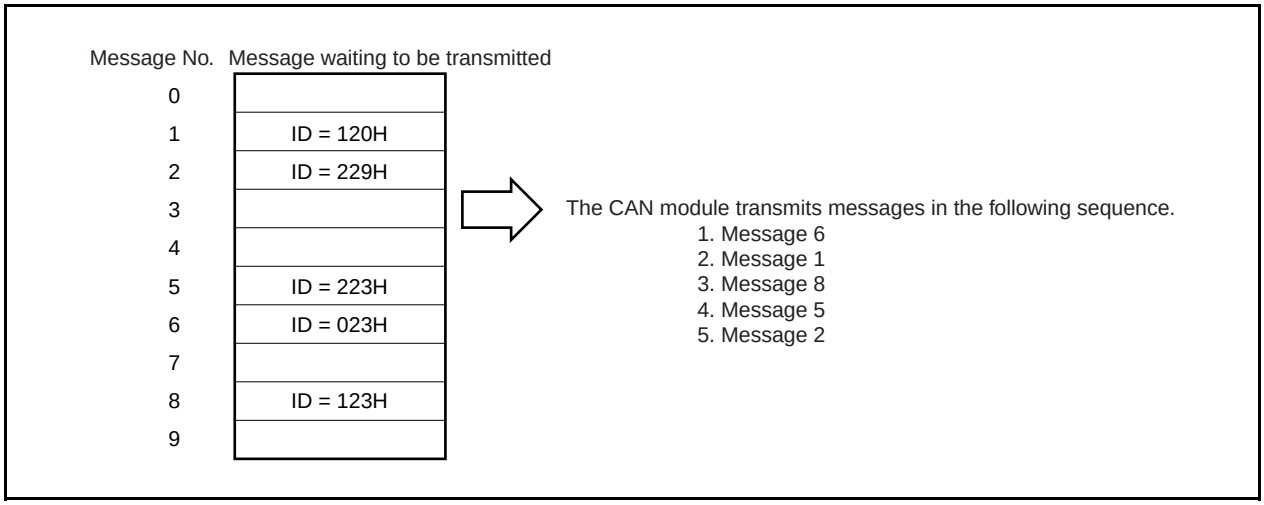
In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- 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 15-30. Message Processing Example



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 ABTTRG 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 TRQ bit is set to 1 for this buffer and for the message buffers that do not belong to the ABT message buffer group, a conflict occurs. When messages are successively transmitted from the automatic block transmission area (message buffers 0 to 7), therefore, the priority of the transmission ID is not searched, and the messages are transmitted sequentially, starting from the buffer with the lowest number. However, the priority among automatic block transmission messages and message buffers other than those in the automatic block transmission area is in compliance with the above rule.
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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.10.2 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.

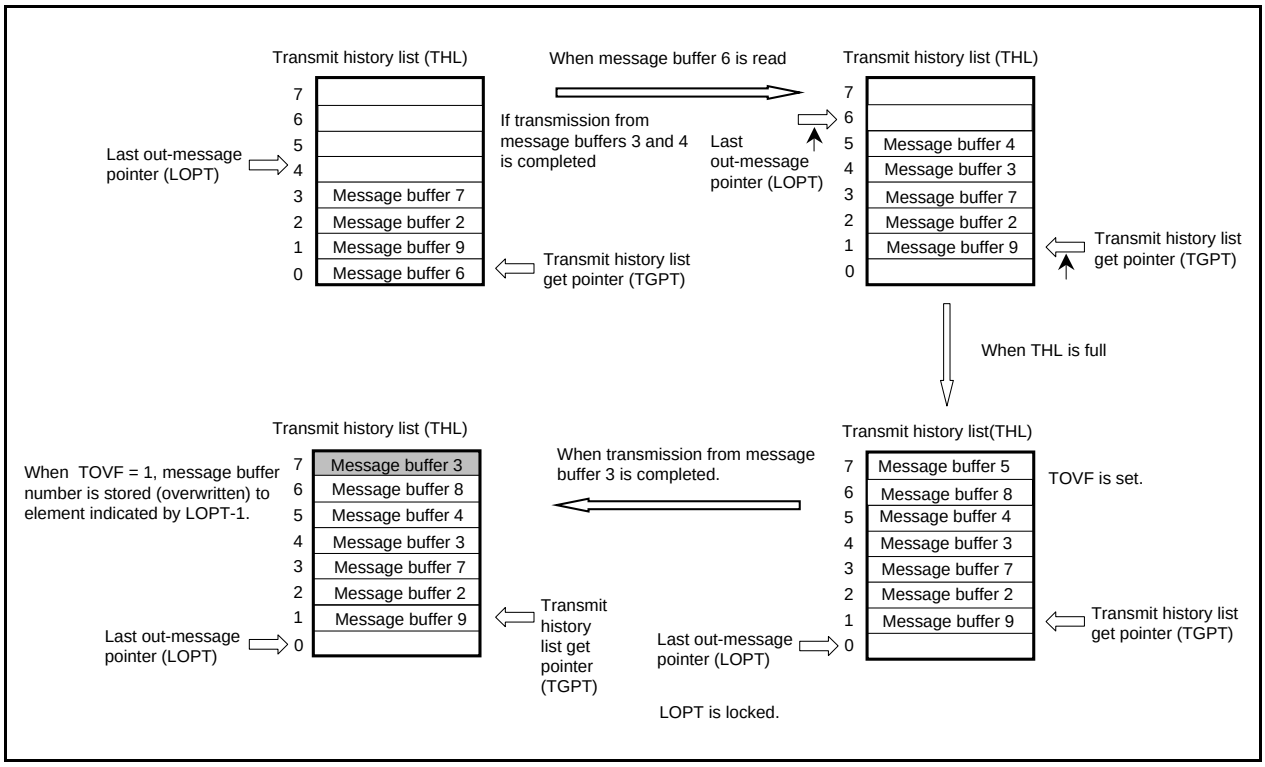
The CnLOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the CnLOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer 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.

Figure 15-31. Transmit History List



15.10.3 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. **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).**

15.10.4 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 15-44**).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the CnGMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the CnGMABT register = 0, clear the TRQ bit of the CnMCTRLm register to 0. 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 15-46).

(3) 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 15-45**).

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.

15.10.5 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.

15.11 Power Saving Modes

15.11.1 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.

(i) 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

(ii) 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.

(iii) 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 an 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).

(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 If this falling edge is at the SOF of a receive frame, no receive operation, including returning ACK, is performed on that frame. No receive operation is performed on the subsequent frames either, unless the clock is supplied to the CAN macro.

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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.11.2 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 PSMODE[1:0] bits of the CnCTRL register.

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 CAN sleep mode before entering the initialization mode.

15.11.3 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 wakeup 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 wakeup 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).

15.12 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 15-20. 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 wakeup 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

15.13 Diagnosis Functions and Special Operational Modes

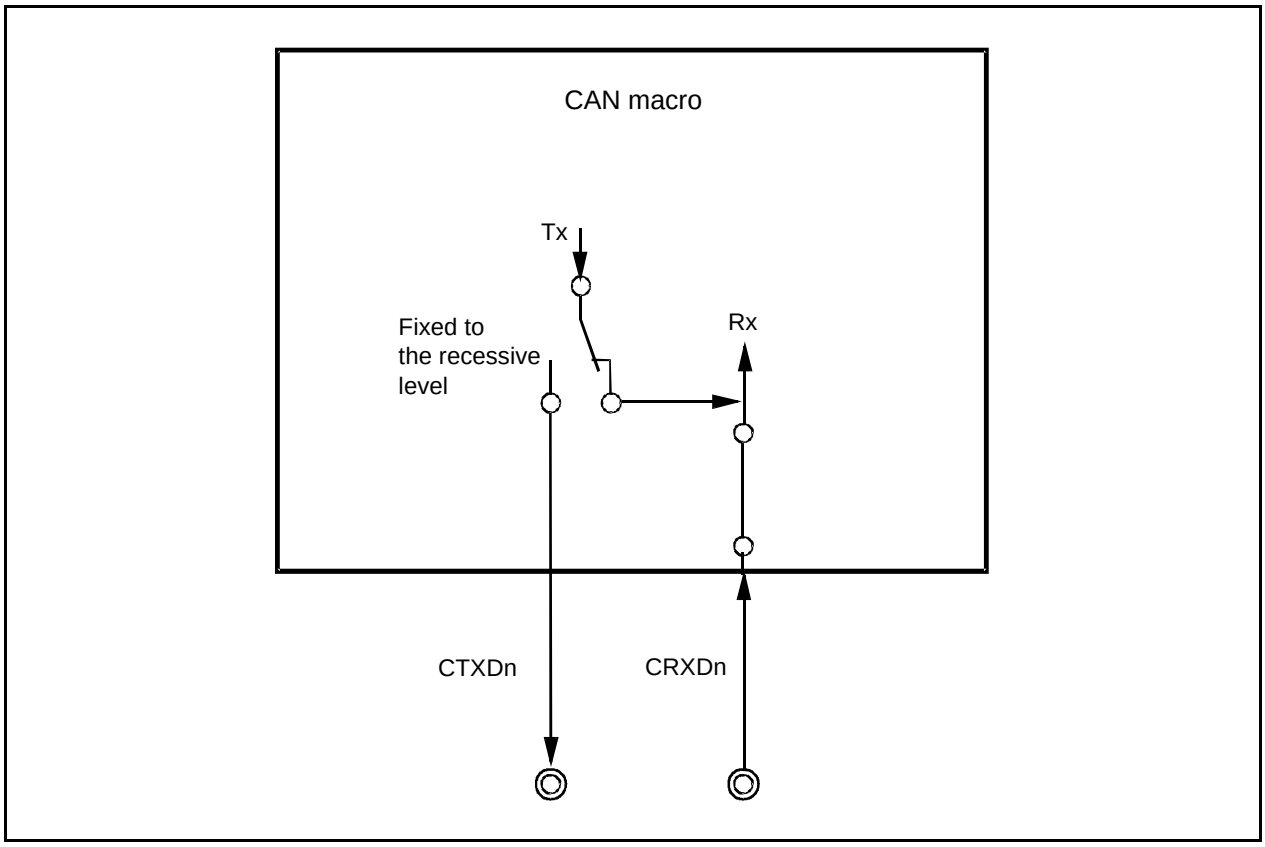
The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of special CAN communication methods.

15.13.1 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 15-32. 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.

15.13.2 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.)

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 influence the operation of re-transmission upon arbitration loss in the other operation modes.

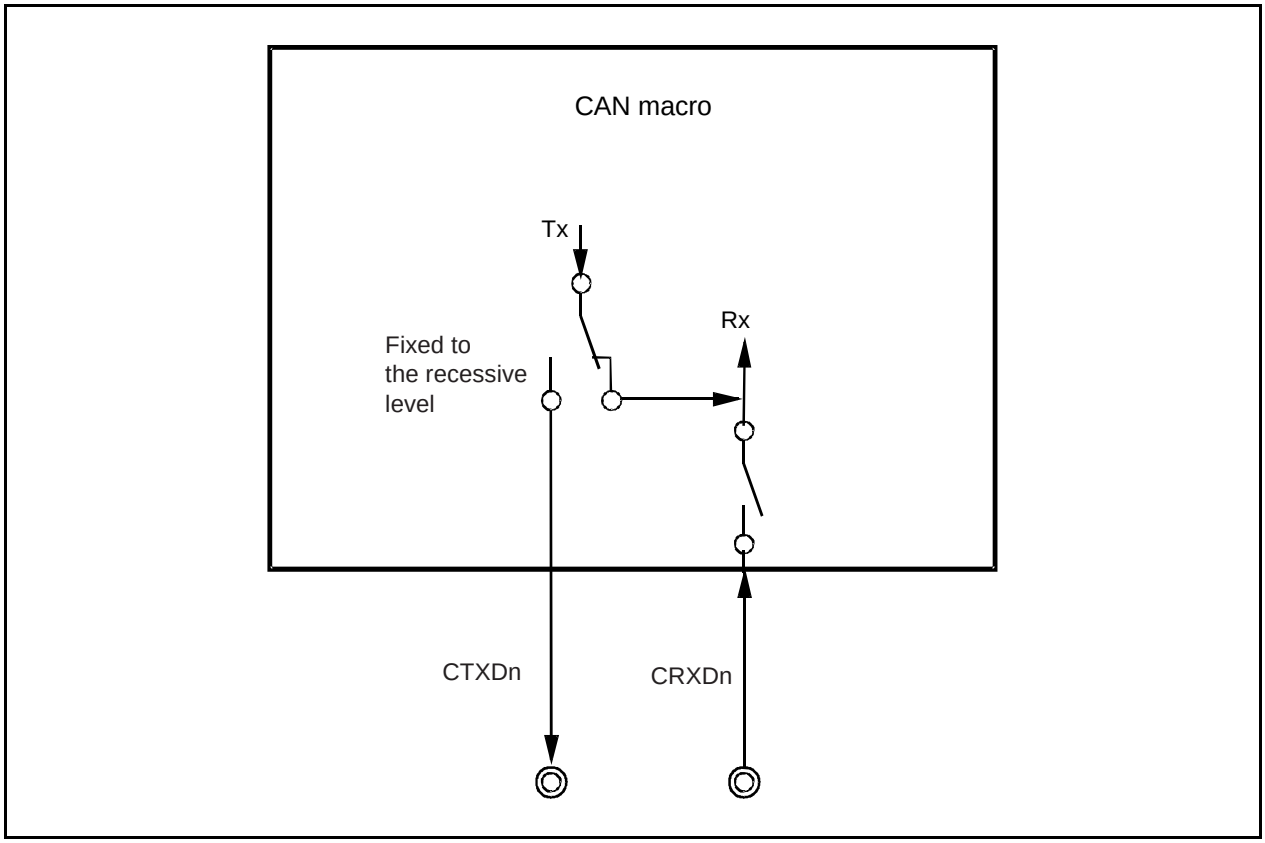
15.13.3 Self-test mode

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 15-33. CAN Module Terminal Connection in Self-Test Mode



15.14 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.

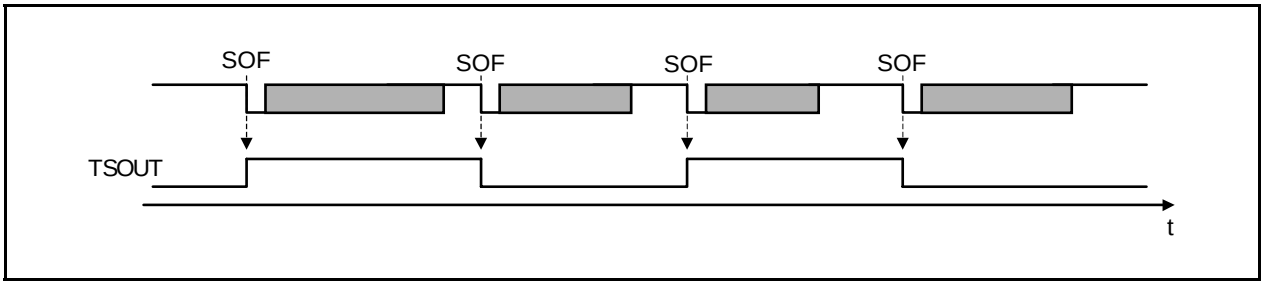
15.14.1 Time stamp function

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 15-34. 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 (μPD70F3231, μPD70F3232, μPD70F3233)

n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)

n = 0 to 3 (μPD70F3238, μPD70F3239)

m = 0 to 31

15.15 Baud Rate Settings

15.15.1 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 15-21 shows the combinations of bit rates that satisfy the above conditions.

Table 15-21. 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

Table 15-21. 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	2	7	7	1000	110	58.8
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

Table 15-21. 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]	
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
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

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 15-21 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.

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

15.15.2 Representative examples of baud rate settings

Tables 15-22 and 15-23 show representative examples of baud rate settings.

Table 15-22. 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

Caution The values in Table 15-22 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.

Table 15-22. 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	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
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 15-22 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

Table 15-23. 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

Caution The values in Table 15-23 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 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

Table 15-23. 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 SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
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
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

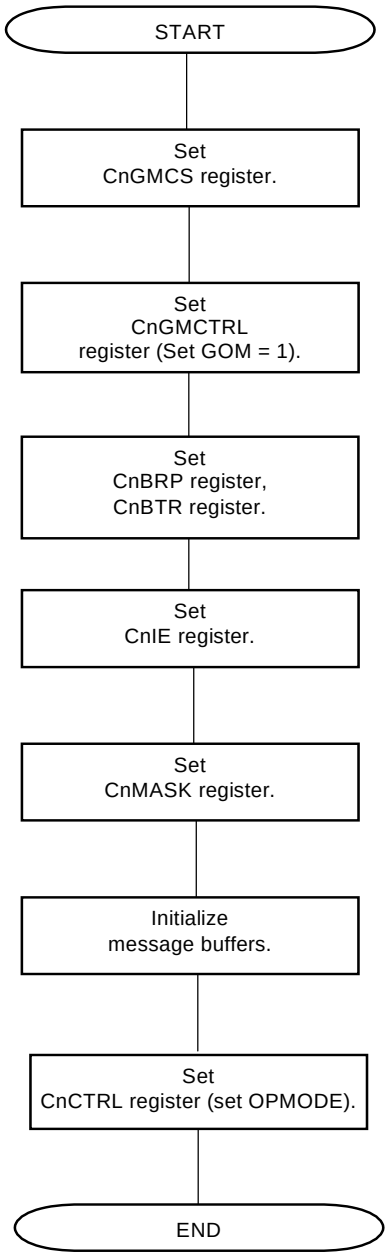
Caution The values in Table 15-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)

15.16 Operation of CAN Controller

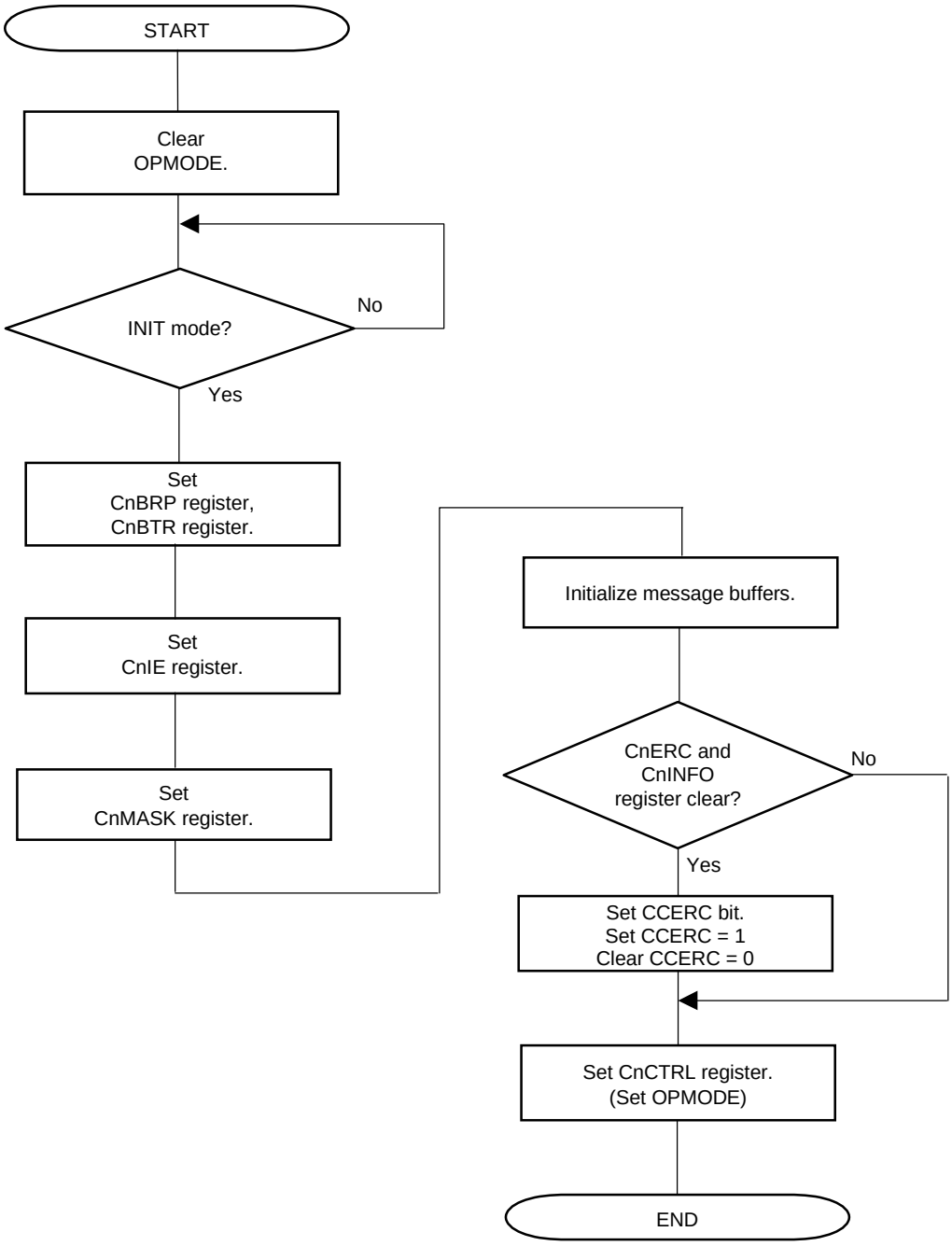
Remark . n = 0 (μPD70F3231, μPD70F3232, μPD70F3233)
n = 0, 1(μPD70F3234, μPD70F3235, μPD70F3236, μPD70F3237)
n = 0 to 3 (μPD70F3238, μPD70F3239)
m = 0 to 31

Figure 15-35. Initialization



Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

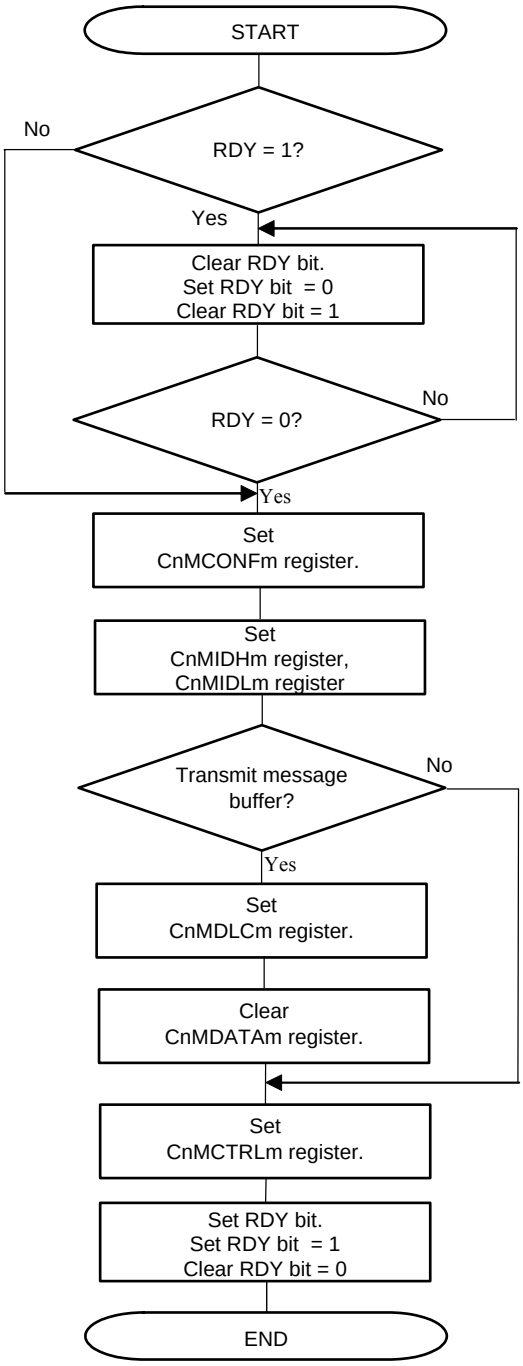
Figure 15-36. Re-initialization



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 15-37. Message Buffer Initialization



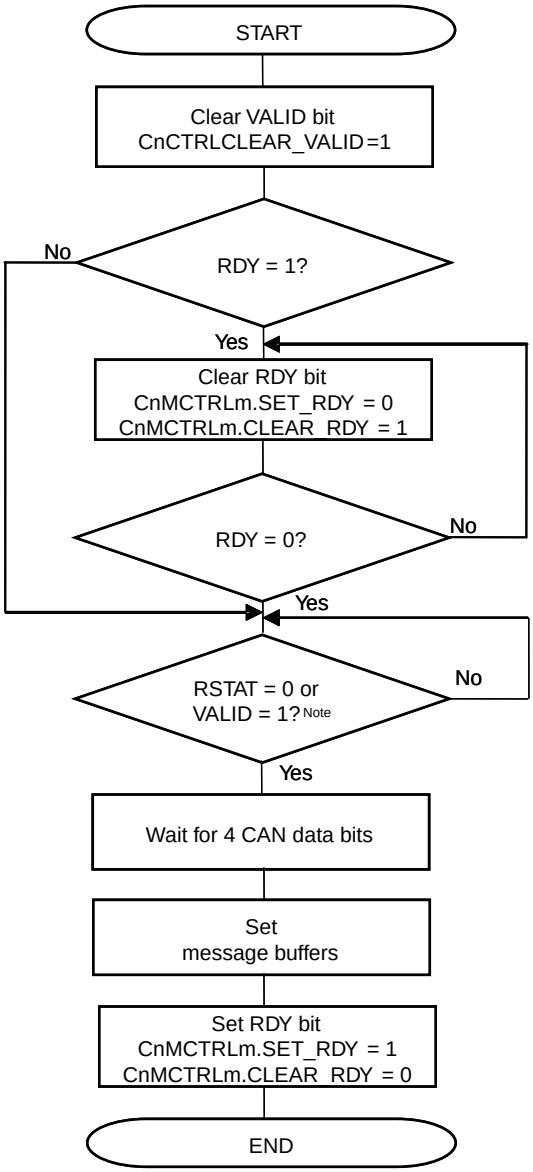
Cautions 1. Before a message buffer is initialized, the RDY bit must be cleared.

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 15-38 shows the processing for a receive message buffer (MT[2:0] bits of CnMCONFm register = 001B to 101B).

Figure 15-38. 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.

Figure 15-39 shows the processing for a transmit message buffer during transmission (MT2 to MT0 bits of CnMCONFm register = 000B).

Figure 15-39. Transmitting Message Buffer Redefinition

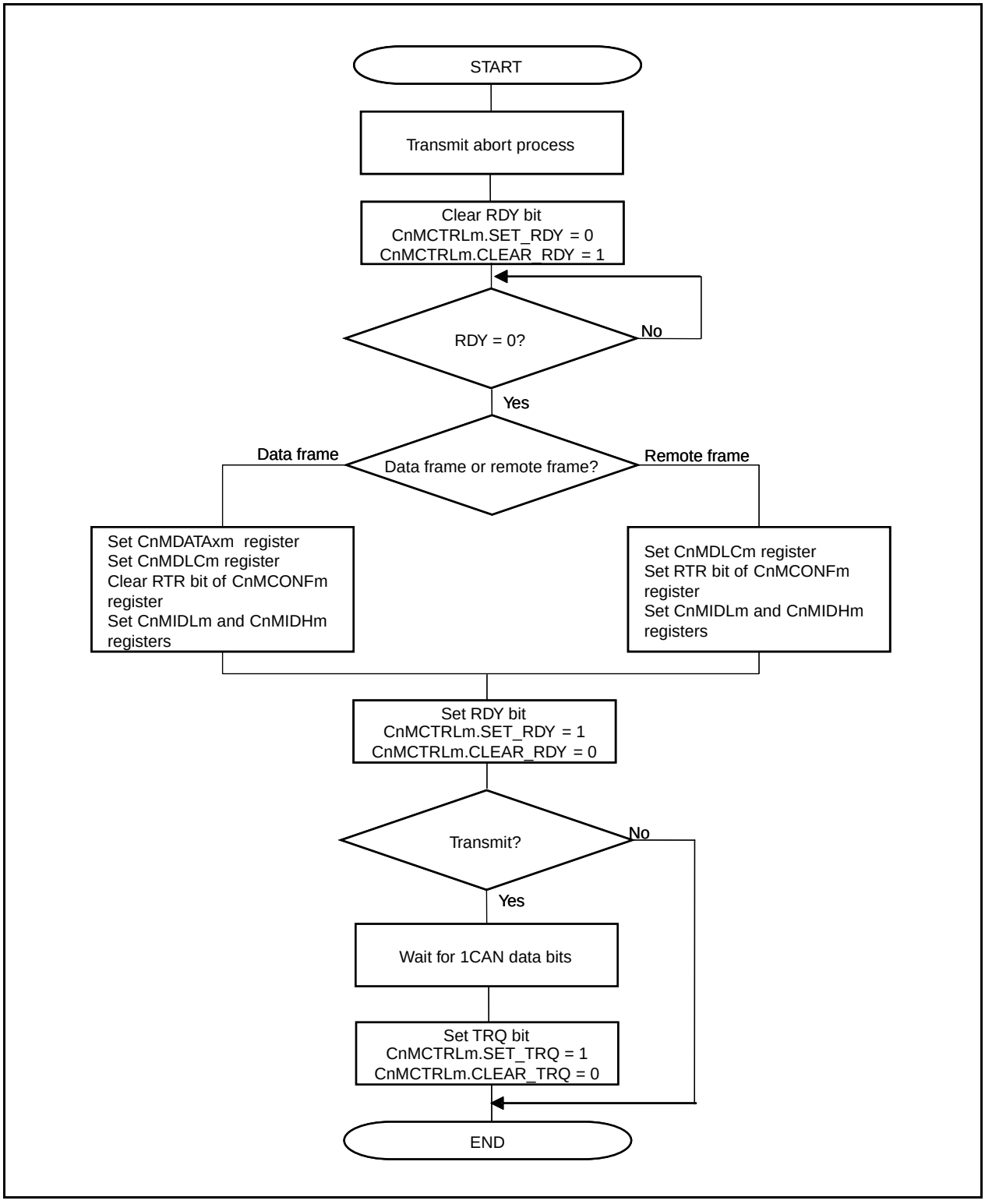
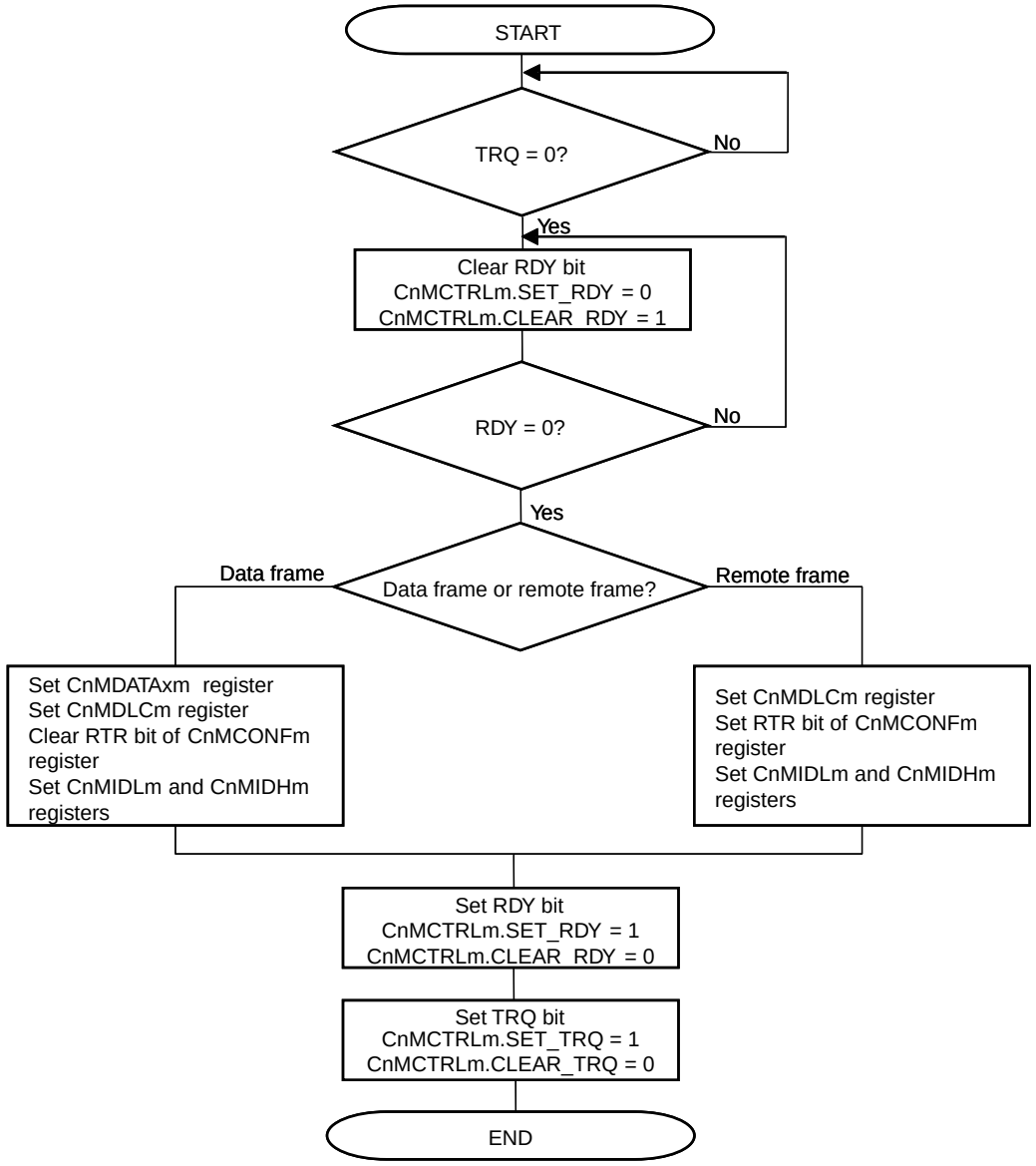


Figure 15-40 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000B).

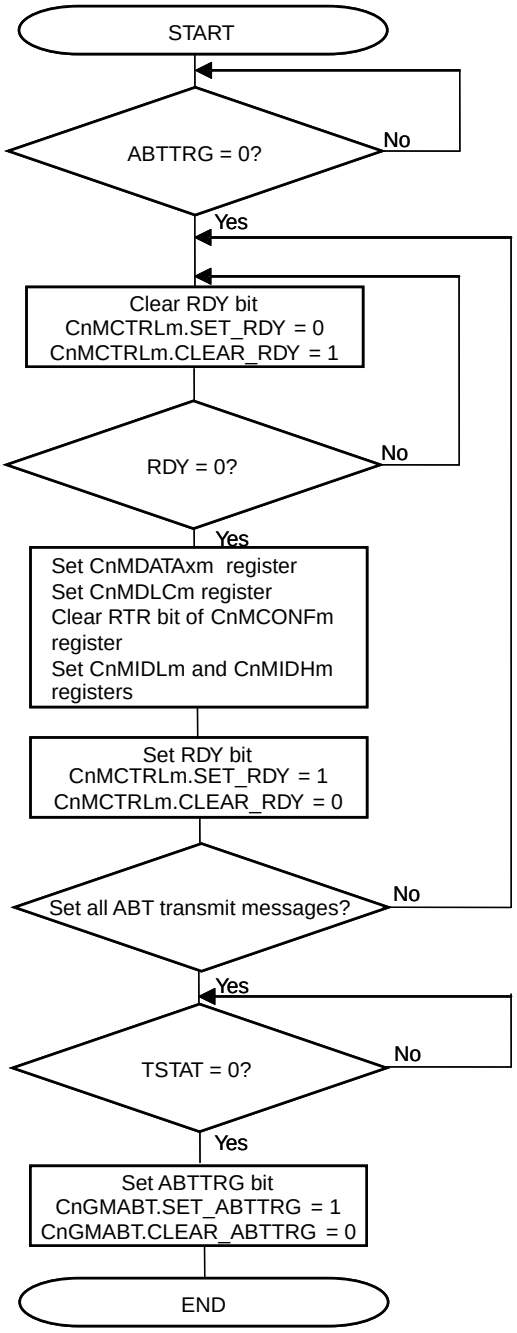
Figure 15-40. Message Transmit Processing (Normal Operation Mode)



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 15-41 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000B).

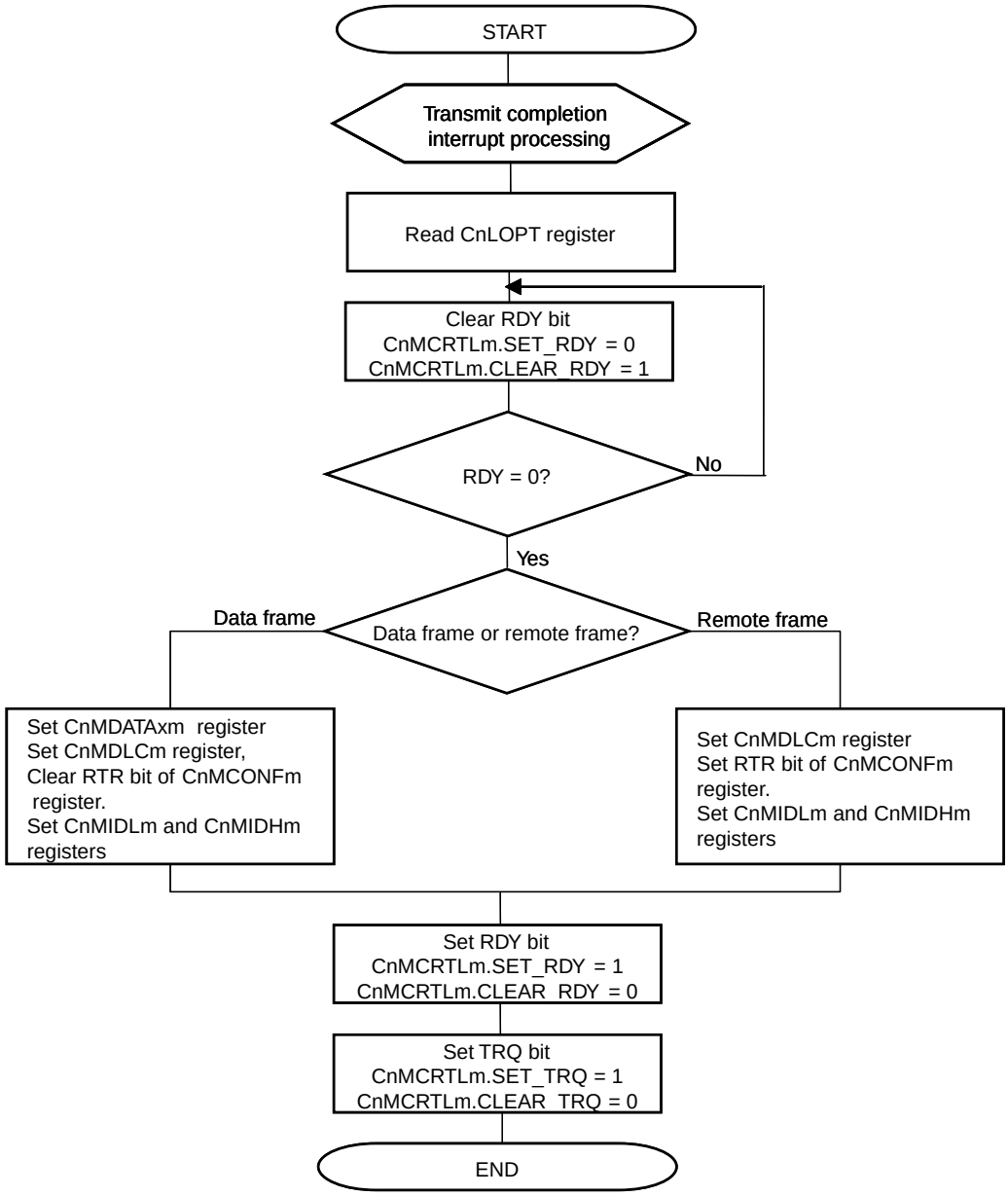
Figure 15-41. Message Transmit Processing (Normal Operation Mode with ABT)



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 15-40**.

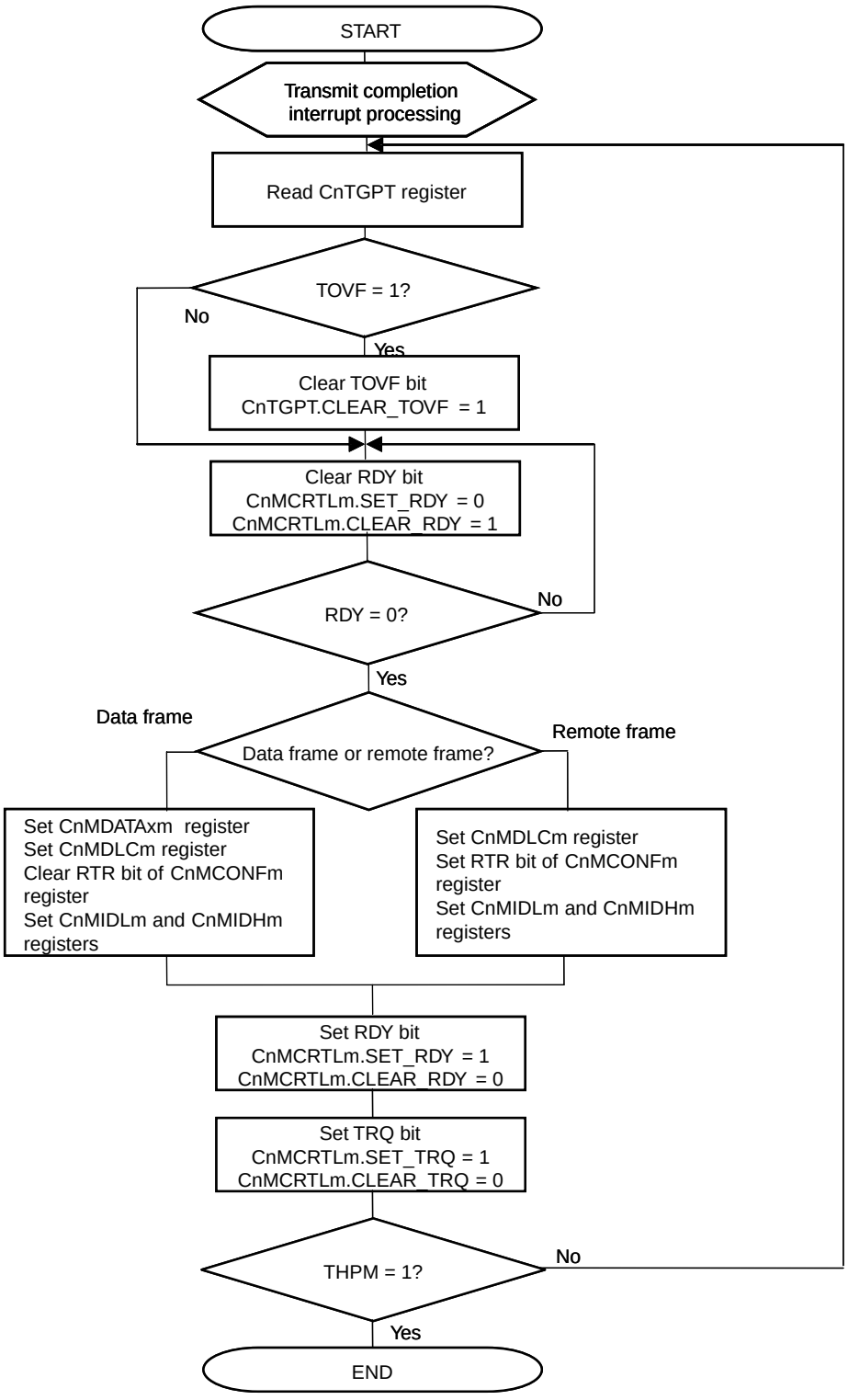
Caution Set (1) ABTTRG bit after TSTAT bit is clear (0) check TSTAT bit and set ABTTRG bit, must be processing successively.

Figure 15-42. Transmission via Interrupt (Using CnLOPT register)



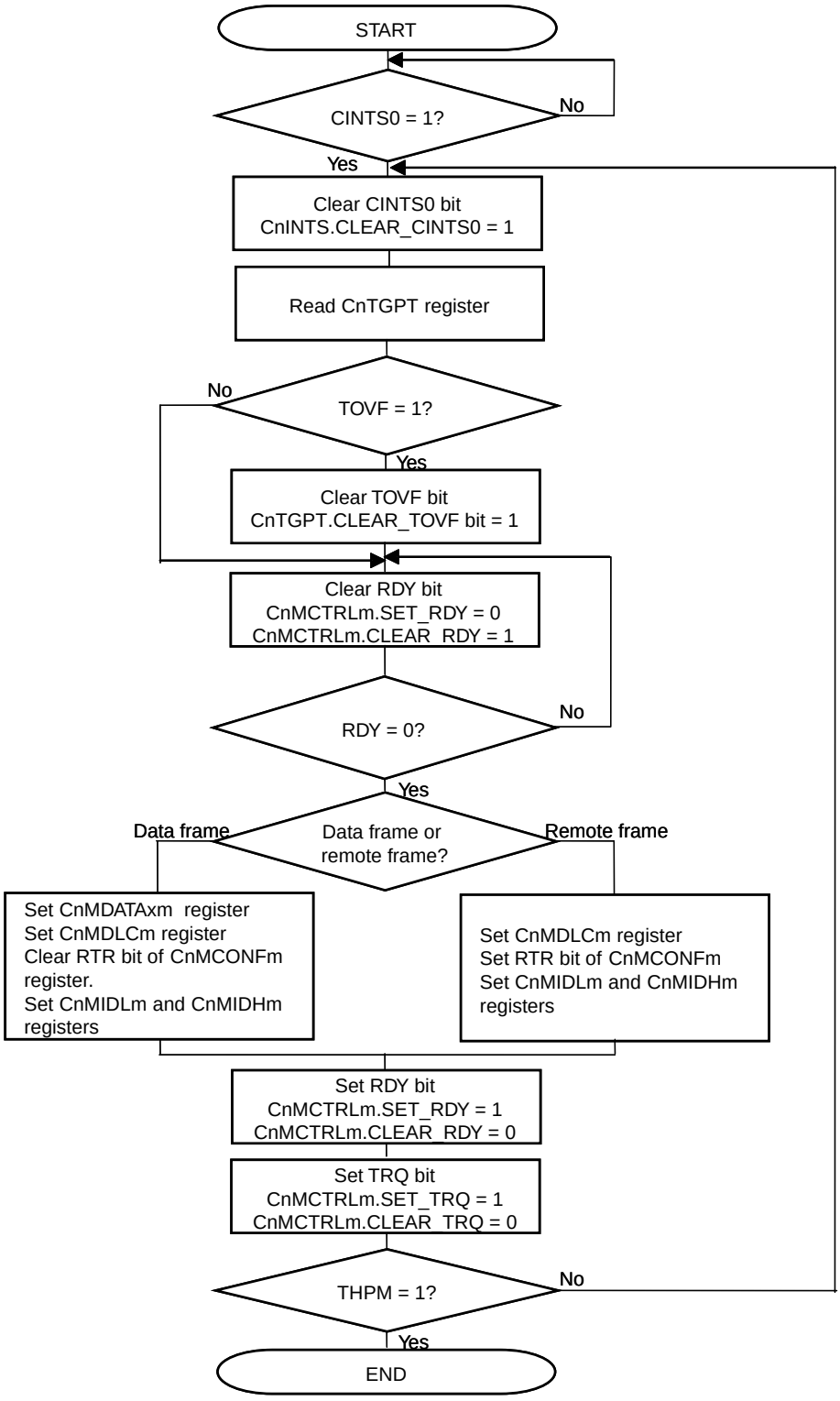
Caution The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 15-43. Transmission via Interrupt (Using CnTGPT Register)



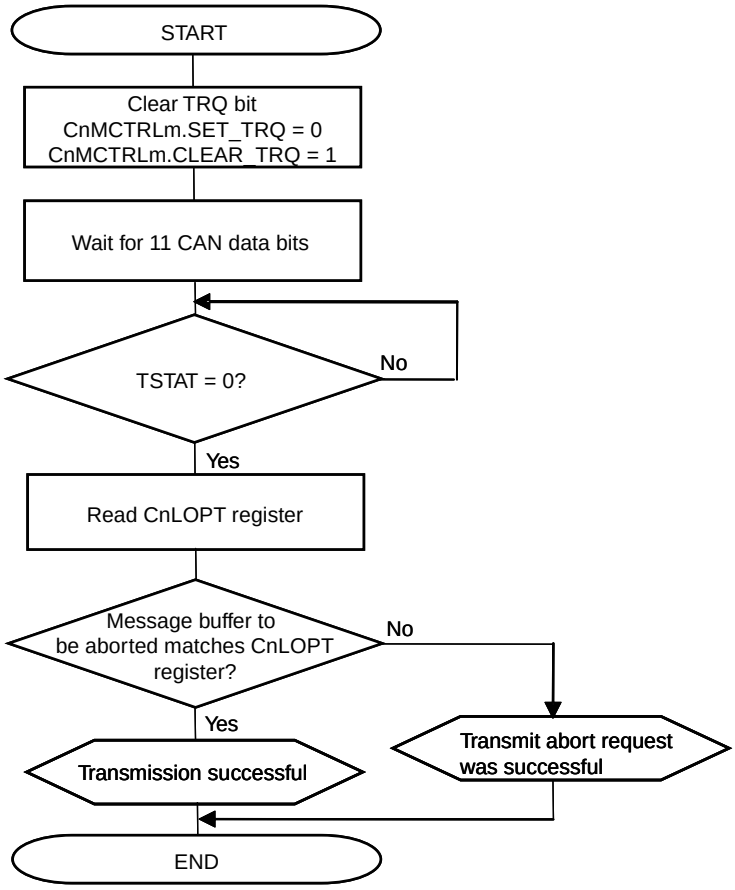
Caution The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 15-44. 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 15-45. 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.

4. In the aborting the transmission progressing, do not make a new transmission request including other message buffer.

In the normal operation with ABT, to abort transmit except transmission with ABT, using this processing flow.

Figure 15-46. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)

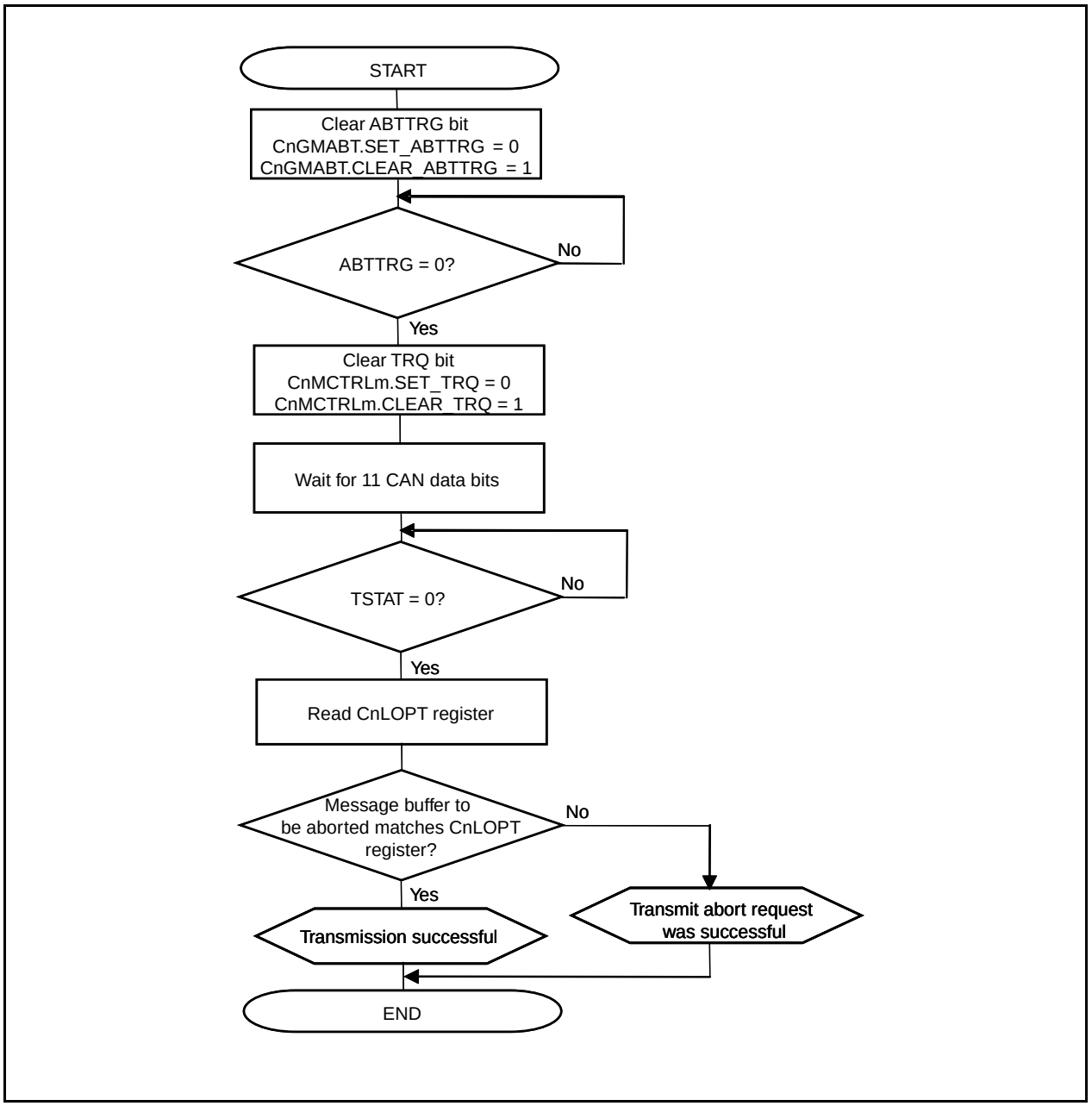
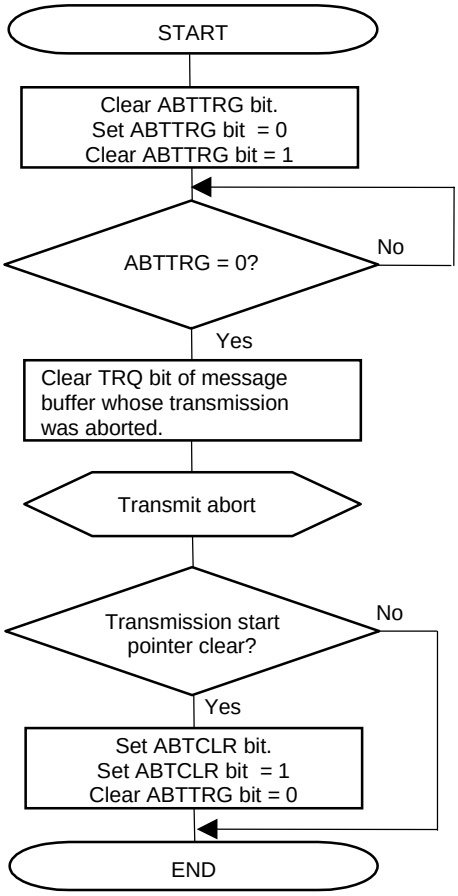


Figure 15-47 (a) shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 15-47. (a) Transmission Abort Processing (Normal Operation Mode with ABT)

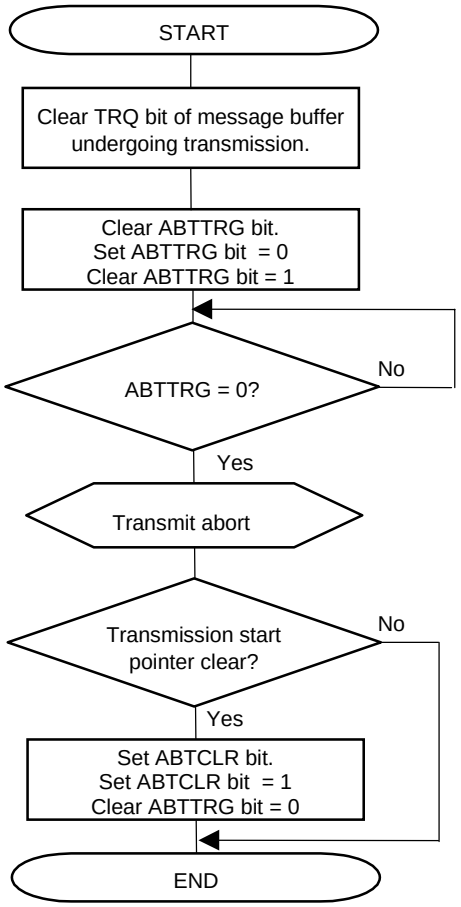


Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG is cleared following the procedure shown in Figure 15-47 (a) or (b). When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 15-46.

