User's Manual

32

V850E2/Px4

User's Manual: Hardware

Renesas microcomputers
V850 Series

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

Renesas Electronics
www.renesas.com

Rev.1.00 Jul 2012

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “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 etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

(2012.4)

Notes for CMOS devices

(1) Voltage application wafeform 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 VIL (MAX) and VIH (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 VIL (MAX) and VIH (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 VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) Precaution against ESD:

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

(4) Status before initialization:

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

(5) Power ON/OFF sequence:

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

(6) Input of signal during power off state:

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

How to Use This Manual

Readers This manual is intended for users who wish to understand the functions of the V850E2/Px4 and design application systems using the following V850E2/Px4 microcontrollers:

Purpose This manual is intended to give users an understanding of the hardware functions of the V850E2/Px4 shown in the *Organization* below.

Organization This manual is divided into two parts: Hardware (this manual) and Architecture (V850E2M 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 overall functions of the V850E2/Px4.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See *V850E2M Architecture User's Manual* available separately.

All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

Section 1 Introduction 1

1.1 Overview 1
1.2 Characteristics 2
1.3 List of Functions 5
1.4 Field of Application 7
1.5 Information for Ordering 7
1.6 Pin Connection Diagram (Top View) 8
1.7 Configuration of Functional Blocks 23
1.7.1 Internal Block Diagram 23
1.7.2 Internal Units 24

Section 2 Port Functions 28

2.1 Port Functions 28
2.2 Overview 29
2.2.1 Terms 30
2.2.2 Pin Function Configuration 31
2.2.3 Pin Data Input/Output 35
2.2.4 Logic Diagram of Port Control 38
2.3 Port Group Configuration Registers 39
2.3.1 Overview 40
2.3.2 Pin Function Configuration Registers 42
2.3.3 Pin Data Input/Output Registers 50
2.3.4 Configuration of Electrical Characteristics Registers 54
2.3.5 Port Register Protection 58
2.4 Port Group Configuration 60
2.4.1 List of Ports and Pins 60
2.5 Functions of Pull-Up and Pull-Down Resistors 94
2.5.1 Details of Pull-Up and Pull-Down Resistors 94
2.6 Port 0 104
2.6.1 Alternative Functions 104
2.6.2 List of Control Registers 105
2.7 Port 1 106
2.7.1 Alternative Functions 106
2.7.2 List of Control Registers 107
2.8 Port 2 108
2.8.1 Alternative Functions 108
2.8.2 List of Control Registers 109
2.9 Port 3 110
2.9.1 Alternative Functions 110
2.9.2 List of Control Registers 111
2.10 Port 4 112
2.10.1 Alternative Functions 112
2.10.2 List of Control Registers 113
2.11 Port 5 114
2.11.1 Alternative Functions 114
2.11.2 List of Control Registers 115
2.12 Port 6 116
2.12.1 Alternative Functions 116
2.12.2 List of Control Registers 117
2.13 Port 7 118
2.13.1 Alternative Functions 118
2.13.2 List of Control Registers 119
2.14 Port 8 120

2.14.1 Alternative Functions ... 120
2.14.2 List of Control Registers ... 120
2.15 JTAG Port 0 ... 121
2.15.1 Alternative Functions ... 121
2.15.2 List of Control Registers ... 121
2.16 Cancelling Noise on Pins ... 122
2.16.1 Details of Noise Cancellation ... 122
2.16.2 Control Registers ... 133
2.17 Edge Detection ... 140
2.17.1 Details of Edge Detection ... 140
2.17.2 Control Register ... 141

Section 3 CPU System Function ... 142

3.1 Overview ... 142
3.1.1 Peripheral Protection Unit ... 144
3.1.2 Important Reminders ... 148
3.1.3 Timing Supervision Unit ... 148
3.2 Operation Modes ... 149
3.2.1 Normal Operation Mode ... 150
3.2.2 Flash Programming Mode ... 150
3.2.3 HALT Mode ... 151
3.3 Address Space ... 152
3.3.1 CPU Data Address and Physical Program Address Space ... 152
3.3.2 Program and Data Space ... 152
3.4 Conditions for Boundary Operations ... 154
3.4.1 Program Space ... 154
3.4.2 Data Space ... 154
3.5 Memory Mapping ... 155
3.5.1 Memory Map for DMA Access ... 155
3.5.2 Memory Map ... 156
3.6 CPU-Related Registers ... 157
3.6.1 Overview of CPU-Related Registers ... 157
3.6.2 CPU-Related Registers in Detail ... 158
3.7 System Error Notification Function ... 160
3.7.1 Overview of System Error Notification Registers ... 160
3.7.2 System Error Notification Function Registers in Detail ... 161

Section 4 Interrupt Functions ... 165

4.1 Features ... 165
4.2 List of Interrupt Sources ... 167
4.3 Interrupt Controller Control Registers ... 175
4.3.1 ICxx: EI Level Interrupt Control Register ... 175
4.3.2 IMRm (m = 0 to 12): EI Level Interrupt Mask Register ... 185
4.3.3 ISPR – In-Service Priority Register ... 195
4.3.4 PMR – Priority Mask Register ... 196
4.3.5 ISPC – In-Service Priority Clear Register ... 197
4.3.6 SCR – Selected Channel Hold Register ... 198
4.3.7 ICSR – Interrupt Controller Status Register ... 199
4.3.8 FNC – FE Level NMI Status Register ... 200
4.3.9 FIC – FE Level INT Status Register ... 201
4.3.10 INTCFGB – FE level Interrupt Switch Register ... 202
4.3.11 INTSTR0B – Interrupt Request Flag Storage Register ... 203
4.3.12 INTSTC0B – Interrupt Request Flag Clearing Register ... 204
4.3.13 INTSTS0B – Interrupt Request Flag Setting Register ... 205
4.4 Interrupt Acknowledgment and Restoring ... 206
4.4.1 FE Level Non-Maskable Interrupt Caused by FENMI Interrupt Request ... 206

4.4.2 Restore from FE Level Non-Maskable Interrupt (FENMI) 208
4.4.3 FE Level Maskable Interrupt Caused by FEINT Interrupt Request 208
4.4.4 Restore from FE Level Maskable Interrupt (FEINT) Servicing 210
4.4.5 EI Level Maskable Interrupt Caused by EIINT Interrupt Request 211
4.4.6 Restore from EI Level Maskable Interrupt (EIINT) 213
4.5 Interrupt Operation 214
4.5.1 Mask Function of EI Level Maskable Interrupt (EIINT) 214
4.5.2 Interrupt Priority Level Judgment 214
4.5.3 Priority Mask Function 220
4.5.4 Pending Interrupt Report Function 221
4.5.5 In-Service Priority Clear Function 221
4.6 Exception Handler Address Switching Function 222
4.7 Interrupt Response Times 222

Section 5 DMA/DTS Module 223

5.1 DMA/DTS in Overview 223
5.2 Definitions 228
5.3 Overview 229
5.3.1 Functions of the DMA Controller (DMAC) 229
5.3.2 Function of the DMA Trigger Factor Register (DTFR) 229
5.3.3 Functions of the Data Transfer Service (DTS) 229
5.3.4 Function of the DTS Factor Selector (DTSFSL) 229
5.3.5 Function of the DTS Selector (DTSSEL) 230
5.3.6 Memory Map for DMA/DTS Access 231
5.3.7 Channel Priority 231
5.3.8 Arbitration of Transfer Requests 232
5.4 DMAC Function 233
5.4.1 Characteristics 233
5.4.2 Setting Registers 235
5.4.3 Availability of Writing to Control Registers 240
5.5 DMAC Control Registers 241
5.5.1 DTRC0: DMA Transfer Request Control Register 0 241
5.5.2 DTRSn (n = 0 to 7): DMA Transfer Request Select Register 242
5.5.3 DSAnL (n = 0 to 7): DMA Source Address Register L 243
5.5.4 DSAnH (n = 0 to 7): DMA Source Address Register H 244
5.5.5 DSCn (n = 0 to 7): DMA Source Chip Select Register 245
5.5.6 DNSAnL (n = 0 to 7): DMA Next Source Address Register L 246
5.5.7 DNSAnH (n = 0 to 7): DMA Next Source Address Register H 247
5.5.8 DNSCn (n = 0 to 7): DMA Next Source Chip Select Register 248
5.5.9 DDAnL (n = 0 to 7): DMA Destination Address Register L 249
5.5.10 DDAnH (n = 0 to 7): DMA Destination Address Register H 250
5.5.11 DDCn (n = 0 to 7): DMA Destination Chip Select Register 251
5.5.12 DNDAnL (n = 0 to 7): DMA Next Destination Address Register L 252
5.5.13 DNDAnH (n = 0 to 7): DMA Next Destination Address Register H 253
5.5.14 DNDCn (n = 0 to 7): DMA Next Destination Chip Select Register 254
5.5.15 DTCn (n = 0 to 7): DMA Transfer Count Register 255
5.5.16 DNTCn (n = 0 to 7): DMA Next Transfer Count Register 256
5.5.17 DTCCn (n =0 to 7): DMA Transfer Count Compare Register 257
5.5.18 DTCTn (n = 0 to 7): DMA Transfer Control Register 258
5.5.19 DTSn (n = 0 to 7): DMA Transfer Status Register 260
5.6 DMAC Function Details 262
5.6.1 DMAC Transfer Setting Flow 262
5.6.2 DMAC Transfer Modes 263
5.6.3 DMAC Channel Priority Control 266
5.6.4 Conditions for Validity of DMA Transfer Requests 267
5.6.5 Next Address Function 268
5.6.6 Suspending/Resuming DMA Transfer 269

5.6.7 Error Responses ... 270
5.7 DTFR Functions ... 271
5.7.1 Features ... 271
5.8 DTFR Control Registers ... 272
5.8.1 DTFRn (n = 0 to 7): DMA Trigger Factor Register ... 272
5.8.2 DRQCLR: DMA Request Clear Register ... 273
5.8.3 DRQSTR: DMA Request Check Register ... 274
5.9 DTS Function ... 275
5.9.1 Features ... 275
5.10 DTS Control Registers ... 278
5.10.1 DTS0TSR: DTS Transfer Status Register [Register Group A] ... 279
5.10.2 DTS0TRC: DTS Transfer Request Control Register [Register Group A] ... 280
5.10.3 DTS0ICR: DTS Initialization Control Register [Register Group A] ... 281
5.10.4 DTS0BTR: DTS Base Table Register [Register Group A] ... 283
5.10.5 DTS0BVR: DTS Base Vector Register [Register Group A] ... 284
5.10.6 DTS0ACR: DTS Active Channel Register [Register Group A] ... 285
5.10.7 DTS0HST: DTS TI Hold Status Register [Register Group A] ... 286
5.10.8 DTS0HC: DTS TI Hold Channel Number Register [Register Group A] ... 287
5.10.9 DTS0SAR: DTS Source Address Register [Register Group B] ... 289
5.10.10 DTS0DAR: DTS Destination Address Register [Register Group B] ... 290
5.10.11 DTS0TCEA: DTS Transfer Counter or Else Address Register [Register Group B] ... 291
5.10.12 DTS0SCS: DTS Source Address Count Size Register [Register Group B] ... 293
5.10.13 DTS0DCS: DTS Destination Address Count Size Register [Register Group B] ... 294
5.10.14 DTS0CIR: DTS Control Information Register [Register Group B] ... 295
5.10.15 DTS0ECSRA: DTS Extension Address Count Size/ Repeat Address Registers [Register Group B] ... 297
5.11 DTS Function Details ... 299
5.11.1 Transfer information (TI) ... 299
5.11.2 DTS Transfer Setting Flow ... 302
5.11.3 Basic Operations of DTS ... 303
5.11.4 Transfer Modes ... 304
5.11.5 Transfer Types ... 305
5.11.6 Special Functions ... 323
5.11.7 Transfer Count ... 325
5.11.8 Chain Function ... 326
5.11.9 TI Hold Function ... 328
5.11.10 Interrupt Output Function ... 329
5.11.11 Interrupt and Chain of Transfers during Block Transfer ... 329
5.11.12 TI Write-back Skipping Function ... 330
5.11.13 DTS Channel Priority Control ... 331
5.11.14 Valid DTS Transfer Request Conditions ... 331
5.11.15 Suspending and Resuming DTS Transfer ... 331
5.11.16 Error Response Support ... 332
5.12 DTSFSL Function ... 333
5.12.1 Features ... 333
5.13 DTSFSL Control Registers ... 334
5.13.1 DTSENn: DTS Transfer Enable Register ... 334
5.13.2 DTSHENm (m = 0 to 3): DTS Hold Enable Register ... 335
5.14 DTSSEL Function (DTS Transfer Completion Interrupt) ... 337
5.14.1 Features ... 337
5.15 DTSSEL Control Registers ... 339
5.15.1 DTSSELn (n = 0 to 14): DTS Transfer Completion Interrupt Register ... 339

Section 6 Memory Modules ... 341

6.1 Features ... 342
6.1.1 Code Flash Memory ... 342
6.1.2 Data Flash Memory ... 343
6.1.3 On-Chip RAM ... 343
6.2 Programming Environment ... 344
6.3 Communications Methods ... 344
6.4 Handling of Pins ... 345
6.4.1 Power Supply ... 345
6.4.2 Pins ... 345
6.4.3 Reset Pin ... 345
6.4.4 FLMD0 Pin ... 346
6.4.5 Port Pins ... 346
6.5 Option-Setting Bytes ... 347
6.5.1 OPBT0 - Option Byte Verification Register ... 347
6.6 Product Identification ... 351
6.7 Setting the FLMD Pin ... 354
6.7.1 Registers ... 354
6.7.2 Setting the FLMDCNT Register ... 356

Section 7 Clock Generation ... 357

7.1 Overview of Clock Generation ... 358
7.2 Configuration ... 359
7.3 Selecting the Input Clock Signal ... 360
7.4 Clock Generating Circuit ... 361
7.5 Clock Output Function (CLKOUT) ... 362
7.5.1 Baud-Rate Generator for CLKOUT Function (BRGA) ... 362
7.5.2 BRGA Registers for CLKOUT Function ... 364
7.6 Clock Monitor A (CLMAn) Function ... 368
7.6.1 Features of the Clock Monitors (CLMAn; n = 0, 1, 2) ... 368
7.6.2 CLMA Enable and Start-Up Options ... 370
7.6.3 Functional Overview ... 370
7.6.4 Functional Description ... 371
7.6.5 Clock Monitor A Registers ... 377

Section 8 Reset Controller ... 382

8.1 Functional Overview ... 382
8.2 Functional Description ... 385
8.2.1 Reset Flags ... 385
8.2.2 Low-Voltage Indicator (LVI) ... 386
8.2.3 External RESET ... 387
8.2.4 Watchdog Timer Reset ... 388
8.2.5 Software Reset ... 388
8.2.6 Clock Monitor Reset ... 389
8.2.7 Self-Diagnostic BIST Reset ... 390
8.2.8 Safety Guardian Reset ... 390
8.2.9 Reset Flag Evaluation ... 391
8.2.10 Protection for Registers of the Reset Controller ... 392
8.3 Registers ... 393
8.3.1 Overview of Reset Controller Registers ... 393
8.3.2 Details of Reset Controller Registers ... 394
8.3.3 Details of Software Reset Control Registers ... 398
8.3.4 Details of Protection Command Registers ... 399
8.4 Power-on Flag/Low Voltage Indicator (POF/LVI) ... 400
8.4.1 Overview of the POF/LVI Function ... 400

8.4.2 Operation ... 401
8.4.3 Overview of POF/LVI Registers ... 404
8.4.4 Details of POF / LVI Control Registers ... 405

Section 9 Safety Functions ... 409

9.1 Peripheral Bus Access Error Detection ... 409
9.2 Memory Access Protection ... 409
9.2.1 Memory Access Protection ... 409
9.3 Registers Related to the Peripheral I/O Bus ... 410
9.3.1 Overview of Registers Related to the Peripheral I/O Bus ... 410
9.3.2 Details of Registers Related to the Peripheral I/O Bus ... 411
9.4 Overview of Self-Diagnostic BIST ... 416
9.4.1 Self-Diagnostic BIST Skip Function (Only for the µPD70F3505A) ... 416
9.5 Self-Diagnostic BIST Related Registers ... 417
9.5.1 Overview of Registers Related to Self-Diagnostic BIST ... 417
9.5.2 Details of Registers Related to Self-Diagnostic BIST ... 418
9.5.3 Procedure for Setting the BSEQ0STCHBT Register ... 425
9.6 ECC Related Registers ... 426
9.6.1 Overview of ECC Related Registers ... 426
9.6.2 Details of ECC Related Registers ... 427
9.7 Resources Required for Initialization ... 438
9.7.1 On-chip RAM ... 438
9.7.2 Registers ... 438

Section 10 Safety Guardian (SGA) ... 442

10.1 SGA Features ... 442
10.2 SGA Functional Overview ... 446
10.3 Functional Description ... 449
10.3.1 SGA Operating States ... 449
10.3.2 SGA Configuration Overview ... 450
10.3.3 Operations for Error Output ... 451
10.3.4 Loop-Back Function ... 452
10.3.5 Pseudo Error Generation ... 453
10.3.6 Error Status ... 454
10.3.7 Writing to Protected Registers ... 455
10.4 Registers ... 456
10.4.1 Overview of SGA Registers ... 456
10.4.2 SGA Registers Details ... 458

Section 11 Data CRC Function A (DCRA) ... 478

11.1 DCRA Features ... 478
11.2 Functional Overview ... 479
11.3 Functional Description ... 481
11.4 Registers ... 482
11.4.1 DCRA Registers Overview ... 482
11.4.2 DCRA Registers Details ... 483

Section 12 Window Watchdog Timer A (WDTA) ... 486

12.1 WDTA Features ... 486
12.2 Functional Overview ... 487
12.3 Functional Description ... 488
12.3.1 WDTA after Reset Release ... 488
12.3.2 WDTA Trigger ... 489
12.3.3 Error Detection ... 490

12.3.4 75% Interrupt Output ... 492
12.3.5 Window Function ... 493
12.4 Registers ... 494
12.4.1 WDTA Registers Overview ... 494
12.4.2 WDTA Registers Details ... 495

Section 13 Timer Array Unit A (TAUA) ... 497

13.1 TAUA Features ... 497
13.2 Functional Overview ... 500
13.2.1 Terms ... 502
13.3 Functional Description ... 503
13.3.1 Functional List of Timer Operations ... 505
13.4 General Operating Procedures ... 506
13.5 Operating Modes ... 507
13.6 Concepts of Synchronous Channel Operation ... 508
13.6.1 Rules ... 508
13.6.2 Simultaneous Start and Stop of Synchronous Channel Counters ... 510
13.7 Simultaneous Rewrite ... 511
13.7.1 Overview ... 511
13.7.2 How to Control Simultaneous Rewrite ... 513
13.7.3 Other General Rules of Simultaneous Rewrite ... 515
13.7.4 Type of Simultaneous Rewrite ... 516
13.8 Channel Output Modes ... 522
13.8.1 General Procedures for Specifying a Channel Output Mode ... 524
13.8.2 Channel Output Modes Controlled Independently by TAUAn Signals ... 525
13.8.3 Channel Output Modes Controlled Synchronously by TAUAn Signals ... 527
13.9 Start Timing in Each Operating Modes ... 532
13.9.1 Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode ... 532
13.9.2 Event Mode ... 533
13.9.3 Other Operating Modes ... 533
13.10 TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts ... 534
13.11 TAUAnTTINm Edge Detection ... 536
13.12 Assigning DMA Window Addresses ... 537
13.13 Independent Channel Operation Functions ... 538
13.13.1 Interval Timer Function ... 538
13.13.2 TAUAnTTINm Input Interval Timer Function ... 545
13.13.3 Delay Count Function ... 551
13.13.4 One-Pulse Output Function ... 555
13.13.5 TAUAnTTINm Input Pulse Interval Measurement Function ... 560
13.13.6 TAUAnTTINm Input Signal Width Measurement Function ... 567
13.13.7 TAUAnTTINm Input Period Count Detection Function ... 575
13.13.8 TAUAnTTINm Input Pulse Interval Judgment Function ... 580
13.13.9 TAUAnTTINm Input Signal Width Judgment Function ... 584
13.13.10 Real-Time Output Function Type 1 ... 589
13.13.11 Real-Time Output Function Type 2 ... 596
13.13.12 Simultaneous Rewrite Trigger Generation Function Type 1 ... 602
13.13.13 One-Phase PWM Output Function ... 609
13.13.14 External Event Count Function ... 615
13.13.15 Clock Divide Function ... 621
13.13.16 TAUAnTTINm Input Position Detection Function ... 627
13.14 Synchronous Channel Operation Functions ... 632
13.14.1 PWM Output Function ... 632
13.14.2 Delay Pulse Output Function ... 643
13.14.3 AD Conversion Trigger Output Function Type 1 ... 657
13.14.4 One-Shot Pulse Output Function ... 659

13.14.5 Offset Trigger Output Function ... 668
13.14.6 Triangle PWM Output Function ... 676
13.14.7 Triangle PWM Output Function with Dead Time ... 687
13.14.8 AD Conversion Trigger Output Function Type 2 ... 702
13.14.9 Non-Complementary Modulation Output Function Type 1 ... 704
13.14.10 Non-Complementary Modulation Output Function Type 2 ... 717
13.14.11 Complementary Modulation Output Function ... 731
13.14.12 Interrupt Sampling Function ... 748
13.15 Registers ... 757
13.15.1 Overview of TAUAn Registers ... 757
13.15.2 Details of TAUAn Prescaler Registers ... 759
13.15.3 Details of TAUAn Control Registers ... 764
13.15.4 Details of TAUAn Output Registers ... 774
13.15.5 Details of TAUAn Channel Output Level Registers ... 781
13.15.6 Details of TAUAn Simultaneous Rewrite Registers ... 782
13.15.7 TAUAn DMA Window Registers ... 785

Section 14 Timer Array Unit J (TAUJ) ... 787

14.1 Features of TAUJ ... 787
14.2 Functional Overview ... 790
14.2.1 Terms ... 792
14.3 Functional Description ... 793
14.3.1 Timer Operation Functions ... 794
14.4 General Operating Procedure ... 795
14.5 Operating Modes ... 796
14.6 Concepts of Synchronous Channel Operation ... 797
14.6.1 Rules ... 797
14.6.2 Simultaneous Start and Stop of Synchronous Channel Counters ... 799
14.7 Simultaneous Rewrite ... 800
14.7.1 Overview ... 800
14.7.2 How to Control Simultaneous Rewrite ... 801
14.7.3 Other General Rules for Simultaneous Rewrite ... 802
14.7.4 Simultaneous Rewrite Procedure ... 803
14.8 Channel Output Modes ... 805
14.8.1 General Procedure for Specifying a Channel Output Mode ... 807
14.8.2 Channel Output Modes Controlled Independently by TAUJn Signals ... 808
14.8.3 Channel Output Modes Controlled Synchronously by TAUJn Signals ... 809
14.9 Start Timing of Operating Modes ... 810
14.9.1 Interval Timer Mode and Capture Mode ... 810
14.9.2 Other Operating Modes ... 811
14.10 TAUJnTTOUTm Output and INTTAUJnIm Generation When Counter Starts or Restarts ... 812
14.11 TAUJnTTINm Edge Detection ... 813
14.12 Independent Channel Operation Functions ... 814
14.12.1 Interval Timer Function ... 814
14.12.2 TAUJnTTINm Input Interval Timer Function ... 821
14.12.3 TAUJnTTINm Input Pulse Interval Measurement Function ... 827
14.12.4 TAUJnTTINm Input Signal Width Measurement Function ... 836
14.12.5 TAUJnTTINm Input Period Count Detection Function ... 844
14.12.6 TAUJnTTINm Input Position Detection Function ... 849
14.13 Synchronous Channel Functions ... 854
14.13.1 PWM Output Functions ... 855
14.14 Registers ... 866
14.14.1 Overview of TAUJn Registers ... 866
14.14.2 Details of TAUJn Prescaler Registers ... 867
14.14.3 Details of TAUJn Control Registers ... 871
14.14.4 Details of TAUJn Output Registers ... 882

14.14.5 Details of TAUJn Channel Output Level Registers 884
14.14.6 Details of TAUJn Reload Data Registers 885

Section 15 TSG2 (TSG20, TSG21) 887

15.1 Functions of TSG2n 887
15.2 Functional Overview 889
15.3 Configuration 890
15.4 Registers 891
15.4.1 TSG2n Registers Overview 891
15.4.2 TSG2n Register Details 893
15.5 Basic Operation 944
15.5.1 Basic Operation of 16-Bit Counter 944
15.5.2 Functions of Compare Registers 946
15.5.3 Compare Register Rewrite Operation 949
15.5.4 List of Outputs in Each Mode 956
15.6 Match Interrupt 962
15.7 Flags 965
15.7.1 Up Count Flag (TSnCUF and TSnSUF) 966
15.7.2 Positive Phase and Inverse Phase Simultaneous Active State Detection Flag (TSnTBF0 to TSnTBF2) 968
15.7.3 Reload Request Flag (TSnRSF) 969
15.7.4 Noise Detection Flag (TSnNDF) 970
15.7.5 Pattern Order Detection Flag (TSnTSF) 971
15.7.6 Pattern Error Detection Flag (TSnPEF) 973
15.7.7 Pattern Reversal Detection Flag (TSnPRF) 974
15.7.8 TSG2nPTSI2 to TSG2nPTSI0 Pin Abnormal Toggle Detection Flag (TSnPTF) 975
15.7.9 TSnSTCI0 and TSnSTCI1 Signal Simultaneous Trigger Detection Flag (TSnTDF) 976
15.7.10 Pattern Phase Difference Detection Flag (TSnPPF) 977
15.7.11 Timer Output Pattern Flag (TSnOPF2 to TSnOPF0) 978
15.7.12 Pattern Switch Detection Signal (TSnPTE) 979
15.8 Interrupt Skipping Function 981
15.8.1 Operation of Interrupt Skipping Function 982
15.8.2 Example of Operation when Peak Interrupt is Generated (in PWM Mode) 986
15.9 A/D Conversion Trigger Function 987
15.9.1 Operation of A/D Conversion Trigger 988
15.10 Error/Warning Interrupt 994
15.10.1 Error Interrupt Function 994
15.10.2 Warning Interrupt Function 997
15.11 Operating Modes 998
15.11.1 PWM Mode 999
15.11.2 HT-PWM Mode (High accuracy Triangular - Pulse Width Modulation Mode) 1010
15.11.3 SP-PWM Mode (Shifted-pulse - Pulse Width Modulation Mode) 1032
15.11.4 120-DC Mode 1044
15.11.5 Software Output Control Function 1078

Section 16 TPBA (TPBA0 and TPBA1) 1080

16.1 Functions of TPBAn 1080
16.2 Functional Overview 1081
16.3 Configuration 1082
16.4 Registers 1083
16.4.1 TPBAn Registers Overview 1083
16.4.2 TPBAn Registers Details 1084

16.5 Basic Operation ... 1097
16.5.1 Basic Operation of Counter ... 1097
16.5.2 Compare Register Rewrite Operation ... 1098
16.5.3 Duty Rewrite Operation ... 1101
16.5.4 Basic Operation Example ... 1104

Section 17 OS Timer (OSTM) ... 1109

17.1 OSTM Features ... 1109
17.2 Functional Overview ... 1111
17.3 Functional Description ... 1112
17.3.1 Clock Signal to Drive Counting ... 1112
17.3.2 Output Modes ... 1114
17.3.3 Interrupt Request Generation ... 1115
17.3.4 Starting and Stopping the Timer ... 1116
17.3.5 Interval Timer Mode ... 1117
17.3.6 Free-Running Comparison Mode ... 1121
17.4 Registers ... 1124
17.4.1 OS Timer Registers Overview ... 1124
17.4.2 OS Timer Registers in Detail ... 1125

Section 18 Encoder Timer (ENCA) ... 1135

18.1 ENCA Features ... 1135
18.2 Functional Overview ... 1137
18.2.1 Block Diagram ... 1138
18.3 ENCA Control Registers ... 1139
18.4 Functional Description ... 1154
18.4.1 Timer Counter Operation ... 1154
18.4.2 Up/Down Control of Timer Counter ... 1158
18.4.3 Control of Timer Counter Clearing ... 1162
18.4.4 Detailed Description of the ENCAnCCR0 Register ... 1168
18.4.5 Detailed Description of the ENCAnCCR1 Register ... 1170
18.4.6 Timer Counter Operation Start/Stop ... 1174
18.5 Setting Sequences ... 1176
18.5.1 Encoder Timer Setting Procedure ... 1176
18.6 Timing Charts for Encoder Operations ... 1179

Section 19 Timer Option Module (TAPA) ... 1185

19.1 Features of the Timer Option Module ... 1185
19.2 Functional Overview ... 1196
19.2.1 Block Diagram ... 1196
19.2.2 Peak and Valley of Timer Counter, and Peak and Valley Interrupts ... 1197
19.3 Registers ... 1198
19.3.1 Registers Overview ... 1198
19.3.2 Registers Details ... 1199
19.4 Basic Functions ... 1205
19.4.1 Hi-Z Control Functions ... 1205
19.4.2 Asynchronous Hi-Z Control for Pin Inputs ... 1206
19.4.3 Selection of INT Signal Output ... 1212
19.4.4 Selecting a Trigger to Start Conversion by the A/D Converter ... 1216

Section 20 CAN Controller (FCN) ... 1222

20.1 FCN Features ... 1222
20.2 Features ... 1224
20.2.1 Overview of Functions ... 1225

20.2.2 Configuration ... 1226
20.3 Internal Registers of FCN ... 1227
20.3.1 CAN Controller Configuration ... 1227
20.3.2 Overview of CAN Controller Registers ... 1228
20.3.3 Register Bit Configuration ... 1230
20.4 Bit Set/Clear Functions ... 1235
20.5 Control Registers ... 1237
20.5.1 FCN Global Registers ... 1237
20.5.2 FCN Module Registers ... 1245
20.5.3 FCN Message Buffer Registers ... 1265
20.6 CAN Controller Initialization ... 1274
20.6.1 Initialization of FCN Module ... 1274
20.6.2 Redefinition of Message Buffer ... 1274
20.6.3 Transition from Initialization Mode to Operating Mode ... 1276
20.7 Reception of Message ... 1277
20.7.1 Message Reception ... 1277
20.7.2 Receive Data Read ... 1278
20.7.3 Receive History List Function ... 1279
20.7.4 Mask Function ... 1281
20.7.5 Multi Buffer Receive Block Function ... 1282
20.7.6 Remote Frame Reception ... 1283
20.8 Transmission of Message ... 1285
20.8.1 Message Transmission ... 1285
20.8.2 Transmit History List Function ... 1287
20.8.3 Automatic Block Transmission (ABT) ... 1289
20.8.4 Transmission Abort Process ... 1291
20.8.5 Remote Frame Transmission ... 1292
20.9 Power Save Modes ... 1293
20.9.1 FCN Sleep Mode ... 1293
20.9.2 FCN Stop Mode ... 1296
20.9.3 Example of Using Power Save Modes ... 1297
20.10 Interrupt Function ... 1298
20.11 Diagnosis Functions and Special Operating Modes ... 1299
20.11.1 Receive-only Mode ... 1299
20.11.2 Single-shot Mode ... 1300
20.11.3 Self-test Mode ... 1301
20.11.4 Receive/Transmit Operations in Each Operating Mode ... 1302
20.12 Time Stamp Function ... 1303
20.12.1 Time Stamp Function ... 1303
20.13 Baud Rate Settings ... 1305
20.13.1 Baud Rate Setting Conditions ... 1305
20.13.2 Typical Examples of Baud Rate Settings ... 1308
20.14 Operation of the CAN Controller ... 1312
20.14.1 Initialization ... 1312
20.14.2 Message Transmission ... 1318
20.14.3 Message Reception ... 1330
20.14.4 Power Save Modes ... 1335

Section 21 Clocked Serial Interface G (CSIG) ... 1342

21.1 CSIG Features ... 1342
21.2 Functional Overview ... 1344
21.3 Functional Description ... 1346
21.3.1 Master/Slave Mode ... 1346
21.3.2 Master/Slave Connections ... 1347
21.3.3 Selection of Serial Communications Clock ... 1348
21.3.4 Data Transfer Modes ... 1349
21.3.5 Data length Selection ... 1350

21.3.6 Serial Data Direction Selection 1352
21.3.7 Communication in Slave Mode 1353
21.3.8 CSIG Interrupts 1354
21.3.9 Handshake Function 1357
21.3.10 Loop-Back Mode 1359
21.3.11 Error Detection 1360
21.4 CSIG Control Registers 1364
21.5 Operating Procedure Example 1378

Section 22 Queue-Equipped Three-Line Serial Interface H (CSIHn) 1380

22.1 CSIHn Features 1381
22.2 Functional Overview 1384
22.3 Functional Description 1386
22.3.1 Operating Modes (Master/Slave) 1387
22.3.2 Master/Slave Connections 1388
22.3.3 Chip Selection (CS) Features 1390
22.3.4 Chip Select Timing Details 1393
22.3.5 Serial Clock Selection 1395
22.3.6 CSIHn Buffer Memory 1396
22.3.7 Data Transfer Modes 1398
22.3.8 Data Length Selection 1400
22.3.9 Serial Data Direction Selection 1403
22.3.10 CSIHn Interrupt Requests 1404
22.3.11 Handshake Function 1412
22.3.12 Error Detection 1416
22.3.13 Loop-Back Mode 1425
22.4 CSIHn Control Registers 1426
22.4.1 CSIH Registers Details 1427
22.5 Operating Procedures 1452
22.5.1 Procedures in Direct Access Mode 1452
22.5.2 Procedures in Transmit-Only Buffer Mode 1457
22.5.3 Procedures in Double-Buffer Mode 1461
22.5.4 Procedures in FIFO Mode 1467

Section 23 Synchronous/Asynchronous Serial Interface H (UARTH) 1471

23.1 UARTHn Features 1471
23.2 Features 1473
23.3 Configuration 1474
23.4 UARTHn Registers 1475
23.5 Interrupt Request Signals 1505
23.5.1 Transmission Interrupt Request URTHnTIT 1505
23.5.2 Reception Interrupt Request URTHnTIR 1506
23.5.3 Status Interrupt Request URTHnTIS 1506
23.6 Operation 1507
23.6.1 Data Formats 1507
23.6.2 Clock Synchronous Mode 1509
23.6.3 Handshake Mode 1510
23.6.4 Consecutive Two-Frame Transfer Mode 1512
23.6.5 Enhanced Bit Detection/ID Compare-Match Detection 1514
23.6.6 BF Transmission/Reception Format 1516
23.6.7 BF Transmission 1518
23.6.8 BF Reception 1519
23.6.9 Transmission Data Consistency Check 1520
23.6.10 UARTHn Transmission 1521
23.6.11 Continuous Transmission Procedure 1522
23.6.12 UARTHn reception 1524

23.6.13 Reception Errors 1526
23.6.14 Parity Types and Operations 1527
23.6.15 Digital Receive Data Noise Filter 1528
23.7 Baud-Rate Generator 1529
23.8 Detecting the Baud Rate in LIN Communications as a Slave 1530

Section 24 FlexRay™ (FLX) 1531

24.1 V850E2/Px4 FLXn Features 1531
24.2 Error Correction Module (ECC) 1535
24.3 FlexRay Controller Overview 1537
24.3.1 Conventions 1537
24.3.2 Definition 1537
24.3.3 References 1537
24.3.4 Terms and Abbreviations 1538
24.3.5 Functional Overview 1539
24.3.6 Block Diagram 1540
24.3.7 Reset Timing 1542
24.4 Programmer's Model 1543
24.4.1 Register Map 1543
24.4.2 HIF Registers 1548
24.4.3 Special Registers 1552
24.4.4 Interrupt Registers 1560
24.4.5 CC Control Registers 1582
24.4.6 CC Status Registers 1606
24.4.7 Message Buffer Control Registers 1624
24.4.8 Message Buffer Status Registers 1630
24.4.9 Identification Registers 1643
24.4.10 Input Buffer 1645
24.4.11 Output Buffer 1654
24.5 Functional Description 1666
24.5.1 Communication Cycle 1666
24.5.2 Communication Modes 1669
24.5.3 Clock Synchronization 1669
24.5.4 Error Handling 1672
24.5.5 Communication Controller States 1674
24.5.6 Network Management 1689
24.5.7 Filtering and Masking 1690
24.5.8 Transmit Process 1693
24.5.9 Receive Process 1696
24.5.10 FIFO Function 1698
24.5.11 Message Handling 1700
24.5.12 Message RAM 1710
24.5.13 Module Interrupt 1719
24.6 Appendix 1721
24.6.1 Assignment of FlexRay Configuration Parameters 1721
24.7 Cautions 1723
24.7.1 Loop Back Mode Operates Only at 10 MBit/s 1723
24.7.2 Noise Following a Dynamic Frame that Delays Idle Detection may Fail to Stop Slot Counting for the Remainder of the Dynamic Segment 1723
24.7.3 Register FLXnRCV Displays Wrong Value 1724
24.7.4 After Reception of a Valid Sync Frame Followed by a Valid Non-Sync Frame in the Same Static Slot, the Received Sync Frame may be Ignored 1725
24.7.5 Sync Frame Overflow Flag FLXnEIR.FLXnSFO may be Set If Slot Counter is Greater Than 1024 1726
24.7.6 Acceptance of Startup Frames Received After Reception of More Than gSyncNodeMax sync Frames 1726

24.7.7 Initial Rate Correction Value of an Integrating Node is Zero If pMicroInitialOffsetA, B = 00_H ... 1727
24.7.8 Incorrect Rate and/or Offset Correction Value If Second Secondary Time Reference Point (STRP) Coincides with the Action Point After Detection of a Valid Frame ... 1727
24.7.9 Flag FLXnSFS.FLXnMRCS is Set Erroneously Although at Least One Valid Sync Frame Pair is Received ... 1728
24.7.10 Rate Correction Set to Zero in Case of SyncCalcResult = MISSING_TERM ... 1729
24.7.11 A Sequence of Received WUS May Generate Redundant FLXnSIR.FLXnWUPA/B Events ... 1730
24.7.12 The Interval of Cycles after Startup Commences is Shortened by the READY Command During Startup and Normal Operation ... 1730
24.7.13 Prohibit Issuing of the READY Command while the POC is in the Ready State ... 1731
24.7.14 There is a Possibility of the First Wakeup Symbol (WUS) after the Reception of a Valid Wakeup Pattern (WUP) being Ignored ... 1731
24.7.15 Initialization of the Slot Mode by the HALT or FREEZE Command is Immediately Executed ... 1732

Section 25 A/D Converter ... 1733

25.1 ADCA Features ... 1733
25.2 Functional Overview ... 1736
25.3 Functional Description ... 1738
25.3.1 Basic Operation ... 1739
25.3.2 Clock Usage ... 1740
25.3.3 Channel and Channel Group ... 1741
25.3.4 A/D Conversion Modes ... 1743
25.3.5 Starting A/D Conversion (Start Triggers) ... 1746
25.3.6 Stopping A/D Conversion (Stop Trigger) ... 1748
25.3.7 Resolution, Sampling Time and Conversion Time ... 1751
25.3.8 Interrupt Generation ... 1754
25.3.9 Storage of A/D Conversion Results ... 1755
25.3.10 Result Check Functions ... 1758
25.3.11 Self-Diagnosis Functions ... 1760
25.3.12 Channel Sample and Hold Function ... 1768
25.3.13 Discharge Function ... 1775
25.3.14 Buffer Amplifier Function ... 1776
25.3.15 Stabilization Control ... 1776
25.4 Registers ... 1777
25.4.1 ADCAn Registers Overview ... 1777
25.4.2 Control Registers ... 1778
25.4.3 Conversion Status Registers ... 1787
25.4.4 Software Trigger Registers ... 1791
25.4.5 A/D Conversion Result Registers ... 1793
25.4.6 A/D Conversion Result Upper/Lower Limit Comparison Registers ... 1801
25.4.7 Diagnosis Function Control Registers ... 1805
25.4.8 Channel Sample and Hold Function Setting Register ... 1808
25.5 Usage Notes ... 1809
25.5.1 Range of Channel Input Voltage ... 1809
25.5.2 Stopping Conversion Operation ... 1809
25.5.3 Restrictions on Using Channel Sample and Hold Function ... 1809
25.5.4 Application Design Notes ... 1809
25.6 How to Read A/D Converter Characteristics Table ... 1815

Section 26 Peripheral Interconnection (PIC) 1822

26.1 Features of Peripheral Interconnection 1822
26.2 Functional Overview 1823
26.3 Peripheral Interconnection Registers 1824
26.3.1 Start of PIC Function 1827
26.4 Connection Functions 1828
26.4.1 Simultaneous Start Trigger Function 1828
26.4.2 ADC Trigger Selection Function 1836
26.4.3 Three-Phase PWM Output Function with Dead Time 1844
26.4.4 High Accuracy Triangle Wave PWM Output Function with Dead Time 1865
26.4.5 Delay Pulse Output Function with Dead Time 1895
26.4.6 Trigger and Pulse Width Measurement Function 1916
26.4.7 A/D Trigger Encoder Capture Function 1930
26.4.8 Two-Phase Encoder Control Function (Control Method 1) 1937
26.4.9 Two-Phase Encoder Control Function (Control Method 2) 1953
26.4.10 Three-Phase Pulse Input Control Function 1970
26.4.11 Three-Phase Encoder Control Function 1990
26.4.12 CAN Time Stamp Function 2003

Section 27 On-chip Debugging Unit (OCD) 2006

27.1 Functional Overview 2006
27.2 Connection with the On-Chip Debugging Emulator 2008
27.3 Notes on On-Chip Debugging 2010

Section 28 Electrical Characteristics 2012

28.1 Electrical Characteristics of the µPD70F3505A 2012
28.1.1 Absolute Maximum Ratings 2012
28.1.2 Capacity 2013
28.1.3 Operating Conditions 2013
28.1.4 Characteristics of Oscillation Circuit 2014
28.1.5 DC Characteristics 2015
28.1.6 AC Characteristics 2019
28.2 Electrical Characteristics of the µPD70F3506 2058
28.2.1 Absolute Maximum Ratings 2058
28.2.2 Capacity 2059
28.2.3 Operating Conditions 2059
28.2.4 Characteristics of Oscillation Circuit 2060
28.2.5 DC Characteristics 2061
28.2.6 AC Characteristics 2065
28.3 Electrical Characteristics of the µPD70F3507 2103
28.3.1 Absolute Maximum Ratings 2103
28.3.2 Capacity 2104
28.3.3 Operating Conditions 2104
28.3.4 Characteristics of Oscillation Circuit 2105
28.3.5 DC Characteristics 2106
28.3.6 AC Characteristics 2110
28.4 Electrical Characteristics of the µPD70F3508 (Target) 2150
28.4.1 Absolute Maximum Ratings 2150
28.4.2 Capacity 2151
28.4.3 Operating Conditions 2151
28.4.4 Characteristics of Oscillation Circuit 2152
28.4.5 DC Characteristics 2153
28.4.6 AC Characteristics 2156
28.5 Electrical Characteristics of the µPD70F3509 (Target) 2194
28.5.1 Absolute Maximum Ratings 2194

28.5.2 Capacity 2195
28.5.3 Operating Conditions 2195
28.5.4 Characteristics of Oscillation Circuit 2196
28.5.5 DC Characteristics 2197
28.5.6 AC Characteristics 2200

Section 1 Introduction

V850E2/Px4 is a product in the V850 Series of single-chip microcomputers from Renesas Electronics. This section gives an overview of the V850E2/Px4.