Figure 15-47 (b) shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

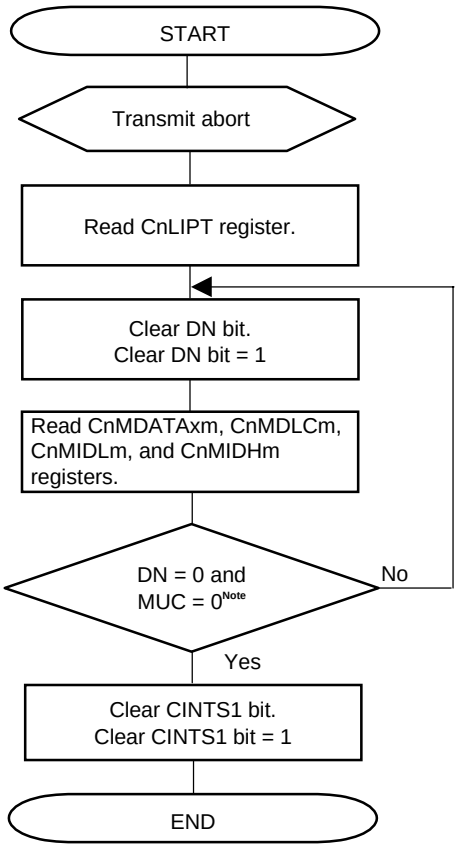
Figure 15-47. (b) Transmission Request Abort Processing (Normal Operation Mode with ABT)



Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

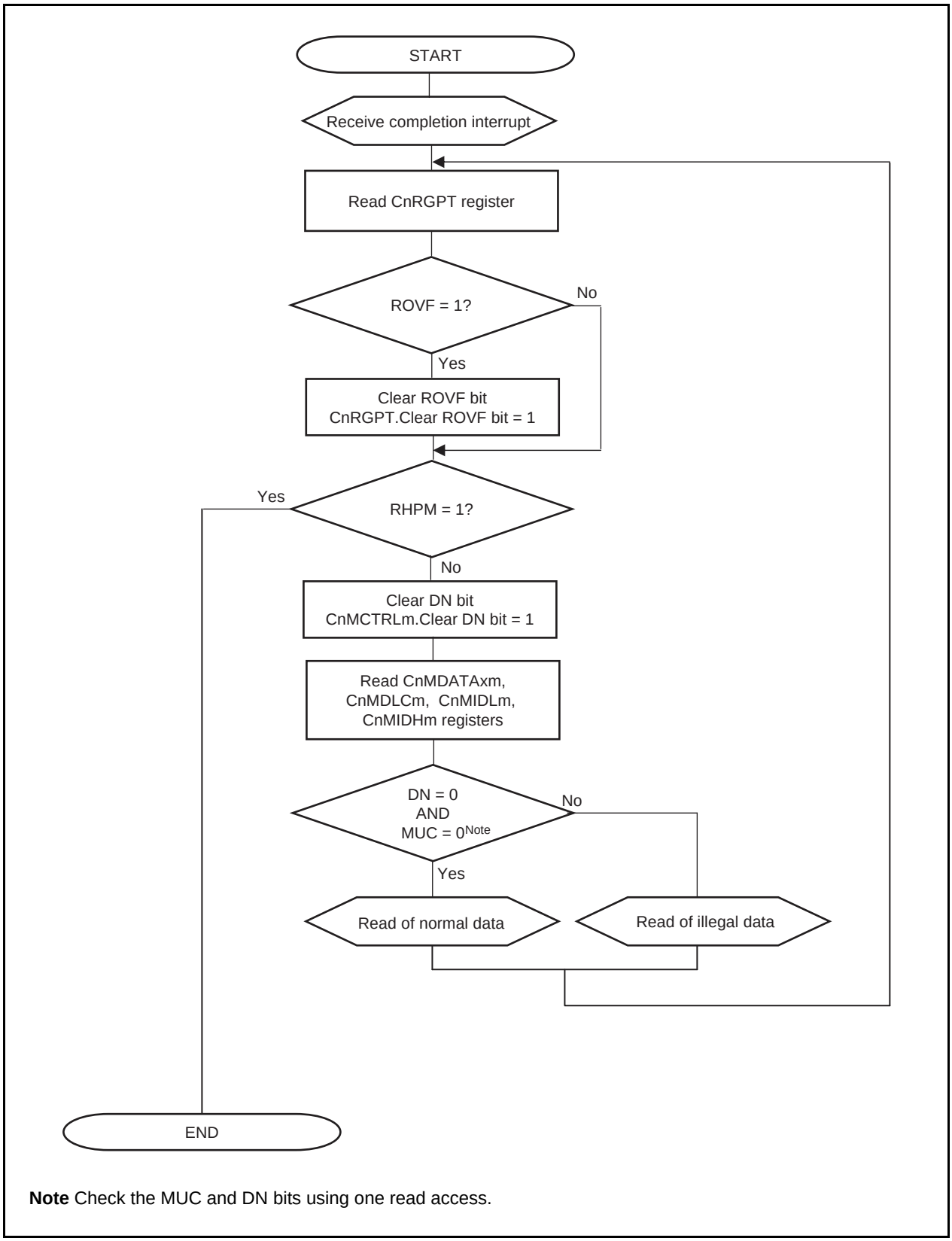
2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared following the procedure shown in Figure 15-47 (a) or (b). When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 15-46.

Figure 15-48. Reception via Interrupt (Using CnLIPT Register)



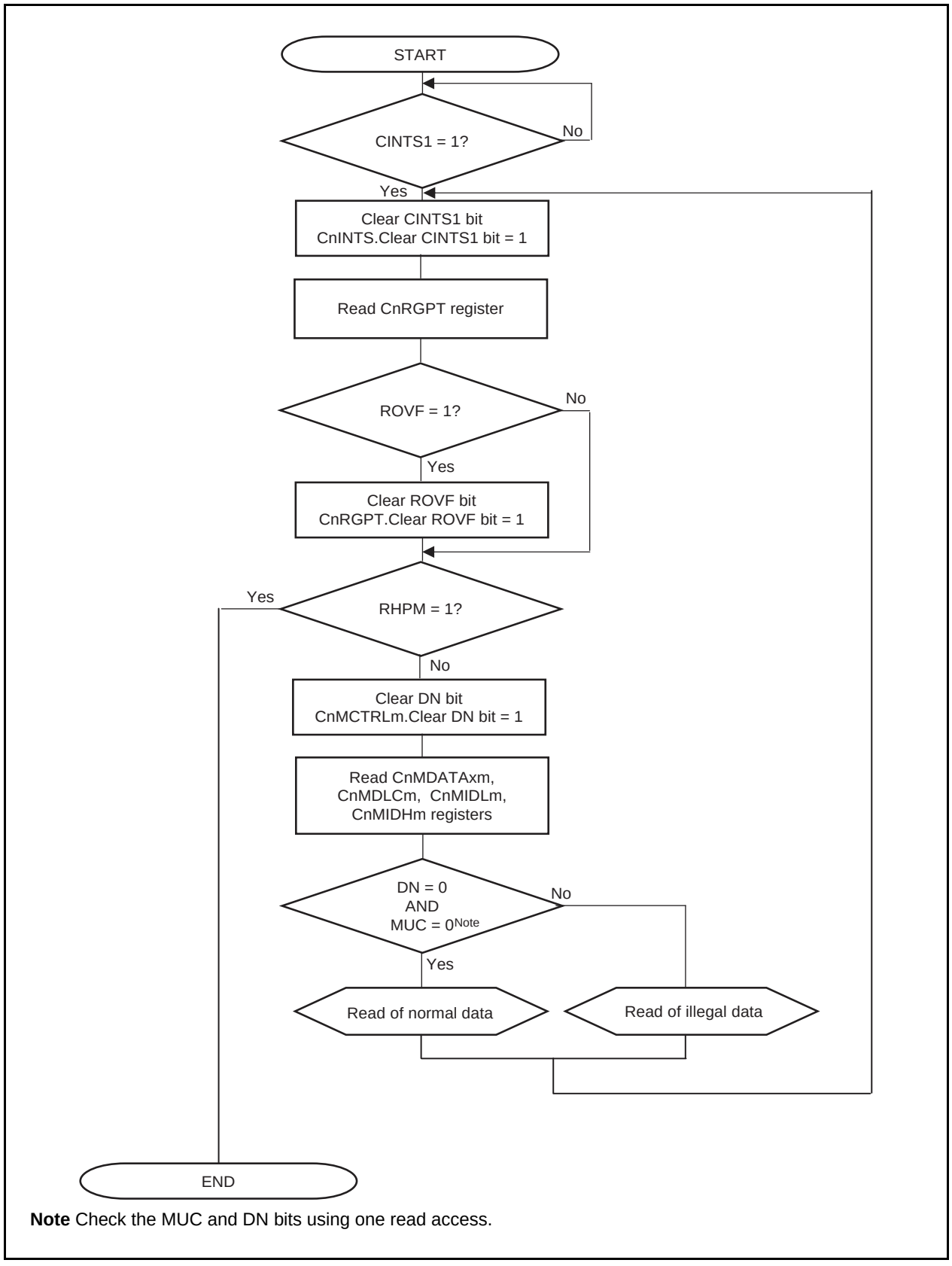
Note Check the MUC and DN bits using one read access.

★ **Figure 15-49. Reception via Interrupt (Using CnRGPT Register)**



Note Check the MUC and DN bits using one read access.

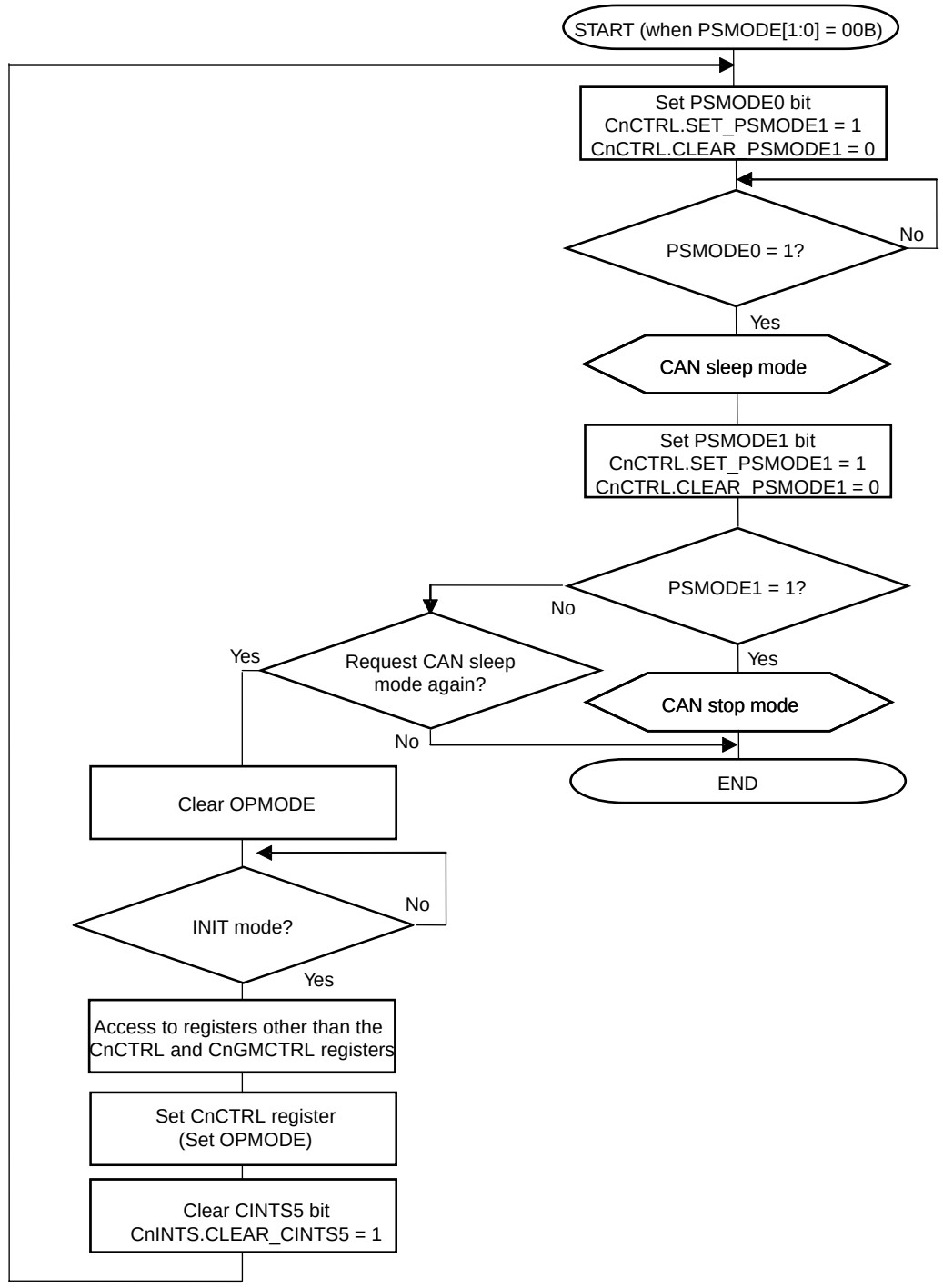
Figure 15-50. Reception via Software Polling



Note Check the MUC and DN bits using one read access.

★

Figure 15-51. Setting CAN Sleep Mode/Stop Mode



Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 15-46 and 15-47.

Figure 15-52. Clear CAN Sleep/Stop Mode

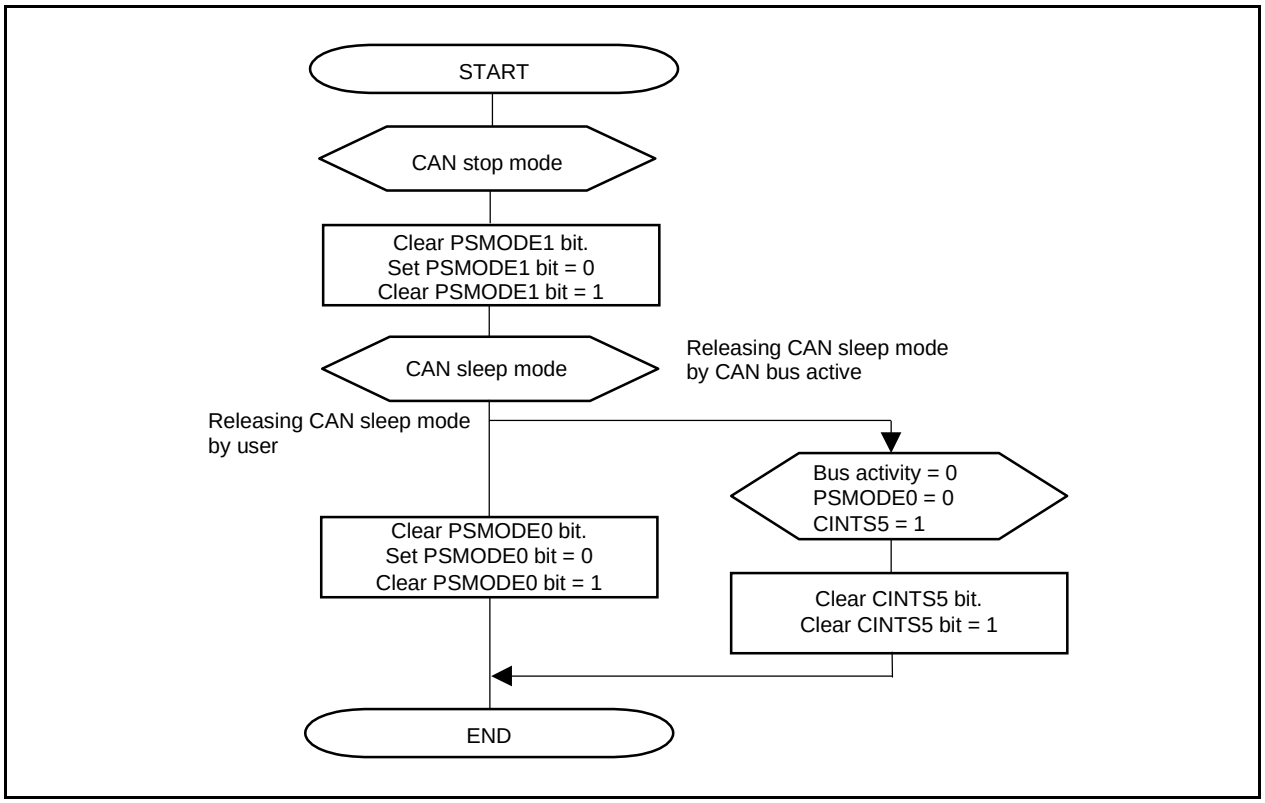


Figure 15-53. Bus-Off Recovery

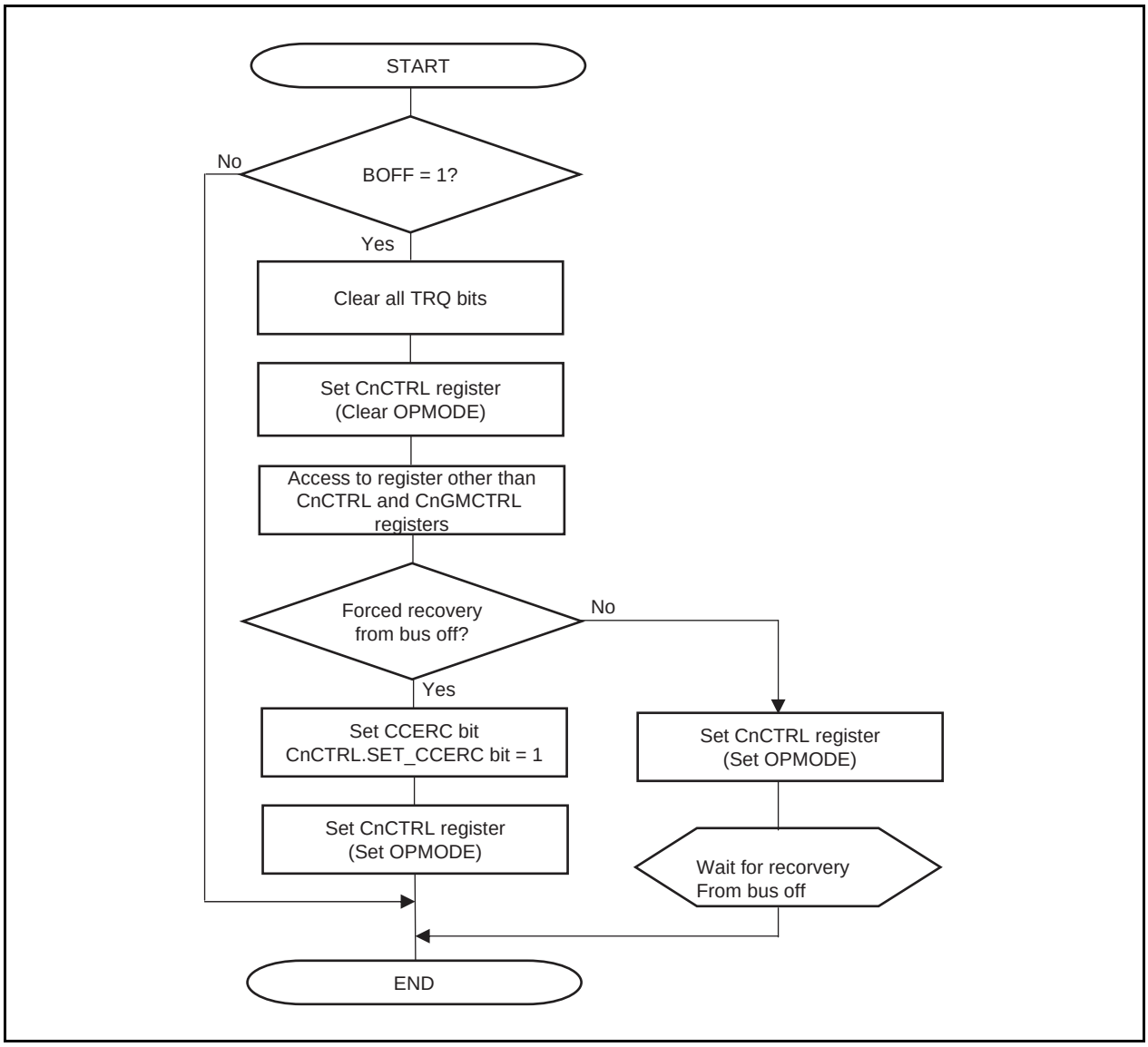


Figure 15-54. Normal Shutdown Process

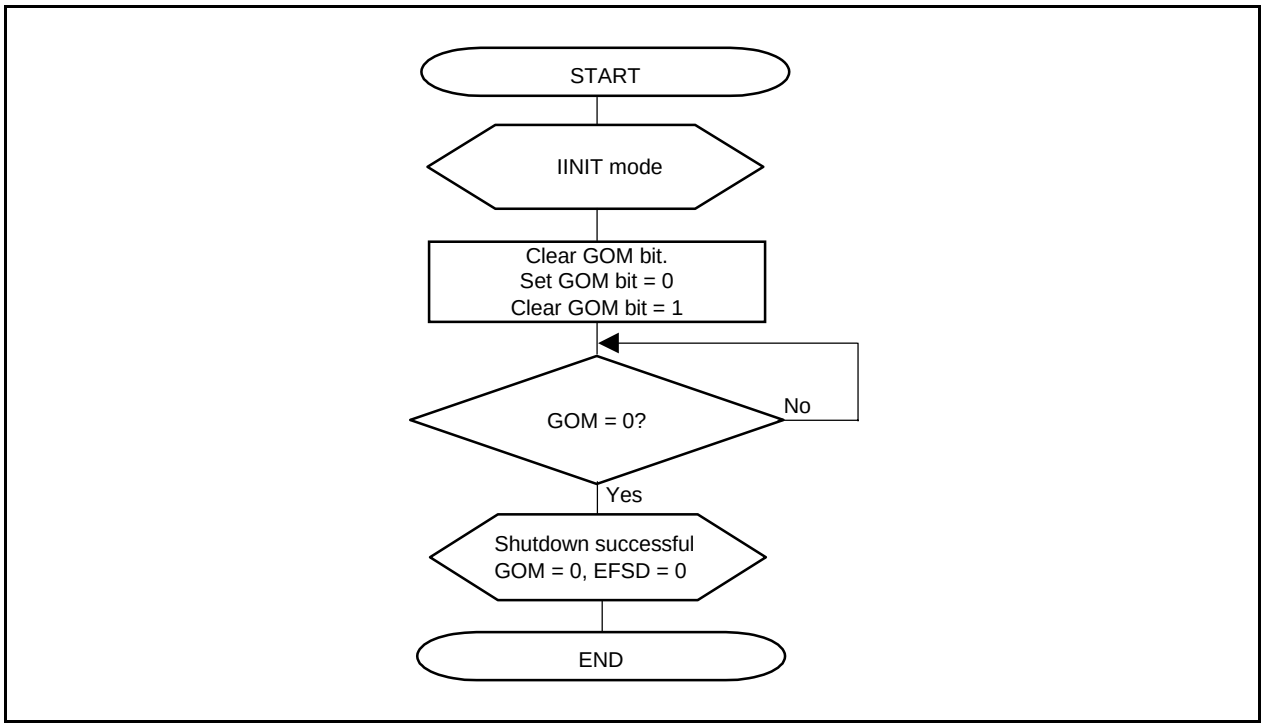
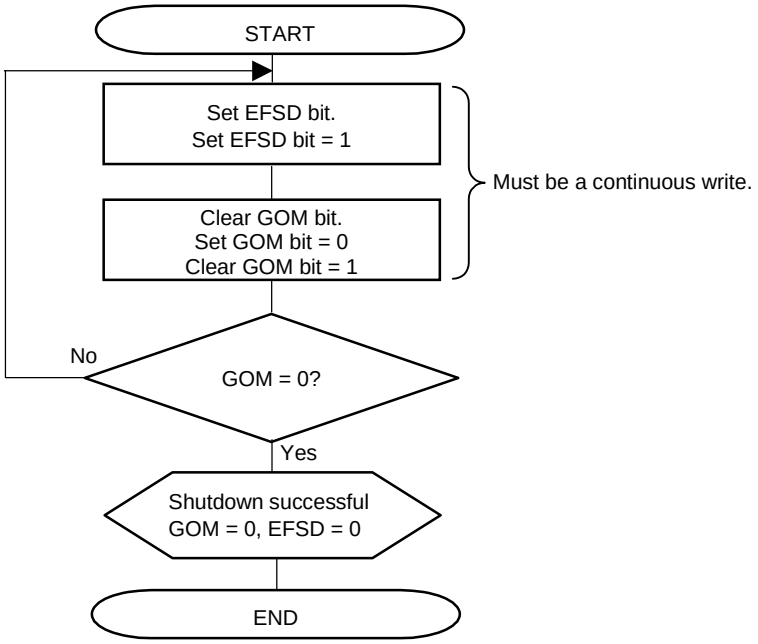


Figure 15-55. Forced Shutdown Process



Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 15-56. Error Handling

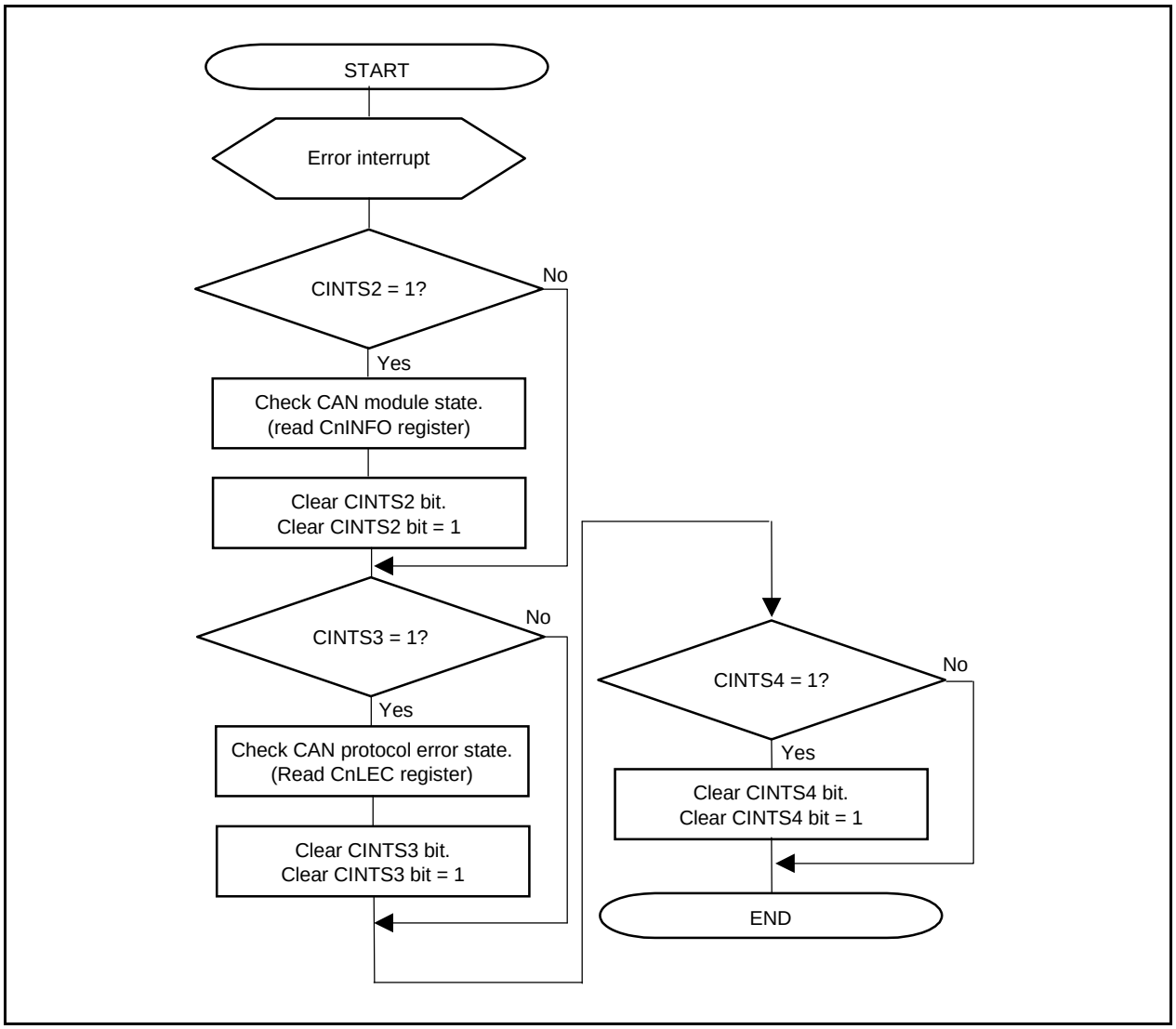
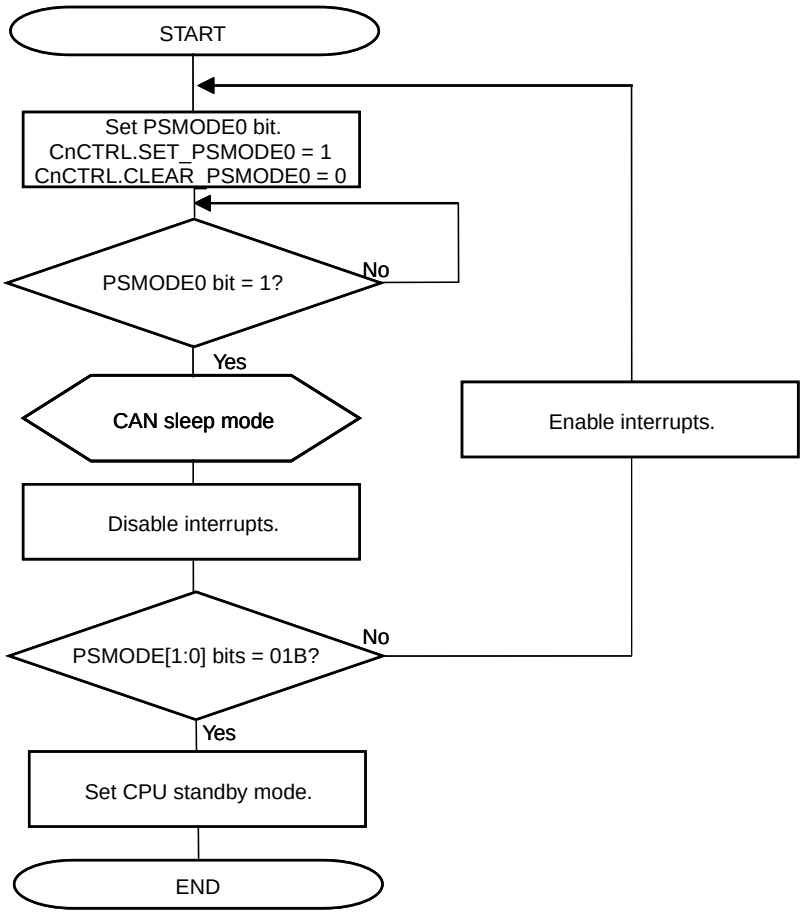
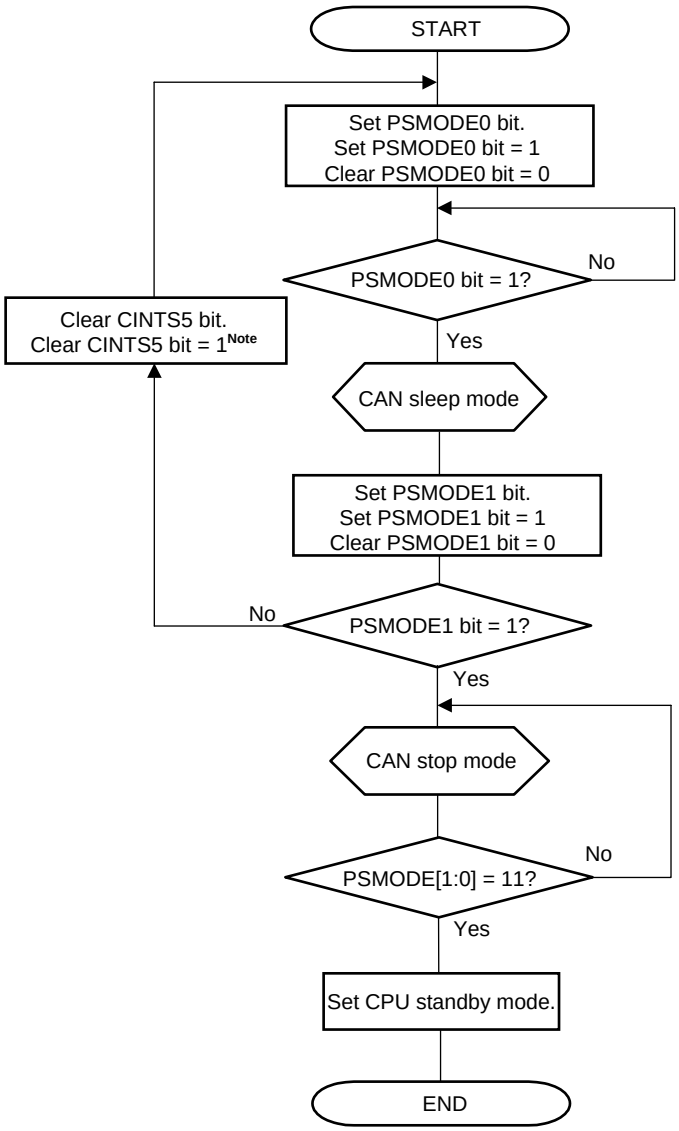


Figure 15-57. Setting CPU Standby (from CAN Sleep Mode)



Caution Check the CAN sleep mode or not, before CPU is set to standby mode. Otherwise, until CPU is set to standby mode, CAN sleep mode may be released by wake-up.

Figure 15-58. Setting CPU Standby (from CAN Stop Mode)



Note During wakeup interrupts

Caution The CAN stop mode can only be released by writing 01 to the CnCTRL.PSMODE1 and CnCTRL.PSMODE0 bits. H cannot be released by changing the CAN bus.

CHAPTER 16 DMA FUNCTION (DMA CONTROLLER)

The V850ES/FG2 and V850ES/FJ2 include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer. The V850ES/FE2 and V850ES/FF2 do not include a direct memory access (DMA) controller.

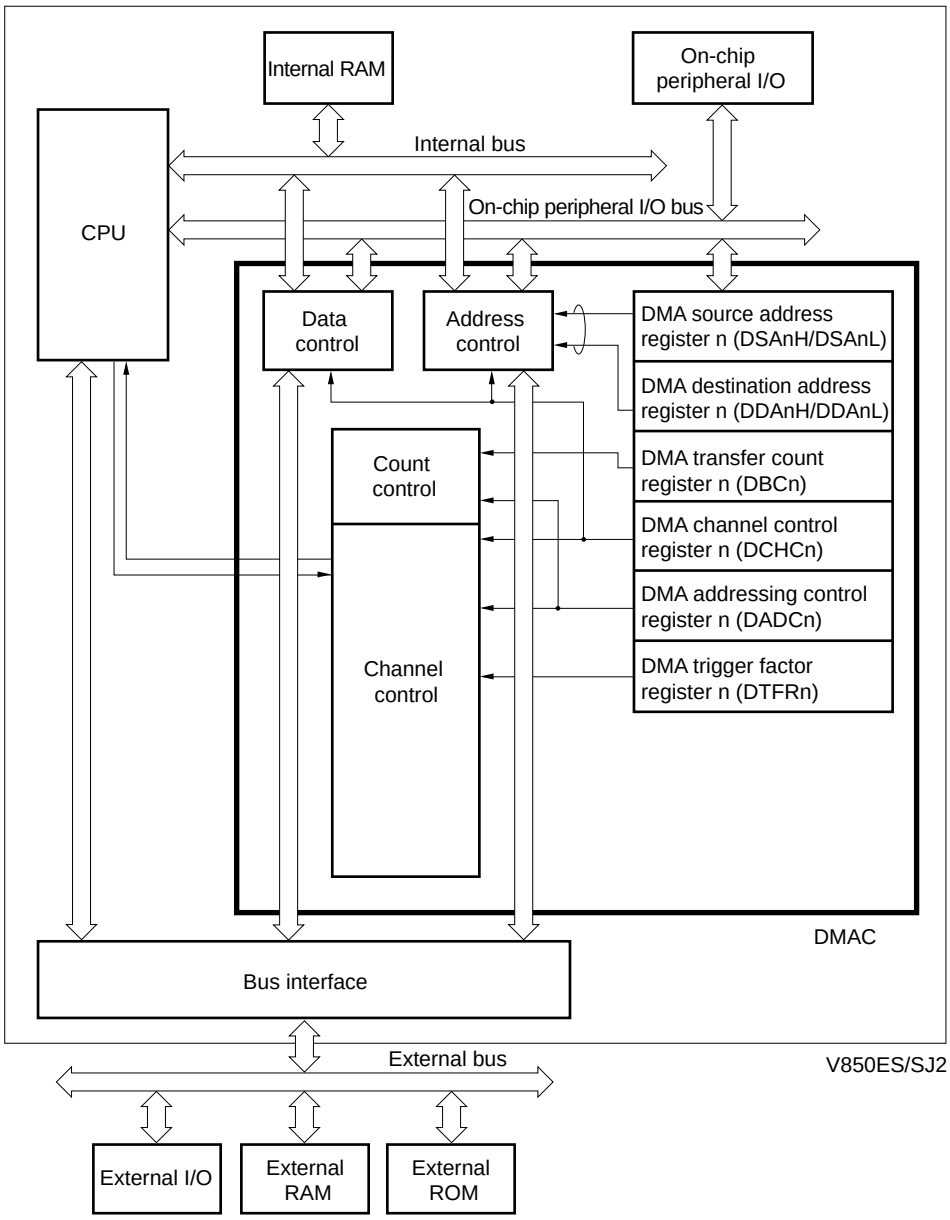
The DMAC controls data transfer between memory and I/O, between memories, or between I/Os based on DMA requests issued by the on-chip peripheral I/O (serial interface, timer/counter, and A/D converter), interrupts from external input pins, or software triggers (memory refers to internal RAM or external memory).

16.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Transfer type: Two-cycle transfer
- Transfer mode: Single transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter) or interrupts from external input pin
  - Requests by software trigger
- Transfer targets
  - Internal RAM ↔ Peripheral I/O
  - Peripheral I/O ↔ Peripheral I/O
  - Internal RAM ↔ External memory
  - External memory ↔ Peripheral I/O
  - External memory ↔ External memory

16.2 Configuration

Figure 16-1. Block Diagram of DMA Controller



Remark n = 0 to 3

16.3 Registers

(1) DMA source address registers 0 to 3 (DSA0 to DSA3)

The DSA0 to DSA3 registers set the DMA source addresses (26 bits each) for DMA channel n (n = 0 to 3).

These registers are divided into two 16-bit registers, DSAnH and DSAnL.

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

After reset: Undefined R/W Address: DSA0H FFFFF082H, DSA1H FFFFF08AH, DSA2H FFFFF092H, DSA3H FFFFF09AH, DSA0L FFFFF080H, DSA1L FFFFF088H, DSA2L FFFFF090H, DSA3L FFFFF098H

DSAnH (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16

DSAnL (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	Specification of DMA transfer source
0	External memory or on-chip peripheral I/O
1	Internal RAM

SA25 to SA16	Set the address (A25 to A16) of the DMA transfer source (default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.

SA15 to SA0	Set the address (A15 to A0) of the DMA transfer source (default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.

Cautions 1. Be sure to clear bits 14 to 10 of the DSAnH register to 0.

2. Set the DSAnH and DSAnL registers at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. When the value of the DSAn register is read, two 16-bit registers, DSAnH and DSAnL, are read. If reading and updating conflict, the value being updated may be read (see 16.13 Cautions).

★ **4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.**

(2) DMA destination address registers 0 to 3 (DDA0 to DDA3)

The DDA0 to DDA3 registers set the DMA destination address (26 bits each) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DDAnH and DDAnL.

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

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH, DDA2H FFFFF096H, DDA3H FFFFF09EH, DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

DDAnH (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

DDAnL (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	Specification of DMA transfer destination
0	External memory or on-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Set an address (A25 to A16) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

DA15 to DA0	Set an address (A15 to A0) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

Cautions 1. Be sure to clear bits 14 to 10 of the DDAnH register to 0.

2. Set the DDAnH and DDAnL registers at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. When the value of the DDAn register is read, two 16-bit registers, DDAnH and DDAnL, are read. If reading and updating conflict, a value being updated may be read (see 16.13 Cautions).

★ **4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.**

(3) DMA byte count registers 0 to 3 (DBC0 to DBC3)

The DBC0 to DBC3 registers are 16-bit registers that set the byte transfer count for DMA channel n (n = 0 to 3). These registers hold the remaining transfer count during DMA transfer.

These registers are decremented by 1 per one transfer regardless of the transfer data unit (8/16 bits), and the transfer is terminated if a borrow occurs.

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

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

DBCn (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15 to BC0	Byte transfer count setting or remaining byte transfer count during DMA transfer
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count
The number of transfer data set first is held when DMA transfer is complete.	

Cautions 1. Set the DBCn register at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

★ **2. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.**

(4) DMA addressing control registers 0 to 3 (DADC0 to DADC3)

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3).

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

Reset sets these registers to 0000H.

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 to 3)

15	14	13	12	11	10	9	8
0	DS0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SAD1	SAD0	DAD1	DAD0	0	0	0	0

DS0	Setting of transfer data size
0	8 bits
1	16 bits

SAD1	SAD0	Setting of count direction of the transfer source address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Setting of count direction of the destination address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

Cautions 1. Be sure to clear bits 15, 13 to 8, and 3 to 0 of the DADCn register to 0.

2. Set the DADCn register at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. The DS0 bit specifies the size of the transfer data, and does not control bus sizing. If 8-bit data (DS0 bit = 0) is set, therefore, the lower data bus is not always used.

4. If the transfer data size is set to 16 bits (DS0 bit = 1), transfer cannot be started from an odd address. Transfer is always started from an address with the first bit of the lower address aligned to 0.

5. If DMA transfer is executed on an on-chip peripheral I/O register (as the transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer on an 8-bit register, be sure to specify 8-bit transfer.

(5) DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

The DCHC0 to DCHC3 registers are 8-bit registers that control the DMA transfer operating mode for DMA channel n.

These registers can be read or written in 8-bit or 1-bit units. (However, bit 7 is read-only and bits 1 and 2 are write-only. If bit 1 or 2 is read, the read value is always 0.)

Reset sets these registers to 00H.

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H, DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

DCHCn (n = 0 to 3)

7	6	5	4	3	2	1	0
TCn[Note 1]	0	0	0	0	INITn[Note 2]	STGn[Note 2]	Enn

TCn	Status flag indicates whether DMA transfer through DMA channel n has completed or not
0	DMA transfer had not completed.
1	DMA transfer had completed.
It is set to 1 on the last DMA transfer completed and cleared to 0 when it is read.	

INITn	If the INITn bit is set to 1 with DMA transfer disabled (Enn bit = 0), the DMA transfer status can be initialized. When re-setting the DMA transfer status (re-setting the DDAnH, DDAnL, DSAnH, DSAnL, DBCn, and DADCn registers) before DMA transfer is completed (before the TCn bit is set to 1), be sure to initialize the DMA channel. When initializing the DMA controller, however, be sure to observe the procedure described in 16.11 Cautions.
Set the INIT bit to 1 when the Enn bit = 0.	

STGn	If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting of whether DMA transfer through DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
DMA transfer is enabled when the Enn bit is set to 1. When DMA transfer is completed (when a terminal count is generated), this bit is automatically cleared to 0. To abort DMA transfer, clear the Enn bit to 0 by software. To resume, set the Enn bit to 1 again. When aborting or resuming DMA transfer, however, be sure to observe the procedure described in 16.11 Cautions.	

Notes 1. The TCn bit is read-only.

2. The INITn and STGn bits are write-only.

Cautions 1. Be sure to clear bits 6 to 3 of the DCHCn register to 0.

2. Before generating a DMA transfer request by software, make sure that the TCn bit is set to 1 and then clear the TCn bit to 0.

3. If the INIT bit setting and the DMA transfer of another channel conflict, initialization may not perform.

4. When DMA transfer is completed (when a terminal count is generated), the Enn bit is cleared to 0 and then the TCn bit is set to 1. If the DCHCn register is read while its bits are being updated, a value indicating "transfer not completed and transfer is disabled" (TCn bit

= 0 and Enn bit = 0) may be read.

(6) DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt request signals from on-chip peripheral I/O.

The interrupt request signals set by these registers serve as DMA transfer start factors.

These registers can be read or written in 8-bit units. However, DFn bit can be read or written in 1-bit units.

Reset sets these registers to 00H.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H, DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 to 3)

DFn[Note]	DMA transfer request flag
0	No DMA transfer request
1	DMA transfer request

Note The DFn bit is a write-only bit. Write 0 to this bit to clear a DMA transfer request if an interrupt that is specified as the cause of starting DMA transfer occurs while DMA transfer is disabled.

Cautions 1. Set the IFCn5 to IFCn0 bits at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

2. An interrupt request that is generated in the standby mode (IDEL1, IDLE2, STOP, or sub-IDLE mode) does not start the DMA transfer cycle (nor is the DFn bit set to 1).

3. If a DMA start factor is selected by the IFCn5 to IFCn0 bits, the DFn bit is set to 1 when an interrupt occurs from the selected on-chip peripheral I/O, regardless of whether the DMA transfer is enabled or disabled. If DMA is enabled in this status, DMA transfer is immediately started.

4. The number of supported DMA start factors differs depending on the product as shown in Table 16-1. For details of the IFCn5 to IFCn0 bits, see Table 16-2 DMA Start Factors.

Table 16-1. Number of DMA Start Factors in Each Product

Part Number	Number of DMA Start Factors	Range of IFCn5 to IFCn0[Note]
μPD70F3237	59	000000B to 111011B (INTLVI to INTCB2T)
μPD70F3238	63	000000B to 111111B (INTLVI to INTC3TRX)
μPD70F3239	63	000000B to 111111B (INTLVI to INTC3TRX)

Note Setting a value other than the value shown in the range of IFCn5 to IFCn0 is prohibited.

Table 16-2. DMA Start Factors (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
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

Remark n = 0 to 3

Table 16-2. DMA Start Factors (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
1	0	1	0	0	0	INTC0TRX
1	0	1	0	0	1	INTKR
1	0	1	0	1	0	INTTQ1OV
1	0	1	0	1	1	INTTQ1CC0
1	0	1	1	0	0	INTTQ1CC1
1	0	1	1	0	1	INTTQ1CC2
1	0	1	1	1	0	INTTQ1CC3
1	0	1	1	1	1	INTUA2R
1	1	0	0	0	0	INTUA2T
1	1	0	0	0	1	INTC1ERR
	1	0	0	1	0	INTC1EWUP
1	1	0	0	1	1	INTC1REC
1	1	0	1	0	0	INTC1TRX
1	1	0	1	0	1	INTTQ2OV
1	1	0	1	1	0	INTTQ2CC0
1	1	0	1	1	1	INTTQ2CC1
1	1	1	0	0	0	INTTQ2CC2
1	1	1	0	0	1	INTTQ2CC3
1	1	1	0	1	0	INTCB2R
1	1	1	0	1	1	INTCB2T
1	1	1	1	0	0	INTC2REC[Note]
1	1	1	1	0	1	INTC2TRX[Note]
1	1	1	1	1	0	INTC3REC[Note]
1	1	1	1	1	1	INTC3TRX[Note]

Note μPD70F3238 and μPD70F3239 only

Remark n = 0 to 3

16.4 DMA Bus States

16.4.1 Types of bus states

The DMAC bus states consist of the following 10 states.

(1) TI state

The TI state is an idle state, during which no access request is issued.
The DMA request signals are sampled at the rising edge of the CLKOUT signal.

(2) T0 state

DMA transfer ready state (state in which a DMA transfer request has been issued and the bus mastership is acquired for the first DMA transfer).

(3) T1R state

The bus enters the T1R state at the beginning of a read operation in two-cycle transfer.
Address driving starts. After entering the T1R state, the bus enters the T2R state.

(4) T1RI state

The T1RI state is a state in which the bus waits for the acknowledge signal corresponding to an external memory read request.
After entering the last T1RI state, the bus invariably enters the T2R state.

(5) T2R state

The T2R state corresponds to the last state of a read operation in two-cycle transfer, or to a wait state.
In the last T2R state, read data is sampled. After entering the last T2R state, the bus enters the T1W state or T2RI state.

(6) T2RI state

State in which the bus is ready for DMA transfer to on-chip peripheral I/O or internal RAM (state in which the bus mastership is acquired for DMA transfer to on-chip peripheral I/O or internal RAM).
After entering the last T2RI state, the bus invariably enters the T1W state.

(7) T1W state

The bus enters the T1W state at the beginning of a write operation in two-cycle transfer.
Address driving starts. After entering the T1W state, the bus enters the T2W state.

(8) T1WI state

State in which the bus waits for the acknowledge signal corresponding to an external memory write request.
After entering the last T1WI state, the bus invariably enters the T2W state.

(9) T2W state

The T2W state corresponds to the last state of a write operation in two-cycle transfer, or to a wait state.
In the last T2W state, the write strobe signal is made inactive.

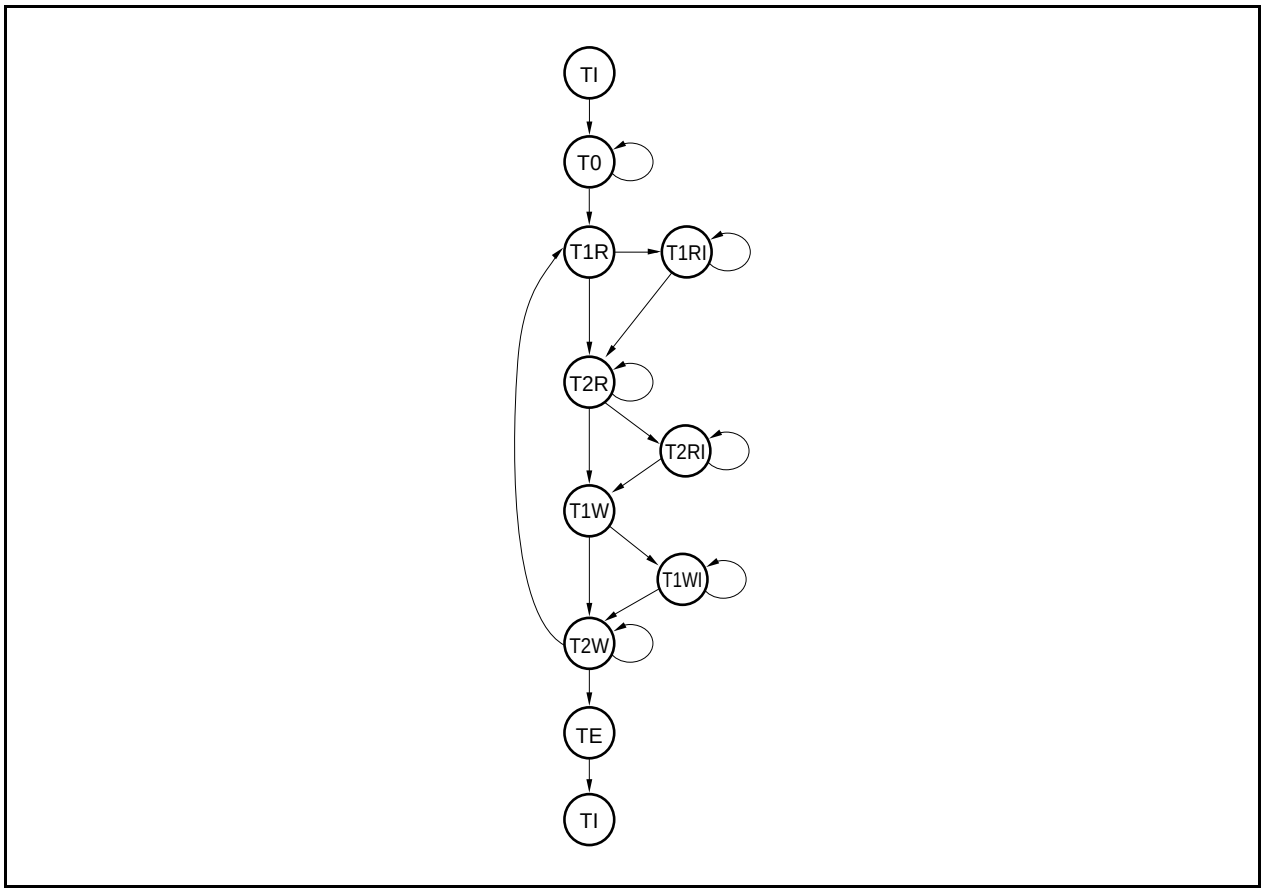
(10) TE state

The TE state corresponds to DMA transfer completion. The DMAC generates the internal DMA transfer completion signal and various internal signals are initialized. After entering the TE state, the bus invariably enters the TI state.

16.4.2 DMAC bus cycle state transition

Each time the processing for a DMA transfer is completed, the bus mastership is released.

Figure 16-2. DMAC Bus Cycle State Transition



16.5 Transfer Targets

Table 16-3 shows the relationship between the transfer targets (√: Transfer enabled, ×: Transfer disabled).

Table 16-3. Relationship Between Transfer Targets

		Transfer Destination			
		Internal ROM	On-Chip Peripheral I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O	×	√	√	√
	Internal RAM	×	√	×	√
	External memory	×	√	√	√
	Internal ROM	×	×	×	×

Caution The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 16-3.

16.6 Transfer Modes

Single transfer is supported as the transfer mode.

In single transfer mode, the bus is released at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

If a new transfer request of the same channel and a transfer request of another channel with a lower priority are generated in a transfer cycle, DMA transfer of the channel with the lower priority is executed after the bus is released to the CPU (the new transfer request of the same channel is ignored in the transfer cycle).

16.7 Transfer Types

As a transfer type, the 2-cycle transfer is supported.

In two-cycle transfer, data transfer is performed in two cycles, a read cycle and a write cycle.

In the read cycle, the transfer source address is output and reading is performed from the source to the DMAC. In the write cycle, the transfer destination address is output and writing is performed from the DMAC to the destination.

An idle cycle of one clock is always inserted between a read cycle and a write cycle. If the data bus width differs between the transfer source and destination for DMA transfer of two cycles, the operation is performed as follows.

<16-bit data transfer>

<1> Transfer from 32-bit bus → 16-bit bus
A read cycle (the higher 16 bits are in a high-impedance state) is generated, followed by generation of a write cycle (16 bits).

<2> Transfer from 16-/32-bit bus to 8-bit bus
A 16-bit read cycle is generated once, and then an 8-bit write cycle is generated twice.

<3> Transfer from 8-bit bus to 16-/32-bit bus
An 8-bit read cycle is generated twice, and then a 16-bit write cycle is generated once.

<4> Transfer between 16-bit bus and 32-bit bus
A 16-bit read cycle is generated once, and then a 16-bit write cycle is generated once.

For DMA transfer executed to an on-chip peripheral I/O register (transfer source/destination), be sure to specify the same transfer size as the register size. For example, for DMA transfer to an 8-bit register, be sure to specify byte (8-bit) transfer.

Remark The bus width of each transfer target (transfer source/destination) is as follows.

- On-chip peripheral I/O: 16-bit bus width
- Internal RAM: 32-bit bus width
- External memory: 8-bit or 16-bit bus width

16.8 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

The priorities are checked for every transfer cycle.

16.9 Time Related to DMA Transfer

The time required responding to a DMA request, and the minimum number of clocks required for DMA transfer are shown below.

Single transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1[Note 1] + Transfer destination memory access (<2>)

DMA Cycle		Minimum Number of Execution Clocks
<1> DMA request response time		4 clocks (MIN.) + Noise elimination time[Note 2]
<2> Memory access	External memory access	Depends on connected memory.
	Internal RAM access	2 clocks[Note 3]
	Peripheral I/O register access	3 clocks + Number of wait cycles specified by VSWC register[Note 4]

Notes 1. One clock is always inserted between a read cycle and a write cycle in DMA transfer.

2. If an external interrupt (INTPn) is specified as the trigger to start DMA transfer, noise elimination time is added (n = 0 to 7).

3. Two clocks are required for a DMA cycle.

4. More wait cycles are necessary for accessing a specific peripheral I/O register (for details, see **3.4.10 (2)**).

16.10 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

(1) Request by software

If the STGn bit is set to 1 while the DCHCn.TCn bit = 1 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

To request the next DMA transfer cycle immediately after that, confirm, by using the DBCn register, that the preceding DMA transfer cycle has been completed, and set the STGn bit to 1 again (n = 0 to 3).

TCn bit = 0, Enn bit = 1
↓
STGn bit = 1 … Starts the first DMA transfer.
↓
Confirm that the contents of the DBCn register have been updated.
STGn bit = 1 … Starts the second DMA transfer.
↓
:
↓
Generation of terminal count … Enn bit = 0, TCn bit = 1, and INTDMAn signal is generated.

(2) Request by on-chip peripheral I/O

If an interrupt request is generated from the on-chip peripheral I/O set by the DTFRn register when the DCHCn.TCn bit = 0 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

Cautions 1. Two start factors (software trigger and hardware trigger) cannot be used for one DMA channel. If two start factors are simultaneously generated for one DMA channel, only one of them is valid. The start factor that is valid cannot be identified.

2. A new transfer request that is generated after the preceding DMA transfer request was generated or in the preceding DMA transfer cycle is ignored (cleared).

3. The transfer request interval of the same DMA channel varies depending on the setting of bus wait in the DMA transfer cycle, the start status of the other channels, or the external bus hold request. In particular, as described in Caution 2, a new transfer request that is generated for the same channel before the DMA transfer cycle or during the DMA transfer cycle is ignored. Therefore, the transfer request intervals for the same DMA channel must be sufficiently separated by the system. When the software trigger is used, completion of the DMA transfer cycle that was generated before can be checked by updating the DBCn register.

16.11 DMA Abort Factors

DMA transfer is aborted if a bus hold occurs.

The same applies if transfer is executed between the internal memory/on-chip peripheral I/O and internal memory/on-chip peripheral I/O.

When the bus hold is cleared, DMA transfer is resumed.

16.12 End of DMA Transfer

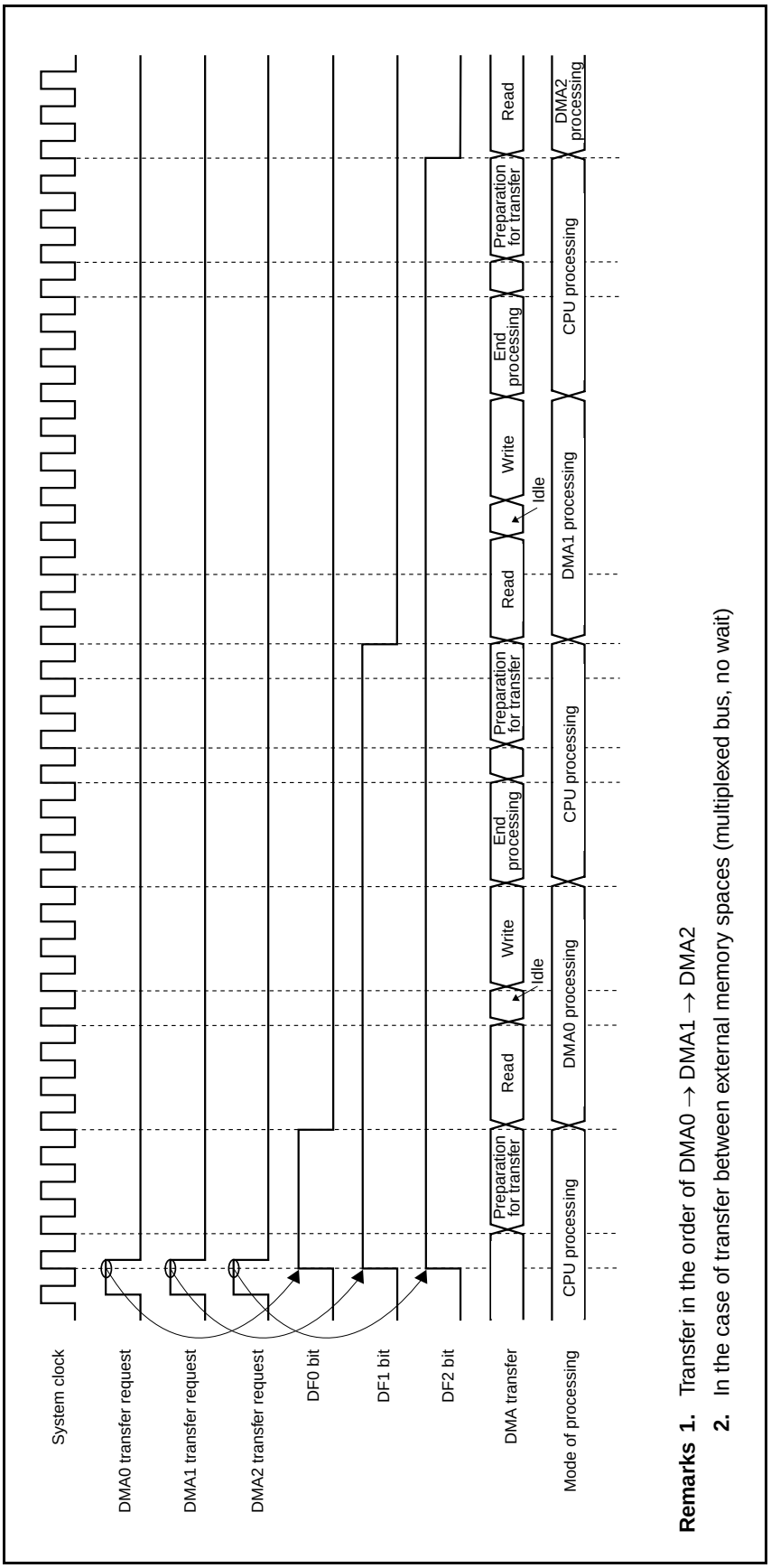
When DMA transfer has been completed the number of times set to the DBCn register and when the DCHCn.Enn bit is cleared to 0 and TCn bit is set to 1, a DMA transfer end interrupt request signal (INTDMAn) is generated for the interrupt controller (INTC) (n = 0 to 3).

The V850ES/FG2 and V850ES/FJ2 do not output a terminal count signal to an external device. Therefore, confirm completion of DMA transfer by using the DMA transfer end interrupt or polling the TCn bit.

16.13 Operation Timing

Figures 16-3 to 16-6 show DMA operation timing.

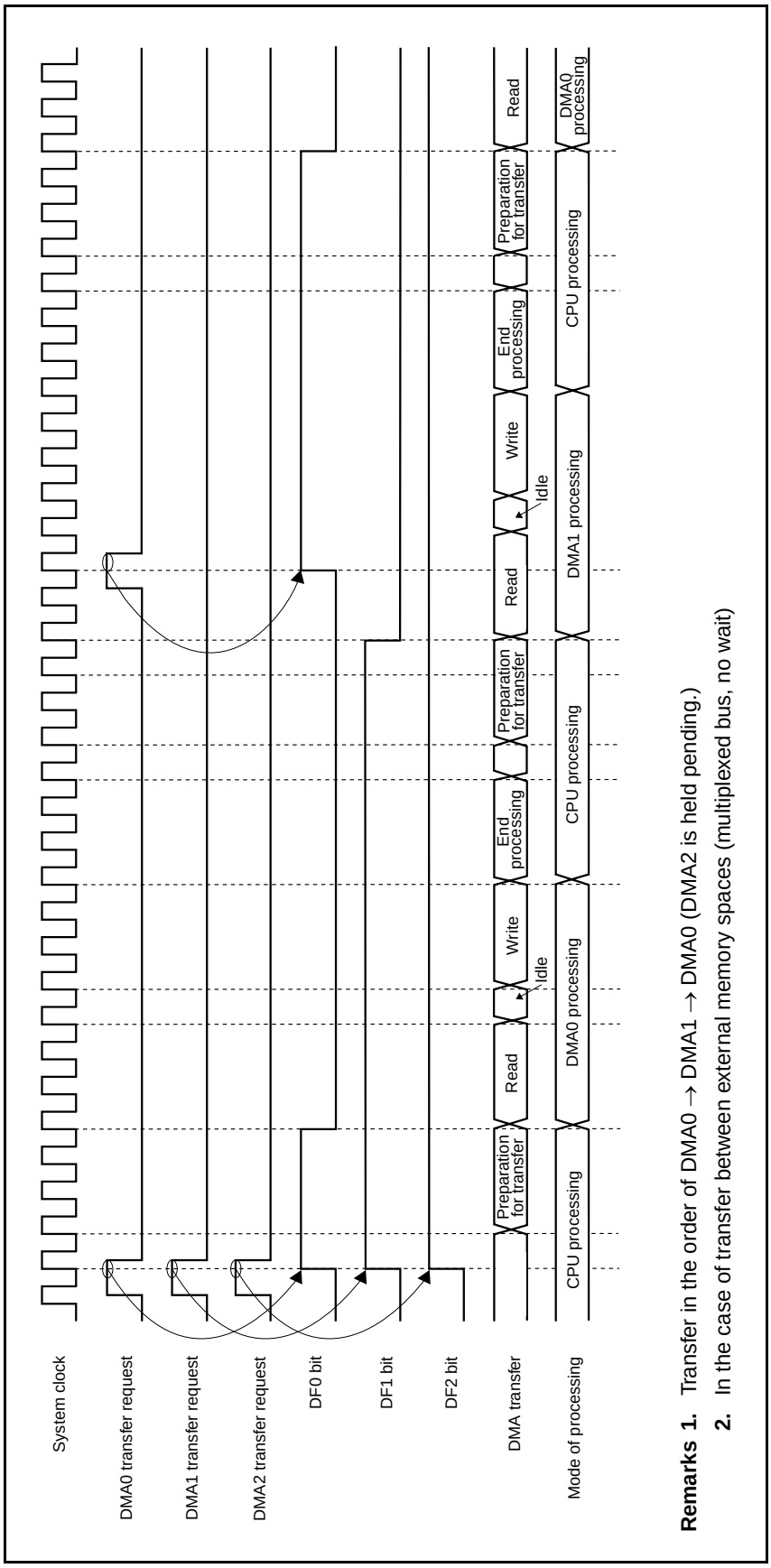
Figure 16-3. Priority of DMA (1)



Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA2

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

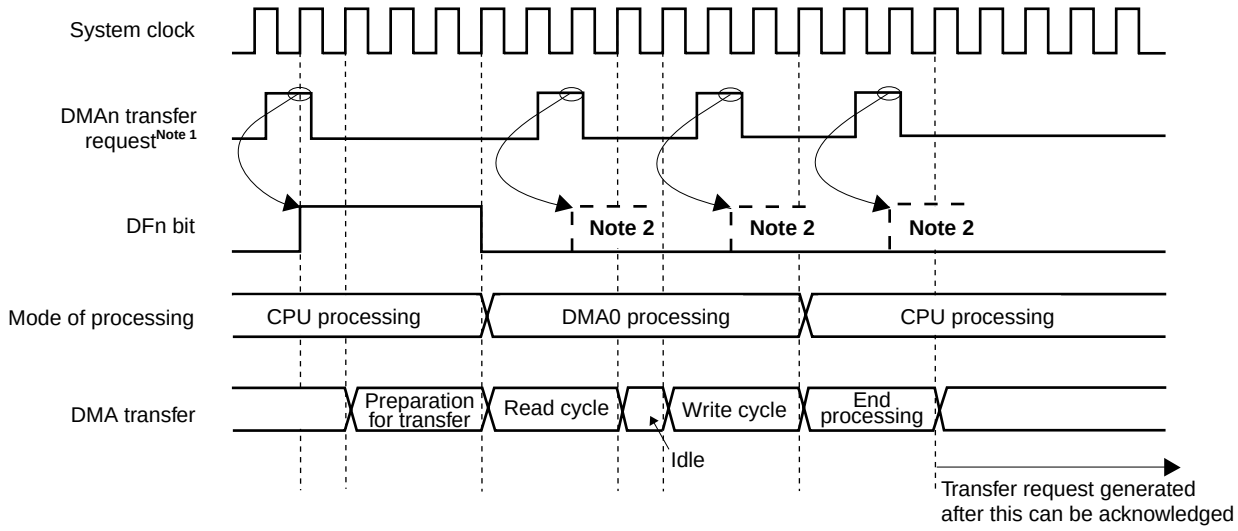
Figure 16-4. Priority of DMA (2)



Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA0 (DMA2 is held pending.)

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 16-5. Period in Which DMA Transfer Request Is Ignored (1)

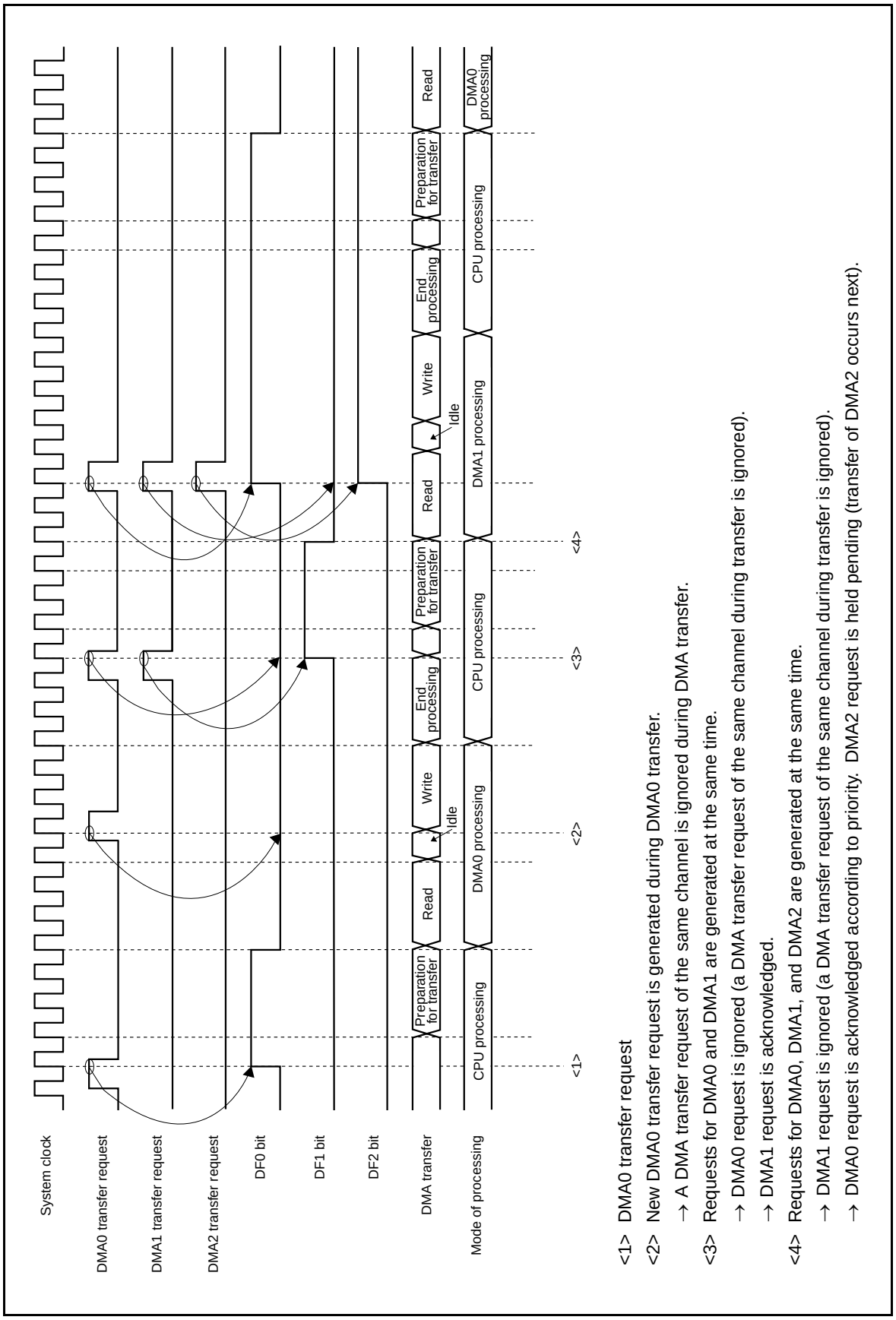


Notes 1. Interrupt from on-chip peripheral I/O, or software trigger (STGn bit)

2. New DMA request of the same channel is ignored between when the first request is generated and the end processing is complete.

Remark In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 16-6. Period in Which DMA Transfer Request Is Ignored (2)



<1> DMA0 transfer request

<2> New DMA0 transfer request is generated during DMA0 transfer.
→ A DMA transfer request of the same channel is ignored during DMA transfer.

<3> Requests for DMA0 and DMA1 are generated at the same time.
→ DMA0 request is ignored (a DMA transfer request of the same channel during transfer is ignored).
→ DMA1 request is acknowledged.

<4> Requests for DMA0, DMA1, and DMA2 are generated at the same time.
→ DMA1 request is ignored (a DMA transfer request of the same channel during transfer is ignored).
→ DMA0 request is acknowledged according to priority. DMA2 request is held pending (transfer of DMA2 occurs next).

16.14 Cautions

(1) Caution for VSWC register

When using the DMAC, be sure to set an appropriate value, in accordance with the operating frequency, to the VSWC register.

When the default value (77H) of the VSWC register is used, or if an inappropriate value is set to the VSWC register, the operation is not correctly performed (for details of the VSWC register, see **3.4.10 (1) (a) System wait control register (VSWC)**).

(2) Caution for DMA transfer executed on internal RAM

When executing the following instructions located in the internal RAM, do not execute a DMA transfer that transfers data to/from the internal RAM (transfer source/destination), because the CPU may not operate correctly afterward.

- Bit manipulation instruction located in internal RAM (SET1, CLR1, or NOT1)
- Data access instruction to misaligned address located in internal RAM

Conversely, when executing a DMA transfer to transfer data to/from the internal RAM (transfer source/destination), do not execute the above two instructions.

(3) Caution for reading DCHCn.TCn bit (n = 0 to 3)

The TCn bit is cleared to 0 when it is read, but it is not automatically cleared even if it is read at a specific timing. To accurately clear the TCn bit, add the following processing.

(a) When waiting for completion of DMA transfer by polling TCn bit

Confirm that the TCn bit has been set to 1 (after TCn bit = 1 is read), and then read the TCn bit three more times.

(b) When reading TCn bit in interrupt servicing routine

Execute reading the TCn bit three times.

(4) DMA transfer initialization procedure (setting DCHCn.INITn bit to 1)

Even if the INITn bit is set to 1 when the channel executing DMA transfer is to be initialized, the channel may not be initialized. To accurately initialize the channel, execute either of the following two procedures.

(a) Temporarily stop transfer of all DMA channels

Initialize the channel executing DMA transfer using the procedure in <1> to <7> below.

Note, however, that TCn bit is cleared to 0 when step <5> is executed. Make sure that the other processing programs do not expect that the TCn bit is 1.

<1> Disable interrupts (DI).

<2> Read the DCHCn.Enn bit of DMA channels other than the one to be forcibly terminated, and transfer the value to a general-purpose register.

<3> Clear the Enn bit of the DMA channels used (including the channel to be forcibly terminated) to 0. To clear the Enn bit of the last DMA channel, execute the clear instruction twice. If the target of DMA transfer (transfer source/destination) is the internal RAM, execute the instruction three times.

Example: Execute instructions in the following order if channels 0, 1, and 2 are used (if the target of transfer is not the internal RAM).

- Clear DCHC0.E00 bit to 0.
- Clear DCHC1.E11 bit to 0.
- Clear DCHC2.E22 bit to 0.
- Clear DCHC2.E22 bit to 0 again.

<4> Set the INITn bit of the channel to be forcibly terminated to 1.

<5> Read the TCn bit of each channel not to be forcibly terminated. If both the TCn bit and the Enn bit read in <2> are 1 (logical product (AND) is 1), clear the saved Enn bit to 0.

<6> After the operation in <5>, write the Enn bit value to the DCHCn register.

<7> Enable interrupts (EI).

Caution Be sure to execute step <5> above to prevent illegal setting of the Enn bit of the channels whose DMA transfer has been normally completed between <2> and <3>.

(b) Repeatedly execute setting INITn bit until transfer is forcibly terminated correctly

<1> Suppress a request from the DMA request source of the channel to be forcibly terminated (stop operation of the on-chip peripheral I/O).
<2> Check that the DMA transfer request of the channel to be forcibly terminated is not held pending, by using the DTFRn.DFn bit. If a DMA transfer request is held pending, wait until execution of the pending request is completed.
<3> When it has been confirmed that the DMA request of the channel to be forcibly terminated is not held pending, clear the Enn bit to 0.
<4> Again, clear the Enn bit of the channel to be forcibly terminated.
If the target of transfer for the channel to be forcibly terminated (transfer source/destination) is the internal RAM, execute this operation once more.
<5> Copy the initial number of transfers of the channel to be forcibly terminated to a general-purpose register.
<6> Set the INITn bit of the channel to be forcibly terminated to 1.
<7> Read the value of the DBCn register of the channel to be forcibly terminated, and compare it with the value copied in <5>. If the two values do not match, repeat operations <6> and <7>.

Remarks 1. When the value of the DBCn register is read in <7>, the initial number of transfers is read if forced termination has been correctly completed. If not, the remaining number of transfers is read.
2. Note that method (b) may take a long time if the application frequently uses DMA transfer for a channel other than the DMA channel to be forcibly terminated.

(5) Procedure of temporarily stopping DMA transfer (clearing Enn bit)

Stop and resume the DMA transfer under execution using the following procedure.

<1> Suppress a transfer request from the DMA request source (stop the operation of the on-chip peripheral I/O).
<2> Check the DMA transfer request is not held pending, by using the DFn bit (check if the DFn bit = 0).
If a request is pending, wait until execution of the pending DMA transfer request is completed.
<3> If it has been confirmed that no DMA transfer request is held pending, clear the Enn bit to 0 (this operation stops DMA transfer).
<4> Set the Enn bit to 1 to resume DMA transfer.
<5> Resume the operation of the DMA request source that has been stopped (start the operation of the on-chip peripheral I/O).

(6) Memory boundary

The operation is not guaranteed if the address of the transfer source or destination exceeds the area of the DMA target (external memory, internal RAM, or on-chip peripheral I/O) during DMA transfer.

(7) Transferring misaligned data

DMA transfer of misaligned data with a 16-bit bus width is not supported.
If an odd address is specified as the transfer source or destination, the least significant bit of the address is forcibly assumed to be 0.

(8) Bus arbitration for CPU

Because the DMA controller has a higher priority bus mastership than the CPU, a CPU access that takes place during DMA transfer is held pending until the DMA transfer cycle is completed and the bus is released to the CPU.

However, the CPU can access the external memory, on-chip peripheral I/O, and internal RAM to/from which DMA transfer is not being executed.

- The CPU can access the internal RAM when DMA transfer is being executed between the external memory and on-chip peripheral I/O.
- The CPU can access the internal RAM and on-chip peripheral I/O when DMA transfer is being executed between the external memory and external memory.

(9) Registers/bits that must not be rewritten during DMA operation

Set the following registers at the following timing when a DMA operation is not under execution.

[Registers]

- DSAnH, DSAnL, DDAnH, DDAnL, DBCn, and DADCn registers
- DTFRn.IFCn5 to DTFRn.IFCn0 bits

[Timing of setting]

- Period from after reset to start of the first DMA transfer
- Time after channel initialization to start of DMA transfer
- Period from after completion of DMA transfer (TCn bit = 1) to start of the next DMA transfer

(10) Be sure to set the following register bits to 0.

- Bits 14 to 10 of DSAnH register
- Bits 14 to 10 of DDAnH register
- Bits 15, 13 to 8, and 3 to 0 of DADCn register
- Bits 6 to 3 of DCHCn register

(11) DMA start factor

Do not start two or more DMA channels with the same start factor. If two or more channels are started with the same factor, a DMA channel with a lower priority may be acknowledged earlier than a DMA channel with a higher priority.

(12) Read values of DSAn and DDAn registers

Values in the middle of updating may be read from the DSAn and DDAn registers during DMA transfer (n = 0 to 3).

For example, if the DSAnH register and then the DSAnL register are read when the DMA transfer source address (DSAn register) is 0000FFFFH and the count direction is incremental (DADCn.SAD1 and DADCn.SAD0 bits = 00), the value of the DSAn register differs as follows, depending on whether DMA transfer is executed immediately after the DSAnH register is read.

(a) If DMA transfer does not occur while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Read value of DSAnL register: DSAnL = FFFFH

(b) If DMA transfer occurs while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Occurrence of DMA transfer

<3> Incrementing DSAn register: DSAn = 00100000H

<4> Read value of DSAnL register: DSAnL = 0000H

(13) DMA trigger Change

When the interrupt request signal selection in the DTFR register is changed and the original interrupt request selection is also the selection for another DMA channel, there is the possibility that a DMA transfer will inadvertently occur.

When the setting of DTFRn register is changed, please make sure to follow the procedure shown below:

(a). In the case that the desired setting of the IFC is NOT set to a different DMA channel

<1> DMA operation of target channel, which is re-written, should be stopped (DCHCn.Enn=0)

<2> Change the setting of DTFRn register (By 8 bit operation)

<3> Confirm DTFRn.DFn=0 (The operation of the interrupt generation factor should be stopped previously)

<4> DMA operation can be enabled (DCHCn.Enn=1)

(b). In the case that the desired setting of the IFC (DMAn) IS set to another DMA channel (DMAm)

<1> DMA operation of target channel, (DMAn) which is re-written, should be stopped (DCHCn.Enn=0).

<2> DMA operation of the channel, which has the same value for the IFCm5-0bits, should be stopped (DCHCm.Emm=0).

<3> Change the setting of DTFRn register (By 8 bit operation)

<4> Confirm DTFRn.DFn=0 and DTFRm.DFm=0 (The operation of the interrupt generation factor should be stopped previously)

<5> DMA operation can be enabled (DCHCn.Enn=1 and DCHCm.Enn=1)

CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

The V850ES/Fx2 is provided with a dedicated interrupt controller (INTC) for interrupt servicing.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850ES/Fx2 can process interrupt request signals from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

The number of supported maskable interrupt sources differs depending on the product, as shown in Table 17-1.

Table 17-1. Number of Maskable Interrupt Sources

Part Number		Number of Maskable Interrupt Sources	Maskable Interrupts[Note]
V850ES/FE2		43	INTLVI to INTWT
V850ES/FF2			
V850ES/FG2		61	INTLVI to INTMA3
V850ES/FJ2	μPD70F3237	72	INTLVI to INTCB2T
	μPD70F3238, 70F3239	82	INTLVI to INTC3TRX

Note Refer to **Table 17-2** for the maskable interrupts.

Caution The explanations in this chapter use the maximum of 82 maskable interrupt sources.

17.1 Features

○ Interrupts
- Non-maskable interrupts: 2 sources
- Maskable interrupts: External: 15, Internal: 67 sources (Number of sources varies depending on the product.)
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals

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

The interrupt/exception sources are listed in Table 17-2.

Table 17-2. Interrupt Source List (1/3)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input Reset input by internal source	RESET	0000H	00000000H	Undefined	–
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	nextPC Note 1	–
Software exception	Exception	–	TRAP0n Note 2	TRAP instruction	–	004nH Note 2	00000040H	nextPC	–
		–	TRAP1n Note 2	TRAP instruction	–	005nH Note 2	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTLVI	Low voltage detection	POCLVI	0080H	00000080H	nextPC	LVIIC
		1	INTP0	External interrupt pin input edge detection (INTP0)	Pin	0090H	00000090H	nextPC	PIC0
		2	INTP1	External interrupt pin input edge detection (INTP1)	Pin	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00B0H	000000B0H	nextPC	PIC2
		4	INTP3	External interrupt pin input edge detection (INTP3)	Pin	00C0H	000000C0H	nextPC	PIC3
		5	INTP4	External interrupt pin input edge detection (INTP4)	Pin	00D0H	000000D0H	nextPC	PIC4
		6	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00E0H	000000E0H	nextPC	PIC5
		7	INTP6	External interrupt pin input edge detection (INTP6)	Pin	00F0H	000000F0H	nextPC	PIC6
		8	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0100H	00000100H	nextPC	PIC7
		9	INTTQ0OV	TMQ0 overflow	TMQ0	0110H	00000110H	nextPC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 capture 0/compare 0 match	TMQ0	0120H	00000120H	nextPC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 capture 1/compare 1 match	TMQ0	0130H	00000130H	nextPC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 capture 2/compare 2 match	TMQ0	0140H	00000140H	nextPC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 capture 3/compare 3 match	TMQ0	0150H	00000150H	nextPC	TQ0CCIC3
		14	INTTP0OV	TMP0 overflow	TMP0	0160H	00000160H	nextPC	TP0OVIC
		15	INTTP0CC0	TMP0 capture 0/compare 0 match	TMP0	0170H	00000170H	nextPC	TP0CCIC0
		16	INTTP0CC1	TMP0 capture 1/compare 1 match	TMP0	0180H	00000180H	nextPC	TP0CCIC1
		17	INTTP1OV	TMP1 overflow	TMP1	0190H	00000190H	nextPC	TP1OVIC
		18	INTTP1CC0	TMP1 capture 0/compare 0 match	TMP1	01A0H	000001AH	nextPC	TP1CCIC0
		19	INTTP1CC1	TMP1 capture 1/compare 1 match	TMP1	01B0H	000001B0H	nextPC	TP1CCIC1
		20	INTTP2OV	TMP2 overflow	TMP2	01C0H	000001C0H	nextPC	TP2OVIC
		21	INTTP2CC0	TMP2 capture 0/compare 0 match	TMP2	01D0H	000001D0H	nextPC	TP2CCIC0

Notes 1. For the restoring in the case of INTWDT2, see "17.2.3 (2) From INTWDT2 signal".

2. n = 0H to FH

Table 17-2. Interrupt Source List (2/3)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	22	INTTP2CC1	TMP2 capture 1/compare 1 match	TMP2	01E0H	000001E0H	nextPC	TP2CCIC1
		23	INTTP3OV	TMP3 overflow	TMP3	01F0H	000001F0H	nextPC	TP3OVIC
		24	INTTP3CC0	TMP3 capture 0/compare 0 match	TMP3	0200H	00000200H	nextPC	TP3CCIC0
		25	INTTP3CC1	TMP3 capture 1/compare 1 match	TMP3	0210H	00000210H	nextPC	TP3CCIC1
		26	INTTM0EQ0	TMM0 compare match	TMM0	0220H	00000220H	nextPC	TM0EQIC0
		27	INTCB0R	CSIB0 reception completion	CSIB0/IIC1	0230H	00000230H	nextPC	CB0RIC
		28	INTCB0T	CSIB0 consecutive transmission write enable	CSIB0	0240H	00000240H	nextPC	CB0TIC
		29	INTCB1R	CSIB1 reception completion	CSIB1	0250H	00000250H	nextPC	CB1RIC
		30	INTCB1T	CSIB1 consecutive transmission write enable	CSIB1	0260H	00000260H	nextPC	CB1TIC
		31	INTUA0R	UARTA0 reception completion	UARTA0/ CSIB4	0270H	00000280H	nextPC	UA0RIC
		32	INTUA0T	UARTA0 transmission enable	UARTA0/ CSIB4	0280H	00000280H	nextPC	UA0TIC
		33	INTUA1R	UARTA1 reception completion/UARTA1 reception error	UARTA1/ IIC2	0290H	00000290H	nextPC	UA1RIC
		34	INTUA1T	UARTA1 transmission enable	UARTA1	02A0H	000002A0H	nextPC	UA1TIC
		35	INTAD	A/D conversion completion	A/D	02BH	000002B0H	nextPC	ADIC
		36	INTC0ERR	AFCAN0 error	AFCAN0	02C0H	000002C0H	nextPC	C0ERRIC
		37	INTC0WUP	AFCAN0 wakeup	AFCAN0	02D0H	000002D0H	nextPC	C0WUPIC
		38	INTC0REC	AFCAN0 reception completion	AFCAN0	02E0H	000002E0H	nextPC	C0RECIC
		39	INTC0TRX	AFCAN0 transmission completion	AFCAN0	02F0H	000002F0H	nextPC	C0TRXIC
		40	INTKR	Key return interrupt request	KR	0300H	00000300H	nextPC	KRIC
		41	INTWTI	Watch timer interval	WT	0310H	00000310H	nextPC	WTIIC
		42	INTWT	Watch timer reference time	WT	0320H	00000320H	nextPC	WTIC
		43	INTP8	External interrupt pin input edge detection (INTP8)	Pin	0330H	00000330H	nextPC	PIC8
		44	INTP9	External interrupt pin input edge detection (INTP9)	Pin	0340H	00000340H	nextPC	PIC9
		45	INTP10	External interrupt pin input edge detection (INTP10)	Pin	0350H	00000350H	nextPC	PIC10
		46	INTTQ1OV	TMQ1 overflow	TMQ1	0360H	00000360H	nextPC	TQ1OVIC
		47	INTTQ1CC0	TMQ1 capture 0/compare 0 match	TMQ1	0370H	00000370H	nextPC	TQ1CCIC0
		48	INTTQ1CC1	TMQ1 capture 1/compare 1 match	TMQ1	0380H	00000380H	nextPC	TQ1CCIC1
		49	INTTQ1CC2	TMQ1 capture 2/compare 2 match	TMQ1	0390H	00000390H	nextPC	TQ1CCIC2
		50	INTTQ1CC3	TMQ1 capture 3/compare 3 match	TMQ1	03A0H	000003A0H	nextPC	TQ1CCIC3

Table 17-2. Interrupt Source List (3/3)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	51	INTUA2R	UARTA2 reception completion/error	UARTA2	03B0H	000003B0H	nextPC	UA2RIC
		52	INTUA2T	UARTA2 transmission enable	UARTA2	03C0H	000003C0H	nextPC	UA2TIC
		53	INTC1ERR	AFCAN1 error	AFCAN1	03D0H	000003D0H	nextPC	C1ERRIC
		54	INTC1WUP	AFCAN1 wakeup	AFCAN1	03E0H	000003E0H	nextPC	C1WUPIC
		55	INTC1REC	AFCAN1 reception completion	AFCAN1	03F0H	000003F0H	nextPC	C1RECIC
		56	INTC1TRX	AFCAN1 transmission completion	AFCAN1	0400H	00000400H	nextPC	C1TRXIC
		57	INTDMA0	DMA0 transfer end	DMA	0410H	00000410H	nextPC	DMAIC0
		58	INTDMA1	DMA1 transfer end	DMA	0420H	00000420H	nextPC	DMAIC1
		59	INTDMA2	DMA2 transfer end	DMA	0430H	00000430H	nextPC	DMAIC2
		60	INTDMA3	DMA3 transfer end	DMA	0440H	00000440H	nextPC	DMAIC3
		61	INTP11	External interrupt pin input edge detection (INTP11)	Pin	0450H	00000450H	nextPC	PIC11
		62	INTP12	External interrupt pin input edge detection (INTP12)	Pin	0460H	00000460H	nextPC	PIC12
		63	INTP13	External interrupt pin input edge detection (INTP13)	Pin	0470H	00000470H	nextPC	PIC13
		64	INTP14	External interrupt pin input edge detection (INTP14)	Pin	0480H	00000480H	nextPC	PIC14
		65	INTTQ2OV	TMQ2 overflow	TMQ2	0490H	00000490H	nextPC	TQ2OVIC
		66	INTTQ2CC0	TMQ2 capture 0/compare 0 match	TMQ2	04A0H	000004A0H	nextPC	TQ2CCIC0
		67	INTTQ2CC1	TMQ2 capture 1/compare 1 match	TMQ2	04B0H	000004B0H	nextPC	TQ2CCIC1
		68	INTTQ2CC2	TMQ2 capture 2/compare 2 match	TMQ2	04C0H	000004C0H	nextPC	TQ2CCIC2
		69	INTTQ2CC3	TMQ2 capture 3/compare 3 match	TMQ2	04D0H	000004D0H	nextPC	TQ2CCIC3
		70	INTCB2R	CSIB2 reception completion/error	CSIB2	04E0H	000004E0H	nextPC	CB2RIC
		71	INTCB2T	CSIB2 continuous transmission write enable	CSIB2	04F0H	000004F0H	nextPC	CB2TIC
		72	INTUA3R	UARTA3 reception completion/error	UARTA3	0500H	00000500H	nextPC	UA3RIC
		73	INTUA3T	UARTA3 transmission enable	UARTA3	0510H	00000510H	nextPC	UA3TIC
		74	INTC2ERR	AFCAN2 error	AFCAN2	0520H	00000520H	nextPC	C2ERRIC
		75	INTC2WUP	AFCAN2 wakeup	AFCAN2	0530H	00000530H	nextPC	C2WUPIC
		76	INTC2REC	AFCAN2 reception completion	AFCAN2	0540H	00000540H	nextPC	C2RECIC
		77	INTC2TRX	AFCAN2 transmission completion	AFCAN2	0550H	00000550H	nextPC	C2TRXIC
		78	INTC3ERR	AFCAN3 error	AFCAN3	0560H	00000560H	nextPC	C3ERRIC
		79	INTC3WUP	AFCAN3 wakeup	AFCAN3	0570H	00000570H	nextPC	C3WUPIC
		80	INTC3REC	AFCAN3 reception completion	AFCAN3	0580H	00000580H	nextPC	C3RECIC
		81	INTC3TRX	AFCAN3 transmission completion	AFCAN3	0590H	00000590H	nextPC	C3TRXIC

Remarks 1. Default Priority: The priority order when two or more maskable interrupt requests are generated at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC or FEPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, and SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value from which the processing starts following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC − 4).

17.2 Non-Maskable Interrupts

17.2.1 Non-maskable interrupt request signal

A non-maskable interrupt request signal is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) state. An NMI is not subject to priority control and takes precedence over all the other interrupt request signals.

This product has the following two non-maskable interrupt request signals.

- NMI pin input (NMI)
- Non-maskable interrupt request signal generated by overflow of watchdog timer (INTWDT2)

The valid edge of the NMI pin can be selected from four types: “rising edge”, “falling edge”, “both edges”, and “no edge detection”.

NMI function becomes valid when the PMC02 bit of the PMC0 register is set to 1 and the INTF02/INTR02 bits of the INTF0 register is set to any value.

The non-maskable interrupt request signal generated by overflow of watchdog timer 2 (INTWDT2) functions when the WDM21 and WDM20 bits of the WDTM2 register are set to “01”.

If two or more non-maskable interrupt request signals are generated at the same time, the interrupt with the higher priority is serviced, as follows (the interrupt request signal with the lower priority is ignored).

INTWDT2 > NMI

If a new NMI or INTWDT2 request signal is issued while a NMI is being serviced, it is serviced as follows.

(1) If new NMI request signal is issued while NMI is being serviced

The new NMI request signal is held pending, regardless of the value of the NP bit of the PSW in the CPU. The pending NMI request signal is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

(2) If INTWDT2 request signal is issued while NMI is being serviced

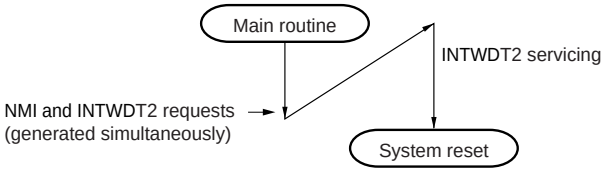
The INTWDT2 request signal is held pending if the NP bit of the PSW remains set (1) while the NMI is being serviced. The pending INTWDT2 request signal is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

If the NP bit of the PSW is cleared (0) while the NMI is being serviced, the newly generated INTWDT2 request signal is executed (the NMI servicing is stopped).

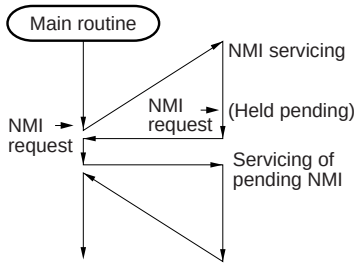
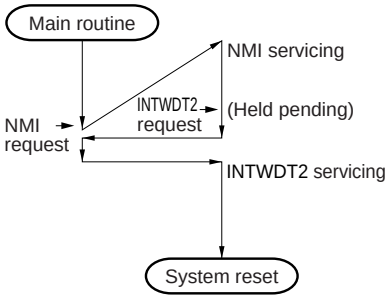
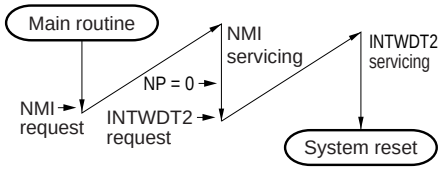
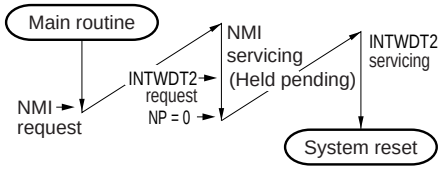
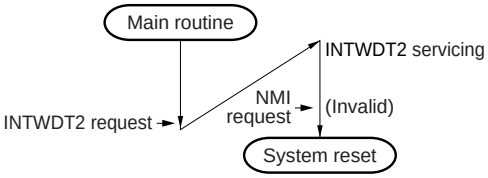
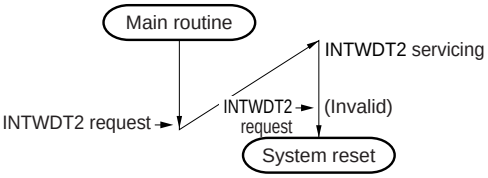
Caution If a non-maskable interrupt request signal is generated, the values of the PC and PSW are saved to the NMI status save registers (FEPC and FEPSW). At this time, execution can be returned by the RETI instruction only if the interrupt was generated by the NMI signal. Execution cannot be returned while an interrupt generated by the INTWDT2 signal is being serviced. Therefore, reset the system after the interrupt has been serviced.

Figure 17-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation

(a) NMI and INTWDT2 request signals generated at the same time



(b) Non-maskable interrupt request signal generated during non-maskable interrupt servicing

Non-maskable interrupt being serviced	Non-maskable interrupt request signal generated during non-maskable interrupt servicing	
	NMI	INTWDT2
NMI	• NMI request generated during NMI servicing 	• INTWDT2 request generated during NMI servicing (NP = 1 retained before INTWDT2 request)  • INTWDT2 request generated during NMI servicing (NP = 0 set before INTWDT2 request)  • INTWDT2 request generated during NMI servicing (NP = 0 set after INTWDT2 request) 
INTWDT2	• NMI request generated during INTWDT2 servicing 	• INTWDT2 request generated during INTWDT2 servicing 

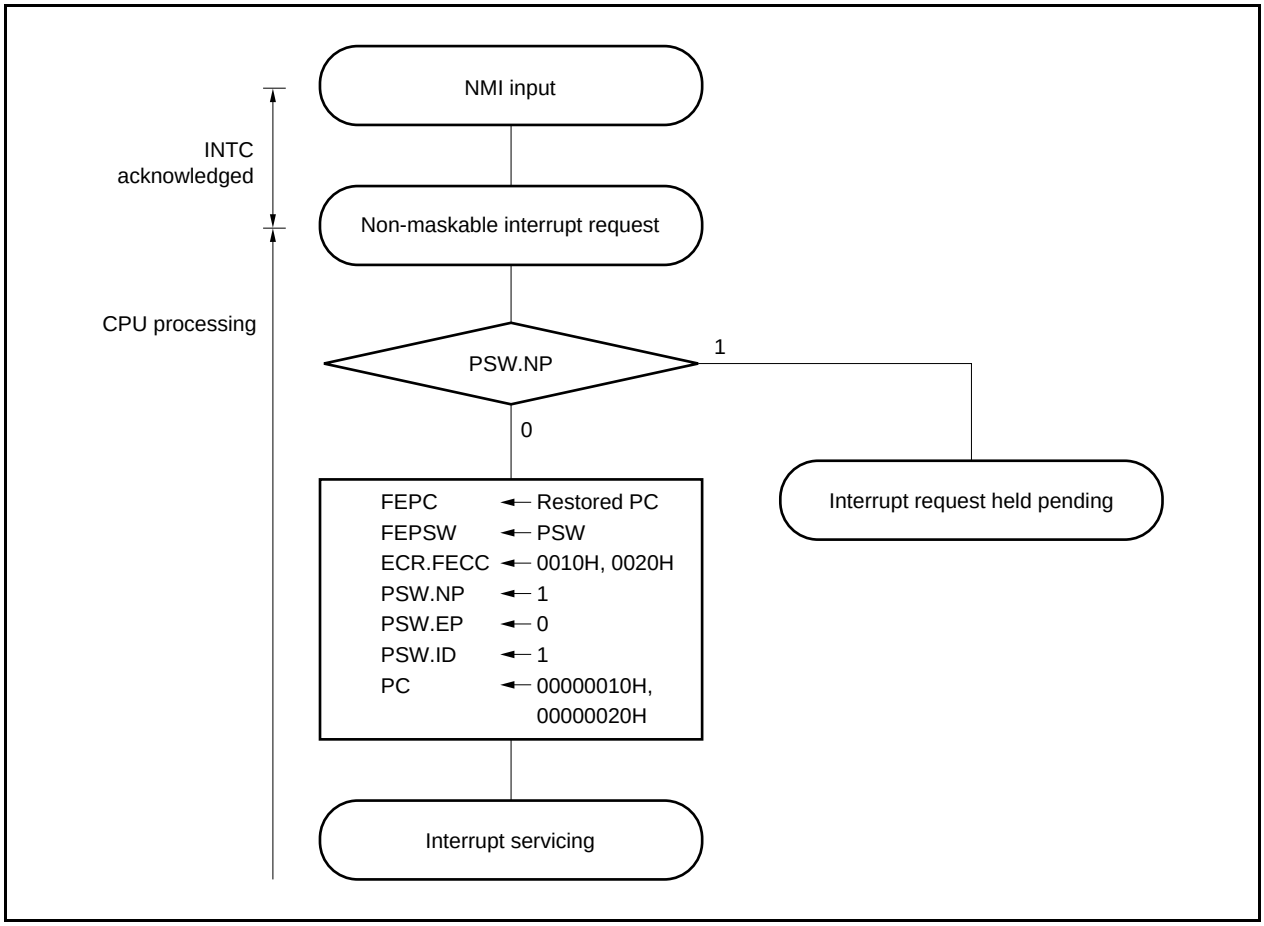
17.2.2 Operation

If a non-maskable interrupt request signal is generated, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to FEPC.
<2> Saves the current PSW to FEPSW.
<3> Writes an exception code (0010H, 0020H) 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 (00000010H, 00000020H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing configuration of a non-maskable interrupt is shown in Figure 17-2.

Figure 17-2. Servicing Configuration of Non-Maskable Interrupt



17.2.3 Restore

(1) From NMI input

Execution is restored from NMI servicing by the RETI instruction.

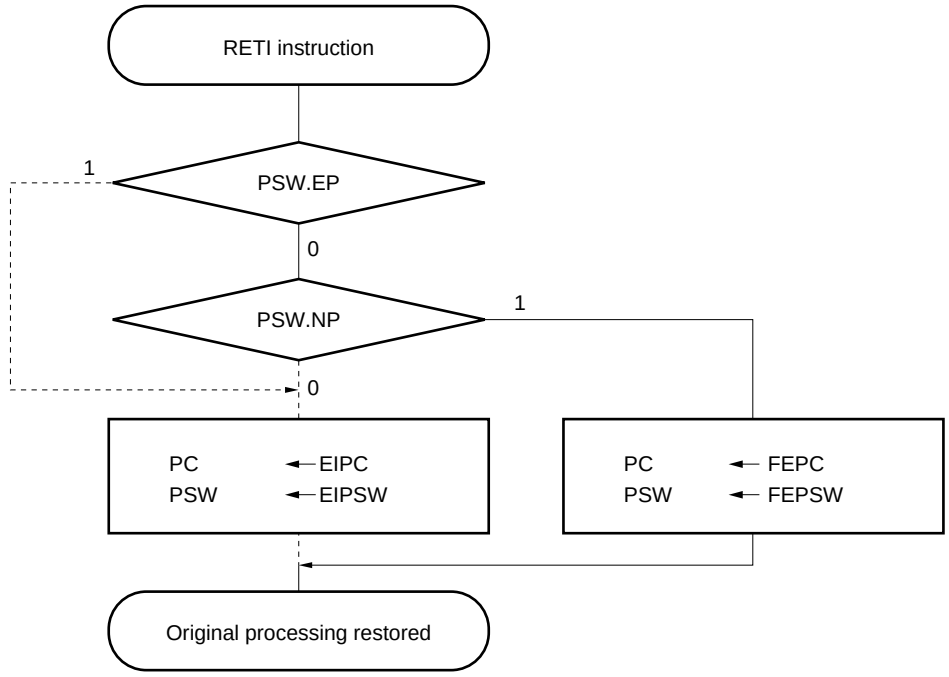
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from FEPC and FEPSW, respectively, because the 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



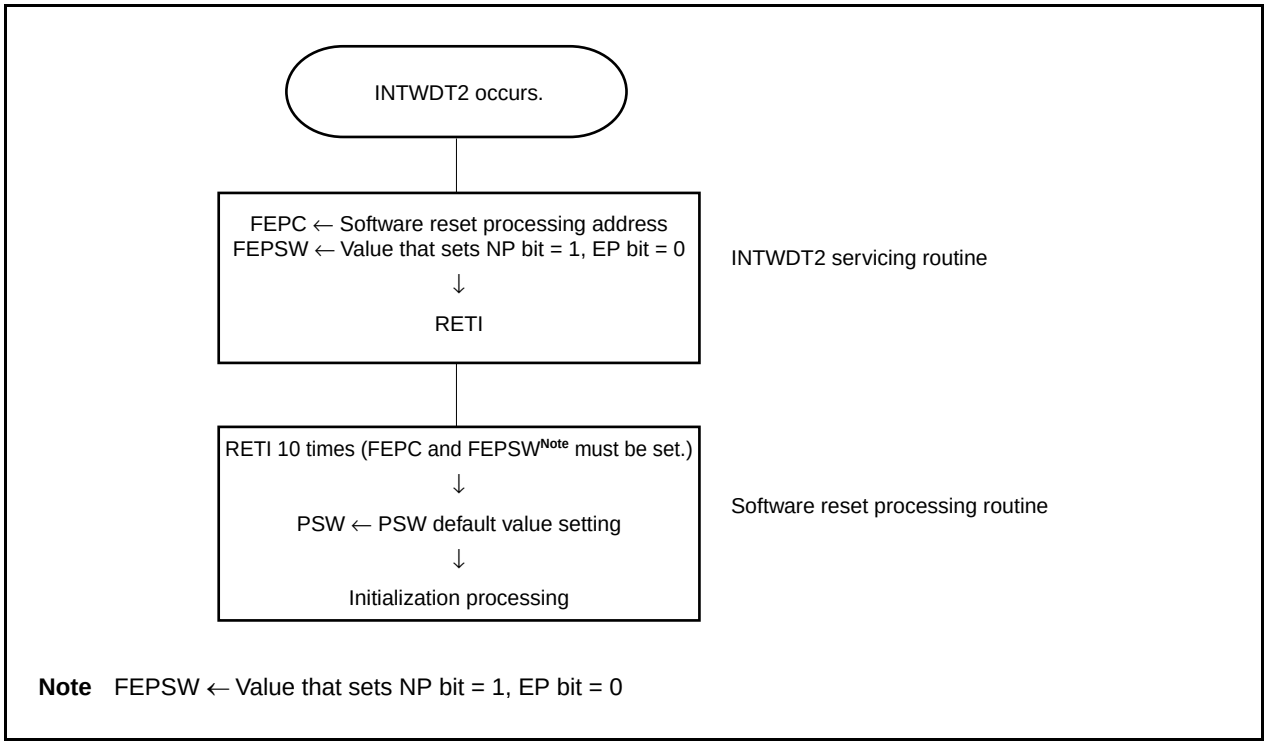
Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

(2) From INTWDT2 signal

Execution cannot be returned from INTWDT2 by the RETI instruction. Execute the following software reset processing.