1.1 Overview

This 32-bit single-chip microcomputer is a best fit for applying high-speed calculation in the precise control of motors and is equipped with V850E2M CPUs from the V850 Series, ROM, RAM, a DMA controller, and various peripheral modules including several timers such as a PWM timer, serial interfaces such as CAN, and an A/D converter.

(1) V850E2M CPU Equipped

This product is equipped with two V850E2M CPUs: one is used as the master CPU and the other is used as a checker CPU, which monitors the master CPU. The two CPUs operate in lockstep.

The V850E2M CPU supports a RISC-type instruction set, and instruction execution speeds are dramatically improved by the execution of one basic instruction per clock cycle and an optimized 7-stage pipeline. Furthermore, support for digital servo-control is provided by a 32-bit hardware multiplier for the multiplication instruction, a saturation sum-of-products instruction, bit-manipulation instructions, and so on.

In addition, the 2-byte length for instructions, instructions corresponding to high-level languages, and other measures improve the efficiency of object code from the C compiler, leading to smaller executable programs. The product's quick interrupt response, including processing by the on-chip interrupt controller, makes it suitable for advanced realtime control fields.

(2) Flash Memory and Data Flash Memory Equipped

This product has a flash memory that provides high-speed access, allowing reprogramming of the product's memory while it is mounted on an application system. This shortens system development periods and dramatically improves systems for maintenance after the product is shipped.

Moreover, the product is equipped with 32 Kbytes of data-flash memory for use, in typical examples, as a substitute for EEPROM.

(3) Products for the Development Environment

The following components of integrated development environments are ready for use with this product: optimized C compiler, debugger, in-circuit emulator, system performance analyzer, realtime OS, etc.

The Nexus function allows realtime debugging.

(4) Protection Function

The following functions are available to guarantee highly reliable CPU operations: memory protection, peripheral device protection, system register protection, and timing supervision.

1.2 Characteristics

CPU core	V850E2M: 2 units (with lockstep operation) One CPU core for normal operation (master CPU), and one CPU core to monitor the first for normal operation (checker CPU)
Number of instructions	Normal instructions: 98 FPU instructions: 73 (the µPD70F3505A is not equipped with these) Debugging instructions: 3
Minimum instruction execution times	20.83, 15.625, 12.5, and 6.25 ns (for internal operation at 48, 64, 80, and 160 MHz)
General purpose registers	32 bits × 32
Instruction set	V850E2v3 Signed multiplication (32 bits × 32 bits → 64 bits): 1 or 2 CPU clock cycles Saturation calculation instruction (with overflow/underflow detection) 32-bit arithmetic/logic shift instruction: 1 CPU clock cycle Bit-manipulation instructions Loading/storing instructions in long/short formats Signed load instruction
Memory space	512-Mbyte address space (common for programs and data)
On-chip memory	Flash memory: 384, 512, or 1024 Kbytes Data flash: 32 Kbytes RAM: 24, 40, or 80 Kbytes
Interrupts/ exceptions	All interrupts: 202 sources (max.) External interrupts: 11 (max.) Internal interrupts: 180 sources (max.) Software interrupts: 11 sources Exceptions: 21 sources (max.) 16 levels available for use in prioritization

DMA controller

8-channel configuration

Transfer unit: 8, 16, 32, or 128 bits

Maximum number of transfers: 32768(2^{15})

Transfer type: 2-cycle transfer (dual-address transfer)

Transfer mode: single transfer/single step transfer

Target to transfer: peripheral I/O ←→ peripheral I/O, RAM ←→ peripheral I/O, RAM ←→ RAM, peripheral I/O ← flash, RAM ← flash, peripheral I/O ← D- flash, RAM ← D- flash

Transfer requests: peripheral I/O and software

Next address setting function

DTS controller

Channel-free (channels 0 to 111) configuration

Unit of transfer: 8, 16, or 32 bits

Maximum number of transfers: 65535(2^{16}-1) times (when the 16-bit register is specified)
255 times (when the 16-bit register is split for use)

Transfer type: 2-cycle transfer (dual-address transfer)

Transfer mode: single transfer/block transfer

Targets for transfer: peripheral I/O ←→ peripheral I/O, RAM ←→ peripheral I/O, RAM ←→ RAM, peripheral I/O ←→ flash, RAM ←→ flash, peripheral I/O ←→ D- flash, RAM ←→ D- flash

Transfer requests: peripheral I/O and software

Chaining function

Comparison function

Handling transfer errors

I/O

I/O port: 43 (µPD70F3505A), 73 (µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509)

Software pull-up function: 43 (µPD70F3505A), 73 (µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509)

Timer function

Timer array unit A (TAUA): 16 channels × 2 units

Timer unit including 16 independent 16-bit counter channels, and a dedicated prescaler

Timer array unit J (TAUJ): 4 channels × 2 units

Timer unit including 4 independent 32-bit counter channels, and a dedicated prescaler

TSG2: 2 units

Timer unit suitable for motor control

Timer pattern buffer (TPBA): 2 units

Timer unit suitable for duty-cycle and period settings

Encoder timer (ENCA): 2 units

Timer unit suitable for 2-phase encoder control

OS timer (OST): 2 units

For use by the OS: 32-bit free running interval timer with timer output function

Window watchdog timer (WDTA0)

- Window watchdog function: window and 75% interrupt
- Reset on error detection, or generation of FENMI/FEINT

Serial interface (SIO)	Synchronous/asynchronous serial interface H (UARTH): 3 channels • Switching between synchronous and asynchronous settings • Available for use as a slave in LIN communications Clocked serial interface (CSIG): 3 channels Clocked serial interface with queue (CSIH): 2 channels (µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509), 1 channel (µPD70F3505A) CAN interface: 2 channels (64 msg) FlexRay interface: 1 unit, 2 channels (not included in the µPD70F3505A)
A/D converter	A/D converter with 10-bit and 12-bit resolution: 22 channels (6 S/H channels + 16 other channels) × 1 unit (µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509) 18 channels (6 S/H channels + 12 other channels) × 1 unit (µPD70F3505A)
Internal connecting function	Peripheral Interconnection (PIC): 1 unit Interlocking operations (simultaneous starting and so on) are obtainable through connection to timers and peripheral I/O
Forcibly stopping output	Timer option unit (TAPA): 4 Unit for controlling the Hi-Z state of pins for the TAUA and TSG2 timers
Generating clock signals	Frequency multiplication by PLL clock synthesizer Clock output functions
Baud rate generator (BRG)	Allows setting of the operating clock frequency to suit the conditions of use
Data CRC	Data CRC (cyclic redundancy checking) can be used to verify or generate data streams protected by CRC with different widths in bits for various lengths.
Safety functions	Flash memory ECC error detection RAM ECC error detection Oscillation stop detection Build In Self Test (BIST) functions Safety guardian function (SGA)
Standby function	HALT mode
POF/LVI function	Facility for detecting abnormal power-supply voltages
On-chip debugger	NEXUS: 1 channel
Packages	100-pin plastic LQFP (0.5 mm pitch) (14 × 14) 144-pin plastic HLQFP (0.5 mm pitch) (20 × 20)

1.3 List of Functions

Series Name			V850E2/Px4				
Product			µPD70F3505A	µPD70F3506	µPD70F3507	µPD70F3508	µPD70F3509
On-chip memory	Code flash		384 Kbytes	512 Kbytes	512 Kbytes	1 Mbyte	1 Mbyte
	Data flash		32 Kbytes	32 Kbytes	32 Kbytes	32 Kbytes	32 Kbytes
	RAM		24 Kbytes	40 Kbytes	40 Kbytes	80 Kbytes	80 Kbytes
	Tuning RAM		8 Kbytes	8 Kbytes	8 Kbytes	8 Kbytes	8 Kbytes
Operating frequency	Ta: 125°C (max.)		–	80 MHz (max.)	80 MHz (max.)	160 MHz (max.)	160 MHz (max.)
	Ta: 110°C (max.)		80 MHz (max.)	–	–	–	–
Oscillation frequency			8 MHz	16 MHz	8/16 MHz	16 MHz	8/16 MHz
FPU*			Not included	Included	Included	Included	Included
Baud-rate generator (BRG)			1 unit	1 unit	1 unit	1 unit	1 unit
DMA + DTS			8 channels + 0 to 111 channels	8 channels + 0 to 111 channels	8 channels + 0 to 111 channels	8 channels + 0 to 111 channels	8 channels + 0 to 111 channels
Timers	Unit functions	TSG2	2 units	2 units	2 units	2 units	2 units
		TAUA	2 units	2 units	2 units	2 units	2 units
		TAUJ	2 units	2 units	2 units	2 units	2 units
		OST	2 units	2 units	2 units	2 units	2 units
		WDTA	1 unit	1 units	1 unit	1 units	1 unit
		ENCA	2 units	2 units	2 units	2 units	2 units
		TPBA	2 units	2 units	2 units	2 units	2 units
		TAPA	4 units	4 units	4 units	4 units	4 units
		PIC	1 unit	1 units	1 unit	1 units	1 unit
Serial interfaces	Synchronous/ asynchronous UARTH		3 channels	3 channels	3 channels	3 channels	3 channels
	CSIG		3 channels	3 channels	3 channels	3 channels	3 channels
	CSIH		1 channel	2 channels	2 channels	2 channels	2 channels
	CAN (number of messages)		2 channels (64 msg)	2 channels (64 msg)	2 channels (64 msg)	2 channels (64 msg)	2 channels (64 msg)
	FlexRay		–	2 channels (1 unit)	2 channels (1 unit)	2 channels (1 unit)	2 channels (1 unit)
A/D converter	Resolution		12 bits	10 bits	12 bits	10 bits	12 bits
	Input channels		18 channels (6 S/H + 12 other)	22 channels (6 S/H + 16 other)	22 channels (6 S/H + 16 other)	22 channels (6 S/H + 16 other)	22 channels (6 S/H + 16 other)
Interrupts	External		10	11	11	11	11
	Internal		167	180	180	180	180
	Software		11	11	11	11	11
I/O ports	Input/output		43	73	73	73	73
NEXUS			1 channel	1 channel	1 channel	1 channel	1 channel
Data CRC (DCRA)			2 channels	2 channels	2 channels	2 channels	2 channels
Safety guardian (SGA)			1 channel	1 channels	1 channel	1 channel	1 channel
Power-on flag			Included	Included	Included	Included	Included
Low voltage detection			Included	Included	Included	Included	Included
Self-diagnosis BIST-SKIP function			Included	Not included	Not included	Not included	Not included

Series Name		V850E2/Px4				
Product		µPD70F3505A	µPD70F3506	µPD70F3507	µPD70F3508	µPD70F3509
Power-supply voltages	For internal use	1.2 V ± 0.1 V	1.2 V ± 0.1 V	1.2 V ± 0.1 V	1.2 V ± 0.1 V	1.2 V ± 0.1 V
	For interfaces	3.0 V to 5.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V
	For A/D converter	5 V ± 0.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V
	For reprogramming flash memory	3.0 V to 5.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V	3.3 V ± 0.3 V	5 V ± 0.5 V
Package		100 pins LQFP 0.5-mm pitch 14 × 14 mm □ 1.4-mm thick	144 pins HLQFP 0.5-mm pitch 20 × 20 mm □ 1.4-mm thick	144 pins HLQFP 0.5-mm pitch 20 × 20 mm □ 1.4-mm thick	144 pins HLQFP 0.5-mm pitch 20 × 20 mm □ 1.4-mm thick	144 pins HLQFP 0.5-mm pitch 20 × 20 mm □ 1.4-mm thick

Note: The µPD70F3505A does not include an FPU. Therefore, the FPU instructions (73 codes) cannot be used, and floating-point calculation exceptions are not generated in response to the execution of source code including FPU instructions.

1.4 Field of Application

- Electrical systems (motor control systems, electric automobiles, etc.)

1.5 Information for Ordering

Product Name	Package	On-Chip ROM	Quality Standard	Operating Ambient Temperature (T_A)	External Oscillator	Maximum Operating Frequency
µPD70F3505AGCA1-UEU-AX	100 pin (0.5-mm pitch) (14 × 14) 1.4-mm thick	Flash memory	Special	-40 to +110°C	8 MHz	80 MHz
µPD70F3506GJA2-GBG-AX	144 pin (0.5-mm pitch) (20 × 20) 1.4-mm thick			-40 to +125°C	16 MHz	
µPD70F3507M1GJA2-GBG-AX					8 MHz	
µPD70F3507M2GJA2-GBG-AX					16 MHz	
µPD70F3508GJA2-GBG-AX					16 MHz	160 MHz
µPD70F3509M1GJA2-GBG-AX					8 MHz	
µPD70F3509M2GJA2-GBG-AX					16 MHz	

Note The products listed above are all lead free.

For details on the quality standards and fields of application, refer to the following document issued by Renesas: NEC Semiconductor Device Quality Standards (document reference number: C11531J). The former company name remains in the document but it is still a valid Renesas document.

1.6 Pin Connection Diagram (Top View)

μPD70F3506, μPD70F3507, μPD70F3508, μPD70F3509

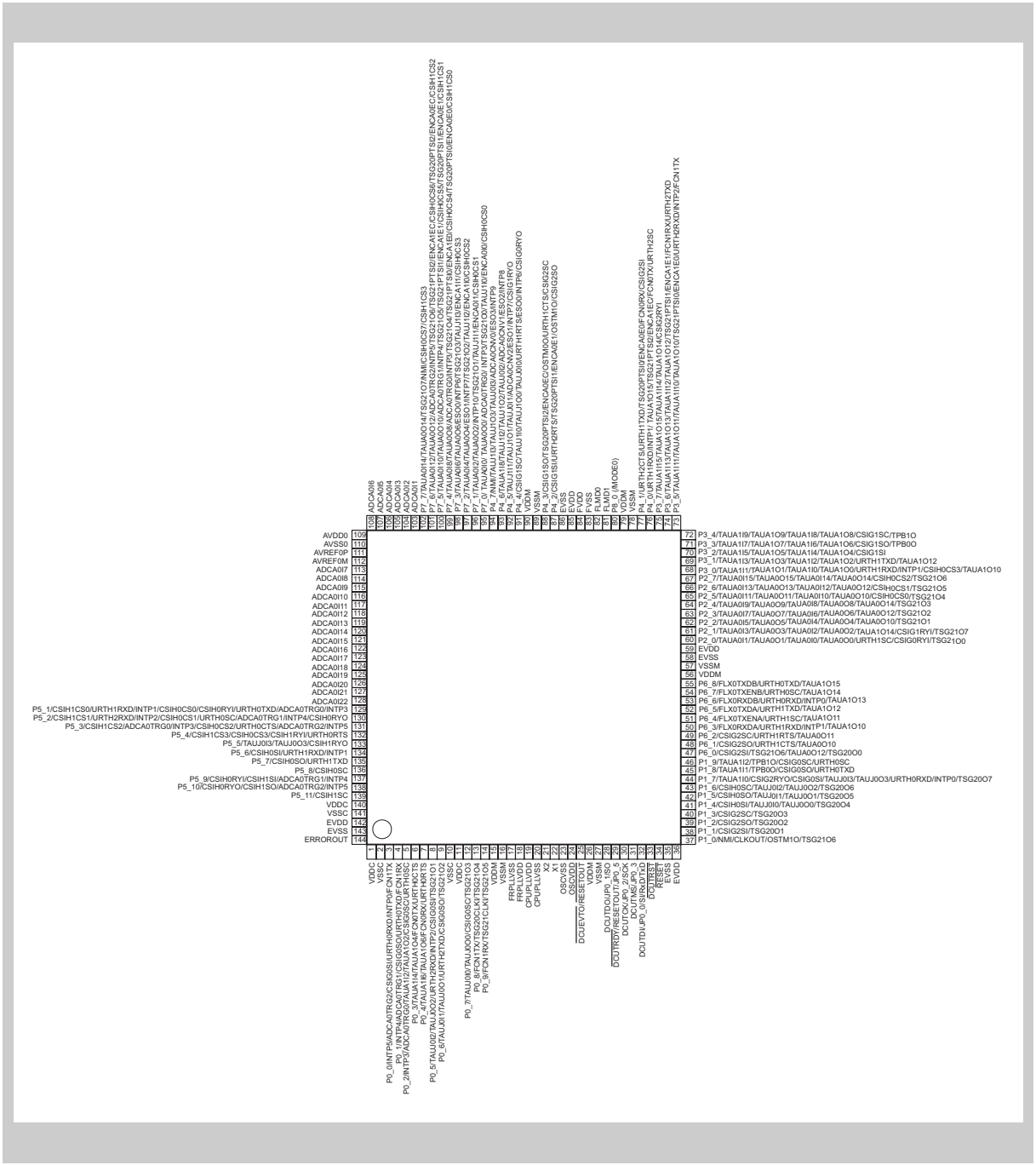


Table1-1 List of Pin Numbers and Names (1/4)

Pin Number	Pin Name	I/O
1	VDDC	V
2	VSSC	G
3	P0_0/INTP5/ADCA0TRG2/CSIG0SI/URTH0RXD/INTP0/FCN1TX	I/O
4	P0_1/INTP4/ADCA0TRG1/CSIG0SO/URTH0TXD/FCN1RX	I/O
5	P0_2/INTP3/ADCA0TRG0/TAUA1I2/TAUA1O2/CSIG0SC/URTH0SC	I/O
6	P0_3/TAUA1I4/TAUA1O4/FCN0TX/URTH0CTS	I/O
7	P0_4/TAUA1I6/TAUA1O6/FCN0RX/URTH0RTS	I/O
8	P0_5/TAUJ0I2/TAUJ0O2/URTH2RXD/INTP2/CSIG0SI/TSG21O1	I/O
9	P0_6/TAUJ0I1/TAUJ0O1/URTH2TXD/CSIG0SO/TSG21O2	I/O
10	VSSC	G
11	VDDC	V
12	P0_7/TAUJ0I0/TAUJ0O0/CSIG0SC/TSG21O3	I/O
13	P0_8/FCN1TX/TSG20CLKI/TSG21O4	I/O
14	P0_9/FCN1RX/TSG21CLKI/TSG21O5	I/O
15	VDDM	V
16	VSSM	G
17	FRPLLVSS	G
18	FRPLLVDD	V
19	CPUPLLVDD	V
20	CPUPLLVSS	G
21	X2	O
22	X1	I
23	OSCVSS	G
24	OSCVDD	V
25	$\overline{\text{DCUEVTO}}/\overline{\text{RESETOUT}}$	O
26	VDDM	V
27	VSSM	G
28	DCUTDO/JP0_1/ SO	O
29	$\overline{\text{DCUTRDY}}$/RESETOUT/JP0_5	O
30	DCUTCK/JP0_2/SCK	I
31	DCUTMS/JP0_3	I
32	DCUTDI/JP0_0/SI/RxD/TxD	I
33	$\overline{\text{DCUTRST}}$	I
34	$\overline{\text{RESET}}$	I
35	EVSS	G
36	EVDD	V
37	P1_0/NMI/CLKOUT/OSTM1O/TSG21O6	I/O
38	P1_1/CSIG2SI/TSG20O1	I/O

Table1-1 List of Pin Numbers and Names (2/4)

Pin Number	Pin Name	I/O
39	P1_2/CSIG2SO/TSG20O2	I/O
40	P1_3/CSIG2SC/TSG20O3	I/O
41	P1_4/CSIH0SI/TAUJ0I0/TAUJ0O0/TSG20O4	I/O
42	P1_5/CSIH0SO/TAUJ0I1/TAUJ0O1/TSG20O5	I/O
43	P1_6/CSIH0SC/TAUJ0I2/TAUJ0O2/TSG20O6	I/O
44	P1_7/TAUA1I0/CSIG2RYO/CSIG0SI/TAUJ0I3/TAUJ0O3/URTH0RXD/INTP0/TSG20O7	I/O
45	P1_8/TAUA1I1/TPB0O/CSIG0SO/URTH0TXD	I/O
46	P1_9/TAUA1I2/TPB1O/CSIG0SC/URTH0SC	I/O
47	P6_0/CSIG2SI/TSG21O6/TAUA0O12/TSG20O0	I/O
48	P6_1/CSIG2SO/URTH1CTS/TAUA0O10	I/O
49	P6_2/CSIG2SC/URTH1RTS/TAUA0O11	I/O
50	P6_3/FLX0RXDA/URTH1RXD/INTP1/TAUA1O10	I/O
51	P6_4/FLX0TXENA/URTH1SC/TAUA1O11	I/O
52	P6_5/FLX0TXDA/URTH1TXD/TAUA1O12	I/O
53	P6_6/FLX0RXDB/URTH0RXD/INTP0/TAUA1O13	I/O
54	P6_7/FLX0TXENB/URTH0SC/TAUA1O14	I/O
55	P6_8/FLX0TXDB/URTH0TXD/TAUA1O15	I/O
56	VDDM	V
57	VSSM	G
58	EVSS	G
59	EVDD	V
60	P2_0/TAUA0I1/TAUA0O1/TAUA0I0/TAUA0O0/URTH1SC/CSIG0RYI/TSG21O0	I/O
61	P2_1/TAUA0I3/TAUA0O3/TAUA0I2/TAUA0O2/TAUA1O14/CSIG1RYI/TSG21O7	I/O
62	P2_2/TAUA0I5/TAUA0O5/TAUA0I4/TAUA0O4/TAUA0O10/TSG21O1	I/O
63	P2_3/TAUA0I7/TAUA0O7/TAUA0I6/TAUA0O6/TAUA0O12/TSG21O2	I/O
64	P2_4/TAUA0I9/TAUA0O9/TAUA0I8/TAUA0O8/TAUA0O14/TSG21O3	I/O
65	P2_5/TAUA0I11/TAUA0O11/TAUA0I10/TAUA0O10/CSIH0CS0/TSG21O4	I/O
66	P2_6/TAUA0I13/TAUA0O13/TAUA0I12/TAUA0O12/CSIH0CS1/TSG21O5	I/O
67	P2_7/TAUA0I15/TAUA0O15/TAUA0I14/TAUA0O14/CSIH0CS2/TSG21O6	I/O
68	P3_0/TAUA1I1/TAUA1O1/TAUA1I0/TAUA1O0/URTH1RXD/INTP1/CSIH0CS3/TAUA1O10	I/O
69	P3_1/TAUA1I3/TAUA1O3/TAUA1I2/TAUA1O2/URTH1TXD/TAUA1O12	I/O
70	P3_2/TAUA1I5/TAUA1O5/TAUA1I4/TAUA1O4/CSIG1SI	I/O
71	P3_3/TAUA1I7/TAUA1O7/TAUA1I6/TAUA1O6/CSIG1SO/TPB0O	I/O
72	P3_4/TAUA1I9/TAUA1O9/TAUA1I8/TAUA1O8/CSIG1SC/TPB1O	I/O
73	P3_5/TAUA1I11/TAUA1O11/TAUA1I10/TAUA1O10/TSG21PTSI0/ENCA1E0/URTH2RXD/ INTP2/FCN1TX	I/O
74	P3_6/TAUA1I13/TAUA1O13/TAUA1I12/TAUA1O12/TSG21PTSI1/ENCA1E1/FCN1RX/ URTH2TXD	I/O
75	P3_7/TAUA1I15/TAUA1O15/TAUA1I14/TAUA1O14/CSIG2RYI	I/O

Table1-1 List of Pin Numbers and Names (3/4)

Pin Number	Pin Name	I/O
76	P4_0/URTH1RXD/INTP1/TAUA1O15/TSG21PTSI2/ENCA1EC/FCN0TX/URTH2SC	I/O
77	P4_1/URTH2CTS/URTH1TXD/TSG20PTSI0/ENCA0E0/FCN0RX/CSIG2SI	I/O
78	VSSM	G
79	VDDM	V
80	P8_0 (/MODE0)	I/O
81	FLMD1	I
82	FLMD0	I
83	FVSS	G
84	FVDD	V
85	EVDD	V
86	EVSS	G
87	P4_2/CSIG1SI/URTH2RTS/TSG20PTSI1/ENCA0E1/OSTM1O/CSIG2SO	I/O
88	P4_3/CSIG1SO/TSG20PTSI2/ENCA0EC/OSTM0O/URTH1CTS/CSIG2SC	I/O
89	VSSM	G
90	VDDM	V
91	P4_4/CSIG1SC/TAUJ1I0/TAUJ1O0/TAUJ0I0/URTH1RTS/ESO0/INTP6/CSIG0RYO	I/O
92	P4_5/TAUJ1I1/TAUJ1O1/TAUJ0I1/ADCA0CNV2/ESO1/INTP7/CSIG1RYO	I/O
93	P4_6/TAUA1I8/TAUJ1I2/TAUJ1O2/TAUJ0I2/ADCA0CNV1/ESO2/INTP8	I/O
94	P4_7/NMI/TAUJ1I3/TAUJ1O3/TAUJ0I3/ADCA0CNV0/ESO3/INTP9	I/O
95	P7_0/TAUA0I0/TAUA0O0/ADCA0TRG0/INTP3/TSG21O0/TAUJ1I0/ENCA0I0/CSIH0CS0	I/O
96	P7_1/TAUA0I2/TAUA0O2/INTP10/TSG21O1/TAUJ1I1/ENCA0I1/CSIH0CS1	I/O
97	P7_2/TAUA0I4/TAUA0O4/ESO1/INTP7/TSG21O2/TAUJ1I2/ENCA1I0/CSIH0CS2	I/O
98	P7_3/TAUA0I6/TAUA0O6/ESO0/INTP6/TSG21O3/TAUJ1I3/ENCA1I1/CSIH0CS3	I/O
99	P7_4/TAUA0I8/TAUA0O8/ADCA0TRG0/INTP3/TSG21O4/TSG21PTSI0/ENCA1E0/ CSIH0CS4/TSG20PTSI0/ENCA0E0/CSIH1CS0	I/O
100	P7_5/TAUA0I10/TAUA0O10/ADCA0TRG1/INTP4/TSG21O5/TSG21PTSI1/ENCA1E1/ CSIH0CS5/TSG20PTSI1/ENCA0E1/CSIH1CS1	I/O
101	P7_6/TAUA0I12/TAUA0O12/ADCA0TRG2/INTP5/TSG21O6/TSG21PTSI2/ENCA1EC/ CSIH0CS6/TSG20PTSI2/ENCA0EC/CSIH1CS2	I/O
102	P7_7/TAUA0I14/TAUA0O14/TSG21O7/NMI/CSIH0CS7/CSIH1CS3	I/O
103	ADCA0I1	I
104	ADCA0I2	I
105	ADCA0I3	I
106	ADCA0I4	I
107	ADCA0I5	I
108	ADCA0I6	I
109	AVDD0	V
110	AVSS0	G
111	AVREF0P	I

Table1-1 List of Pin Numbers and Names (4/4)

Pin Number	Pin Name	I/O
112	AVREF0M	I
113	ADCA0I7	I
114	ADCA0I8	I
115	ADCA0I9	I
116	ADCA0I10	I
117	ADCA0I11	I
118	ADCA0I12	I
119	ADCA0I13	I
120	ADCA0I14	I
121	ADCA0I15	I
122	ADCA0I16	I
123	ADCA0I17	I
124	ADCA0I18	I
125	ADCA0I19	I
126	ADCA0I20	I
127	ADCA0I21	I
128	ADCA0I22	I
129	P5_1/CSIH1CS0/URTH1RXD/INTP1/CSIH0CS0/CSIH0RYI/URTH0TXD/ADCA0TRG0/ INTP3	I/O
130	P5_2/CSIH1CS1/URTH2RXD/INTP2/CSIH0CS1/URTH0SC/ADCA0TRG1/INTP4/ CSIH0RYO	I/O
131	P5_3/CSIH1CS2/ADCA0TRG0/INTP3/CSIH0CS2/URTH0CTS/ADCA0TRG2/INTP5	I/O
132	P5_4/CSIH1CS3/CSIH0CS3/CSIH1RYI/URTH0RTS	I/O
133	P5_5/TAUJ0I3/TAUJ0O3/CSIH1RYO	I/O
134	P5_6/CSIH0SI/URTH1RXD/INTP1	I/O
135	P5_7/CSIH0SO/URTH1TXD	I/O
136	P5_8/CSIH0SC	I/O
137	P5_9/CSIH0RYI/CSIH1SI/ADCA0TRG1/INTP4	I/O
138	P5_10/CSIH0RYO/CSIH1SO/ADCA0TRG2/INTP5	I/O
139	P5_11/CSIH1SC	I/O
140	VDDC	V
141	VSSC	G
142	EVDD	V
143	EVSS	G
144	ERROROUT	O

Pin Name	
ADCA0I1 - ADCA0I22	**:** Analog input
ADCA0TRG0 - ADCA0TRG2	**:** ADC trigger input
ADCA0CNV0 - ADCA0CNV2	**:** ADC status output
AVDD0	**:** Analog power supply
AVREF0P	**:** Analog reference voltage (positive)
AVREF0M	**:** Analog reference voltage (minimal)
AVSS0	**:** Analog ground
CLKOUT	**:** Clock output
CPUPLLVDD	**:** CPU PLL power supply
CPUPLLVSS	**:** CPU PLL ground
CSIG0SC - CSIG2SC	**:** Serial clock
CSIG0SI - CSIG2SI	**:** Serial input
CSIG0SO - CSIG2SO	**:** Serial output
CSIG0RYI - CSIG2RYI	**:** Serial ready/busy input signal
CSIG0RYO - CSIG2RYO	**:** Serial ready/busy output signal
CSIH0SC, CSIH1SC	**:** Serial clock
CSIH0SI, CSIH1SI	**:** Serial input
CSIH0SO, CSIH1SO	**:** Serial output
CSIH0RYI, CSIH1RYI	**:** Serial ready/busy input signal
CSIH0RYO, CSIH1RYO	**:** Serial ready/busy output signal
CSIH0CS0 - CSIH0CS7	**:** Serial chip select output signal
CSIH1CS0 - CSIH1CS3	**:** Serial chip select output signal
$\overline{\text{DCUEVTO}}$	**:** NEXUS event output
DCUTCK	**:** NEXUS input clock
DCUTDI	**:** NEXUS input
DCUTDO	**:** NEXUS output
DCUTMS	**:** NEXUS mode select
$\overline{\text{DCUTRDY}}$	**:** NEXUS ready output
$\overline{\text{DCUTRST}}$	**:** NEXUS reset input
ENCA0I0, ENCA1I0, ENCA0I1, ENCA1I1	**:** Timer encoder capture input
ENCA0E0, ENCA1E0, ENCA0E1, ENCA1E1	**:** Timer encoder count pulse input
ENCA0EC, ENCA1EC	**:** Timer encoder clear input
ERROROUT	**:** Error output signal
ESO0-ESO3	**:** Timer output compulsion stop input

EVDD : I/O Power supply

EVSS : I/O Ground

FCN0RX, FCN1RX : Receive data for controller area network

FCN0TX, FCN1TX : Transmit data for controller area network

FLX0RXDA, FLX0RXDB : FlexRay receive data

FLX0TXDA, FLX0TXDB : FlexRay transmit data

FLX0TXENA, FLX0TXENB : FlexRay request to send (output)

FLMD0 : Flash programming mode

FLMD1 : Flash programming mode

FRPLLVDD : FlexRay PLL power supply

FRPLLVSS : FlexRay PLL ground

FVDD : Code flash/data flash memory power supply

FVSS : Code flash/data flash ground

INTP0 - INTP10 : External interrupt input

MODE0 : Mode

NMI : Non-maskable interrupt input

OSTM0O, OSTM1O : Operation system timer output

OSCVDD : OSC power supply

OSCVSS : OSC ground

P0_0 - P0_9 : Port 0

P1_0 - P1_9 : Port 1

P2_0 - P2_7 : Port 2

P3_0 - P3_7 : Port 3

P4_0 - P4_7 : Port 4

P5_1 - P5_11 : Port 5

P6_0 - P6_8 : Port 6

P7_0 - P7_7 : Port 7

P8_0 : Port 8

JP0_0 - JP0_3, JP0_5 : Port JP0 (JTAG)

$\overline{\text{RESET}}$: Reset input

RESETOUT, RESETOUT : Reset output

RxD : Receive data for flash writing in (UART)

SCK : Serial clock for flash writing in (CSI)

SI : Serial input for flash writing in (CSI)

SO : Serial output for flash writing in (CSI)

TAUA0I0 - TAUA0I15, TAUA1I0 - TAUA1I15, TAUJ0I0 - TAUJ0I3, TAUJ1I0 - TAUJ1I3	: Timer input
TAUA0O0 - TAUA0O15, TAUA1O0 - TAUA1O15, TAUJ0O0 - TAUJ0O3, TAUJ1O0 - TAUJ1O3	: Timer output
TPB0O, TPB1O	: Timer pattern buffer pulse output
TSG20O0 - TSG20O7, TSG21O0 - TSG21O7	: Timer pulse output
TSG20PTSI0 - TSG20PTSI2, TSG21PTSI0 - TSG21PTSI2	: Timer pattern input
TSG20CLKI, TSG21CLKI	: Timer count enable input signal
TxD	: Transmit data for flash writing in (UART)
URTH0RXD - URTH2RXD	: Receive data
URTH0TXD - URTH2TXD	: Transmit data
URTH0SC - URTH2SC	: UART input clock
URTH0CTS - URTH2CTS	: UART clear to send (input)
URTH0RTS - URTH2RTS	: UART request to send (output)
VDDC	: SUB CPU power supply
VSSC	: SUB CPU ground
VDDM	: Main CPU power supply
VSSM	: Main CPU ground
X1, X2	: Crystal

μPD70F3505A

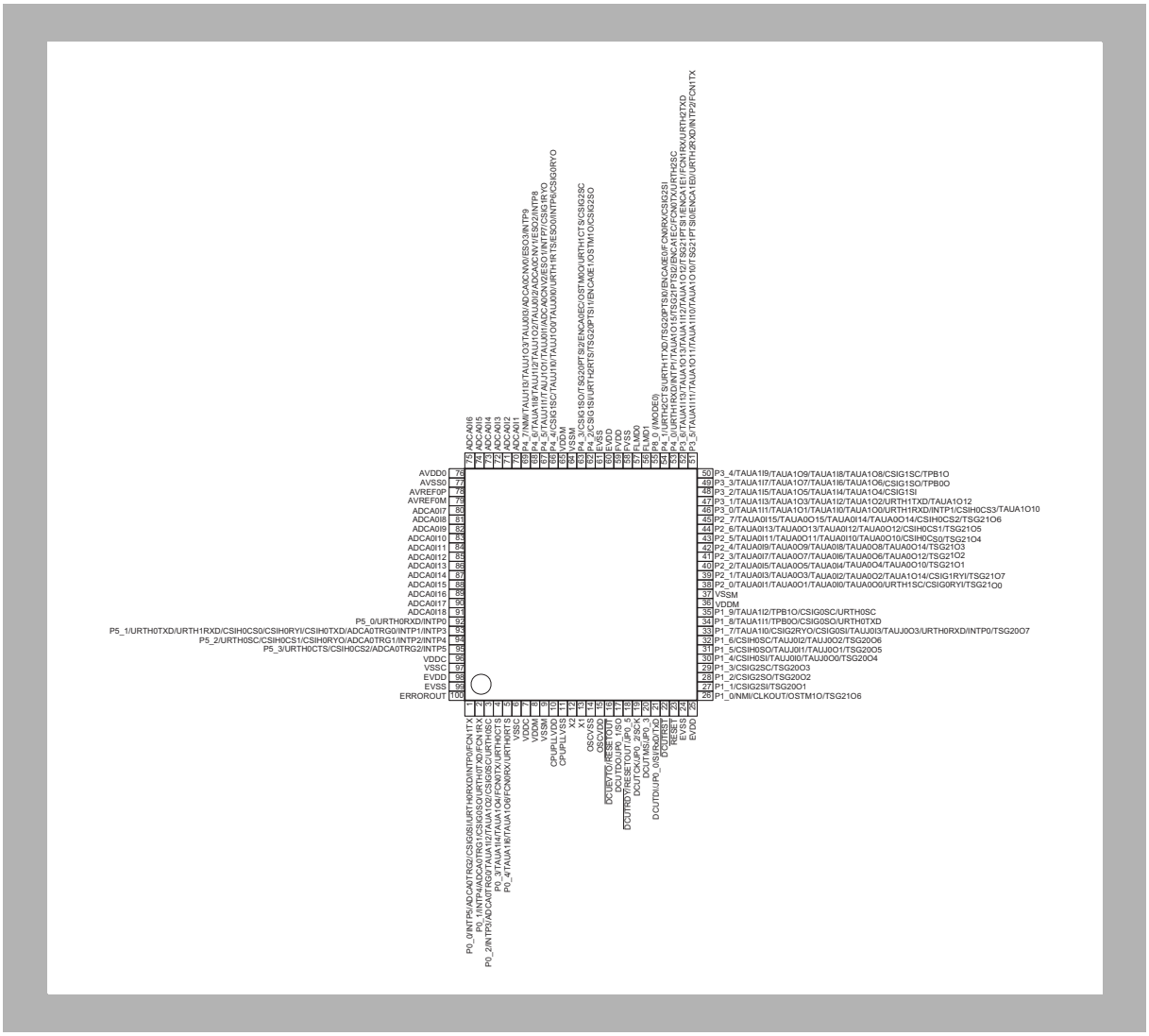


Table1-2 List of Pin Numbers and Names (1/3)

Pin Number	Pin Name	I/O
1	P0_0/INTP5/ADCA0TRG2/CSIG0SI/URTH0RXD/INTP0/FCN1TX	I/O
2	P0_1/INTP4/ADCA0TRG1/CSIG0SO/URTH0TXD/FCN1RX	I/O
3	P0_2/INTP3/ADCA0TRG0/TAUA1I2/TAUA1O2/CSIG0SC/URTH0SC	I/O
4	P0_3/TAUA1I4/TAUA1O4/FCN0TX/URTH0CTS	I/O
5	P0_4/TAUA1I6/TAUA1O6/FCN0RX/URTH0RTS	I/O
6	VSSC	G
7	VDDC	V
8	VDDM	V
9	VSSM	G
10	CPUPLLVDD	V
11	CPUPLLVSS	G
12	X2	O
13	X1	I
14	OSCVSS	G
15	OSCVDD	V
16	$\overline{\text{DCUEVTO}}$/$\overline{\text{RESETOUT}}$	O
17	DCUTDO/JP0_1/SO	I/O
18	$\overline{\text{DCUTRDY}}$/RESETOUT/JP0_5	I/O
19	DCUTCK/JP0_2/SCK	I/O
20	DCUTMS/JP0_3	I/O
21	DCUTDI/JP0_0/SI/ RxD/TxD	I/O
22	$\overline{\text{DCUTRST}}$	I
23	$\overline{\text{RESET}}$	I
24	EVSS	G
25	EVDD	V
26	P1_0/NMI/CLKOUT/OSTM1O/TSG21O6	I/O
27	P1_1/CSIG2SI/TSG20O1	I/O
28	P1_2/CSIG2SO/TSG20O2	I/O
29	P1_3/CSIG2SC/TSG20O3	I/O
30	P1_4/CSIH0SI/TAUJ0I0/TAUJ0O0/TSG20O4	I/O
31	P1_5/CSIH0SO/TAUJ0I1/TAUJ0O1/TSG20O5	I/O
32	P1_6/CSIH0SC/TAUJ0I2/TAUJ0O2/TSG20O6	I/O
33	P1_7/TAUA1I0/CSIG2RYO/CSIG0SI/TAUJ0I3/TAUJ0O3/URTH0RXD/INTP0/TSG20O7	I/O
34	P1_8/TAUA1I1/TPB0O/CSIG0SO/URTH0TXD	I/O
35	P1_9/TAUA1I2/TPB1O/CSIG0SC/URTH0SC	I/O
36	VDDM	V
37	VSSM	G
38	P2_0/TAUA0I1/TAUA0O1/TAUA0I0/TAUA0O0/URTH1SC/CSIG0RYI/TSG21O0	I/O

Table1-2 List of Pin Numbers and Names (2/3)

Pin Number	Pin Name	I/O
39	P2_1/TAUA0I3/TAUA0O3/TAUA0I2/TAUA0O2/TAUA1O14/CSIG1RYI/TSG21O7	I/O
40	P2_2/TAUA0I5/TAUA0O5/TAUA0I4/TAUA0O4/TAUA0O10/TSG21O1	I/O
41	P2_3/TAUA0I7/TAUA0O7/TAUA0I6/TAUA0O6/TAUA0O12/TSG21O2	I/O
42	P2_4/TAUA0I9/TAUA0O9/TAUA0I8/TAUA0O8/TAUA0O14/TSG21O3	I/O
43	P2_5/TAUA0I11/TAUA0O11/TAUA0I10/TAUA0O10/CSIH0CS0/TSG21O4	I/O
44	P2_6/TAUA0I13/TAUA0O13/TAUA0I12/TAUA0O12/CSIH0CS1/TSG21O5	I/O
45	P2_7/TAUA0I15/TAUA0O15/TAUA0I14/TAUA0O14/CSIH0CS2/TSG21O6	I/O
46	P3_0/TAUA1I1/TAUA1O1/TAUA1I0/TAUA1O0/URTH1RXD/INTP1/CSIH0CS3/TAUA1O10	I/O
47	P3_1/TAUA1I3/TAUA1O3/TAUA1I2/TAUA1O2/URTH1TXD/TAUA1O12	I/O
48	P3_2/TAUA1I5/TAUA1O5/TAUA1I4/TAUA1O4/CSIG1SI	I/O
49	P3_3/TAUA1I7/TAUA1O7/TAUA1I6/TAUA1O6/CSIG1SO/TPB0O	I/O
50	P3_4/TAUA1I9/TAUA1O9/TAUA1I8/TAUA1O8/CSIG1SC/TPB1O	I/O
51	P3_5/TAUA1I11/TAUA1O11/TAUA1I10/TAUA1O10/TSG21PTSI0/ENCA1E0/URTH2RXD/ INTP2/FCN1TX	I/O
52	P3_6/TAUA1I13/TAUA1O13/TAUA1I12/TAUA1O12/TSG21PTSI1/ENCA1E1/FCN1RX/ URTH2TXD	I/O
53	P4_0/URTH1RXD/INTP1/TAUA1O15/TSG21PTSI2/ENCA1EC/FCN0TX/URTH2SC	I/O
54	P4_1/URTH2CTS/URTH1TXD/TSG20PTSI0/ENCA0E0/FCN0RX/CSIG2SI	I/O
55	P8_0 (/MODE0)	I/O
56	FLMD1	I
57	FLMD0	I
58	FVSS	G
59	FVDD	V
60	EVDD	V
61	EVSS	G
62	P4_2/CSIG1SI/URTH2RTS/TSG20PTSI1/ENCA0E1/OSTM1O/CSIG2SO	I/O
63	P4_3/CSIG1SO/TSG20PTSI2/ENCA0EC/OSTM0O/URTH1CTS/CSIG2SC	I/O
64	VSSM	G
65	VDDM	V
66	P4_4/CSIG1SC/TAUJ1I0/TAUJ1O0/TAUJ0I0/URTH1RTS/ESO0/INTP6/CSIG0RYO	I/O
67	P4_5/TAUJ1I1/TAUJ1O1/TAUJ0I1/ADCA0CNV2/ESO1/ INTP7/CSIG1RYO	I/O
68	P4_6/TAUA1I8/TAUJ1I2/TAUJ1O2/TAUJ0I2/ADCA0CNV1/ESO2/INTP8	I/O
69	P4_7/NMI/TAUJ1I3/TAUJ1O3/TAUJ0I3/ADCA0CNV0/ESO3/INTP9	I/O
70	ADCA0I1	I
71	ADCA0I2	I
72	ADCA0I3	I
73	ADCA0I4	I
74	ADCA0I5	I
75	ADCA0I6	I