Figure 22-4. Software Reset Processing



★ **Note** FEPSW ← Value that sets NP bit = 1, EP bit = 0

17.2.4 NP flag

The NP flag is a status flag that indicates that non-maskable interrupt servicing is under execution.

This flag is set when a non-maskable interrupt request signal has been acknowledged, and after that, non-maskable interrupt request is reserved.

After reset: 00000020H

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

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt currently being serviced

17.2.5 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 a 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, noise elimination using the system clock is not performed.

17.2.6 Function to detect edge of NMI pin

The valid edge of the NMI pin can be selected from four types: “rising edge”, “falling edge”, “both edges”, and “no edge detection”.

Specify the valid edge of the NMI pin by using the INTR0 and INTF0 registers.

After reset, NMI function is not valid unless the PMC02 bit of the PMC0 register is set to 1 and the INTF02/INTR02 bits of the INTF0 register is set to any value.

To use the P00/NMI pin as an I/O port pin, specify that the valid edge of the NMI pin is “no edge detection”.

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

The INTF0 register is an 8-bit register that specifies detection of the falling edge of an NMI via bit 2.

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 the INTF0n and INTR0n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0

Remark For how to specify a valid edge, see **Table 17-3**.

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

The INTR0 register is an 8-bit register that specifies detection of the rising edge of the NMI pin via bit 2.

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 the INTF0n and INTR0n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0

Remark For how to specify a valid edge, see **Table 17-3**.

Table 17-3. NMI Valid Edge Specification

INTF02	INTR02	NMI Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

17.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/Fx2 has up to 82 maskable interrupt sources.

If two or more maskable interrupt request signals are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request signal has been acknowledged, the acknowledgment of other maskable interrupt request signals is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request signal in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be serviced as multiple interrupts.

To enable multiple interrupt servicing, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

17.3.1 Operation

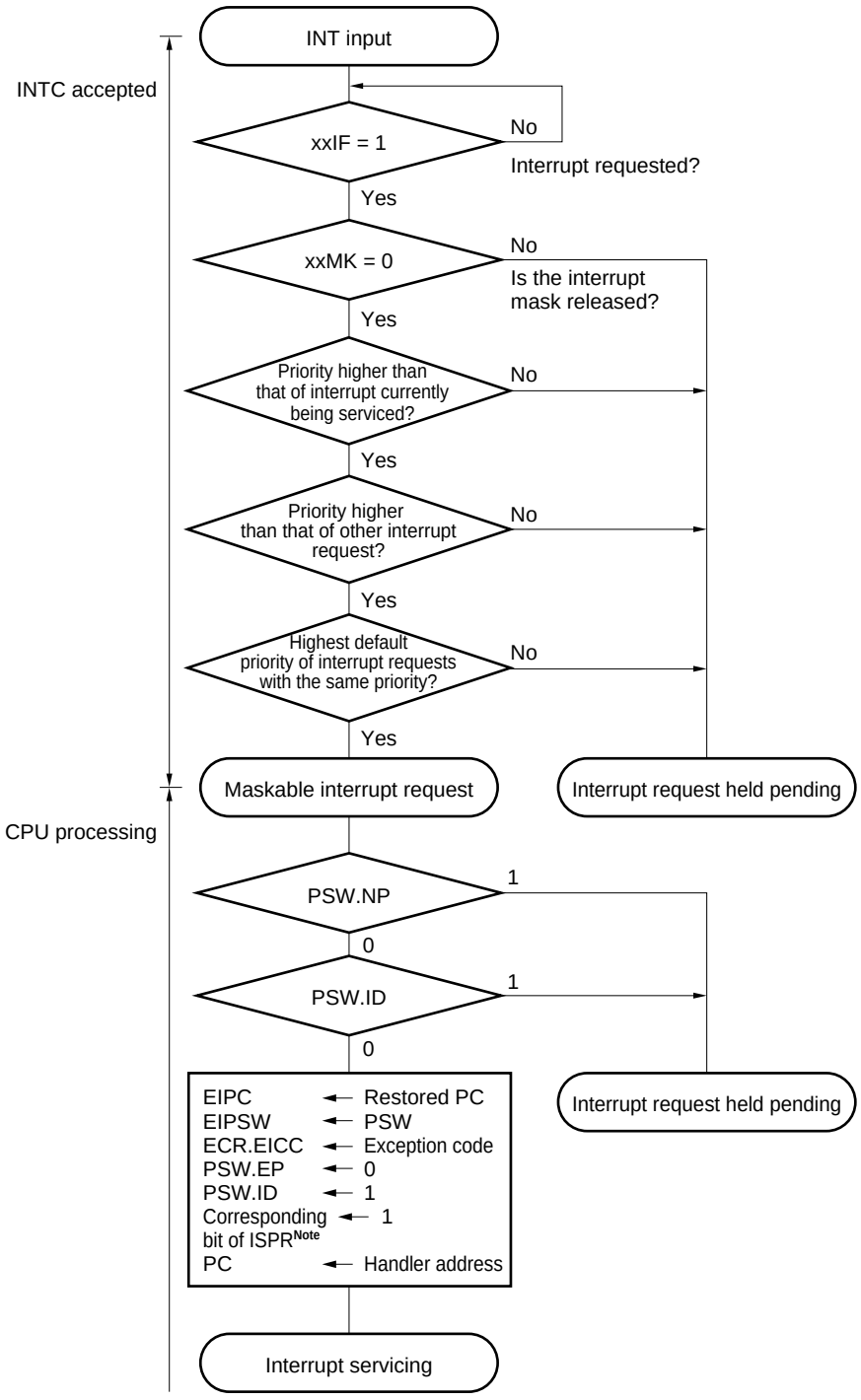
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the 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 signal masked by INTC and the maskable interrupt request signal generated while another interrupt is being serviced (while PSW.NP = 1 or PSW.ID = 1) are held pending inside the INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or 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 17-4. Maskable Interrupt Servicing



Note For details of the ISPR register, see **17.3.6 In-service priority register (ISPR)**.

17.3.2 Restore

Execution is restored from maskable interrupt servicing by the RETI instruction.

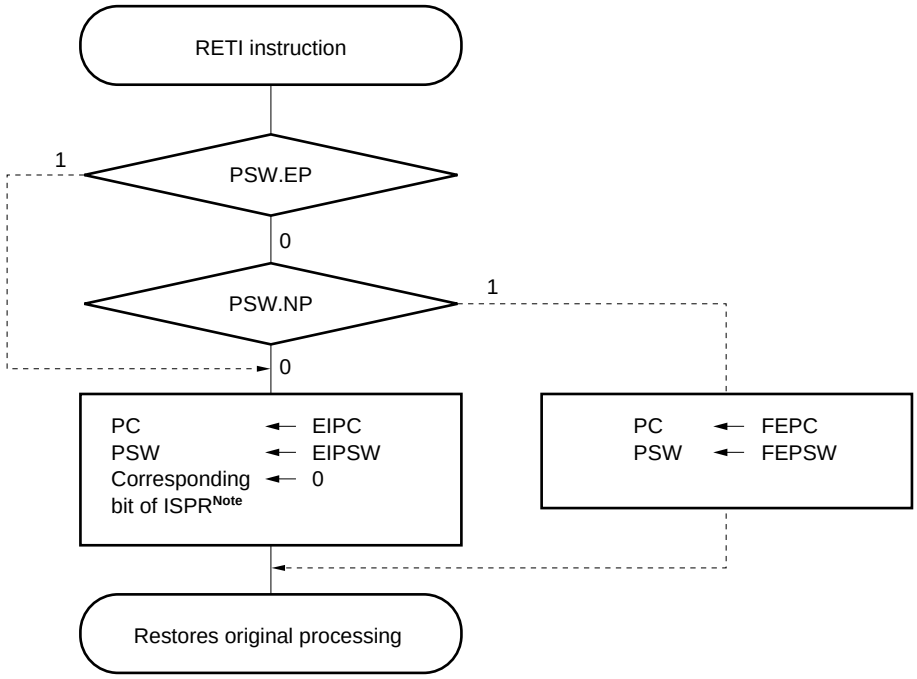
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<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 17-5 illustrates the processing of the RETI instruction.

Figure 17-5. RETI Instruction Processing



Note For details of the ISPR register, see **17.3.6 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

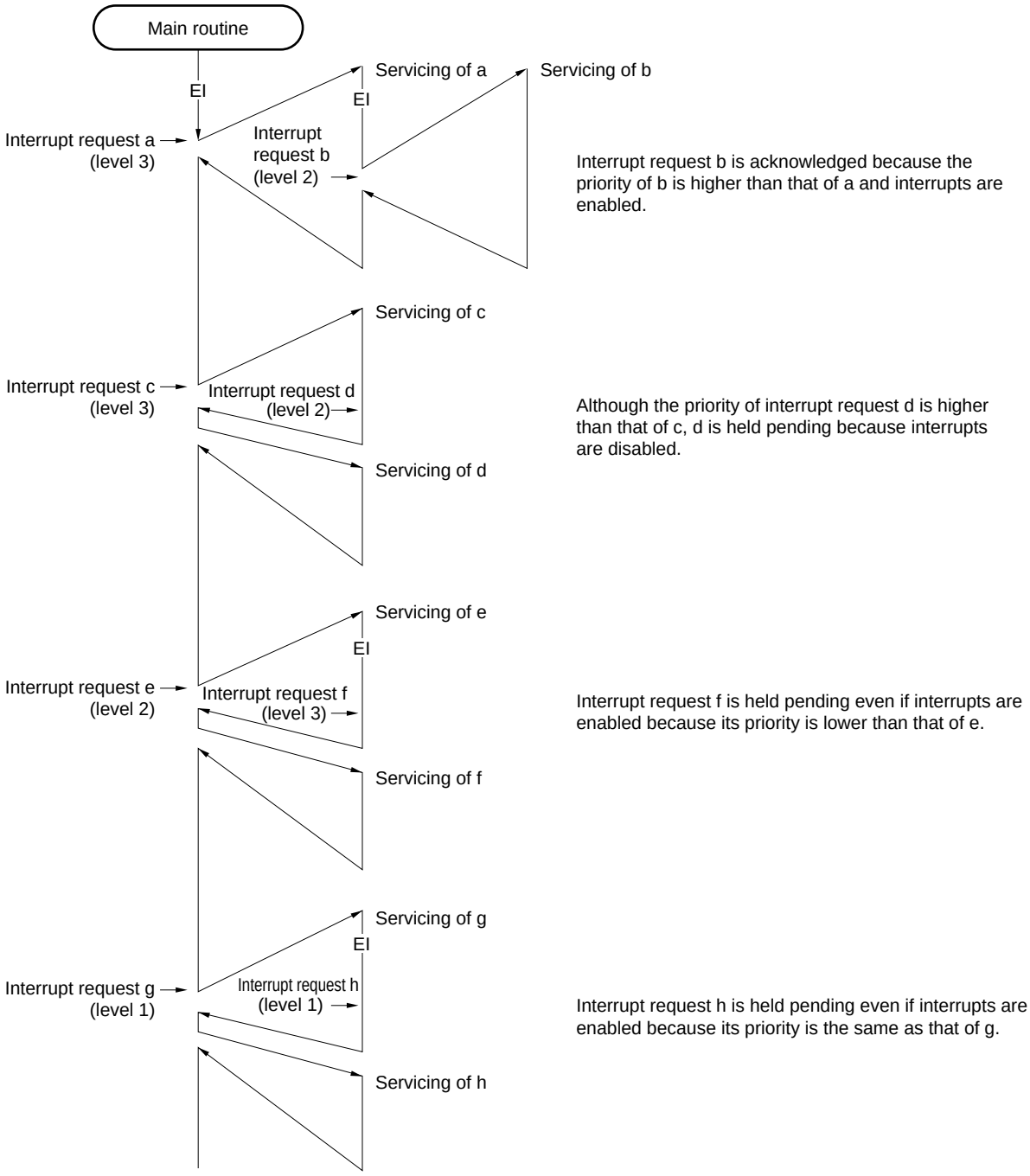
The INTC performs multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupt servicing can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupt request signals are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, see **Table 17-2 Interrupt Source List**. Programmable priority control customizes interrupt request signals into eight levels by setting the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the ID flag of the PSW is automatically set to 1. Therefore, when multiple interrupt servicing is to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt servicing program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (see **Table 17-4 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 17-4 Interrupt Control Registers (xxICn)**)

Figure 17-6. Example of Processing in Which Interrupt Request Signal Is Issued While Another Interrupt Is Being Serviced (1/2)

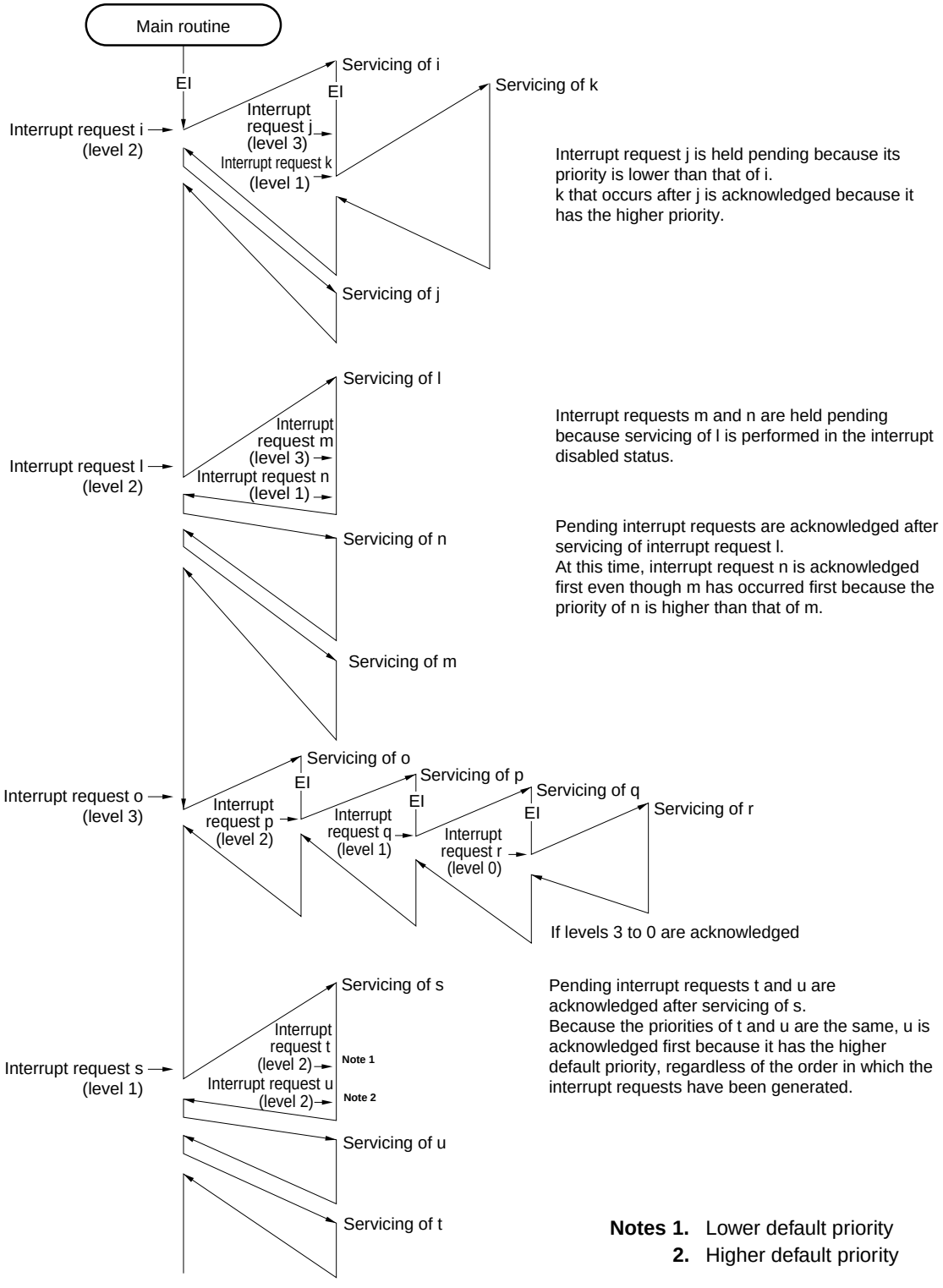


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. **"**a" to "u" in the figure are assumed names given to interrupt request signals for the sake of explanation.

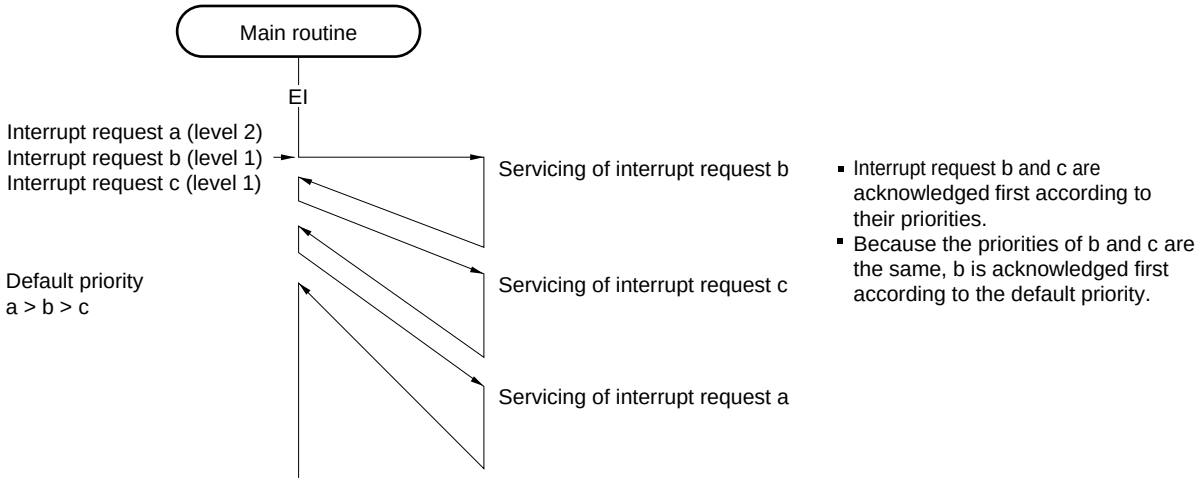
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 17-6. Example of Processing in Which Interrupt Request Signal Is Issued While Another Interrupt Is Being Serviced (2/2)



Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 17-7. Example of Servicing Interrupt Request Signals Generated Simultaneously



Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. **“**a” to “c” in the figure are assumed names given to interrupt request signals for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt request signals.

17.3.4 Interrupt control registers (xxICn)

An xxICn register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

These registers can be read or written in 8-bit or 1-bit units.

Reset input sets these registers to 47H.

Caution Disable interrupts (DI) to read the xxIFn bit of the xxICn register. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read when acknowledging an interrupt and reading the bit conflict.

After reset: 47H R/W Address: FFFFF112H to FFFFF1B2H

	7	6	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged.

Remark xx: Identification name of each peripheral unit (see **Table 17-4 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 17-4 Interrupt Control Registers (xxICn)**).

The addresses and bits of the interrupt control registers are as follows.

Table 17-4. Interrupt Control Registers (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	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
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
FFFFF15EH	C0TRXIC	C0TRXIF	C0TRXMK	0	0	0	C0TRXPR2	C0TRXPR1	C0TRXPR0
FFFFF160H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF162H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0

Table 17-4. Interrupt Control Registers (xxICn) (2/2)

Address	Register	Bit							
		7	6	5	4	3	2	1	0
FFFFF164H	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0
FFFFF166H	PIC8	PIF8	PMK8	0	0	0	PPR82	PPR81	PPR80
FFFFF168H	PIC9	PIF9	PMK9	0	0	0	PPR92	PPR91	PPR90
FFFFF16AH	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100
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
FFFFF176H	UA2RIC	UA2RIF	UA2RMK	0	0	0	UA2RPR2	UA2RPR1	UA2RPR0
FFFFF178H	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
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	PIC11	PIF11	PMK11	0	0	0	PPR112	PPR111	PPR110
FFFFF18CH	PIC12	PIF12	PMK12	0	0	0	PPR122	PPR121	PPR120
FFFFF18EH	PIC13	PIF13	PMK13	0	0	0	PPR132	PPR131	PPR130
FFFFF190H	PIC14	PIF14	PMK14	0	0	0	PPR142	PPR141	PPR140
FFFFF192H	TQ2OVIC	TQ2OVIF	TQ2OVMK	0	0	0	TQ2OVPR2	TQ2OVPR1	TQ2OVPR0
FFFFF194H	TQ2CCIC0	TQ2CCIF0	TQ2CCMK0	0	0	0	TQ2CCPR02	TQ2CCPR01	TQ2CCPR00
FFFFF196H	TQ2CCIC1	TQ2CCIF1	TQ2CCMK1	0	0	0	TQ2CCPR12	TQ2CCPR11	TQ2CCPR10
FFFFF198H	TQ2CCIC2	TQ2CCIF2	TQ2CCMK2	0	0	0	TQ2CCPR22	TQ2CCPR21	TQ2CCPR20
FFFFF19AH	TQ2CCIC3	TQ2CCIF3	TQ2CCMK3	0	0	0	TQ2CCPR32	TQ2CCPR31	TQ2CCPR30
FFFFF19CH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF19EH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0
FFFFF1A0H	UA3RIC	UA3RIF	UA3RMK	0	0	0	UA3RPR2	UA3RPR1	UA3RPR0
FFFFF1A2H	UA3TIC	UA3TIF	UA3TMK	0	0	0	UA3TPR2	UA3TPR1	UA3TPR0
FFFFF1A4H	C2ERRIC	C2ERRIF	C2ERRMK	0	0	0	C2ERRPR2	C2ERRPR1	C2ERRPR0
FFFFF1A6H	C2WUPIC	C2WUPIF	C2WUPMK	0	0	0	C2WUPPR2	C2WUPPR1	C2WUPPR0
FFFFF1A8H	C2RECIC	C2RECIF	C2RECMK	0	0	0	C2RECPR2	C2RECPR1	C2RECPR0
FFFFF1AAH	C2TRXIC	C2TRXIF	C2TRXMK	0	0	0	C2TRXPR2	C2TRXPR1	C2TRXPR0
FFFFF1ACH	C3ERRIC	C3ERRIF	C3ERRMK	0	0	0	C3ERRPR2	C3ERRPR1	C3ERRPR0
FFFFF1AEH	C3WUPIC	C3WUPIF	C3WUPMK	0	0	0	C3WUPPR2	C3WUPPR1	C3WUPPR0
FFFFF1B0H	C3RECIC	C3RECIF	C3RECMK	0	0	0	C3RECPR2	C3RECPR1	C3RECPR0
FFFFF1B2H	C3TRXIC	C3TRXIF	C3TRXMK	0	0	0	C3TRXPR2	C3TRXPR1	C3TRXPR0

17.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR4, IMR5L)

The IMR0 to IMR4, IMR5L registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR4 and IMR5L 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 4).

The IMR5L register can be read or written in 8-bit or 1-bit units.

If the higher 8 bits of the IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units (m = 0 to 4).

Reset input sets these registers to FFFFH.

Bits 7 to 2 of the IMR5L register are fixed to 1. If these bits are not 1, the operation cannot be guaranteed.

Reset input sets these registers to FFFFH.

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).

(1/2)

After reset: FFH R/W Address: FFFF10AH

	7	6	5	4	3	2	1	0
IMR5L	1	1	1	1	1	1	C3TRXMK	C3RECMK

Caution Set bits 7 to 2 of the IMR5L register to 1.

After reset: FFFFH R/W Address: FFFF108H

	15	14	13	12	11	10	9	8
IMR4	C3WUPMK	C3ERRMK	C2TRXMK	C2RECMK	C2WUPMK	C2ERRMK	UA3TMK	UA3RMK
	7	6	5	4	3	2	1	0
	CB2TMK	CB2RMK	TQ2CCMK3	TQ2CCMK2	TQ2CCMK1	TQ2CCMK0	TQ2OVMK	PMK14

After reset: FFFFH R/W Address: FFFF106H

	15	14	13	12	11	10	9	8
IMR3	PMK13	PMK12	PMK11	DMAMK3	DMAMK2	DMAMK1	DMAMK0	C1TRXMK
	7	6	5	4	3	2	1	0
	C1RECMK	C1WUPMK	C1ERRMK	UA2TMK	UA2RMK	TQ1CCMK3	TQ1CCMK2	TQ1CCMK1

After reset: FFFFH R/W Address: FFFF104H

	15	14	13	12	11	10	9	8
IMR2	TQ1CCMK0	TQ1OVMK	PMK10	PMK9	PMK8	WTMK	WTIMK	KRMK
	7	6	5	4	3	2	1	0
	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	ADMK	UA1TMK	UA1RMK	UA0TMK

(2/2)

After reset: FFFFH R/W Address: FFFF102H

IMR1

15	14	13	12	11	10	9	8
UA0RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK	TM0EQMK0	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 (see **Table 17-4 Interrupt Control Registers (xxICn)**).

n: Peripheral unit number (see **Table 17-4 Interrupt Control Registers (xxICn)**)

17.3.6 In-service priority register (ISPR)

The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt request signal is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request signal having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution If an interrupt is acknowledged while the ISPR register is being read in the interrupt enabled (EI) state, the value of the ISPR register after the bits of the register have been set by acknowledging the interrupt may be read. To accurately read the value of the ISPR register before an interrupt is acknowledged, read the register while interrupts are disabled (DI).

After reset: 00H R Address: FFFFF1FAH

	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently acknowledged
0	Interrupt request signal with priority n not acknowledged
1	Interrupt request signal with priority n acknowledged

Remark n = 0 to 7 (priority level)

17.3.7 ID flag

This is the interrupt disable flag and controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt request signals. The ID flag is assigned to the PSW.

After reset: 00000020H

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

ID	Specification of maskable interrupt servicing[Note]
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled

Note Interrupt disable flag (ID) function

This bit is set to 1 by the DI instruction and 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 request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set to 1 by hardware.

An interrupt request signal generated during the acknowledgment disabled period (ID = 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)

The WDTM2 register is a special register and can only be written in a specific sequence.

This register can be read or written in 8-bit units (for details, see **CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2**).

Reset input sets this register to 67H.

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 watchdog timer operation mode
0	0	Stops operation
0	1	Non-maskable interrupt request mode
1	×	Reset mode (initial-value)

Remark For the WDCS24 to WDCS20 bits refer to **CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2**.

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 a specific time). Noise elimination by analog delay or digital noise elimination can be selected for the INTP3 pin.

17.3.10 Function to detect edge of INTP0 to INTP14 pins

The valid edge of the INTP0 to INTP14 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)

The INTF0 register is an 8-bit register that specifies detection of the falling edge of the non-maskable interrupt pin (NMI) via bit 2 or external interrupt pins (INTP0 to INTP3) via bits 3 to 6.

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

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF0n and INTR0n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0

Remark For how to specify a valid edge, see **Table 17-5**.

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

The INTR0 register is an 8-bit register that specifies detection of the rising edge of the non-maskable interrupt pin (NMI) via bit 2 or external interrupt pins (INTP0 to INTP3) via bits 3 to 6.

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

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF0n and INTR0n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0

Remark For how to specify a valid edge, see **Table 17-5**.

Table 17-5. Valid Edge Specification (INTF0n, INTR0n)

INTF0n	INTR0n	Valid Edge Specification (n = 2 to 6)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 2: NMI pin control
n = 3: INTP0 pin control
n = 4: INTP1 pin control
n = 5: INTP2 pin control
n = 6: INTP3 pin control

Caution Be sure to clear the INTF0n and INTR0n bits to 00 when these registers are not specified as NMI or INTP0 to INTP3.

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

The INTF1 register is an 8-bit register that specifies detection of the falling edge of external interrupt pins (INTP9, INTP10) via bits 0 and 1.

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

Reset input cleats this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF1n and INTR1n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC02H

	7	6	5	4	3	2	1	0
INTF1	0	0	0	0	0	0	INTF11	INTF10

Remark For how to specify a valid edge, see **Table 17-6**.

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

The INTR1 register is an 8-bit register that specifies detection of the rising edge of external interrupt pins (INTP9, INTP10) via bits 0 and 1.

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

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF1n and INTR1n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC22H

	7	6	5	4	3	2	1	0
INTR1	0	0	0	0	0	0	INTR11	INTR10

Remark For how to specify a valid edge, see **Table 17-6**.

Table 17-6. Valid Edge Specification (INTF1n, INTR1n)

INTF1n Bit	INTR1n Bit	Valid Edge Specification (n = 0, 1)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0: INTP9 pin control
n = 1: INTP10 pin control

Caution Be sure to clear the INTF1n and INTR1n bits to 00 when these registers are not used as INTP9 and INTP10.

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

The INTF3 register is a 16-bit register that specifies detection of the falling edge of external interrupt pins (INTP7, INTP8) via bits 1 and 9. This register can be read or written in 16-bit units.

However, when the higher 8 bits of INTF3 register are used as the INTF3H register and the lower 8 bits as the INTF3L register, they can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 0000H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF3n and INTR3n bits to 0, and then set the port mode.

After reset: 0000H R/W Address: FFFFFC06H, FFFFFC07H

	15	14	13	12	11	10	9	8
INTF3 (INTF3H[Note])	0	0	0	0	0	0	INTF39	0
	7	6	5	4	3	2	1	0
(INTF3L)	0	0	0	0	0	0	INTF31	0

Note When bits 8 to 15 of the INTF3 register are read or written in 8-bit or 1-bit units, specify them as bits 0 to 7 of the INTF3H register

Remark For how to specify a valid edge, see **Table 17-7**.

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

The INTR3 register is a 16-bit register that specifies detection of the rising edge of external interrupt pins (INTP7, INTP8) via bits 1 and 9. This register can be read or written in 16-bit units.

However, when the higher 8 bits of INTR3 register are used as the INTR3H register and the lower 8 bits as the INTR3L register, they can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 0000H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF3n and INTR3n bits to 0, and then set the port mode.

After reset: 0000H R/W Address: FFFFFC26H, FFFFFC27H

	15	14	13	12	11	10	9	8
INTR3 (INTR3H[Note])	0	0	0	0	0	0	INTR39	0
	7	6	5	4	3	2	1	0
(INTR3L)	0	0	0	0	0	0	INTR31	0

Note When bits 8 to 15 of the INTR3 register are read or written in 8-bit or 1-bit units, specify them as bits 0 to 7 of the INTR3H register

Remark For how to specify a valid edge, see **Table 17-7**.

Table 17-7. Valid Edge Specification (INTF3n, INTR3n)

INTF3n Bit	INTR3n Bit	Valid Edge Specification (n = 1, 9)
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 1: INTP7 pin control
n = 9: INTP8 pin control

Caution Be sure to clear the INTF3n and INTR3n bits to 00 when these registers are not specified as INTP7 and INTP8.

(7) External interrupt falling edge specification register 6L (INTF6L)

The INTF6L register is an 8-bit register that specifies detection of the falling edge of external interrupt pins (INTP11 to INTP13) via bits 0 to 2. This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF6n and INTR6n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC0CH

	7	6	5	4	3	2	1	0
INTF6L	0	0	0	0	0	INTF62	INTF61	INTF60

Remark For how to specify a valid edge, see **Table 17-8**.

(8) External interrupt rising edge specification register 6L (INTR6L)

The INTR6L register is an 8-bit register that specifies detection of the rising edge of external interrupt pins (INTP11 to INTP13) via bits 0 to 2. This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF6n and INTR6n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC2CH

	7	6	5	4	3	2	1	0
INTR6L	0	0	0	0	0	INTR62	INTR61	INTR60

Remark For how to specify a valid edge, see **Table 17-8**.

Table 17-8. Valid Edge Specification (INTF6n, INTR6n)

INTF6n Bit	INTR6n Bit	Valid Edge Specification (n = 0 to 2)
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0: INTP11 pin control
n = 1: INTP12 pin control
n = 2: INTP13 pin control

Caution Be sure to clear the INTF6n and INTR6n bits to 00 when these registers are not specified as INTP11 to INTP13.

(9) External interrupt falling edge specification register 8 (INTF8)

The INTF8 register is an 8-bit register that specifies detection of the falling edge of an external interrupt pin (INTP8) via bit 0. This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF80 and INTR80 bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC10H

	7	6	5	4	3	2	1	0
INTF8	0	0	0	0	0	0	0	INTF80

Remark For how to specify a valid edge, see **Table 17-9**.

(10) External interrupt rising edge specification register 8 (INTR8)

The INTR8 register is an 8-bit register that specifies detection of the rising edge of an external interrupt pin (INTP8) using bit 0. This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF80 and INTR80 bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC30H

	7	6	5	4	3	2	1	0
INTR8	0	0	0	0	0	0	0	INTR80

Remark For how to specify a valid edge, see **Table 17-9**.

Table 17-9. Valid Edge Specification (INTF80, INTR80)

INTF80 Bit	INTR80 Bit	Valid Edge Specification
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark INTP14 pin control

Caution Be sure to clear the INTF80 and INTR80 bits to 00 when these registers are not specified as INTP14.

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

The INTF9H register is an 8-bit register that specifies detection of the falling edge of external interrupt pins (INTP4 to INTP6) via bits 5 to 7.

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

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF9n and INTR9n bits to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC13H

	7	6	5	4	3	2	1	0
INTF9H	INTH915	INTF914	INTF913	0	0	0	0	0

Remark For how to specify a valid edge, see **Table 17-10**.

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

The INTR9H register is an 8-bit register that specifies detection of the rising edge of external interrupt pins (INTP4 to INTP6) via bits 5 to 7.

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

Reset input clears this register to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF9n and INTR9n bits to 0, and then set the port mode.

After reset: 00H R/W Address: 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, see **Table 17-10**.

Table 17-10. Valid Edge Specification (INTF9n, INTR9n)

INTF9n Bit	INTR9n Bit	Valid Edge Specification (n = 13 to 15)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 13: INTP4 pin control
n = 14: INTP5 pin control
n = 15: INTP6 pin control

Caution Be sure to clear the INTF9n and INTR9n bits to 00 when these registers are not used as INTP4 to INTP6.

(13) Noise elimination control register (NFC)

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are performed using the NFC register.

When digital noise elimination is selected, the sampling clock for digital sampling can be selected from among $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$, $f_{XX}/1,024$, and f_{XT}. Sampling is performed 2 or 3 times.

Even when digital noise elimination is selected, using fXT as the sampling clock makes it possible to use the INTP3 interrupt request signal to release the IDLE1, IDLE2 and Software STOP modes.

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

Reset input clears this register to 00H.

Caution After the sampling clock has been changed; it takes "set number by NFSTS bit" sampling clocks to initialize the digital noise eliminator. Therefore, if an INTP3 valid edge is input within these "set number by NFSTS bit" sampling clocks after the sampling clock has been changed, an interrupt request signal may be generated. Therefore, be careful about the following points when using the interrupt and DMA functions.

- **When using the interrupt function, after the "set number by NFSTS bit" sampling clocks have elapsed, enable interrupts after the interrupt request flag (bit 7 of PIC3) has been cleared.**
- **When using the DMA function (started by INTP3), enable DMA after "set number by NFSTS bit" sampling clocks have elapsed.**

After reset: 00H R/W Address: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	Settings of INTP3 pin noise elimination
0	Analog noise elimination (60 ns (TYP.))
1	Digital noise elimination

NFSTS	Setting of sampling performed for digital noise elimination
0	Sampling performed = 3
1	Sampling performed = 2

NFC2	NFC1	NFC0	Digital sampling clock
0	0	0	$f_{XX}/64$
0	0	1	$f_{XX}/128$
0	1	0	$f_{XX}/256$
0	1	1	$f_{XX}/512$
1	0	0	$f_{XX}/1,024$
1	0	1	f_{XT} (subclock)
Other than above			Setting prohibited

Remarks 1. Since sampling is performed 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 signal is generated if noise synchronized with the sampling clock is input.

17.4 Software Exceptions

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

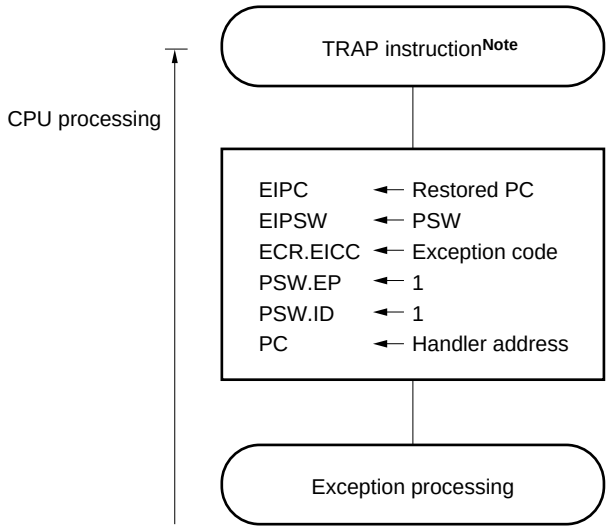
17.4.1 Operation

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 17-8 illustrates the processing of a software exception.

Figure 17-8. 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

Execution is restored from software exception processing by the RETI instruction.

When the RETI instruction is executed, the CPU carries out the following processing and shifts control to the restored PC's address.

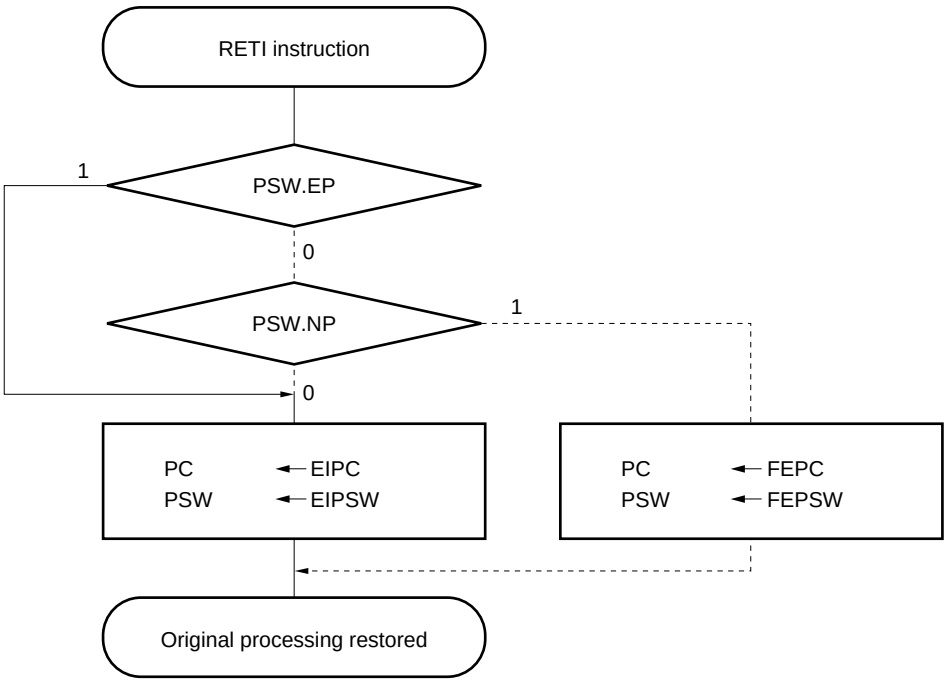
<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 1.

<2> Transfers control to the address of the restored PC and PSW.

Caution DBPC and DBPSW can be access from execution of the DBTRAP instruction or illegal instruction till execution of the DBRET instruction only.

Figure 17-9 illustrates the processing of the RETI instruction.

Figure 17-9. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set 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 EP flag

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.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	**EP**	ID	SAT	CY	OV	S	Z

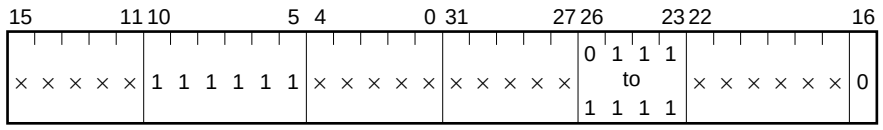
EP	Exception processing status
0	Exception processing not in progress.
1	Exception processing in progress.

17.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/FX2, 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.



×: 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.

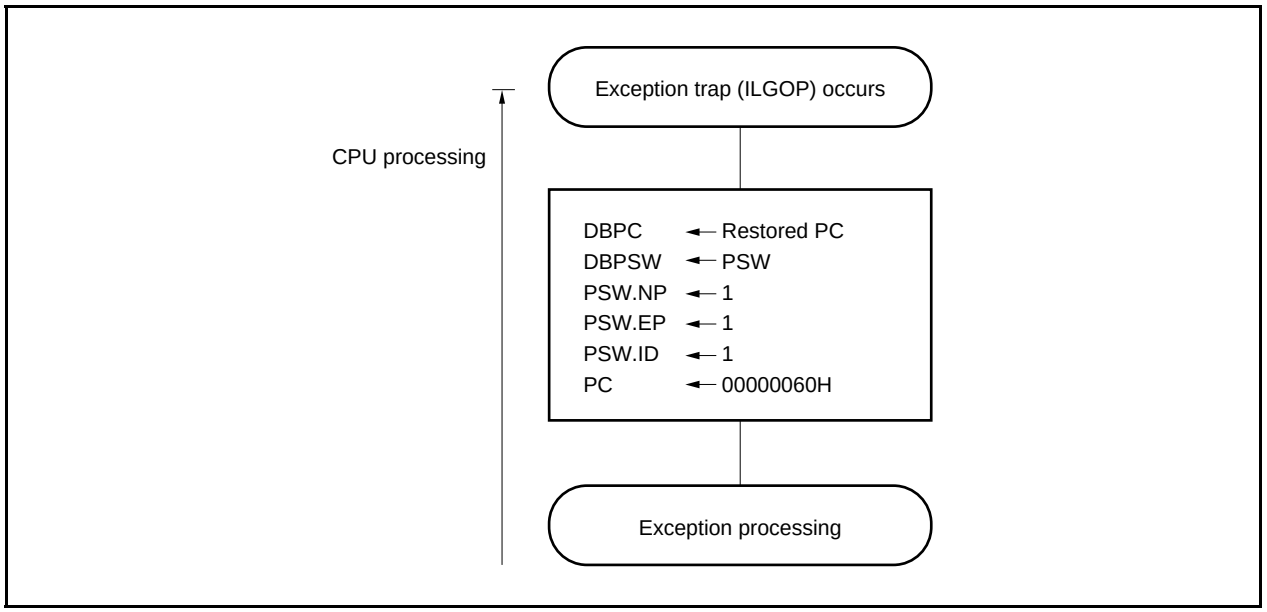
(1) Operation

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 17-10 illustrates the processing of the exception trap.

Figure 17-10. Exception Trap Processing



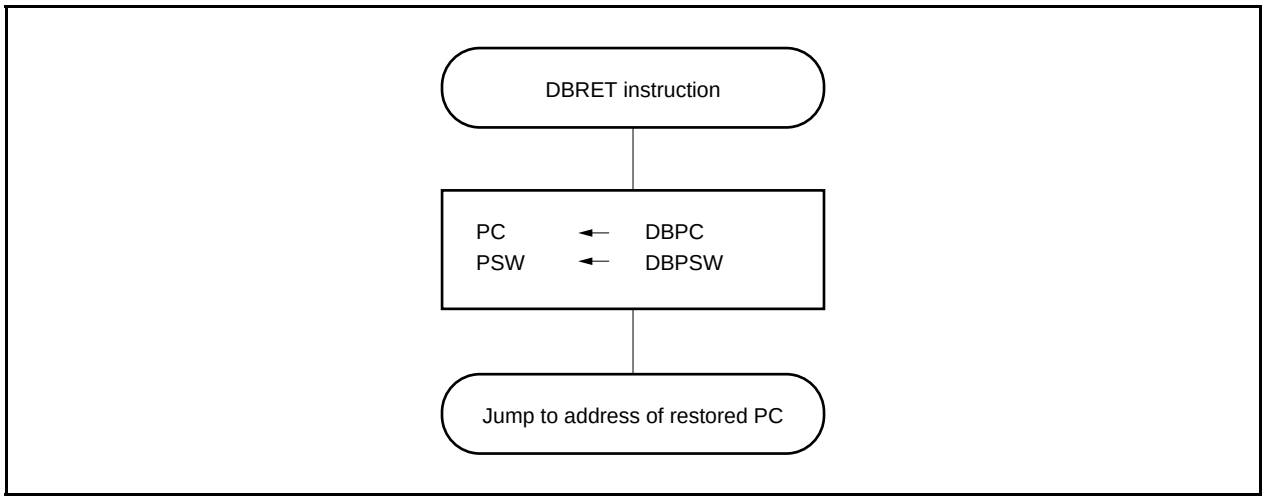
(2) Restore

Execution is restored from an exception trap by the DBRET instruction. When the DBRET instruction is executed, 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 17-11 illustrates the processing for restoring from an exception trap.

Figure 17-11. Processing of Restoration 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.

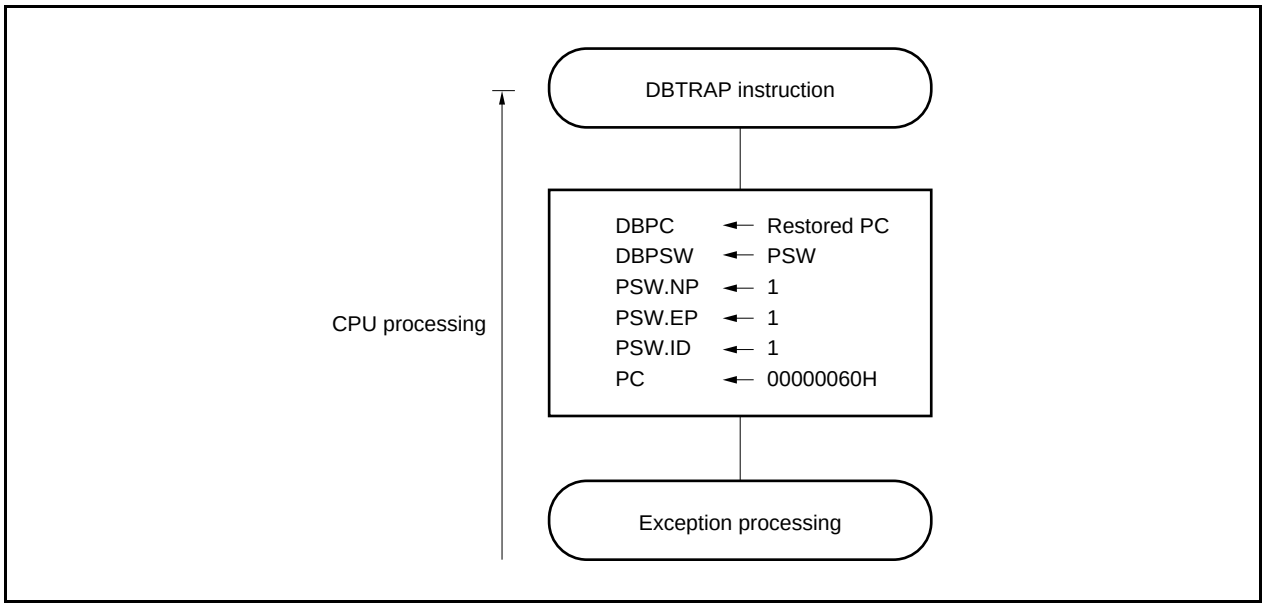
Upon occurrence of a debug trap, the CPU performs the following processing.

(1) Operation

<1> Saves 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) for the debug trap to the PC and transfers control.

Figure 17-12 illustrates the processing of the debug trap.

Figure 17-12. Debug Trap Processing



(2) Restore

Execution is restored from a debug trap by the DBRET instruction.

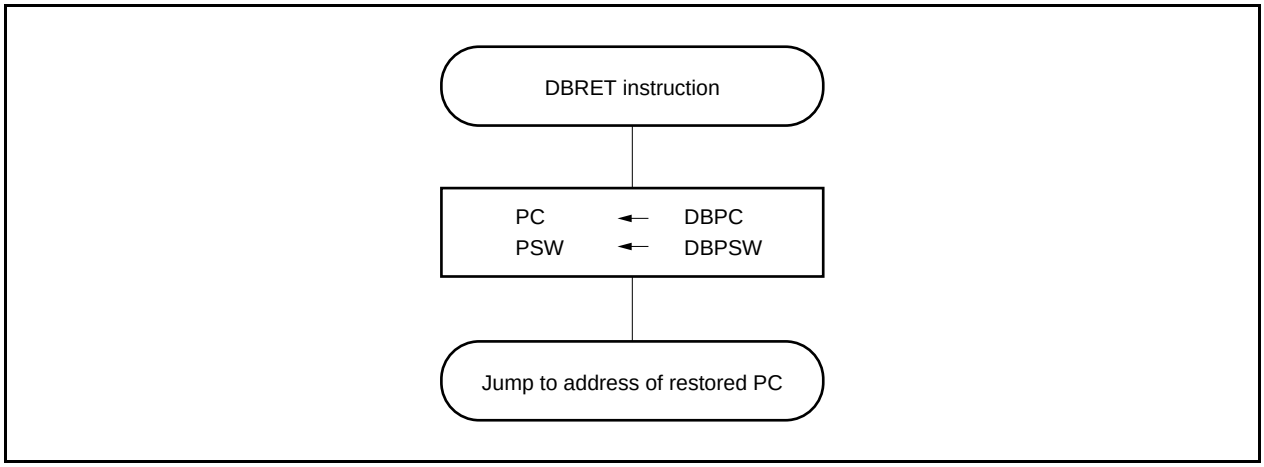
When the DBRET instruction is executed, the CPU carries out the following processing and transfers control to the address of the restored PC.

<1> Reads the restored PC and PSW from DBPC and DBPSW.

<2> Transfers control to the fetched address of the restored PC and PSW.

Table 17-13 illustrates the processing for restoring from a debug trap.

Figure 17-13. Processing of Restoration from Debug Trap

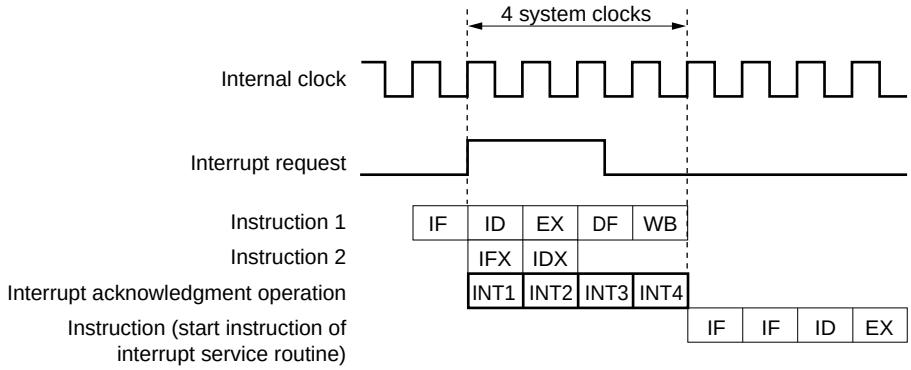


17.6 Interrupt Acknowledgment Time of CPU

Except the following cases, the interrupt acknowledgment time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 4 clocks after the preceding interrupt.

- In software STOP mode
- When the external bus is accessed
- When interrupt request non-sampling instructions are successively executed (see **17.7 Periods in Which Interrupts Are Not Acknowledged by CPU**.)
- When an interrupt control register is accessed

Figure 17-14. Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt acknowledgment time (internal system clock)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE1/IDLE2/Software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to peripheral I/O register and programmable peripheral I/O register
Maximum	6	6 + 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)
- The store, SET1, NOT1, or CLR1 instructions for the following registers.
- Interrupt-related registers:
    - Interrupt control register (xxICn), interrupt mask registers 0 to 4 (IMR0 to IMR4)
    - In-service priority register (ISPR)
    - Command register (PRCMD)
    - Power save control register (PSC)
    - On-chip debug mode register (OCDM)
    - Peripheral emulation register 1 (PEMU1)

Remark xx: Identification name of each peripheral unit (see **Table 17-4 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 17-4 Interrupt Control Registers (xxICn)**).