Table1-2 List of Pin Numbers and Names (3/3)

Pin Number	Pin Name	I/O
76	AVDD0	V
77	AVSS0	G
78	AVREF0P	I
79	AVREF0M	I
80	ADCA0I7	I
81	ADCA0I8	I
82	ADCA0I9	I
83	ADCA0I10	I
84	ADCA0I11	I
85	ADCA0I12	I
86	ADCA0I13	I
87	ADCA0I14	I
88	ADCA0I15	I
89	ADCA0I16	I
90	ADCA0I17	I
91	ADCA0I18	I
92	P5_0/URTH0RXD/INTP0	I/O
93	P5_1/URTH0TXD/URTH1RXD/CSIH0CS0/CSIH0RYI/ADCA0TRG0/INTP1/INTP3	I/O
94	P5_2/URTH0SC/CSIH0CS1/CSIH0RYO/ADCA0TRG1/INTP2/INTP4	I/O
95	P5_3/URTH0CTS/CSIH0CS2/ADCA0TRG2/INTP5	I/O
96	VDDC	V
97	VSSC	G
98	EVDD	V
99	EVSS	G
100	ERROROUT	O

Pin Name	
ADCA0I1 - ADCA0I18	**:** Analog input
ADCA0TRG0 - ADCA0TRG2	**:** ADC trigger input
ADCA0CNV0 - ADCA0CNV2	**:** ADC status output
AVDD0	**:** Analog power supply
AVREF0P	**:** Analog reference voltage (positive)
AVREF0M	**:** Analog reference voltage (minimal)
AVSS0	**:** Analog ground
CLKOUT	**:** Clock output
CPUPLLVDD	**:** CPU PLL power supply
CPUPLLVSS	**:** CPU PLL ground
CSIG0SC - CSIG2SC	**:** Serial clock
CSIG0SI - CSIG2SI	**:** Serial input
CSIG0SO - CSIG2SO	**:** Serial output
CSIG0RYI - CSIG1RYI	**:** Serial ready/busy input signal
CSIG0RYO - CSIG2RYO	**:** Serial ready/busy output signal
CSIH0SC	**:** Serial clock
CSIH0SI	**:** Serial input
CSIH0SO	**:** Serial output
CSIH0CS0 - CSIH0CS3	**:** Serial chip select output signal
CSIH0RYI	**:** Serial ready/busy input signal
CSIH0RYO	**:** Serial ready/busy output signal
$\overline{\text{DCUEVTO}}$	**:** NEXUS event output
DCUTCK	**:** NEXUS input clock
DCUTDI	**:** NEXUS input
DCUTDO	**:** NEXUS output
DCUTMS	**:** NEXUS mode select
$\overline{\text{DCUTRDY}}$	**:** NEXUS ready output
$\overline{\text{DCUTRST}}$	**:** NEXUS reset input
ENCA0E0, ENCA1E0, ENCA0E1, ENCA1E1	**:** Timer encoder count pulse input
ENCA0EC, ENCA1EC	**:** Timer encoder clear input
ERROROUT	**:** Error output signal
ESO0 - ESO3	**:** Timer output compulsion stop input
EVDD	**:** I/O power supply
EVSS	**:** I/O ground
FCN0RX, FCN1RX	**:** Receive data for controller area network
FCN0TX, FCN1TX	**:** Transmit data for controller area network

FLMD0 : Flash programming mode

FLMD1 : Flash programming mode

FVDD : Code flash/data flash memory power supply

FVSS : Code flash/data flash ground

INTP0 - INTP9 : External interrupt input

JP0_0 - JP0_3, JP0_5 : Port JP0 (JTAG)

MODE0 : Mode

NMI : Non-maskable interrupt input

OSTM0O, OSTM1O : Operation system timer output

OSCVDD : OSC power supply

OSCVSS : OSC ground

P0_0 - P0_4 : Port 0

P1_0 - P1_9 : Port 1

P2_0 - P2_7 : Port 2

P3_0 - P3_6 : Port 3

P4_0 - P4_7 : Port 4

P5_0 - P5_3 : Port 5

P8_0 : Port 8

$\overline{\text{RESET}}$: Reset input

RESETOUT : Reset output

RxD : Receive data for flash writing in (UART)

SCK : Serial clock for flash writing in (CSI)

SI : Serial input for flash writing in (CSI)

SO : Serial output for flash writing in (CSI)

TAUA0I0 - TAUA0I15, TAUA1I0 - TAUA1I13, TAUJ0I0 - TAUJ0I3, TAUJ1I0 - TAUJ1I3 : Timer input

TAUA0O0 - TAUA0O15, TAUA1O0 - TAUA1O15, TAUJ0O0 - TAUJ0O3, TAUJ1O0 - TAUJ1O3 : Timer output

TPB0O,TPB1O : Timer pattern buffer pulse output

TSG20O1 - TSG20O7, TSG21O0 - TSG21O7 : Timer pulse output

TSG20PTSI0 - TSG20PTSI2, TSG21PTSI0 - TSG21PTSI2 : Timer pattern input

TxD : Transmit data for flash writing in (UART)

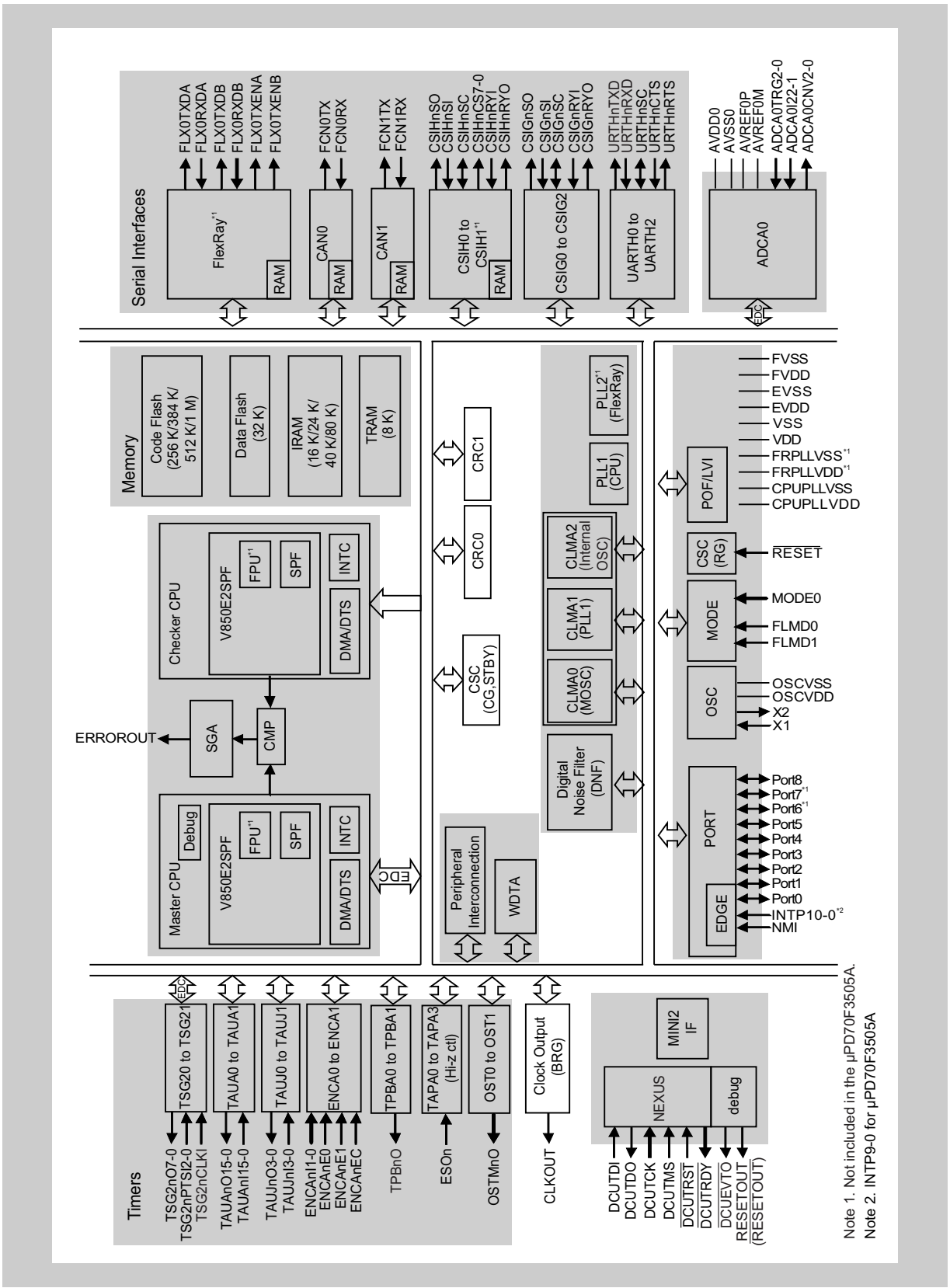
URTH0RXD - URTH2RXD : Receive data

URTH0TXD - URTH2TXD : Transmit data

URTH0SC - URTH2SC	**:** UART input clock
URTH0CTS - URTH2CTS	**:** UART clear to send (input)
URTH0RTS - URTH2RTS	**:** UART request to send (output)
VDDC	**:** SUB CPU power supply
VSSC	**:** SUB CPU ground
VDDM	**:** Main CPU power supply
VSSM	**:** Main CPU ground
X1, X2	**:** Crystal

1.7 Configuration of Functional Blocks

1.7.1 Internal Block Diagram



1.7.2 Internal Units

(1) CPU

Processing of almost all instructions is executed in a single clock cycle under 7-stage pipeline control.

The CPU includes a multiplier (16 bits × 16 bits → 32 bits, or 32 bits × 32 bits → 64 bits) and barrel shifter (32 bits) to speed up complicated instruction processing.

(2) DMA, DTS Controller (DMAC, DTS)

These controllers transfer data between memory and peripheral I/O with 2-cycle transfer. Each module has two transfer modes: single transfer and single step transfer (DMA)/block transfer (DTS).

(3) ROM

The ROM consists of code flash memory and data flash memory. Memory capacity per product is given in the following table.

The CPU is able to access ROM in a single clock cycle when fetching instructions.

Table1-3 Memory Capacity and Addresses of Code Flash and Data Flash

Product Name	Data Flash Memory Capacity / Code Flash Memory Capacity	Address Range
µPD70F3505A	32 Kbytes	0200_0000_H to 0200_7FFF_H
	384 Kbytes	0000_0000_H to 0005_FFFF_H
µPD70F3506, µPD70F3507	32 Kbytes	0200_0000_H to 0200_7FFF_H
	512 Kbytes	0000_0000_H to 0007_FFFF_H
µPD70F3508, µPD70F3509	32 Kbytes	0200_0000_H to 0200_7FFF_H
	1024 Kbytes	0000_0000_H to $000F_FFFF_H$

(4) RAM

Mappings of RAM per product are listed below.

Table1-4 RAM Capacities and Address Ranges

Product Name	RAM Capacity	Address Range
µPD70F3505A	24 Kbytes	$FEDF_A000_H$ to $FEDF_FFFF_H$
µPD70F3506, µPD70F3507	40 Kbytes	$FEDF_6000_H$ to $FEDF_FFFF_H$
µPD70F3508, µPD70F3509	80 Kbytes	$FEDE_C000_H$ to $FEDF_FFFF_H$

(5) Interrupt Controller (INTC)

This module handles peripheral I/O and external hardware interrupt requests (µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509: INTP10-0, µPD70F3505A: INTP9-0). These interrupts can be prioritized at 16 levels to control multiple forms of handling for the interrupt sources.

(6) Clock Generation

An external oscillator is connected to the X1 and X2 pins as an input clock. The signal from the oscillator is input to the PLL1 synthesizer for frequency multiplying (the multiple is switched by optional bytes). The PLL1 output is supplied as the internal system clock and the frequency-divided clock is supplied as the peripheral clock. Moreover, the PLL2 output is used as the FlexRay clock (The µPD70F3505A does not include FlexRay). Set operating frequency for PCLK at 64 MHz or less to use FlexRay (only applicable for the µPD70F3506 and µPD70F3507).

(7) Timer Units

Timer units include 16-bit timer array unit A (TAUA), 32-bit timer array unit J (TAUJ), timer S (TSG2), the encoder timer (ENCA), timer pattern buffer (TPBA), and OS timer (OST), and can provide capturing and comparison, measurement of pulse intervals and frequency, and programmable pulse output.

(8) Timer Option Units (TAPA)

These units control forcible stopping of the output from timers (TAUA, TAUJ, TSG2).

(9) Peripheral Interconnection (PIC)

Interlocking operation is available by connecting to a timer for simultaneous starting or peripheral I/O.

(10) Serial Interface (SIO)

Serial interfaces include synchronous/asynchronous interface H (UARTH), clocked serial interface (CSIG), clocked serial interface with queue (CSIH), CAN and FlexRay (FLX0).
UARTH executes data transfer through the URTHnTXD and URTHnRXD pins (n = 0 to 2).
CSIG executes data transfer through the CSIGnSO, CSIGnSI, and CSIGnSC pins (n = 0 to 2).
CSIH executes data transfer through the CSIHnSO, CSIHnSI, and CSIHnSC pins (n = 0 to 1; with the µPD70F3505A, n = 0).
CAN executes data transfer through the FCNnRX and FCNnTX pins (n = 0, 1).
FlexRay executes data transfer through the FLX0TXDA, FLX0TXDB, FLX0RXDA, and FLX0RXDB pins (The µPD70F3505A does not include FlexRay).

(11) Data CRC Function

Data CRC (cyclic redundancy checking) is used to verify or generate data streams protected by CRC with different widths in bits for various lengths.

(12) CLKOUT Function

A baud rate generator is included to allow output of desired frequencies from the CLKOUT pin.

(13) Nexus Function

A single Nexus channel is included as a debugging interface.

(14) A/D Converter (ADCA0)

The high-speed and high-resolution 10-bit and 12-bit A/D converters have the following numbers of analog input pins: 22 for the µPD70F3506, µPD70F3507, µPD70F3508, and µPD70F3509, 18 for the µPD70F3505A.

(15) Ports

Port pins capable of operation as general-purpose port pins and as control pins are listed below.

Port	I/O	Control Function
Port 0	Bitwise I/O	Serial interface (CSIG0) I/O Serial interface (UARTH0, CAN0, CAN1) I/O Serial interface (CAN0, CAN1) I/O Timer array unit A (TAUA1) I/O Timer array unit J (TAUJ0) I/O TSG2 (TSG20, TSG21) I/O External interrupt (INTP0, INTP3 to INTP5) input A/D converter (ADCA0) input
Port 1	Bitwise I/O	External interrupt (NMI, INTPO) input Baud rate generator (BRG0 (CLKOUT)) output Operating system timer (OST1) output Serial interface (CSIG2, CSIG0) I/O Serial interface (CSIH0) I/O Serial interface (UARTH0) I/O Timer array unit A (TAUA1) input Timer array unit J (TAUJ0) I/O Timer pattern buffer (TPBA0, TPBA1) output TSG2 (TSG20, TSG21) output
Port 2	Bitwise I/O	Timer array unit A (TAUA0) I/O Serial interface (CSIG0, CSIG1) I/O Serial interface (CSIH0) I/O Serial interface (UARTH1) I/O TSG2 (TSG21) I/O
Port 3	Bitwise I/O	External interrupt (INTP1) input Timer array unit A (TAUA1) I/O Serial interface (CSIG1, CSIG2) I/O Serial interface (CSIH0) I/O Serial interface (UARTH1, UARTH2) I/O Serial interface (CAN1) I/O Timer pattern buffer (TPBA0, TPBA1) output TSG2 (TSG21) input Encoder timer (ENCA1) input
Port 4	Bitwise I/O	External interrupt (NMI, INTP1, INTP7 to INTP9) input Serial interface (CAN0) output Serial interface (CSIG0 to CSIG2) I/O Serial interface (UARTH1, UARTH2) I/O Timer array unit A (TAUA1) output Timer array unit J (TAUJ0, TAUJ1) I/O Encoder timer (ENCA0, ENCA1) input TSG2 (TSG20, TSG21) input A/D converter (ADCA0) input Timer option unit (TAPA0 to TAPA3) input
Port 5	Bitwise I/O	External interrupt (INTP1 to INTP5) input Serial interface (UARTH0, UARTH1, UARTH2) I/O Serial interface (CSIH0, CSIH1) I/O Timer array unit J (TAUJ0) I/O A/D converter (ADCA0) input

Port	I/O	Control Function
Port 6	Bitwise I/O	External interrupt (INTP0, INTP1) input Serial interface (CSIG2) I/O Serial interface (UARTH0, UARTH1) I/O Serial interface (FLX0) I/O Timer S (TSG20, TSG21) I/O Timer array unit A (TAUA0, TAUA1) output
Port 7	Bitwise I/O	External interrupt (NMI, INTP3 to INTP7, INTP10) input Serial interface (CSIH0, CSIH1) I/O Timer array unit A (TAUA0) I/O Timer array unit J (TAUJ1) output Timer S (TSG20, TSG21) I/O Encoder timer (ENCA0, ENCA1) input A/D converter (ADCA0) input Timer option unit (TAPA0, TAPA1) input
Port 8	Bitwise I/O	Mode input

Caution The µPD70F3505A does not have ports 6 and 7.

Section 2 Port Functions

This section contains a generic description of the port control functions.

2.1 Port Functions

Port groups This product has the following numbers of port groups.

Table 2-1 Port Groups

Port Groups		
Product name	µPD70F3505A	µPD70F3506, µPD70F3507, µPD70F3508, µPD70F3509
Number of port groups	7	9
Name	P0_0-P0_4, P1_0-P1_9, P2_0-P2_7 , P3_0-P3_6, P4_0-P4_7, P5_0-P5_3, P8_0	P0_0-P0_9, P1_0-P1_9, P2_0-P2_7, P3_0-P3_7, P4_0-P4_7, P5_1-P5_11, P6_0-P6_8, P7_0-P7_7, P8_0

Port groups index n Throughout this section, the individual port groups are identified by the index “n”, for example, PMCn for the port mode control register of Pn.

Register addresses All port n register addresses are given as address offsets from the individual base addresses <PORTn_base0> and <PORTn_base1>.

The base addresses <PORTn_base0> and <PORTn_base1> are specified in the following table.

Table 2-2 Port n Register Addresses <PORTn_base0> and <PORTn_base1>

<PORTn_base0> Address	<PORTn_base1> Address
FF40 0000_H	FFFF 8000_H

2.2 Overview

The microcontroller has various pins for input/output functions, known as ports.

The ports are organized in port groups.

The microcontroller also has several control registers to allocate non-general purpose input/output functions to the pins.

For a description of the terms pin, port, and port group, see section 2.2.1, Terms.

Features summary

- I/O ports
- Multiplexed with input/output pin functions for other peripheral modules
- Able to specify input/output in bit units
- Edge detection
- Software pull-up
- Noise cancellation

2.2.1 Terms

In this section, the following terms are used.

- **Pin**

  Denotes the physical pin. Every pin is denoted by a unique pin number.

  A pin can be switched to several modes for use.

  The pin function allocated to a pin depends on the selected mode.

- **Port group**

  Denotes a group of pins. The pins of a port group have a common set of port mode control registers.

- **Port mode/Port**

  A pin in port mode works as a general purpose input/output pin. It is then called “port”.

  The corresponding name is Pn_m. For example, P0_1 denotes port 1 of port group 0. It is referenced as “port P0_1”.

- **Alternative mode**

  In alternative mode, a pin can be used for in various non-general purpose input/output functions, for example as the input/output pin of on-chip peripherals.

  The corresponding pin name depends on the selected function.

  For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

  Note that two different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function in which the pin is being operated.

- **Port type**

  “Port type” is set by setting a configuration register for the given port.

2.2.2 Pin Function Configuration

The pins can operate in three different general modes.

- Port mode (PMCn.PMCn_m = 0)
  In port mode, the pin operates as a general purpose I/O port. PMn.PMn_m selects input or output.
- Software I/O control alternative mode (PMCn.PMCn_m = 1, PIPCn.PIPCn_m = 0)
  In software I/O control alternative mode, the pin is operated by an alternative function. The selection between input or output is done by software via the PMn.PMn_m control bits.
- Direct I/O control alternative mode (PMCn.PMCn_m = 1, PIPCn.PIPCn_m = 1)
  In direct I/O control alternative mode, the pin operates for an alternative function. In contrast to the software I/O control alternative mode, input or output are also controlled by the hardware for the alternative function, so software settings are not required for this.

An overview of the register settings is given in the tables below.

Table 2-3 Pin Function Configuration (Overview)

Function	Register			I/O
	PMCn_m	PMn_m	PIPCn_m	
Port mode (Output)	0	0	X	O
Port mode (Input)		1*		I
Alternative output modes 1 to 4	1	0	0	O
Alternative input modes 1 to 4		1	0	I
Direct I/O control		X	1	Controlled by alternative function

Note * The input buffer must be enabled. (PIBCn.PIBCn_m = 1)
X: Invalid setting

If a pin is operated in an alternative mode (PMCn.PMCn_m = 1), one out of up to four different alternative functions can be selected by the PFCn and PFCEn registers.

- Software I/O control alternative functions (PIPCn.PIPCn_m = 0)
  - Outputs (PMn_m = 0): ALT-OUT1 to ALT-OUT4
  - Inputs (PMn_m = 1): ALT-IN1 to ALT-IN4
- Direct I/O control alternative functions (PIPCn.PIPCn_m = 1)
  - Input/output of ALT-OUT1 to ALT-OUT4 and ALT-IN1 to ALT-IN4 is directly controlled by the alternative function (hardware).

Table 2-4 Alternative Mode Selection Overview (PMCn.PMCn_m = 1)

Function	Register				I/O
	PIPC*	PM*	PFCE	PFC	
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)		1			I
Alternative output mode 2 (ALT-OUT2)		0	0	1	O
Alternative input mode 2 (ALT-IN2)		1			I
Alternative output mode 3 (ALT-OUT3)		0	1	0	O
Alternative input mode 3 (ALT-IN3)		1			I
Alternative output mode 4 (ALT-OUT4)		0	1	1	O
Alternative input mode 4 (ALT-IN4)		1			I

Note * If PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the alternative function and PM is ignored.

Pins that have multiple functions multiplexed with a control mode (ALT-IN) and pins allocated to multiple port pins (excluding ESO2/INTP8 and ESO3/INTP9) are listed in the following table.

Caution 1. Do not use alternative function pins simultaneously (for example, ADCA0TRG0 and INTP3). However, edges for the INTP pins are detectable as the edges on the URTH0RXD, URTH1RXD, and URTH2RXD pins, respectively, in the case of the multiplexed pins for URTH0RXD/INTP0, URTH1RXD/INTP1, and URTH2RXD/INTP2.

Caution 2. Do not use the same alternative function pin with multiple port pins (For example, if ADCA0TRG0/INTP3 is in use, ALT_IN1 of P0_2, ALT_IN4 of P5_1, and so on).

Table 2-5 Pins on which a Single Control Mode and Multiple Alternate Functions are Multiplexed (1/2)

Alternative Function Pin	Port Pin	Control Mode
ADCA0TRG0/INTP3	P0_2	ALT-IN1
	P5_1	ALT-IN4
	P5_3	ALT-IN2
	P7_0	ALT-IN2
	P7_4	ALT-IN2
ADCA0TRG1/INTP4	P0_1	ALT-IN1
	P5_2	ALT-IN4
	P5_9	ALT-IN4
	P7_5	ALT-IN2
ADCA0TRG2/INTP5	P0_0	ALT-IN1
	P5_3	ALT-IN4
	P5_10	ALT-IN4
	P7_6	ALT-IN2
ESO0/INTP6	P4_4	ALT-IN4
	P7_3	ALT-IN2
ESO1/INTP7	P4_5	ALT-IN4
	P7_2	ALT-IN2
ESO2/INTP8	P4_6	ALT-IN4
ESO3/INTP9	P4_7	ALT-IN4
URTH0RXD/INTP0	P0_0	ALT-IN4
	P1_7	ALT-IN4
	P5_0	ALT-IN3
	P6_6	ALT-IN2
URTH1RXD/INTP1	P3_0	ALT-IN3
	P4_0	ALT-IN1
	P5_1	ALT-IN2
	P5_6	ALT-IN2
	P6_3	ALT-IN2
URTH2RXD/INTP2	P0_5	ALT-IN2
	P3_5	ALT-IN4
	P5_2	ALT-IN2
TSG20PTSI0/ENCA0E0	P4_1	ALT-IN2
	P7_4	ALT-IN4
TSG20PTSI1/ENCA0E1	P4_2	ALT-IN2
	P7_5	ALT-IN4
TSG20PTSI2/ENCA0EC	P4_3	ALT-IN2
	P7_6	ALT-IN4
TSG21PTSI0/ENCA1E0	P3_5	ALT-IN3
	P7_4	ALT-IN3

Table 2-5 Pins on which a Single Control Mode and Multiple Alternate Functions are Multiplexed (2/2)

Alternative Function Pin	Port Pin	Control Mode
TSG21PTSI1/ENCA1E1	P3_6	ALT-IN3
	P7_5	ALT-IN3
TSG21PTSI2/ENCA1EC	P4_0	ALT-IN3
	P7_6	ALT-IN3

2.2.3 Pin Data Input/Output

The registers used for data input and output are described below.

The sources of the data for output and of data read via the PPRn register will differ with the pin mode.

Output data

In port mode (PMCn.PMCn_m = 0), the data of Pn.Pn_m is output to pin Pn_m.

With a direct I/O control alternative function (PIPCn.PIPCn_m = 1), output is controlled by the alternative function.

Input data

A read operation of the PPRn register returns either the value of the Pn_m pin, the associated bit of the port register Pn.Pn_m, or the data output by an alternative function. The source of the data read via PPRn depends on the pin mode and the setting of several control bits. The table below summarizes the different PPRn read modes.

Caution When the PCLK signal is at or above 64 MHz, secure at least 3 cycles of the PCLK before reading a PPR immediately after register settings are made and the input buffers for registers connected to the ports are enabled. Also, read the registers twice and discard the value read the first time.

Table 2-6 PPRn_m Read Values

PMCn_m	PMn_m	PIBCn_m	PIPCn_m	Mode	PPRn_m Read Value
0	1	0	×	Port input, input buffer disabled	Pn.Pn_m register
		1		Port input, input buffer enabled	Pn_m pin
	0	×		Port output	Pn.Pn_m register*1
1	1	×	0	Input for alternative function under software I/O control	Pn_m pin
	0			Output for alternative function under software I/O control	Internal signal output for alternative function*2
	×		1	Input and output for alternative function under direct I/O control	If alternative functions sets port in • input: PPRn_m = Pn_m pin • output: PPRn_m = alternative function output*1, *2

Note 1. If PBDCn_m = 1, the Pn_m pin level is read via PPRn_m.
Note 2. When the PPRn_m bit is read while PMCn_m = 1 for the internal signal output of an alternative function, the level read is not that on the Pn_m pin but the level of the internal signal for the alternative function.

The control registers in the table above have following effects:

- PBDCn.PBDCn_m
  This bit forces to read the Pn_m pin level via PPRn_m, thus enabling a bi-directional mode, where the level of pin Pn_m can also be read back if the port is operated in output mode.

- PMCn.PMCn_m
  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).

- PMn.PMn_m
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) in port mode (PMCn_m = 0) and software I/O control alternative function mode (PMCn_m = 1, PIPCn_m = 0).

- PIBCn.PIBCn_m
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). If the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit, otherwise the Pn_m pin level is returned.

- PIPCn.PIPCn_m
  This bit selects between the software I/O control alternative function mode and direct I/O control alternative function.

Pn register write

The data to be output via port Pn_m in port mode (PMCn.PMCn_m = 0) is held in the port register Pn. The Pn data can be manipulated in two different ways:

- Direct Pn write
  New data can be written directly to the Pn register.
- Indirect Pn bit set/reset/not
  An indirect way to set (Pn_m = 1), reset (Pn_m = 0), or invert ($\overline{Pn_m} \rightarrow$ Pn_m) a Pn bit is possible using the following two registers:
  - Port set reset register PSRn
    If the bit PSRn.PSRn_(m+16) = 1, the value of bit PSRn.PSRn_m determines the value of Pn.Pn_m. Thus Pn_m can be set/reset without a direct write to Pn.
  - Port NOT register PNOTn
    Setting PNOTn.PNOTn_m = 1 inverts the bit Pn.Pn_m without a direct write to Pn_m.

Access to single bits of the Pn register is possible so that operations to manipulate (set, reset, not) bits for the Pn register do not indirectly affect any of the other Pn_m bits.

Take care because the simultaneous use of both methods to manipulate Pn_m bits is possible.

Caution When the following conditions are satisfied, the output on the Pn_m pin (e.g. ALT_OUTx) reflects the internal level of a multiplexed input function (e.g. ALT_INx).

- An output function (e.g. ALT_OUTx) is multiplexed with an input function (e.g. ALT_INx).
- The pin is in use for the multiplexed output function (PMCn.PMCn_m = 1 and PMn.PMn_m = 0).
- Enabling of the bi-directional mode (PBDCn.PBDCn_m = 1) leads to the level on Pn_m being readable from PPRn.PPRn_m.

2.2.4 Logic Diagram of Port Control

The figure below is a logic diagram of the port control function.

Caution This figure only shows the logic for reference and does not show the actual circuits.

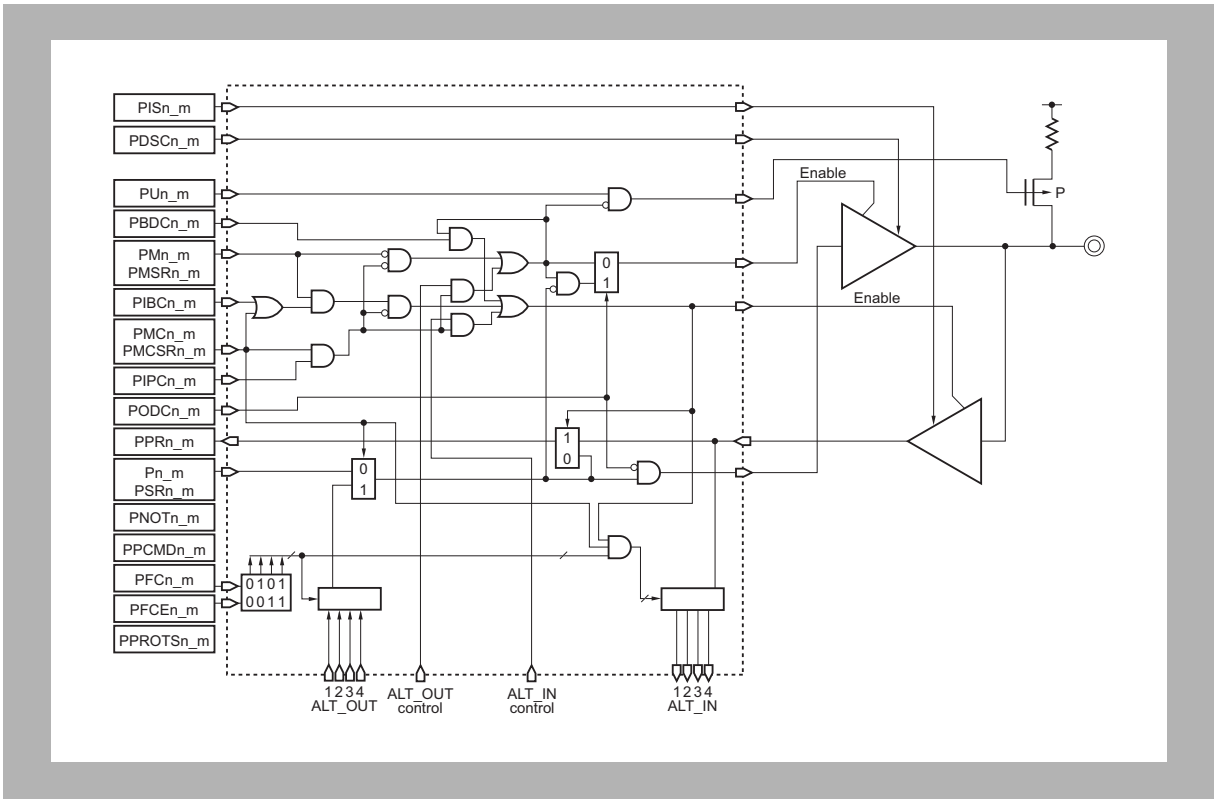


Figure 2-1 Logic Diagram of Port Control

2.3 Port Group Configuration Registers

This section starts with an overview of all configuration registers, and then presents all registers in detail. The configuration registers are grouped as follows:

- 2.3.2 Pin Function Configuration Registers
- 2.3.3 Pin Data Input/Output Registers
- 2.3.4 Configuration of Electrical Characteristics Registers

2.3.1 Overview

The following registers are used for the configuration of the individual pins of the port groups.

Table 2-7 Port Group Configuration Registers

Register Name	Symbol	Address
Port input buffer control register	PIBCn	<PORTn_base0> + 4000_H + n × 4
Port bi-direction control register	PBDCn	<PORTn_base0> + 4100_H + n × 4
Port IP control register	PIPCn	<PORTn_base0> + 4200_H + n × 4
Pull-up option register	PUn	<PORTn_base0> + 4300_H + n × 4
Port drive strength control register	PDSCn	<PORTn_base0> + 4600_H + n × 4
Port open drain control register	PODCn	<PORTn_base0> + 4500_H + n × 4
Port input buffer selection register	PISn	<PORTn_base0> + 4700_H + n × 4
Port register protection command register	PPCMDn	<PORTn_base0> + $4C00_H$ + n × 4
Port protection status register	PPROTSn	<PORTn_base0> + $4B00_H$ + n × 4
Port register	Pn	<PORTn_base1> + n × 4
Port set reset register	PSRn	<PORTn_base1> + 100_H + n × 4
Port pin read register	PPRn	<PORTn_base1> + 200_H + n × 4
Port mode register	PMn	<PORTn_base1> + 300_H + n × 4
Port mode control register	PMCn	<PORTn_base1> + 400_H + n × 4
Port function control register	PFCn	<PORTn_base1> + 500_H + n × 4
Port function control expansion register	PFCEn	<PORTn_base1> + 600_H + n × 4
Port NOT register	PNOTn	<PORTn_base1> + 700_H + n × 4
Port mode set reset register	PMSRn	<PORTn_base1> + 800_H + n × 4
Port mode control set reset register	PMCSRn	<PORTn_base1> + 900_H + n × 4

<PORTn_base> The base address <PORTn_base> of the port control registers is defined in table 2-2, Port n Register Addresses <PORTn_base0> and <PORTn_base1>.

Table 2-8 JTAG Port Group Configuration Registers

Register Name	Symbol	Address
JTAG port register	JP0	FF44 0000$_H$
JTAG port set / reset register	JPSR0	FF44 0010$_H$
JTAG port pin read register	JPPR0	FF44 0020$_H$
JTAG port mode register	JPM0	FF44 0030$_H$
JTAG port NOT register	JPNOT0	FF44 0070$_H$
JTAG port mode set register	JPMSR0	FF44 0080$_H$
JTAG port input buffer control register	JPIBC0	FF44 0400$_H$
JTAG pull-up option register	JPU0	FF44 0430$_H$

Note For details on JTAG port group configuration registers, see section 2.3.2, Pin Function Configuration Registers, section 2.3.3, Pin Data Input/Output Registers, and section 2.3.4, Configuration of Electrical Characteristics Registers. For the valid bits and access widths of JTAG port group configuration registers, see table 2-53, List of JTAG Port 0 Control Registers.

Caution Many registers include bits to which no function has been allocated. Unless there is a specific indication to the contrary, do not write values other than the initial values to such bits. Operation is not guaranteed if other values are set in these bits.

2.3.2 Pin Function Configuration Registers

(1) PMCn – Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode (n = 0 to 7).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCn_15	PMCn_14	PMCn_13	PMCn_12	PMCn_11	PMCn_10	PMCn_9	PMCn_8	PMCn_7	PMCn_6	PMCn_5	PMCn_4	PMCn_3	PMCn_2	PMCn_1	PMCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-9 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

(2) PMCSRn – Port Mode Control Set Reset Register n

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits specify whether the value PMCn.PMCn_m is set by a write operation to PMCn.PMCn_m or is defined by the lower 16 bits of PMCSRn (n = 0 to 7).

Access Readable and writable in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of register PMCn.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00000000_H

A reset from any source will initialize the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMCSRn_31	PMCSRn_30	PMCSRn_29	PMCSRn_28	PMCSRn_27	PMCSRn_26	PMCSRn_25	PMCSRn_24	PMCSRn_23	PMCSRn_22	PMCSRn_21	PMCSRn_20	PMCSRn_19	PMCSRn_18	PMCSRn_17	PMCSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCSRn_15	PMCSRn_14	PMCSRn_13	PMCSRn_12	PMCSRn_11	PMCSRn_10	PMCSRn_9	PMCSRn_8	PMCSRn_7	PMCSRn_6	PMCSRn_5	PMCSRn_4	PMCSRn_3	PMCSRn_2	PMCSRn_1	PMCSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-10 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_[31:16]	PMCSRn_m specifies whether the value of the corresponding lower bit PMCSRn_m value is written to PMCn_m. 0: PMCn_m is independent of PMCSRn_m 1: PMCn_m is PMCSRn_m Example: If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_[15:0]	Specifies the PMCn_m value if the corresponding upper bit PMCSRn_(m + 16) is 1. 0: PMCn_m = 0 1: PMCn_m = 1

(3) PIPCn – Port IP Control Register

This register specifies whether the I/O direction of pin Pn_m is controlled by the port mode register PMn.PMn_m or by an alternative function.

If pin Pn_m is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires to directly control the I/O direction of Pn_m, PIPCn.PIPCn_m must be set to 1 as well. This hands over I/O control to the alternative function and overrules the PMn.PMn_m setting (n = 0 to 7).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-11 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O control mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

Caution 1. The following pins control buffer I/O from peripheral I/O. Set PIPCn to 1 if any of these pins is to be used.

- CSIGnSC, CSIHnSC, URTHnSC
  (serial clock input/output pins)

  Corresponding bits of the PIPCn register can be set to 0 as long as the I/O direction corresponding to the master or slave mode of the serial interface is set in the PMn register (i.e. PMn_m = 0 for master mode or PMn_m = 1 for slave mode).
- TAUAnO10 to TAUAnO15, TSG2nO1 to TSG2nO6
  (target pins for Hi-Z control)

  When a pin is to be used as a timer output pin for the timer option module (TAPA) and Hi-Z control is not to be applied, the corresponding bit in the PIPCn register can be set to 0 as long as the pin is set as an output pin by the corresponding bit in the PMn register (PMn_m = 0).

Caution 2. To use the data consistency checking function of the CSIG or CSIH module, set the bit in the PIPCn register that is allocated to the CSIGnSO pin or CSIHnSO pin to 0 (operation of data consistency checking is not guaranteed if the bit is set to 1).

(4) PMn – Port Mode Register

The PMn register specifies whether the individual pins of the port group n are in input mode or in output mode (n = 0 to 8).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value $FFFF_H$

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-12 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

Note 1. To use a port in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).

Note 2. By default, PMn_m, it specifies the I/O direction in port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m=1), since PIPCn.PIPCn_m = 0 after reset.

(5) PMSRn – Port Mode Set Reset Register n

This register provides an alternative method to write data to the PMn register.

The upper 16 bits specify whether the value PMn.PMn_m is set by a write operation to PMn.PMn_m or is defined by the lower 16 bits of PMSRn (n = 0 to 8).

Access Readable and writable in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of register PMn.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value $0000FFFF_H$

A reset from any source will initialize the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-13 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	PMSRn_m specifies whether the value of the corresponding lower bit PMSRn_m value is written to PMn_m. 0: PMn_m is independent of PMSRn_m 1: PMn_m is PMSRn_m Example: If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Specifies the PMn_m value if the corresponding upper bit PMSRn_(m + 16) is 1. 0: PMn_m = 0 1: PMn_m = 1

(6) PIBCn – Port Input Buffer Control Register

In input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables/disables the port pin's input buffer (n = 0 to 8).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIBCn_15	PIBCn_14	PIBCn_13	PIBCn_12	PIBCn_11	PIBCn_10	PIBCn_9	PIBCn_8	PIBCn_7	PIBCn_6	PIBCn_5	PIBCn_4	PIBCn_3	PIBCn_2	PIBCn_1	PIBCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-14 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled

Caution Settings in this register are overruled in bi-directional mode (PBDCn.PBDCn_m = 1).

(7) PFCn – Port Function Control Register

This register, together with register PFCEn, specifies an alternative function of the pins.

Some alternative functions require direct I/O control of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m (n = 0 to 7).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFCn_15	PFCn_14	PFCn_13	PFCn_12	PFCn_11	PFCn_10	PFCn_9	PFCn_8	PFCn_7	PFCn_6	PFCn_5	PFCn_4	PFCn_3	PFCn_2	PFCn_1	PFCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-15 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of a pin. See table 2-4, Alternative Mode Selection Overview (PMCn.PMCn_m = 1).

(8) PFCEn – Port Function Control Expansion Register

This register, together with register PFCn, specifies an alternative function of the pins.

Some alternative functions require direct I/O control of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m (n = 0 to 7).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn_11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-16 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of a pin. See table table 2-4, Alternative Mode Selection Overview (PMCn.PMCn_m = 1).

2.3.3 Pin Data Input/Output Registers

(1) PBDCn – Port Bi-Direction Control Register

This register enables the input buffer, thus the Pn_m pin level is always read via PPRn.PPRn_m (n = 0, 1, and 3 to 6).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-17 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn_[15:0]	Enables/disables bi-directional mode of the corresponding pin. 0: Bi-directional mode disabled 1: Bi-directional mode enabled

Caution To use the data-consistency checking function of the CSIG or CSIH module, set the bit in the PBDCn register that is allocated to the CSIGnSO or CSIHnSO pin to 1.

(2) PPRn – Port Pin Read Register

This register reflects the actual level of pin Pn_m, the value of the Pn.Pn_m bit or the level of an alternative output function. The value which is read depends on various control settings as described in table 2-6, PPRn_m Read Values (n = 0 to 8).

Access This register is read-only and only readable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2-18 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	Pin Pn_m, Pn.Pn_m value or alternative function output.

(3) Pn – Port Register

This register holds the data Pn.Pn_m to be output via the related port Pn_m in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0) (n = 0 to 8).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-19 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of pin m (m = 0 to 15). 0: Outputs low level 1: Outputs high level

Note The bits of this register can be manipulated by different means; refer to section 2.2.3, Pin Data Input/Output, under the keyword “Pn register write”.