CHAPTER 18 KEY INTERRUPT FUNCTION

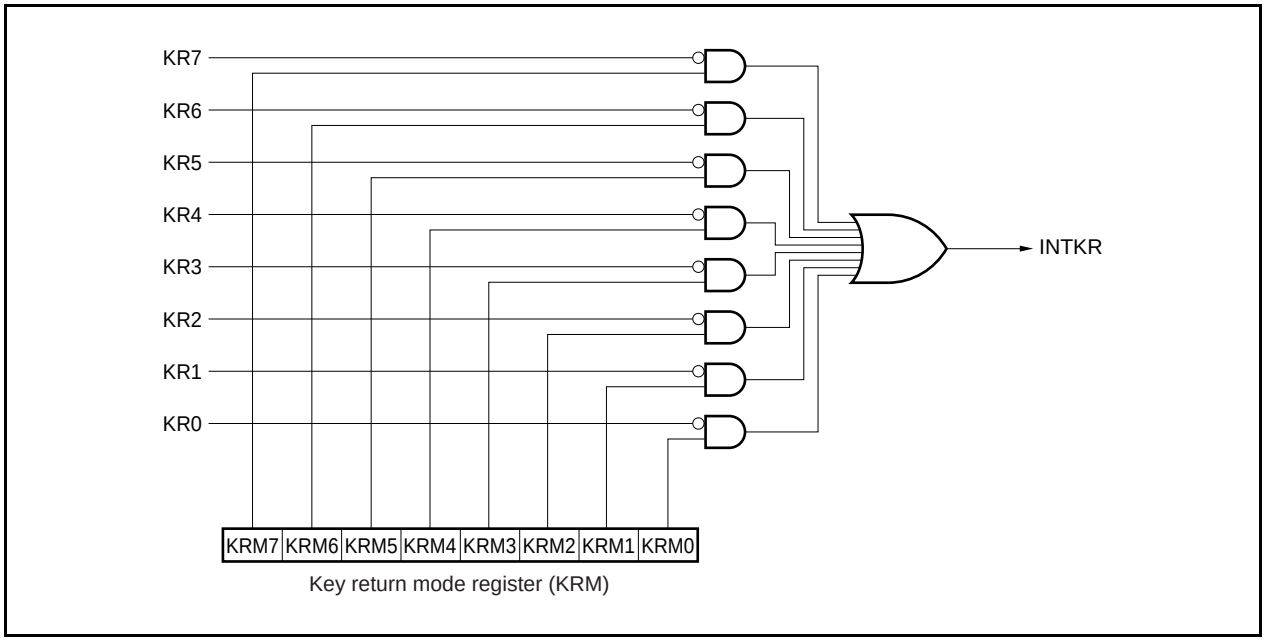
18.1 Function

A key interrupt request signal (INTKR) can be generated by inputting a falling edge to the eight key input pins (KR0 to KR7) by setting the key return mode register (KRM).

Table 18-1. Assignment of Key Return Detection Pins

Flag	Pin Description
KRM0	Controls KR0 signal in 1-bit units
KRM1	Controls KR1 signal in 1-bit units
KRM2	Controls KR2 signal in 1-bit units
KRM3	Controls KR3 signal in 1-bit units
KRM4	Controls KR4 signal in 1-bit units
KRM5	Controls KR5 signal in 1-bit units
KRM6	Controls KR6 signal in 1-bit units
KRM7	Controls KR7 signal in 1-bit units

Figure 18-1. Key Return Block Diagram



18.2 Control Register

(1) Key return mode register (KRM)

The KRM register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Control of key return mode
0	Does not detect key return signal
1	Detects key return signal

Caution Rewrite the KRM register after once clearing the KRM register to 00H.

Remark For the alternate-function pin settings, see **Table 4-25 Register Settings to Use Port Pins as Alternate-Function Pins (3/7)**

18.3 Cautions

(1) If a low level is input to any of the KR0 to KR7 pins, the INTKR signal is not generated even if the falling edge of another pin is input.

(2) The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0).

★ (3) If the KRM register is changed, an interrupt request signal (INTKR) may be generated. To prevent this, change the KRM register after disabling interrupts (DI) or masking, then clear the interrupt request flag (KRIC.KRIF bit) to 0, and enable interrupts (EI) or clear the mask.

(4) To use the key interrupt function, be sure to set the port pin to the key return pin and then enable the operation with the KRM register. To switch from the key return pin to the port pin, disable the operation with the KRM register and then set the port pin.

CHAPTER 19 STANDBY FUNCTION

19.1 Overview

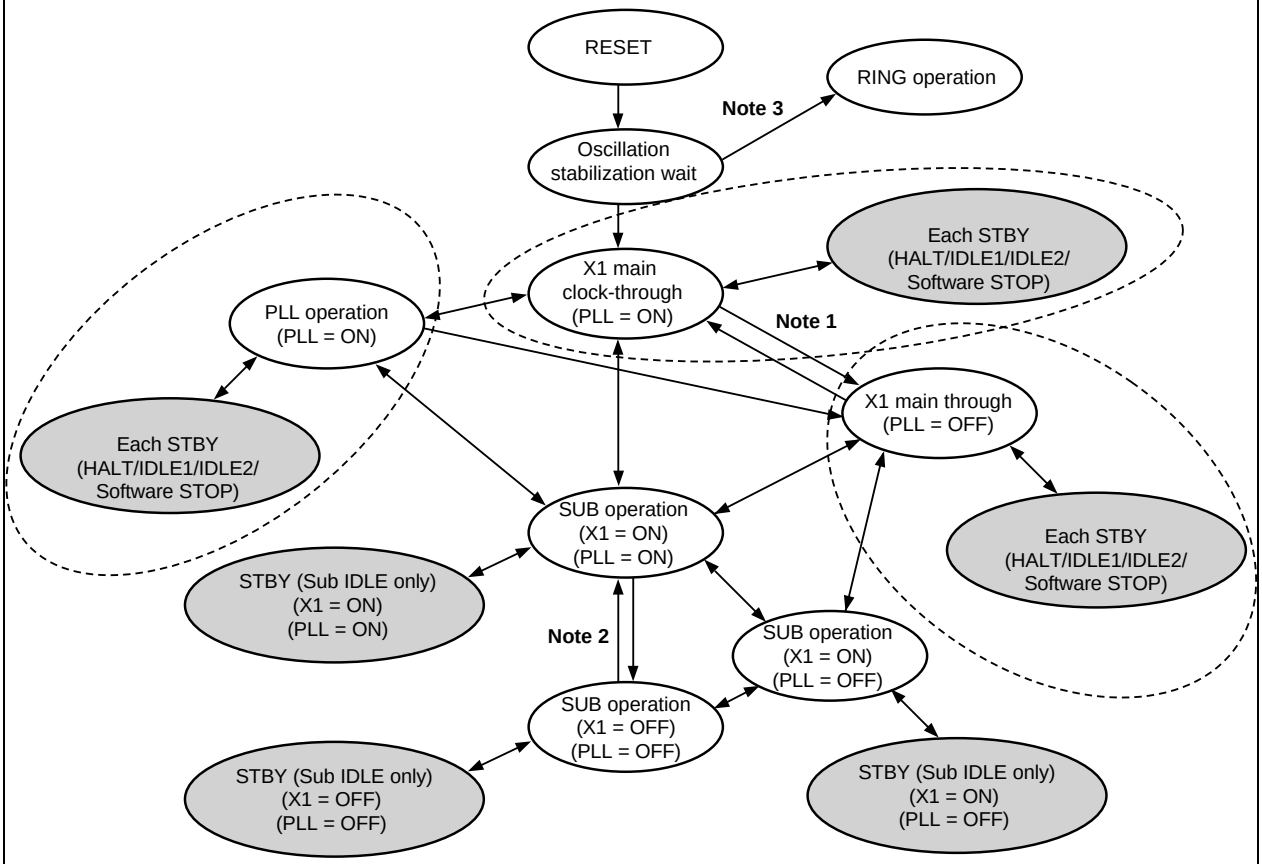
The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed below.

Table 19-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode in which only the operating clock of the CPU is stopped
IDLE1 mode	Mode in which all the internal operations of the chip except the oscillator, PLL[Note], and flash memory are stopped
IDLE2 mode	Mode in which all the internal operations of the chip except the oscillator are stopped
Software STOP mode	Mode in which all the internal operations of the chip except the subclock oscillator are stopped
Subclock operation mode	Mode in which the subclock is used as the internal system clock
Sub-IDLE mode	Mode in which all the internal operations of the chip except the oscillator, PLL[Note], and flash memory are stopped, in the subclock operation mode

Note The PLL holds the previous operating status (in clock-through mode or PLL mode).

★ **Figure 19-1. Status Transition Diagram**



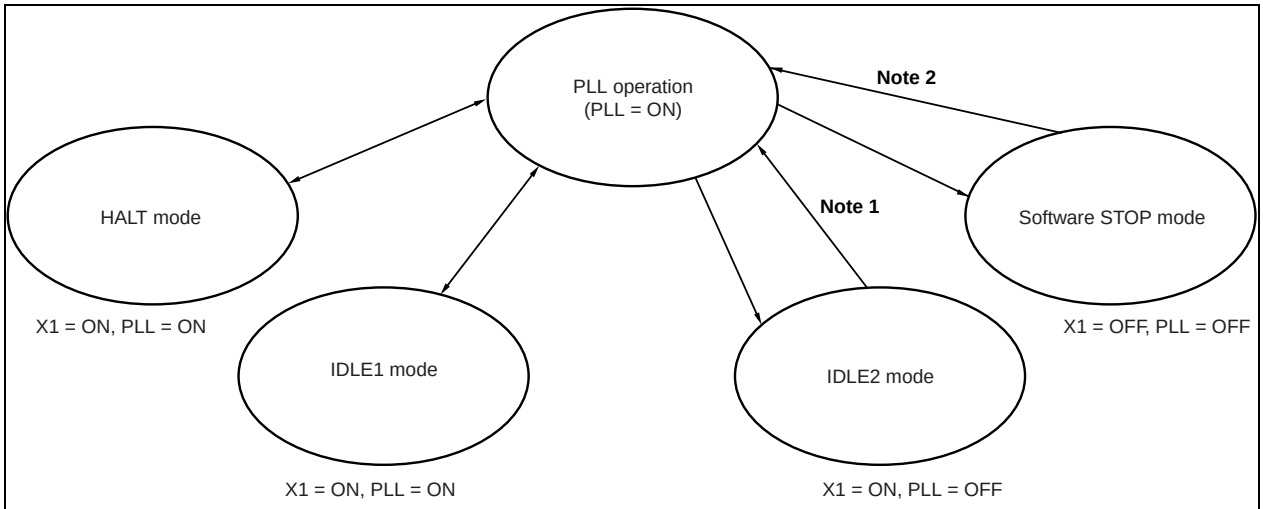
Notes 1. PLL lockup time is required (LOCK bit of LOCKR register = 1 → 0).

2. Oscillation stabilization time must be secured by program.
Each standby -> PLL operation (PLL=ON)
Each standby -> X1 main clock-through (PLL = ON)

3. If the watchdog timer overflows (reset) while the oscillation stabilization time is being counted, the CPU starts clock operation with the ring oscillator.

Remark For PLLS and OSTS, refer to **CHAPTER 6 CLOCK GENERATION FUNCTION** and **CHAPTER 19.8(3) Oscillation stabilization time selection function**.

Figure 19-2. Standby Transition from PLL Operation (PLL = ON)

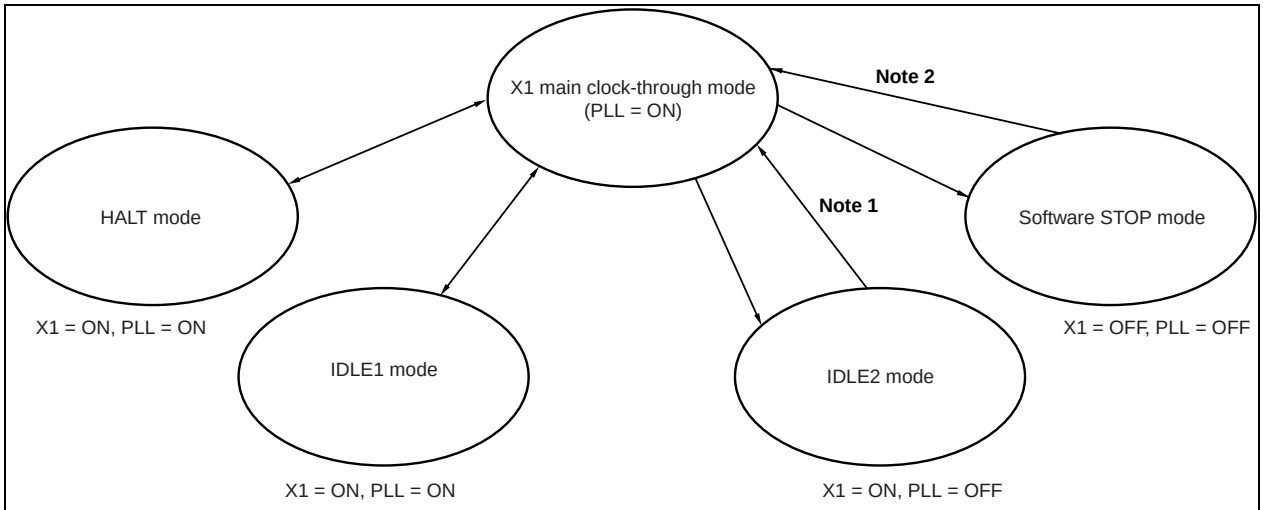


Notes 1. After the lapse of the time set by the OSTS register, the CPU returns to the PLL mode.

2. After the lapse of the time set by the OSTS register, the CPU returns to the PLL mode. If the watchdog timer overflows (reset) while the oscillation stabilization time is being counted, the CPU starts clock operation with the ring oscillator.

Remark For OSTS, refer to **CHAPTER 19.8(3) Oscillation stabilization time selection function**.

Figure 19-3. Standby Transition from X1 Main Clock-Through Operation (PLL = ON)

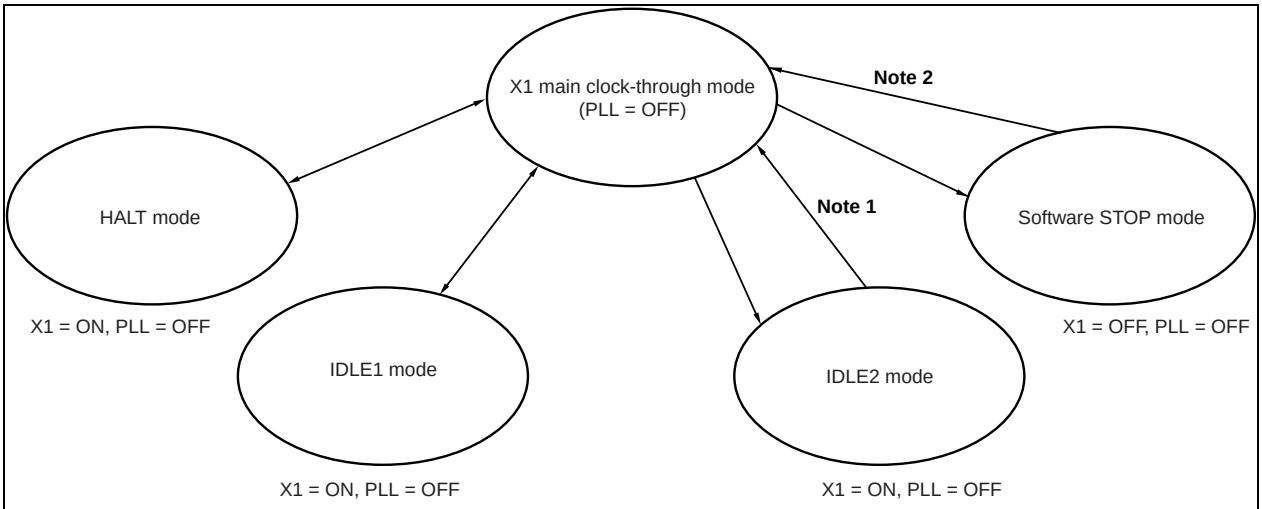


Notes 1. After the lapse of the time set by the OSTS register, the CPU returns to the through mode.

2. After the lapse of the time set by the OSTS register, the CPU returns to the through mode. If the watchdog timer overflows (reset) while the oscillation stabilization time is counted, the CPU starts its clock operation with the ring oscillator.

Remark For OSTS, refer to **CHAPTER 19.8(3) Oscillation stabilization time selection function**.

Figure 19-4. Standby Transition from X1 Main Clock-Through Operation (PLL = OFF)

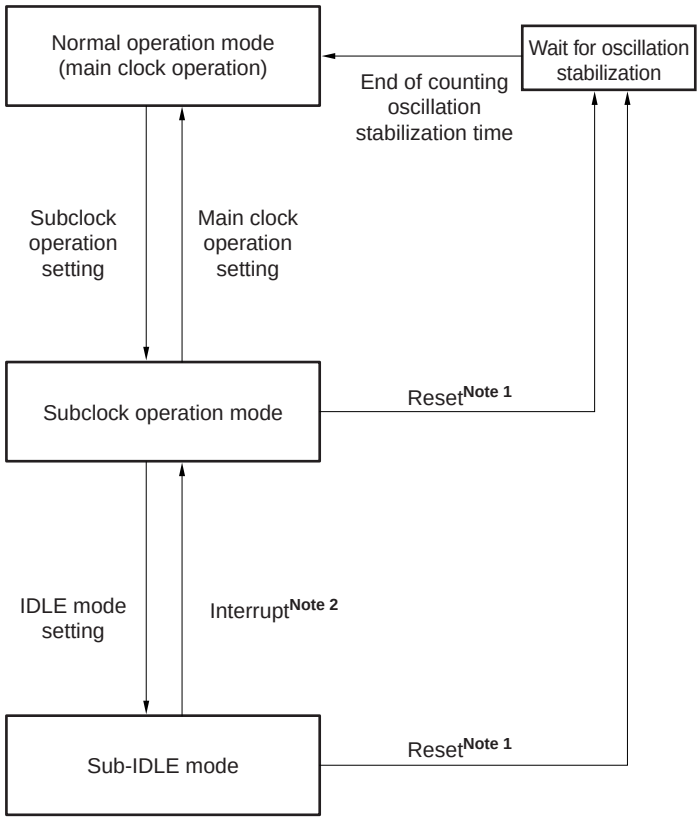


Notes 1. After the lapse of the time set by the OSTS register, the CPU returns to the through mode.

2. After the lapse of the time set by the OSTS register, the CPU returns to the through mode. If the watchdog timer overflows (reset) while the oscillation stabilization time is counted, the CPU starts its clock operation with the ring oscillator.

Remark For OSTS, refer to **CHAPTER 19.8(3) Oscillation stabilization time selection function**.

Figure 19-5. Status Transition Diagram (During Subclock Operation)



Notes 1. Reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), and clock monitor (CLM)

2. Non-maskable interrupt request signal (NMI or INTWDT2), unmasked external interrupt request signal, or internal maskable interrupt request signal that can operate in the sub-IDLE mode and is not masked

19.2 HALT Mode

19.2.1 Setting and operation status

When a dedicated instruction (HALT instruction) is executed in the normal operation mode, the HALT mode is set.

In this mode, the clock oscillator continues operating, but 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 contents of the internal RAM before the HALT mode was set are retained. However, the on-chip peripheral functions that are not dependent upon the instruction processing of the CPU continue operating.

Table 19-3 shows the operation status in the HALT mode.

The HALT mode can reduce the average current consumption of the system if it is used with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an interrupt request signal is held pending, the HALT mode is set but is released immediately by the pending interrupt request.

19.2.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal, unmasked internal interrupt request of a peripheral function that can operate in the HALT mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM)).

When the HALT mode has been released, the normal operation mode is restored.

(1) Non-maskable interrupt request signal and unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the HALT mode is set in an interrupt routine, however, the operation is performed as follows.

(a) If an interrupt request signal having a priority lower than that of the interrupt request currently being serviced is generated, the HALT mode is released, but the interrupt request with the lower priority is not acknowledged. The interrupt request signal itself is held.

(b) If an interrupt request signal (including a non-maskable interrupt request signal) having a priority higher than that of the interrupt request currently being serviced is generated, the HALT mode is released, and this interrupt request signal is acknowledged.

Table 19-2. Operation After HALT Mode Is Released by Interrupt Request Signal

Releasing Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address, or the next instruction is executed.	The next instruction is executed.

(2) Releasing by reset input

The operation is the same as the normal reset operation.

Table 19-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status	
		Without Subclock	With Subclock
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Operable	
Interrupt controller		Operable	
Timer P (TMP0 to TMP3)		Operable	
Timer Q (TMQ0 to TMQ3)		Operable	
Timer M (TMM0)		Operable when other than f_{XT} is selected as the count clock	Operable
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0 to CSIB2	Operable	
	UARTA0 to UARTA3	Operable	
CAN controller		Operable	
A/D converter		Operable	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Holds status before HALT mode is 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 HALT mode was set.	

19.3 IDLE1 Mode

19.3.1 Setting and operation status

The IDLE1 mode is set when the PSM1 and PSM0 bits of the PSMR register are cleared to “00” and the STP bit of the PSC register is set to 1 in the normal operation mode.

In the IDLE1 mode, the clock oscillator, PLL, and flash memory continue operating, but clock supply to the CPU and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the IDLE1 mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate on the subclock or external clock continue operating.

Table 19-5 shows the operation status in the IDLE1 mode.

The IDLE1 mode can reduce current consumption more than the HALT mode because the operations of the on-chip peripheral functions are stopped. Because the main clock oscillator is not stopped, however, the normal mode can be restored without having to secure oscillation stabilization time, in the same manner as in the HALT mode, when the IDLE1 mode is released.

Caution Insert five or more NOP instructions after the store instruction that manipulates the PSC register to set the IDLE2 mode.

19.3.2 Releasing IDLE1 mode

The IDLE1 mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal, unmasked internal interrupt request signal of a peripheral function that can operate in the IDLE1 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM)).

When the IDLE1 mode has been released, the normal operation mode is restored.

Cautions 1. Interrupt request signals that are set (disabled) by the NMI1M, NMI0M, and INTM bits of the PSC register are invalid and do not release the IDLE1 mode.

2. When digital noise elimination is selected by setting of NFC register, and the sampling clock can be selected from among f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, IDLE1 mode can not released using INTP3 pin. For detail, refer to 17.3.10 (13) Noise elimination control register.

(1) Non-maskable interrupt request signal and unmasked maskable interrupt request signal

The IDLE1 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE1 mode is set in an interrupt routine, however, the operation is performed as follows.

(a) If an interrupt request signal having a priority lower than that of the interrupt request currently being serviced is generated, the IDLE1 mode is released, but the interrupt request with the lower priority is not acknowledged. The interrupt request signal itself is held.

(b) If an interrupt request signal (including a non-maskable interrupt request signal) having a priority higher than that of the interrupt request currently being serviced is generated, the IDLE1 mode is released, and this interrupt request signal is acknowledged.

Table 19-4. Operation After IDLE1 Mode Is Released by Interrupt Request Signal

Releasing Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address, or the next instruction is executed.	The next instruction is executed.

(2) Releasing by reset input

The operation is the same as the normal reset operation.

Table 19-5. Operation Status in IDLE1 Mode

Item \ Setting of IDLE1 Mode		Operation Status	
		Without Subclock	With Subclock
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (however, can be used to release standby mode).	
Timer P (TMP0 to TMP3)		Stops operation	
Timer Q (TMQ0 to TMQ3)		Stops operation	
Timer M (TMM0)		Operable when f_R/8 is selected as the count clock	Operable when fR/8 or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0 to CSIB2	Operable when $\overline{SCKBn}$ input clock is selected as the operating clock (n = 0 to 2)	
	UART0-UART3	Stop operation (However, operable when ASCKA0 input clock is selected as the operating clock)	
CAN controller		Stops operation	
A/D converter		Stops operation[Note]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Holds status before IDLE1 mode is 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 IDLE1 mode was set.	

Note To realize low power consumption, stop the A/D converter before shifting to the IDLE1 mode.

19.4 IDLE2 Mode

19.4.1 Setting and operation status

The IDLE2 mode is set when the PSM1 and PSM0 bits of the PSMR register are set to “10” and the STP bit of the PSC register is set to 1 in the normal operation mode.

In the IDLE2 mode, the clock oscillator continues operating, but clock supply to the CPU, PLL, flash memory, and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the IDLE2 mode was set are retained. Not only the CPU but also the other on-chop peripheral functions stop operating. However, the on-chip peripheral functions that can operate on the subclock or external clock continue operating.

Table 19-7 shows the operation status in the IDLE2 mode.

The IDLE2 mode can reduce current consumption more than the IDLE1 mode because the operations of the on-chop peripheral functions and flash memory are stopped. Because the PLL and flash memory are stopped, however, setup time for the PLL and flash memory is required after the IDLE2 mode is released.

Caution Insert five or more NOP instructions after the store instruction that manipulates the PSC register to set the IDLE2 mode.

19.4.2 Releasing IDLE2 mode

The IDLE2 mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal, unmasked internal interrupt request of a peripheral function that can operate in the IDLE2 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operation status before the IDLE2 mode was set.

When the IDLE2 mode has been released, the normal operation mode is restored.

Cautions 1. Interrupt request signals that are set (disabled) by the NMI1M, NMI0M, and INTM bits of the PSC register are invalid and do not release the IDLE2 mode.

2. When digital noise elimination is selected by setting of NFC register, and the sampling clock can be selected from among f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, IDLE2 mode can not released using INTP3 pin. For detail, refer to 17.3.10 (13) Noise elimination control register.

(1) Non-maskable interrupt request signal and unmasked maskable interrupt request signal

The IDLE2 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE2 mode is set in an interrupt routine, however, the operation is performed as follows.

(a) If an interrupt request signal having a priority lower than that of the interrupt request currently being serviced is generated, the IDLE2 mode is released, but the interrupt request with the lower priority is not acknowledged. The interrupt request signal itself is held.

(b) If an interrupt request signal (including a non-maskable interrupt request signal) having a priority higher than that of the interrupt request currently being serviced is generated, the IDLE2 mode is released, and this interrupt request signal is acknowledged.

Table 19-6. Operation After IDLE2 Mode Is Released by Interrupt Request Signal

Releasing Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after the specified setup time is secured.	
Maskable interrupt request signal	Execution branches to the handler address, or the next instruction is executed after the specified setup time is secured.	The next instruction is executed after the specified setup time is secured.

(2) Releasing by reset input

The operation is the same as the normal reset operation.

Table 19-7. Operation Status in IDLE2 Mode

Setting of IDLE2 Mode / Item		Operation Status	
		Without Subclock	With Subclock
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (however, can be used to release standby mode).	
Timer P (TMP0 to TMP3)		Stops operation	
Timer Q (TMQ0 to TMQ3)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0 to CSIB2	Operable when $\overline{SCKBn}$ input clock is selected as the operating clock (n = 0 to 2)	
	UART0-UART3	Stop operation (However, operable when ASCKA0 input clock is selected as the operating clock)	
CAN controller		Stops operation	
A/D converter		Stops operation[Note]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Holds status before IDLE2 mode is 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 IDLE2 mode was set.	

Note To realize low power consumption, stop the A/D converter before shifting to the IDLE2 mode.

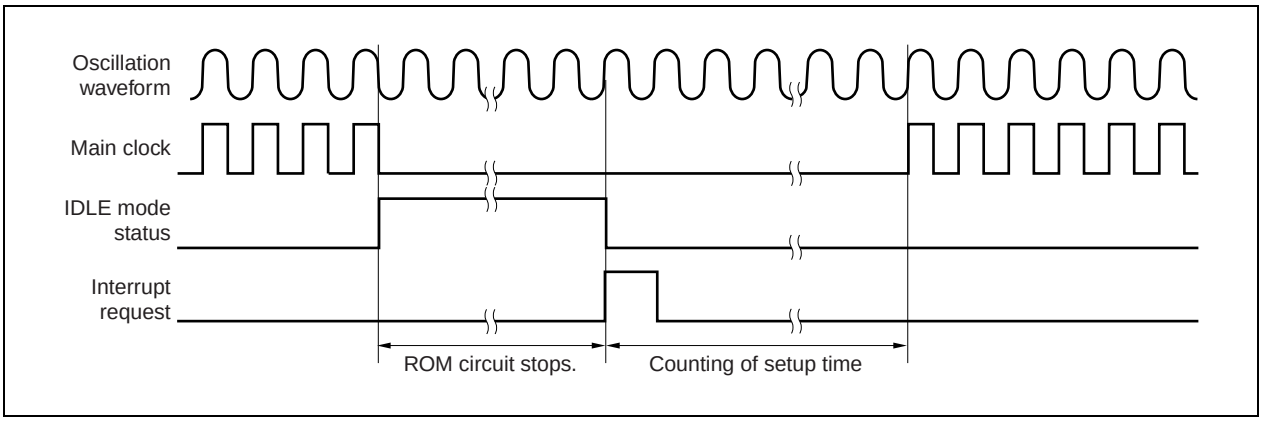
19.4.3 Securing setup time after release of IDLE2 mode

The main clock oscillator stops operating when the IDLE2 mode is set. Therefore, secure the setup time of ROM (flash memory) after releasing the IDLE2 mode.

(1) Releasing by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The setup time is secured by setting the OSTS register.

When a source that releases the IDLE2 mode occurs, an internal dedicated timer starts counting in accordance with the setting of the OSTS register. When this counter overflows, the normal operation mode is restored.



(2) Releasing by reset input ($\overline{\text{RESET}}$ pin input or WDT2RES occurrence)

The operation is the same as the normal reset operation.

The oscillation stabilization time is the default value of the OSTS register, $2^{16}/f_X$.

19.5 Software STOP Mode

19.5.1 Setting and operation status

The software STOP mode is set when the PSM1 and PSM0 bits of the PSMR register are set to “01” or “11”, and the STP bit of the PSC register is set to 1 in the normal operation mode.

In the software STOP mode, the subclock oscillator continues operating, but the main clock oscillator stops operating. Moreover, clock supply to the CPU and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the software STOP mode was set are retained. Not only the CPU but also the other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate on the subclock or external clock continue operating.

Table 19-9 shows the operation status in the software STOP mode.

The software STOP mode can reduce current consumption more than the IDLE2 mode because the operation of the main clock oscillator is stopped. When the subclock oscillator, Ring-OSC, and external clock are not used, the current consumption can be substantially reduced with only a leakage current flowing.

Caution Insert five or more NOP instructions after the store instruction that manipulates the PSC register to set the software STOP mode.

19.5.2 Releasing software STOP mode

The software STOP mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal, unmasked internal interrupt request signal of a peripheral function that can operate in the software STOP mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, or low voltage detector (LVI)).

When the software STOP mode has been released, the normal operation mode is restored.

Cautions 1. Interrupt request signals that are set (disabled) by the NMI1M, NMI0M, and INTM bits of the PSC register are invalid and do not release the software STOP mode.

2. When digital noise elimination is selected by setting of NFC register, and the sampling clock can be selected from among f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, Software STOP mode can not released using INTP3 pin. For detail, refer to 17.3.10 (13) Noise elimination control register.

(1) Non-maskable interrupt request signal and unmasked maskable interrupt request signal

The software STOP mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the software STOP mode is set in an interrupt routine, however, the operation is performed as follows.

(a) If an interrupt request signal having a priority lower than that of the interrupt request currently being serviced is generated, the software STOP mode is released, but the interrupt request with the lower priority is not acknowledged. The interrupt request signal itself is held.

(b) If an interrupt request signal (including a non-maskable interrupt request signal) having a priority higher than that of the interrupt request currently being serviced is generated, the software STOP mode is released, and this interrupt request signal is acknowledged.

Table 19-8. Operation After Software STOP Mode Is Released by Interrupt Request Signal

Releasing Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after the oscillation stabilization time is secured.	
Maskable interrupt request signal	Execution branches to the handler address, or the next instruction is executed the after the oscillation stabilization time is secured.	The next instruction is executed after the oscillation stabilization time is secured.

(2) Releasing by reset input

The operation is the same as the normal reset operation.

Table 19-9. Operation Status in Software STOP Mode

Item \ Setting of Software STOP Mode		Operation Status	
		Without Subclock	With Subclock
Main clock oscillator		Stops oscillation	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation	
Timer P (TMP0 to TMP3)		Stops operation	
Timer Q (TMQ0 to TMQ3)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Stops operation	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when f_R is selected as the count clock	
Serial interface	CSIB0 to CSIB2	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the operating clock (n = 0 to 2)	
	UART0-UART3	Stop operation (However, operable when ASCKA0 input clock is selected as the operating clock)	
CAN controller		Stops operation	
A/D converter		Stops operation Note	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Holds status before software STOP mode is 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 software STOP mode was set.	

Notes: 1. If the STOP mode is set while the A/D converter is operating, the A/D converter is automatically stopped and it starts operating again after the STOP mode is released. However, in that case, the A/D conversion results up to the second conversion after the STOP mode is released are invalid (the third or later conversion results are valid). All the A/D conversion results before the STOP mode is set are invalid.

2. Even if the STOP mode is set while the A/D converter is operating, the power consumption is reduced equivalently to when the A/D converter is stopped before the STOP mode is set.

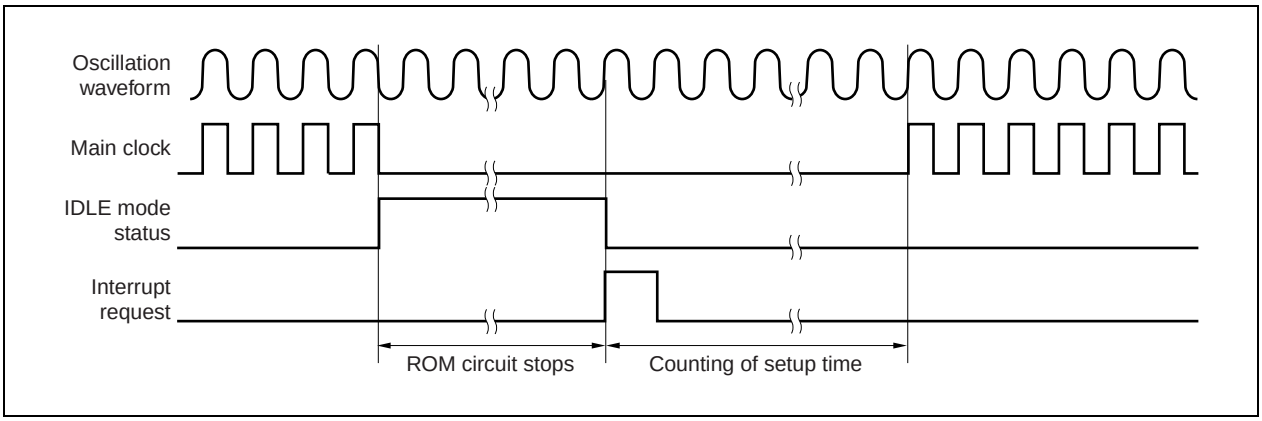
19.5.3 Securing setup time after release of software STOP mode

The main clock oscillator stops operating when the software STOP mode is set. Therefore, secure the oscillation stabilization time of the main clock oscillator after releasing the software STOP mode.

(1) Releasing by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The oscillation stabilization time is secured by setting the OSTS register.

When a source that releases the software STOP mode occurs, an internal dedicated timer starts counting in accordance with the setting of the OSTS register. When this counter overflows, the normal operation mode is restored.



(2) Releasing by reset input

The operation is the same as the normal reset operation.

The oscillation stabilization time is the default value of the OSTS register, $2^{16}/f_X$.

19.6 Subclock Operation Mode

19.6.1 Setting and operation status

The subclock operation mode is set when the CK3 bit of the PCC register is set to 1 in the normal operation mode.

When the subclock operation mode is set, the internal system clock is changed from the main clock to the subclock. Check that the system clock has been changed by using the CLS bit of the PCC register.

When the MCK bit of the PCC register is set to 1, the operation of the main clock oscillator is stopped. Consequently, the entire system operates on only the subclock.

In the subclock operation mode, the subclock is used as the internal system clock, so that the current consumption can be reduced from that in the normal operation mode. In addition, a current consumption close to that in the software STOP mode can be realized by stopping the operation of the main clock oscillator.

Table 19-10 shows the operation status in the subclock operation mode.

Caution Changing the set value of the CK2 to CK0 bits of the PCC register is prohibited when the CK3 bit is manipulated (0→1 or 1→0) (set the CK3 bit by using a bit manipulation instruction). For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

19.6.2 Releasing subclock operation mode

The subclock operation mode is released by clearing the CK3 bit to 0 or by a reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM)).

When the main clock is stopped (MCK bit = 1), clear the MCK bit to 0, secure the oscillation stabilization time of the main clock by software, and then clear the CK3 bit to 0.

When the subclock operation mode is released, the normal operation mode is restored.

Cautions 1. Changing the set value of the CK2 to CK0 bits of the PCC register is prohibited when the CK3 bit is manipulated (0→1 or 1→0) (set the CK3 bit by using a bit manipulation instruction). For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

2. When digital noise elimination is selected, and the sampling clock can be selected from among f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, subclock operation mode can not released

Table 19-10. Operation Status in Subclock Operation Mode

Item \ Setting of Subclock Operation Mode		Operation Status	
		With Main Clock	Without Main Clock
Subclock oscillator		Oscillation enabled	
Ring-OSC generator		Oscillation enabled	
PLL		Operable	Stops operation[Note]
CPU		Operable	
DMA		Operable	
Interrupt controller		Operable	
Timer P (TMP0 to TMP3)		Operable	Stops operation
Timer Q (TMQ0 to TMQ3)		Operable	Stops operation
Timer M (TMM0)		Operable	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_R is selected as the count clock
Serial interface	CSIB1 to CSIB2	Operable	Operable when $\overline{SCKBn}$ input clock is selected as the operating clock (n = 0 to 2)
	UART0-UART3	Operable	Stop operation (However, operable when ASCKA0 input clock is selected as the operating clock)
CAN controller		Operable	Stops operation
A/D converter		Operable	Stops operation
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Settable	
Internal data		Settable	

Note When stopping the main clock, be sure to stop the PLL (by clearing the PLLON bit of the PLLCTL register to 0).

Caution: When the CPU is operating on the subclock and main clock oscillation is stopped, accessing a register in which a wait occurs is disabled. If a wait is generated, it can be released only by reset (see 3.4.10 (2)).

19.7 Sub-IDLE Mode

19.7.1 Setting and operation status

The sub-IDLE mode is set when the PSM1 and PSM0 bits of the PSMR register are set to “10” and the STP bit of the PSC register is set to 1 in the subclock operation mode.

In the sub-IDLE mode, the clock oscillator continues operating, but clock supply to the CPU, flash memory, and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the sub-IDLE mode was set are retained. Not only the CPU but also the other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate on the subclock or external clock continue operating.

The sub-IDLE mode can reduce current consumption more than the subclock operation mode because the operations of the CPU, flash memory, and other on-chip peripheral functions are stopped.

If the sub-IDLE mode is set after the main clock is stopped, a current consumption close to that in the software STOP mode can be realized.

Table 19-12 shows the operation status in the sub-IDLE mode.

Caution Insert five or more NOP instructions after the store instruction that manipulates the PSC register to set the sub-IDLE mode.

19.7.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal, unmasked internal interrupt request of a peripheral function that can operate in the sub-IDLE mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operation status before the sub-IDLE mode was set.

When the sub-IDLE mode is released by an interrupt request signal, the subclock operation mode is restored. When the sub-IDLE mode is released by reset, the normal operation mode is restored.

Cautions: 1. Interrupt request signals that are set (disabled) by the NMI1M, NMI0M, and INTM bits of the PSC register are invalid and do not release the sub-IDLE mode.

2. When digital noise elimination is selected by setting of NFC register, and the sampling clock can be selected from among f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, sub IDLE mode can not released using INTP3 pin. For detail, refer to 17.3.10 (13) Noise elimination control register.

(1) Non-maskable interrupt request signal and unmasked maskable interrupt request signal

The sub-IDLE mode is released by a non-maskable interrupt signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal.

If the sub-IDLE mode is set in an interrupt routine, however, the operation is performed as follows.

(a) If an interrupt request signal having a priority lower than that of the interrupt request currently being serviced is generated, the sub-IDLE mode is released, but the interrupt request with the lower priority is not acknowledged. The interrupt request signal itself is held.

(b) If an interrupt request signal (including a non-maskable interrupt request signal) having a priority higher than that of the interrupt request currently being serviced is generated, the sub-IDLE mode is released, and this interrupt request signal is acknowledged.

Table 19-11. Operation After Sub-IDLE Mode Is Released by Interrupt Request Signal

Releasing Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address, or the next instruction is executed.	The next instruction is executed.

(2) Releasing by reset input

The operation is the same as the normal reset operation.

Table 19-12. Operation Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode / Item		Operation Status	
		With Main Clock	Without Main Clock
Subclock oscillator		Oscillation enabled	
Ring-OSC generator		Oscillation enabled	
PLL		Operation enabled	Stops operation[Note]
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (however, can be used to release standby mode)	
Timer P (TMP0 to TMP3)		Stops operation	
Timer Q (TMQ0 to TMQ3)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ or f_{XT} is selected as the count clock	
Watch timer		Operation enabled	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when f_R is selected as the count clock	
Serial interface	CSIB0 to CSIB2	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the operating clock (n = 0 to 2)	
	UART0-UART3	Stop operation (However, operable when ASCKA0 input clock is selected as the operating clock)	
CAN controller		Stops operation	
A/D converter		Stops operation	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION** (same operation status as in IDLE1 and IDLE2 modes).	
Port function		Holds status before sub-IDLE mode is 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 sub-IDLE mode was set.	

Note When stopping the main clock, be sure to stop the PLL (by clearing the PLLON bit of the PLLCTL register to 0).

19.8 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 specifies the standby mode. This register is a special register and can be written only in a combination of specific sequences (refer to **3.4.9 Special registers**).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	4	3	2	1	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	Standby mode release control by occurrence of INTWDT2 signal
0	Enable releasing standby mode by INTWDT2 signal.
1	Disable releasing standby mode by INTWDT2 signal.

NMI0M	Standby mode release control by NMI pin input
0	Enable releasing standby mode by NMI pin input.
1	Disable releasing standby mode by NMI pin input.

INTM	Standby mode release control by maskable interrupt request signal
0	Enable releasing standby mode by maskable interrupt request signal.
1	Disable releasing standby mode by maskable interrupt request signal.

STP	Setting of standby mode[Note]
0	Normal mode
1	Standby mode

Note Standby modes that can be set by the STP bit: IDLE1 mode, IDLE2 mode, software STOP mode, and sub-IDLE mode

Caution Before setting the IDLE1, IDLE2, software STOP mode, or sub-IDLE mode, set the PSM1 and PSM0 bits of the PSMR register.

(2) Power save mode register (PSMR)

This is an 8-bit register that controls the operating status of the power save mode and the operation of the clock.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	0
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	Specification of operation in software standby mode
0	0	IDLE1 mode, Sub-IDLE mode
0	1	Software STOP mode, Sub-IDLE mode
1	0	IDLE2 mode, Sub-IDLE mode
1	1	Software STOP mode

Cautions 1. Be sure to clear bits 2 to 7 to 0.

2. The PSM0 and PSM1 bits are valid only when the STP bit = 1.

Remark IDLE1: Mode used to stop all the operations except the oscillator and some circuits (flash memory and PLL). When IDLE1 mode is released, the normal operation mode is restored without the lapse of the oscillation stabilization time, in the same manner as in the HALT mode.

IDLE2: In this mode, all operations except the oscillator operation are stopped. After the IDLE2 mode is released, the normal operation mode is restored following the lapse of the setup time specified by the OSTS register (flash memory and PLL).

Sub-IDLE: Mode used to stop all the operations except the oscillator. After the sub-IDLE mode is released, the setup time (for flash memory and PLL) specified by the OSTS register elapses, and then the normal operation mode is restored.

STOP: Mode used to stop all the operations except the subclock oscillator. After the software STOP mode is released, the oscillation stabilization time specified by the OSTS register elapses, and then the normal operation mode is restored.

(3) 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.

Reset input sets this register to 06H.

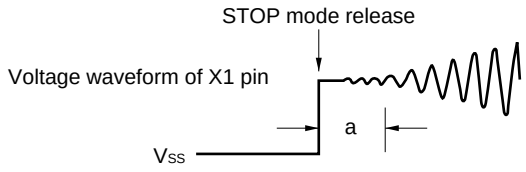
After reset: 06H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time[Note]
0	0	0	$2^{10}/f_X$
0	0	1	$2^{11}/f_X$
0	1	0	$2^{12}/f_X$
0	1	1	$2^{13}/f_X$
1	0	0	$2^{14}/f_X$
1	0	1	$2^{15}/f_X$
1	1	0	$2^{16}/f_X$
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 $\overline{\text{RESET}}$ 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 $2^{16}/f_X$ (because the initial value of the OSTS register = 06H).

Remark f_X = Oscillation frequency

CHAPTER 20 RESET FUNCTION

20.1 Overview

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

The reset function is outlined below.

(1) Five types of reset sources

- Reset function by $\overline{\text{RESET}}$ pin input
- Reset function by overflow of watchdog timer 2 (WDT2RES)
- System reset by low voltage detector (LVI) (see **CHAPTER 23 LOW VOLTAGE DETECTOR**)
- System reset by clock monitor (CLM) (see **CHAPTER 21 CLOCK MONITOR**)
- System reset by power-on clear circuit (POC) (see **CHAPTER 22 POWER-ON CLEAR CIRCUIT**)

It can be check reset source by reset source flag register (RESF) after reset has been released.

(2) Emergency operation mode

If the WDT2 overflows during the main clock oscillation stabilization time inserted after reset, a main clock oscillation anomaly is judged and the CPU starts operating on the internal oscillation clock.

Caution: When the CPU is being operated via the internal oscillator, 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 3.4.10 (2) Accessing specific on-chip peripheral I/O registers.

20.2 Register to Check Reset Source

(1) Reset source flag register (RESF)

The RESF register is a special register that can be written only by a combination of specific sequences (see **3.4.9 Special registers**).

The RESF register indicates the source from which a reset signal is generated.

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

$\overline{\text{RESET}}$ pin input clears this register to 00H. The default value differs if the source of reset is other than the $\overline{\text{RESET}}$ pin signal.

After reset: 00H[Note] R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	Generation of reset signal from WDT2
0	Not generated
1	Generated

CLMRF	Generation of reset signal from clock monitor
0	Not generated
1	Generated

LVIRF	Generation of reset signal from low voltage detector
0	Not generated
1	Generated

Note This register holds 00H after a reset by the $\overline{\text{RESET}}$ pin, or sets its reset flags (WDT2RF, CLMRF, and LVIRF bits) after a reset by the WDT2RES signal, low voltage detector (LVI), or clock monitor (CLM) (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.

20.3 Operation

20.3.1 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 is initialized.

When the level of the $\overline{\text{RESET}}$ pin input is changed from low to high, the reset status is released.

If the reset status is released by input to the $\overline{\text{RESET}}$ pin, the oscillation stabilization time (reset value of the OSTS register: $2^{16}/f_X$) elapses, and then the CPU starts program execution.

Table 20-1. Hardware Status When Signal Is Input to $\overline{\text{RESET}}$ Pin

Item	During Reset	After Reset
Main clock oscillator (f_X)	Stops oscillation	Starts oscillation
Subclock oscillator (f_{XT}): X'tal RC	 X'tal ->Continues oscillation RC ->Stops oscillation	 X'tal ->Continues oscillation RC ->Starts oscillation
Ring-OSC generator	Stops oscillation	Starts oscillation
Peripheral clock (f_X to f_X/1,024)	Stops operation	Starts operation after oscillation stabilization time
Internal system clock (f_{XX}), CPU clock (f_{CPU})	Stops operation	Starts operation after oscillation stabilization time (initialized to f_{XX}/8)
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 (port/alternate-function pins)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status. OCDM register is reset (01H).	
Other on-chip peripheral functions	Stop operation	Start operation after oscillation stabilization time

Note Because the V850ES/FX2 supports a boot swap function, the firmware uses part of the internal RAM after the internal system reset is released. For details see **20.4 RAM usage after RESET release.**

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, refer to CHAPTER 4 PORT FUNCTIONS.

Figure 20-1. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

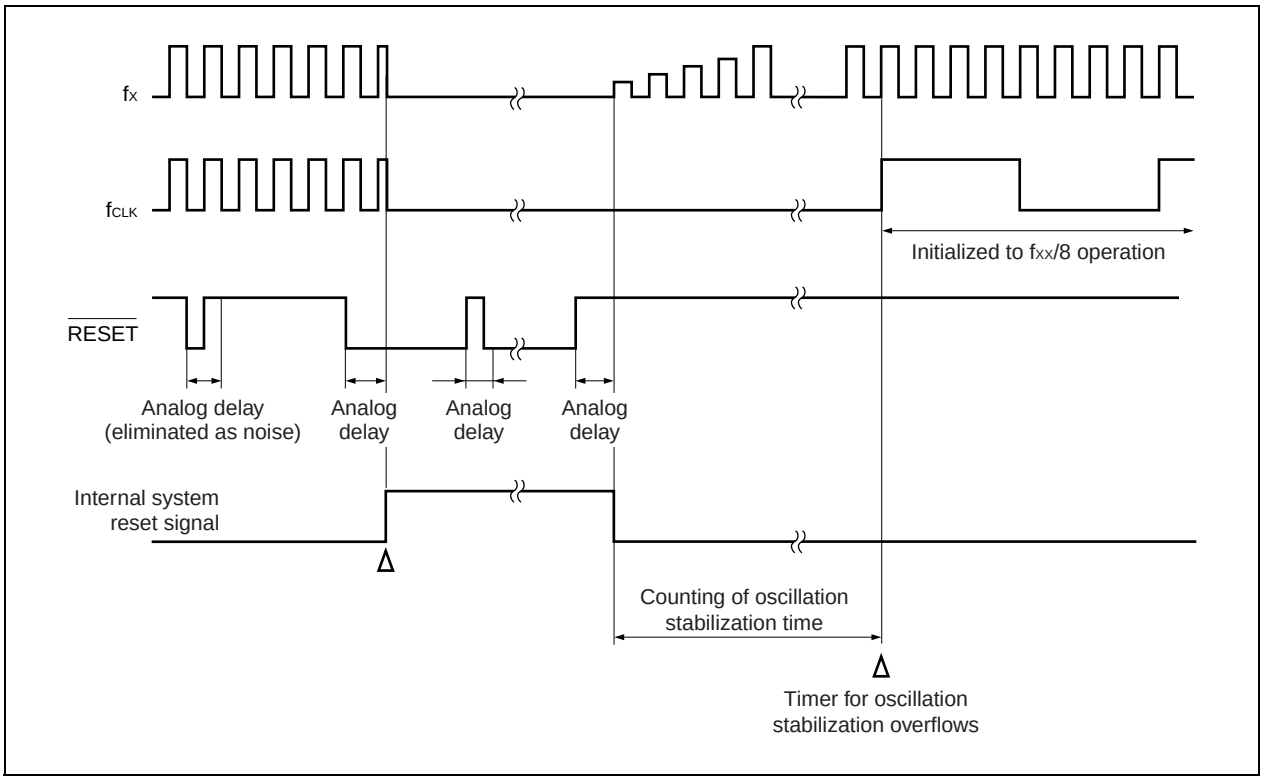
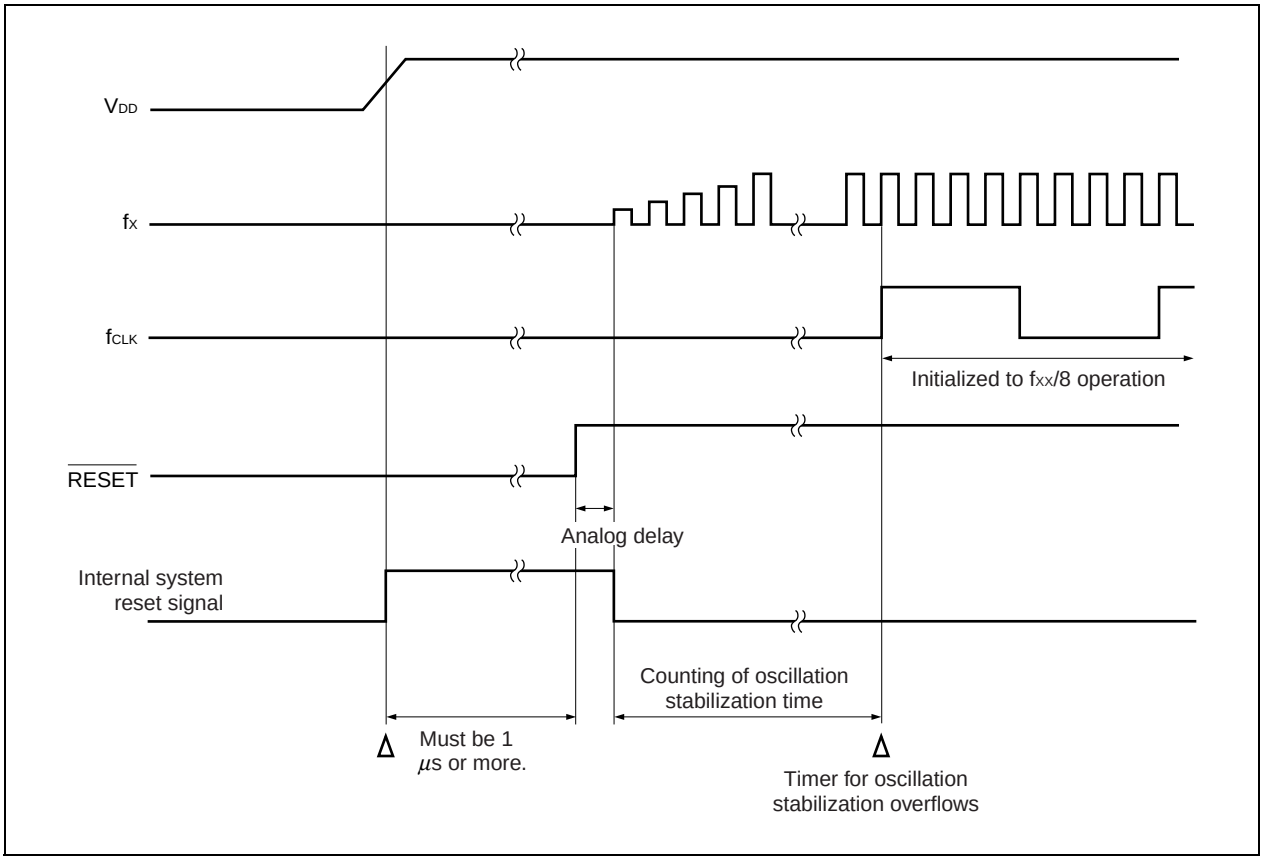


Figure 20-2. Timing of Power-on Reset Operation



20.3.2 Reset operation by WDT2RES signal

If the mode in which a reset operation is performed when watchdog timer 2 overflows is set, if watchdog timer 2 overflows (generating the WDT2RES signal), the system is reset and each hardware is initialized to the specified status.

After watchdog timer 2 overflows, the reset status lasts for a specific time (analog delay). Then the reset status is automatically released. After the reset status is released, the oscillation stabilization time of the main clock oscillator elapses (default value of OSTS register: $2^{16}/f_X$), and the CPU starts program execution.

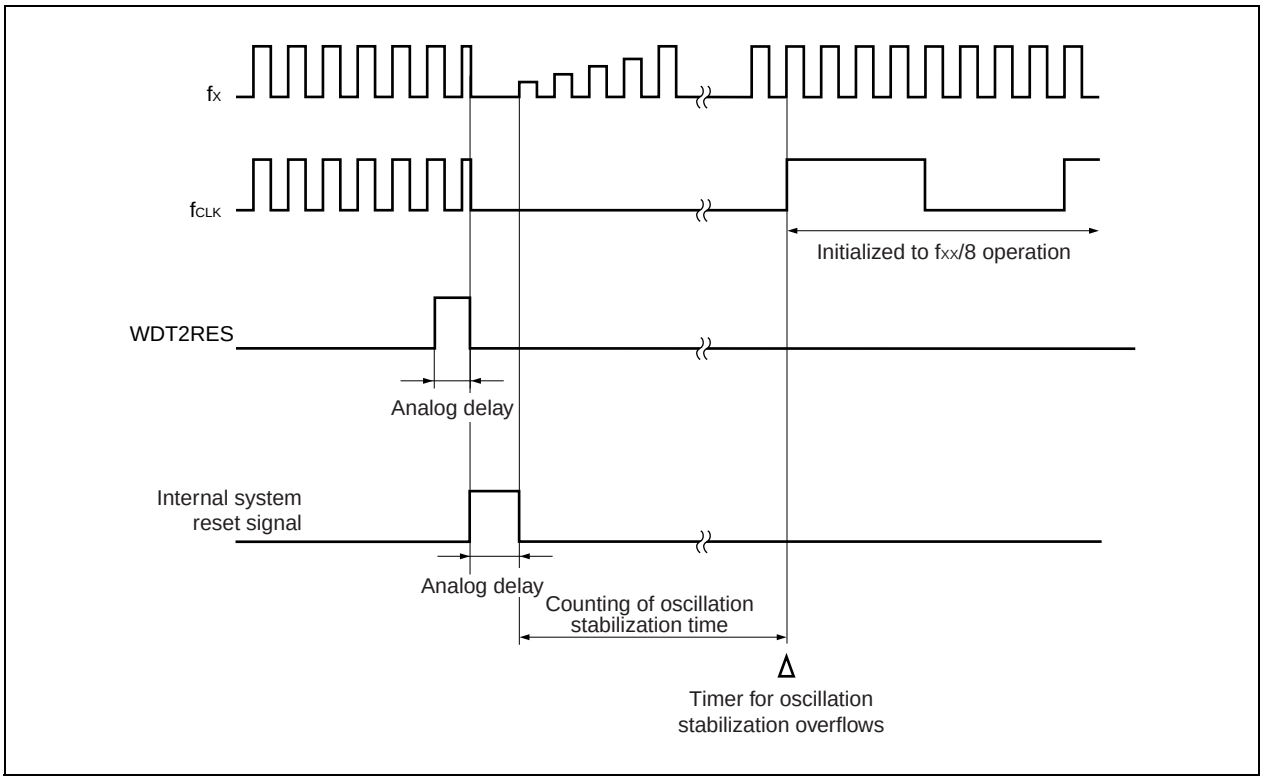
The main clock oscillator is stopped during the reset period.

Table 20-2. Hardware Status After Generation of WDT2RES Signal

Item	During Reset	After Reset
Main clock oscillator (f_X)	Stops oscillation	Starts oscillation
Subclock oscillator (f_{XT}): X'tal RC	 X'tal ->Continues oscillation RC ->Stops oscillation	 X'tal ->Continues oscillation RC ->Starts oscillation
Ring-OSC generator	Stops oscillation	Starts oscillation
Peripheral clock (f_X to f_X/1,024)	Stops operation	Starts operation after oscillation stabilization time
Internal system clock (f_{XX}), CPU clock (f_{CPU})	Stops operation	Starts operation after oscillation stabilization time (initialized to f_{XX}/8)
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 (port/alternate-function pins)	High impedance	
On-chip peripheral I/O registers	Initialized to specified value. OCDM register retains its value.	
Other on-chip peripheral functions	Stop operation	Start operation after oscillation stabilization time

Note Because the V850ES/FX2 supports a boot swap function, the firmware uses part of the internal RAM after the internal system reset is released. For details see **20.4 RAM usage after RESET release.**

Figure 20-3. Timing of Reset Operation by Generation of WDT2RES Signal



20.3.3 Reset operation by power-on clear (only on-chip products of the power-on clear function)

If the supply voltage falls below the voltage detected by comparison supply voltage and detection voltage when power-on clear operation is enabled (incl. when power input), a system reset is executed, and the hardware is initialized to the initial status.

The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the detection voltage. Then, reset status is released automatically. After reset release, the oscillation stabilization time of main clock oscillator (default value of OSTS register: 2^16/fx) is ensured, and CPU is started program execution. For details, refer to **CHAPTER 22 POWER-ON CLEAR**.

20.3.4 Reset operation by low-voltage detector

If the supply voltage falls below the voltage detected by the low-voltage detector when LVI operation is enabled, a system reset is executed (when the LVIM.LVIMD bit is set to 1), and the hardware is initialized to the initial status.

The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the LVI detection voltage.

After reset release, the oscillation stabilization time of main clock oscillator (default value of OSTS register: 2^16/fx) is ensured, and CPU is started program execution. For details, refer to CHAPTER 23 LOW-VOLTAGE DETECTOR.

20.3.5 Reset operation by clock monitor

When operation of the clock monitor is enabled, the main clock is monitored by using the internal oscillator. Then, when oscillation stop of the main clock is detected, system reset is executed and each hardware is initialized to the initial status. For details, refer to **CHAPTER 21 CLOCK MONITOR**.

20.4 RAM usage after RESET release

Because the V850ES/Fx2 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 of the RAM are not retained even when power-on reset is executed.

The used RAM areas after RESET are for all products the

first 150 bytes at the lower side and

the last 100 bytes at the upper side.

The detailed affected addresses are:

Part number (RAM)	150 BYTE LOWER SIDE		100 BYTE UPPER SIDE	
D70F3231 (6K)	3FFD800	3FFD895	3FFEF9B	3FFEFFF
D70F3232 (12K)	3FFC000	3FFC095	3FFEF9B	3FFEFFF
D70F3233 (12K)	3FFC000	3FFC095	3FFEF9B	3FFEFFF
D70F3234 (6K)	3FFD800	3FFD895	3FFEF9B	3FFEFFF
D70F3235 (12K)	3FFC000	3FFC095	3FFEF9B	3FFEFFF
D70F3236 (16K)	3FFB000	3FFB095	3FFEF9B	3FFEFFF
D70F3237 (16K)	3FFB000	3FFB095	3FFEF9B	3FFEFFF
D70F3238 (20K)	3FFA000	3FFA095	3FFEF9B	3FFEFFF
D70F3239 (20K)	3FFA000	3FFA095	3FFEF9B	3FFEFFF

CHAPTER 21 CLOCK MONITOR

21.1 Function of Clock Monitor

The clock monitor samples the main 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 is stopped (from when the PCC.MCK bit = 1 during subclock operation, until the PCC.CLS bit = 0 during main clock operation)
- When the sampling clock is stopped (Ring-OSC)
- When the CPU operates with Ring-OSC

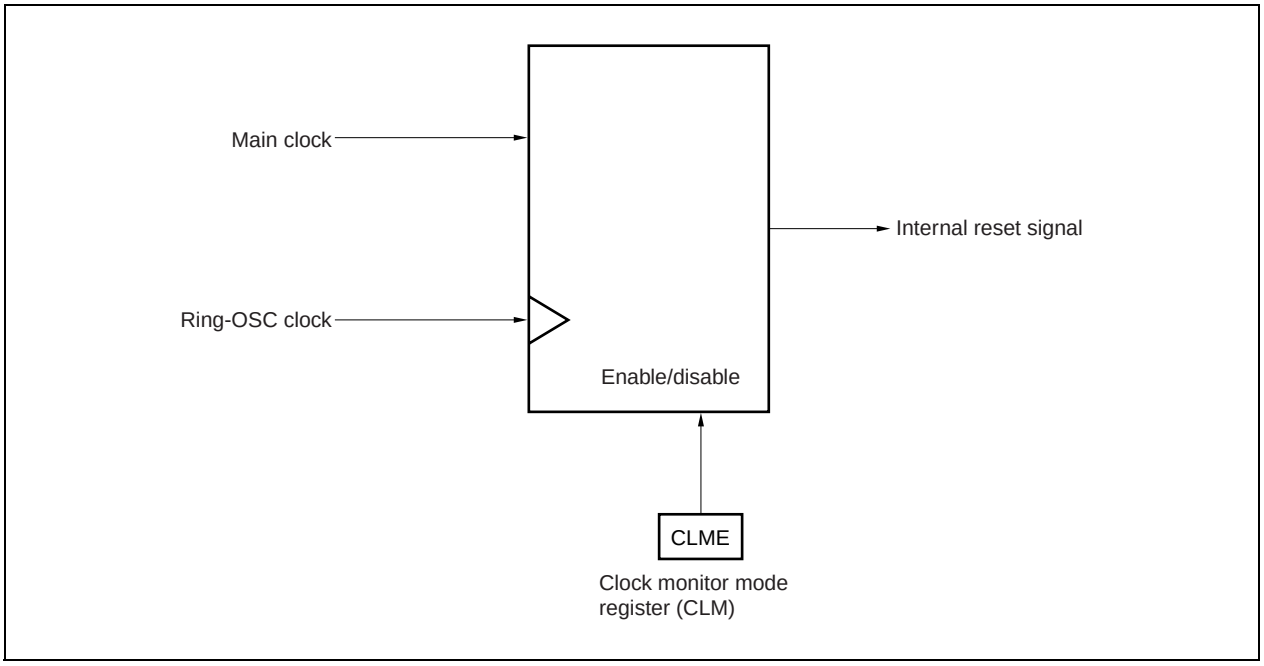
21.2 Configuration of Clock Monitor

The clock monitor consists of the following hardware.

Table 21-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 21-1. CLM Block Diagram



21.3 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 a special register and can be written only in a combination of specific sequences (refer to 3.4.9 Special registers).

This register is used to set the operation mode of the clock monitor.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

After reset: 00H R/W Address: FFFFF870H

	7	6	5	4	3	2	1	0
CLM	0	0	0	0	0	0	0	CLME

CLME	Clock monitor operation enable or disable
0	Disable clock monitor operation.
1	Enable clock monitor operation.

Cautions 1. Once the CLME bit has been set to 1, it cannot be cleared to 0 by any means other than reset.

2. If reset is occurred for clock monitor, CLME bit is clear (0), and RESF, CLMRF bit is set (1).

21.4 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 conditions>

- While oscillation stabilization time is being counted after software STOP mode is released
- When the main clock is stopped (from when the PCC.MCK bit = 1 during subclock operation, until the PCC.CLS bit = 0 during main clock operation)
- When the sampling clock is stopped (Ring-OSC)
- When the CPU operates using Ring-OSC

Table 21-2. Operation Status of Clock Monitor (When CLM.CLME Bit = 1, During Ring-OSC Operation)

CPU Operating 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]
	IDLE1 mode, IDLE2 mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
	Software STOP mode	Stops	Oscillates[Note 1]	Stops
Subclock (MCK bit of PCC register = 0)	Sub-IDLE mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
Subclock (MCK bit of PCC register = 1)	Sub-IDLE mode	Stops	Oscillates[Note 1]	Stops
Ring-OSC clock	–	Stops	Stops[Note 1]	Stops
During reset	–	Stops	Stops	Stops

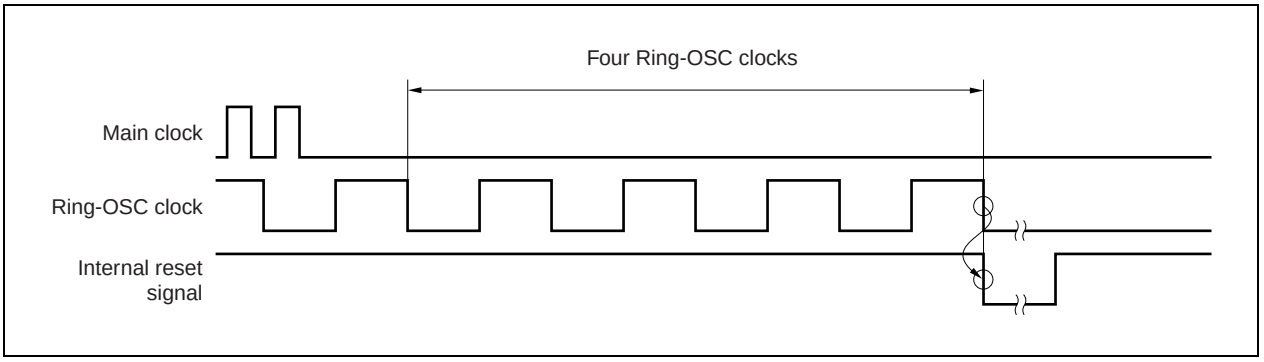
Notes 1. Ring-OSC can be stopped by setting the RSTOP bit of the RCM register to 1 only when “Ring-OSC: Can be stopped” is specified by an option function.

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 21-2.

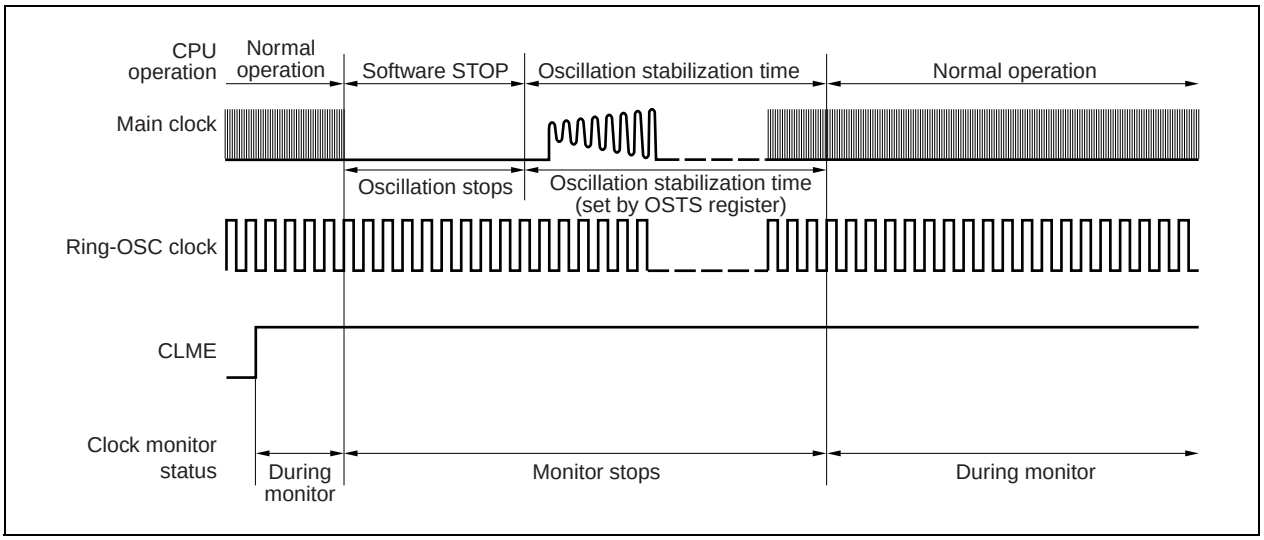
Figure 21-2. When Oscillation of Main Clock Is Stopped



(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.

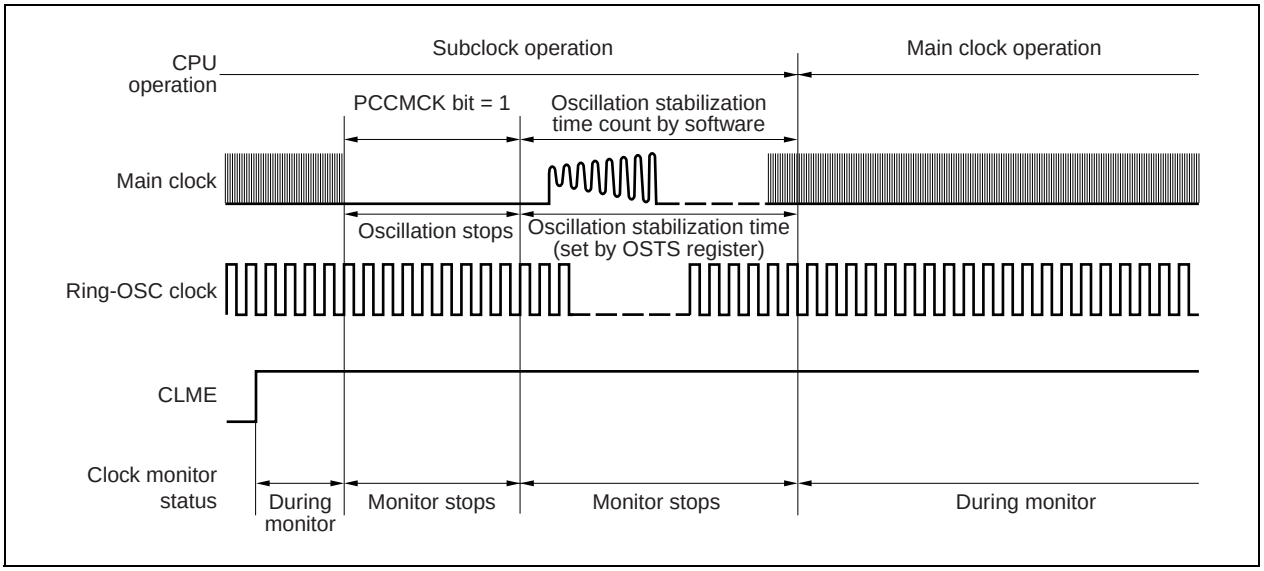
Figure 21-3. Operation in Software STOP Mode or After Software STOP Mode Is Released



(3) Operation when main clock is stopped (arbitrary)

During subclock operation (CLS bit of the PCC register = 1) or when the main clock is stopped by setting the MCK bit of the PCC register to 1, the monitor operation is stopped until the main clock operation is started (CLS bit of PCC register = 0). The monitor operation is automatically started when the main clock operation is started.

Figure 21-4. Operation When Main Clock Is Stopped (Arbitrary)



(4) Operation while CPU is operating on Ring-OSC clock (CCLSF bit of CCLS register = 1)

The monitor operation is not stopped when the CCLSF bit is 1, even if the CLME bit is set to 1.

CHAPTER 22 POWER-ON CLEAR CIRCUIT

22.1 Functions of Power-on Clear Circuit

The power-on clear (POC) circuit has the following functions.

- Generates an internal reset signal upon power application.
- Compares the supply voltage (V_{DD}) and detected voltage (V_{POC0}), and generates an internal reset signal when $V_{DD} < V_{POC0}$.

  The following choice can be made depending on the product.
  - POC is disabled.
  - POC can be used (detected voltage: V_{POC0} = 3.7 V (Typ.))

Caution If the internal reset signal is generated by the POC circuit, the reset source flag register (RESF) is cleared (to 00H).

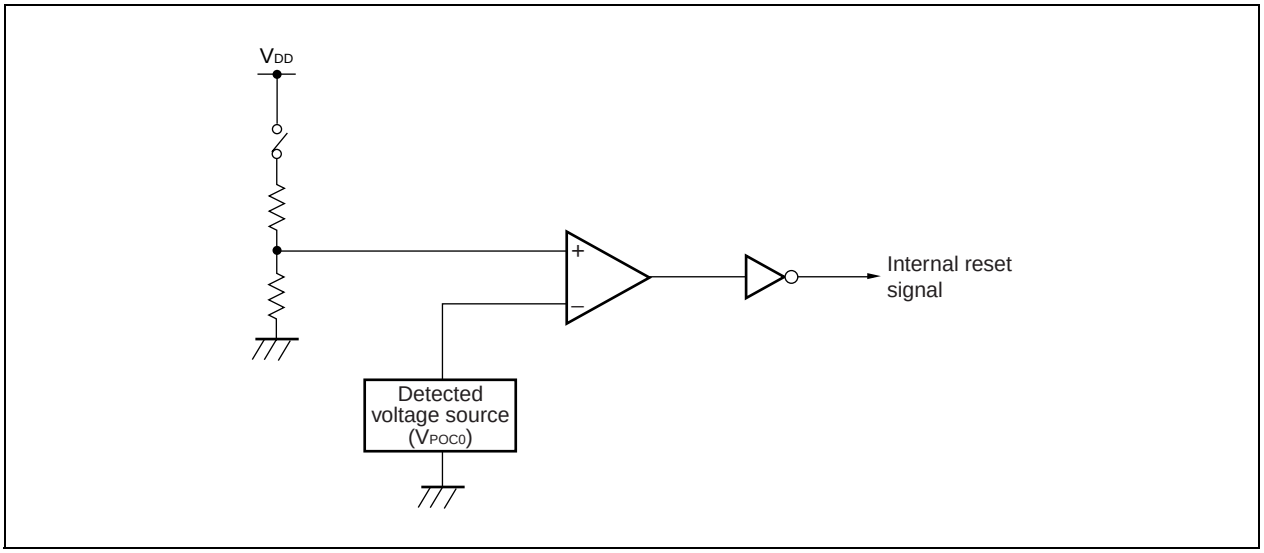
Remarks 1. This product has several hardware units that generate an internal reset signal. When reset is effected by watchdog timer 2 (WDT2RES), the low-voltage detector (LVI), or the clock monitor (CLM), a flag that identifies the reset source is provided in the reset source flag register (RESF). If an internal reset signal is generated by WDT2RES, LVI, or the clock monitor, RESF is not cleared (00H) but the corresponding flag is set (1). For details of RESF, refer to **CHAPTER 20 RESET FUNCTION**.

2. The time when it consumes to the program start from the power supply input is The time from power supply input to reset released + 16 ms in case of the frequency that is connected outside is 5 MHz. But this time is influenced by the outside factor (The condition of power supply that supplies to microcomputer).

22.2 Configuration of Power-on Clear Circuit

Figure 22-1 shows the block diagram of the power-on clear circuit.

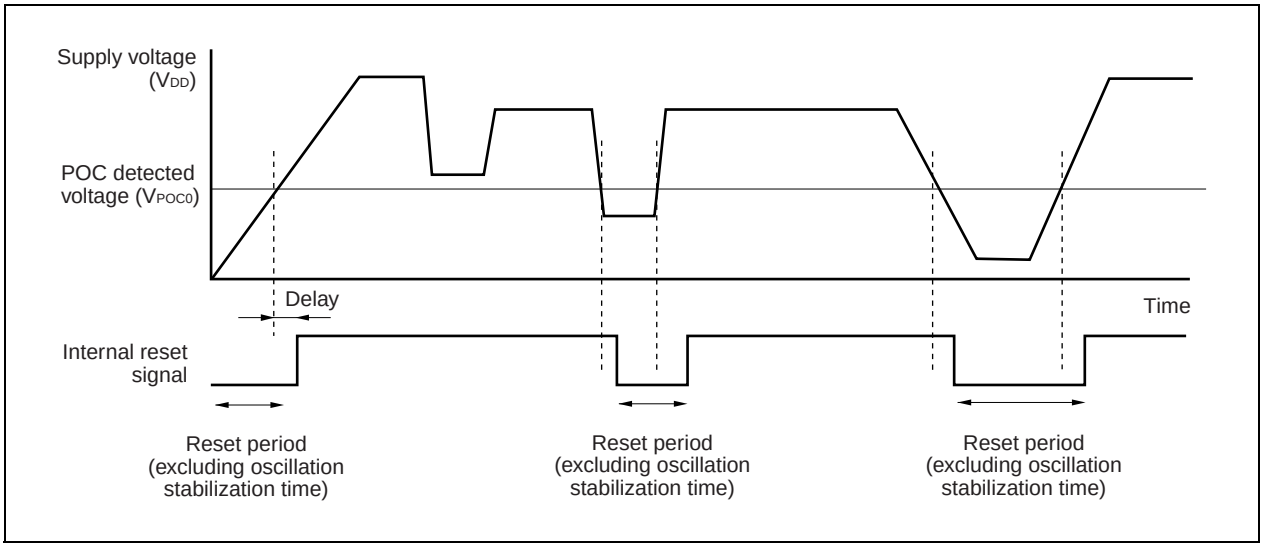
Figure 22-1. Block Diagram of Power-on Clear Circuit



22.3 Operation of Power-on Clear Circuit

The power-on clear circuit compares the supply voltage (V_{DD}) and detected voltage (V_{POC0}), and generates an internal reset signal when $V_{DD} < V_{POC0}$.

Figure 22-2. Timing of Internal Reset Signal Generation by Power-on Clear Circuit



CHAPTER 23 LOW-VOLTAGE DETECTOR

23.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

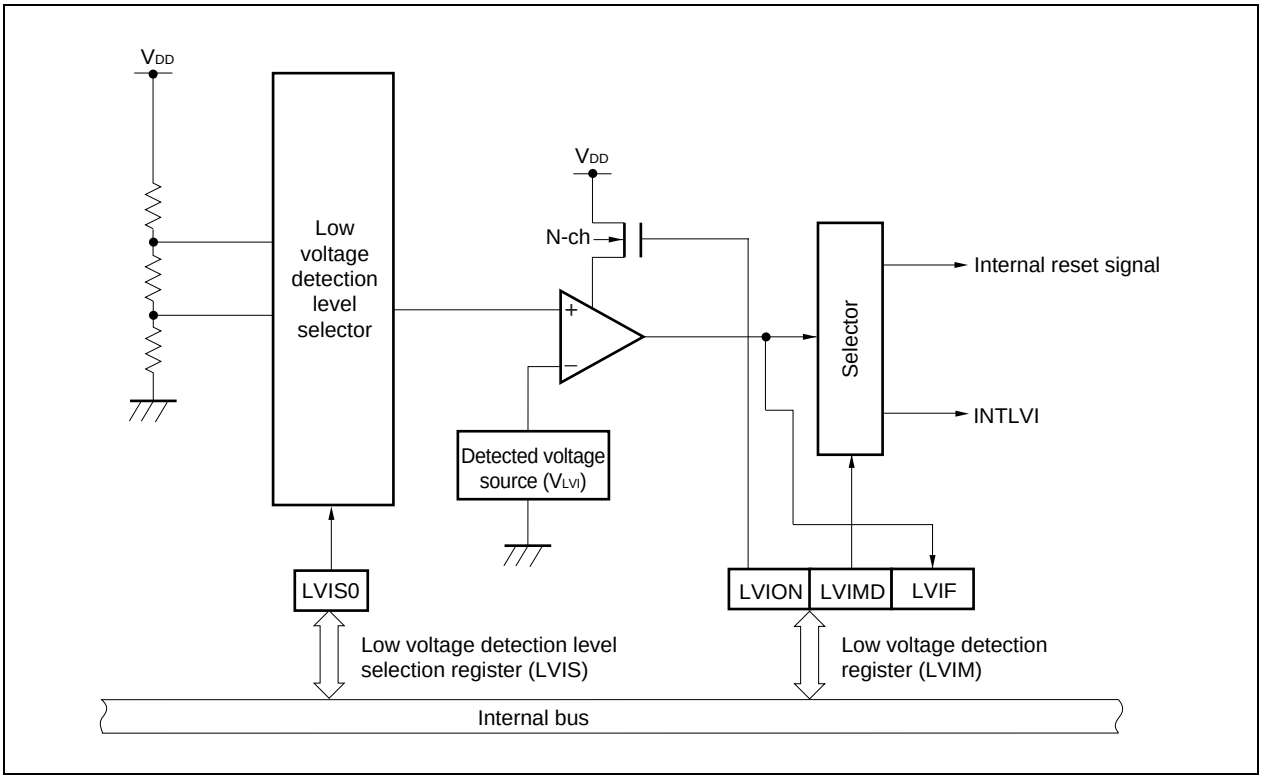
- Compares the supply voltage (V_{DD}) and detected voltage (V_{LVII}) and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- Interrupt or reset signal can be selected by software.
- Can operate in STOP mode too.
- 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. For details of RESF, refer to **CHAPTER 20 RESET FUNCTION**.

23.2 Configuration of Low-Voltage Detector

Figure 23-1 shows the block diagram of the low-voltage detector.

Figure 23-1. Block Diagram of Low-Voltage Detector



23.3 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)

This register is a special register and can be written only in a combination of specific sequences (refer to 3.4.9 Special registers).

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF890H

	7	6	5	4	3	2	1	0
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Low voltage detection operation enable or disable
0	Disable operation.
1	Enable operation.

LVIMD	Selection of operation mode of low voltage detection
0	Generate interrupt request signal INTLVI when supply voltage < detected voltage.
1	Generate internal reset signal LVIRES when supply voltage < detected voltage.

LVIF	Low voltage detection flag
0	When supply voltage > detected voltage, or when operation is disabled
1	Supply voltage of connected power supply < detected voltage

Cautions 1. After setting the LVION bit to 1, wait for 0.2 ms (Max.) before checking the voltage using the LVIF bit.

2. The value of the LVIF flag is output as the output signal INTLVI when the LVION bit = 1 and LVIMD bit = 0.

3. The LVIF bit is read-only.

4. Be sure to clear bits 2 to 6 to 0.

(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.

After reset: 00H R/W Address: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Detection level
0	4.4 V (Typ.)
1	4.2 V (Typ.)

Cautions: 1. This register cannot be written until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

2. Be sure to clear bits 7 to 1 to 0.

(3) Internal RAM data status register (RAMS)

The RAMS register is a flag register that indicates whether the internal RAM is valid or not.

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.

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

After reset: 01H R/W Address: FFFFF892H

	7	6	5	4	3	2	1	0
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	Internal RAM data valid/invalid
0	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.

After reset: 00H R/W Address: FFFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	Pseudo specification of RAM retention voltage detection signal
0	Do not detect voltage lower than RAM retention voltage.
1	Detect voltage lower than RAM retention voltage (set RAMF flag).

Caution This bit is not automatically cleared.

[Usage]

When an in-circuit emulator is used, pseudo emulation of RAMF is realized by rewriting this register on the debugger.

<1> CPU break (CPU operation stops.)

<2> Set the EVARAMIN bit to 1 by using a register write command.
By setting the EVARAMIN bit to 1, the RAMF bit is set to 1 on hardware (the internal RAM data is invalid).

<3> Clear the EVARAMIN bit to 0 by using a register write command again.
Unless this operation is performed (clearing the EVARAMIN bit to 0), the RAMF bit cannot be cleared to 0 by a CPU operation instruction.

<4> Run the CPU and resume emulation.

23.4 Operation of Low-Voltage Detector

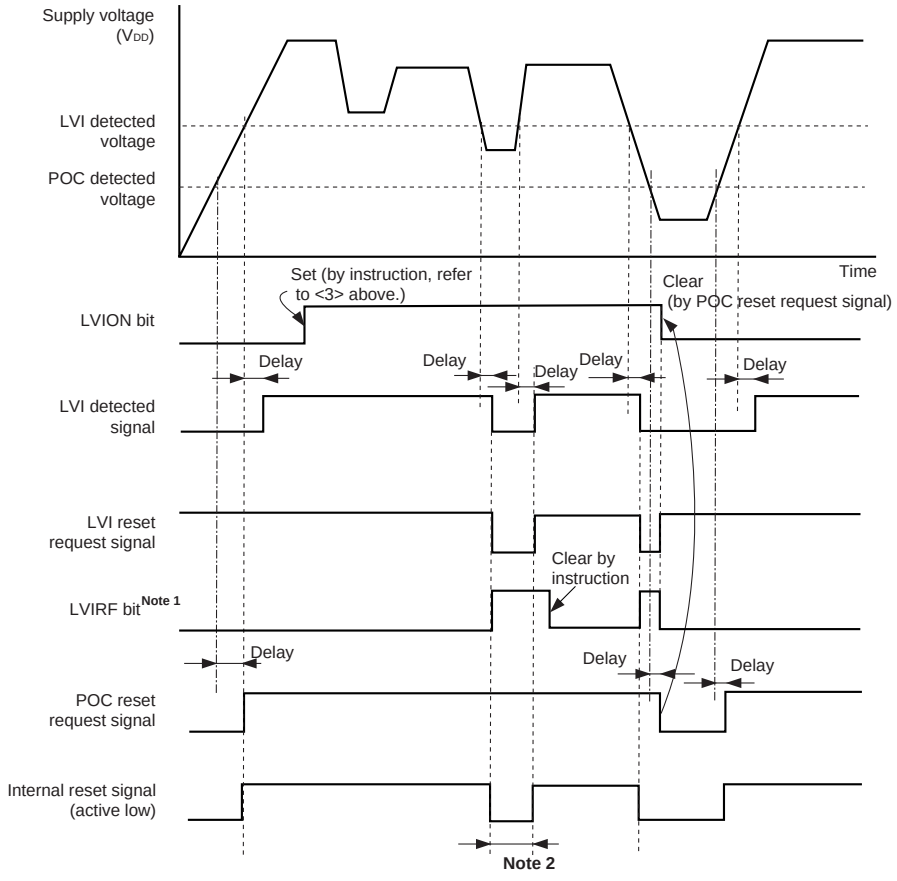
Depending on the setting of the LVIMD bit, an interrupt signal (INTLVI) or an internal reset signal is generated. How to specify each operation is described below, together with timing charts.

23.4.1 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 0.2 ms (Max.) 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 23-2. 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 20 RESET FUNCTION**.

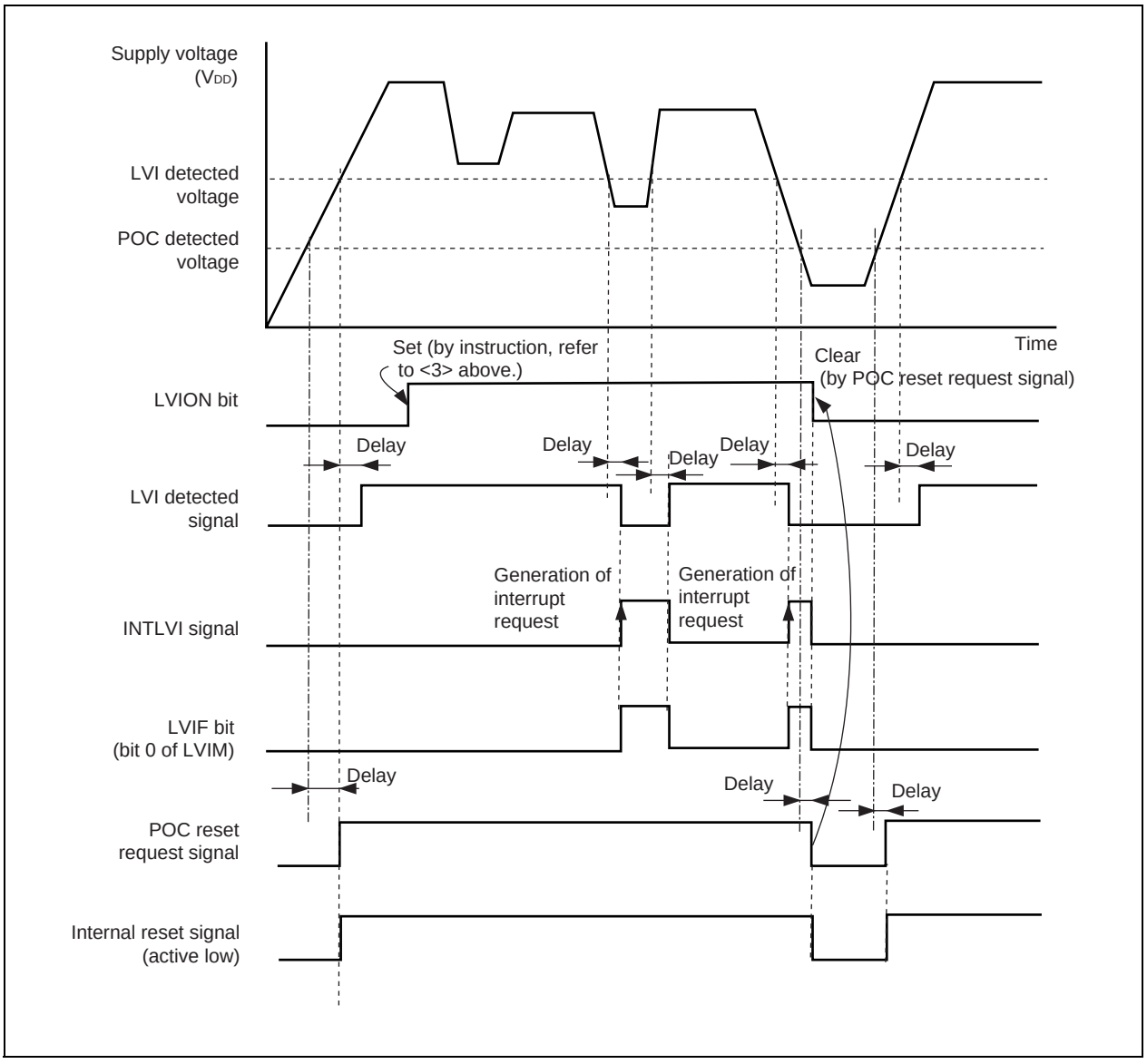
2. During the period in which the supply voltage is the set low voltage or lower, the internal reset signal is retained (internal reset state).

23.4.2 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 0.2 ms (Max.) 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 23-3. Operation Timing of Low-Voltage Detector (LVIM = 0)

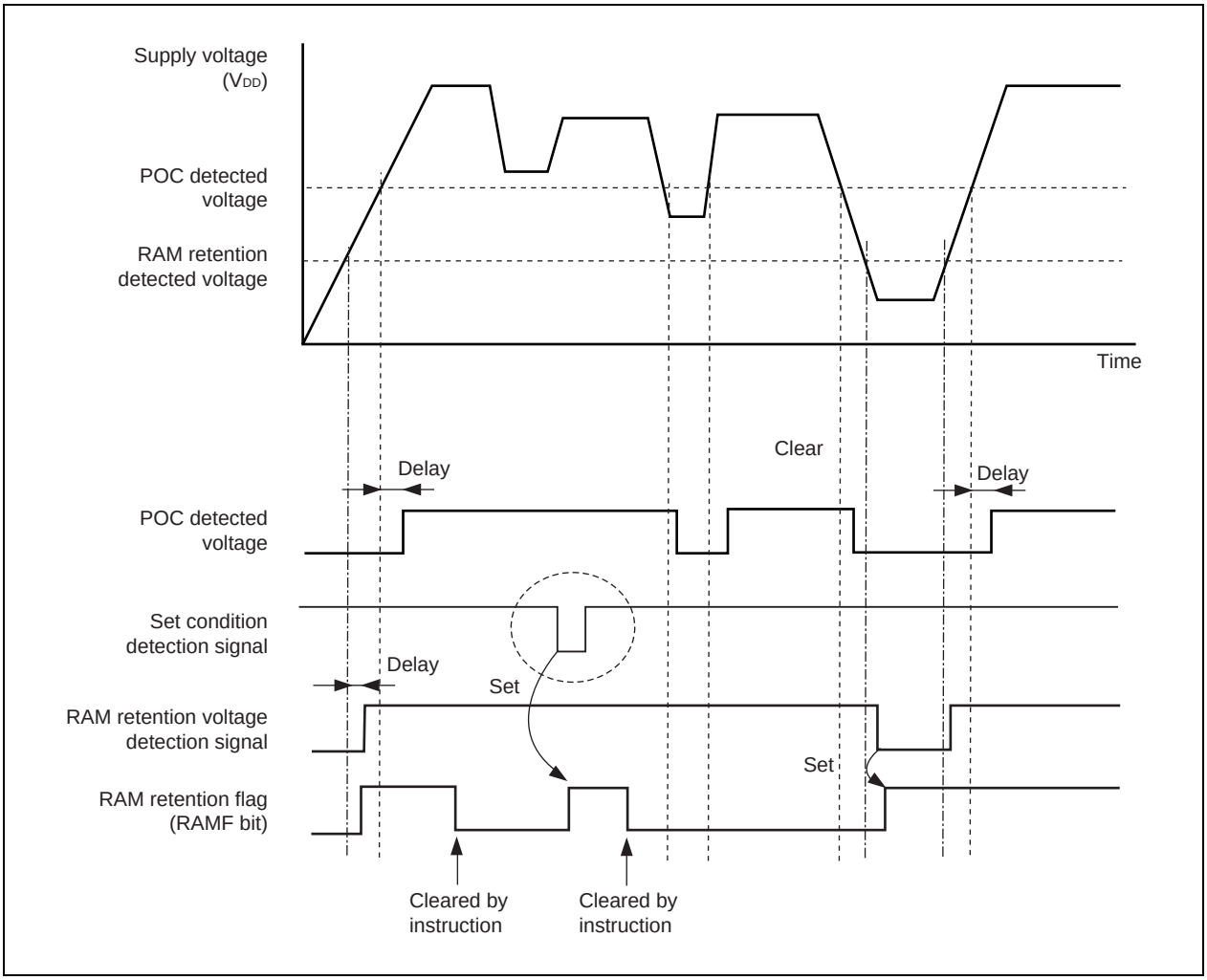


23.5 RAM Retention Voltage Detection Operation

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.

When the POC function is not used and when the RAM retention voltage detection function is used, be sure to input an external reset signal if the detected voltage falls below the operating voltage.

Figure 23-4. Operation Timing of RAM Retention Voltage Detection Function



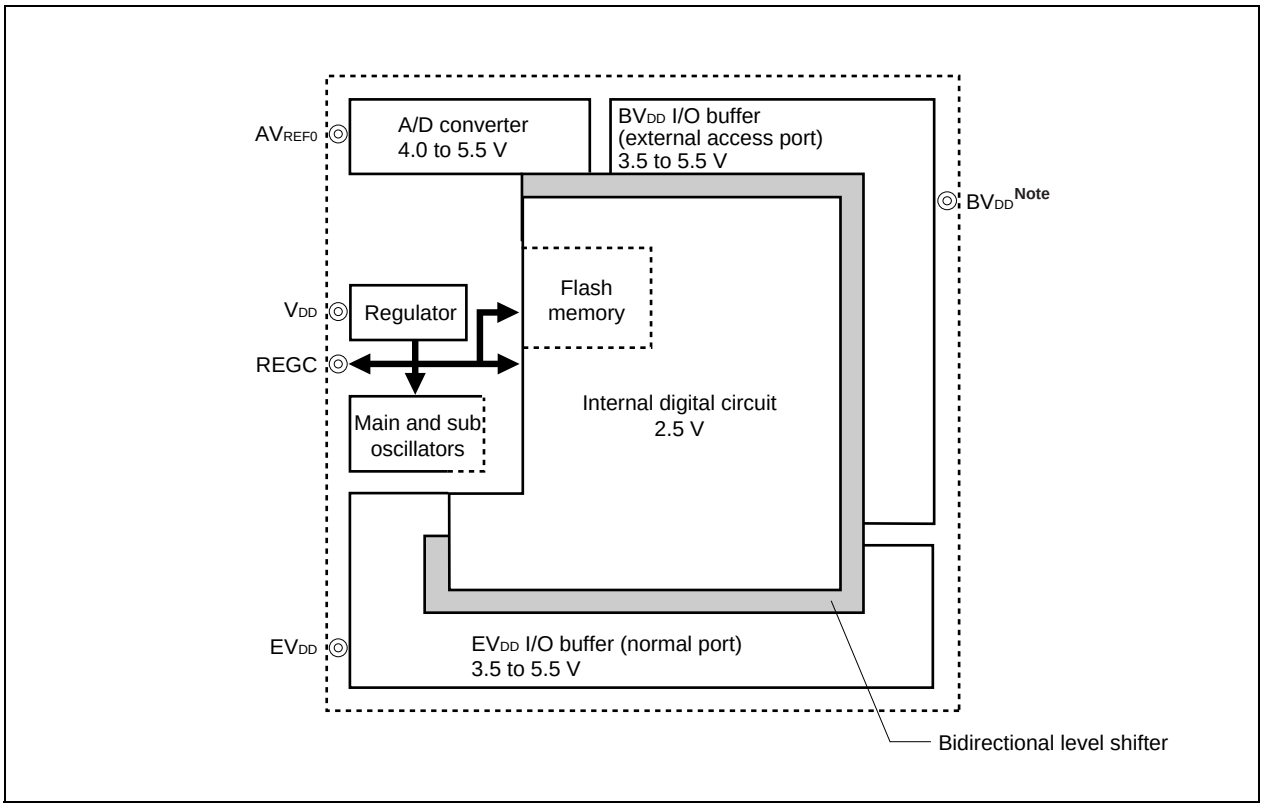
CHAPTER 24 REGULATOR

24.1 Overview

This product has an on-chip regulator to lower the power consumption and noise.

This regulator supplies a voltage lower than the supply voltage V_{DD} to the oscillator block and internal logic circuits (except the A/D converter and I/O buffers). The output voltage of the regulator is set to 2.5 V (Typ.).

★ **Figure 24-1. Regulator**



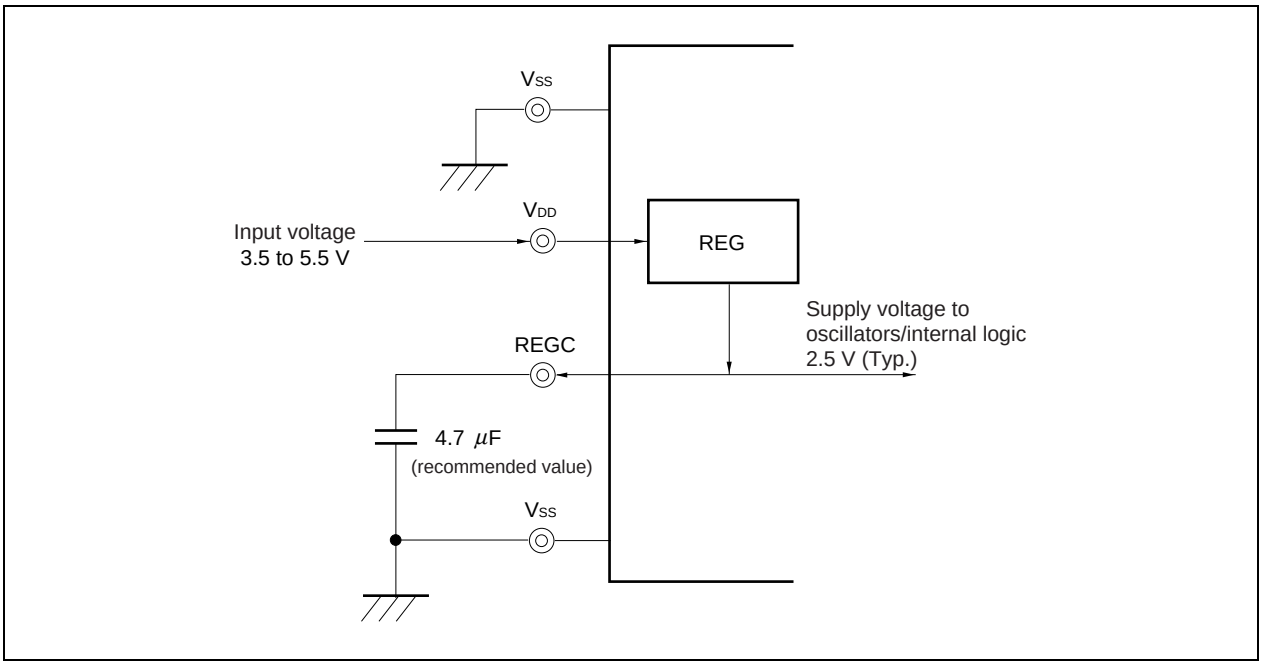
Note: BV_{DD} not available for V850ES/FE2 and V850ES/FF2

24.2 Operation

The regulator of this product operates in all operation modes (normal operation, HALT, IDLE1, IDLE2, software STOP, and sub-IDLE modes, and during reset).

To stabilize the output voltage of the regulator, connect a capacitor (4.7 µF (recommended value)) to the REGC pin. Connect the REGC pin as illustrated below.

★ **Figure 24-2. Connection of REGC Pin (REGC = Capacitance)**



CHAPTER 25 FLASH MEMORY

The following products are flash memory versions of the V850ES/Fx2

Remark: For the whole chapter it shall be agreed that V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2.

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.

- μPD70F3231, 70F3232, 70F3234
  On-chip 128 KB flash memory
- μPD70F3233, 70F3235, 70F3237
  On-chip 256 KB flash memory
- μPD70F3236
  On-chip 384 KB flash memory
- μPD70F3238
  On-chip 376 KB flash memory
- μPD70F3239
  On-chip 512 KB flash memory

When fetching an instruction, 4 bytes of the flash memory can be accessed in 1 clock in the same manner as the mask ROM versions.

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 V850ES/Fx2 on the target system
- For differentiating software in small-scale production of various models.
- For data adjustment when starting mass production

25.1 Features

- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 512/376(384)/256/128 KB
- Write voltage: Erase/write with a single power supply
- Rewriting method
- Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
- Rewriting flash memory by user program (self programming)
- Flash memory write prohibit function supported (security function)
- Safe rewriting of entire flash memory area by self programming using boot swap function
- Interrupts can be acknowledged during self programming.

25.1.1 Erasure unit

The units in which the 128, 256, 384, 376 or 512 KB flash memory can be erased are as follows.

(1) All-area erasure

The areas of flash memory xx000000H to xx01FFFFH, xx000000H to xx03FFFFH, xx000000H to xx05FFFFH, xx000000H to xx05DFFFH and xx000000H to xx07FFFFH can be erased at the same time.

(2) Block erasure

The flash memory can be erased in block units

Block / Flash	**128K Flash**	**256K Flash**	**384K Flash**	**376K Flash**	**512K Flash**
Block 15					4 KB
Block 14					4 KB
Block 13					4 KB
Block 12					4 KB
Block 11			32 KB		60 KB
Block 10			32 KB		60 KB
Block 9			32 KB	60 KB	60 KB
Block 8			32 KB	60 KB	60 KB
Block 7		8 KB	8 KB	8 KB	8 KB
Block 6		8 KB	8 KB	8 KB	8 KB
Block 5		56 KB	56 KB	56 KB	56 KB
Block 4		56 KB	56 KB	56 KB	56 KB
Block 3	8 KB	8 KB	8 KB	8 KB	8 KB
Block 2	56 KB	56 KB	56 KB	56 KB	56 KB
Block 1	8 KB	8 KB	8 KB	8 KB	8 KB
Block 0	56 KB	56 KB	56 KB	56 KB	56 KB

25.1.2 Functional Outline

The internal flash memory of the V850ES/Fx2 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850ES/Fx2 has already been mounted on the target system or not (off-board/on-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed after production/shipment of the target system. A boot swap function that rewrites the entire flash memory area safely is also supported. In addition, interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 25-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. (During self-programming, instructions cannot be fetched from or data access cannot be made to the internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 25-2. Basic Functions

Function	Functional Outline	Support (○: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	○	○
Chip erasure	The contents of the entire memory area are erased all at once.	○	×
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	○	○
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash programmer.	○	× (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	○	○
Security setting	Use of the block erase command, chip erase command, and program command can be prohibited.	○	× (Only values set by on-board/off-board programming can be retained)

The following table lists the security functions. The block erase command prohibit, chip erase command prohibit, and program command prohibit functions are enabled by default after shipment, and security can be set by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

Table 25-3. Security Functions

Function	Function Outline	Rewriting Operation When Prohibited (○: Executable, ×: Not Executable)	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	Block erase command: × Chip erase command: ○ Program command: ○	Can always be rewritten regardless of setting of prohibition
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks are prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.	Block erase command: × Chip erase command: × Program command: ○	
Program command prohibit	Write and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.	Block erase command: × Chip erase command: ○ Program command: ×	

25.2 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 V850ES/Fx2 mounted on the target system. Mount a connector that connects the dedicated flash programmer on the target system.

(2) Off-board programming

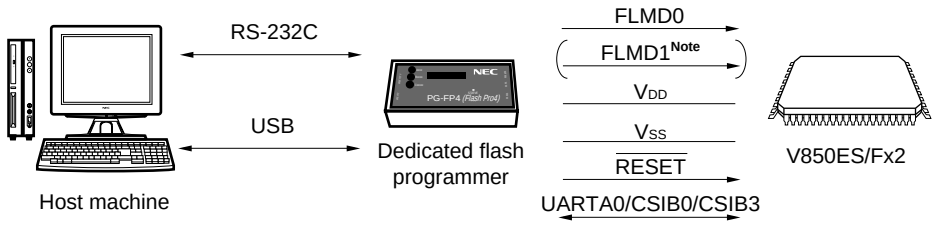
The flash memory of the V850ES/Fx2 can be written before the device is mounted on the target system, by using a dedicated program adapter (FA series).

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

25.3 Programming Environment

The environment necessary to write a program to the flash memory of the V850ES/Fx2 is shown below.

Figure 25-1. Environment to Write Program to Flash Memory



Note Connect the FLMD1 pin to the flash programmer or connect to a GND via a pull-down resistor on the board.

A host machine is required for controlling the dedicated flash programmer.

UARTA0 or CSIB0 is used as the interface between the dedicated flash programmer and the V850ES/Fx2 to manipulate the flash programmer by writing or erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

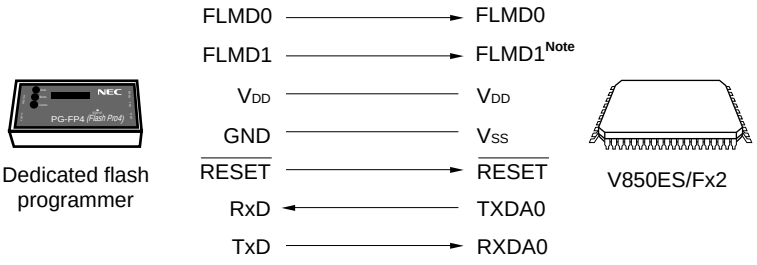
25.4 Communication Mode

Serial communication is performed between the dedicated flash programmer and the V850ES/Fx2 by using UARTA0 or CSIB0 of the V850ES/Fx2.