(4) PNOTn – Port NOT Register

This register allows to invert a bit Pn_m of the port register Pn without directly writing to Pn (n = 0 to 8).

Access Writing in 16-bit units is the only form of access. The register is always read as 0000_H.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PNOTn_15	PNOTn_14	PNOTn_13	PNOTn_12	PNOTn_11	PNOTn_10	PNOTn_9	PNOTn_8	PNOTn_7	PNOTn_6	PNOTn_5	PNOTn_4	PNOTn_3	PNOTn_2	PNOTn_1	PNOTn_0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 2-20 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted: 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted (Pn_m → Pn_m)

(5) PSRn – Port Set Reset Register

This register provides an alternative method to write data to the Pn register.

The upper 16 bits specify whether the value Pn.Pn_m is set by a write operation to Pn.Pn_m or is defined by the lower 16 bits of PSRn (n = 0 to 8).

Access Readable and writable in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of register Pn.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00000000_H

A reset from any source will initialize the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-21 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	PSRn_m specifies whether the value of the corresponding lower bit PSRn_m value is written to Pn_m. 0: Pn_m is independent of PSRn_m 1: Pn_m is PSRn_m Example: If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit PSRn_(m+16) is 1. 0: Pn_m = 0 1: Pn_m = 1

2.3.4 Configuration of Electrical Characteristics Registers

(1) PUn – Pull-Up Option Register

This register specifies whether a pull-up resistor is connected to an input pin (n = 0 to 8).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-22 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether a pull-up resistor is connected to the corresponding pin. 0: No pull-up resistor connected 1: Pull-up resistor connected

Caution The pull-up resistor has no effect when the pin is operated in output mode.

(2) PDSCn – Port Drive Strength Control Register

This register specifies port pin output drive strength (n = 6).

Access Readable and writable in 32-bit units.

Writing to this register is protected by a special sequence of instructions. Refer to section 2.3.5, Port Register Protection.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00000000_H

A reset from any source will initialize the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDSCn_31	PDSCn_30	PDSCn_29	PDSCn_28	PDSCn_27	PDSCn_26	PDSCn_25	PDSCn_24	PDSCn_23	PDSCn_22	PDSCn_21	PDSCn_20	PDSCn_19	PDSCn_18	PDSCn_17	PDSCn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDSCn_15	PDSCn_14	PDSCn_13	PDSCn_12	PDSCn_11	PDSCn_10	PDSCn_9	PDSCn_8	PDSCn_7	PDSCn_6	PDSCn_5	PDSCn_4	PDSCn_3	PDSCn_2	PDSCn_1	PDSCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-23 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 0	PDSCn_[31:0]	Specifies the drive strength of the port-pin output buffer. 0: Low drive strength (25 MHz) 1: High drive strength (40 MHz)

Caution Set this bit to 1 when using a FlexRay output pin (FLX0TXENA, FLX0TXDA, FLX0TXENB, or FLX0TXDB) for an alternative function.

(3) PODCn – Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function (n = 2 and 3).

Access Readable and writable in 32-bit units.

Writing to this register is protected by a special sequence of instructions. Refer to section 2.3.5, Port Register Protection.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00000000_H

A reset from any source will initialize the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PODCn_31	PODCn_30	PODCn_29	PODCn_28	PODCn_27	PODCn_26	PODCn_25	PODCn_24	PODCn_23	PODCn_22	PODCn_21	PODCn_20	PODCn_19	PODCn_18	PODCn_17	PODCn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-24 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 0	PODCn_[31:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

(4) PISn – Port Input Buffer Selection Register

This register specifies the input buffer characteristics (n = 6).

Access Readable and writable in 16-bit units.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-25 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer characteristics. 0: Schmitt characteristics 1: CMOS characteristics

Caution Set this bit to 1 when using a FlexRay input pin (FLX0RXDA or FLX0RXDB) for an alternative function.

Note The difference between the input buffer characteristics is whether one has hysteresis width or not (only the Schmitt characteristics has hysteresis width). For details on the values, see section 28, Electrical Characteristics.

2.3.5 Port Register Protection

(1) PPCMDn – Port Register Protection Command Register

This register is a command register for protected port register (n = 2, 3, and 6).

Access Writable in 8-bit units. When read, it is always read as 0.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W

Table 2-26 PPCMDm Register Contents

Bit Position	Bit Name	Function
7 to 0	-	Command capable of writing to protected port registers

(2) PPROTSn – Port Protection Status Registers

This register shows the state of the write sequence for a protected port register (n = 2, 3, and 6).

Access Readable in 8-bit units. Values written are ignored.

Address Refer to table 2-7, Port Group Configuration Registers.

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PPROTSn_0
R	R	R	R	R	R	R	R

Table 2-27 PPROTSn Register Contents

Bit Position	Bit Name	Function
0	PPROTSn_0	Confirms errors in the write sequence for a protected port register. 0: No protection error occurred 1: Protection error occurred

(3) Protected Port Registers

PDSCn - Port Driving Ability Control Register

PODCn - Port Open Drain Control Register

(4) Sequence for Writing to a Protected Port Register

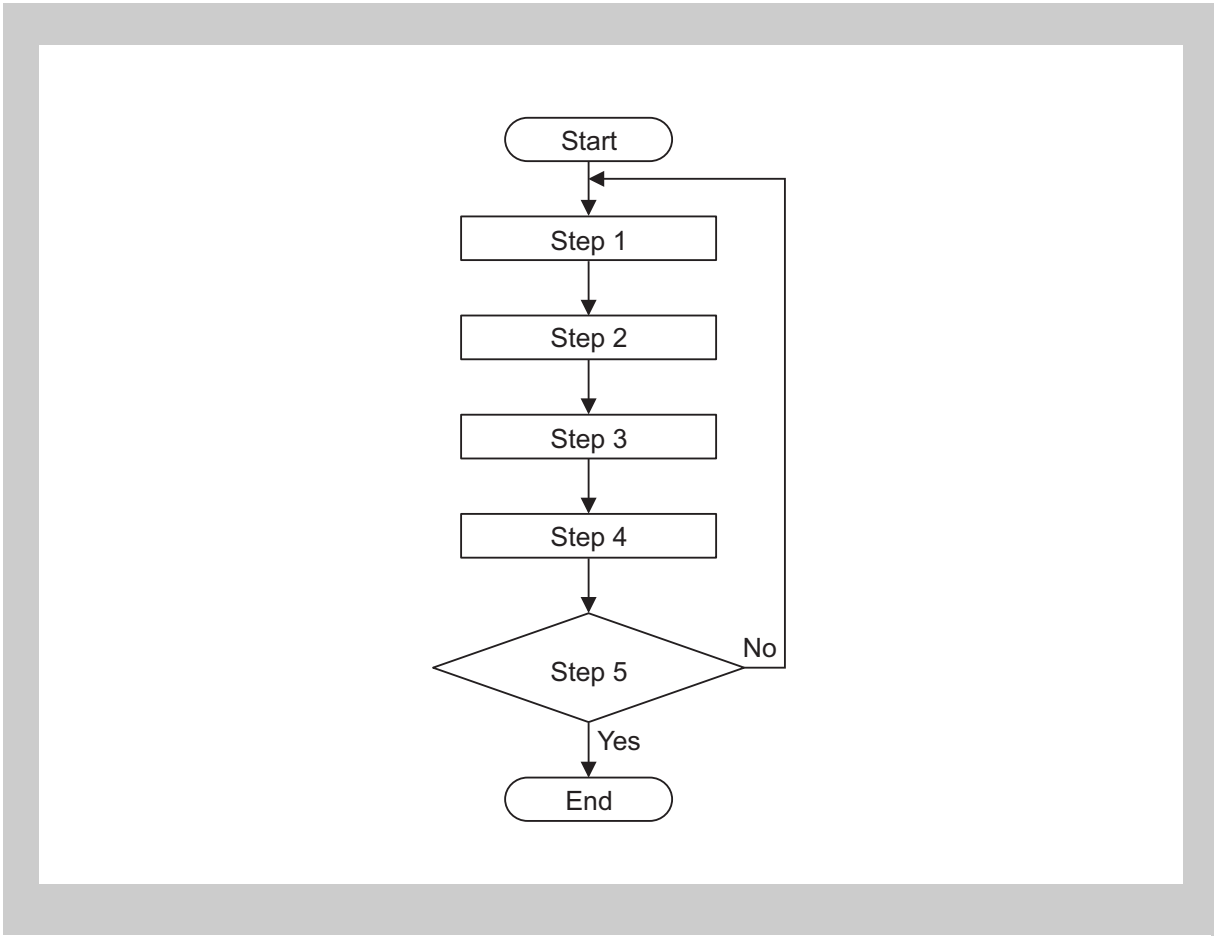


Figure 2-2 Sequence for Writing to a Protected Port Register

Step 1 Initialize the sequence of writing by writing $A5_H$ to register PPCMD.

Step 2 Write the set value in a 32-bit unit to a protected register (this will not update the register).

Step 3 Write the inverse of the set value in a 32-bit unit to the same protected register (this will not update the register).

Step 4 Again write the original set value in a 32-bit unit to the same protected register (this will update the register).

Step 5 Check if the value in the PPROTSn.PPROTSn_0 bit is 0; this will confirm that the set value has actually been written to the protected port register.

2.4 Port Group Configuration

This section shows an overview of the port groups. See table 2-28, List of Port Groups.

Table 2-32, List of States of Port Pins, shows the changes in pin function when this microcontroller is reset, or is at each stand-by mode.

In each sub-section, the configuration register settings of each port group are listed (see section 2.6, Port 0, and subsequent passages).

2.4.1 List of Ports and Pins

(1) List of Port Groups

Table 2-28 List of Port Groups (1/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Characteristics
P0	P0_0	ADCA0 TRG2/ INTP5	-	CSIG0SI	URTH0 RXD/ INTP0/ FCN1TX	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P0_1	ADCA0 TRG1/ INTP4	-	CSIG0SO	FCN1RX/ URTH0 TXD	
	P0_2	ADCA0 TRG0/ INTP3	TAUA1I2/ TAUA1O2	CSIG0SC	URTH0SC	
	P0_3	-	TAUA1I4/ TAUA1O4	FCN0TX	URTH0 CTS	
	P0_4	-	TAUA1I6/ TAUA1O6	FCN0RX	URTH0 RTS	
	P0_5	TAUJ0I2/ TAUJ0O2	URTH2 RXD/ INTP2	CSIG0SI	TSG21O1	
	P0_6	TAUJ0I1/ TAUJ0O1	URTH2 TXD	CSIG0SO	TSG21O2	
	P0_7	TAUJ0I0/ TAUJ0O0	-	CSIG0SC	TSG21O3	
	P0_8	FCN1TX	TSG20CLKI	-	TSG21O4	
	P0_9	FCN1RX	TSG21CLKI	-	TSG21O5	

Table 2-28 List of Port Groups (2/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Characteristics
P1	P1_0	NMI/ CLKOUT	OSTM1O	-	TSG21O6	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P1_1	-	CSIG2SI	-	TSG20O1	
	P1_2	-	CSIG2SO	-	TSG20O2	
	P1_3	-	CSIG2SC	-	TSG20O3	
	P1_4	-	CSIH0SI	TAUJ0I0/ TAUJ0O0	TSG20O4	
	P1_5	-	CSIH0SO	TAUJ0I1/ TAUJ0O1	TSG20O5	
	P1_6	-	CSIH0SC	TAUJ0I2/ TAUJ0O2	TSG20O6	
	P1_7	TAUA1I0/ CSIG2RYO	CSIG0SI	TAUJ0I3/ TAUJ0O3	URTH0 RXD/ TSG20O7/ INTP0	
	P1_8	TAUA1I1/ TPB0O	CSIG0SO	-	URTH0 TXD	
	P1_9	TAUA1I2/ TPB1O	CSIG0SC	-	URTH0 SC	
P2	P2_0	TAUA0I1/ TAUA0O1	TAUA0I0/ TAUA0O0	URTH1SC	CSIG0RYI/ TSG21O0	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P2_1	TAUA0I3/ TAUA0O3	TAUA0I2/ TAUA0O2	TAUA1O14	CSIG1RYI/ TSG21O7	
	P2_2	TAUA0I5/ TAUA0O5	TAUA0I4/ TAUA0O4	TAUA0O10	TSG21O1	
	P2_3	TAUA0I7/ TAUA0O7	TAUA0I6/ TAUA0O6	TAUA0O12	TSG21O2	
	P2_4	TAUA0I9/ TAUA0O9	TAUA0I8/ TAUA0O8	TAUA0O14	TSG21O3	
	P2_5	TAUA0I11/ TAUA0O11	TAUA0I10/ TAUA0O10	CSIH0CS0	TSG21O4	
	P2_6	TAUA0I13/ TAUA0O13	TAUA0I12/ TAUA0O12	CSIH0CS1	TSG21O5	
	P2_7	TAUA0I15/ TAUA0O15	TAUA0I14/ TAUA0O14	CSIH0CS2	TSG21O6	

Table 2-28 List of Port Groups (3/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Characteristics
P3	P3_0	TAUA1I1/ TAUA1O1	TAUA1I0/ TAUA1O0	URTH1 RXD/ INTP1/ CSIH0CS3	TAUA1O10	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P3_1	TAUA1I3/ TAUA1O3	TAUA1I2/ TAUA1O2	URTH1 TXD	TAUA1O12	
	P3_2	TAUA1I5/ TAUA1O5	TAUA1I4/ TAUA1O4	CSIG1SI	-	
	P3_3	TAUA1I7/ TAUA1O7	TAUA1I6/ TAUA1O6	CSIG1SO	TPB0O	
	P3_4	TAUA1I9/ TAUA1O9	TAUA1I8/ TAUA1O8	CSIG1SC	TPB1O	
	P3_5	TAUA1I11/ TAUA1O11	TAUA1I10/ TAUA1O10	TSG21 PTSI0/ ENCA1E0	URTH2 RXD/ INTP2/ FCN1TX	
	P3_6	TAUA1I13/ TAUA1O13	TAUA1I12/ TAUA1O12	TSG21 PTSI1/ ENCA1E1	FCN1RX/ URTH2 TXD	
	P3_7	TAUA1I15/ TAUA1O15	TAUA1I14/ TAUA1O14	-	CSIG2RYI	
P4	P4_0	URTH1 RXD/ INTP1/ TAUA1O15	-	TSG21 PTSI2/ ENCA1EC/ FCN0TX	URTH2SC	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P4_1	URTH2 CTS/ URTH1 TXD	TSG20 PTSI0/ ENCA0E0	FCN0RX	CSIG2SI	
	P4_2	CSIG1SI/ URTH2 RTS	TSG20 PTSI1/ ENCA0E1/ OSTM1O	-	CSIG2SO	
	P4_3	CSIG1SO	TSG20 PTSI2/ ENCA0EC/ OSTM0O	URTH1 CTS	CSIG2SC	
	P4_4	CSIG1SC	TAUJ1I0/ TAUJ1O0	TAUJ0I0/ URTH1 RTS	ESO0/ INTP6/ CSIG0RYO	
	P4_5	-	TAUJ1I1/ TAUJ1O1	TAUJ0I1/ ADCA0 CNV2	ESO1/ INTP7/ CSIG1RYO	
	P4_6	TAUA1I8	TAUJ1I2/ TAUJ1O2	TAUJ0I2/ ADCA0 CNV1	ESO2/ INTP8	
	P4_7	NMI	TAUJ1I3/ TAUJ1O3	TAUJ0I3/ ADCA0 CNV0	ESO3/ INTP9	

Table 2-28 List of Port Groups (4/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Characteristics
P5	P5_0	-	-	URTH0RXD /INTP0	-	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P5_1	CSIH1CS0	URTH1 RXD/ INTP1/ CSIH0CS0	CSIH0RYI/ URTH0 TXD	ADCA0 TRG0/ INTP3	
	P5_2	CSIH1CS1	URTH2 RXD/ INTP2/ CSIH0CS1	URTH0SC	ADCA0 TRG1/ INTP4/ CSIH0RYO	
	P5_3	CSIH1CS2	ADCA0 TRG0/ INTP3/ CSIH0CS2	URTH0 CTS	ADCA0 TRG2/ INTP5	
	P5_4	CSIH1CS3	CSIH0CS3	CSIH1RYI/ URTH0 RTS	-	
	P5_5	TAUJ0I3/ TAUJ0O3	-	CSIH1RYO	-	
	P5_6	CSIH0SI	URTH1RXD /INTP1	-	-	
	P5_7	CSIH0SO	URTH1 TXD	-	-	
	P5_8	CSIH0SC	-	-	-	
	P5_9	CSIH0RYI	CSIH1SI	-	ADCA0 TRG1/ INTP4	
	P5_10	CSIH0RYO	CSIH1SO	-	ADCA0 TRG2/ INTP5	
	P5_11	-	CSIH1SC	-	-	
P6	P6_0	CSIG2SI/ TSG21O6	-	TAUA0O12	TSG20O0	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P6_1	CSIG2SO	URTH1CTS	TAUA0O10	-	
	P6_2	CSIG2SC	URTH1RTS	TAUA0O11	-	
	P6_3	FLX0RXDA	URTH1RXD /INTP1	TAUA1O10	-	
	P6_4	FLX0 TXENA	URTH1SC	TAUA1O11	-	
	P6_5	FLX0TXDA	URTH1TXD	TAUA1O12	-	
	P6_6	FLX0RXDB	URTH0RXD /INTP0	TAUA1O13	-	
	P6_7	FLX0 TXENB	URTH0SC	TAUA1O14	-	
	P6_8	FLX0TXDB	URTH0TXD	TAUA1O15	-	

Table 2-28 List of Port Groups (5/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Characteristics
P7	P7_0	TAUA0I0/ TAUA0O0	ADCA0 TRG0/ INTP3/ TSG21O0	TAUJ1I0	ENCA0I0/ CSIH0CS0	• Input/output port • Inputs and outputs are specifiable in 1-bit units.
	P7_1	TAUA0I2/ TAUA0O2	INTP10/ TSG21O1	TAUJ1I1	ENCA0I1/ CSIH0CS1	
	P7_2	TAUA0I4/ TAUA0O4	ESO1/ INTP7/ TSG21O2	TAUJ1I2	ENCA1I0/ CSIH0CS2	
	P7_3	TAUA0I6/ TAUA0O6	ESO0/ INTP6/ TSG21O3	TAUJ1I3	ENCA1I1/ CSIH0CS3	
	P7_4	TAUA0I8/ TAUA0O8	ADCA0 TRG0/ INTP3/ TSG21O4	TSG21 PTSI0/ ENCA1E0/ CSIH0CS4	TSG20 PTSI0/ ENCA0E0/ CSIH1CS0	
	P7_5	TAUA0I10/ TAUA0O10	ADCA0 TRG1/ INTP4/ TSG21O5	TSG21 PTSI1/ ENCA1E1/ CSIH0CS5	TSG20 PTSI1/ ENCA0E1/ CSIH1CS1	
	P7_6	TAUA0I12/ TAUA0O12	ADCA0 TRG2/ INTP5/ TSG21O6	TSG21 PTSI2/ ENCA1EC/ CSIH0CS6	TSG20 PTSI2/ ENCA0EC/ CSIH1CS2	
	P7_7	TAUA0I14/ TAUA0O14	TSG21O7	NMI/ CSIH0CS7	CSIH1CS3	
P8	P8_0	-	-	-	-	• Input/output port • Inputs and outputs are specifiable in 1-bit units. • Multiplexed with MODE0
JP0	JP0_0	-	-	-	-	• JTAG port • Used for the JTAG interface
	JP0_1	-	-	-	-	
	JP0_2	-	-	-	-	
	JP0_3	-	-	-	-	
	JP0_5	-	-	-	-	

(2) List of Pins Other than Port Pins

Table 2-29 List of Pins Other than Port Pins (1/9)

	Pin Name	Function
INTC	NMI	Input for non-maskable external interrupt requests
	INTP0	Input for maskable external interrupt requests
	INTP1	
	INTP2	
	INTP3	
	INTP4	
	INTP5	
	INTP6	
	INTP7	
	INTP8	
	INTP9	
	INTP10	

Table 2-29 List of Pins Other than Port Pins (2/9)

	Pin Name	Function
TAUA0	TAUA0I0	Input for the TAUA0 channel
	TAUA0I1	
	TAUA0I2	
	TAUA0I3	
	TAUA0I4	
	TAUA0I5	
	TAUA0I6	
	TAUA0I7	
	TAUA0I8	
	TAUA0I9	
	TAUA0I10	
	TAUA0I11	
	TAUA0I12	
	TAUA0I13	
	TAUA0I14	
	TAUA0I15	
	TAUA0O0	Output for the TAUA0 channel
	TAUA0O1	
	TAUA0O2	
	TAUA0O3	
	TAUA0O4	
	TAUA0O5	
	TAUA0O6	
	TAUA0O7	
	TAUA0O8	
	TAUA0O9	
	TAUA0O10	
	TAUA0O11	
	TAUA0O12	
	TAUA0O13	
	TAUA0O14	
	TAUA0O15	

Table 2-29 List of Pins Other than Port Pins (3/9)

	Pin Name	Function
TAUA1	TAUA1I0	Input for the TAUA1 channel
	TAUA1I1	
	TAUA1I2	
	TAUA1I3	
	TAUA1I4	
	TAUA1I5	
	TAUA1I6	
	TAUA1I7	
	TAUA1I8	
	TAUA1I9	
	TAUA1I10	
	TAUA1I11	
	TAUA1I12	
	TAUA1I13	
	TAUA1I14	
	TAUA1I15	
	TAUA1O0	Output for the TAUA1 channel
	TAUA1O1	
	TAUA1O2	
	TAUA1O3	
	TAUA1O4	
	TAUA1O5	
	TAUA1O6	
	TAUA1O7	
	TAUA1O8	
	TAUA1O9	
	TAUA1O10	
	TAUA1O11	
	TAUA1O12	
	TAUA1O13	
	TAUA1O14	
	TAUA1O15	

Table 2-29 List of Pins Other than Port Pins (4/9)

	Pin Name	Function
TAUJ0	TAUJ0I0	Input for the TAUJ0 channel
	TAUJ0I1	
	TAUJ0I2	
	TAUJ0I3	
	TAUJ0O0	Output for the TAUJ0 channel
	TAUJ0O1	
	TAUJ0O2	
	TAUJ0O3	
TAUJ1	TAUJ1I0	Input for the TAUJ1 channel
	TAUJ1I1	
	TAUJ1I2	
	TAUJ1I3	
	TAUJ1O0	Output for the TAUJ1 channel
	TAUJ1O1	
	TAUJ1O2	
	TAUJ1O3	
TSG20	TSG20O0	Pulse output for the TSG20
	TSG20O1	
	TSG20O2	
	TSG20O3	
	TSG20O4	
	TSG20O5	
	TSG20O6	
	TSG20O7	
	TSG20PTSI0	Pattern input for the TSG20
	TSG20PTSI1	
	TSG20PTSI2	
	TSG20CLKI	Input for the TSG20 count enable signal

Table 2-29 List of Pins Other than Port Pins (5/9)

	Pin Name	Function
TSG21	TSG21O0	Pulse output for the TSG21
	TSG21O1	
	TSG21O2	
	TSG21O3	
	TSG21O4	
	TSG21O5	
	TSG21O6	
	TSG21O7	
	TSG21PTSI0	Pattern input for the TSG21
	TSG21PTSI1	
	TSG21PTSI2	
	TSG21CLKI	Input for the TSG21 count enable signal
ENC0	ENCA0I0	Input for the capture signal of the ENC0 encoder counter
	ENCA0I1	
	ENCA0E0	Encoder input for the ENC0 encoder counter
	ENCA0E1	
	ENCA0EC	Clearing input for the ENC0 encoder counter
ENC1	ENCA1I0	Input for the capture signal of the ENC1 encoder counter
	ENCA1I1	
	ENCA1E0	Encoder input for the ENC1 encoder counter
	ENCA1E1	
	ENCA1EC	Clearing input for the ENC1 encoder counter
TPB0	TPB0O	Pulse output from the TPB0 timer pattern buffer
TPB1	TPB1O	Pulse output from the TPB1 timer pattern buffer
OST0	OSTM0O	Output for OS timer 0
OST1	OSTM1O	Output for OS timer 1
TAPA	ESO0	Input for forcibly stopping timer output
	ESO1	
	ESO2	
	ESO3	
BRG	CLKOUT	Clock output
CAN0	FCN0TX	Output for data transmission from CAN0
	FCN0RX	Input for data reception by CAN0
CAN1	FCN1TX	Output for data transmission from CAN1
	FCN1RX	Input for data reception by CAN1

Table 2-29 List of Pins Other than Port Pins (6/9)

	Pin Name	Function
FlexRay	FLX0TXDA	Output for data transmission from the FlexRay module
	FLX0TXDB	
	FLX0RXDA	Input for data reception by the FlexRay module
	FLX0RXDB	
	FLX0TXENA	Output for the communications enable signal from the FlexRay module
	FLX0TXENB	
UARTH0	URTH0TXD	Output for serial data transmission from UART0
	URTH0RXD	Input for serial data reception by UART0
	URTH0CTS	Input for hand-shake signal in UART0 transmission
	URTH0RTS	Output for hand-shake signal in UART0 reception
	URTH0SC	Input or output for the UART0 serial clock
UARTH1	URTH1TXD	Output for serial data transmission from UART1
	URTH1RXD	Input for serial data reception by UART1
	URTH1CTS	Input for hand-shake signal in UART1 transmission
	URTH1RTS	Output for hand-shake signal in UART1 reception
	URTH1SC	Input or output for the UART1 serial clock
UARTH2	URTH2TXD	Output for serial data transmission from UART2
	URTH2RXD	Input for serial data reception by UART2
	URTH2CTS	Input for hand-shake signal in UART2 transmission
	URTH2RTS	Output for hand-shake signal in UART2 reception
	URTH2SC	Input or output for the UART2 serial clock
CSIG0	CSIG0SC	Input or output for the CSIG0 serial clock
	CSIG0SI	Input for serial data reception by CSIG0
	CSIG0SO	Output for serial data transmission from CSIG0
	CSIG0RYI	Input for serial ready/busy for CSIG0
	CSIG0RYO	Output for serial ready/busy for CSIG0
CSIG1	CSIG1SC	Input or output for the CSIG1 serial clock
	CSIG1SI	Input for serial data reception by CSIG1
	CSIG1SO	Output for serial data transmission from CSIG1
	CSIG1RYI	Input for serial ready/busy for CSIG1
	CSIG1RYO	Output for serial ready/busy for CSIG1

Table 2-29 List of Pins Other than Port Pins (7/9)

	Pin Name	Function
CSIG2	CSIG2SC	Input or output for the CSIG2 serial clock
	CSIG2SI	Input for serial data reception by CSIG2
	CSIG2SO	Output for serial data transmission from CSIG2
	CSIG2RYI	Input for serial ready/busy for CSIG2
	CSIG2RYO	Output for serial ready/busy for CSIG2
CSIH0	CSIH0SC	Input or output for the CSIH0 serial clock
	CSIH0SI	Input for serial data reception by CSIH0
	CSIH0SO	Output for serial data transmission from CSIH0
	CSIH0RYI	Input for the serial ready/busy signal of CSIH0
	CSIH0RYO	Output for the serial ready/busy signal of CSIH0
	CSIH0CS0	Output for the serial chip selection signal of CSIH0
	CSIH0CS1	
	CSIH0CS2	
	CSIH0CS3	
	CSIH0CS4	
	CSIH0CS5	
	CSIH0CS6	
	CSIH0CS7	
CSIH1	CSIH1SC	Input or output for the CSIH1 serial clock
	CSIH1SI	Input for serial data reception by CSIH1
	CSIH1SO	Output for serial data transmission from CSIH1
	CSIH1RYI	Input for the serial ready/busy signal of CSIH1
	CSIH1RYO	Output for the serial ready/busy signal of CSIH1
	CSIH1CS0	Output for the serial chip selection signal of CSIH1
	CSIH1CS1	
	CSIH1CS2	
	CSIH1CS3	

Table 2-29 List of Pins Other than Port Pins (8/9)

	Pin Name	Function
ADCA0	ADCA0I1	Analog inputs for the A/D converter
	ADCA0I2	
	ADCA0I3	
	ADCA0I4	
	ADCA0I5	
	ADCA0I6	
	ADCA0I7	
	ADCA0I8	
	ADCA0I9	
	ADCA0I10	
	ADCA0I11	
	ADCA0I12	
	ADCA0I13	
	ADCA0I14	
	ADCA0I15	
	ADCA0I16	
	ADCA0I17	
	ADCA0I18	
	ADCA0I19	
	ADCA0I20	
	ADCA0I21	
	ADCA0I22	
	ADCA0TRG0	External trigger inputs for the A/D converter
	ADCA0TRG1	
	ADCA0TRG2	
	ADCA0CNV0	Status outputs for the A/D converter
	ADCA0CNV1	
	ADCA0CNV2	
	AVDD0	Source of positive power supply for the A/D converter
	AVSS0	Ground potential for the A/D converter
	AVREF0P	Source of reference power supply for the A/D converter
	AVREF0M	Reference ground potential for the A/D converter
SGA	ERROROUT	Output for error signal from the safety guardian

Table 2-29 List of Pins Other than Port Pins (9/9)

	Pin Name	Function
Nexus	DCUTCK	Input for the debugger clock
	DCUTDI	Input for debugger data
	DCUTDO	Output for debugger data
	DCUTMS	Input for debugger mode-selection signal
	$\overline{\text{DCUTRST}}$	Input for debugger reset
	$\overline{\text{DCUTRDY}}$	Output for debugger-ready signal
Other debugger	$\overline{\text{DCUEVTO}}$	Output for event-trigger signal
	RESETOUT	Output for debugger reset
	$\overline{\text{RESETOUT}}$	
RESET	$\overline{\text{RESET}}$	Input for system reset
CLOCK	X1	For connection of crystal for system clock oscillation
	X2	
MODE	FLMD0	Operating mode specifying pin/flash write control pin
	FLMD1	Operation mode specifying pin
	MODE0	Operation mode specifying pin
POWER	OSCVDD	Source of positive power supply (3.3 V/5.0 V) for the OSC
	OSCVSS	Ground potential for the OSC
	CPUPLLVDD	Source of positive power supply (1.2 V) for the PLL (for the CPU)
	CPUPLLVSS	Ground potential for the PLL (for the CPU)
	FRPLLVDD	Source of positive power supply (1.2 V) for the FlexRay pins
	FRPLLVSS	Ground potential for the FlexRay pins
	VDDC	Source of positive power supply (1.2 V) for an internal unit (the checker)
	VSSC	Ground potential for an internal unit (the checker)
	VDDM	Source of positive power supply (1.2 V) for an internal unit (the master)
	VSSM	Ground potential for an internal unit (the master)
	EVDD	Source of positive power supply (3.3 V/5.0 V) for the external pins
	EVSS	Ground potential for external pins
	FVDD	Source of positive power supply (3.3 V/5.0 V) for the flash memory
	FVSS	Ground potential for the flash memory

(3) Handling of Unused Port Pins

Table 2-30 Handling of Unused Port Pins (1/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	I/O circuit type	Handling of Unused Pins
P0	P0_0	ADCA0 TRG2/ INTP5	-	CSIG0SI	URTH0 RXD/ INTP0/ FCN1TX	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P0_1	ADCA0 TRG1/ INTP4	-	CSIG0SO	FCN1RX/ URTH0 TXD		
	P0_2	ADCA0 TRG0/ INTP3	TAUA1I2/ TAUA1O2	CSIG0SC	URTH0SC		
	P0_3	-	TAUA1I4/ TAUA1O4	FCN0TX	URTH0 CTS		
	P0_4	-	TAUA1I6/ TAUA1O6	FCN0RX	URTH0 RTS		
	P0_5	TAUJ0I2/ TAUJ0O2	URTH2 RXD/ INTP2	CSIG0SI	TSG21O1		
	P0_6	TAUJ0I1/ TAUJ0O1	URTH2 TXD	CSIG0SO	TSG21O2		
	P0_7	TAUJ0I0/ TAUJ0O0	-	CSIG0SC	TSG21O3		
	P0_8	FCN1TX	TSG20 CLKI	-	TSG21O4		
	P0_9	FCN1RX	TSG21 CLKI	-	TSG21O5		
P1	P1_0	NMI/ CLKOUT	OSTM1O	-	TSG21O6	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P1_1	-	CSIG2SI	-	TSG20O1		
	P1_2	-	CSIG2SO	-	TSG20O2		
	P1_3	-	CSIG2SC	-	TSG20O3		
	P1_4	-	CSIH0SI	TAUJ0I0/ TAUJ0O0	TSG20O4		
	P1_5	-	CSIH0SO	TAUJ0I1/ TAUJ0O1	TSG20O5		
	P1_6	-	CSIH0SC	TAUJ0I2/ TAUJ0O2	TSG20O6		
	P1_7	TAUA1I0/ CSIG2RYO	CSIG0SI	TAUJ0I3/ TAUJ0O3	URTH0 RXD/ TSG20O7/ INTP0		
	P1_8	TAUA1I1/ TPB0O	CSIG0SO	-	URTH0 TXD		
	P1_9	TAUA1I2/ TPB1O	CSIG0SC	-	URTH0SC		

Table 2-30 Handling of Unused Port Pins (2/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	I/O circuit type	Handling of Unused Pins
P2	P2_0	TAUA0I1/ TAUA0O1	TAUA0I0/ TAUA0O0	URTH1SC	CSIG0RYI/ TSG21O0	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P2_1	TAUA0I3/ TAUA0O3	TAUA0I2/ TAUA0O2	TAUA1O14	CSIG1RYI/ TSG21O7		
	P2_2	TAUA0I5/ TAUA0O5	TAUA0I4/ TAUA0O4	TAUA0O10	TSG21O1		
	P2_3	TAUA0I7/ TAUA0O7	TAUA0I6/ TAUA0O6	TAUA0O12	TSG21O2		
	P2_4	TAUA0I9/ TAUA0O9	TAUA0I8/ TAUA0O8	TAUA0O14	TSG21O3		
	P2_5	TAUA0I11/ TAUA0O11	TAUA0I10/ TAUA0O10	CSIH0CS0	TSG21O4		
	P2_6	TAUA0I13/ TAUA0O13	TAUA0I12/ TAUA0O12	CSIH0CS1	TSG21O5		
	P2_7	TAUA0I15/ TAUA0O15	TAUA0I14/ TAUA0O14	CSIH0CS2	TSG21O6		
P3	P3_0	TAUA1I1/ TAUA1O1	TAUA1I0/ TAUA1O0	URTH1 RXD/ INTP1/ CSIH0CS3	TAUA1O10	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P3_1	TAUA1I3/ TAUA1O3	TAUA1I2/ TAUA1O2	URTH1 TXD	TAUA1O12		
	P3_2	TAUA1I5/ TAUA1O5	TAUA1I4/ TAUA1O4	CSIG1SI	-		
	P3_3	TAUA1I7/ TAUA1O7	TAUA1I6/ TAUA1O6	CSIG1SO	TPB0O		
	P3_4	TAUA1I9/ TAUA1O9	TAUA1I8/ TAUA1O8	CSIG1SC	TPB1O		
	P3_5	TAUA1I11/ TAUA1O11	TAUA1I10/ TAUA1O10	TSG21 PTSI0/ ENCA1E0	URTH2 RXD/ INTP2/ FCN1TX		
	P3_6	TAUA1I13/ TAUA1O13	TAUA1I12/ TAUA1O12	TSG21 PTSI1/ ENCA1E1	FCN1RX/ URTH2 TXD		
	P3_7	TAUA1I15/ TAUA1O15	TAUA1I14/ TAUA1O14	-	CSIG2RYI		

Table 2-30 Handling of Unused Port Pins (3/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	I/O circuit type	Handling of Unused Pins
P4	P4_0	URTH1 RXD/ INTP1/ TAUA1O15	-	TSG21 PTSI2/ ENCA1EC/ FCN0TX	URTH2SC	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P4_1	URTH2 CTS/ URTH1 TXD	TSG20 PTSI0/ ENCA0E0	FCN0RX	CSIG2SI		
	P4_2	CSIG1SI/ URTH2 RTS	TSG20 PTSI1/ ENCA0E1/ OSTM1O	-	CSIG2SO		
	P4_3	CSIG1SO	TSG20 PTSI2/ ENCA0EC/ OSTM0O	URTH1 CTS	CSIG2SC		
	P4_4	CSIG1SC	TAUJ1I0/ TAUJ1O0	TAUJ0I0/ URTH1 RTS	ESO0/ INTP6/ CSIG0RYO		
	P4_5	-	TAUJ1I1/ TAUJ1O1	TAUJ0I1/ ADCA0 CNV2	ESO1/ INTP7/ CSIG1RYO		
	P4_6	TAUA1I8	TAUJ1I2/ TAUJ1O2	TAUJ0I2/ ADCA0 CNV1	ESO2/ INTP8		
	P4_7	NMI	TAUJ1I3/ TAUJ1O3	TAUJ0I3/ ADCA0 CNV0	ESO3/ INTP9		

Table 2-30 Handling of Unused Port Pins (4/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	I/O circuit type	Handling of Unused Pins
P5	P5_0	-	-	URTH0 RXD/INTP0	-	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P5_1	CSIH1CS0	URTH1 RXD/ INTP1/ CSIH0CS0	CSIH0RYI/ URTH0 TXD	ADCA0 TRG0/ INTP3		
	P5_2	CSIH1CS1	URTH2 RXD/ INTP2/ CSIH0CS1	URTH0SC	ADCA0 TRG1/ INTP4/ CSIH0RYO		
	P5_3	CSIH1CS2	ADCA0 TRG0/ INTP3/ CSIH0CS2	URTH0 CTS	ADCA0 TRG2/ INTP5		
	P5_4	CSIH1CS3	CSIH0CS3	CSIH1RYI/ URTH0 RTS	-		
	P5_5	TAUJ0I3/ TAUJ0O3	-	CSIH1RYO	-		
	P5_6	CSIH0SI	URTH1 RXD/ INTP1	-	-		
	P5_7	CSIH0SO	URTH1 TXD	-	-		
	P5_8	CSIH0SC	-	-	-		
	P5_9	CSIH0RYI	CSIH1SI	-	ADCA0 TRG1/ INTP4		
	P5_10	CSIH0RYO	CSIH1SO	-	ADCA0 TRG2/ INTP5		
	P5_11	-	CSIH1SC	-	-		
P6	P6_0	CSIG2SI/ TSG21O6	-	TAUA0O12	TSG20O0	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P6_1	CSIG2SO	URTH1 CTS	TAUA0O10	-		
	P6_2	CSIG2SC	URTH1 RTS	TAUA0O11	-		
	P6_3	FLX0RXDA	URTH1 RXD/ INTP1	TAUA1O10	-		
	P6_4	FLX0 TXENA	URTH1SC	TAUA1O11	-		
	P6_5	FLX0TXDA	URTH1 TXD	TAUA1O12	-		
	P6_6	FLX0RXDB	URTH0 RXD/ INTP0	TAUA1O13	-		
	P6_7	FLX0 TXENB	URTH0SC	TAUA1O14	-		
	P6_8	FLX0TXDB	URTH0 TXD	TAUA1O15	-		

Table 2-30 Handling of Unused Port Pins (5/5)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	I/O circuit type	Handling of Unused Pins
P7	P7_0	TAUA0I0/ TAUA0O0	ADCA0 TRG0/ INTP3/ TSG21O0	TAUJ1I0	ENCA0I0/ CSIH0CS0	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
	P7_1	TAUA0I2/ TAUA0O2	INTP10/ TSG21O1	TAUJ1I1	ENCA0I1/ CSIH0CS1		
	P7_2	TAUA0I4/ TAUA0O4	ESO1/ INTP7/ TSG21O2	TAUJ1I2	ENCA1I0/ CSIH0CS2		
	P7_3	TAUA0I6/ TAUA0O6	ESO0/ INTP6/ TSG21O3	TAUJ1I3	ENCA1I1/ CSIH0CS3		
	P7_4	TAUA0I8/ TAUA0O8	ADCA0 TRG0/ INTP3/ TSG21O4	TSG21 PTSI0/ ENCA1E0/ CSIH0CS4	TSG20 PTSI0/ ENCA0E0/ CSIH1CS0		
	P7_5	TAUA0I10/ TAUA0O10	ADCA0 TRG1/ INTP4/ TSG21O5	TSG21 PTSI1/ ENCA1E1/ CSIH0CS5	TSG20 PTSI1/ ENCA0E1/ CSIH1CS1		
	P7_6	TAUA0I12/ TAUA0O12	ADCA0 TRG2/ INTP5/ TSG21O6	TSG21 PTSI2/ ENCA1EC/ CSIH0CS6	TSG20 PTSI2/ ENCA0EC/ CSIH1CS2		
	P7_7	TAUA0I14/ TAUA0O14	TSG21O7	NMI/ CSIH0CS7	CSIH1CS3		
P8	P8_0	-	-	-	-	Type 5-W	Set the pins as inputs (PMn_m = 1) in port mode (PMCn_m = 0), and connect them to on-board pull-up resistors (PUn_m = 1). Pins shall be left open-circuit.
JP0	JP0_0	-	-	-	-	Type TDI	Set the pins as inputs (JPM0_m = 1), and connect them to on-board pull-up resistors (JPU0_m = 1). Pins shall be left open-circuit.
	JP0_1	-	-	-	-	Type TDO	
	JP0_2	-	-	-	-	Type TCK	
	JP0_3	-	-	-	-	Type TMS	
	JP0_5	-	-	-	-	Type $\overline{\text{RDY}}$	

(4) Handling of Unused Pins Other than Port Pins

Table 2-31 Handling of Unused Pins Other than Port Pins (1/2)

Pin Name	I/O circuit type	Handling
FLMD0	Type 2-W	Must be used. Connect this pin to EVSS via a resistance no less than 82 kΩ.
FLMD1	Type 2-I	Must be used. Connect this pin to EVSS directly.
$\overline{\text{RESET}}$	Type 2-I	Must be used.
$\overline{\text{DCUTRST}}$	Type 2-X	Connect this pin to EVSS.
DCUTCK	Type TCK	Leave these pins open-circuit.
DCUTDI	Type TDI	
DCUTDO	Type TDO	
DCUTMS	Type TMS	
$\overline{\text{DCUTRDY}}$	Type $\overline{\text{RDY}}$	
$\overline{\text{DCUEVTO}}$	Type 3	
ERROROUT	Type ERROROUT	Leave this pin open-circuit.
X1	-	Must be used.
X2	-	
ADCA0I1	Type 7	When the ADC is not in use: directly connect the pins to EVDD or EVSS. Unused channels when the ADC is in use: directly connect the pins to AVDD0 or AVSS0.
ADCA0I2		
ADCA0I3		
ADCA0I4		
ADCA0I5		
ADCA0I6		
ADCA0I7		
ADCA0I8		
ADCA0I9		
ADCA0I10		
ADCA0I11		
ADCA0I12		
ADCA0I13		
ADCA0I14		
ADCA0I15		
ADCA0I16		
ADCA0I17		
ADCA0I18		
ADCA0I19*		
ADCA0I20*		
ADCA0I21*		
ADCA0I22*		
AVREF0P	-	When the ADC is not in use: connect to EVDD.