(1) UARTA0

Transfer rate: 9,600 - 153,600 bps

Figure 25-2. Communication with Dedicated Flash Programmer (UARTA0)



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

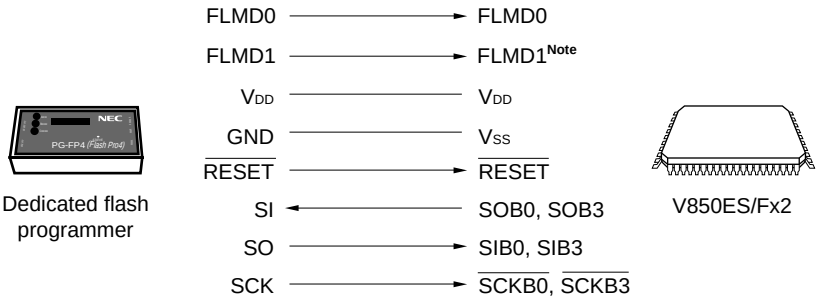
Cautions 1. Process the pins not shown in accordance with processing of unused pins (see 2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Handling of Unused Pins). To connect a resistor, a resistor of 1 k to 10 kΩ is recommended.

2. Please do not input high level in $\overline{\text{DRST}}$ pin.

(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 25-3. Communication with Dedicated Flash Programmer (CSIB0)



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

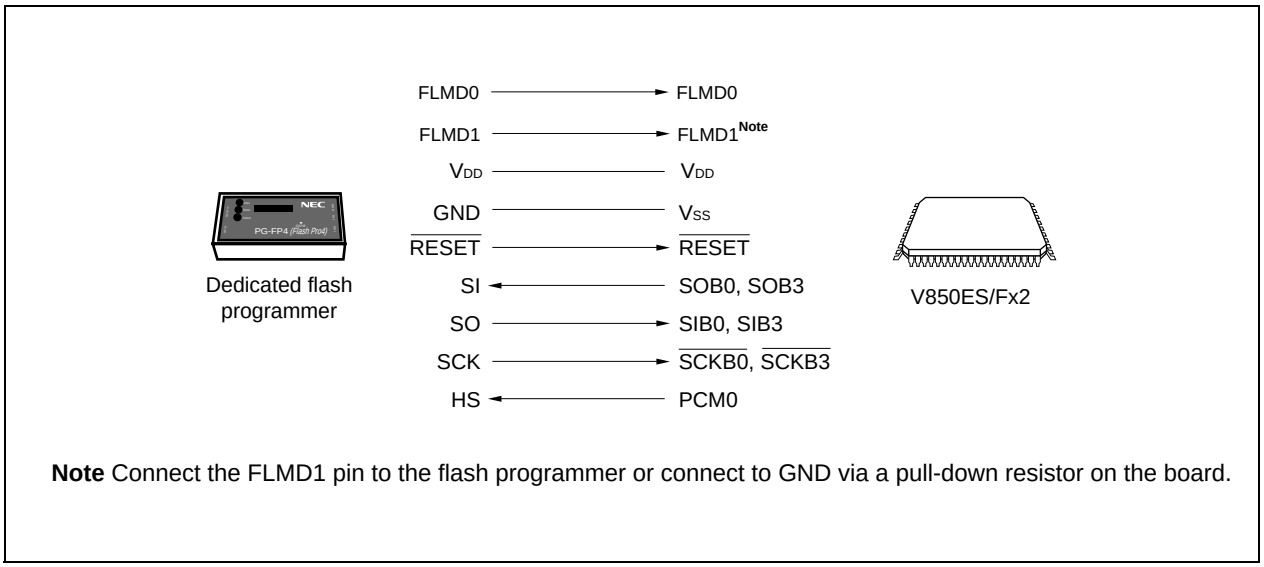
Cautions 1. Process the pins not shown in accordance with processing of unused pins (see 2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Handling of Unused Pins). To connect a resistor, a resistor of 1 k to 10 kΩ is recommended.

2 Please do not input high level in $\overline{\text{DRST}}$ pin.

(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 25-4. Communication with Dedicated Flash Programmer (CSIB0+HS)



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

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 Supplies and Handling of Unused Pins). To connect a resistor, a resistor of 1 k to 10 kΩ is recommended.

2. Please do not input high level in $\overline{\text{DRST}}$ pin.

The dedicated flash programmer outputs a transfer clock and the V850ES/Fx2 operates as a slave.

If the PG-FP4 is used as the flash programmer, it generates the following signals for the V850ES/Fx2. For details, refer to the **PG-FP4 User's Manual (U15260E)**.

Table 25-4. Signal Connections of Dedicated Flash Programmer (PG-FP4)

PG-FP4			V850ES/Fx2 Note 1	Processing for Connection		
Signal Name	I/O	Pin Function	Pin Name	UARTA0	CSIB0	CSIB0 + HS,
FLMD0	Output	Write enable/disable	FLMD0	◎	◎	◎
FLMD1	Output	Write enable/disable	FLMD1	◎Note 2	◎Note 2	◎Note 2
VDD	–	V_{DD} voltage generation/voltage monitor	V_{DD}	◎	◎	◎
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output to V850ES/Fx2	X1, X2	×Note 3	×Note 3	×Note 3
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI/RxD	Input	Receive signal	SOB0, TXDA0	◎	◎	◎
SO/TxD	Output	Transmit signal	SIB0, RXDA0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCKB0}}$	×	◎	◎
HS	Input	Handshake signal for CSIB0 + HS communication	PCM0	×	×	◎

Notes 1. V850ES/Fx2 stands for V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2,

2. Wire these pins as shown in Figure 25-6, or connect then to GND via pull-down resistor on board.

3. Clock cannot be supplied via the CLK pin of the flash programmer. Create an oscillator on board and supply the clock.

Remark ◎: Must be connected.

×: Does not have to be connected.

★ **Figure 25-5. Example of Wiring of V850ES/FJ2 Flash Writing Adapter (FA-144GJ-UEN) (In CSIB0 + HS Mode) (1/2)**

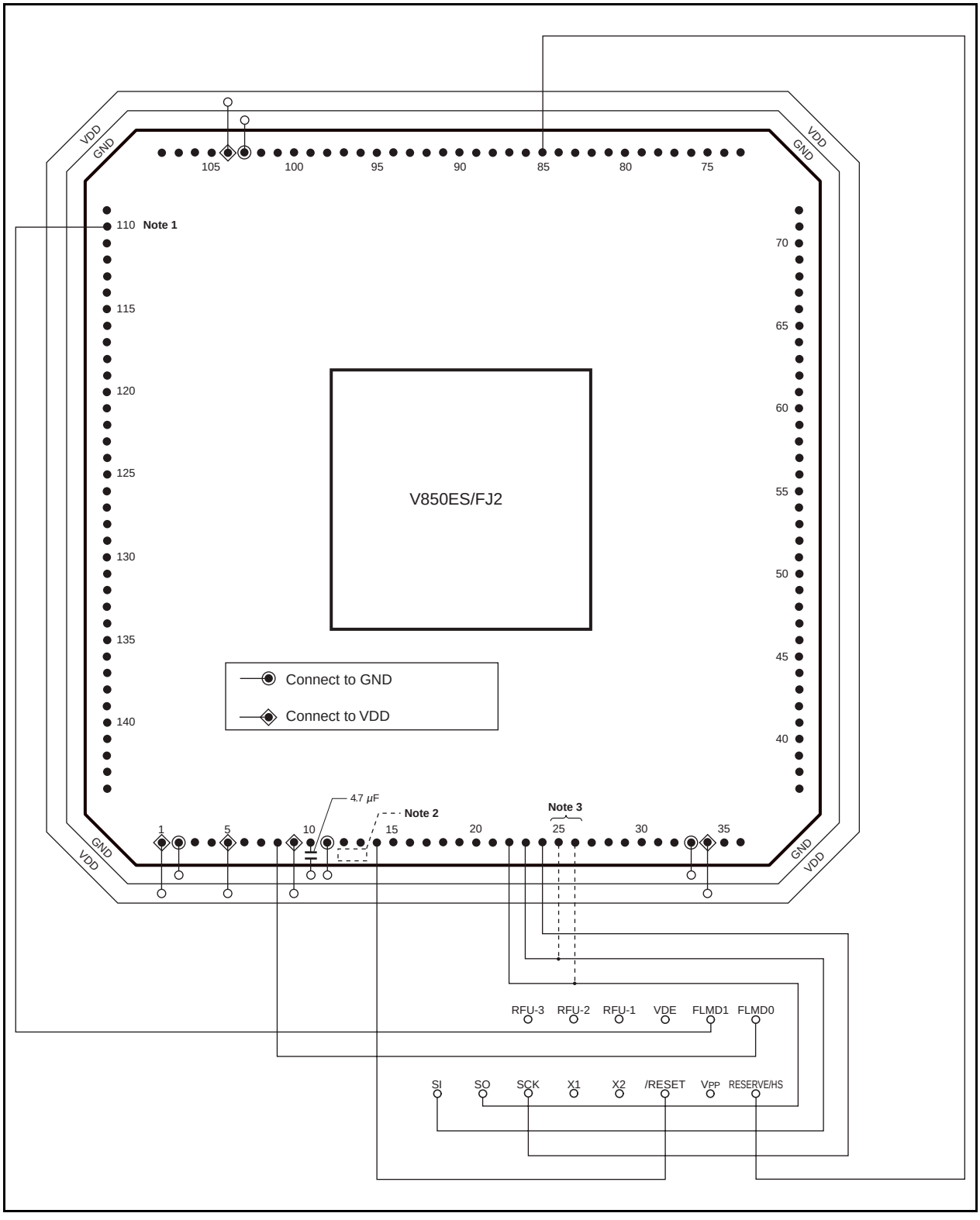
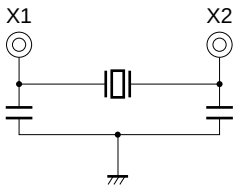


Figure 25-5. Example of Wiring of V850ES/FJ2 Flash Writing Adapter (FA-144GJ-UEN) (In CSIB0 + HS Mode) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by the broken lines). Here is an example of the oscillator.

Example



3. Pins used when UARTA0 is used.

Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

Remarks 1. Process the pins not shown in accordance with processing of unused pins (see **2.4 Pin I/O Circuit Types and Recommended Connection of Unused Pins**).

2. This adapter is for the 144-pin plastic LQFP package.

25.5 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.

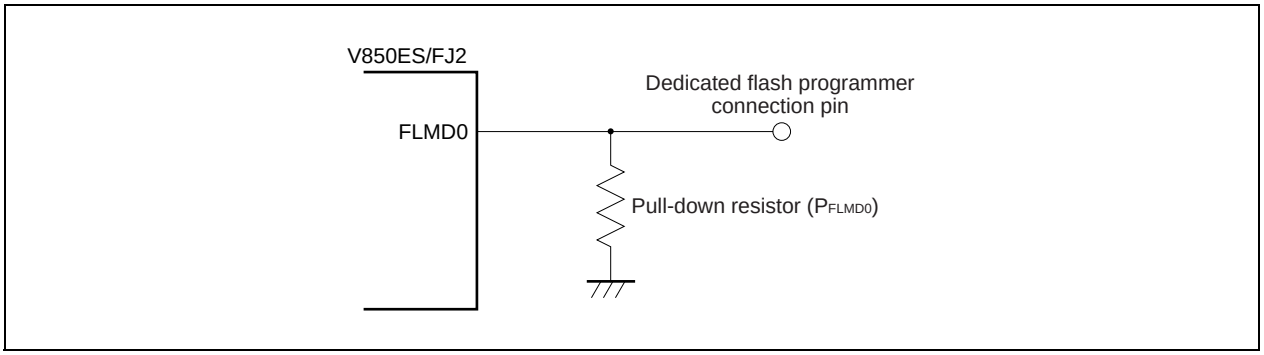
When the flash memory programming mode is set, all the pins not used for flash memory programming are in the same status as that immediately after reset. Therefore, all the ports go into an output high-impedance state, and the pins must be processed correctly if the external device does not recognize the output high-impedance state.

25.5.1 FLMD0 pin

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of VDD level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, see **25.7.5 (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.

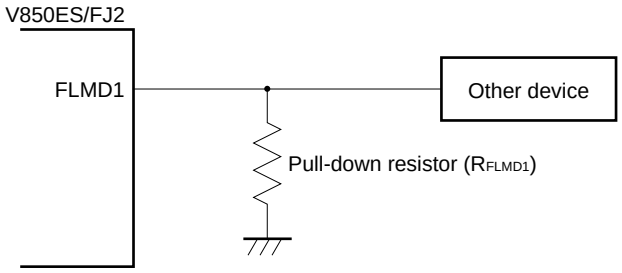
Figure 25-6. Example of Connection of FLMD0 Pin



25.5.2 FLMD1 pin

If 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. If V_{DD} 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 25-7. Example of Connection of FLMD1 Pin



Caution If a V_{DD} signal is input to the FLMD1 pin from other device during on-board writing and immediately after reset, isolate this signal.

Table 25-5. 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

25.5.3 Serial interface pins

The pins used by each serial interface are shown in the table below.

Table 25-6. Pins Used by Each Serial Interface

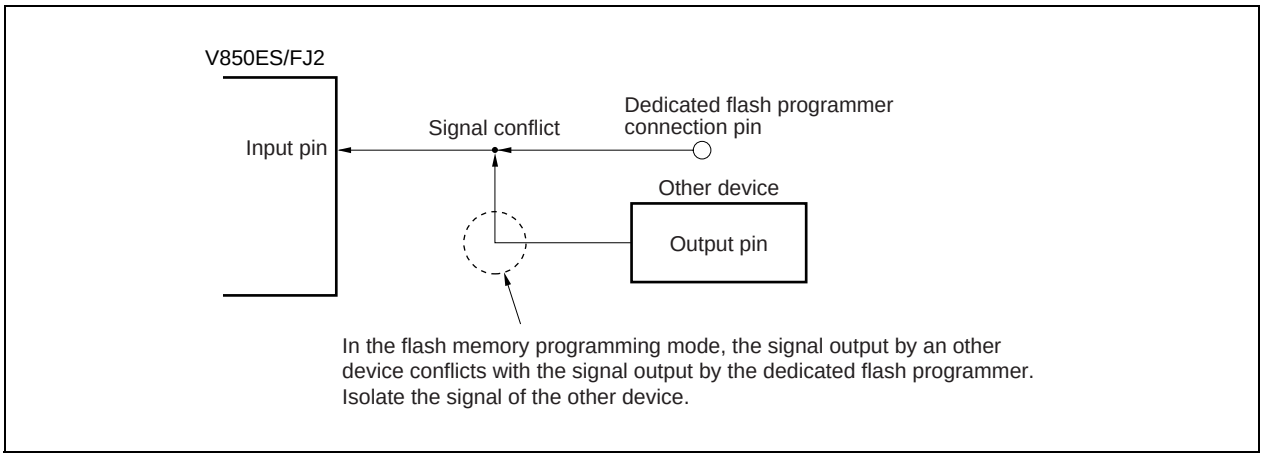
Serial Interface	Pins
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
UARTA0	TXDA0, RXDA0

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) connected to another device (output), a signal conflict occurs. To avoid this signal conflict, isolate the connection with the other device, or place the other device in an output high-impedance state.

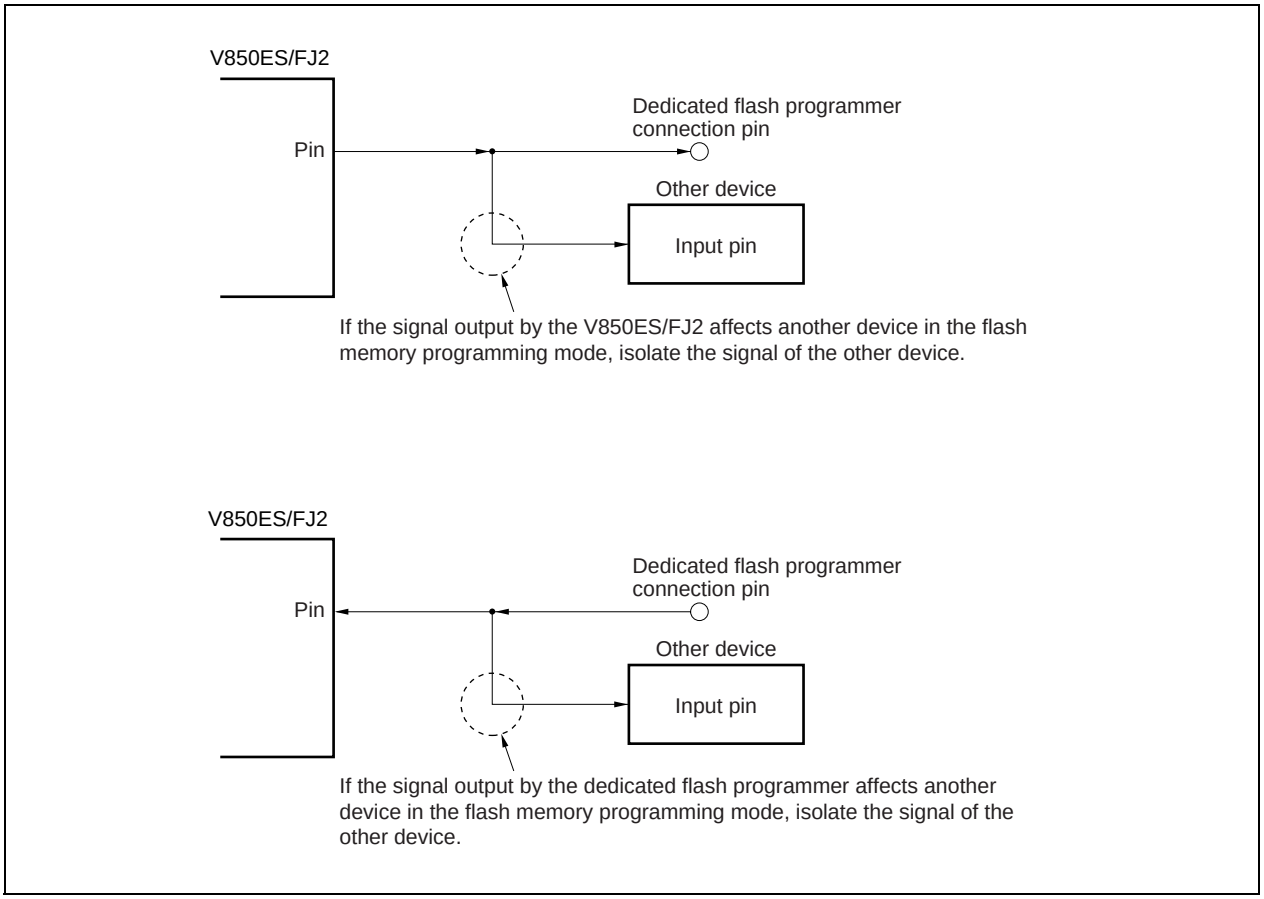
Figure 25-8. Signal Conflict (Input Pin of Serial Interface)



(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 signal input to the other device is ignored.

Figure 25-9. Abnormal Operation of Other Device

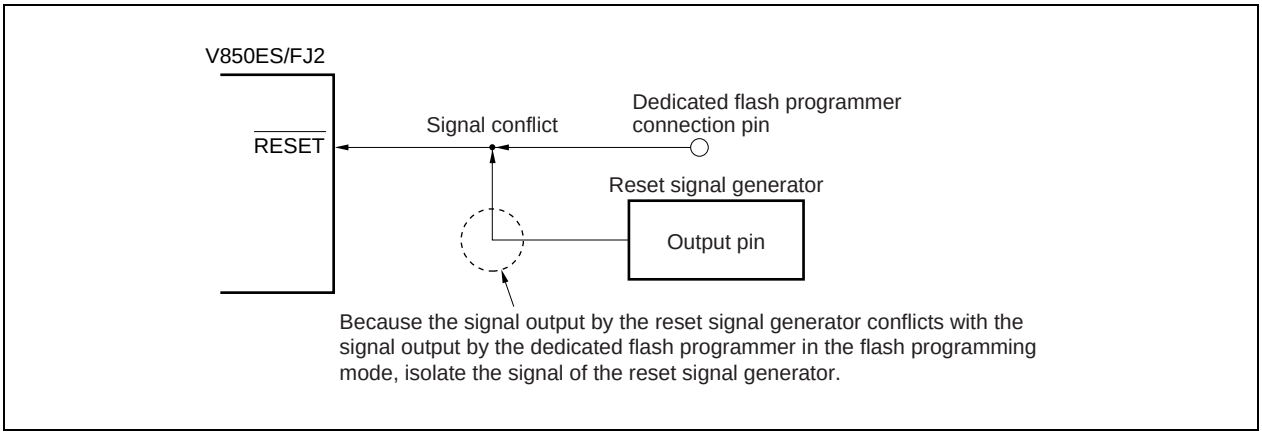


25.5.4 $\overline{\text{RESET}}$ pin

When the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ 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 25-10. Signal Conflict ($\overline{\text{RESET}}$ Pin)



25.5.5 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 V_{DD} or V_{SS} via a resistor.

25.5.6 Other signal pins

Connect X1, X2, XT1, XT2, and REGC in the same status as that in the normal operation mode.

During flash memory programming, input a low level to the DRST pin or leave it open. Do not input a high level.

25.5.7 Power supply

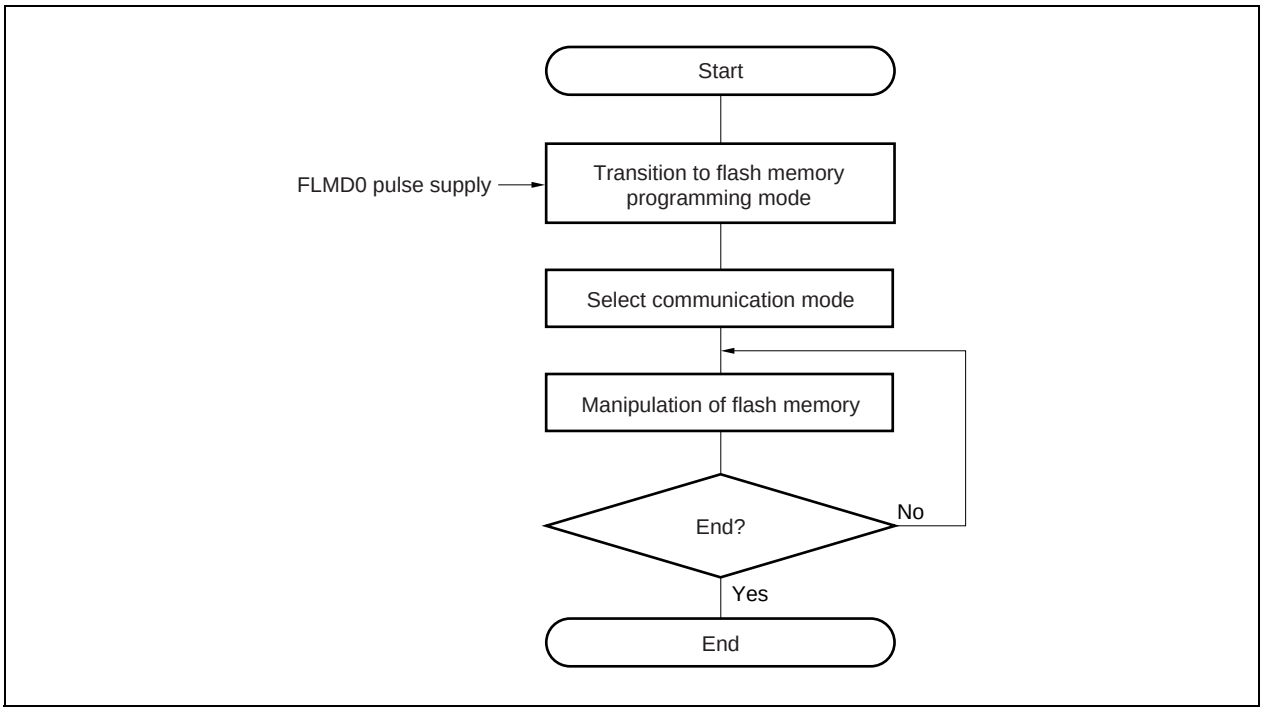
Supply the same power to the power supply pins (VDD, VSS, EVDD, EVSS, BVDD, BVSS, AVSS and AVREF0) as in the normal operation mode.

25.6 Programming Method

25.6.1 Flash memory control

The procedure to manipulate the flash memory is illustrated below.

Figure 25-11. Flash Memory Manipulation Procedure

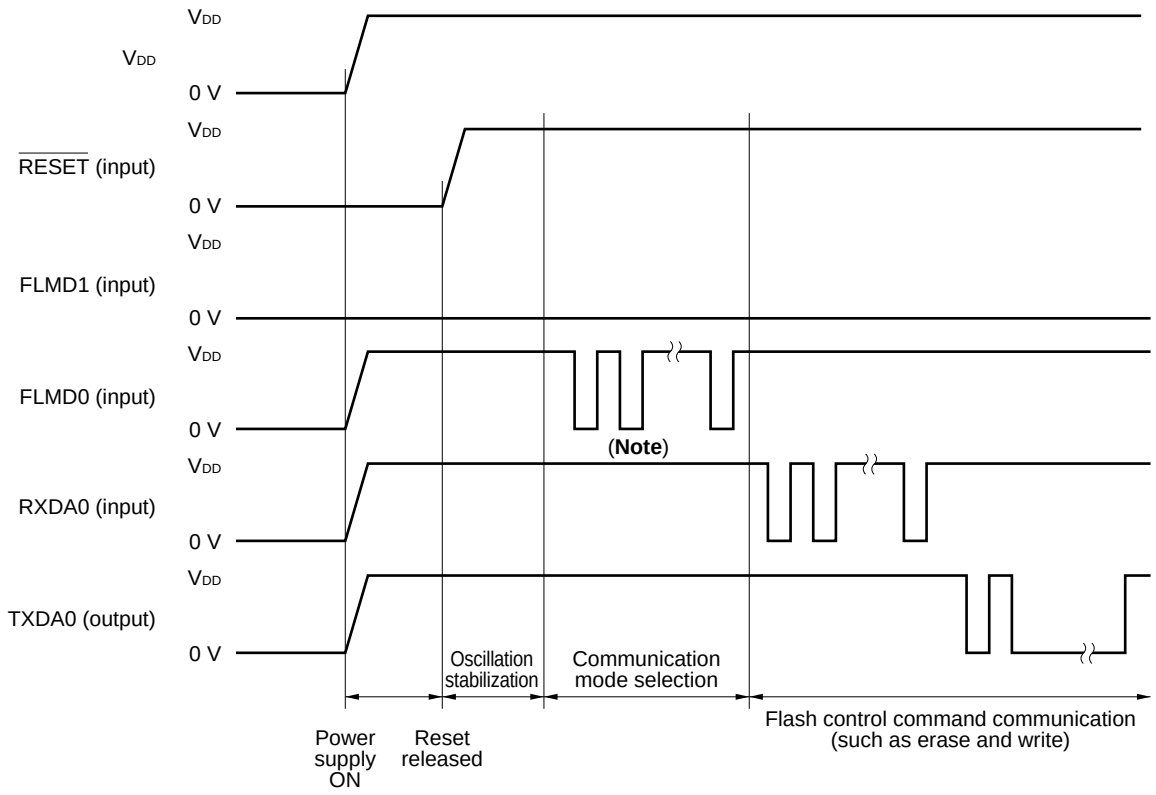


25.6.2 Selecting communication mode

In the V850ES/Fx2., the communication mode is selected by inputting pulses (up to 11 pulses) to the FLMD0 pin after the flash memory programming mode is set. These FLMD0 pulses are generated by the dedicated flash programmer.

The relationship between the number of pulses and the communication mode is shown in the figure below.

Figure 25-12. Flash Memory Programming Mode



Note The number of clocks to be inserted differs depending on the communication mode.

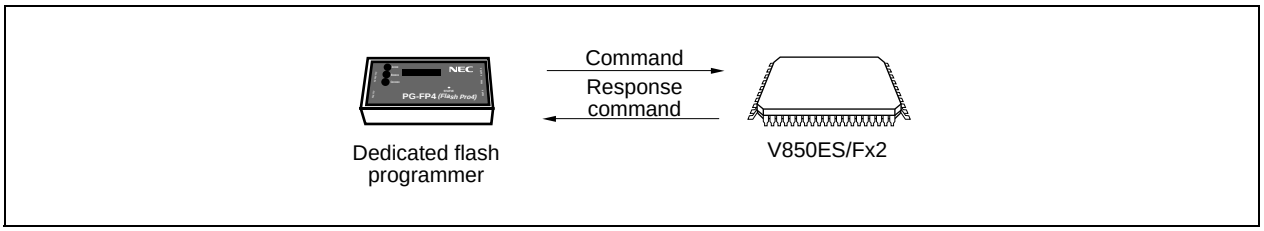
FLMD0 Pulse	Communication Mode	Remark
8	CSIB0	V850ES/Fx2 operates as slave. MSB first
11	CSIB0 + HS	V850ES/FJ2 operates as slave. MSB first
0	UARTA0	Communication rate: 9,600 bps (after reset), LSB first
Others	-	Setting prohibited

Caution When UARTA is selected, the receive clock is calculated based on the reset command that is sent from the dedicated flash programmer after reception of the FLMD0 pulse.

25.6.3 Communication commands

The V850ES/Fx2. communicates with the dedicated flash programmer via commands. The commands sent by the dedicated flash programmer to the V850ES/Fx2 are called commands, and the response signals sent by the V850ES/Fx2 to the flash programmer are called response commands.

Figure 25-13. Communication Commands



The following table lists the flash memory control commands of the V850ES/Fx2. All these commands are issued by the programmer, and the V850ES/Fx2 performs the corresponding processing.

Table 25-7. Flash Memory Control Commands

Classification	Command Name	Support			Function
		CSIB	CSIB + HS	UARTA	
Blank check	Block blank check command	√	√	√	Checks erasure status of 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 status.
	Oscillation frequency setting command	√	√	√	Sets oscillation frequency.
	Baud rate setting command	–	–	√	Sets baud rate when UART 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 V850ES/Fx2 returns a response command in response to the command issued by the flash programmer. The response commands sent by the V850ES/Fx2 are listed below.

Table 25-8. Response Commands

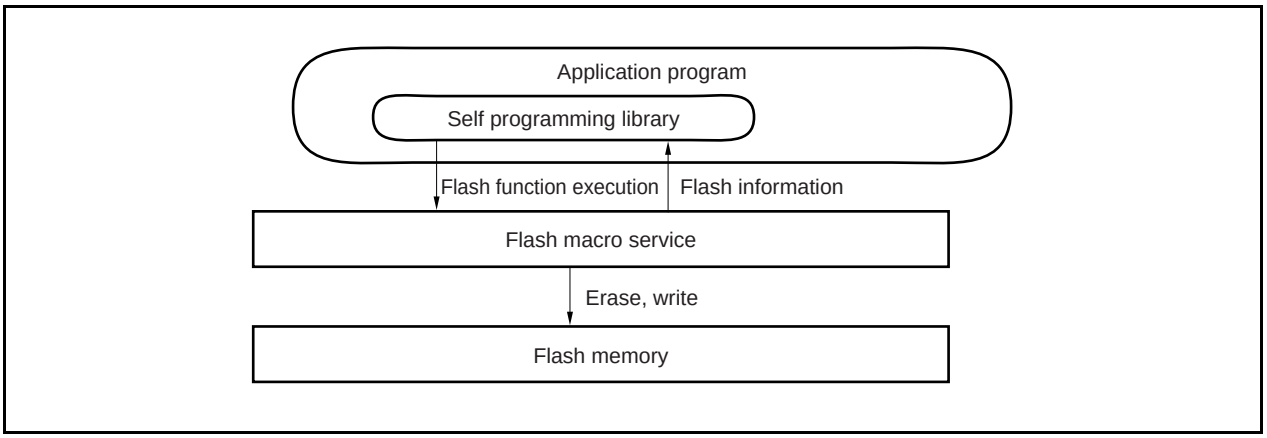
Response Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

25.7 Rewriting by Self Programming

25.7.1 Overview

The V850ES/Fx2 supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

Figure 25-14. Concept of Self Programming

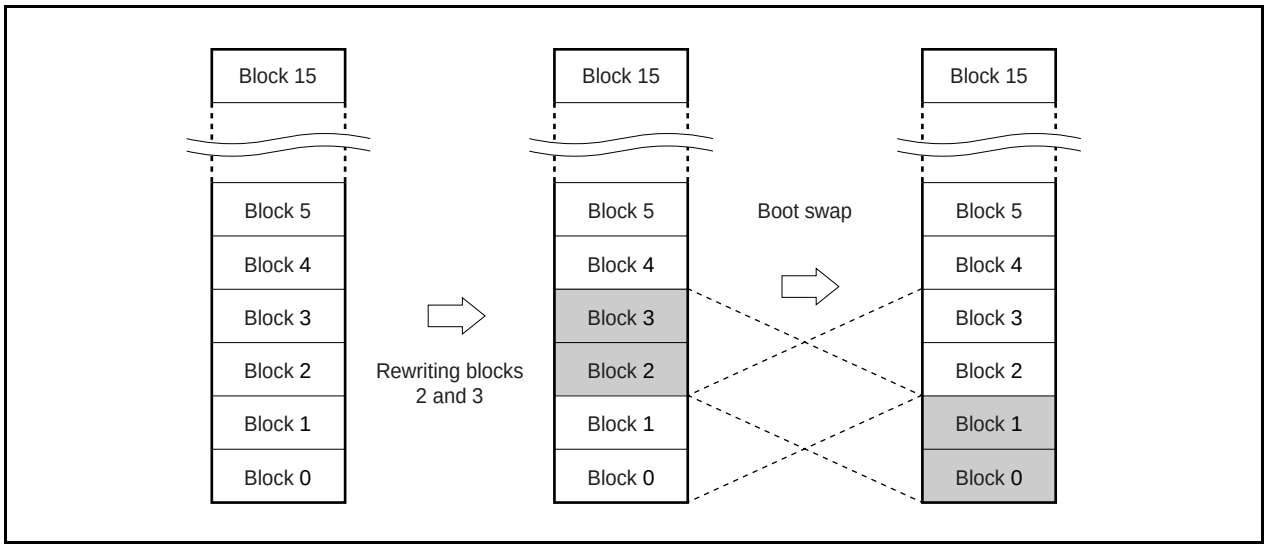


25.7.2 Features

(1) Secure self programming (boot swap function)

The V850ES/Fx2 supports a boot swap function that can exchange the physical memory of blocks 0 and 1 with the physical memory of blocks 2 and 3. By writing the start program to be rewritten to blocks 2 and 3 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because the correct user program always exists in blocks 0 and 1.

Figure 25-15. Rewriting Entire Memory Area (Boot Swap)



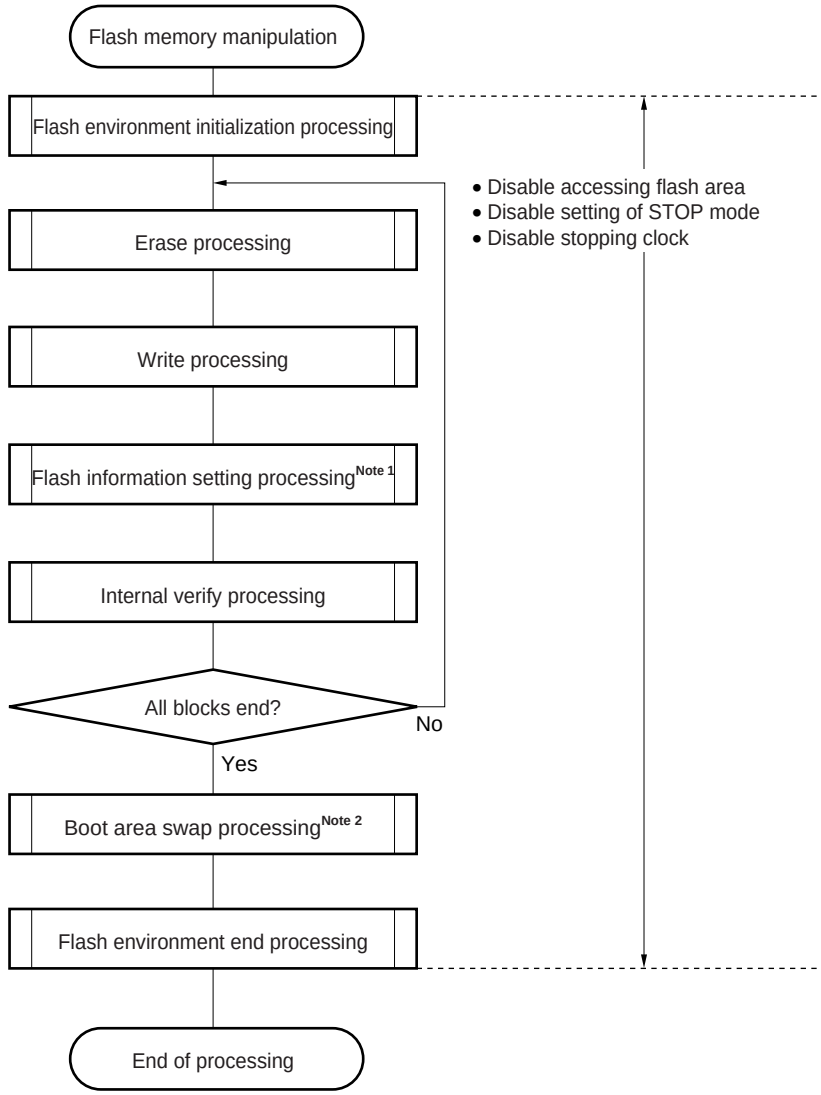
(2) Interrupt support

Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the V850ES/Fx2, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

25.7.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

Figure 25-16. Standard Self Programming Flow



Notes 1. If a security setting is not performed, flash information setting processing does not have to be executed.

2. If boot swap is not used, flash information setting processing and boot swap processing do not have to be executed.

25.7.4 Flash functions

Table 25-9. Flash Function List

Function Name	Outline	Support
FlashEnv	Initialization of flash control macro	√
FlashBlockErase	Erasure of only specified one block	√
FlashWordWrite	Writing from specified address	√
FlashBlockIVerify	Internal verification of specified block	√
FlashBlockBlankCheck	Blank check of specified block	√
FlashFLMDCheck	Check of FLMD pin	√
FlashStatusCheck	Status check of operation specified immediately before	√
FlashGetInfo	Reading of flash information	√
FlashSetInfo	Setting of flash information	√
FlashBootSwap	Swapping of boot area	√
FlashSetUserHandler	User interrupt handler registration function	√

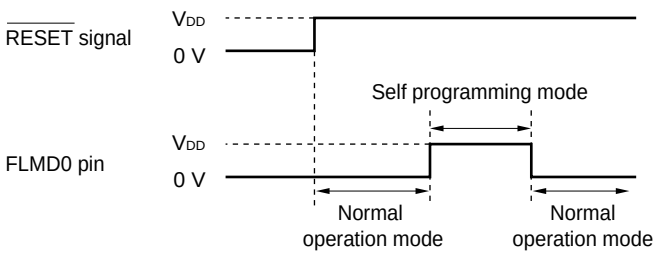
25.7.5 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode when reset is released and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V when reset is released and a normal operation is executed. It is also necessary to apply a voltage of V_{DD} level to the FLMD0 pin during the self programming mode period via port control before the memory is rewritten.

When self programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

Figure 25-17. Mode Change Timing



Caution Make sure that the FLMD0 pin is at 0 V when reset is released.

25.7.6 Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 25-10. Internal Resources Used

Resource Name	Description
Entry RAM area (124 bytes of either internal RAM/external RAM)	Routines and parameters used for the flash macro service are located in this area. The entry program and default parameters are copied by calling a library initialization function.
Stack area (user stack + 300 bytes)	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code (1900 bytes)	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.
NMI interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status the interrupt servicing start address must be registered in advance by a registration function.

CHAPTER 26 OPTION FUNCTION

26.1 Mask Options

This product series has an option data area where a block subject to mask options is specified.

When writing a program to a flash memory version or a mask ROM version, be sure to set the option data corresponding to the following option in the program at address 007AH as default data.

The data in this area cannot be rewritten during program execution.

	7	6	5	4	3	2	1	0
007AH	Subclock7	Subclock6	0	0	0	MP2	WDTMD1	RMOPIN

Subclock7	Subclock6	Selection of subclock
0	0	Sub-Crystal connection
0	1	Setting prohibited
1	0	Setting prohibited
1	1	RC oscillation connection

MP2	POC function for Mask ROM
0	Without POC function
1	With POC function
Remarks:	**1.** MP2 bit is only used for mask ROM product. **2.** Setting of MP2 bit does not affect flash ROM product function. **3.** For flash product POC function is distinguished by device name “M1” and “M2”.

WDTMD1	WDT2 mask option
0	Count clock for WDT2 can be selected by software and Overflow signal can be selected from INTWDT2 or WDT2RES.
1	Count clock is fixed to Ring-OSC and overflow signal is fixed to WDT2RES.

RMOPIN	Ring-OSC mask option
0	Ring-OSC can be stopped by software
1	Ring-OSC cannot be stopped

Caution: Do not make any settings other than the above.

Remark: In case of mask products, set the option data same as flash memory products.

Some examples for possible settings are described in Table 26-1 (Selection is not complete).

Table 26-1. Example settings for mask option (assortment)

Address	Set Value	Setting
007AH	00H	Ring-OSC: Can be stopped. WDT2: Count clock can be selected. Overflow signal can be selected from INTWDT2 or WDT2RES. Subclock: Crystal resonator connection POC function for mask ROM product disabled
	03H	Ring-OSC: Cannot be stopped. WDT2: Count clock is fixed to Ring-OSC. Overflow signal is fixed to WDT2RES. Subclock: Crystal resonator connection POC function for mask ROM product disabled
	C2H	Ring-OSC: Can be stopped. WDT2: Count clock is fixed to Ring-OSC. Overflow signal is fixed to WDT2RES. Subclock: RC oscillation connection POC function for mask ROM product disabled
	C7H	Ring-OSC: Cannot be stopped. WDT2: Count clock is fixed to Ring-OSC. Overflow signal is fixed to WDT2RES. Subclock: RC oscillation POC function for mask ROM product

CHAPTER 27 ON-CHIP DEBUG FUNCTION

The V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2 include an on-chip debug unit. By connecting an N-Wire emulator, on-chip debugging can be executed with the V850ES/FE2, V850ES/FF2, V850ES/FG2 and V850ES/FJ2 alone.

Cautions: 1 The on-chip debug function is provided only in the flash memory version. It is not provided with the mask ROM version. However, the OCDM register also exists in the mask ROM version and it controls the pull-down resistor connected to the P05/INTP2 pin, so set the OCDM register even for the mask ROM version.

2. The following debug functions are supported and 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.

27.1 Functional Outline

27.1.1 Type of on-chip debug unit

The on-chip debug unit is RCU1 (Run Control Unit 1).

27.1.2 Debug functions

(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 4-3 Alternate-Function Pins of Port 0** for details.

(3) Forced reset function

The V850ES/FE2, V850ES/FF2, V850ES/FG2 AND V850ES/FJ2 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 (IE-V850E1-CD-NW) of NEC Electronics is shown below.

NMI0 mask function: NMI pin
NMI1 mask function: WDT2 interrupt
NMI2 mask function: –
STOP mask function: –
HOLD mask function: $\overline{\text{HLDRQ}}$ pin
RESET mask function: $\overline{\text{RESET}}$ pin, WDT2 reset, POC reset[Note], LVI reset, clock monitor reset
DBINT mask function: –
WAIT mask function: Masks $\overline{\text{WAIT}}$ pin

Note This applies only to the products with a power-on clear function.

(10) Timer function

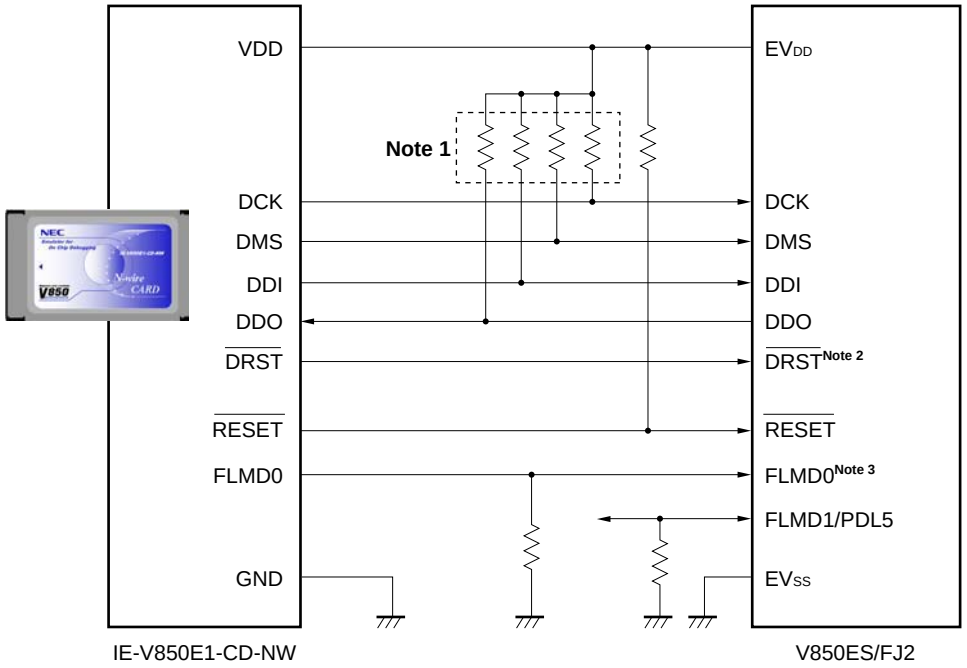
The execution time of the user program can be measured.

(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

27.2 Connection Circuit Example



Notes 1. Example of pin processing when N-Wire emulator is connected

2. A pull-down resistor is provided on chip.

3. For flash memory rewriting

27.3 Interface Signals

The interface signals are described below.

(1) $\overline{\text{DRST}}$

This is a reset input signal for the on-chip debug unit. It is a negative-logic signal that asynchronously initializes the debug control unit.

The IE-V850E1-CD-NW raises the $\overline{\text{DRST}}$ signal when it detects V_{DD} of the target system after the integrated debugger is started, and starts the on-chip debug unit of the device.

When the $\overline{\text{DRST}}$ signal goes high, a reset signal is also generated in the CPU.

When starting debugging by starting the integrated debugger, a CPU reset is always generated.

(2) DCK

This is a clock input signal. It supplies a 20 MHz clock from the IE-V850E1-CD-NW. In the on-chip debug unit, the DMS and DDI signals are sampled at the rising edge of the DCK signal, and the data DDO is output at its falling edge.

(3) DMS

This is a transfer mode select signal. The transfer status in the debug unit changes depending on the level of the DMS signal.

(4) DDI

This is a data input signal. It is sampled in the on-chip debug unit at the rising edge of DCK.

(5) DDO

This is a data output signal. It is output from the on-chip debug unit at the falling edge of the DCK signal.

(6) EV_{DD}

This signal is used to detect VDD of the target system. If VDD from the target system is not detected, the signals output from the IE-V850E1-CD-NW ($\overline{DRST}$, DCK, DMS, DDI, FLMD0, and $\overline{RESET}$) go into a high-impedance state.

(7) FLMD0

The flash self programming function is used for the function to download data to the flash memory via the integrated debugger. During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from IE-V850E1-CD-NW

Connect the FLMD0 signal of the IE-V850E1-CD-NW to the FLMD0 pin.

In the normal mode, nothing is driven by the IE-V850E1-CD-NW (high impedance).

During a break, the IE-V850E1-CD-NW raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed.

<2> To control from port

Connect any port of the device to the FLMD0 pin.

The same port as the one used by the user program to realize the flash self programming function may be used.

On the console of the integrated debugger, make a setting to raise the port pin to high level before executing the download function, or lower the port pin after executing the download function.

For details, refer to the **ID850QB Ver. 2.80 Integrated Debugger Operation User's Manual (U16973E)**.

(8) $\overline{RESET}$

This is a system reset input pin. If the $\overline{DRST}$ pin is made invalid by the value of the OCDM0 bit of the OCDM register set by the user program, on-chip debugging cannot be executed. Therefore, reset is effected by the IE-V850E1-CD-NW, using the $\overline{RESET}$ pin, to make the $\overline{DRST}$ pin valid (initialization).

27.4 Register

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

This register is used to select the normal operation mode or on-chip debug mode. This register is a special register and can be written only in a combination of specific sequences (refer to **3.4.9 Special registers**).

If the OCDM0 bit is 1 and if the $\overline{\text{DRST}}$ pin is high, the on-chip debug mode is selected.

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

After reset: 01H[Note 1] R/W Address: FFFFF9FCH

	7	6	5	4	3	2	1	0
OCDM	0	0	0	0	0	0	0	OCDM0

OCDM0	Specification of alternate-function pin of on-chip debug function[Note 2]
0	Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P05/INTP2/DRST pin.
1	When DRST pin is low: Normal operation mode (in which a pin that functions alternately as an on-chip debug function pin is used as a port/peripheral function pin) When DRST pin is high: On-chip debug mode (in which a pin that functions alternately as an on-chip debug function pin is used as an on-chip debug mode pin)

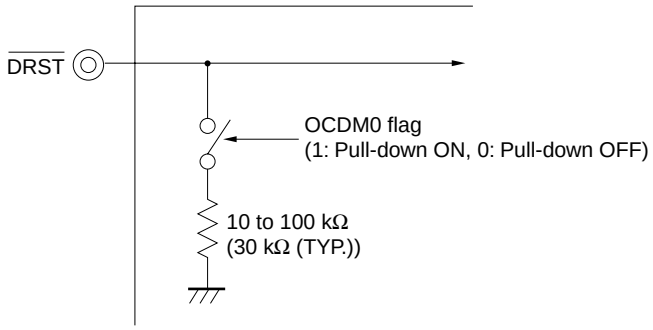
Notes 1. $\overline{\text{RESET}}$ input sets this register to 01H.
On reset by power-on clear: OCDM0 = 0
On occurrence of internal source reset (other than power-on clear): The OCDM register holds the value before occurrence of reset.

2. P05/INTP2/$\overline{\text{DRST}}$
P52/KR2/TIQ03/TOQ03/DDI
P53/KR3/TIQ00/TOQ00/DDO
P54/KR4/DCK
P55/KR5/DMS

Cautions 1. When using the DDI, DDO, DCK, and DMS pins not as on-chip debug pins but as port pins after external reset, any of the following actions must be taken.

- **Input a low level to the P05/INTP2/$\overline{\text{DRST}}$ pin.**
- **Set the OCDM0 bit. In this case, take the following actions.**
  <1> Clear the OCDM0 bit to 0.
  <2> Fix the P05/INTP2/$\overline{\text{DRST}}$ pin to the low level until <1> is completed.

2. The $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0. The mask ROM version does not have an on-chip debug function but it has the above pull-down resistor. With the mask ROM version also, therefore, the on-chip pull-down resistor must be disconnected by clearing the OCDM0 bit to 0.



27.5 Operation

The on-chip debug function is made invalid under the conditions shown in the table below.
When this function is not used, keep the $\overline{\text{DRST}}$ pin low until the OCDM.OCDM0 flag is cleared to 0.

OCDM0 Flag / $\overline{\text{DRST}}$ Pin	0	1
L	Invalid	Invalid
H	Invalid	Valid

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

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 27-1**). The $\overline{\text{DRST}}$ pin is internally pulled down while the OCDM0 bit is 1, and therefore, it can be left open.

After POC reset, the default value of the OCDM0 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 27-2**).

Figure 27-1. Timing Chart of Selecting Normal Operation Mode

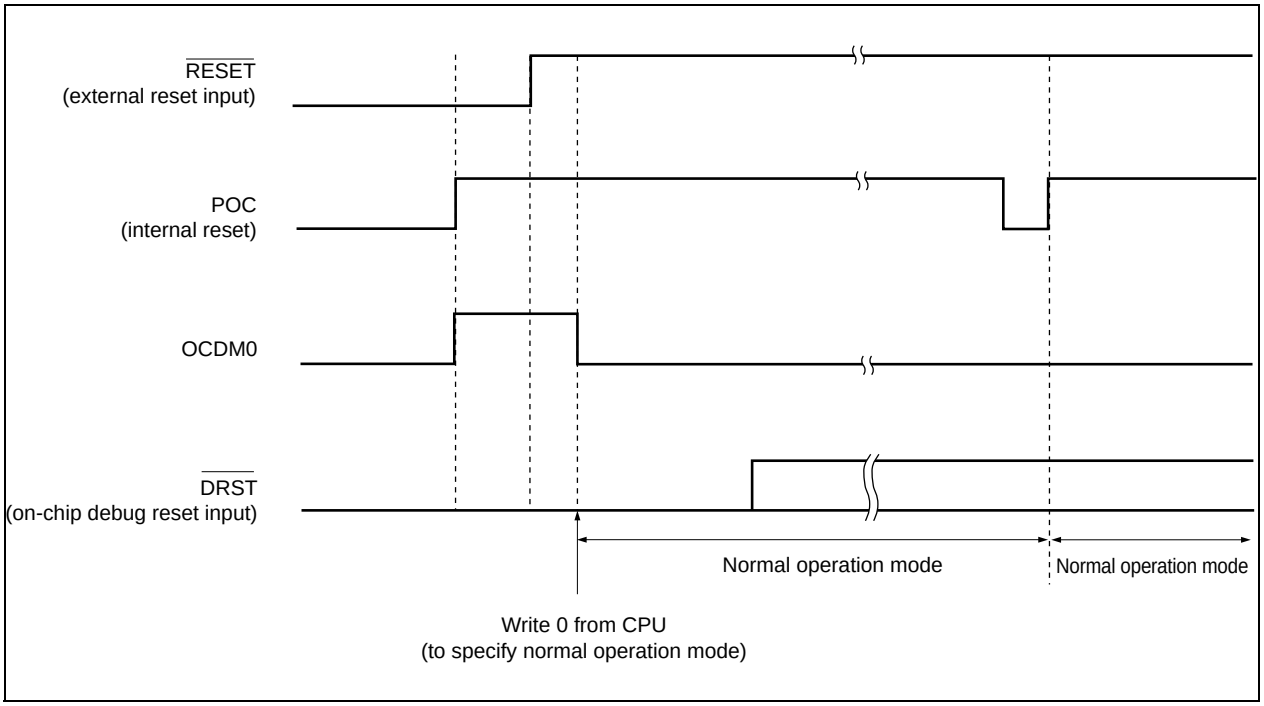
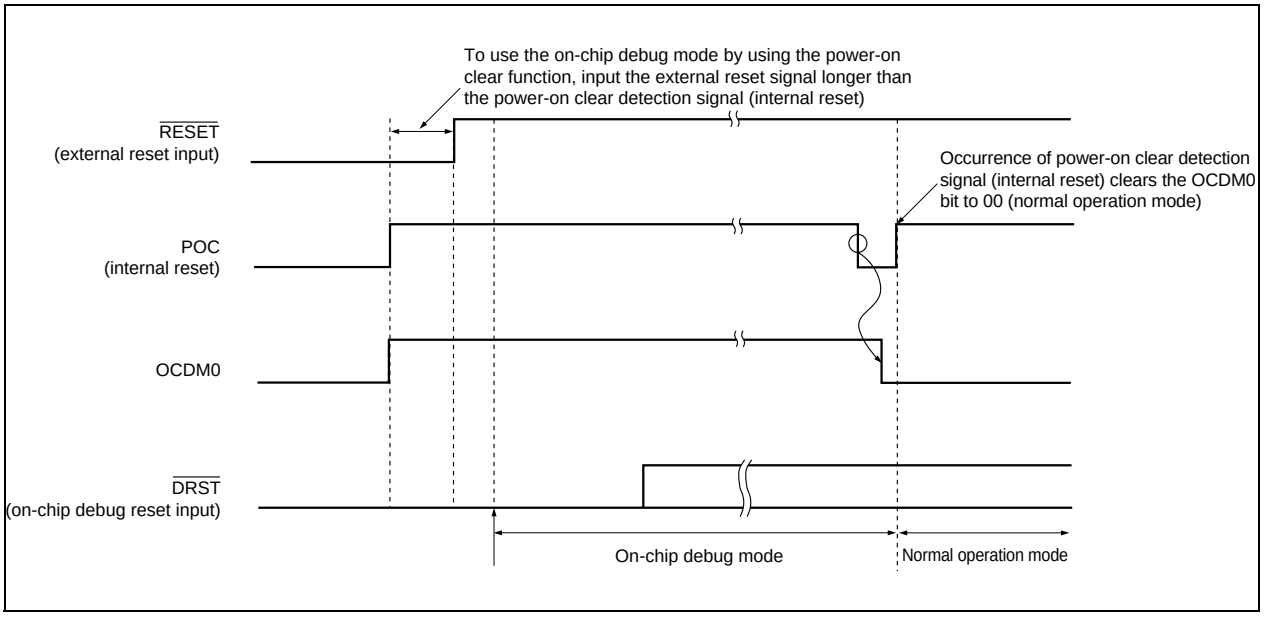


Figure 27-2. Timing Chart of Selecting On-Chip Debug Mode



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. (After power-on, from power-voltage is upper 4V, please release the RESET input pin.).