Table 2-31 Handling of Unused Pins Other than Port Pins (2/2)

Pin Name	I/O circuit type	Handling
AVREF0M	-	When the ADC is not in use: connect to EVSS.
AVDD0	-	When the ADC is not in use: connect to EVDD.
AVSS0	-	When the ADC is not in use: connect to EVSS.
VDDC	-	Must be used.
VSSC	-	
VDDM	-	
VSSM	-	
OSCVDD	-	
OSCVSS	-	
CPUPLLVDD	-	
CPUPLLVSS	-	
FRPLLVDD*	-	
FRPLLVSS*	-	
FVDD	-	
FVSS	-	
EVDD	-	
EVSS	-	

Note * The µPD70F3505A does not have these pins.

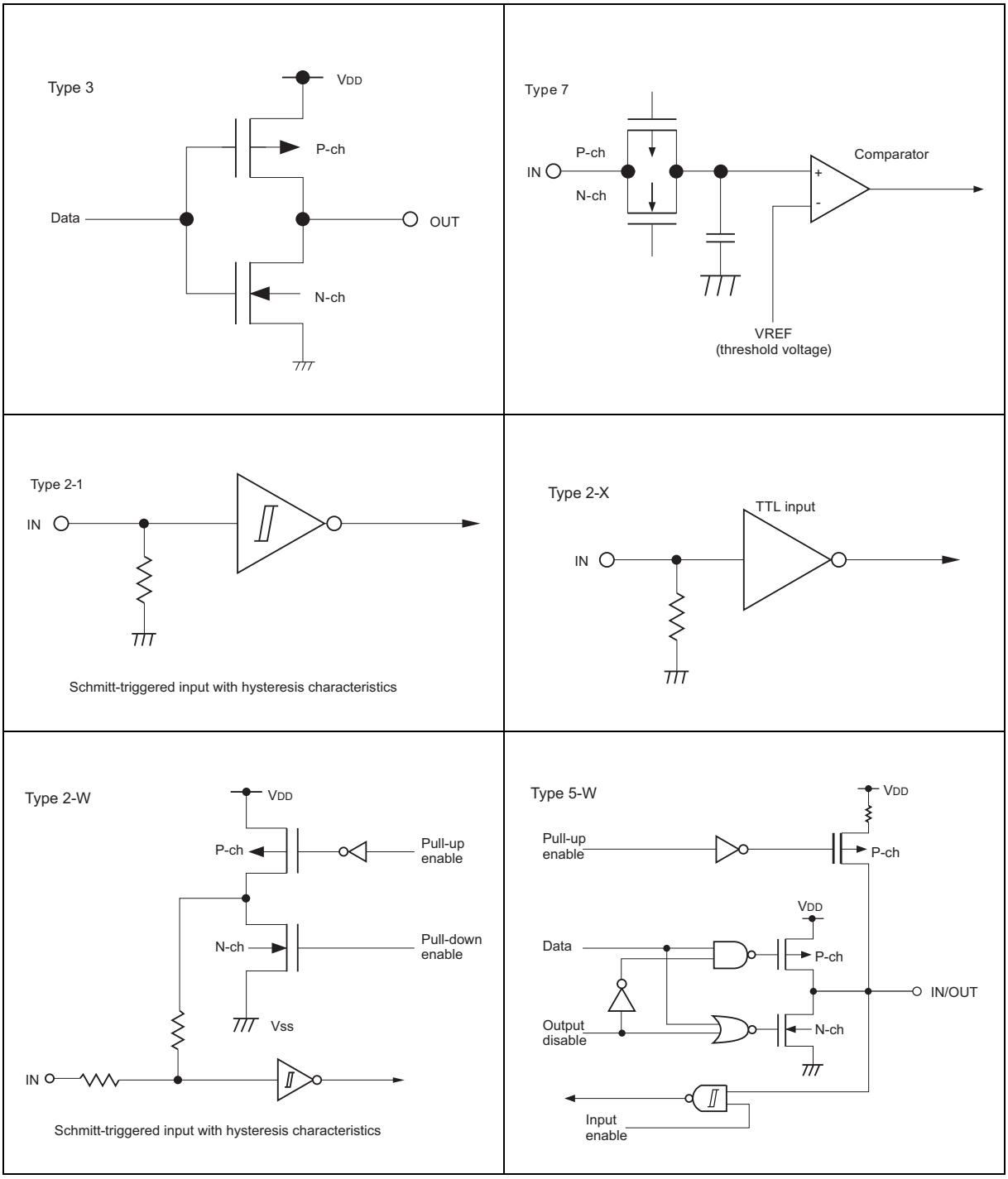


Figure 2-3 I/O circuit type(1/2)

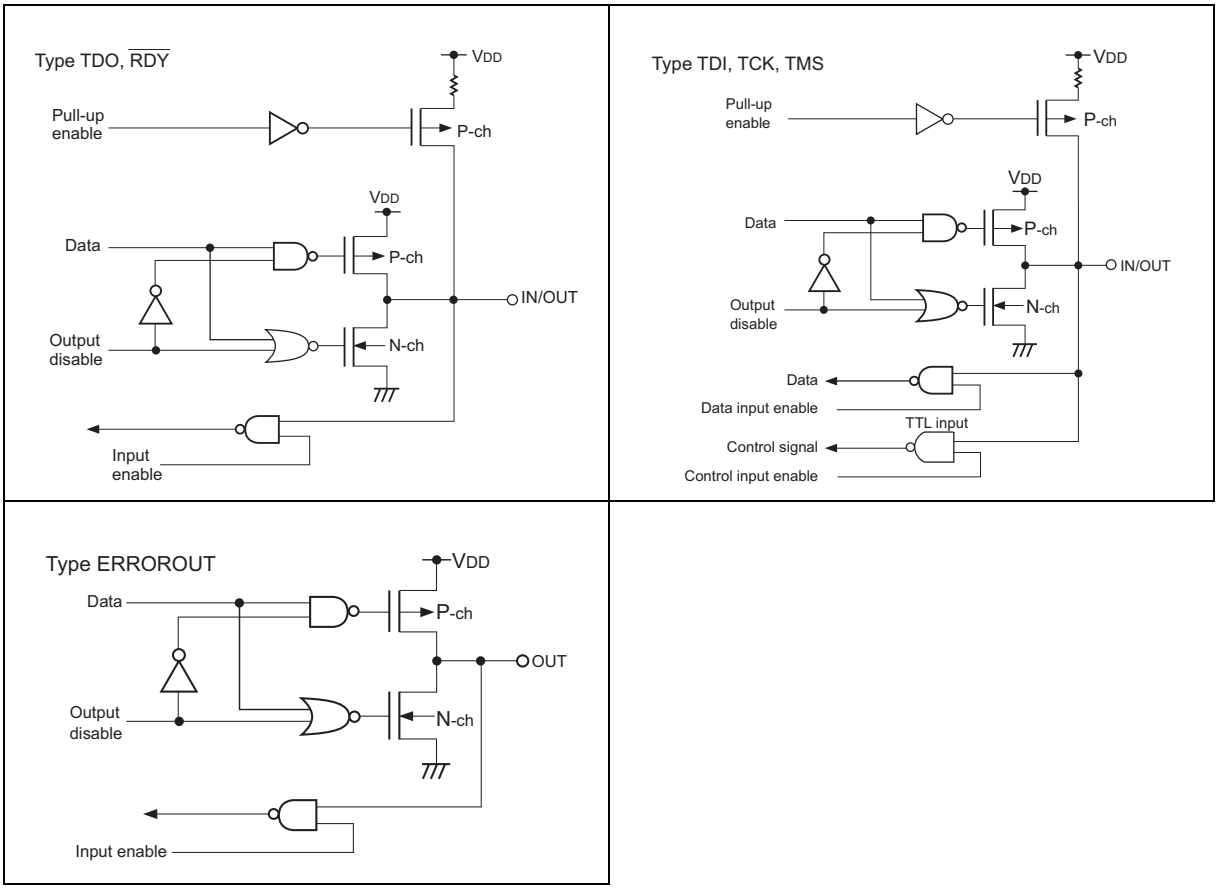


Figure 2-3 I/O circuit type(2/2)

(5) States of Port Pins

Table 2-32 List of States of Port Pins (1/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P0	P0_0	ADCA0 TRG2/ INTP5	-	CSIG0 SI	URTH0 RXD/ INTP0/ FCN1 TX	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P0_1	ADCA0 TRG1/ INTP4	-	CSIG0 SO	FCN1 RX/ URTH0 TXD				
	P0_2	ADCA0 TRG0/ INTP3	TAUA1 I2/ TAUA1 O2	CSIG0 SC	URTH0 SC				
	P0_3	-	TAUA1 I4/ TAUA1 O4	FCN0 TX	URTH0 CTS				
	P0_4	-	TAUA1 I6/ TAUA1 O6	FCN0 RX	URTH0 RTS				
	P0_5	TAUJ0 I2/ TAUJ0 O2	URTH2 RXD/ INTP2	CSIG0 SI	TSG21 O1				
	P0_6	TAUJ0 I1/ TAUJ0 O1	URTH2 TXD	CSIG0 SO	TSG21 O2				
	P0_7	TAUJ0 I0/ TAUJ0 O0	-	CSIG0 SC	TSG21 O3				
	P0_8	FCN1 TX	TSG20 CLKI	-	TSG21 O4				
	P0_9	FCN1 RX	TSG21 CLKI	-	TSG21 O5				

Table 2-32 List of States of Port Pins (2/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P1	P1_0	NMI/ CLK OUT	OSTM 1O	-	TSG21 O6	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P1_1	-	CSIG2 SI	-	TSG20 O1				
	P1_2	-	CSIG2 SO	-	TSG20 O2				
	P1_3	-	CSIG2 SC	-	TSG20 O3				
	P1_4	-	CSIH0 SI	TAUJ0 I0/ TAUJ0 O0	TSG20 O4				
	P1_5	-	CSIH0 SO	TAUJ0 I1/ TAUJ0 O1	TSG20 O5				
	P1_6	-	CSIH0 SC	TAUJ0 I2/ TAUJ0 O2	TSG20 O6				
	P1_7	TAUA1 I0/ CSIG2 RYO	CSIG0 SI	TAUJ0 I3/ TAUJ0 O3	URTH0 RXD/ TSG20 O7/ INTP0				
	P1_8	TAUA1 I1/ TPB0O	CSIG0 SO	-	URTH0 TXD				
	P1_9	TAUA1 I2/ TPB1O	CSIG0 SC	-	URTH0 SC				

Table 2-32 List of States of Port Pins (3/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P2	P2_0	TAUA0I1/TAUA0O1	TAUA0I0/TAUA0O0	URTH1SC	CSIG0RYI/TSG21O0	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P2_1	TAUA0I3/TAUA0O3	TAUA0I2/TAUA0O2	TAUA1O14	CSIG1RYI/TSG21O7				
	P2_2	TAUA0I5/TAUA0O5	TAUA0I4/TAUA0O4	TAUA0O10	TSG21O1				
	P2_3	TAUA0I7/TAUA0O7	TAUA0I6/TAUA0O6	TAUA0O12	TSG21O2				
	P2_4	TAUA0I9/TAUA0O9	TAUA0I8/TAUA0O8	TAUA0O14	TSG21O3				
	P2_5	TAUA0I11/TAUA0O11	TAUA0I10/TAUA0O10	CSIH0CS0	TSG21O4				
	P2_6	TAUA0I13/TAUA0O13	TAUA0I12/TAUA0O12	CSIH0CS1	TSG21O5				
	P2_7	TAUA0I15/TAUA0O15	TAUA0I14/TAUA0O14	CSIH0CS2	TSG21O6				

Table 2-32 List of States of Port Pins (4/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P3	P3_0	TAUA1I1/TAUA1O1	TAUA1I0/TAUA1O0	URTH1RXD/INTP1/CSIH0CS3	TAUA1O10	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P3_1	TAUA1I3/TAUA1O3	TAUA1I2/TAUA1O2	URTH1TXD	TAUA1O12				
	P3_2	TAUA1I5/TAUA1O5	TAUA1I4/TAUA1O4	CSIG1SI	-				
	P3_3	TAUA1I7/TAUA1O7	TAUA1I6/TAUA1O6	CSIG1SO	TPB0O				
	P3_4	TAUA1I9/TAUA1O9	TAUA1I8/TAUA1O8	CSIG1SC	TPB1O				
	P3_5	TAUA1I11/TAUA1O11	TAUA1I10/TAUA1O10	TSG21PTSI0/ENCA1E0	URTH2RXD/INTP2/FCN1TX				
	P3_6	TAUA1I13/TAUA1O13	TAUA1I12/TAUA1O12	TSG21PTSI1/ENCA1E1	FCN1RX/URTH2TXD				
	P3_7	TAUA1I15/TAUA1O15	TAUA1I14/TAUA1O14	-	CSIG2RYI				

Table 2-32 List of States of Port Pins (5/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P4	P4_0	URTH1 RXD/ INTP1/ TAUA1 O15	-	TSG21 PTSI2/ ENCA1 EC/ FCN0 TX	URTH2 SC	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P4_1	URTH2 CTS/ URTH1 TXD	TSG20 PTSI0/ ENCA0 E0	FCN0 RX	CSIG2 SI				
	P4_2	CSIG1 SI/ URTH2 RTS	TSG20 PTSI1/ ENCA0 E1/ OSTM 1O	-	CSIG2 SO				
	P4_3	CSIG1 SO	TSG20 PTSI2/ ENCA0 EC/ OSTM 0O	URTH1 CTS	CSIG2 SC				
	P4_4	CSIG1 SC	TAUJ1 I0/ TAUJ1 O0	TAUJ0 I0/ URTH1 RTS	ESO0/ INTP6/ CSIG0 RYO				
	P4_5	-	TAUJ1 I1/ TAUJ1 O1	TAUJ0 I1/ ADCA0 CNV2	ESO1/ INTP7/ CSIG1 RYO				
	P4_6	TAUA1 I8	TAUJ1 I2/ TAUJ1 O2	TAUJ0 I2/ ADCA0 CNV1	ESO2/ INTP8				
	P4_7	NMI	TAUJ1 I3/ TAUJ1 O3	TAUJ0 I3/ ADCA0 CNV0	ESO3/ INTP9				

Table 2-32 List of States of Port Pins (6/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P5	P5_0	-	-	URTH0 RXD/ INTP0	-	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P5_1	CSIH1 CS0	URTH1 RXD/ INTP1/ CSIH0 CS0	CSIH0 RYI/ URTH0 TXD	ADCA0 TRG0/ INTP3				
	P5_2	CSIH1 CS1	URTH2 RXD/ INTP2/ CSIH0 CS1	URTH0 SC	ADCA0 TRG1/ INTP4/ CSIH0 RYO				
	P5_3	CSIH1 CS2	ADCA0 TRG0/ INTP3/ CSIH0 CS2	URTH0 CTS	ADCA0 TRG2/ INTP5				
	P5_4	CSIH1 CS3	CSIH0 CS3	CSIH1 RYI/ URTH0 RTS	-				
	P5_5	TAUJ0 I3/ TAUJ0 O3	-	CSIH1 RYO	-				
	P5_6	CSIH0 SI	URTH1 RXD/ INTP1	-	-				
	P5_7	CSIH0 SO	URTH1 TXD	-	-	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P5_8	CSIH0 SC	-	-	-				
	P5_9	CSIH0 RYI	CSIH1 SI	-	ADCA0 TRG1/ INTP4				
	P5_10	CSIH0 RYO	CSIH1 SO	-	ADCA0 TRG2/ INTP5				
	P5_11	-	CSIH1 SC	-	-				

Table 2-32 List of States of Port Pins (7/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P6	P6_0	CSIG2 SI/ TSG21 O6	-	TAUA0 O12	TSG20 O0	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P6_1	CSIG2 SO	URTH1 CTS	TAUA0 O10	-				
	P6_2	CSIG2 SC	URTH1 RTS	TAUA0 O11	-				
	P6_3	FLX0 RXDA	URTH1 RXD/ INTP1	TAUA1 O10	-				
	P6_4	FLX0 TXENA	URTH1 SC	TAUA1 O11	-				
	P6_5	FLX0 TXDA	URTH1 TXD	TAUA1 O12	-				
	P6_6	FLX0 RXDB	URTH0 RXD/ INTP0	TAUA1 O13	-				
	P6_7	FLX0 TXENB	URTH0 SC	TAUA1 O14	-				
	P6_8	FLX0 TXDB	URTH0 TXD	TAUA1 O15	-				

Table 2-32 List of States of Port Pins (8/8)

Port Group Name	Pin Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	HALT Mode
P7	P7_0	TAUA0I0/TAUA0O0	ADCA0TRG0/INTP3/TSG21O0	TAUJ1I0	ENCA0I0/CSIH0CS0	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
	P7_1	TAUA0I2/TAUA0O2	INTP10/TSG21O1	TAUJ1I1	ENCA0I1/CSIH0CS1				
	P7_2	TAUA0I4/TAUA0O4	ESO1/INTP7/TSG21O2	TAUJ1I2	ENCA1I0/CSIH0CS2				
	P7_3	TAUA0I6/TAUA0O6	ESO0/INTP6/TSG21O3	TAUJ1I3	ENCA1I1/CSIH0CS3				
	P7_4	TAUA0I8/TAUA0O8	ADCA0TRG0/INTP3/TSG21O4	TSG21PTSI0/ENCA1E0/CSIH0CS4	TSG20PTSI0/ENCA0E0/CSIH1CS0				
	P7_5	TAUA0I10/TAUA0O10	ADCA0TRG1/INTP4/TSG21O5	TSG21PTSI1/ENCA1E1/CSIH0CS5	TSG20PTSI1/ENCA0E1/CSIH1CS1				
	P7_6	TAUA0I12/TAUA0O12	ADCA0TRG2/INTP5/TSG21O6	TSG21PTSI2/ENCA1EC/CSIH0CS6	TSG20PTSI2/ENCA0EC/CSIH1CS2				
	P7_7	TAUA0I14/TAUA0O14	TSG21O7	NMI/CSIH0CS7	CSIH1CS3				
P8	P8_0	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling Alternative output: can operate Alternative input: can operate
JP0	JP0_0	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	Port output: Retained Port input: Non-sampling
	JP0_1	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	
	JP0_2	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	
	JP0_3	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	
	JP0_5	-	-	-	-	Hi-Z	Hi-Z	Hi-Z	

(6) States of Pins Other than Port Pins

Table 2-33 List of States of Pins Other than Port Pins (1/3)

Pin Name	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	Halt Mode
FLMD0	-	-	-	-
FLMD1	-	-	-	-
$\overline{\text{RESET}}$	Low	High	High	High
$\overline{\text{DCUTRST}}$	*1	*1	Operation	Operation
$\overline{\text{DCUEVTO}}$	Low	Low	Operation*2	Operation*2
DCUTDI	Hi-Z	Hi-Z	Operation*4, *5	Operation*4
DCUTDO	Hi-Z	Hi-Z	Operation*4, *5	Operation*4
DCUTCK	Hi-Z	Hi-Z	Operation*4, *5	Operation*4
DCUTMS	Hi-Z	Hi-Z	Operation*4, *5	Operation*4
$\overline{\text{DCUTRDY}}$	Hi-Z	Hi-Z	Operation*4, *5	Operation*4
ERROROUT *3	Hi-Z	Low	Low	Operable
X1	Operation	Operation	Operation	Operation
X2	Operation	Operation	Operation	Operation

Note 1. For input of the high level to $\overline{\text{DCUTRST}}$, confirm that the low level is being output from $\overline{\text{DCUTRDY}}$ and execute the Nexus start-up sequence beforehand. See section 27.3, Notes on On-Chip Debugging, for the operating procedure.

Note 2. The output on $\overline{\text{DCUEVTO}}$ is at the low level while the internal reset signal (CPURES) is at the low level. If $\overline{\text{DCUTRST}}$ is set to the high level, this becomes the output from $\overline{\text{DCUEVTO}}$.

Note 3. ERROROUT pin is Hi-Z during an external reset and at the low level during other resets. However, the ERROROUT pin is also at the low level during an external reset in debugging mode.

Note 4. Hi-Z when $\overline{\text{DCUTRST}}$ = L.

Note 5. With development tools (such as the RAM monitor), place the DCUTRST at the high level after DCUTRDY goes to the low level.

Table 2-33 List of States of Pins Other than Port Pins (2/3)

Pin Name	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	Halt Mode
ADCA0I1	-	-	Operable	Operable
ADCA0I2				
ADCA0I3				
ADCA0I4				
ADCA0I5				
ADCA0I6				
ADCA0I7				
ADCA0I8				
ADCA0I9				
ADCA0I10				
ADCA0I11				
ADCA0I12				
ADCA0I13				
ADCA0I14				
ADCA0I15				
ADCA0I16				
ADCA0I17				
ADCA0I18				
ADCA0I19				
ADCA0I20				
ADCA0I21				
ADCA0I22				

Table 2-33 List of States of Pins Other than Port Pins (3/3)

Pin Name	Reset	BIST for Self-diagnosis is Running	Immediately after Releasing the CPU Core from Reset	Halt Mode
AVREF0P	-	-	-	-
AVREF0M				
AVDD0				
AVSS0				
VDDC				
VSSC				
VDDM				
VSSM				
OSCVDD				
OSCVSS				
CPUPLLVDD				
CPUPLLVSS				
FRPLLVDD				
FRPLLVSS				
FVDD				
FVSS				
EVDD				
EVSS				

(7) Pin States under Specified Conditions

1. Port pins are in the Hi-Z state during an external reset in on-chip debugging mode.
   Self-diagnosis BIST is not executed after release from an external reset in on-chip debugging mode.
2. Pin states in flash memory programming mode are identical to those in single chip mode.
   Self-diagnosis BIST is not executed in flash memory programming mode.

2.5 Functions of Pull-Up and Pull-Down Resistors

This section describes target pins for pull-up and pull-down resistors.

2.5.1 Details of Pull-Up and Pull-Down Resistors

(1) Pull-Up and Pull-Down Resistors for Port Pins

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (1/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P0	P0_0	ADCA0 TRG2/ INTP5	-	CSIG0SI	URTH0 RXD/ INTP0/ FCN1TX	Software pull-up (initial value: off)
	P0_1	ADCA0 TRG1/ INTP4	-	CSIG0SO	FCN1RX/ URTH0 TXD	
	P0_2	ADCA0 TRG0/ INTP3	TAUA1I2/ TAUA1O2	CSIG0SC	URTH0SC	
	P0_3	-	TAUA1I4/ TAUA1O4	FCN0TX	URTH0 CTS	
	P0_4	-	TAUA1I6/ TAUA1O6	FCN0RX	URTH0 RTS	
	P0_5	TAUJ0I2/ TAUJ0O2	URTH2 RXD/ INTP2	CSIG0SI	TSG21O1	
	P0_6	TAUJ0I1/ TAUJ0O1	URTH2 TXD	CSIG0SO	TSG21O2	
	P0_7	TAUJ0I0/ TAUJ0O0	-	CSIG0SC	TSG21O3	
	P0_8	FCN1TX	TSG20 CLKI	-	TSG21O4	
	P0_9	FCN1RX	TSG21 CLKI	-	TSG21O5	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (2/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P1	P1_0	NMI/ CLKOUT	OSTM1O	-	TSG21O6	Software pull-up (initial value: off)
	P1_1	-	CSIG2SI	-	TSG20O1	
	P1_2	-	CSIG2SO	-	TSG20O2	
	P1_3	-	CSIG2SC	-	TSG20O3	
	P1_4	-	CSIH0SI	TAUJ0I0/ TAUJ0O0	TSG20O4	
	P1_5	-	CSIH0SO	TAUJ0I1/ TAUJ0O1	TSG20O5	
	P1_6	-	CSIH0SC	TAUJ0I2/ TAUJ0O2	TSG20O6	
	P1_7	TAUA1I0/ CSIG2 RYO	CSIG0SI	TAUJ0I3/ TAUJ0O3	URTH0 RXD/ TSG20O7/ INTP0	
	P1_8	TAUA1I1/ TPB0O	CSIG0SO	-	URTH0 TXD	
	P1_9	TAUA1I2/ TPB1O	CSIG0SC	-	URTH0SC	
P2	P2_0	TAUA0I1/ TAUA0O1	TAUA0I0/ TAUA0O0	URTH1SC	CSIG0RYI/ TSG21O0	Software pull-up (initial value: off)
	P2_1	TAUA0I3/ TAUA0O3	TAUA0I2/ TAUA0O2	TAUA1O 14	CSIG1RYI/ TSG21O7	
	P2_2	TAUA0I5/ TAUA0O5	TAUA0I4/ TAUA0O4	TAUA0O 10	TSG21O1	
	P2_3	TAUA0I7/ TAUA0O7	TAUA0I6/ TAUA0O6	TAUA0O 12	TSG21O2	
	P2_4	TAUA0I9/ TAUA0O9	TAUA0I8/ TAUA0O8	TAUA0O 14	TSG21O3	
	P2_5	TAUA0I11T AUA0 O11	TAUA0I10T AUA0 O10	CSIH0 CS0	TSG21O4	
	P2_6	TAUA0I13T AUA0 O13	TAUA0I12T AUA0 O12	CSIH0 CS1	TSG21O5	
	P2_7	TAUA0I15T AUA0 O15	TAUA0I14T AUA0 O14	CSIH0 CS2	TSG21O6	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (3/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P3	P3_0	TAUA1I1/ TAUA1O1	TAUA1I0/ TAUA1O0	URTH1 RXD/ INTP1/ CSIH0 CS3	TAUA1 O10	Software pull-up (initial value: off)
	P3_1	TAUA1I3/ TAUA1O3	TAUA1I2/ TAUA1O2	URTH1 TXD	TAUA1 O12	
	P3_2	TAUA1I5/ TAUA1O5	TAUA1I4/ TAUA1O4	CSIG1SI	-	
	P3_3	TAUA1I7/ TAUA1O7	TAUA1I6/ TAUA1O6	CSIG1SO	TPB0O	
	P3_4	TAUA1I9/ TAUA1O9	TAUA1I8/ TAUA1O8	CSIG1SC	TPB1O	
	P3_5	TAUA1I11/ TAUA1O11	TAUA1I10/ TAUA1 O10	TSG21 PTSI0/ ENCA1E0	URTH2 RXD/ INTP2/ FCN1TX	
	P3_6	TAUA1I13/ TAUA1 O13	TAUA1I12/ TAUA1 O12	TSG21 PTSI1/ ENCA1E1	FCN1RX/ URTH2 TXD	
	P3_7	TAUA1I15/ TAUA1 O15	TAUA1I14/ TAUA1 O14	-	CSIG2RYI	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (4/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P4	P4_0	URTH1 RXD/ INTP1/ TAUA1O 15	-	TSG21 PTSI2/ ENCA1EC/ FCN0TX	URTH2SC	Software pull-up (initial value: off)
	P4_1	URTH2 CTS/ URTH1 TXD	TSG20 PTSI0/ ENCA0E0	FCN0RX	CSIG2SI	
	P4_2	CSIG1SI/ URTH2 RTS	TSG20 PTSI1/ ENCA0E1/ OSTM1O	-	CSIG2SO	
	P4_3	CSIG1SO	TSG20 PTSI2/ ENCA0EC/ OSTM0O	URTH1 CTS	CSIG2SC	
	P4_4	CSIG1SC	TAUJ1I0/ TAUJ1O0	TAUJ0I0/ URTH1 RTS	ESO0/ INTP6/ CSIG0 RYO	
	P4_5	-	TAUJ1I1/ TAUJ1O1	TAUJ0I1/ ADCA0 CNV2	ESO1/ INTP7/ CSIG1 RYO	
	P4_6	TAUA1I8	TAUJ1I2/ TAUJ1O2	TAUJ0I2/ ADCA0 CNV1	ESO2/ INTP8	
	P4_7	NMI	TAUJ1I3/ TAUJ1O3	TAUJ0I3/ ADCA0 CNV0	ESO3/ INTP9	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (5/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P5	P5_0	-	-	URTH0 RXD/INTP0	-	Software pull-up (initial value: off)
	P5_1	CSIH1 CS0	URTH1 RXD/ INTP1/ CSIH0 CS0	CSIH0RYI/ URTH0 TXD	ADCA0 TRG0/ INTP3	
	P5_2	CSIH1 CS1	URTH2 RXD/ INTP2/ CSIH0 CS1	URTH0SC	ADCA0 TRG1/ INTP4/ CSIH0 RYO	
	P5_3	CSIH1 CS2	ADCA0 TRG0/ INTP3/ CSIH0CS2	URTH0 CTS	ADCA0 TRG2/ INTP5	
	P5_4	CSIH1CS3	CSIH0CS3	CSIH1RYI/ URTH0 RTS	-	
	P5_5	TAUJ0I3/ TAUJ0O3	-	CSIH1 RYO	-	
	P5_6	CSIH0SI	URTH1 RXD/ INTP1	-	-	
	P5_7	CSIH0SO	URTH1 TXD	-	-	
	P5_8	CSIH0SC	-	-	-	
	P5_9	CSIH0RYI	CSIH1SI	-	ADCA0 TRG1/ INTP4	
	P5_10	CSIH0 RYO	CSIH1SO	-	ADCA0 TRG2/ INTP5	
	P5_11	-	CSIH1SC	-	ESO0/ INTP6	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (6/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P6	P6_0	CSIG2SI/ TSG21O6	-	TAUA0 O12	TSG20O0	Software pull-up (initial value: off)
	P6_1	CSIG2SO	URTH1 CTS	TAUA0 O10	-	
	P6_2	CSIG2SC	URTH1 RTS	TAUA0 O11	-	
	P6_3	FLX0 RXDA	URTH1 RXD/ INTP1	TAUA1 O10	-	
	P6_4	FLX0 TXENA	URTH1SC	TAUA1 O11	-	
	P6_5	FLX0 TXDA	URTH1 TXD	TAUA1 O12	-	
	P6_6	FLX0 RXDB	URTH0 RXD/ INTP0	TAUA1 O13	-	
	P6_7	FLX0 TXENB	URTH0SC	TAUA1 O14	-	
	P6_8	FLX0 TXDB	URTH0 TXD	TAUA1 O15	-	

Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins (7/7)

Port Group Name	Port Name	Alternative Mode 1	Alternative Mode 2	Alternative Mode 3	Alternative Mode 4	Pull-Up Resistor, Pull-Down Resistor
P7	P7_0	TAUA0I0/ TAUA0O0	ADCA0 TRG0/ INTP3/ TSG21O0	TAUJ1I0	ENCA0I0/ CSIH0 CS0	Software pull-up (initial value: off)
	P7_1	TAUA0I2/ TAUA0O2	INTP10/ TSG21O1	TAUJ1I1	ENCA0I1/ CSIH0 CS1	
	P7_2	TAUA0I4/ TAUA0O4	ESO1/ INTP7/ TSG21O2	TAUJ1I2	ENCA1I0/ CSIH0 CS2	
	P7_3	TAUA0I6/ TAUA0O6	ESO0/ INTP6/ TSG21O3	TAUJ1I3	ENCA1I1/ CSIH0 CS3	
	P7_4	TAUA0I8/ TAUA0O8	ADCA0 TRG0/ INTP3/ TSG21O4	TSG21 PTSI0/ ENCA1E0/ CSIH0 CS4	TSG20 PTSI0/ ENCA0E0/ CSIH1 CS0	
	P7_5	TAUA0I10/ TAUA0O 10	ADCA0 TRG1/ INTP4/ TSG21O5	TSG21 PTSI1/ ENCA1E1/ CSIH0 CS5	TSG20 PTSI1/ ENCA0E1/ CSIH1 CS1	
	P7_6	TAUA0I12/ TAUA0O 12	ADCA0 TRG2/ INTP5/ TSG21O6	TSG21 PTSI2/ ENCA1EC/ CSIH0 CS6	TSG20 PTSI2/ ENCA0EC/ CSIH1 CS2	
	P7_7	TAUA0I14/ TAUA0 O14	TSG21O7	NMI/ CSIH0 CS7	CSIH1 CS3	
P8	P8_0	-	-	-	-	Software pull-up (initial value: off)
JP0	JP0_0	-	-	-	-	Software pull-up (initial value: off)
	JP0_1	-	-	-	-	Software pull-up (initial value: off)
	JP0_2	-	-	-	-	Software pull-up (initial value: off)
	JP0_3	-	-	-	-	Software pull-up (initial value: off)
	JP0_5	-	-	-	-	Software pull-up (initial value: off)

(2) Pull-Up and Pull-Down Resistors for Pins Other than Port Pins

Table 2-35 List of Pull-Up and Pull-Down Resistors for Pins Other than Port Pins (1/3)

Pin Name	Pull-Up Resistor, Pull Down Resistor
FLMD0	Software pull-up/pull-down (initial value: pull-down on)
FLMD1	Always pull-down
$\overline{\text{RESET}}$	Always pull-down
$\overline{\text{DCUTRST}}$	Always pull-down
$\overline{\text{DCUEVTO}}$	-
DCUTDI	Connected to an internal pull-up resistor
DCUTDO	This is connected to an internal pull-up resistor. However, the pull-up resistor is not connected when DCUTDO is an output.
DCUTCK	Connected to an internal pull-up resistor
DCUTMS	Connected to an internal pull-up resistor
$\overline{\text{DCUTRDY}}$	-
ERROROUT	-
X1	-
X2	-

Table 2-35 List of Pull-Up and Pull-Down Resistors for Pins Other than Port Pins (2/3)

Pin Name	Pull-Up Resistor, Pull Down Resistor
ADCA0I1	Software pull-down (initial value: off)
ADCA0I2	
ADCA0I3	
ADCA0I4	
ADCA0I5	
ADCA0I6	
ADCA0I7	
ADCA0I8	
ADCA0I9	
ADCA0I10	
ADCA0I11	
ADCA0I12	
ADCA0I13	
ADCA0I14	
ADCA0I15	
ADCA0I16	
ADCA0I17	
ADCA0I18	
ADCA0I19	
ADCA0I20	
ADCA0I21	
ADCA0I22	
AVREF0P	-
AVREF0M	-
AVDD0	-
AVSS0	-

Table 2-35 List of Pull-Up and Pull-Down Resistors for Pins Other than Port Pins (3/3)

Pin Name	Pull-Up Resistor, Pull Down Resistor
VDDC	-
VSSC	
VDDM	
VSSM	
OSCVDD	
OSCVSS	
CPUPLLVDD	
CPUPLLVSS	
FRPLLVDD	
FRPLLVSS	
FVDD	
FVSS	
EVDD	
EVSS	

2.6 Port 0

This section describes alternative functions and control registers of port 0.

2.6.1 Alternative Functions

(1) Alternative Functions of Port 0

Table 2-36 List of Alternative Functions of Port 0

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC=1, PFCE=0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P0_0	INTP5/ ADCA0 TRG2	-	-	-	CSIG0SI	-	URTH0 RXD/ INTP0	FCN1TX
P0_1	INTP4/ ADCA0 TRG1	-	-	-	CSIG0SO *1	CSIG0SO	FCN1RX	URTH0 TXD
P0_2	INTP3/ ADCA0 TRG0	-	TAUA1I2	TAUA1O2	CSIG0SC	CSIG0SC	URTH0SC	URTH0SC
P0_3	-	-	TAUA1I4	TAUA1O4	-	FCN0TX	URTH0 CTS	-
P0_4	-	-	TAUA1I6	TAUA1O6	FCN0RX	-	-	URTH0 RTS
P0_5	TAUJ0I2	TAUJ0O2	URTH2 RXD/ INTP2	-	CSIG0SI	-	-	TSG21O1
P0_6	TAUJ0I1	TAUJ0O1	-	URTH2 TXD	CSIG0SO *1	CSIG0SO	-	TSG21O2
P0_7	TAUJ0I0	TAUJ0O0	-	-	CSIG0SC	CSIG0SC	-	TSG21O3
P0_8	-	FCN1TX	TSG20 CLKI	-	-	-	-	TSG21O4
P0_9	FCN1RX	-	TSG21 CLKI	-	-	-	-	TSG21O5

Note 1. Corresponds to data consistency checking.

2.6.2 List of Control Registers

(1) Port 0 Control Registers

Table 2-37 List of Port 0 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PSR0	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PPR0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PM0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMC0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PFC0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PFCE0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PNOT0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMSR0	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMCSR0	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PIBC0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PBDC0	—	√	—	—	—	—	—	—	—	—	—	—	√	—	—	—	—	√	—
PIPC0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	—	√	√	√	√
PU0	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√

2.7 Port 1

This section describes alternative functions and control registers of port 1.

2.7.1 Alternative Functions

(1) Alternative Functions of Port 1

Table 2-38 List of Alternative Functions of Port 1

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P1_0	NMI	CLKOUT	-	OSTM1O	-	-	-	TSG21O6
P1_1	-	-	CSIG2SI	-	-	-	-	TSG20O1
P1_2	-	-	CSIG2SO *1	CSIG2SO	-	-	-	TSG20O2
P1_3	-	-	CSIG2SC	CSIG2SC	-	-	-	TSG20O3
P1_4	-	-	CSIH0SI	-	TAUJ0I0	TAUJ0O0	-	TSG20O4
P1_5	-	-	CSIH0SO *1	CSIH0SO	TAUJ0I1	TAUJ0O1	-	TSG20O5
P1_6	-	-	CSIH0SC	CSIH0SC	TAUJ0I2	TAUJ0O2	-	TSG20O6
P1_7	TAUA1I0	CSIG2RYO	CSIG0SI	-	TAUJ0I3	TAUJ0O3	URTH0 RXD/ INTP0	TSG20O7
P1_8	TAUA1I1	TPB0O	CSIG0SO *1	CSIG0SO	-	-	-	URTH0 TXD
P1_9	TAUA1I2	TPB1O	CSIG0SC	CSIG0SC	-	-	URTH0SC	URTH0SC

Note 1. Corresponds to data consistency checking.

2.7.2 List of Control Registers

(1) Port 1 Control Registers

Table 2-39 List of Port 1 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PSR1	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PPR1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PM1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMC1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PFC1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PFCE1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PNOT1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMSR1	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PMCSR1	—	—	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PIBC1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
PBDC1	—	√	—	—	—	—	—	—	—	—	√	—	—	√	—	—	√	—	—
PIPC1	—	√	—	—	—	—	—	—	—	√	—	—	√	√	√	√	√	√	√
PU1	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√

2.8 Port 2

This section describes alternative functions and control registers of port 2.

2.8.1 Alternative Functions

(1) Alternative Functions of Port 2

Table 2-40 List of Alternative Functions of Port 2

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P2_0	TAUA0I1	TAUA0O1	TAUA0I0	TAUA0O0	URTH1SC	URTH1SC	CSIG0RYI	TSG21O0
P2_1	TAUA0I3	TAUA0O3	TAUA0I2	TAUA0O2	-	TAUA1O14	CSIG1RYI	TSG21O7
P2_2	TAUA0I5	TAUA0O5	TAUA0I4	TAUA0O4	-	TAUA0O10	-	TSG21O1
P2_3	TAUA0I7	TAUA0O7	TAUA0I6	TAUA0O6	-	TAUA0O12	-	TSG21O2
P2_4	TAUA0I9	TAUA0O9	TAUA0I8	TAUA0O8	-	TAUA0O14	-	TSG21O3
P2_5	TAUA0I11	TAUA0O11	TAUA0I10	TAUA0O10	-	CSIH0CS0	-	TSG21O4
P2_6	TAUA0I13	TAUA0O13	TAUA0I12	TAUA0O12	-	CSIH0CS1	-	TSG21O5
P2_7	TAUA0I15	TAUA0O15	TAUA0I14	TAUA0O14	-	CSIH0CS2	-	TSG21O6

2.8.2 List of Control Registers

(1) Port 2 Control Register

Table 2-41 List of Port 2 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PSR2	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PPR2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PM2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMC2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFC2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFCE2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PNOT2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMSR2	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMCSR2	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIBC2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIPC2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PU2	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PODC2	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PPROTS2	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PPCMD2	√	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√

2.9 Port 3

This section describes alternative functions and control registers of port 3.

2.9.1 Alternative Functions

(1) Port 3 Control Registers

Table 2-42 List of Alternative Functions of Port 3

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P3_0	TAUA1I1	TAUA1O1	TAUA1I0	TAUA1O0	URTH1 RXD/ INTP1	CSIH0CS3	-	TAUA1O10
P3_1	TAUA1I3	TAUA1O3	TAUA1I2	TAUA1O2	-	URTH1 TXD	-	TAUA1O12
P3_2	TAUA1I5	TAUA1O5	TAUA1I4	TAUA1O4	CSIG1SI	-	-	-
P3_3	TAUA1I7	TAUA1O7	TAUA1I6	TAUA1O6	CSIG1SO *1	CSIG1SO	-	TPB0O
P3_4	TAUA1I9	TAUA1O9	TAUA1I8	TAUA1O8	CSIG1SC	CSIG1SC	-	TPB1O
P3_5	TAUA1I11	TAUA1O11	TAUA1I10	TAUA1O10	TSG21 PTSI0/ ENCA1E0	-	URTH2 RXD/ INTP2	FCN1TX
P3_6	TAUA1I13	TAUA1O13	TAUA1I12	TAUA1O12	TSG21 PTSI1/ ENCA1E1	-	FCN1RX	URTH2 TXD
P3_7	TAUA1I15	TAUA1O15	TAUA1I14	TAUA1O14	-	-	CSIG2RYI	-

Note 1. Corresponds to data consistency checking.

2.9.2 List of Control Registers

(1) List of Port 3 Control Registers

Table 2-43 List of Port 3 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PSR3	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PPR3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PM3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMC3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFC3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFCE3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PNOT3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMSR3	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMCSR3	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIBC3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PBDC3	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—
PIPC3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	—	—	√	√
PU3	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PODC3	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PPROTS3	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PPCMD3	√	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√

2.10 Port 4

This section describes alternative functions and control registers of port 4.

2.10.1 Alternative Functions

(1) Alternative Functions of Port 4

Table 2-44 List of Alternative Functions of Port 4

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P4_0	URTH1 RXD/ INTP1	TAUA1O15	-	-	TSG21 PTSI2/ ENCA1EC	FCN0TX	URTH2SC	URTH2SC
P4_1	URTH2 CTS	URTH1 TXD	TSG20 PTSI0/ ENCA0E0	-	FCN0RX	-	CSIG2SI	-
P4_2	CSIG1SI	URTH2 RTS	TSG20 PTSI1/ ENCA0E1	OSTM1O	-	-	CSIG2SO *1	CSIG2SO
P4_3	CSIG1SO *1	CSIG1SO	TSG20 PTSI2/ ENCA0EC	OSTM0O	URTH1 CTS	-	CSIG2SC	CSIG2SC
P4_4	CSIG1SC	CSIG1SC	TAUJ1I0	TAUJ1O0	TAUJ0I0	URTH1 RTS	ESO0/ INTP6	CSIG0RYO
P4_5	-	-	TAUJ1I1	TAUJ1O1	TAUJ0I1	ADCA0 CNV2	ESO1/ INTP7	CSIG1RYO
P4_6	TAUA1I8	-	TAUJ1I2	TAUJ1O2	TAUJ0I2	ADCA0 CNV1	ESO2/ INTP8	-
P4_7	NMI	-	TAUJ1I3	TAUJ1O3	TAUJ0I3	ADCA0 CNV0	ESO3/ INTP9	-

Note 1. Corresponds to data consistency checking.

2.10.2 List of Control Registers

(1) Port 4 Control Registers

Table 2-45 List of Port 4 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PSR4	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PPR4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PM4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMC4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFC4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFCE4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PNOT4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMSR4	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMCSR4	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIBC4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PBDC4	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	—	—
PIPC4	—	√	—	—	—	—	—	—	—	—	—	—	—	—	√	√	—	—	√
PU4	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√

2.11 Port 5

This section describes alternative functions and control registers of port 5.

2.11.1 Alternative Functions

(1) Alternative Functions of Port 5

Table 2-46 List of Alternative Functions of Port 5

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P5_0	-	-	-	-	URTH0 RXD/INTP0	-	-	-
P5_1	-	CSIH1CS0	URTH1 RXD/ INTP1	CSIH0CS0	CSIH0RYI	URTH0 TXD	ADCA0 TRG0/ INTP3	-
P5_2	-	CSIH1CS1	URTH2 RXD/ INTP2	CSIH0CS1	URTH0SC	URTH0SC	ADCA0 TRG1/ INTP4	CSIH0RYO
P5_3	-	CSIH1CS2	ADCA0 TRG0/ INTP3	CSIH0CS2	URTH0 CTS	-	ADCA0 TRG2/ INTP5	-
P5_4	-	CSIH1CS3	-	CSIH0CS3	CSIH1RYI	URTH0 RTS	-	-
P5_5	TAUJ0I3	TAUJ0O3	-	-	-	CSIH1RYO	-	-
P5_6	CSIH0SI	-	URTH1 RXD/ INTP1	-	-	-	-	-
P5_7	CSIH0SO *1	CSIH0SO	-	URTH1 TXD	-	-	-	-
P5_8	CSIH0SC	CSIH0SC	-	-	-	-	-	-
P5_9	CSIH0RYI	-	CSIHISI	-	-	-	ADCA0 TRG1/ INTP4	-
P5_10	-	CSIH0RYO	CSIH1SO *1	CSIH1SO	-	-	ADCA0 TRG2/ INTP5	-
P5_11	-	-	CSIH1SC	CSIH1SC	-	-	-	-

Note 1. Corresponds to data consistency checking.

2.11.2 List of Control Registers

(1) Port 5 Control Registers

Table 2-47 List of Port 5 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PSR5	—	—	√	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PPR5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PM5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PMC5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PFC5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PFCE5	—	√	—	—	—	—	—	—	√	√	—	—	—	√	√	√	√	√	√
PNOT5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PMSR5	—	—	√	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PMCSR5	—	—	√	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
				—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PIBC5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√
PBDC5	—	√	—	—	—	—	—	—	√	—	—	√	—	—	—	—	—	—	—
PIPC5	—	√	—	—	—	—	—	√	—	—	√	—	—	—	—	—	√	—	—
PU5	—	√	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√

2.12 Port 6

This section describes alternative functions and control registers of port 6.

2.12.1 Alternative Functions

(1) Alternative Functions of Port 6

Table 2-48 List of Alternative Functions of Port 6

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P6_0	CSIG2SI	TSG21O6	-	-	-	TAUA0O12	-	TSG20O0
P6_1	CSIG2SO *1	CSIG2SO	URTH1 CTS	-	-	TAUA0O10	-	-
P6_2	CSIG2SC	CSIG2SC	-	URTH1 RTS	-	TAUA0O11	-	-
P6_3	FLX0RXDA	-	URTH1 RXD/ INTP1	-	-	TAUA1O10	-	-
P6_4	-	FLX0 TXENA	URTH1SC	URTH1SC	-	TAUA1O11	-	-
P6_5	-	FLX0TXDA	-	URTH1 TXD	-	TAUA1O12	-	-
P6_6	FLX0RXDB	-	URTH0 RXD/ INTP0	-	-	TAUA1O13	-	-
P6_7	-	FLX0 TXENB	URTH0SC	URTH0SC	-	TAUA1O14	-	-
P6_8	-	FLX0TXDB	-	URTH0 TXD	-	TAUA1O15	-	-

Note 1. Corresponds to data consistency checking.

2.12.2 List of Control Registers

(1) Port 6 Control Registers

Table 2-49 List of Port 6 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PSR6	—	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PPR6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PM6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PMC6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PFC6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PFCE6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PNOT6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PMSR6	—	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PMCSR6	—	—	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PIBC6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PBDC6	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—
PIPC6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PU6	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
PDSC6	—	—	√	—	—	—	—	—	—	—	√	√	—	√	√	—	—	—	—
				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PIS6	—	√	—	—	—	—	—	—	—	—	—	—	√	—	—	√	—	—	—
PPROTS6	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PPCMD6	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√

2.13 Port 7

This section describes alternative functions and control registers of port 7.

2.13.1 Alternative Functions

(1) Alternative Functions of Port 7

Table 2-50 List of Alternative Functions of Port 7

Port Mode (PMC = 0)	Control Mode (PMC = 1)							
	Alternative Mode 1 (PFC = 0, PFCE = 0)		Alternative Mode 2 (PFC = 1, PFCE = 0)		Alternative Mode 3 (PFC = 0, PFCE = 1)		Alternative Mode 4 (PFC = 1, PFCE = 1)	
	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)	Input (PM = 1)	Output (PM = 0)
P7_0	TAUA0I0	TAUA0O0	ADCA0 TRG0/ INTP3	TSG21O0	TAUJ1I0	-	ENCA0I0	CSIH0CS0
P7_1	TAUA0I2	TAUA0O2	INTP10	TSG21O1	TAUJ1I1	-	ENCA0I1	CSIH0CS1
P7_2	TAUA0I4	TAUA0O4	ESO1/ INTP7	TSG21O2	TAUJ1I2	-	ENCA1I0	CSIH0CS2
P7_3	TAUA0I6	TAUA0O6	ESO0/ INTP6	TSG21O3	TAUJ1I3	-	ENCA1I1	CSIH0CS3
P7_4	TAUA0I8	TAUA0O8	ADCA0 TRG0/ INTP3	TSG21O4	TSG21 PTSI0/ ENCA1E0	CSIH0CS4	TSG20 PTSI0/ ENCA0E0	CSIH1CS0
P7_5	TAUA0I10	TAUA0O10	ADCA0 TRG1/ INTP4	TSG21O5	TSG21 PTSI1/ ENCA1E1	CSIH0CS5	TSG20 PTSI1/ ENCA0E1	CSIH1CS1
P7_6	TAUA0I12	TAUA0O12	ADCA0 TRG2/ INTP5	TSG21O6	TSG21 PTSI2/ ENCA1EC	CSIH0CS6	TSG20 PTSI2/ ENCA0EC	CSIH1CS2
P7_7	TAUA0I14	TAUA0O14	-	TSG21O7	NMI	CSIH0CS7	-	CSIH1CS3

2.13.2 List of Control Registers

(1) Port 7 Control Registers

Table 2-51 List of Port 7 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PSR7	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PPR7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PM7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMC7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFC7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PFCE7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PNOT7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMSR7	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PMCSR7	—	—	√	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
				—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIBC7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√
PIPC7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	—
PU7	—	√	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√

2.14 Port 8

This section describes alternative functions and control registers of port 8.

2.14.1 Alternative Functions

(1) Alternative Functions of Port 8

Port 8 does not have any alternative functions.

2.14.2 List of Control Registers

(1) Port 8 Control Registers

Table 2-52 List of Port 8 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
P8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PSR8	—	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PPR8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PM8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PNOT8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PMSR8	—	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PIBC8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√
PU8	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√

2.15 JTAG Port 0

This section describes alternative functions and control registers of JTAG port 0.

2.15.1 Alternative Functions

(1) Alternative Functions of JTAG Port 0

JTAG port 0 does not have any alternative functions.

2.15.2 List of Control Registers

(1) JTAG Port 0 Control Registers

Table 2-53 List of JTAG Port 0 Control Registers

Name	Number of Bits			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	8	16	32																
JP0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPSR0	—	—	√	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
				—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPPR0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPM0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPNOT0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPMSR0	—	—	√	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
				—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPIBC0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√
JPU0	√	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√

2.16 Cancelling Noise on Pins

This section describes the noise-filtering insertion pin, filter groups, noise-cancellation sampling clock, and control registers.

2.16.1 Details of Noise Cancellation

(1) Digital Filter Function

The input signal is sampled with the sampling frequency fs.
If the input is sampled at the same level in a specified number of successive samples, the signal level is judged as valid and the filter output signal is changed accordingly.
If the external signal level changes within the specified number of samples (at the same level), the new signal level is judged to be noise–or a spike–and the level of the output signal from the filter does not change.
The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.
Both parameters can be specified.

- DNFAnCTL.DNFAnPRS[2:0] allows to select the sampling frequency according to
  $fs = f_{DNFATCKI} / 2^{DNFAnPRS[2:0]}$
  where $f_{DNFATCKI}$ is the frequency of the DNFATCKI clock.
- DNFAnCTL.DNFAnNFSTS[1:0] specifies the number (2 to 5) of same level samples.

External signal pulses shorter than “(DNFAnNFSTS[1:0] - 1) × 1/fs” are always judged to be invalid and cancelled.

Caution If noise is sampled as the same level over the specified number of successive samples, the noise will not be cancelled.

External signal pulses longer than “DNFAnNFSTS[1:0] × 1/fs” are always judged to be valid and passed on to the filter output.

External signal pulses with a width between “(DNFAnNFSTS[1:0] -1) × 1/fs” and “DNFAnNFSTS[1:0] × 1/fs” may be cancelled or judged to be valid.

(2) Noise Filter Insertion Pins and Filter Groups

Table 2-54 List of Noise Filter Insertion Pins and Filter Groups (1/5)

Port Group Name	Port Pin	Filter Insertion Pin	Filter Group
P0	P0_0	INTP0	DNF_INTP-G0
		INTP5	DNF_INTP-G1
		ADCA0TRG2	ADTRG
	P0_1	INTP4	DNF_INTP-G1
		ADCA0TRG1	ADTRG
	P0_2	INTP3	DNF_INTP-G1
		TAUA1I2	DNF_TAUA1-G1
		ADCA0TRG0	ADTRG
	P0_3	TAUA1I4	DNF_TAUA1-G2
	P0_4	TAUA1I6	DNF_TAUA1-G3
	P0_5	TAUJ0I2	DNF_TAUJ0-G2
		INTP2	DNF_INTP-G0
	P0_6	TAUJ0I1	DNF_TAUJ0-G1
	P0_7	TAUJ0I0	DNF_TAUJ0-G0
	P0_8	TSG20CLKI	DNF_TSG20-G2
	P0_9	TSG21CLKI	DNF_TSG21-G2
P1	P1_0	NMI	DNF_NMI-G0
	P1_4	TAUJ0I0	DNF_TAUJ0-G0
	P1_5	TAUJ0I1	DNF_TAUJ0-G1
	P1_6	TAUJ0I2	DNF_TAUJ0-G2
	P1_7	TAUA1I0	DNF_TAUA1-G0
		TAUJ0I3	DNF_TAUJ0-G3
		INTP0	DNF_INTP-G0
	P1_8	TAUA1I1	DNF_TAUA1-G0
	P1_9	TAUA1I2	DNF_TAUA1-G1

Table 2-54 List of Noise Filter Insertion Pins and Filter Groups (2/5)

Port Group Name	Port Pin	Filter Insertion Pin	Filter Group
P2	P2_0	TAUA0I0	DNF_TAUA0-G0
		TAUA0I1	DNF_TAUA0-G0
		CSIG0RYI	DNF_CSI-G0
	P2_1	TAUA0I2	DNF_TAUA0-G1
		TAUA0I3	DNF_TAUA0-G1
		CSIG1RYI	DNF_CSI-G0
	P2_2	TAUA0I4	DNF_TAUA0-G2
		TAUA0I5	DNF_TAUA0-G2
	P2_3	TAUA0I6	DNF_TAUA0-G3
		TAUA0I7	DNF_TAUA0-G3
	P2_4	TAUA0I8	DNF_TAUA0-G4
		TAUA0I9	DNF_TAUA0-G4
	P2_5	TAUA0I10	DNF_TAUA0-G5
		TAUA0I11	DNF_TAUA0-G5
	P2_6	TAUA0I12	DNF_TAUA0-G6
		TAUA0I13	DNF_TAUA0-G6
	P2_7	TAUA0I14	DNF_TAUA0-G7
		TAUA0I15	DNF_TAUA0-G7
P3	P3_0	TAUA1I0	DNF_TAUA1-G0
		TAUA1I1	DNF_TAUA1-G0
		INTP1	DNF_INTP-G0
	P3_1	TAUA1I2	DNF_TAUA1-G1
		TAUA1I3	DNF_TAUA1-G1
	P3_2	TAUA1I4	DNF_TAUA1-G2
		TAUA1I5	DNF_TAUA1-G2
	P3_3	TAUA1I6	DNF_TAUA1-G3
		TAUA1I7	DNF_TAUA1-G3
	P3_4	TAUA1I8	DNF_TAUA1-G4
		TAUA1I9	DNF_TAUA1-G4
	P3_5	TAUA1I10	DNF_TAUA1-G5
		TAUA1I11	DNF_TAUA1-G5
		TSG21PTSI0/ ENCA1E0	DNF_TSG21-G0
		INTP2	DNF_INTP-G0
	P3_6	TAUA1I12	DNF_TAUA1-G6
		TAUA1I13	DNF_TAUA1-G6
		TSG21PTSI1/ ENCA1E1	DNF_TSG21-G0
	P3_7	TAUA1I14	DNF_TAUA1-G7
		TAUA1I15	DNF_TAUA1-G7
		CSIG2RYI	DNF_CSI-G0

Table 2-54 List of Noise Filter Insertion Pins and Filter Groups (3/5)

Port Group Name	Port Pin	Filter Insertion Pin	Filter Group
P4	P4_0	INTP1	DNF_INTP-G0
		TSG21PTSI2/ ENCA1EC	DNF_TSG21-G1
	P4_1	TSG20PTSI0/ ENCA0E0	DNF_TSG20-G0
	P4_2	TSG20PTSI1/ ENCA0E1	DNF_TSG20-G0
	P4_3	TSG20PTSI2/ ENCA0EC	DNF_TSG20-G1
	P4_4	TAUJ1I0	DNF_TAUJ1-G0
		TAUJ0I0	DNF_TAUJ0-G0
		ESO0	ESO
		INTP6	DNF_INTP-G2
	P4_5	TAUJ1I1	DNF_TAUJ1-G1
		TAUJ0I1	DNF_TAUJ0-G1
		ESO1	ESO
		INTP7	DNF_INTP-G2
	P4_6	TAUA1I8	DNF_TAUA1-G4
		TAUJ1I2	DNF_TAUJ1-G2
		TAUJ0I2	DNF_TAUJ0-G2
		ESO2	ESO
		INTP8	DNF_INTP-G2
	P4_7	NMI	DNF_NMI-G0
		TAUJ1I3	DNF_TAUJ1-G3
		TAUJ0I3	DNF_TAUJ0-G3
		ESO3	ESO
		INTP9	DNF_INTP-G2

Table 2-54 List of Noise Filter Insertion Pins and Filter Groups (4/5)

Port Group Name	Port Pin	Filter Insertion Pin	Filter Group
P5	P5_0	INTP0	DNF_INTP-G0
	P5_1	INTP1	DNF_INTP-G0
		INTP3	DNF_INTP-G1
		CSIH0RYI	DNF_CSI-G0
		ADCA0TRG0	ADTRG
	P5_2	INTP2	DNF_INTP-G0
		INTP4	DNF_INTP-G1
		ADCA0TRG1	ADTRG
	P5_3	INTP3	DNF_INTP-G1
		INTP5	DNF_INTP-G1
		ADCA0TRG0	ADTRG
		ADCA0TRG2	ADTRG
	P5_4	CSIH1RYI	DNF_CSI-G0
	P5_5	TAUJ0I3	DNF_TAUJ0-G3
	P5_6	INTP1	DNF_INTP-G0
	P5_9	INTP4	DNF_INTP-G1
		CSIH0RYI	DNF_CSI-G0
		ADCA0TRG1	ADTRG
	P5_10	INTP5	DNF_INTP-G1
		ADCA0TRG2	ADTRG
P6	P6_3	INTP1	DNF_INTP-G0
	P6_6	INTP0	DNF_INTP-G0

Table 2-54 List of Noise Filter Insertion Pins and Filter Groups (5/5)

Port Group Name	Port Pin	Filter Insertion Pin	Filter Group
P7	P7_0	TAUA0I0	DNF_TAUA0-G0
		INTP3	DNF_INTP-G1
		TAUJ1I0	DNF_TAUJ1-G0
		ENCA0I0	DNF_ENC0-G0
		ADCA0TRG0	ADTRG
	P7_1	TAUA0I2	DNF_TAUA0-G1
		INTP10	DNF_INTP-G2
		TAUJ1I1	DNF_TAUJ1-G1
		ENCA0I1	DNF_ENC0-G0
	P7_2	TAUA0I4	DNF_TAUA0-G2
		ESO1	ESO
		INTP7	DNF_INTP-G2
		TAUJ1I2	DNF_TAUJ1-G2
		ENCA1I0	DNF_ENC1-G0
	P7_3	TAUA0I6	DNF_TAUA0-G3
		ESO0	ESO
		INTP6	DNF_INTP-G2
		TAUJ1I3	DNF_TAUJ1-G3
		ENCA1I1	DNF_ENC1-G0
	P7_4	TAUA0I8	DNF_TAUA0-G4
		INTP3	DNF_INTP-G1
		ADCA0TRG0	ADTRG
		TSG20PTSI0/ ENCA0E0	DNF_TSG20-G0
		TSG21PTSI0/ ENCA1E0	DNF_TSG21-G0
	P7_5	TAUA0I10	DNF_TAUA0-G5
		INTP4	DNF_INTP-G1
		ADCA0TRG1	ADTRG
		TSG20PTSI1/ ENCA0E1	DNF_TSG20-G0
		TSG21PTSI1/ ENCA1E1	DNF_TSG21-G0
	P7_6	TAUA0I12	DNF_TAUA0-G6
		INTP5	DNF_INTP-G1
		ADCA0TRG2	ADTRG
		TSG20PTSI2/ ENCA0EC	DNF_TSG20-G1
		TSG21PTSI2/ ENCA1EC	DNF_TSG21-G1
	P7_7	TAUA0I14	DNF_TAUA0-G7
		NMI	DNF_NMI-G0

(3) Noise Filter Insertion Pins and Filter Groups for Pins Other than Port Pins

Table 2-55 List of Filter Insertion Pins and Filter Groups for Pins Other than Port

Pin Name	Filter Insertion Pin	Filter Group
FLMD0	FLMD0	FLMD0
FLMD1	FLMD1	FLMD1
MODE0	MODE0	MODE0
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
$\overline{\text{DCUTRST}}$	$\overline{\text{DCUTRST}}$	$\overline{\text{DCUTRST}}$

(4) Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation

Table 2-56 List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (1/5)

Filter Group	Function Block	Target Pin	Type (Analog or Digital)	Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (Clock Selection Group)
$\overline{\text{RESET}}$	RESET	$\overline{\text{RESET}}$	Analog filter	Min. 100 ns, Max. 500 ns
$\overline{\text{DCUTRST}}$	Nexus	$\overline{\text{DCUTRST}}$		Min. 100 ns, Max. 500 ns
FLMD0, FLMD1	MODE	FLMD0, FLMD1		Min. 50 ns, Max. 250 ns
MODE0		MODE0		Min. 50 ns, Max. 250 ns
ESO	TAPA	ESO0		Min. 50 ns, Max. 250 ns
		ESO1		
		ESO2		
		ESO3		
ADTRG	ADCA0	ADCA0TRG0		
		ADCA0TRG1		
		ADCA0TRG2		

Table 2-56 List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (2/5)

Filter Group	Function Block	Target Pin	Type (Analog or Digital)	Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (Clock Selection Group)
DNF_NMI-G0	NMI	NMI	Digital filter	<INTP group> 1) PCLK 2) PCLK/2
DNF_INTP-G0	INTC	INTP0		
		INTP1		
		INTP2		
DNF_INTP-G1		INTP3		
		INTP4		
		INTP5		
DNF_INTP-G2		INTP6		
		INTP7		
		INTP8		
		INTP9		
		INTP10		

Table 2-56 List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (3/5)

Filter Group	Function Block	Target Pin	Type (Analog or Digital)	Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (Clock Selection Group)
DNF_TAUA0-G0	TAUA0	TAUA0I0	Digital filter	<TAUA0 group> 1) PCLK 2) PCLK/2 3) TAUA0 ch0 CKEN
		TAUA0I1		
DNF_TAUA0-G1		TAUA0I2		
		TAUA0I3		
DNF_TAUA0-G2		TAUA0I4		
		TAUA0I5		
DNF_TAUA0-G3		TAUA0I6		
		TAUA0I7		
DNF_TAUA0-G4		TAUA0I8		
		TAUA0I9		
DNF_TAUA0-G5		TAUAI10		
		TAUA0I11		
DNF_TAUA0-G6		TAUA0I12		
		TAUA0I13		
DNF_TAUA0-G7		TAUA0I14		
		TAUA0I15		

Table 2-56 List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (4/5)

Filter Group	Function Block	Target Pin	Type (Analog or Digital)	Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (Clock Selection Group)
DNF_TAUA1-G0	TAUA1	TAUA1I0	Digital filter	<TAUA1 group> 1) PCLK 2) PCLK/2 3) TAUA1 ch0 CKEN
		TAUA1I1		
DNF_TAUA1-G1		TAUA1I2		
		TAUA1I3		
DNF_TAUA1-G2		TAUA1I4		
		TAUA1I5		
DNF_TAUA1-G3		TAUA1I6		
		TAUA1I7		
DNF_TAUA1-G4		TAUA1I8		
		TAUA1I9		
DNF_TAUA1-G5		TAUA1I10		
		TAUA1I11		
DNF_TAUA1-G6		TAUA1I12		
		TAUA1I13		
DNF_TAUA1-G7		TAUA1I14		
		TAUA1I15		
DNF_TAUJ0-G0	TAUJ0	TAUJ0I0	Digital filter	<TAUJ0 group> 1) PCLK 2) PCLK/2 3) TAUJ0 ch0 CKEN
DNF_TAUJ0-G1		TAUJ0I1		
DNF_TAUJ0-G2		TAUJ0I2		
DNF_TAUJ0-G3		TAUJ0I3		
DNF_TAUJ1-G0	TAUJ1	TAUJ1I0		<TAUJ1 group> 1) PCLK 2) PCLK/2 3) TAUJ1 ch0 CKEN
DNF_TAUJ1-G1		TAUJ1I1		
DNF_TAUJ1-G2		TAUJ1I2		
DNF_TAUJ1-G3		TAUJ1I3		

Table 2-56 List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (5/5)

Filter Group	Function Block	Target Pin	Type (Analog or Digital)	Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation (Clock Selection Group)
DNF_TSG20-G0	TSG20	TSG20PTSI0/ ENCA0E0 TSG20PTSI1/ ENCA0E1	Digital filter	<TSG2,ENC group> 1) PCLK 2) PCLK/2
DNF_TSG20-G1		TSG20PTSI2/ ENCA0EC		
DNF_TSG20-G2		TSG20CLKI		
DNF_TSG21-G0	TSG21	TSG21PTSI0/ ENCA1E0 TSG21PTSI1/ ENCA1E1		<TSG2,ENC group> 1) PCLK 2) PCLK/2
DNF_TSG21-G1		TSG21PTSI2/ ENCA1EC		
DNF_TSG21-G2		TSG21CLKI		
DNF_ENC0-G0	ENC0	ENCA0I0 ENCA0I1		
DNC_ENC1-G0	ENC1	ENCA1I0 ENCA1I1		
DNF_CSI-G0	CSIG0	CSIG0RYI		<CSIgroup> 1) PCLK 2) PCLK/2
	CSIG1	CSIG1RYI		
	CSIG2	CSIG2RYI		
	CSIH0	CSIH0RYI		
	CSIH1	CSIH1RYI		

Register addresses

Addresses of control registers for controlling the digital noise filter are given as the respective offsets from the base address indicated by <DNFn_base>.

The register base address for the various registers, i.e. <DNFn_base>, is given below.

Table 2-57 Base Address <DNFn_base> of Control Registers for Controlling the Digital Noise Filter

<DNFn_base> Address
FF41 0000_H

2.16.2 Control Registers

The following registers control and run the digital noise filter.

Table 2-58 The List of Registers for the Digital Noise Canceller

Register Function	Name	Address
Digital noise canceller control register	DNFAnCTL	<DNFn_base> + n × 100_H
Digital noise canceller enable register	DNFAnEN	<DNFn_base> + n × 100_H + 4
Digital noise filter sampling clock control register	DNFSCTL	FF420034H

n = 0 to 36

Caution 1. Do not change the settings of associated control registers (the DNFAnCTL, DNFAnEN, DNFSCTL registers) while DNFAnEN.DNFAnNFENm = 1, which enables the digital filter. Changing the settings can lead to the filter generating unanticipated output.

Caution 2. Many registers include bits to which no function has been allocated. Unless there is a specific indication to the contrary, do not write values other than the initial values to such bits. Operation is not guaranteed if other values are set in these bits.

(1) DNFAnCTL – Digital Noise Canceller Control Register

This register is used to select the sampling clock for the digital noise canceller.

Access Readable and writable in 8-bit units.

Address Refer to table 2-58, The List of Registers for the Digital Noise Canceller.

Initial value 00_H

7	6	5	4	3	2	1	0
0	DNFAnNFSTS[1:0]		0	0	DNFAnPRS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-59 Contents of Register DNFAnCTL

Bit Position	Bit Name	Function
6, 5	DNFAnNFSTS [1:0]	Number of samples at the same level for judgment to validate or invalidate an external pulse 00: 2 01: 3 10: 4 11: 5
2 to 0	DNFAnPRS [2:0]	Digital filter sampling clock selection 000: DNFATCKI / 1 001: DNFATCKI / 2 010: DNFATCKI / 4 011: DNFATCKI / 8 100: DNFATCKI / 16 101: DNFATCKI / 32 110: DNFATCKI / 64 111: DNFATCKI / 128 DNFATCKI is the clock signal selected in the DNFSCTL register.

Register Name/ Address	Filter Group	Target Pin
DNFA0CTL FF410000	DNF_NMI-G0	NMI
DNFA1CTL FF410100	DNF_INTP-G0	INTP0, INTP1, INTP2
DNFA2CTL FF410200	DNF_INTP-G1	INTP3, INTP4, INTP5
DNFA3CTL FF410300	DNF_INTP-G2	INTP6, INTP7, INTP8, INTP9, INTP10
DNFA4CTL FF410400	DNF_TAUA0-G0	TAUA0I0, TAUA0I1
DNFA5CTL FF410500	DNF_TAUA0-G1	TAUA0I2, TAUA0I3
DNFA6CTL FF410600	DNF_TAUA0-G2	TAUA0I4, TAUA0I5
DNFA7CTL FF410700	DNF_TAUA0-G3	TAUA0I6, TAUA0I7
DNFA8CTL FF410800	DNF_TAUA0-G4	TAUA0I8, TAUA0I9
DNFA9CTL FF410900	DNF_TAUA0-G5	TAUA0I10, TAUA0I11
DNFA10CTL FF410A00	DNF_TAUA0-G6	TAUA0I12, TAUA0I13
DNFA11CTL FF410B00	DNF_TAUA0-G7	TAUA0I14, TAUA0I15
DNFA12CTL FF410C00	DNF_TAUA1-G0	TAUA1I0, TAUA1I1
DNFA13CTL FF410D00	DNF_TAUA1-G1	TAUA1I2, TAUA1I3
DNFA14CTL FF410E00	DNF_TAUA1-G2	TAUA1I4, TAUA1I5

Register Name/ Address	Filter Group	Target Pin
DNFA15CTL FF410F00	DNF_TAUA1-G3	TAUA1I6, TAUA1I7
DNFA16CTL FF411000	DNF_TAUA1-G4	TAUA1I8, TAUA1I9
DNFA17CTL FF411100	DNF_TAUA1-G5	TAUA1I10, TAUA1I11
DNFA18CTL FF411200	DNF_TAUA1-G6	TAUA1I12, TAUA1I13
DNFA19CTL FF411300	DNF_TAUA1-G7	TAUA1I14, TAUA1I15
DNFA20CTL FF411400	DNF_TAUJ0-G0	TAUJ0I0
DNFA21CTL FF411500	DNF_TAUJ0-G1	TAUJ0I1
DNFA22CTL FF411600	DNF_TAUJ0-G2	TAUJ0I2
DNFA23CTL FF411700	DNF_TAUJ0-G3	TAUJ0I3
DNFA24CTL FF411800	DNF_TAUJ1-G0	TAUJ1I0
DNFA25CTL FF411900	DNF_TAUJ1-G1	TAUJ1I1
DNFA26CTL FF411A00	DNF_TAUJ1-G2	TAUJ1I2
DNFA27CTL FF411B00	DNF_TAUJ1-G3	TAUJ1I3
DNFA28CTL FF411C00	DNF_TSG20-G0	TSG20PTSI0/ ENCA0E0 TSG20PTSI1/ ENCA0E1
DNFA29CTL FF411D00	DNF_TSG20-G1	TSG20PTSI2/ ENCA0EC
DNFA30CTL FF411E00	DNF_TSG20-G2	TSG20CLKI
DNFA31CTL FF411F00	DNF_TSG21-G0	TSG21PTSI0/ ENCA1E0 TSG21PTSI1/ ENCA1E1
DNFA32CTL FF412000	DNF_TSG21-G1	TSG21PTSI2/ ENCA1EC
DNFA33CTL FF412100	DNF_TSG21-G2	TSG21CLKI
DNFA34CTL FF412200	DNF_ENC0-G0	ENCA0I0 ENCA0I1
DNFA35CTL FF412300	DNF_ENC1-G0	ENCA1I0 ENCA1I1
DNFA36CTL FF412400	DNF_CSI-G0	CSIG0RYI CSIG1RYI CSIG2RYI CSIH0RYI CSIH1RYI

(2) DNFAnEN – Digital Noise Canceller Enable Register

This register specifies enabling and disabling of the digital noise cancellers.

Access Readable and writable in 16-bit units.

The higher- and lower-order bytes (DNFAnNFEN[15:8] and DNFAnNFEN[7:0]) are also accessible in 1- or 8-bit units.

Address Refer to table 2-58, The List of Registers for the Digital Noise Canceller.

Initial value 0000_H

15	14	13	12	11	10	9	8
DNFAnNFEN[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNFAnNFEN[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-60 DNFAnEN Register Contents

Bit Position	Bit Name	Function
15 to 0	DNFAnNFEN [15:0]	Each bit enables or disables digital noise cancellation for the signal on the corresponding pin. • DNFA0EN to DNFA3EN registers If the NMI or INTP0 to INTP10 is in use, set the corresponding bit to 1. - DNFA0EN.DNFA0NFEN[0] bit - DNFA1EN.DNFA1NFEN[2-0] bits - DNFA2EN.DNFA2NFEN[2-0] bits - DNFA3EN.DNFA3NFEN[4-0] bits • DNFA2EN register If the ADCA0TRG0, ADCA0TRG1, or ADCA0TRG2 multiplexed function is in use, set the corresponding bit to 0. - DNFA2EN.DNFA2NFEN[2-0] bits • DNFA3EN register If any of the ESO0 to ESO3 multiplexed functions is in use, set the corresponding bit to 0. - DNFA3EN.DNFA3NFEN[3-0] bits • DNFA4EN to DNFA36EN registers 0: Digital noise cancellation is disabled. 1: Digital noise cancellation is enabled.

Caution 1. When writing to DNFAnEN registers, be sure to write 0 to DNFAnEN.DNAFnENm bits for which “0” is the entry as shown in table 2-61, List of DNFAnEN Registers. Operation is not guaranteed if any such bit is set to 1.

Caution 2. When the output signal from the digital filter is set to an input for an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFAnEN.DNFAnNFENm = 1) for the port pin to switch to the alternative function.

DNFAnNFSTS[1:0] × 1/fs + 4 × $1/f_{DNFATCKI}$

Caution 3. When a digital filter is used with an interrupt acting as an event output signal, only enable the digital filter (by setting DNFAnEN.DNFAnNFENm = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.

DNFAnNFSTS[1:0] × 1/fs + 5 × $1/f_{DNFATCKI}$

Table 2-61 List of DNFAnEN Registers (1/2)

Register Name, Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DNFA0EN FF410004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA0NFEN 0(NMI)
DNFA1EN FF410104	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA1NFEN 2(INTP2)	DNFA1NFEN 1(INTP1)	DNFA1NFEN 0(INTP0)
DNFA2EN FF410204	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA2NFEN 2(INTP5/ ADCA0TRG2)	DNFA2NFEN 1(INTP4/ ADCA0TRG1)	DNFA2NFEN 0(INTP3/ ADCA0TRG0)
DNFA3EN FF410304	0	0	0	0	0	0	0	0	0	0	0	DNFA3NFEN 4(INTP10)	DNFA3NFEN 3(INTP9/ ESO3)	DNFA3NFEN 2(INTP8/ ESO2)	DNFA3NFEN 1(INTP7/ ESO1)	DNFA3NFEN 0(INTP6/ ESO0)
DNFA4EN FF410404	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA4NFEN 1(TAUA0I1)	DNFA4NFEN 0(TAUA0I0)
DNFA5EN FF410504	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA5NFEN 1(TAUA0I3)	DNFA5NFEN 0(TAUA0I2)
DNFA6EN FF410604	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA6NFEN 1(TAUA0I5)	DNFA6NFEN 0(TAUA0I4)
DNFA7EN FF410704	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA7NFEN 1(TAUA0I7)	DNFA7NFEN 0(TAUA0I6)
DNFA8EN FF410804	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA8NFEN 1(TAUA0I9)	DNFA8NFEN 0(TAUA0I8)
DNFA9EN FF410904	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA9NFEN 1(TAUA0I11)	DNFA9NFEN 0(TAUA0I10)
DNFA10EN FF410A04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA10NFEN 1(TAUA0I13)	DNFA10NFEN 0(TAUA0I12)
DNFA11EN FF410B04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA11NFEN 1(TAUA0I15)	DNFA11NFEN 0(TAUA0I14)
DNFA12EN FF410C04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA12NFE N1(TAUA1I1)	DNFA12NFE N0(TAUA1I0)
DNFA13EN FF410D04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA13NFE N1(TAUA1I3)	DNFA13NFE N0(TAUA1I2)
DNFA14EN FF410E04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA14NFE N1(TAUA1I5)	DNFA14NFE N0(TAUA1I4)
DNFA15EN FF410F04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA15NFE N1(TAUA1I7)	DNFA15NFE N0(TAUA1I6)
DNFA16EN FF411004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA16NFE N1(TAUA1I9)	DNFA16NFE N0(TAUA1I8)
DNFA17EN FF411104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA17NFEN 1(TAUA1I11)	DNFA17NFEN 0(TAUA1I10)
DNFA18EN FF411204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA18NFEN 1(TAUA1I13)	DNFA18NFEN 0(TAUA1I12)
DNFA19EN FF411304	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA19NFEN 1(TAUA1I15)	DNFA19NFEN 0(TAUA1I14)

Table 2-61 List of DNFAnEN Registers (2/2)

Register Name, Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DNFA20EN FF411404	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA20NFEN0(TAUJ0I0)
DNFA21EN FF411504	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA21NFEN0(TAUJ0I1)
DNFA22EN FF411604	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA22NFEN0(TAUJ0I2)
DNFA23EN FF411704	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA23NFEN0(TAUJ0I3)
DNFA24EN FF411804	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA24NFEN0(TAUJ1I0)
DNFA25EN FF411904	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA25NFEN0(TAUJ1I1)
DNFA26EN FF411A04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA26NFEN0(TAUJ1I2)
DNFA27EN FF411B04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA27NFEN0(TAUJ1I3)
DNFA28EN FF411C04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA28NFEN1(TSG20PTSI1/ ENCA0E1)	DNFA28NFEN0(TSG20PTSI0/ ENCA0E0)
DNFA29EN FF411D04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA29NFEN0(TSG20PTSI2/ ENCA0EC)
DNFA30EN FF411E04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA30NFEN0(TSG20CLKI)
DNFA31EN FF411F04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA31NFEN1(TSG21PTSI1/ ENCA1E1)	DNFA31NFEN0(TSG21PTSI0/ ENCA1E0)
DNFA32EN FF412004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA32NFEN0(TSG21PTSI2/ ENCA1EC)
DNFA33EN FF412104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA33NFEN0(TSG21CLKI)
DNFA34EN FF412204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA34NFEN1(ENCA0I1)	DNFA34NFEN0(ENCA0I0)
DNFA35EN FF412304	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DNFA35NFEN1(ENCA1I1)	DNFA35NFEN0(ENCA1I0)
DNFA36EN FF412404	0	0	0	0	0	0	0	0	0	0	0	DNFA36NFEN4 (CSIG2RYI)	DNFA36NFEN3 (CSIG1RYI)	DNFA36NFEN2 (CSIG0RYI)	DNFA36NFEN1 (CSIH1RYI)	DNFA36NFEN0 (CSIH0RYI)

(3) Digital Noise Filter Sampling Clock Control Register

This register selects the digital noise filter sampling clock.

Access Readable and writable in 16-bit units.

Address Refer to table 2-58, The List of Registers for the Digital Noise Canceller.

Initial value 0000_H

15	14	13	12	11	10	9	8
DNFSCKSL7[1:0]		DNFSCKSL6[1:0]		DNFSCKSL5[1:0]		DNFSCKSL4[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
0	0	0	DNFSCKSL20	0	DNFSCKSL10	0	DNFSCKSL00
R	R	R	R/W	R	R/W	R	R/W

Table 2-62 DNFSCTL Register Contents

Bit Position	Bit Name	Function
15, 14	DNFSCKSL 71, 70	Selects the sampling clock of the TAUJ1 group. 00: PCLK 01: PCLK/2 10: TAUJ1 ch0 CKEN signal 11: Setting prohibited
13, 12	DNFSCKSL 61, 60	Selects the sampling clock of the TAUJ0 group. 00: PCLK 01: PCLK/2 10: TAUJ0 ch0 CKEN signal 11: Setting prohibited
11, 10	DNFSCKSL 51, 50	Selects the sampling clock of the TAUA1 group. 00: PCLK 01: PCLK/2 10: TAUA1 ch0 CKEN signal 11: Setting prohibited
9, 8	DNFSCKSL 41, 40	Selects the sampling clock of the TAUA0 group. 00: PCLK 01: PCLK/2 10: TAUA0 ch0 CKEN signal 11: Setting prohibited
4	DNFSCKSL 20	Selects the sampling clock of the CSI group. 0: PCLK 1: PCLK/2
2	DNFSCKSL 10	Selects the sampling clock of the TSG2 and ENC groups. 0: PCLK 1: PCLK/2
0	DNFSCKSL 00	Selects the sampling clock of the INTP group. 0: PCLK 1: PCLK/2

Caution Only set this register while the operation of the corresponding DNF is not enabled. Operation cannot be guaranteed if these settings are made while the operation of a corresponding DNF is enabled.

2.17 Edge Detection

This section describes the pins and control registers used with edge detection.

2.17.1 Details of Edge Detection

(1) Pins for Use in Edge Detection

Table 2-63 List of Pins for Use in Edge Detection

Target Pin	Control Register Name
NMI	FCLA0CTL0
INTP0	FCLA1CTL0
INTP1	FCLA1CTL1
INTP2	FCLA1CTL2
INTP3	FCLA1CTL3
INTP4	FCLA1CTL4
INTP5	FCLA1CTL5
INTP6	FCLA1CTL6
INTP7	FCLA1CTL7
INTP8	FCLA2CTL0
INTP9	FCLA2CTL1
INTP10	FCLA2CTL2

2.17.2 Control Register

(1) FCLAnCTLm-Filter Control Register

This register specifies operation of the edge detection function.

Access Readable and writable in 8-bit units.

Address Refer to table 2-65, List of FCLAnCTLm Registers.

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCLAn INTFm	FCLAn INTRm
R	R	R	R	R	R	R/W	R/W

Table 2-64 FCLAnCTLm Register Contents

Bit Position	Bit Name	Function
1	FCLAnINTFm	Controls falling edge detection of the input signal 0: Falling edge detection disabled 1: Falling edge detection enabled
0	FCLAnINTRm	Controls rising edge detection of the input signal 0: Rising edge detection disabled 1: Rising edge detection enabled

Note If FCLAnINTFm = 0 and FCLAnINTRm = 0, no edge detection is done.
If FCLAnINTFm = 1 and FCLAnINTRm = 1, both edges are recognized as valid.

Table 2-65 List of FCLAnCTLm Registers

FCLAn	Register Name	Register Address	Generated Interrupt
FCLA0	FCLA0CTL0	FF414000	NMI
FCLA1	FCLA1CTL0	FF414100	INTP0
	FCLA1CTL1	FF414104	INTP1
	FCLA1CTL2	FF414108	INTP2
	FCLA1CTL3	FF41410C	INTP3
	FCLA1CTL4	FF414110	INTP4
	FCLA1CTL5	FF414114	INTP5
	FCLA1CTL6	FF414118	INTP6
	FCLA1CTL7	FF41411C	INTP7
FCLA2	FCLA2CTL0	FF414200	INTP8
	FCLA2CTL1	FF414204	INTP9
	FCLA2CTL2	FF414208	INTP10

Caution Many registers include bits to which no function has been allocated. Unless there is a specific indication to the contrary, do not write values other than the initial values to such bits. Operation is not guaranteed if other values are set in these bits.

Section 3 CPU System Function

This section describes the registers of the CPU, the operation modes, the address space and the memory areas.

For details on the functions of the CPU, refer to manuals as indicated below.

Table 3-1 Descriptions of CPU Functions

Function	V850E2M Architecture (R01US0001E)	This Manual
V850E2 CPU (including instruction set)	√	–
Floating-point calculation unit (FPU)	√	–
Processor protection functions (MPU, SRP, PPU, TSU)	√	–
DMA/DTS controller (DMAC)	–	√
Interrupt controller (INTC)	–	√
Code flash/data flash memory	–	√
On-chip RAM	–	√

3.1 Overview

CPU

- Core architecture: V850E2M architecture class
- Instruction execution times:

Device	Minimum Instruction Execution Time	Maximum CPU Clock
µPD70F3505A	12.5 ns	80 MHz
µPD70F3506	12.5 ns	80 MHz
µPD70F3507	12.5 ns	80 MHz
µPD70F3508	6.25 ns	160 MHz
µPD70F3509	6.25 ns	160 MHz

- Internal 32-bit architecture
- 7-stage pipeline
- 32 × 32-bit general purpose registers
- Floating point calculation unit (FPU) (except for the µPD70F3505A)
- 2-way superscalar
- Memory space:
  - 512-Mbyte linear program space
  - 4-Gbyte linear data space

- Flash and cache:
  - 8 Kbytes, 2-way associative (4 Kbytes per way)
- Interrupt controller (INTC)
- DMA/DTS controller (DMAC)
- Processor protection functions
  - Memory protection unit (MPU)
    Protection against illegal execution from or data manipulation of CPU memory areas (up to 5 areas in instruction space, up to 6 areas in data space)
  - System register protection (SRP)
    Protection against damage to the system registers by a non-trusted programs
  - Peripheral protection unit (PPU)
    Protection against illegal access to peripheral modules
  - Timing supervision unit (TSU)
    Protection against inappropriate CPU time possession by a non-trusted program, and resources and time of disabling interrupts can be managed

Caution The memory protection unit (MPU) for this product may not be able to detect unauthorized access in the form of misaligned access at the upper boundary when the setting for data protection area 5 is other than the setting for base address 0. Make sure that such misaligned access does not proceed.