27.6 ROM Security Function

27.6.1 Security ID

The flash memory versions of the V850ES/FE2, V850ES/FF2, V850ES/FG2 AND V850ES/FJ2 perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the N-Wire emulator.

Set the ID code in the 10-byte on-chip flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

- When the N-Wire emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the N-Wire emulator enable flag is 0, even if the ID codes match.

If the IDs match, the security is released and reading flash memory and using the N-Wire emulator are enabled.

(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.

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.

27.6.2 Setting

Example When the following values are set to addresses 0x70 to 0x79

Address	Value
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0xF1
0x78	0x23
0x79	0xD4
0x7A	Flash mask option

← (See **Chapter 26: Option Function**.)

The following shows program examples when the CA850 is used.

[Program example 1]

Following the “ILGOP” section (address 0x60); enter the 10-byte security code and 1-byte system reserved area data (00H).

```
#---------------------------------------
# ILGOP handler
#---------------------------------------
   .section   "ILGOP"   -- Interrupt handler address 0x60
                        -- Input ILGOP handler code
   .org       0x10      -- Skip handler address to 0x70

#---------------------------------------
#  SECURITYID (continue ILGOP handler)
#---------------------------------------
   .word                0x78563412   --0-3 byte code
   .word                0xF1DEBC9A   --4-7 byte code
   .hword               0xD423       --8-9 byte code
   .byte                0x00         --Flash mask option code
```

Caution When using the CA850 Ver. 3.00 or later, specify the option for disabling the generation of the security ID

The security ID addition function by linker is added from the CA850 Ver. 3.00. As a result, errors occur during linking in the above program example.

Error message:

```
F4264: start address (0x00000070) of section "SECURITY_ID" overlaps
       previous section "ILGOP" ended before address (0xXXXXXXXX).
```

[Program example 2]

Enter the 10-byte security code using the “SECURITY_ID” section (address 0x70).

```
#---------------------------------------
#  SECURITY_ID
#---------------------------------------
   .section   "SECURITY_ID"
   .word               0x78563412   --0-3 byte code
   .word               0xF1DEBC9A   --4-7 byte code
   .hword              0xD423       --8-9 byte code
```

Caution Data that can be set to the “SECURITY_ID” section is limited to 10 bytes. For this reason, data cannot be set to the system reserved area (0x7A) following the security code. Consequently, when using a device that needs to set data to the system reserved area, set the security code and system reserved area data using the method shown in “Program example 1”.

27.7 Connection to N-Wire Emulator

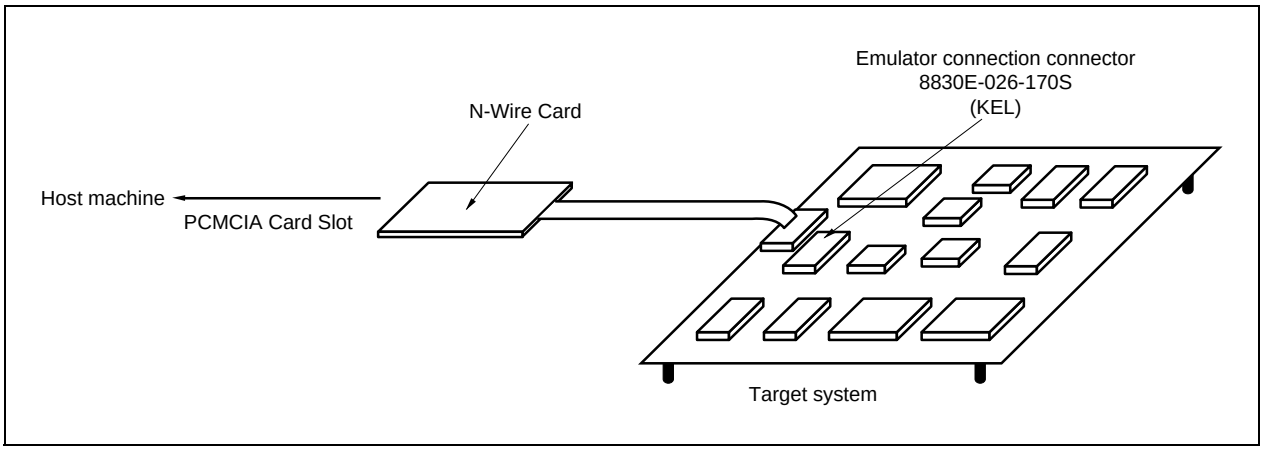
To connect the N-Wire emulator, a connector for emulator connection and a connection circuit must be mounted on the target system.

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.

27.7.1 KEL connector

O Product name
- 8830E-026-170S (KEL): Straight type
- 8830E-026-170L (KEL): Right-angle type

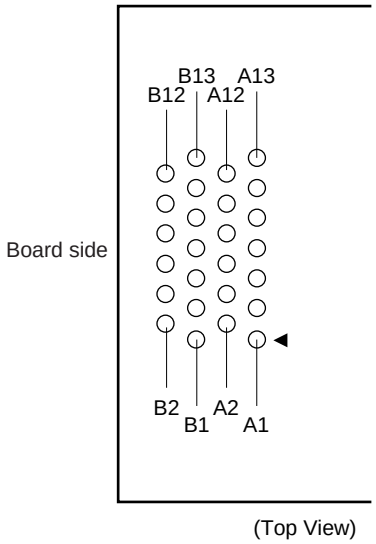
Figure 27-3. Connection to N-Wire Emulator (NEC Electronics IE-V850E1-CD-NW: N-Wire Card)



(1) Pin configuration

Figure 27-4 shows the pin configuration of the connector for emulator connection (target system side), and Table 27-1 shows the pin functions.

Figure 27-4. 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 27-1. 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	DDO	Output	Data output for N-Wire interface
A11	$\overline{\text{DRST}}$	Input	On-chip debug unit reset input
A12	$\overline{\text{RESET}}$	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 27-5 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 depends upon the emulator to be used.

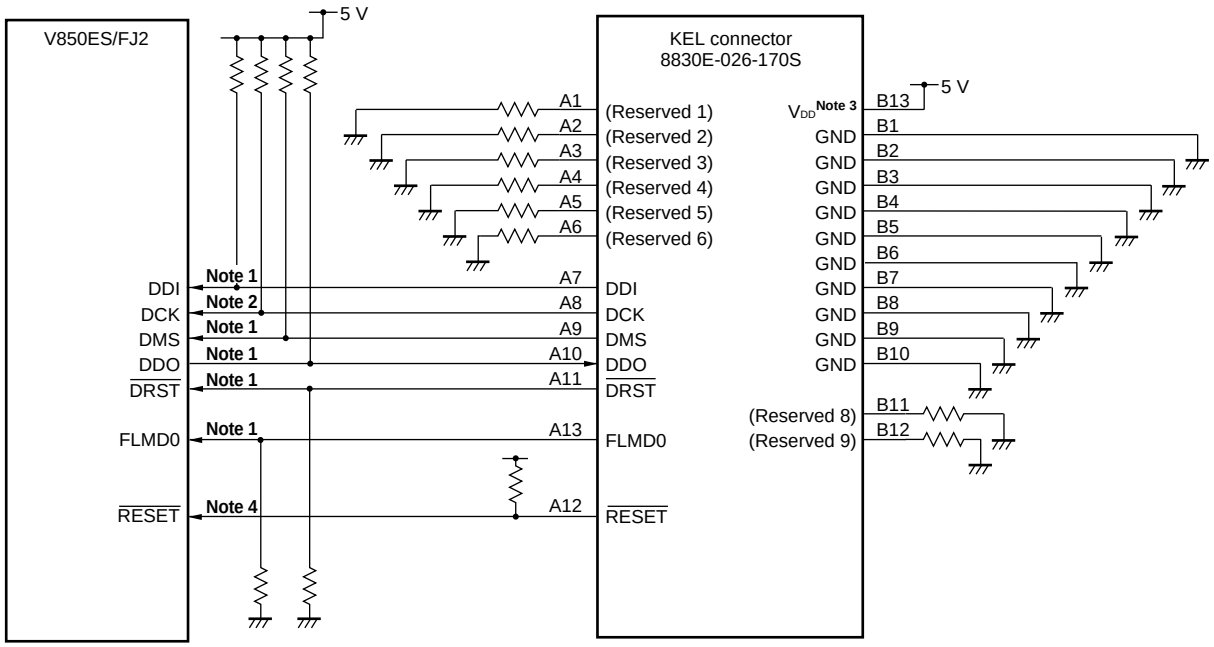
2. The pattern of the target board must satisfy the following conditions.

- **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 27-5. 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 27-5 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.

27.8 Cautions

(1) If a reset signal is input (from the target system or a reset signal from an internal reset source) during RUN (program execution), the break function may malfunction.

(2) Even if the reset signal is masked by the mask function, the I/O buffer (port pin) may be reset if a reset signal is input from a pin.

(3) 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.

(4) Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by DMA or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.

(5) The POC reset operation cannot be emulated.

(6) When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator due to a break, etc., the watchdog timer 2 does not stop and a reset or non-maskable interrupt occurs. When a reset occurs, the debugger hangs up.

(a) The main clock or subclock is used as the source clock for watchdog timer 2.
(b) The internal oscillation clock is stopped (RCM.RSTOP bit = 1).

To avoid this, perform either of the following.

- When an emulator is used, the internal oscillation clock is used as the source clock.
- When an emulator is used, disable the internal oscillator oscillation.

(7) When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator due to a break, etc., TMM does not stop even if the peripheral break function is set to “Break”.

(a) Either the INTWT, internal oscillation clock (f_R/8), or subclock are selected as the TMM source clock.
(b) The main clock is stopped.

To avoid this, perform either of the following.

- When an emulator is used, the main clock (f_{XX}, f_{XX}/2, f_{XX}/4, f_{XX}/64, f_{XX}/512) is used as the source clock.
- When an emulator is used, disable the main clock oscillation.

(8) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.

(9) When break command is based, and application software accesses for UARTA/CSIB/AFCAN peripheral I/O register, to restart without reset, CSIB, UARTA and AFCAN that may be not correct operation.

(10) 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).

APPENDIX A REGISTER INDEX

(1/15)

Symbol	Function Register Name	Unit	Page
ADA0CR0	A/D conversion result register 0	ADC	469
ADA0CR0H	A/D conversion result register 0H	ADC	469
ADA0CR1	A/D conversion result register 1	ADC	469
ADA0CR10	A/D conversion result register 10	ADC	469
ADA0CR10H	A/D conversion result register 10H	ADC	469
ADA0CR11	A/D conversion result register 11	ADC	469
ADA0CR11H	A/D conversion result register 11H	ADC	469
ADA0CR12	A/D conversion result register 12	ADC	469
ADA0CR12H	A/D conversion result register 12H	ADC	469
ADA0CR13	A/D conversion result register 13	ADC	469
ADA0CR13H	A/D conversion result register 13H	ADC	468
ADA0CR14	A/D conversion result register 14	ADC	468
ADA0CR14H	A/D conversion result register 14H	ADC	469
ADA0CR15	A/D conversion result register 15	ADC	469
ADA0CR15H	A/D conversion result register 15H	ADC	469
ADA0CR16	A/D conversion result register 16	ADC	469
ADA0CR16H	A/D conversion result register 16H	ADC	469
ADA0CR17	A/D conversion result register 17	ADC	469
ADA0CR17H	A/D conversion result register 17H	ADC	469
ADA0CR18	A/D conversion result register 18	ADC	469
ADA0CR18H	A/D conversion result register 18H	ADC	469
ADA0CR19	A/D conversion result register 19	ADC	469
ADA0CR19H	A/D conversion result register 19H	ADC	469
ADA0CR1H	A/D conversion result register 1H	ADC	469
ADA0CR2	A/D conversion result register 2	ADC	469
ADA0CR20	A/D conversion result register 20	ADC	469
ADA0CR20H	A/D conversion result register 20H	ADC	469
ADA0CR21	A/D conversion result register 21	ADC	469
ADA0CR21H	A/D conversion result register 21H	ADC	469
ADA0CR22	A/D conversion result register 22	ADC	469
ADA0CR22H	A/D conversion result register 22H	ADC	469
ADA0CR23	A/D conversion result register 23	ADC	469
ADA0CR23H	A/D conversion result register 23H	ADC	469
ADA0CR2H	A/D conversion result register 2H	ADC	469
ADA0CR3	A/D conversion result register 3	ADC	469
ADA0CR3H	A/D conversion result register 3H	ADC	469
ADA0CR4	A/D conversion result register 4	ADC	469
ADA0CR4H	A/D conversion result register 4H	ADC	469
ADA0CR5	A/D conversion result register 5	ADC	469

(2/15)

Symbol	Function Register Name	Unit	Page
ADA0CR5H	A/D conversion result register 5H	ADC	469
ADA0CR6	A/D conversion result register 6	ADC	469
ADA0CR6H	A/D conversion result register 6H	ADC	469
ADA0CR7	A/D conversion result register 7	ADC	469
ADA0CR7H	A/D conversion result register 7H	ADC	469
ADA0CR8	A/D conversion result register 8	ADC	469
ADA0CR8H	A/D conversion result register 8H	ADC	469
ADA0CR9	A/D conversion result register 9	ADC	469
ADA0CR9H	A/D conversion result register 9H	ADC	469
ADA0M0	A/D converter mode register 0	ADC	465
ADA0M1	A/D converter mode register 1	ADC	466
ADA0M2	A/D converter mode register 2	ADC	467
ADA0PFM	Power-fail comparison mode register	ADC	472
ADA0PFT	Power-fail comparison threshold value register	ADC	472
ADA0S	A/D converter channel specification register 0	ADC	468
AWC	Address wait control register	BCU	302
BCC	Bus cycle control register	BCU	303
BPC	Peripheral I/O area select control register	CPU	176
BSC	Bus size configuration register bus	BCU	293
C0BRP	CAN0 module bit rate prescaler register	CAN	678
C0BTR	CAN0 module bit rate register	CAN	679
C0CTRL	CAN0 module control register	CAN	668
C0ERC	CAN0 module error counter register	CAN	674
C0GMABT	CAN0 global block transmission control register	CAN	663
C0GMABTD	CAN0 global block transmission delay setting register	CAN	665
C0GMCS	CAN0 global clock select register	CAN	662
C0GMCTRL	CAN0 global control register	CAN	660
C0IE	CAN0 module interrupt enable register	CAN	675
C0INFO	CAN0 module information register	CAN	673
C0INTS	CAN0 module interrupt status register	CAN	677
C0LEC	CAN0 module last error information register	CAN	672
C0LIPT	CAN0 module last in-pointer register	CAN	681
C0LOPT	CAN0 module last out-pointer register	CAN	683
C0MASK1H	CAN0 module mask 1 register H	CAN	666
C0MASK1L	CAN0 module mask 1 register L	CAN	666
C0MASK2H	CAN0 module mask 2 register H	CAN	666
C0MASK2L	CAN0 module mask 2 register L	CAN	666
C0MASK3H	CAN0 module mask 3 register H	CAN	666
C0MASK3L	CAN0 module mask 3 register L	CAN	666
C0MASK4H	CAN0 module mask 4 register H	CAN	666
C0MASK4L	CAN0 module mask 4 register L	CAN	665
C0MCONFm	CAN0 message configuration register m	CAN	690
C0MCTRLm	CAN0 message control register m	CAN	692

(3/15)

Symbol	Function Register Name	Unit	Page
C0MDATA01m	CAN0 message data byte 01 register m	CAN	687
C0MDATA0m	CAN0 message data byte 0 register m	CAN	687
C0MDATA1m	CAN0 message data byte 1 register m	CAN	687
C0MDATA23m	CAN0 message data byte 23 register m	CAN	687
C0MDATA2m	CAN0 message data byte 2 register m	CAN	687
C0MDATA3m	CAN0 message data byte 3 register m	CAN	687
C0MDATA45m	CAN0 message data byte 45 register m	CAN	687
C0MDATA4m	CAN0 message data byte 4 register m	CAN	687
C0MDATA5m	CAN0 message data byte 5 register m	CAN	687
C0MDATA67m	CAN0 message data byte 67 register m	CAN	687
C0MDATA6m	CAN0 message data byte 6 register m	CAN	687
C0MDATA7m	CAN0 message data byte 7 register m	CAN	687
C0MDLCm	CAN0 message data length code register m	CAN	689
C0MIDHm	CAN0 message ID register Hm	CAN	691
C0MIDLm	CAN0 message ID register Lm	CAN	691
C0RGPT	CAN0 module receive history list register	CAN	682
C0TGPT	CAN0 module transmit history list register	CAN	684
C0TS	CAN0 module time stamp register	CAN	685
C1BRP	CAN1 module bit rate prescaler register	CAN	678
C1BTR	CAN1 module bit rate register	CAN	679
C1CTRL	CAN1 module control register	CAN	668
C1ERC	CAN1 module error counter register	CAN	674
C1GMABT	CAN1 global block transmission control register	CAN	663
C1GMABTD	CAN1 global block transmission delay setting register	CAN	665
C1GMCS	CAN1 global clock select register	CAN	662
C1GMCTRL	CAN1 global control register	CAN	660
C1IE	CAN1 module interrupt enable register	CAN	675
C1INFO	CAN1 module information register	CAN	673
C1INTS	CAN1 module interrupt status register	CAN	677
C1LEC	CAN1 module last error information register	CAN	672
C1LIPT	CAN1 module last in-pointer register	CAN	681
C1LOPT	CAN1 module last out-pointer register	CAN	683
C1MASK1H	CAN1 module mask 1 register H	CAN	666
C1MASK1L	CAN1 module mask 1 register L	CAN	666
C1MASK2H	CAN1 module mask 2 register H	CAN	666
C1MASK2L	CAN1 module mask 2 register L	CAN	666
C1MASK3H	CAN1 module mask 3 register H	CAN	666
C1MASK3L	CAN1 module mask 3 register L	CAN	666
C1MASK4H	CAN1 module mask 4 register H	CAN	665
C1MASK4L	CAN1 module mask 4 register L	CAN	666
C1MCONFm	CAN1 message configuration register m	CAN	690
C1MCTRLm	CAN1 message control register m	CAN	692

(4/15)

Symbol	Function Register Name	Unit	Page
C1MDATA01m	CAN1 message data byte 01 register m	CAN	687
C1MDATA0m	CAN1 message data byte 0 register m	CAN	687
C1MDATA1m	CAN1 message data byte 1 register m	CAN	687
C1MDATA23m	CAN1 message data byte 23 register m	CAN	687
C1MDATA2m	CAN1 message data byte 2 register m	CAN	687
C1MDATA3m	CAN1 message data byte 3 register m	CAN	687
C1MDATA45m	CAN1 message data byte 45 register m	CAN	687
C1MDATA4m	CAN1 message data byte 4 register m	CAN	687
C1MDATA5m	CAN1 message data byte 5 register m	CAN	687
C1MDATA67m	CAN1 message data byte 67 register m	CAN	687
C1MDATA6m	CAN1 message data byte 6 register m	CAN	687
C1MDATA7m	CAN1 message data byte 7 register m	CAN	687
C1MDLCm	CAN1 message data length code register m	CAN	689
C1MIDHm	CAN1 message ID register Hm	CAN	691
C1MIDLm	CAN1 message ID register Lm	CAN	691
C1RGPT	CAN1 module receive history list register	CAN	682
C1TGPT	CAN1 module transmit history list register	CAN	684
C1TS	CAN1 module time stamp register	CAN	685
C2BRP	CAN2 module bit rate prescaler register	CAN	678
C2BTR	CAN2 module bit rate register	CAN	679
C2CTRL	CAN2 module control register	CAN	668
C2ERC	CAN2 module error counter register	CAN	674
C2ERRIC	Interrupt control register	CAN	686
C2GMABT	CAN2 global block transmission control register	CAN	663
C2GMABTD	CAN2 global block transmission delay setting register	CAN	665
C2GMCS	CAN2 global clock select register	CAN	662
C2GMCTRL	CAN2 global control register	CAN	660
C2IE	CAN2 module interrupt enable register	CAN	675
C2INFO	CAN2 module information register	CAN	673
C2INTS	CAN2 module interrupt status register	CAN	677
C2LEC	CAN2 module last error information register	CAN	672
C2LIPT	CAN2 module last in-pointer register	CAN	681
C2LOPT	CAN2 module last out-pointer register	CAN	683
C2MASK1H	CAN2 module mask 1 register H	CAN	666
C2MASK1L	CAN2 module mask 1 register L	CAN	666
C2MASK2H	CAN2 module mask 2 register H	CAN	666
C2MASK2L	CAN2 module mask 2 register L	CAN	666
C2MASK3H	CAN2 module mask 3 register H	CAN	666
C2MASK3L	CAN2 module mask 3 register L	CAN	666
C2MASK4H	CAN2 module mask 4 register H	CAN	666
C2MASK4L	CAN2 module mask 4 register L	CAN	666
C2MCONFm	CAN2 message configuration register m	CAN	690
C2MCTRLm	CAN2 message control register m	CAN	692

(5/15)

Symbol	Function Register Name	Unit	Page
C2MDATA01m	CAN2 message data byte 01 register m	CAN	687
C2MDATA0m	CAN2 message data byte 0 register m	CAN	687
C2MDATA1m	CAN2 message data byte 1 register m	CAN	687
C2MDATA23m	CAN2 message data byte 23 register m	CAN	687
C2MDATA2m	CAN2 message data byte 2 register m	CAN	687
C2MDATA3m	CAN2 message data byte 3 register m	CAN	687
C2MDATA45m	CAN2 message data byte 45 register m	CAN	687
C2MDATA4m	CAN2 message data byte 4 register m	CAN	687
C2MDATA5m	CAN2 message data byte 5 register m	CAN	687
C2MDATA67m	CAN2 message data byte 67 register m	CAN	687
C2MDATA6m	CAN2 message data byte 6 register m	CAN	687
C2MDATA7m	CAN2 message data byte 7 register m	CAN	687
C2MDLCm	CAN2 message data length code register m	CAN	689
C2MIDHm	CAN2 message ID register Hm	CAN	691
C2MIDLm	CAN2 message ID register Lm	CAN	691
C2RGPT	CAN2 module receive history list register	CAN	682
C2TGPT	CAN2 module transmit history list register	CAN	684
C2TS	CAN2 module time stamp register	CAN	685
C3BRP	CAN3 module bit rate prescaler register	CAN	678
C3BTR	CAN3 module bit rate register	CAN	679
C3CTRL	CAN3 module control register	CAN	668
C3ERC	CAN3 module error counter register	CAN	674
C3GMABT	CAN3 global block transmission control register	CAN	663
C3GMABTD	CAN3 global block transmission delay setting register	CAN	665
C3GMCS	CAN3 global clock select register	CAN	662
C3GMCTRL	CAN3 global control register	CAN	660
C3IE	CAN3 module interrupt enable register	CAN	675
C3INFO	CAN3 module information register	CAN	673
C3INTS	CAN3 module interrupt status register	CAN	677
C3LEC	CAN3 module last error information register	CAN	672
C3LIPT	CAN3 module last in-pointer register	CAN	681
C3LOPT	CAN3 module last out-pointer register	CAN	683
C3MASK1H	CAN3 module mask 1 register H	CAN	666
C3MASK1L	CAN3 module mask 1 register L	CAN	666
C3MASK2H	CAN3 module mask 2 register H	CAN	666
C3MASK2L	CAN3 module mask 2 register L	CAN	666
C3MASK3H	CAN3 module mask 3 register H	CAN	666
C3MASK3L	CAN3 module mask 3 register L	CAN	666
C3MASK4H	CAN3 module mask 4 register H	CAN	666
C3MASK4L	CAN3 module mask 4 register L	CAN	666
C3MCONFm	CAN3 message configuration register m	CAN	690
C3MCTRLm	CAN3 message control register m	CAN	692

(6/15)

Symbol	Function Register Name	Unit	Page
C3MDATA01m	CAN3 message data byte 01 register m	CAN	687
C3MDATA0m	CAN3 message data byte 0 register m	CAN	687
C3MDATA1m	CAN3 message data byte 1 register m	CAN	687
C3MDATA23m	CAN3 message data byte 23 register m	CAN	687
C3MDATA2m	CAN3 message data byte 2 register m	CAN	687
C3MDATA3m	CAN3 message data byte 3 register m	CAN	687
C3MDATA45m	CAN3 message data byte 45 register m	CAN	687
C3MDATA4m	CAN3 message data byte 4 register m	CAN	687
C3MDATA5m	CAN3 message data byte 5 register m	CAN	687
C3MDATA67m	CAN3 message data byte 67 register m	CAN	687
C3MDATA6m	CAN3 message data byte 6 register m	CAN	687
C3MDATA7m	CAN3 message data byte 7 register m	CAN	687
C3MDLCm	CAN3 message data length code register m	CAN	689
C3MIDHm	CAN3 message ID register Hm	CAN	691
C3MIDLm	CAN3 message ID register Lm	CAN	691
C3RGPT	CAN3 module receive history list register	CAN	682
C3TGPT	CAN3 module transmit history list register	CAN	684
C3TS	CAN3 module time stamp register	CAN	685
CB0CTL0	CSIB0 control register 0	CSI	533
CB0CTL1	CSIB0 control register 1	CSI	535
CB0CTL2	CSIB0 control register 2	CSI	536
CB0RX	CSIB0 receive data register	CSI	532
CB0RXL	CSIB0 receive data register L	CSI	532
CB0STR	CSIB0 status register	CSI	537
CB0TX	CSIB0 transmit data register	CSI	532
CB0TXL	CSIB0 transmit data register L	CSI	532
CB1CTL0	CSIB1 control register 0	CSI	533
CB1CTL1	CSIB1 control register 1	CSI	535
CB1CTL2	CSIB1 control register 2	CSI	536
CB1RX	CSIB1 receive data register	CSI	532
CB1RXL	CSIB1 receive data register L	CSI	532
CB1STR	CSIB1 status register	CSI	537
CB1TX	CSIB1 transmit data register	CSI	532
CB1TXL	CSIB1 transmit data register L	CSI	532
CB2CTL0	CSIB2 control register 0	CSI	533
CB2CTL1	CSIB2 control register 1	CSI	535
CB2CTL2	CSIB2 control register 2	CSI	536
CB2RX	CSIB2 receive data register	CSI	532
CB2RXL	CSIB2 receive data register L	CSI	532
CB2STR	CSIB2 status register	CSI	537
CB2TX	CSIB2 transmit data register	CSI	532
CB2TXL	CSIB2 transmit data register L	CSI	532
CCLS	CPU operating clock status register	BCU	318
CLM	Clock monitor mode register	CM	860

(7/15)

Symbol	Function Register Name	Unit	Page
DADC0	DMA addressing control register 0	DMA	760
DADC1	DMA addressing control register 1	DMA	760
DADC2	DMA addressing control register 2	DMA	760
DADC3	DMA addressing control register 3	DMA	760
DBC0	DMA transfer count register 0	DMA	759
DBC1	DMA transfer count register 1	DMA	759
DBC2	DMA transfer count register 2	DMA	759
DBC3	DMA transfer count register 3	DMA	759
DCHC0	DMA channel control register 0	DMA	759
DCHC1	DMA channel control register 1	DMA	761
DCHC2	DMA channel control register 2	DMA	761
DCHC3	DMA channel control register 3	DMA	761
DDA0H	DMA destination address register 0H	DMA	758
DDA0L	DMA destination address register 0L	DMA	758
DDA1H	DMA destination address register 1L	DMA	758
DDA1L	DMA destination address register 1H	DMA	758
DDA2H	DMA destination address register 2H	DMA	758
DDA2L	DMA destination address register 2L	DMA	758
DDA3H	DMA destination address register 3H	DMA	758
DDA3L	DMA destination address register 3L	DMA	758
DSA0H	DMA source address register 0H	DMA	757
DSA0L	DMA source address register 0L	DMA	757
DSA1H	DMA source address register 1H	DMA	757
DSA1L	DMA source address register 1L	DMA	757
DSA2H	DMA source address register 2H	DMA	757
DSA2L	DMA source address register 2L	DMA	757
DSA3H	DMA source address register 3H	DMA	757
DSA3L	DMA source address register 3L	DMA	757
DTFR0	DMA trigger source register 0	DMA	763
DTFR1	DMA trigger source register 1	DMA	763
DTFR2	DMA trigger source register 2	DMA	763
DTFR3	DMA trigger source register 3	DMA	763
DWC0	Data wait control register 0	BCU	300
IMR0	Interrupt mask register 0	INTC	804
IMR0H	Interrupt mask register 0H	INTC	804
IMR0L	Interrupt mask register 0L	INTC	804
IMR1	Interrupt mask register 1	INTC	804
IMR1H	Interrupt mask register 1H	INTC	804
IMR1L	Interrupt mask register 1L	INTC	804
IMR2	Interrupt mask register 2	INTC	804
IMR2H	Interrupt mask register 2H	INTC	804
IMR2L	Interrupt mask register 2L	INTC	804
IMR3	Interrupt mask register 3	INTC	804
IMR3H	Interrupt mask register 3H	INTC	804
IMR3L	Interrupt mask register 3L	INTC	804

(8/15)

Symbol	Function Register Name	Unit	Page
IMR4	Interrupt mask register 4	INTC	804
IMR4H	Interrupt mask register 4H	INTC	804
IMR4L	Interrupt mask register 4L	INTC	804
IMR5L	Interrupt mask register 5L	INTC	804
INTF0	External interrupt falling edge specification register 0	INTC	808
INTF1	External interrupt falling edge specification register 1	INTC	810
INTF3	External interrupt falling edge specification register 3	INTC	812
INTF3H	External interrupt falling edge specification register 3H	INTC	812
INTF3L	External interrupt falling edge specification register 3L	INTC	812
INTF6L	External interrupt falling edge specification register 6L	INTC	814
INTF8	External interrupt falling edge specification register 8	INTC	815
INTF9H	External interrupt falling edge specification register 9H	INTC	816
INTR0	External interrupt rising edge specification register 0	INTC	809
INTR1	External interrupt rising edge specification register 1	INTC	811
INTR3	External interrupt rising edge specification register 3	INTC	813
INTR3H	External interrupt rising edge specification register 3H	INTC	813
INTR3L	External interrupt rising edge specification register 3L	INTC	813
INTR6L	External interrupt rising edge specification register 6L	INTC	814
INTR8	External interrupt rising edge specification register 8	INTC	815
INTR9H	External interrupt rising edge specification register 9H	INTC	816
ISPR	In-service priority register	INTC	806
KRIC	Interrupt control register	INTC	686
KRM	Key return mode register	KR	828
LOCKR	Lock register	BCU	321
LVIIC	Interrupt control register	INTC	686
LVIM	Low-voltage detection register	LVD	867
LVIS	Low-voltage detection level select register	LVD	868
NFC	Noise elimination control register	INTC	817
OCDM	On-chip debug mode register	DEBUG	903
OSTS	Oscillation stabilization time select register	WDT	457
P0	Port 0	PORT	196
P00NFC	TIP00 noise eliminator control register	TIMER	338
P01NFC	TIP01 noise eliminator control register	TIMER	338
P1	Port 1	PORT	202
P10NFC	TIP10 noise eliminator control register	TIMER	338
P11NFC	TIP11 noise eliminator control register	TIMER	338
P12	Port 12	PORT	258
P20NFC	TIP20 noise eliminator control register	TIMER	338
P21NFC	TIP21 noise eliminator control register	TIMER	338
P3	Port 3	PORT	207
P30NFC	TIP30 noise eliminator control register	TIMER	338
P31NFC	TIP31 noise eliminator control register	TIMER	338
P3H	Port 3H	PORT	207
P3L	Port 3L	PORT	207

(9/15)

Symbol	Function Register Name	Unit	Page
P4	Port 4	PORT	217
P5	Port 5	PORT	221
P6	Port 6	PORT	228
P6H	Port 6H	PORT	228
P6L	Port 6L	PORT	228
P7	Port 7	PORT	237
P7H	Port 7H	PORT	237
P7L	Port 7L	PORT	237
P8	Port 8	PORT	240
P9	Port 9	PORT	246
P9H	Port 9H	PORT	246
P9L	Port 9L	PORT	246
PCC	Processor clock control register	BCU	316
PCD	Port CD	PORT	260
PCLM	Programmable clock mode register	BCU	323
PCM	Port CM	PORT	262
PCS	Port CS	PORT	266
PCT	Port CT	PORT	270
PDL	Port DL	PORT	275
PDLH	Port DLH	PORT	275
PDLL	Port DLL	PORT	275
PEMU1	Peripheral emulation register 1	LVD	869
PFC0	Port function control register 0	PORT	198
PFC3L	Port function control register 3L	PORT	211
PFC5	Port function control register 5	PORT	223
PFC6	Port function control register 6	PORT	232
PFC6H	Port function control register 6H	PORT	232
PFC6L	Port function control register 6L	PORT	232
PFC9	Port function control register 9	PORT	251
PFC9H	Port function control register 9H	PORT	251
PFC9L	Port function control register 9L	PORT	251
PFCE3L	Port function control expansion register 3L	PORT	211
PFCE5	Port function control expansion register 5	PORT	223
PFCE9	Port function control expansion register 9	PORT	252
PFCE9H	Port function control expansion register 9H	PORT	252
PFCE9L	Port function control expansion register 9L	PORT	252
PIC0	Interrupt control register	INTC	783
PIC1	Interrupt control register	INTC	783
PIC10	Interrupt control register	INTC	783
PIC11	Interrupt control register	INTC	783
PIC12	Interrupt control register	INTC	783
PIC13	Interrupt control register	INTC	783
PIC14	Interrupt control register	INTC	783
PIC2	Interrupt control register	INTC	783

(10/15)

Symbol	Function Register Name	Unit	Page
PIC3	Interrupt control register	INTC	783
PIC4	Interrupt control register	INTC	783
PIC5	Interrupt control register	INTC	783
PIC6	Interrupt control register	INTC	783
PIC7	Interrupt control register	INTC	783
PIC8	Interrupt control register	INTC	783
PIC9	Interrupt control register	INTC	783
PLLCTL	PLL control register	BCU	320
PLLS	PLL lockup time specification register	BCU	322
PM0	Port mode register 0	PORT	196
PM1	Port mode register 1	PORT	202
PM12	Port mode register 12	PORT	258
PM3	Port mode register 3	PORT	208
PM3H	Port mode register 3H	PORT	208
PM3L	Port mode register 3L	PORT	208
PM4	Port mode register 4	PORT	217
PM5	Port mode register 5	PORT	221
PM6	Port mode register 6	PORT	229
PM6H	Port mode register 6H	PORT	229
PM6L	Port mode register 6L	PORT	229
PM7	Port mode register 7	PORT	238
PM7H	Port mode register 7H	PORT	238
PM7L	Port mode register 7L	PORT	238
PM8	Port mode register 8	PORT	240
PM9	Port mode register 9	PORT	247
PM9H	Port mode register 9H	PORT	247
PM9L	Port mode register 9L	PORT	247
PMC0	Port mode control register 0	PORT	197
PMC1	Port mode control register 1	PORT	203
PMC3	Port mode control register 3	PORT	209
PMC3H	Port mode control register 3 H	PORT	209
PMC3L	Port mode control register 3 L	PORT	209
PMC4	Port mode control register 4	PORT	218
PMC5	Port mode control register 5	PORT	222
PMC6	Port mode control register 6	PORT	230
PMC6H	Port mode control register 6 H	PORT	230
PMC6L	Port mode control register 6 L	PORT	230
PMC8	Port mode control register 8	PORT	241
PMC9	Port mode control register 9	PORT	248
PMC9H	Port mode control register 9 H	PORT	248
PMC9L	Port mode control register 9 L	PORT	248
PMCCM	Port mode control register CM	PORT	264
PMCCS	Port mode control register CS	PORT	268
PMCCT	Port mode control register CT	PORT	272

(11/15)

Symbol	Function Register Name	Unit	Page
PMCD	Port mode register CD	PORT	260
PMCDL	Port mode control register DL	PORT	277
PMCDLH	Port mode control register DLH	PORT	277
PMCDLL	Port mode control register DLL	PORT	277
PMCM	Port mode register CM	PORT	263
PMCS	Port mode register CS	PORT	267
PMCT	Port mode register CT	PORT	271
PMDL	Port mode register DL	PORT	276
PMDLH	Port mode register DLH	PORT	276
PMDLL	Port mode register DLL	PORT	276
PRCMD	Command register	CPU	179
PRSCM0	Prescaler compare register 0	WT	455
PRSM0	Prescaler mode register 0	WT	454
PSC	Power save control register	Standby	849
PSMR	Power save mode register	Standby	850
PU0	Pull-up resistor option register 0	PORT	198
PU1	Pull-up resistor option register 1	PORT	203
PU3	Pull-up resistor option register 3	PORT	213
PU3H	Pull-up resistor option register 3H	PORT	213
PU3L	Pull-up resistor option register 3L	PORT	213
PU4	Pull-up resistor option register 4	PORT	218
PU5	Pull-up resistor option register 5	PORT	225
PU6	Pull-up resistor option register 6	PORT	234
PU6H	Pull-up resistor option register 6H	PORT	234
PU6L	Pull-up resistor option register 6L	PORT	234
PU8	Pull-up resistor option register 8	PORT	242
PU9	Pull-up resistor option register 9	PORT	255
PU9H	Pull-up resistor option register 9H	PORT	255
PU9L	Pull-up resistor option register 9L	PORT	255
Q00NFC	TIQ00 noise eliminator control register	TIMER	397
Q01NFC	TIQ01 noise eliminator control register	TIMER	397
Q02NFC	TIQ02 noise eliminator control register	TIMER	397
Q03NFC	TIQ03 noise eliminator control register	TIMER	397
Q10NFC	TIQ10 noise eliminator control register	TIMER	397
Q11NFC	TIQ11 noise eliminator control register	TIMER	397
Q12NFC	TIQ12 noise eliminator control register	TIMER	397
Q13NFC	TIQ13 noise eliminator control register	TIMER	397
Q20NFC	TIQ20 noise eliminator control register	TIMER	397
Q21NFC	TIQ21 noise eliminator control register	TIMER	397
Q22NFC	TIQ22 noise eliminator control register	TIMER	397
Q23NFC	TIQ23 noise eliminator control register	TIMER	397
RAMS	Internal RAM data status register	LVD	868
RCM	Ring OSC mode register	BCU	318
RESF	Reset source flag register	RESET	853

(12/15)

Symbol	Function Register Name	Unit	Page
SAR	Successive approximation register	ADC	463
SELCNT0	Selector operation control register 0	TIMER	374
SELCNT1	Selector operation control register 1	TIMER	376
SYS	System status register	CPU	180
TM0CMP0	TMM0 compare register 0	TIMER	442
TM0CTL0	TMM0 control register 0	TIMER	443
TM0EQIC0	Interrupt control register	INTC	783
TP0CCIC0	Interrupt control register	INTC	783
TP0CCIC1	Interrupt control register	INTC	783
TP0CCR0	TMP0 capture/compare register 0	TIMER	327
TP0CCR1	TMP0 capture/compare register 1	TIMER	328
TP0CNT	TMP0 counter read buffer register	TIMER	238
TP0CTL0	TMP0 control register 0	TIMER	330
TP0CTL1	TMP0 control register 1	TIMER	332
TP0IOC0	TMP0 I/O control register 0	TIMER	334
TP0IOC1	TMP0 I/O control register 1	TIMER	335
TP0IOC2	TMP0 I/O control register 2	TIMER	336
TP0OPT0	TMP0 option register	TIMER	337
TP0OVIC	Interrupt control register	INTC	783
TP1CCIC0	Interrupt control register	INTC	783
TP1CCIC1	Interrupt control register	INTC	783
TP1CCR0	TMP1 capture/compare register 0	TIMER	327
TP1CCR1	TMP1 capture/compare register 1	TIMER	328
TP1CNT	TMP1 counter read buffer register	TIMER	329
TP1CTL0	TMP1 control register 0	TIMER	330
TP1CTL1	TMP1 control register 1	TIMER	332
TP1IOC0	TMP1 I/O control register 0	TIMER	334
TP1IOC1	TMP1 I/O control register 1	TIMER	335
TP1IOC2	TMP1 I/O control register 2	TIMER	336
TP1OPT0	TMP1 option register	TIMER	337
TP1OVIC	Interrupt control register	INTC	783
TP2CCIC0	Interrupt control register	INTC	783
TP2CCIC1	Interrupt control register	INTC	783
TP2CCR0	TMP2 capture/compare register 0	TIMER	327
TP2CCR1	TMP2 capture/compare register 1	TIMER	328
TP2CNT	TMP2 counter read buffer register	TIMER	329
TP2CTL0	TMP2 control register 0	TIMER	330
TP2CTL1	TMP2 control register 1	TIMER	332
TP2IOC0	TMP2 I/O control register 0	TIMER	334
TP2IOC1	TMP2 I/O control register 1	TIMER	335
TP2IOC2	TMP2 I/O control register 2	TIMER	336
TP2OPT0	TMP2 option register	TIMER	337
TP2OVIC	Interrupt control register	INTC	783
TP3CCIC0	Interrupt control register	INTC	783

(13/15)

Symbol	Function Register Name	Unit	Page
TP3CCIC1	Interrupt control register	INTC	783
TP3CCR0	TMP3 capture/compare register 0	TIMER	327
TP3CCR1	TMP3 capture/compare register 1	TIMER	328
TP3CNT	TMP3 counter read buffer register	TIMER	329
TP3CTL0	TMP3 control register 0	TIMER	330
TP3CTL1	TMP3 control register 1	TIMER	332
TP3IOC0	TMP3 I/O control register 0	TIMER	334
TP3IOC1	TMP3 I/O control register 1	TIMER	335
TP3IOC2	TMP3 I/O control register 2	TIMER	336
TP3OPT0	TMP3 option register	TIMER	337
TP3OVIC	Interrupt control register	INTC	783
TQ0CCIC0	Interrupt control register	INTC	783
TQ0CCIC1	Interrupt control register	INTC	783
TQ0CCIC2	Interrupt control register	INTC	783
TQ0CCIC3	Interrupt control register	INTC	783
TQ0CCR0	TMQ1 capture/compare register 0	TIMER	383
TQ0CCR1	TMQ1 capture/compare register 1	TIMER	384
TQ0CCR2	TMQ1 capture/compare register 2	TIMER	385
TQ0CCR3	TMQ1 capture/compare register 3	TIMER	386
TQ0CNT	TMQ0 counter read buffer register	TIMER	387
TQ0CTL0	TMQ0 control register 0	TIMER	388
TQ0CTL1	TMQ0 control register 1	TIMER	390
TQ0IOC0	TMQ0 I/O control register 0	TIMER	392
TQ0IOC1	TMQ0 I/O control register 1	TIMER	393
TQ0IOC2	TMQ0 I/O control register 2	TIMER	395
TQ0OPT0	TMQ0 option register 0	TIMER	396
TQ0OVIC	Interrupt control register	INTC	783
TQ1CCIC0	Interrupt control register	INTC	783
TQ1CCIC1	Interrupt control register	INTC	783
TQ1CCIC2	Interrupt control register	INTC	783
TQ1CCIC3	Interrupt control register	INTC	783
TQ1CCR0	TMQ0 capture/compare register 0	TIMER	383
TQ1CCR1	TMQ0 capture/compare register 1	TIMER	384
TQ1CCR2	TMQ0 capture/compare register 2	TIMER	385
TQ1CCR3	TMQ0 capture/compare register 3	TIMER	386
TQ1CNT	TMQ1 counter read buffer register	TIMER	387
TQ1CTL0	TMQ1 control register 0	TIMER	388
TQ1CTL1	TMQ1 control register 1	TIMER	390
TQ1IOC0	TMQ1 I/O control register 0	TIMER	392
TQ1IOC1	TMQ1 I/O control register 1	TIMER	393
TQ1IOC2	TMQ1 I/O control register 2	TIMER	395
TQ1OPT0	TMQ1 timer option register 0	TIMER	396
TQ1OVIC	Interrupt control register	INTC	783
TQ2CCIC0	Interrupt control register	INTC	783

(14/15)

Symbol	Function Register Name	Unit	Page
TQ2CCIC1	Interrupt control register	INTC	783
TQ2CCIC2	Interrupt control register	INTC	783
TQ2CCIC3	Interrupt control register	INTC	783
TQ2CCR0	TMQ2 capture/compare register 0	TIMER	383
TQ2CCR1	TMQ2 capture/compare register 1	TIMER	384
TQ2CCR2	TMQ2 capture/compare register 2	TIMER	385
TQ2CCR3	TMQ2 capture/compare register 3	TIMER	386
TQ2CNT	TMQ2 counter read buffer register	TIMER	387
TQ2CTL0	TMQ2 control register 0	TIMER	388
TQ2CTL1	TMQ2 control register 1	TIMER	390
TQ2IOC0	TMQ2 I/O control register 0	TIMER	392
TQ2IOC1	TMQ2 I/O control register 1	TIMER	393
TQ2IOC2	TMQ2 I/O control register 2	TIMER	395
TQ2OPT0	TMQ2 option register	TIMER	396
TQ2OVIC	Interrupt control register	INTC	783
UA0CTL0	UARTA0 control register 0	UART	500
UA0CTL1	UARTA0 control register 1	UART	502
UA0CTL2	UARTA0 control register 2	UART	503
UA0OPT0	UARTA0 option control register 0	UART	504
UA0RIC	Interrupt control register	INTC	783
UA0RX	UARTA0 receive data register	UART	507
UA0STR	UARTA0 status register	UART	505
UA0TIC	Interrupt control register	INTC	783
UA0TX	UARTA0 transmit data register	UART	507
UA1CTL0	UARTA1 control register 0	UART	500
UA1CTL1	UARTA1 control register 1	UART	502
UA1CTL2	UARTA1 control register 2	UART	503
UA1OPT0	UARTA1 option control register 0	UART	504
UA1RIC	Interrupt control register	INTC	783
UA1RX	UARTA1 receive data register	UART	507
UA1STR	UARTA1 status register	UART	505
UA1TIC	Interrupt control register	INTC	783
UA1TX	UARTA1 receive data register	UART	507
UA2CTL0	UARTA2 control register 0	UART	500
UA2CTL1	UARTA2 control register 1	UART	502
UA2CTL2	UARTA2 control register 2	UART	503
UA2OPT0	UARTA2 option control register 0	UART	504
UA2RIC	Interrupt control register	INTC	783
UA2RX	UARTA2 receive data register	UART	507
UA2STR	UARTA2 status register	UART	505
UA2TIC	Interrupt control register	INTC	783
UA2TX	UARTA2 transmit data register	UART	507
UA3CTL0	UARTA3 control register 0	UART	500
UA3CTL1	UARTA3 control register 1	UART	502

(15/15)

Symbol	Function Register Name	Unit	Page
UA3CTL2	UARTA3 control register 2	UART	503
UA3OPT0	UARTA3 option control register 0	UART	504
UA3RIC	Interrupt control register	INTC	783
UA3RX	UARTA3 receive data register	UART	507
UA3STR	UARTA3 status register	UART	505
UA3TIC	Interrupt control register	INTC	783
UA3TX	UARTA3 transmit data register	UART	507
VSWC	System wait control register	CPU	181
WDTE	Watchdog timer enable register	WDT	460
WDTM2	Watchdog timer mode register 2	WDT	458
WTIC	Interrupt control register	INTC	783
WTIIC	Interrupt control register	INTC	783
WTM	Watch timer operation mode register	WT	450

APPENDIX B INSTRUCTION SET LIST

B.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z	0 0 1 0	Z = 1	Zero
NZ	1 0 1 0	Z = 0	Not zero
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
S/N	0 1 0 0	S = 1	Negative
NS/P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	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 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2][Note 6]xGR[reg1][Note 6]		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2][Note 6]xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1][Note 6]ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR 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				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm[Note 15]	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr01010 1iiiii	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	rrrrr01000 1iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

B.3 Description of Operating Precautions

If a conflict occurs between the decode operation of the instruction (<2> in the examples mentioned below) immediately before the **sld** instruction (<3> in the examples) following a special instruction (<1> in the examples) and an interrupt request before execution of the special instruction is complete, the execution result of the special instruction may not be stored in a register as expected.
This situation may only occur when the same register is used as the destination register of the special instruction and the **sld** instruction, and when the register value is referenced by the instruction followed by the **sld** instruction.

Conditions under which the conflict occurs:
The situation may occur when all the following conditions (1) to (3) are satisfied.

(1) Either condition (I) or (II) is satisfied

Condition (I):
The same register is used as the destination register of a special instruction (see below) and the subsequent **sld** instruction and as the source register (reg1) of an instruction shown below followed by the **sld** instruction (See Example 1).

mov **reg**1,reg2	not **reg**1,reg2	satsubr **reg**1,reg2	satsub **reg**1,reg2
satadd **reg**1,reg2	or **reg**1,reg2	xor **reg**1,reg2	and **reg**1,reg2
tst **reg**1,reg2	subr **reg**1,reg2	sub **reg**1,reg2	add **reg**1,reg2
cmp **reg**1,reg2	mulh **reg**1,reg2		

Condition (II):
The same register is used as the destination register of a special instruction (see below) and the subsequent **sld** instruction and as the source register (reg2) of an instruction shown below followed by the **sld** instruction (See Examples 2 and 3).

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

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

(2) When the execution result of the special instruction (see above) has not been stored in the destination register before execution of the instruction (instruction of condition (I) or (II)) immediately before the **sld** instruction starts in the CPU pipeline.

(3) When the decode operation of the instruction (instruction of condition (I) or (II)) immediately before the **sld** instruction and interrupt request servicing conflict.

Examples of instruction sequences that may cause the conflict:

Example 1:

<1> ld.w [r11], **r10**
:
<2> mov **r10**, r28
<3> sld.w 0x28, r10

This situation occurs when the decode operation of **mov** (<2>) is done immediately before **sld** (<3>) and an interrupt request servicing conflict happens before the execution of the special instruction **ld** (<1>) is completed.

Example 2:

(1) ld.w [r11], **r10**
:
<2> cmp imm5, **r10**
<3> sld.w 0x28, r10
<4> bz label

This situation occurs when the decode operation of **comp** (<2>) is done immediately before **sld** (<3>) and an interrupt request servicing conflict happens before the execution of the special instruction **ld** (<1>) is completed. As a result, the compare result of **comp** becomes illegal, which may cause an illegal operation of the branch instruction **bz** (<4>).

Example 3:

<1> ld.w [r11], **r10**
:
<2> add imm5, **r10**
<3> sld.w 0x28, r10
<4> setf r16

This situation occurs when the decode operation of **add** (<2>) is done immediately before **sld** (<3>) and an interrupt request servicing conflict happens before the execution of the special instruction **ld** (<1>) is completed. As a result, the results of add and the flag become illegal, which may cause illegal operation of the **setf** (<4>).

Workaround

(1) Do not use the **sld** instruction (e. g. by avoiding code optimization that makes use of **sld**).

(2) If a code sequence as described above is used (a **sld** instruction following an instruction that can be executed in parallel), insert a **nop** instruction before the **sld** instruction.

(3) If a code sequence as described above is used (a **sld** instruction following an instruction that can be executed in parallel), exchange the order of the previous two instructions as long as the program algorithm is not disturbed:

Example:
1. (before implementing workaround)
ld.w [r11], r10

```
...
add r11, r12
mov r10, r28
sld.w 0x28, r10

2. (after implementing workaround)
ld.w [r11], r10
...
mov r10, r28
add r11, r12
sld.w 0x28, r10
```

(4) When assembler code is used:
Avoid the critical code sequences as described above.

Please regard this item as a usage restriction on the CPU function. A compiler that can automatically suppress the generation of the instruction sequence that may cause the bug will be provided.
Please consult an NEC Electronics sales representative or distributor for further details.

Support for system developed:

[Support for system already developed]
When the system has already been developed a judgment is necessary whether or not the restriction applies to the system.
Please consult an NEC Electronics sales representative or distributor for further details.