For details, refer to section 3.1.2, Important Reminders.

Instruction set

- V850E2 instruction set compatible to former V850 instruction sets plus additional powerful instructions for reduced code size and increasing execution speed
- Signed multiplication operations in 1 clock
  - 16 bits × 16 bits → 32 bits
  - 32 bits × 32 bits → 32 bits or 64 bits
  - 32 bits × 32 bits → 64 bits
- Saturated operation instructions with overflow/underflow detection
- 32-bit shift instructions in 1 clock
- Bit manipulation instructions (SET1, CLR1, NOT1, TST1)
- Load/store instructions with long/short format
- Signed load instructions
- MAC operation
  32 bits × 32 bits + 64 bits → 64 bits
- Floating-point operations
  The FPU of V850E/Px4 complies with ANSI/IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic.

3.1.1 Peripheral Protection Unit

PPU base address

The addresses of the peripheral protection unit registers described in the "V850E2M Architecture Manual (R01US0001E)" are given as offset addresses. The base address is as follows:

<PPU_base> = FFFF 5100_H

PPU areas and registers

The control registers for each protected area comprise 4 registers.

- PPVn – Validating general peripheral device protection
- PPTn – Specification protection type of general peripheral device
- PPPn – Specification of OS peripheral device
- PPSn – Specification of special peripheral device

These registers are numbered with n = 0 to 8.

The bits in these 32-bit wide registers are named with m = 0 to 31.

- PPVn.PPVnm
- PPTn.PPTnm
- PPPn.PPPnm
- PPSn.PPSnm

The protected address ranges, their control registers and bits and the modules in the respective address are listed in table 3-2, PPU Protected Areas and Modules.

Table 3-2 PPU Protected Areas and Modules (1/4)

Protection Range Size	Protection Control		Module Name	Address Range
	Registers PPVn, PPTn, PPPn, PPSn n =	Bits PPVnm, PPTnm, PPPnm, PPSnm m =		
1120 B	0	0	INTC	FFFF 6000_H to FFFF $645F_H$
4 B		3	System error	FFFF $64B0_H$ to FFFF $64B3_H$
1024B		16	TSU/PPU	FFFF 5000_H to FFFF $53FF_H$
256 B		23	DMAC	FFFF 7300_H to FFFF $73FF_H$
256 B		24		FFFF 7400_H to FFFF $74FF_H$
512 B		28		FFFF $7B00_H$ to FFFF $7CFF_H$
512 B		29		FFFF $7D00_H$ to FFFF $7EFF_H$
64 KB	1	0	Port Pnm control	FF40 0000_H to FF40 $FFFF_H$
		1	Port filters control	FF41 0000_H to FF41 $FFFF_H$
		2	Clock generator Reset controller	FF42 0000_H to FF42 $FFFF_H$
		3	Code flash ECC, data flash access wait, FLMD control	FF43 0000_H to FF43 $FFFF_H$
		4	Port JPnm control	FF44 0000_H to FF44 $FFFF_H$

Table 3-2 PPU Protected Areas and Modules (2/4)

Protection Range Size	Protection Control		Module Name	Address Range
	Registers PPVn, PPTn, PPPn, PPSn n =	Bits PPVnm, PPTnm, PPPnm, PPSnm m =		
64 KB	1	5	POF/LVI control PBUS error/access control	FF45 0000$_H$ to FF45 FFFF$_H$
		6	FCN, FlexRay*, on-chip RAM ECC	FF46 0000$_H$ to FF46 FFFF$_H$
		7	Option byte, product information	FF47 0000$_H$ to FF47 FFFF$_H$
		8	FCN0	FF48 0000$_H$ to FF48 FFFF$_H$
		9		FF49 0000$_H$ to FF49 FFFF$_H$
		10	FCN1	FF4A 0000$_H$ to FF4A FFFF$_H$
		11		FF4B 0000$_H$ to FF4B FFFF$_H$
		24	FlexRay*	FF58 0000$_H$ to FF58 FFFF$_H$
		28	UARTH0	FF5C 0000$_H$ to FF5C FFFF$_H$
		29	UARTH1	FF5D 0000$_H$ to FF5D FFFF$_H$
		30	UARTH2	FF5E 0000$_H$ to FF5E FFFF$_H$
64 KB	2	12	CSIH0	FF6C 0000$_H$ to FF6C FFFF$_H$
		13	CSIH1*	FF6D 0000$_H$ to FF6D FFFF$_H$
		16	CSIG0	FF70 0000$_H$ to FF70 FFFF$_H$
		17	CSIG1	FF71 0000$_H$ to FF71 FFFF$_H$
		18	CSIG2	FF72 0000$_H$ to FF72 FFFF$_H$
4 KB	3	0	OSTM0	FF80 0000$_H$ to FF80 0FFF$_H$
		1	OSTM1	FF80 1000$_H$ to FF80 1FFF$_H$
		2	CLMA0	FF80 2000$_H$ to FF80 2FFF$_H$
		3	CLMA1	FF80 3000$_H$ to FF80 3FFF$_H$
		4	CLMA2	FF80 4000$_H$ to FF80 4FFF$_H$
		6	WDTA0	FF80 6000$_H$ to FF80 6FFF$_H$
		8	TAUA0	FF80 8000$_H$ to FF80 8FFF$_H$
		9	TAUA1	FF80 9000$_H$ to FF80 9FFF$_H$
		17	TAUJ0	FF81 1000$_H$ to FF81 1FFF$_H$
		18	TAUJ1	FF81 2000$_H$ to FF81 2FFF$_H$
		21	TAPA0	FF81 5000$_H$ to FF81 5FFF$_H$
		22	TAPA1	FF81 6000$_H$ to FF81 6FFF$_H$
		23	TAPA2	FF81 7000$_H$ to FF81 7FFF$_H$
		24	TAPA3	FF81 8000$_H$ to FF81 8FFF$_H$
		25	ENCA0	FF81 9000$_H$ to FF81 9FFF$_H$
		26	ENCA1	FF81 A000$_H$ to FF81 AFFF$_H$
		28	PIC function	FF81 C000$_H$ to FF81 CFFF$_H$
		29	ADCA0	FF81 D000$_H$ to FF81 DFFF$_H$
		31	DTS control	FF81 F000$_H$ to FF81 FFFF$_H$

Table 3-2 PPU Protected Areas and Modules (3/4)

Protection Range Size	Protection Control		Module Name	Address Range
	Registers PPVn, PPTn, PPPn, PPSn n =	Bits PPVnm, PPTnm, PPPnm, PPSnm m =		
4 KB	4	14	TSG20	FF82 $E000_H$ to FF82 $EFFF_H$
		15	TSG21	FF82 $F000_H$ to FF82 $FFFF_H$
		18	TPBA0	FF83 2000_H to FF83 $2FFF_H$
		19	TPBA1	FF83 3000_H to FF83 $3FFF_H$
		22	DCRA0	FF83 6000_H to FF83 $6FFF_H$
		23	DCRA1	FF83 7000_H to FF83 $7FFF_H$
		25	BRGA0	FF83 9000_H to FF83 $9FFF_H$
		26	INTC	FF83 $A000_H$ to FF83 $AFFF_H$
		27	Self-diagnosis BIST	FF83 $B000_H$ to FF83 $BFFF_H$
		28	SGA	FF83 $C000_H$ to FF83 $CFFF_H$
		29		FF83 $D000_H$ to FF83 $DFFF_H$
		30		FF83 $E000_H$ to FF83 $EFFF_H$
		31	OSTM input clock select, SGA error pulse output control, Back-up RAM register	FF83 $F000_H$ to FF83 $FFFF_H$
256 B	5	0	Port	FFFF 8000_H to FFFF $80FF_H$
		1		FFFF 8100_H to FFFF $81FF_H$
		2		FFFF 8200_H to FFFF $82FF_H$
		3		FFFF 8300_H to FFFF $83FF_H$
		4		FFFF 8400_H to FFFF $84FF_H$
		5		FFFF 8500_H to FFFF $85FF_H$
		6		FFFF 8600_H to FFFF $86FF_H$
		7		FFFF 8700_H to FFFF $87FF_H$
		8		FFFF 8800_H to FFFF $88FF_H$
		9		FFFF 8900_H to FFFF $89FF_H$

Table 3-2 PPU Protected Areas and Modules (4/4)

Protection Range Size	Protection Control		Module Name	Address Range
	Registers PPVn, PPTn, PPPn, PPSn n =	Bits PPVnm, PPTnm, PPPnm, PPSnm m =		
256 B	7	0	OSTM0	FFFF $C000_H$ to FFFF $C0FF_H$
		1	OSTM1	FFFF $C100_H$ to FFFF $C1FF_H$
		2	TAUJ0	FFFF $C200_H$ to FFFF $C2FF_H$
		3	TAUJ1	FFFF $C300_H$ to FFFF $C3FF_H$
		4	TAUA0	FFFF $C400_H$ to FFFF $C4FF_H$
		5		FFFF $C500_H$ to FFFF $C5FF_H$
		8	TAUA1	FFFF $C800_H$ to FFFF $C8FF_H$
		9		FFFF $C900_H$ to FFFF $C9FF_H$
		12	TSG20	FFFF $CC00_H$ to FFFF $CCFF_H$
		16	TSG21	FFFF $D000_H$ to FFFF $D0FF_H$
		20	TAPA0	FFFF $D400_H$ to FFFF $D4FF_H$
		21	TAPA1	FFFF $D500_H$ to FFFF $D5FF_H$
		22	TAPA2	FFFF $D600_H$ to FFFF $D6FF_H$
		23	TAPA3	FFFF $D700_H$ to FFFF $D7FF_H$
		24	ENCA0	FFFF $D800_H$ to FFFF $D8FF_H$
		25	ENCA1	FFFF $D900_H$ to FFFF $D9FF_H$
		27	PIC function	FFFF $DB00_H$ to FFFF $DBFF_H$
		28	ADCA0	FFFF $DC00_H$ to FFFF $DCFF_H$
256 B	8	0	CSIH0	FFFF $E000_H$ to FFFF $E0FF_H$
		1	CSIH1*	FFFF $E100_H$ to FFFF $E1FF_H$
		4	CSIG0	FFFF $E400_H$ to FFFF $E4FF_H$
		5	CSIG1	FFFF $E500_H$ to FFFF $E5FF_H$
		6	CSIG2	FFFF $E600_H$ to FFFF $E6FF_H$
		10	UARTH0	FFFF $EA00_H$ to FFFF $EAFF_H$
		11	UARTH1	FFFF $EB00_H$ to FFFF $EBFF_H$
		12	UARTH2	FFFF $EC00_H$ to FFFF $ECFF_H$
		16	TPBA0	FFFF $F000_H$ to FFFF $F0FF_H$
		17		FFFF $F100_H$ to FFFF $F1FF_H$
		20	TPBA1	FFFF $F400_H$ to FFFF $F4FF_H$
		21		FFFF $F500_H$ to FFFF $F5FF_H$
		26	DCRA0	FFFF $FA00_H$ to FFFF $FAFF_H$
		27	DCRA1	FFFF $FB00_H$ to FFFF $FBFF_H$
		28	POF/LVI control	FFFF $FC00_H$ to FFFF $FCFF_H$
		29	BRGA0	FFFF $FD00_H$ to FFFF $FDFF_H$
		30	LIN communications slave operation baud-rate detection	FFFF $FE00_H$ to FFFF $FEFF_H$

Note * The μPD70F3505A does not have this module.

3.1.2 Important Reminders

Please be aware of the following points regarding functions as described in the V850E2M Architecture Manual (R01US0001E).

(1) Data Protection

When using data protection by the DPA5U and DPA5L registers, blocking of access and generation of an exception will not always occur in response to access that violates access protection by crossing the upper boundary for data protection (misaligned access). Therefore, use one of the following settings with data protection by the DPA5U and DPA5L registers.

- Set the protection area such that there are no data to which access protection will apply in the protection area and in the 3 bytes from the upper boundary of the protection area.
- Set the address of the lower boundary of the data protection area to 00000000_H.
- Avoid misaligned access to the range within 3 bytes from the upper boundary of the data protection area.

(2) Branching Instructions and Loading Instructions

When a branching instruction and a loading instruction are consecutive in memory, parallel execution of the pipelines means that the loading instruction which immediately follows the branching instruction is executed (but discarded by the CPU) even if the preceding instruction leads to a branch.

An ECC error occurs if the loading instruction that immediately follows the branching instruction is in an area of on-chip RAM that is not mounted or not initialized (this does not apply when the loading instruction is in the flash area).

Applicable branching instructions are Bcond, JARL, JMP, and JR

Applicable loading instructions are LD and SLD

Use any one of the following methods to avoid the above phenomenon.

- Ensure that there is no access to a loading instruction that immediately follows a branching instruction and is in an area of on-chip RAM which is not mounted.
  Initialize the on-chip RAM area before using it.
- Insert an NOP instruction between each branching instruction and the loading instruction following it (write this in assembler because it may be eliminated by optimization if the C language is used).
- In the ECC error detection interrupt processing program, treat the error as invalid if the address where the interrupt occurred was in an area of on-chip RAM that has not been mounted.

3.1.3 Timing Supervision Unit

TSU base address

The addresses of the timing supervision unit registers described in the “V850E2M Architecture Manual (R01US0001E)” are given as offset addresses. The base address is as follows:

<TSU_base> = $FFFF\ 5000_H$

3.2 Operation Modes

This section describes the operation modes of this product and how the modes are selected.

The following operation modes are available:

- Normal operation mode
- Flash programming mode

After reset release by $\overline{\text{RESET}}$ pin, the microcontroller starts to fetch instructions from an internal boot ROM which contains the internal firmware. The firmware checks the FLMD0, the FLMD1, and MODE0 pins, to set the operation mode after reset release according to the table below.

Note. When the FLMD0 pin is in use, connect a resistor with a value of at least 82 kΩ to pull it down to the EVSS level.

Table 3-3 Selecting an Operation Mode

Pin			Operation Mode
FLMD0	**FLMD1**	**MODE0 (P8_0)**	
0	0	P8_0	Operation mode (single-chip mode)
1	0	P8_0	Flash programming mode
Other than the above			Settings are prohibited

3.2.1 Normal Operation Mode

This mode enables to access the on-chip ROM.

In single-chip mode, instruction processing starts with a branch to the reset entry address in the on-chip ROM after release from the system-reset state.

3.2.2 Flash Programming Mode

This mode allows the programming of on-chip flash memory by the flash writer. Flash memory is programmable through Nexus communications.

For details, see Section 6, Memory Modules.

3.2.3 HALT Mode

(1) Setting and Operation of HALT Mode

In normal operation mode, HALT mode is selected by executing a dedicated instruction (the HALT instruction). In HALT mode, pipeline operation of the CPU is stopped but clock signals continue to be supplied to the clock oscillation circuit and other on-chip peripheral modules.

This allows the stopping of program execution while retaining the on-chip RAM in the same condition as before the HALT mode setting. Furthermore, on-chip peripheral modules that are independent of instruction processing by the CPU continue to operate. The average current drawn by the overall system can be reduced by alternation between the HALT and normal operating modes.

Table 3-4 State of Operations in HALT Mode

Function	State of Operations
Clock generator	Operating
CPU system clock	Operating
Peripheral clock (PCLK)	Operating
CPU	Stopped
Peripheral I/O	Operating
Internal data	The states of internal data for which reprogramming is not possible while the CPU is stopped, such as the values of registers, status bits, and other data in the CPU and on-chip RAM, are retained from before the setting of HALT mode.

(2) Cancelling HALT Mode

The transition from HALT mode to normal operating mode is in response to the input of a reset, or to the generation of an interrupt or other exception. Following the acceptance of any interrupt or exception, program counting by the program counter resumes from the address of the instruction after the HALT instruction.

Input of any of the following resets or the generation of any of the following interrupts or exceptions constitutes a request for release from HALT mode.

- Reset input ($\overline{\text{RESET}}$, $\overline{\text{WDTA0RES}}$, $\overline{\text{CLMA0RES}}$ to $\overline{\text{CLMA2RES}}$, $\overline{\text{LVIRES}}$, $\overline{\text{SGARES}}$, $\overline{\text{DBRES}}$)
- EI level maskable interrupt (EIINT)
- FE level maskable interrupt (FEINT)
- FE level non-maskable interrupt (FENMI)
- Peripheral device protection exception (PPI)
- Timing supervision exception (TSI)
- System error exception (SYSERR)

3.3 Address Space

This section describes the address space of the CPU (sizes and addresses of the CPU address space and physical address space).

The address range of data space and program space together with their wraparound properties are presented.

3.3.1 CPU Data Address and Physical Program Address Space

The CPU supports the following address spaces.

- 4-GB CPU data address space
  With the 32-bit general purpose registers, addresses for a 4-Gbyte memory can be generated. This is the maximum address space supported by the CPU.
- 512-Mbyte physical program address space
  The CPU provides 512-Mbyte physical address space to access instruction codes in program memory. That means that a maximum of 512-Mbyte internal or external program memory can be accessed.

3.3.2 Program and Data Space

The figure below shows the assignment of the CPU address space to data and program space.

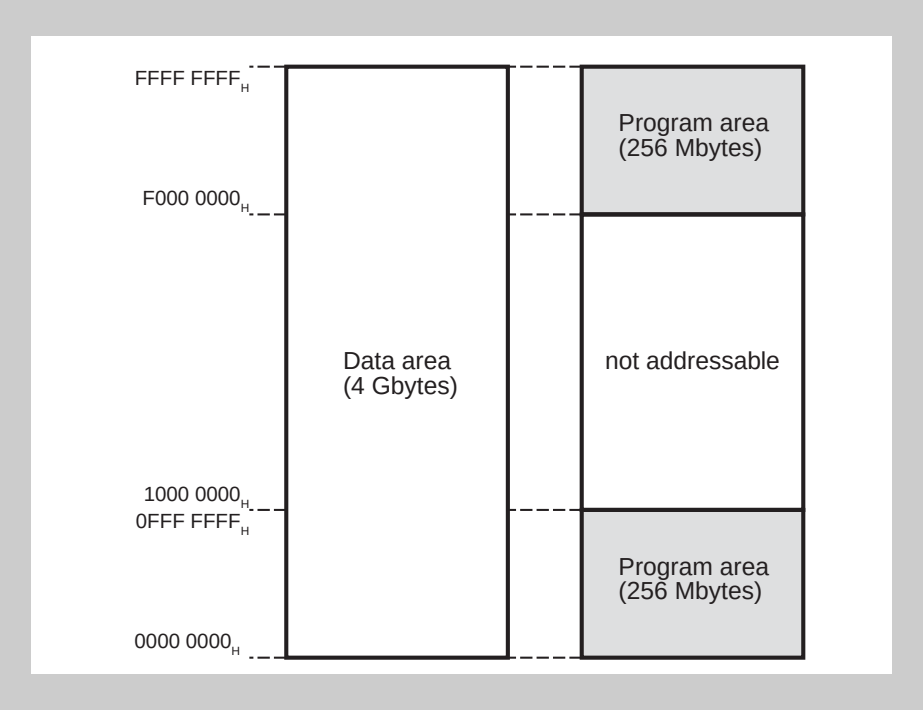


Figure 3-1 CPU Address Space

(1) Wrap-Around of Data Space

If an operand address calculation exceeds 32 bits, only the lower 32 bits of the result are considered. Therefore, the addresses FFFF FFFF$_H$ and 0000 0000$_H$ are contiguous addresses. This results in a wrap-around of the data space:

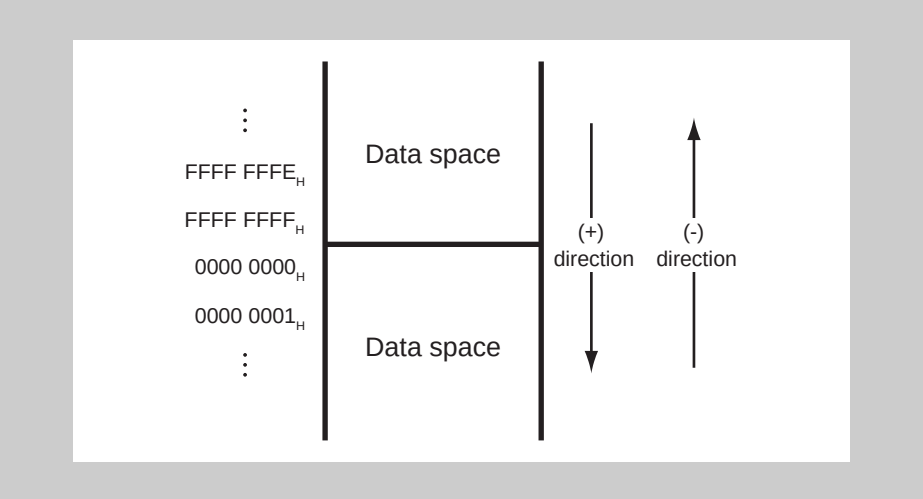


Figure 3-2 Wrap-Around of Data Space

(2) Wrap-Around of Program Space

If an instruction address calculation exceeds 28 bits, only the lower 28 bits of the result are considered. Therefore, the addresses 0000 0000$_H$ and 0FFF FFFF$_H$ are contiguous addresses. This results in a wrap-around of the program space:

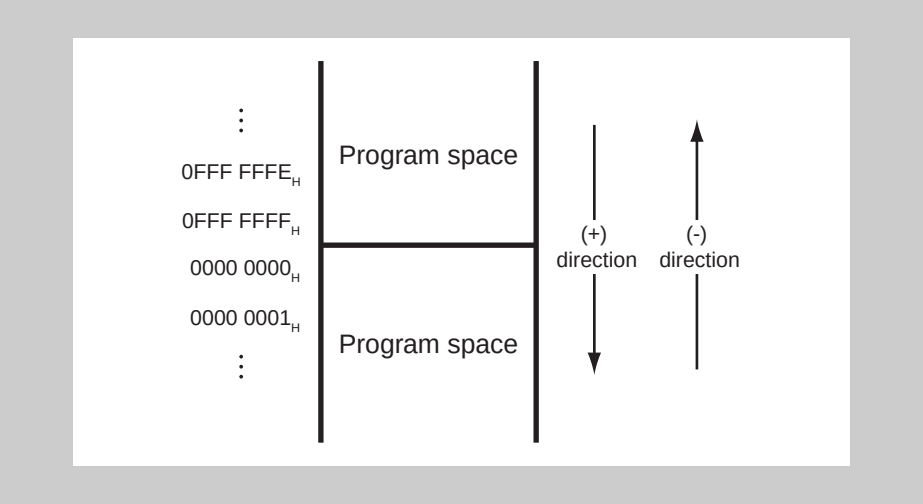


Figure 3-3 Wrap-Around of Program Space

3.4 Conditions for Boundary Operations

3.4.1 Program Space

Branching to the peripheral I/O area or consecutive fetching from the RAM area to peripheral I/O area should not proceed. Such branching or consecutive fetching can lead to the fetching of undefined data.

Caution Executing an executable instruction allocated to a location within 48 bytes of the last address of a memory area (code flash or on-chip RAM) may lead to an ECC error (undefined instructions are not executed).

3.4.2 Data Space

This product is capable of handling misaligned data.

This allows the allocation of data to any address regardless of the unit of data (word or half-word). However, for a word or half-word of data, at least 2 bus cycles are generated unless the data are aligned with the corresponding boundary, and this decreases bus efficiency.

(1) Access to Data with Half-Word Length

If the lowest-order address bit is 1, 2 cycles of byte-length bus access are generated.

(2) Access to Data with Word Length

- If the lowest-order address bit is 1, 3 bus cycles are generated in the following order: byte length, half-word length, and byte length.
- If the 2 lower-order address bits are 10, 2 bus cycles of half-word access are generated.

3.5 Memory Mapping

This section describes memory maps for the CPU and the address map for DMA.

3.5.1 Memory Map for DMA Access

For details, refer to section 5.3.6, Memory Map for DMA/DTS Access.

3.5.2 Memory Map

(1) V850E2/Px4 Memory Map

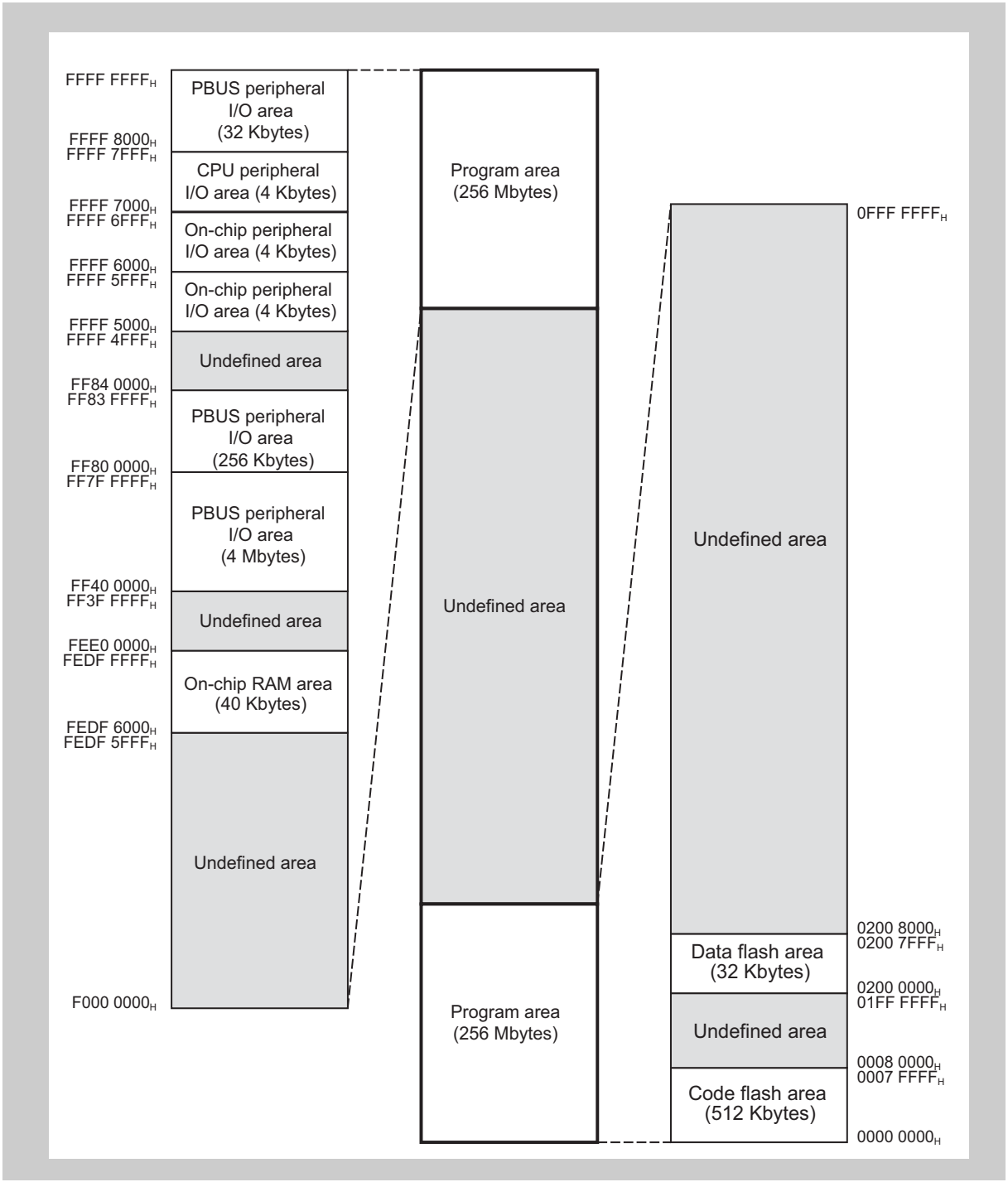


Figure 3-4 V850E2/Px4 Memory Map (for the μPD70F3507)

3.6 CPU-Related Registers

This section describes CPU related registers.

3.6.1 Overview of CPU-Related Registers

Table 3-5 List of CPU-Related Registers

Register Name	Symbol	Address
Processor element identifier register	PEID	FFFF 6490$_H$
Data flash access wait setting register	DCLKWAIT	FF43 6000$_H$

3.6.2 CPU-Related Registers in Detail

(1) PEID - Processor Element Identifier Register

The PEID register returns the processor-element identifier of the last accessing CPU.

Access Readable in 16 bit-units

Address FFFF 6490$_H$

Initial value 0001$_H$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEID[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3-6 PEID Register

Bit Position	Bit Name	Function
15 to 0	PEID[15:0]	These bits indicate a processor element ID and are always read as 0001$_H$.

(2) DCLKWAIT - Data Flash Access Wait Setting Register

This register is used to set a waiting time for access to data-flash memory.

Access Readable/writable in 8-bit units.

Address FF43 6000$_H$

Initial value Initialized to the value 17$_H$ by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	WAIT[4:0]				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3-7 DCLKWAIT Register

Bit Position	Bit Name	Function
4 to 0	WAIT4 - WAIT0	These bits are used to set a waiting time for access to data-flash memory. Specify the setting that corresponds to the operating frequency of the CPU.

Setting Value	Description
08$_H$	Setting when the CPU is operating at 48 MHz
0A$_H$	Setting when the CPU is operating at 64 MHz
0B$_H$	Setting when the CPU is operating at 80 MHz
13$_H$	Setting when the CPU is operating at 160 MHz
17$_H$	(Initial value)
Other than the above	Settings are prohibited.

Caution After release from the reset state, set the waiting time for access by the time of the first access to data-flash memory.

3.7 System Error Notification Function

This section describes the registers involved in system error notification.

3.7.1 Overview of System Error Notification Registers

Table 3-8 List of System Error Notification Function Registers

Register Name	Symbol	Address
System error control register	SEG_CONT	FFFF 64B0$_H$
	SEG_CONTL	
System error flag register	SEG_FLAG	FFFF 64B2$_H$
	SEG_FLAGL	

3.7.2 System Error Notification Function Registers in Detail

(1) SEG_CONT – System Error Control Register (SEG_CONT, SEG_CONTL)

This register is used to set up the generation of SYSERR exceptions due to system error sources.

Access SEG_CONT is readable/writable in 16-bit units.

SEG_CONTL consists of the 8 lower-order bits of SEG_CONT and is readable/writable in 8- and 1-bit units.

Address FFFF 64B0$_H$

Initial value 0000$_H$

	15	14	13	12	11	10	9	8
SEG_CONT	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
(SEG_CONTL)	SEG_ CONT DMAE	0	SEG_ CONT SEGE	SEG_ CONT RAME	0	SEG_ CONT EXTE	SEG_ CONT FCHE	0
	R/W	R	R/W	R/W	R	R/W	R/W	R

Table 3-9 SEG_CONT Register (1/2)

Bit Position	Bit Name	Function
7	SEG_ CONT DMAE	DMA error notification enable This bit is used to set the behavior when DMA or DTS access leads to the following errors. • Uncorrectable ECC error in access to data in the code-flash area • Uncorrectable ECC error in access to data in the on-chip RAM area • Access to the reserved area (undefined area) 0: SYSERR exception is not generated (initial value). 1: SYSERR exception is generated.
5	SEG_ CONT SEGE	Reserved area access notification enable This bit is used to set the behavior in response to attempted CPU access to data in the reserved area (undefined area). 0: SYSERR exception is not generated (initial value). 1: SYSERR exception is generated. Note. The reserved area (undefined area) in the address range FFFF 0000$_H$ to FFFF 4FFF$_H$ is for use in future expansion. The user is unable to use this area. A system error will occur if access is attempted.

Table 3-9 SEG_CONT Register (2/2)

Bit Position	Bit Name	Function
4	SEG_CONTRAME	CPU on-chip RAM area error notification enable This bit is used to set the behavior when CPU access to data in the on-chip RAM area leads to an uncorrectable error. 0: SYSERR exception is not generated (initial value). 1: SYSERR exception is generated.
2	SEG_CONTEXTE	EXT area error notification enable This bit is used to set the behavior when CPU access to data in data-flash memory or peripheral I/O leads to the following errors. • An error indicated by data-flash memory at the time the memory is read (the error occurs due to the execution of 8- and 16-bit loading, i.e. ld.b and ld.h, from the data-flash area). • Access to peripheral I/O causes a slave error (SLV), undefined area access error (UAA) or access time out (ATO) to be generated. The error is specified in the peripheral I/O bus PSELG control register (APC). 0: SYSERR exception is not generated (initial value). 1: SYSERR exception is generated.
1	SEG_CONTFCHE	Code flash error notification enable This bit is used to set the behavior when CPU access to data in the code-flash area leads to an uncorrectable ECC error. 0: SYSERR exception is not generated (initial value). 1: SYSERR exception is generated.

Caution When a bit is set to 1, a SYSERR exception is generated when a corresponding error occurs. When a bit is set to 0, the error flag corresponding to the error source is set, but a SYSERR exception is not generated.

(2) SEG_FLAG – System Error Flag Register (SEG_FLAG, SEG_FLAGL)

This register is used to store results of the detection of system errors. When a system error is detected, the flag corresponding to the error source is set to 1. Setting flags by directly writing to the register is also possible, providing a way to intentionally generate SYSERR exceptions. A flag is cleared by reading it as 1 and then writing 0 to it. Even when 0 is later written to a flag that was read as 0, the flags will still be set to reflect any errors that occurred between the reading and writing.

Access SEG_FLAG is readable/writable in 16-bit units.

SEG_FLAGL consists of the 8 lower-order bits of SEG_FLAG and is readable/writable in 8- and 1-bit units.

Address FFFF 64B2$_H$

Initial value 0000$_H$

	15	14	13	12	11	10	9	8
SEG_FLAG	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
(SEG_FLAGL)	SEG_ FLAG DMAF	0	SEG_ FLAG SEGF	SEG_ FLAG RAMF	0	SEG_ FLAG EXTF	SEG_ FLAG FCHF	0
	R/W	R	R/W	R/W	R	R/W	R/W	R

Table 3-10 SEG_FLAG Register (1/2)

Bit Position	Bit Name	Function
7	SEG_ FLAG DMAF	DMA/DTS error flag This flag is set when DMA or DTS access leads to the following errors. • Uncorrectable error generated from flash ROM in response to attempted access to data in the code-flash area • Uncorrectable error in response to access to data in the on-chip RAM area • Access to the reserved area (undefined area) 0: DMA/DTS error is not generated 1: DMA/DTS error is generated
5	SEG_ FLAG SEGF	Reserved area error flag This flag is set in response to attempted CPU access to data in the reserved area (undefined area). 0: Error in the reserved area is not generated 1: Error in the reserved area is generated Note. The reserved area (undefined area) in the address range FFFF 0000$_H$ to FFFF 4FFF$_H$ is for use in future expansion. The user is unable to use this area. A system error will occur if access is attempted.

Table 3-10 SEG_FLAG Register (2/2)

Bit Position	Bit Name	Function
4	SEG_ FLAG RAMF	CPU on-chip RAM area error flag This flag is set when CPU access to data in the on-chip RAM area leads to an uncorrectable ECC error. 0: Error in the on-chip RAM area is not generated 1: Error in the on-chip RAM area is generated
2	SEG_ FLAG EXTF	EXT area error flag This flag is set when CPU access to data in data-flash memory or peripheral I/O area leads to the following errors. • An error indicated by data-flash memory at the time the memory is read (the error occurs due to the execution of 8- and 16-bit loading, i.e. ld.b and ld.h, from the data-flash area). • Access to peripheral I/O causes a slave error (SLV), undefined area access error (UAA) or access time out (ATO) to be generated. The error is specified in the peripheral I/O bus PSELG control register (APC). 0: Error in the EXT area is not generated 1: Error in the EXT area is generated
1	SEG_ FLAG FCHF	Code flash error flag This flag is set when CPU access to data in the code-flash area leads to an uncorrectable ECC error. 0: Flash ROM error is not generated. 1: Flash ROM error is generated.

Section 4 Interrupt Functions

4.1 Features

An event where a specific event forces branching from a currently running program to a different program is called an exception. This section describes the exception processing functions of this product.

Table 4-1 List of Exception Sources

Name	Symbol	Source	Priority	Exception Level
CPU initialization	RESET	Reset input	1	–
Reserved	–	–	2	–
FE level non-maskable interrupt	FENMI	FENMI input	3	FE
System error exception	SYSERR	System error (SYSERR) input*1	4	FE
Peripheral device protection exception	PPI	Peripheral device protection violation	5	FE
Timing supervision exception	TSI	Timing supervision violation	6	FE
FE level maskable interrupt	FEINT	FEINT input	7	FE
Floating-point operation exception (imprecise)*4	FPI	FPU instruction	8	EI
EI level maskable interrupt	EIINT	EIINT input	9	EI
Reserved	–	–	10	–
Execution protection exception	MIP	Execution protection violation	11	FE
Memory error exception	MEP	Instruction access error input*2	12	FE
Data protection exception	MDP	Data protection violation	13	FE
Floating-point operation exception (precise)*4	FPP	FPU instruction		EI
Coprocessor unusable exception*3	UCPOP	Coprocessor instruction		FE
Reserved instruction exception	RIEX	Reserved instruction		FE
Reserved	–	–		–
FE level software exception	FETRAPEX	FETRAP instruction (vector = 1_H to F_H)		FE
EI level software exception	EITRAP0	TRAP0n instruction (vector = 00_H to $0F_H$)		EI
EI level software exception	EITRAP1	TRAP1n instruction (vector = 10_H to $1F_H$)		EI
System call exception	SYSCALLEX	SYSCALL instruction (vector = 00_H to FF_H)		EI

Note 1. Refer to 3.7, System Error Notification Function.

Note 2. An instruction-access error is generated at the point where instruction fetching for the execution of an instruction includes an address where an error is actually generated. Listed below are the error sources.

- Code flash ECC error
- On-chip RAM ECC error
- Reserved area error (access to an address in the range $FFFF0000_H$ to $FFFF4FFF_H$)

Note 3. Not supported on the μPD70F3506, μPD70F3507, μPD70F3508 and μPD70F3509.

Note 4. Not supported on the μPD70F3505A.

The following three exceptions from table 4-1 are called interrupts, and are thus described in this section. All other exceptions and their handling are described in the V850E2M Architecture Manual (R01US0001E).

- FE level non-maskable interrupt (FENMI): One source

  (generating factors: INTCME, INTWDTA0NMI, or INTCLMA0-2)

  - Resume disabled, recover disabled (for error report)

- FE level maskable interrupt (FEINT): One source

  (generating factors: INTISG pin or NMI pin)

  - Resume enabled, recover enabled*
  - Highest priority interrupt (except FENMI)

- EI level maskable interrupt (EIINT): 202 sources

  - Resume enabled, recover enabled.*
  - Interrupt masking can be specified per interrupt channel.
  - 16 interrupt priority levels can be specified for each interrupt channel
  - In this section the EIINT that corresponds to interrupt channel n is indicated by EIINTn.

Note * Resume: Indicates whether execution restart from the last position at which program execution was interrupted is possible.

Recover: Indicates whether recovery to the processor status (status of processor resources including general-purpose registers and system registers) at the time of program execution interruption is possible.

These interrupt sources are listed in table 4-2.

4.2 List of Interrupt Sources

Table 4-2 List of Interrupt Sources (1/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
FE	FENMI	FENMI	Interrupt occurred with INTCME, INTWDTA0NMI, or INTCLMA0-2.*2	–	–	00000020_H	$+0020_H$	–	–	FNC	FFFF $645C_H$
FE	FEINT	FEINT	SGA interrupt (INTISG)*3 or NMI pin interrupt	–	–	00000010_H	$+0010_H$	currentPC	FERET	FIC	FFFF $645E_H$
EI	EIINT0	INTCME*4	CPU comparison error	Safety	0	00000080_H	$+0080_H$	currentPC	EIRET	ICCME	FFFF 6000_H
EI	EIINT1	INTISG*4	SGA interrupt	Safety	1	00000090_H	$+0090_H$	currentPC	EIRET	ICISG	FFFF 6002_H
EI	EIINT2	INTWDTA0NMI*4	NMI output from watchdog	WDTA0	2	$000000A0_H$	$+00A0_H$	currentPC	EIRET	ICWDTA0NMI	FFFF 6004_H
EI	EIINT3	INTWDTA0	WDTA0 75% interrupt	WDTA0	3	$000000B0_H$	$+00B0_H$	currentPC	EIRET	ICWDTA0	FFFF 6006_H
EI	EIINT4	INTCLMA0*4	Detection of CLMA0 abnormality	CLMA0	4	$000000C0_H$	$+00C0_H$	currentPC	EIRET	ICCLMA0	FFFF 6008_H
EI	EIINT5	INTCLMA1*4	Detection of CLMA1 abnormality	CLMA1	5	$000000D0_H$	$+00D0_H$	currentPC	EIRET	ICCLMA1	FFFF $600A_H$
EI	EIINT6	INTCLMA2*4	Detection of CLMA2 abnormality	CLMA2	6	$000000E0_H$	$+00E0_H$	currentPC	EIRET	ICCLMA2	FFFF $600C_H$
EI	EIINT7	INTECCCFEC	Detection or correction of an ECC error in code flash	Safety	7	$000000F0_H$	$+00F0_H$	currentPC	EIRET	ICECCCFEC	FFFF $600E_H$
EI	EIINT8	INTECCLRED	Detection of an ECC error in on-chip RAM	Safety	8	00000100_H	$+0100_H$	currentPC	EIRET	ICECCLRED	FFFF 6010_H
EI	EIINT9	INTECCLREC	Detection or correction of an ECC error in on-chip RAM	Safety	9	00000110_H	$+0110_H$	currentPC	EIRET	ICECCLREC	FFFF 6012_H
EI	EIINT10	INTECCCDRED2	Detection, due to cache flushing, of an ECC error in RAM (level 2)	Safety	10	00000120_H	$+0120_H$	currentPC	EIRET	ICECCCDRED2	FFFF 6014_H
EI	EIINT11	INTECCCTRED2	Detection, due to cache flushing, of an ECC error in TAG (level 2)	Safety	11	00000130_H	$+0130_H$	currentPC	EIRET	ICECCCTRED2	FFFF 6016_H
EI	EIINT12	INTECCCDRED1	Detection, due to cache flushing, of an ECC error in RAM (level 1)	Safety	12	00000140_H	$+0140_H$	currentPC	EIRET	ICECCCDRED1	FFFF 6018_H
EI	EIINT13	INTECCCTRED1	Detection, due to cache flushing, of an ECC error in TAG (level 1)	Safety	13	00000150_H	$+0150_H$	currentPC	EIRET	ICECCCTRED1	FFFF $601A_H$
EI	EIINT14	INTECCFRED*5	Detection of an ECC error in FLXA0 RAM	Safety	14	00000160_H	$+0160_H$	currentPC	EIRET	ICECCFRED	FFFF $601C_H$
EI	EIINT15	INTECCCNED	Detection of an ECC error in CAN RAM	Safety	15	00000170_H	$+0170_H$	currentPC	EIRET	ICECCCNED	FFFF $601E_H$
EI	EIINT16	INTLVI0	Low voltage detection	LVI	16	00000180_H	$+0180_H$	currentPC	EIRET	ICLVI0	FFFF 6020_H
EI	EIINT17	INTP0	INTP0 edge detection	Port	17	00000190_H	$+0190_H$	currentPC	EIRET	ICP0	FFFF 6022_H
EI	EIINT18	INTP1	INTP1 edge detection	Port	18	$000001A0_H$	$+01A0_H$	currentPC	EIRET	ICP1	FFFF 6024_H
EI	EIINT19	INTP2	INTP2 edge detection	Port	19	$000001B0_H$	$+01B0_H$	currentPC	EIRET	ICP2	FFFF 6026_H
EI	EIINT20	INTP3	INTP3 edge detection	Port	20	$000001C0_H$	$+01C0_H$	currentPC	EIRET	ICP3	FFFF 6028_H
EI	EIINT21	INTP4	INTP4 edge detection	Port	21	$000001D0_H$	$+01D0_H$	currentPC	EIRET	ICP4	FFFF $602A_H$

Table 4-2 List of Interrupt Sources (2/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT22	INTP5	INTP5 edge detection	Port	22	$000001E0_H$	$+01E0_H$	currentPC	EIRET	ICP5	FFFF $602C_H$
EI	EIINT23	INTP6	INTP6 edge detection	Port	23	$000001F0_H$	$+01F0_H$	currentPC	EIRET	ICP6	FFFF $602E_H$
EI	EIINT24	INTP7	INTP7 edge detection	Port	24	00000200_H	$+0200_H$	currentPC	EIRET	ICP7	FFFF 6030_H
EI	EIINT25	INTP8	INTP8 edge detection	Port	25	00000210_H	$+0210_H$	currentPC	EIRET	ICP8	FFFF 6032_H
EI	EIINT26	INTP9	INTP9 edge detection	Port	26	00000220_H	$+0220_H$	currentPC	EIRET	ICP9	FFFF 6034_H
EI	EIINT27	INTP10*5	INTP10 edge detection	Port	27	00000230_H	$+0230_H$	currentPC	EIRET	ICP10	FFFF 6036_H
EI	EIINT28	INTDMA0	DMA transfer completed 0	DMA	28	00000240_H	$+0240_H$	currentPC	EIRET	ICDMA0	FFFF 6038_H
EI	EIINT29	INTDMACT0	DMA transfer count matched 0	DMA	29	00000250_H	$+0250_H$	currentPC	EIRET	ICDMACT0	FFFF $603A_H$
EI	EIINT30	INTDMA1	DMA transfer completed 1	DMA	30	00000260_H	$+0260_H$	currentPC	EIRET	ICDMA1	FFFF $603C_H$
EI	EIINT31	INTDMACT1	DMA transfer count matched 1	DMA	31	00000270_H	$+0270_H$	currentPC	EIRET	ICDMACT1	FFFF $603E_H$
EI	EIINT32	INTDMA2	DMA transfer completed 2	DMA	32	00000280_H	$+0280_H$	currentPC	EIRET	ICDMA2	FFFF 6040_H
EI	EIINT33	INTDMACT2	DMA transfer count matched 2	DMA	33	00000290_H	$+0290_H$	currentPC	EIRET	ICDMACT2	FFFF 6042_H
EI	EIINT34	INTDMA3	DMA transfer completed 3	DMA	34	$000002A0_H$	$+02A0_H$	currentPC	EIRET	ICDMA3	FFFF 6044_H
EI	EIINT35	INTDMACT3	DMA transfer count matched 3	DMA	35	$000002B0_H$	$+02B0_H$	currentPC	EIRET	ICDMACT3	FFFF 6046_H
EI	EIINT36	INTDMA4	DMA transfer completed 4	DMA	36	$000002C0_H$	$+02C0_H$	currentPC	EIRET	ICDMA4	FFFF 6048_H
EI	EIINT37	INTDMACT4	DMA transfer count matched 4	DMA	37	$000002D0_H$	$+02D0_H$	currentPC	EIRET	ICDMACT4	FFFF $604A_H$
EI	EIINT38	INTDMA5	DMA transfer completed 5	DMA	38	$000002E0_H$	$+02E0_H$	currentPC	EIRET	ICDMA5	FFFF $604C_H$
EI	EIINT39	INTDMACT5	DMA transfer count matched 5	DMA	39	$000002F0_H$	$+02F0_H$	currentPC	EIRET	ICDMACT5	FFFF $604E_H$
EI	EIINT40	INTDMA6	DMA transfer completed 6	DMA	40	00000300_H	$+0300_H$	currentPC	EIRET	ICDMA6	FFFF 6050_H
EI	EIINT41	INTDMACT6	DMA transfer count matched 6	DMA	41	00000310_H	$+0310_H$	currentPC	EIRET	ICDMACT6	FFFF 6052_H
EI	EIINT42	INTDMA7	DMA transfer completed 7	DMA	42	00000320_H	$+0320_H$	currentPC	EIRET	ICDMA7	FFFF 6054_H
EI	EIINT43	INTDMACT7	DMA transfer count matched 7	DMA	43	00000330_H	$+0330_H$	currentPC	EIRET	ICDMACT7	FFFF 6056_H
EI	EIINT44	INTOSTM0	OST0 interrupt signal	OSTM0	44	00000340_H	$+0340_H$	currentPC	EIRET	ICOSTM0	$FFFF6058_H$
EI	EIINT45	INTOSTM1	OST1 interrupt signal	OSTM1	45	00000350_H	$+0350_H$	currentPC	EIRET	ICOSTM1	FFFF $605A_H$
EI	EIINT46	INTTAUJ0I0	TAUJ0 CH0 interrupt signal	TAUJ0	46	00000360_H	$+0360_H$	currentPC	EIRET	ICTAUJ0I0	FFFF $605C_H$
EI	EIINT47	INTTAUJ0I1	TAUJ0 CH1 interrupt signal	TAUJ0	47	00000370_H	$+0370_H$	currentPC	EIRET	ICTAUJ0I1	FFFF $605E_H$
EI	EIINT48	INTTAUJ0I2	TAUJ0 CH2 interrupt signal	TAUJ0	48	00000380_H	$+0380_H$	currentPC	EIRET	ICTAUJ0I2	FFFF 6060_H
EI	EIINT49	INTTAUJ0I3	TAUJ0 CH3 interrupt signal	TAUJ0	49	00000390_H	$+0390_H$	currentPC	EIRET	ICTAUJ0I3	FFFF 6062_H
EI	EIINT50	INTTAUJ1I0	TAUJ1 CH0 interrupt signal	TAUJ1	50	$000003A0_H$	$+03A0_H$	currentPC	EIRET	ICTAUJ1I0	FFFF 6064_H
EI	EIINT51	INTTAUJ1I1	TAUJ1 CH1 interrupt signal	TAUJ1	51	$000003B0_H$	$+03B0_H$	currentPC	EIRET	ICTAUJ1I1	FFFF 6066_H
EI	EIINT52	INTTAUJ1I2	TAUJ1 CH2 interrupt signal	TAUJ1	52	$000003C0_H$	$+03C0_H$	currentPC	EIRET	ICTAUJ1I2	FFFF 6068_H

Table 4-2 List of Interrupt Sources (3/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT53	INTTAUJ1I3	TAUJ1 CH3 interrupt signal	TAUJ1	53	$000003D0_H$	$+03D0_H$	currentPC	EIRET	ICTAUJ1I3	FFFF $606A_H$
EI	EIINT54	INTTAUA0I0	TAUA0 CH0 interrupt signal	TAUA0	54	$000003E0_H$	$+03E0_H$	currentPC	EIRET	ICTAUA0I0	FFFF $606C_H$
EI	EIINT55	INTTAUA0I1	TAUA0 CH1 interrupt signal	TAUA0	55	$000003F0_H$	$+03F0_H$	currentPC	EIRET	ICTAUA0I1	FFFF $606E_H$
EI	EIINT56	INTTAUA0I2	TAUA0 CH2 interrupt signal	TAUA0	56	00000400_H	$+0400_H$	currentPC	EIRET	ICTAUA0I2	FFFF 6070_H
EI	EIINT57	INTTAUA0I3	TAUA0 CH3 interrupt signal	TAUA0	57	00000410_H	$+0410_H$	currentPC	EIRET	ICTAUA0I3	FFFF 6072_H
EI	EIINT58	INTTAUA0I4	TAUA0 CH4 interrupt signal	TAUA0	58	00000420_H	$+0420_H$	currentPC	EIRET	ICTAUA0I4	FFFF 6074_H
EI	EIINT59	INTTAUA0I5	TAUA0 CH5 interrupt signal	TAUA0	59	00000430_H	$+0430_H$	currentPC	EIRET	ICTAUA0I5	FFFF 6076_H
EI	EIINT60	INTTAUA0I6	TAUA0 CH6 interrupt signal	TAUA0	60	00000440_H	$+0440_H$	currentPC	EIRET	ICTAUA0I6	FFFF 6078_H
EI	EIINT61	INTTAUA0I7	TAUA0 CH7 interrupt signal	TAUA0	61	00000450_H	$+0450_H$	currentPC	EIRET	ICTAUA0I7	FFFF $607A_H$
EI	EIINT62	INTTAUA0I8	TAUA0 CH8 interrupt signal	TAUA0	62	00000460_H	$+0460_H$	currentPC	EIRET	ICTAUA0I8	FFFF $607C_H$
EI	EIINT63	INTTAUA0I9	TAUA0 CH9 interrupt signal	TAUA0	63	00000470_H	$+0470_H$	currentPC	EIRET	ICTAUA0I9	FFFF $607E_H$
EI	EIINT64	INTTAUA0I10	TAUA0 CH10 interrupt signal	TAUA0	64	00000480_H	$+0480_H$	currentPC	EIRET	ICTAUA0I10	FFFF 6080_H
EI	EIINT65	INTTAUA0I11	TAUA0 CH11 interrupt signal	TAUA0	65	00000490_H	$+0490_H$	currentPC	EIRET	ICTAUA0I11	FFFF 6082_H
EI	EIINT66	INTTAUA0I12	TAUA0 CH12 interrupt signal	TAUA0	66	$000004A0_H$	$+04A0_H$	currentPC	EIRET	ICTAUA0I12	FFFF 6084_H
EI	EIINT67	INTTAUA0I13	TAUA0 CH13 interrupt signal	TAUA0	67	$000004B0_H$	$+04B0_H$	currentPC	EIRET	ICTAUA0I13	FFFF 6086_H
EI	EIINT68	INTTAUA0I14	TAUA0 CH14 interrupt signal	TAUA0	68	$000004C0_H$	$+04C0_H$	currentPC	EIRET	ICTAUA0I14	FFFF 6088_H
EI	EIINT69	INTTAUA0I15	TAUA0 CH15 interrupt signal	TAUA0	69	$000004D0_H$	$+04D0_H$	currentPC	EIRET	ICTAUA0I15	FFFF $608A_H$
EI	EIINT70	INTTAUA1I0	TAUA1 CH0 interrupt signal	TAUA1	70	$000004E0_H$	$+04E0_H$	currentPC	EIRET	ICTAUA1I0	FFFF $608C_H$
EI	EIINT71	INTTAUA1I1	TAUA1 CH1 interrupt signal	TAUA1	71	$000004F0_H$	$+04F0_H$	currentPC	EIRET	ICTAUA1I1	FFFF $608E_H$
EI	EIINT72	INTTAUA1I2	TAUA1 CH2 interrupt signal	TAUA1	72	00000500_H	$+0500_H$	currentPC	EIRET	ICTAUA1I2	FFFF 6090_H
EI	EIINT73	INTTAUA1I3	TAUA1 CH3 interrupt signal	TAUA1	73	00000510_H	$+0510_H$	currentPC	EIRET	ICTAUA1I3	FFFF 6092_H
EI	EIINT74	INTTAUA1I4	TAUA1 CH4 interrupt signal	TAUA1	74	00000520_H	$+0520_H$	currentPC	EIRET	ICTAUA1I4	FFFF 6094_H
EI	EIINT75	INTTAUA1I5	TAUA1 CH5 interrupt signal	TAUA1	75	00000530_H	$+0530_H$	currentPC	EIRET	ICTAUA1I5	FFFF 6096_H
EI	EIINT76	INTTAUA1I6	TAUA1 CH6 interrupt signal	TAUA1	76	00000540_H	$+0540_H$	currentPC	EIRET	ICTAUA1I6	FFFF 6098_H
EI	EIINT77	INTTAUA1I7	TAUA1 CH7 interrupt signal	TAUA1	77	00000550_H	$+0550_H$	currentPC	EIRET	ICTAUA1I7	FFFF $609A_H$
EI	EIINT78	INTTAUA1I8	TAUA1 CH8 interrupt signal	TAUA1	78	00000560_H	$+0560_H$	currentPC	EIRET	ICTAUA1I8	FFFF $609C_H$
EI	EIINT79	INTTAUA1I9	TAUA1 CH9 interrupt signal	TAUA1	79	00000570_H	$+0570_H$	currentPC	EIRET	ICTAUA1I9	FFFF $609E_H$
EI	EIINT80	INTTAUA1I10	TAUA1 CH10 interrupt signal	TAUA1	80	00000580_H	$+0580_H$	currentPC	EIRET	ICTAUA1I10	FFFF $60A0_H$
EI	EIINT81	INTTAUA1I11	TAUA1 CH11 interrupt signal	TAUA1	81	00000590_H	$+0590_H$	currentPC	EIRET	ICTAUA1I11	FFFF $60A2_H$
EI	EIINT82	INTTAUA1I12	TAUA1 CH12 interrupt signal	TAUA1	82	$000005A0_H$	$+05A0_H$	currentPC	EIRET	ICTAUA1I12	FFFF $60A4_H$
EI	EIINT83	INTTAUA1I13	TAUA1 CH13 interrupt signal	TAUA1	83	$000005B0_H$	$+05B0_H$	currentPC	EIRET	ICTAUA1I13	FFFF $60A6_H$

Table 4-2 List of Interrupt Sources (4/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT84	INTTAUA1I14	TAUA1 CH14 interrupt signal	TAUA1	84	$000005C0_H$	$+05C0_H$	currentPC	EIRET	ICTAUA1I14	FFFF $60A8_H$
EI	EIINT85	INTTAUA1I15	TAUA1 CH15 interrupt signal	TAUA1	85	$000005D0_H$	$+05D0_H$	currentPC	EIRET	ICTAUA1I15	FFFF $60AA_H$
EI	EIINT86	INTADCA0ERR	ADC error interrupt signal	ADCA0	86	$000005E0_H$	$+05E0_H$	currentPC	EIRET	ICADCA0ERR	FFFF $60AC_H$
EI	EIINT87	INTADCA0I0	ADC coversion completed CG0	ADCA0	87	$000005F0_H$	$+05F0_H$	currentPC	EIRET	ICADCA0I0	FFFF $60AE_H$
EI	EIINT88	INTADCA0I1	ADC coversion completed CG1	ADCA0	88	00000600_H	$+0600_H$	currentPC	EIRET	ICADCA0I1	FFFF $60B0_H$
EI	EIINT89	INTADCA0I2	ADC coversion completed CG2	ADCA0	89	00000610_H	$+0610_H$	currentPC	EIRET	ICADCA0I2	FFFF $60B2_H$
EI	EIINT90	INTADCA0LLT	Timing of latest conversion by the ADC	ADCA0	90	00000620_H	$+0620_H$	currentPC	EIRET	ICADCA0LLT	FFFF $60B4_H$
EI	EIINT91	INTCSIH0IRE	CSIH0 reception error	CSIH0	91	00000630_H	$+0630_H$	currentPC	EIRET	ICCSIH0IRE	FFFF $60B6_H$
EI	EIINT92	INTCSIH0IR	CSIH0 reception complete	CSIH0	92	00000640_H	$+0640_H$	currentPC	EIRET	ICCSIH0IR	FFFF $60B8_H$
EI	EIINT93	INTCSIH0IC	CSIH0 communication complete	CSIH0	93	00000650_H	$+0650_H$	currentPC	EIRET	ICCSIH0IC	FFFF $60BA_H$
EI	EIINT94	INTCSIH0IJC	CSIH0 job complete	CSIH0	94	00000660_H	$+0660_H$	currentPC	EIRET	ICCSIH0IJC	FFFF $60BC_H$
EI	EIINT95	INTCSIH1IRE*5	CSIH1 reception error	CSIH1	95	00000670_H	$+0670_H$	currentPC	EIRET	ICCSIH1IRE	FFFF $60BE_H$
EI	EIINT96	INTCSIH1IR*5	CSIH1 reception complete	CSIH1	96	00000680_H	$+0680_H$	currentPC	EIRET	ICCSIH1IR	FFFF $60C0_H$
EI	EIINT97	INTCSIH1IC*5	CSIH1 communication complete	CSIH1	97	00000690_H	$+0690_H$	currentPC	EIRET	ICCSIH1IC	FFFF $60C2_H$
EI	EIINT98	INTCSIH1IJC*5	CSIH1 job complete	CSIH1	98	$000006A0_H$	$+06A0_H$	currentPC	EIRET	ICCSIH1IJC	FFFF $60C4_H$
EI	EIINT99	INTCSIG0IRE	CSIG0 reception error	CSIG0	99	$000006B0_H$	$+06B0_H$	currentPC	EIRET	ICCSIG0IRE	FFFF $60C6_H$
EI	EIINT100	INTCSIG0IR	CSIG0 reception complete	CSIG0	100	$000006C0_H$	$+06C0_H$	currentPC	EIRET	ICCSIG0IR	FFFF $60C8_H$
EI	EIINT101	INTCSIG0IC	CSIG0 communication complete	CSIG0	101	$000006D0_H$	$+06D0_H$	currentPC	EIRET	ICCSIG0IC	FFFF $60CA_H$
EI	EIINT102	INTCSIG1IRE	CSIG1 reception error	CSIG1	102	$000006E0_H$	$+06E0_H$	currentPC	EIRET	ICCSIG1IRE	FFFF $60CC_H$
EI	EIINT103	INTCSIG1IR	CSIG1 reception complete	CSIG1	103	$000006F0_H$	$+06F0_H$	currentPC	EIRET	ICCSIG1IR	FFFF $60CE_H$
EI	EIINT104	INTCSIG1IC	CSIG1 communication complete	CSIG1	104	00000700_H	$+0700_H$	currentPC	EIRET	ICCSIG1IC	FFFF $60D0_H$
EI	EIINT105	INTCSIG2IRE	CSIG2 reception error	CSIG2	105	00000710_H	$+0710_H$	currentPC	EIRET	ICCSIG2IRE	FFFF $60D2_H$
EI	EIINT106	INTCSIG2IR	CSIG2 reception complete	CSIG2	106	00000720_H	$+0720_H$	currentPC	EIRET	ICCSIG2IR	FFFF $60D4_H$
EI	EIINT107	INTCSIG2IC	CSIG2 communication complete	CSIG2	107	00000730_H	$+0730_H$	currentPC	EIRET	ICCSIG2IC	FFFF $60D6_H$
EI	EIINT108	INTURTH0IS	UARTH0 status interrupt	UARTH0	108	00000740_H	$+0740_H$	currentPC	EIRET	ICURTH0IS	FFFF $60D8_H$
EI	EIINT109	INTURTH0IR	UARTH0 reception complete	UARTH0	109	00000750_H	$+0750_H$	currentPC	EIRET	ICURTH0IR	FFFF $60DA_H$
EI	EIINT110	INTURTH0IT	UARTH0 transmission complete	UARTH0	110	00000760_H	$+0760_H$	currentPC	EIRET	ICURTH0IT	FFFF $60DC_H$
EI	EIINT111	INTURTH1IS	UARTH1 status interrupt	UARTH1	111	00000770_H	$+0770_H$	currentPC	EIRET	ICURTH1IS	FFFF $60DE_H$
EI	EIINT112	INTURTH1IR	UARTH1 reception complete	UARTH1	112	00000780_H	$+0780_H$	currentPC	EIRET	ICURTH1IR	FFFF $60E0_H$
EI	EIINT113	INTURTH1IT	UARTH1 transmission complete	UARTH1	113	00000790_H	$+0790_H$	currentPC	EIRET	ICURTH1IT	FFFF $60E2_H$
EI	EIINT114	INTURTH2IS	UARTH2 status interrupt	UARTH2	114	$000007A0_H$	$+07A0_H$	currentPC	EIRET	ICURTH2IS	FFFF $60E4_H$

Table 4-2 List of Interrupt Sources (5/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT115	INTURTH2IR	UARTH2 reception complete	UARTH2	115	$000007B0_H$	$+07B0_H$	currentPC	EIRET	ICURTH2IR	$FFFF\ 60E6_H$
EI	EIINT116	INTURTH2IT	UARTH2 transmission complete	UARTH2	116	$000007C0_H$	$+07C0_H$	currentPC	EIRET	ICURTH2IT	$FFFF\ 60E8_H$
EI	EIINT117	INTFCN0ERR	CAN0 error	FCN0	117	$000007D0_H$	$+07D0_H$	currentPC	EIRET	ICFCN0ERR	$FFFF\ 60EA_H$
EI	EIINT118	INTFCN0WUP	CAN0 wake up	FCN0	118	$000007E0_H$	$+07E0_H$	currentPC	EIRET	ICFCN0WUP	$FFFF\ 60EC_H$
EI	EIINT119	INTFCN0REC	CAN0 reception complete	FCN0	119	$000007F0_H$	$+07F0_H$	currentPC	EIRET	ICFCN0REC	$FFFF\ 60EE_H$
EI	EIINT120	INTFCN0TRX	CAN0 transmission complete	FCN0	120	00000800_H	$+0800_H$	currentPC	EIRET	ICFCN0TRX	$FFFF\ 60F0_H$
EI	EIINT121	INTFCN1ERR	CAN1 error	FCN1	121	00000810_H	$+0810_H$	currentPC	EIRET	ICFCN1ERR	$FFFF\ 60F2_H$
EI	EIINT122	INTFCN1WUP	CAN1 wake up	FCN1	122	00000820_H	$+0820_H$	currentPC	EIRET	ICFCN1WUP	$FFFF\ 60F4_H$
EI	EIINT123	INTFCN1REC	CAN1 reception complete	FCN1	123	00000830_H	$+0830_H$	currentPC	EIRET	ICFCN1REC	$FFFF\ 60F6_H$
EI	EIINT124	INTFCN1TRX	CAN1 transmission complete	FCN1	124	00000840_H	$+0840_H$	currentPC	EIRET	ICFCN1TRX	$FFFF\ 60F8_H$
EI	EIINT125	INTBRG0	BRG interrupt	BRG	125	00000850_H	$+0850_H$	currentPC	EIRET	ICBRG0	$FFFF\ 60FA_H$
EI	EIINT126	Reserved				00000860_H	$+0860_H$	currentPC	EIRET		
EI	EIINT127	INTFL	Flash programming complete interrupt (code/data)	Code flash/ Data flash	127	00000870_H	$+0870_H$	currentPC	EIRET	ICFL	$FFFF\ 60FE_H$
EI	EIINT128	INTSWN	NEXUS software interrupt	Software	128	00000880_H	$+0880_H$	currentPC	EIRET	ICSWN	$FFFF\ 6100_H$
EI	EIINT129	INTTSG20I00	TSG20 PWM cycle interrupt	TSG20	129	00000890_H	$+0890_H$	currentPC	EIRET	ICTSG20I00	$FFFF\ 6102_H$
EI	EIINT130	INTTSG20I01	TSG20 compare match 1 interrupt	TSG20	130	$000008A0_H$	$+08A0_H$	currentPC	EIRET	ICTSG20I01	$FFFF\ 6104_H$
EI	EIINT131	INTTSG20I02	TSG20 compare match 2 interrupt	TSG20	131	$000008B0_H$	$+08B0_H$	currentPC	EIRET	ICTSG20I02	$FFFF\ 6106_H$
EI	EIINT132	INTTSG20I03	TSG20 compare match 3 interrupt	TSG20	132	$000008C0_H$	$+08C0_H$	currentPC	EIRET	ICTSG20I03	$FFFF\ 6108_H$
EI	EIINT133	INTTSG20I04	TSG20 compare match 4 interrupt	TSG20	133	$000008D0_H$	$+08D0_H$	currentPC	EIRET	ICTSG20I04	$FFFF\ 610A_H$
EI	EIINT134	INTTSG20I05	TSG20 compare match 5 interrupt	TSG20	134	$000008E0_H$	$+08E0_H$	currentPC	EIRET	ICTSG20I05	$FFFF\ 610C_H$
EI	EIINT135	INTTSG20I06	TSG20 compare match 6 interrupt	TSG20	135	$000008F0_H$	$+08F0_H$	currentPC	EIRET	ICTSG20I06	$FFFF\ 610E_H$
EI	EIINT136	INTTSG20I07	TSG20 compare match 7 interrupt	TSG20	136	00000900_H	$+0900_H$	currentPC	EIRET	ICTSG20I07	$FFFF\ 6110_H$
EI	EIINT137	INTTSG20I08	TSG20 compare match 8 interrupt	TSG20	137	00000910_H	$+0910_H$	currentPC	EIRET	ICTSG20I08	$FFFF\ 6112_H$
EI	EIINT138	INTTSG20I09	TSG20 compare match 9 interrupt	TSG20	138	00000920_H	$+0920_H$	currentPC	EIRET	ICTSG20I09	$FFFF\ 6114_H$
EI	EIINT139	INTTSG20I10	TSG20 compare match 10 interrupt	TSG20	139	00000930_H	$+0930_H$	currentPC	EIRET	ICTSG20I10	$FFFF\ 6116_H$
EI	EIINT140	INTTSG20I11	TSG20 compare match 11 interrupt	TSG20	140	00000940_H	$+0940_H$	currentPC	EIRET	ICTSG20I11	$FFFF\ 6118_H$
EI	EIINT141	INTTSG20I12	TSG20 compare match 12 interrupt	TSG20	141	00000950_H	$+0950_H$	currentPC	EIRET	ICTSG20I12	$FFFF\ 611A_H$
EI	EIINT142	INTTSG20IPEK	TSG20 peak interrupt	TSG20	142	00000960_H	$+0960_H$	currentPC	EIRET	ICTSG20IPEK	$FFFF\ 611C_H$
EI	EIINT143	INTTSG20IVLY	TSG20 valley interrupt	TSG20	143	00000970_H	$+0970_H$	currentPC	EIRET	ICTSG20IVLY	$FFFF\ 611E_H$
EI	EIINT144	INTTSG20IER	TSG20 error interrupt	TSG20	144	00000980_H	$+0980_H$	currentPC	EIRET	ICTSG20IER	$FFFF\ 6120_H$

Table 4-2 List of Interrupt Sources (6/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT145	INTTSG20IWN	TSG20 warning interrupt	TSG20	145	00000990_H	$+0990_H$	currentPC	EIRET	ICTSG20IWN	FFFF 6122_H
EI	EIINT146	INTTSG21I00	TSG21 PWM cycle interrupt	TSG21	146	$000009A0_H$	$+09A0_H$	currentPC	EIRET	ICTSG21I00	FFFF 6124_H
EI	EIINT147	INTTSG21I01	TSG21 compare match 1 interrupt	TSG21	147	$000009B0_H$	$+09B0_H$	currentPC	EIRET	ICTSG21I01	FFFF 6126_H
EI	EIINT148	INTTSG21I02	TSG21 compare match 2 interrupt	TSG21	148	$000009C0_H$	$+09C0_H$	currentPC	EIRET	ICTSG21I02	FFFF 6128_H
EI	EIINT149	INTTSG21I03	TSG21 compare match 3 interrupt	TSG21	149	$000009D0_H$	$+09D0_H$	currentPC	EIRET	ICTSG21I03	FFFF $612A_H$
EI	EIINT150	INTTSG21I04	TSG21 compare match 4 interrupt	TSG21	150	$000009E0_H$	$+09E0_H$	currentPC	EIRET	ICTSG21I04	FFFF $612C_H$
EI	EIINT151	INTTSG21I05	TSG21 compare match 5 interrupt	TSG21	151	$000009F0_H$	$+09F0_H$	currentPC	EIRET	ICTSG21I05	FFFF $612E_H$
EI	EIINT152	INTTSG21I06	TSG21 compare match 6 interrupt	TSG21	152	$00000A00_H$	$+0A00_H$	currentPC	EIRET	ICTSG21I06	FFFF 6130_H
EI	EIINT153	INTTSG21I07	TSG21 compare match 7 interrupt	TSG21	153	$00000A10_H$	$+0A10_H$	currentPC	EIRET	ICTSG21I07	FFFF 6132_H
EI	EIINT154	INTTSG21I08	TSG21 compare match 8 interrupt	TSG21	154	$00000A20_H$	$+0A20_H$	currentPC	EIRET	ICTSG21I08	FFFF 6134_H
EI	EIINT155	INTTSG21I09	TSG21 compare match 9 interrupt	TSG21	155	$00000A30_H$	$+0A30_H$	currentPC	EIRET	ICTSG21I09	FFFF 6136_H
EI	EIINT156	INTTSG21I10	TSG21 compare match 10 interrupt	TSG21	156	$00000A40_H$	$+0A40_H$	currentPC	EIRET	ICTSG21I10	FFFF 6138_H
EI	EIINT157	INTTSG21I11	TSG21 compare match 11 interrupt	TSG21	157	$00000A50_H$	$+0A50_H$	currentPC	EIRET	ICTSG21I11	FFFF $613A_H$
EI	EIINT158	INTTSG21I12	TSG21 compare match 12 interrupt	TSG21	158	$00000A60_H$	$+0A60_H$	currentPC	EIRET	ICTSG21I12	FFFF $613C_H$
EI	EIINT159	INTTSG21IPEK	TSG21 peak interrupt	TSG21	159	$00000A70_H$	$+0A70_H$	currentPC	EIRET	ICTSG21IPEK	FFFF $613E_H$
EI	EIINT160	INTTSG21IVLY	TSG21 valley interrupt	TSG21	160	$00000A80_H$	$+0A80_H$	currentPC	EIRET	ICTSG21IVLY	FFFF 6140_H
EI	EIINT161	INTTSG21IER	TSG21 error interrupt	TSG21	161	$00000A90_H$	$+0A90_H$	currentPC	EIRET	ICTSG21IER	FFFF 6142_H
EI	EIINT162	INTTSG21IWN	TSG21 warning interrupt	TSG21	162	$00000AA0_H$	$+0AA0_H$	currentPC	EIRET	ICTSG21IWN	FFFF 6144_H
EI	EIINT163	INTENCA0IOV	ENCA0 overflow interrupt	ENCA0	163	$00000AB0_H$	$+0AB0_H$	currentPC	EIRET	ICENCA0IOV	FFFF 6146_H
EI	EIINT164	INTENCA0I0	ENCA0 capture/compare match 0 interrupt	ENCA0	164	$00000AC0_H$	$+0AC0_H$	currentPC	EIRET	ICENCA0I0	FFFF 6148_H
EI	EIINT165	INTENCA0I1	ENCA0 capture/compare match 1 interrupt	ENCA0	165	$00000AD0_H$	$+0AD0_H$	currentPC	EIRET	ICENCA0I1	FFFF $614A_H$
EI	EIINT166	INTENCA0IUD	ENCA0 underflow interrupt	ENCA0	166	$00000AE0_H$	$+0AE0_H$	currentPC	EIRET	ICENCA0IUD	FFFF $614C_H$
EI	EIINT167	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	167	$00000AF0_H$	$+0AF0_H$	currentPC	EIRET	ICENCA0IEC	FFFF $614E_H$
EI	EIINT168	INTENCA1IOV	ENCA1 overflow interrupt	ENCA1	168	$00000B00_H$	$+0B00_H$	currentPC	EIRET	ICENCA1IOV	FFFF 6150_H
EI	EIINT169	INTENCA1I0	ENCA1 capture/compare match 0 interrupt	ENCA1	169	$00000B10_H$	$+0B10_H$	currentPC	EIRET	ICENCA1I0	FFFF 6152_H
EI	EIINT170	INTENCA1I1	ENCA1 capture/compare match 1 interrupt	ENCA1	170	$00000B20_H$	$+0B20_H$	currentPC	EIRET	ICENCA1I1	FFFF 6154_H
EI	EIINT171	INTENCA1IUD	ENCA1 underflow interrupt	ENCA1	171	$00000B30_H$	$+0B30_H$	currentPC	EIRET	ICENCA1IUD	FFFF 6156_H
EI	EIINT172	INTENCA1IEC	ENCA1 encoder clear interrupt	ENCA1	172	$00000B40_H$	$+0B40_H$	currentPC	EIRET	ICENCA1IEC	FFFF 6158_H
EI	EIINT173	INTTPBA0IPRD	TPBA0 period-matched detection interrupt	TPB0	173	$00000B50_H$	$+0B50_H$	currentPC	EIRET	ICTPBA0IPRD	FFFF $615A_H$
EI	EIINT174	INTTPBA0IDTY	TPBA0 duty-cycle-matched detection	TPB0	174	$00000B60_H$	$+0B60_H$	currentPC	EIRET	ICTPBA0IDTY	FFFF $615C_H$

Table 4-2 List of Interrupt Sources (7/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT175	INTTPBA0IPAT	TPBA0 number-of-patterns matched detection interrupt	TPB0	175	$00000B70_H$	$+0B70_H$	currentPC	EIRET	ICTPBA0IPAT	FFFF $615E_H$
EI	EIINT176	INTTPBA1IPRD	TPBA1 period-matched detection interrupt	TPB1	176	$00000B80_H$	$+0B80_H$	currentPC	EIRET	ICTPBA1IPRD	FFFF 6160_H
EI	EIINT177	INTTPBA1IDTY	TPBA1 duty-cycle-matched detection	TPB1	177	$00000B90_H$	$+0B90_H$	currentPC	EIRET	ICTPBA1IDTY	FFFF 6162_H
EI	EIINT178	INTTPBA1IPAT	TPBA1 number-of-patterns matched detection interrupt	TPB1	178	$00000BA0_H$	$+0BA0_H$	currentPC	EIRET	ICTPBA1IPAT	FFFF 6164_H
EI	EIINT179	Reserved				$00000BB0_H$	$+0BB0_H$	currentPC	EIRET		
EI	EIINT180	Reserved				$00000BC0_H$	$+0BC0_H$	currentPC	EIRET		
EI	EIINT181	INTTAPA0IPEK0	TAPA0 peak interrupt	TAPA0	181	$00000BD0_H$	$+0BD0_H$	currentPC	EIRET	ICTAPA0IPEK0	FFFF $616A_H$
EI	EIINT182	INTTAPA0IVLY0	TAPA0 valley interrupt	TAPA0	182	$00000BE0_H$	$+0BE0_H$	currentPC	EIRET	ICTAPA0IVLY0	FFFF $616C_H$
EI	EIINT183	INTTAPA1IPEK0	TAPA1 peak interrupt	TAPA1	183	$00000BF0_H$	$+0BF0_H$	currentPC	EIRET	ICTAPA1IPEK0	FFFF $616E_H$
EI	EIINT184	INTTAPA1IVLY0	TAPA1 valley interrupt	TAPA1	184	$00000C00_H$	$+0C00_H$	currentPC	EIRET	ICTAPA1IVLY0	FFFF 6170_H
EI	EIINT185	INTFLX0I0*5	FLXA0 FRINTP0 interrupt 0	FLXA0	185	$00000C10_H$	$+0C10_H$	currentPC	EIRET	ICFLX0I0	FFFF 6172_H
EI	EIINT186	INTFLX0I1*5	FLXA0 FRINTP1 interrupt 1	FLXA0	186	$00000C20_H$	$+0C20_H$	currentPC	EIRET	ICFLX0I1	FFFF 6174_H
EI	EIINT187	INTFLX0I2*5	FLXA0 timer interrupt 0	FLXA0	187	$00000C30_H$	$+0C30_H$	currentPC	EIRET	ICFLX0I2	FFFF 6176_H
EI	EIINT188	INTFLX0I3*5	FLXA0 timer interrupt 1	FLXA0	188	$00000C40_H$	$+0C40_H$	currentPC	EIRET	ICFLX0I3	FFFF 6178_H
EI	EIINT189	INTFLX0SIFL08*5	FLXA0 transfer input buffer completed	FLXA0	189	$00000C50_H$	$+0C50_H$	currentPC	EIRET	ICFLX0SIFL08	FFFF $617A_H$
EI	EIINT190	INTFLX0SIFL09*5	FLXA0 transfer output buffer completed	FLXA0	190	$00000C60_H$	$+0C60_H$	currentPC	EIRET	ICFLX0SIFL09	FFFF $617C_H$
EI	EIINT191	INTFLX0SIFL02*5	FLXA0 cycle start interrupt	FLXA0	191	$00000C70_H$	$+0C70_H$	currentPC	EIRET	ICFLX0SIFL02	FFFF $617E_H$
EI	EIINT192	INTFLX0SIFL07*5	FLXA0 network management vector changed	FLXA0	192	$00000C80_H$	$+0C80_H$	currentPC	EIRET	ICFLX0SIFL07	FFFF 6180_H
EI	EIINT193	INTSW0	Software interrupt 0	Software	193	$00000C90_H$	$+0C90_H$	currentPC	EIRET	ICSW0	FFFF 6182_H
EI	EIINT194	INTSW1	Software interrupt 1	Software	194	$00000CA0_H$	$+0CA0_H$	currentPC	EIRET	ICSW1	FFFF 6184_H
EI	EIINT195	INTSW2*6	Software interrupt 2	Software	195	$00000CB0_H$	$+0CB0_H$	currentPC	EIRET	ICSW2	FFFF 6186_H
EI	EIINT196	INTSW3*6	Software interrupt 3	Software	196	$00000CC0_H$	$+0CC0_H$	currentPC	EIRET	ICSW3	FFFF 6188_H
EI	EIINT197	INTSW4*6	Software interrupt 4	Software	197	$00000CD0_H$	$+0CD0_H$	currentPC	EIRET	ICSW4	FFFF $618A_H$
EI	EIINT198	INTSW5*6	Software interrupt 5	Software	198	$00000CE0_H$	$+0CE0_H$	currentPC	EIRET	ICSW5	FFFF $618C_H$
EI	EIINT199	INTSW6*6	Software interrupt 6	Software	199	$00000CF0_H$	$+0CF0_H$	currentPC	EIRET	ICSW6	FFFF $618E_H$
EI	EIINT200	INTSW7*6	Software interrupt 7	Software	200	$00000D00_H$	$+0D00_H$	currentPC	EIRET	ICSW7	FFFF 6190_H

Table 4-2 List of Interrupt Sources (8/8)

Level	Symbol	Symbol for Source of Interrupt	Generating Condition or Factor	Generating Unit	Default Priority	Exception Code	Handler Offset*1	PC Value on Return	Instruction for Return	Interrupt Control Register	
										Symbol	Address
EI	EIINT201	INTSW8*6	Software interrupt 8	Software	201	$00000D10_H$	$+0D10_H$	currentPC	EIRET	ICSW8	$FFFF\ 6192_H$
EI	EIINT202	INTSW9*6	Software interrupt 9	Software	202	$00000D20_H$	$+0D20_H$	currentPC	EIRET	ICSW9	$FFFF\ 6194_H$

Note 1. The base address is set by the "exception handler address switching function".
Note 2. If the INTCFGB register is set for level output of EI, FENMI interrupts will not be generated.
Note 3. If the INTCFGB register is set for level output of EI, the INTISG interrupt factor will not lead to the generation of FEINT.
Note 4. If the INTCFGB register is set for level output of FE, EI level interrupts will not be generated.
Note 5. Not supported by the μPD70F3505A.
Note 6. With interrupts INTSW0 to INTSW9, only writing 1 to the RFSW[0:9] bits in the ICSW0 to ICSW9 registers by software acts as the interrupt source.

4.3 Interrupt Controller Control Registers

Meaning of xx xx denotes the identifying names of the individual peripheral units.

4.3.1 ICxx: EI Level Interrupt Control Register

An ICxx register is provided for every EI level-maskable interrupt (EIINT) channel, and is used to configure control conditions.

Access Readable/writable in 1-, 8-, or 16-bit units.

However, bit-wise access through SET1, CLR1, NOT1 instructions to bits 15 to 13, bits 11 to 8 and bits 6 to 4 is prohibited.

Address FFFF 6000_H to FFFF 6195_H

Initial value $008F_H$

A reset from any source will initialize the bits.

Caution Do not access ICxx registers that are not listed in table 4-2.

ICxx	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(ICxxH)	0	0	0	RFxx	0	0	0	0
	R	R	R	R/W	R	R	R	R

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(ICxxL)	MKxx	0	0	0	P3xx	P2xx	P1xx	P0xx
	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
12	RFxx	This is an interrupt request flag. The RFxx bit is writable from a program. If the RFxx bit is set (to 1), an EI level maskable interrupt n (EINTn) will be generated in the same way as when an interrupt request is received. 0: No interrupt request is made (initial value). 1: Interrupt request is made.
7	MKxx	This is an interrupt mask bit. Setting the MKxx bit masks interrupt requests set in the interrupt request flag (RFxx), i.e. it may be used to obstruct interrupt requests from the given channel to the CPU core. If the MKxx bit is set for a channel, the ICSR.PMF bit will not hold indications of interrupts for that channel. The MKxx bit does not mask input from an interrupt input pin, so the interrupt request flag still gets set even if the MKxx bit is set. The setting of this bit also reflects the setting in the interrupt mask register (IMR). 0: Enables interrupt handling. 1: Prohibits interrupt handling (initial value).
3 to 0	P3xx to P0xx	These bits specify the interrupt priority as one of 16 levels, with 0 as the highest and 15 as the lowest. When multiple EI level-interrupt requests are made simultaneously, the interrupt from the source with the highest priority setting in these bits is selected and conveyed to the CPU core. When P3xx to P0xx bits for different sources specify the same priority level, the source with the lower channel number takes priority. This order is fixed.

Table 4-3 Addresses and Bits of the Interrupt Control Registers (1/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICCMEL	$FFFF6000_H$	MKCME	0	0	0	P3CME	P2CME	P1CME	P0CME
ICCMEH	$FFFF6001_H$	0	0	0	RFCME	0	0	0	0
ICISGL	$FFFF6002_H$	MKISG	0	0	0	P3ISG	P2ISG	P1ISG	P0ISG
ICISGH	$FFFF6003_H$	0	0	0	RFISG	0	0	0	0
ICWDTA0NMIL	$FFFF6004_H$	MKWDTA0NMI	0	0	0	P3WDTA0NMI	P2WDTA0NMI	P1WDTA0NMI	P0WDTA0NMI
ICWDTA0NMIH	$FFFF6005_H$	0	0	0	RFWDTA0NMI	0	0	0	0
ICWDTA0L	$FFFF6006_H$	MKWDTA0	0	0	0	P3WDTA0	P2WDTA0	P1WDTA0	P0WDTA0
ICWDTA0H	$FFFF6007_H$	0	0	0	RFWDTA0	0	0	0	0
ICCLMA0L	$FFFF6008_H$	MKCLMA0	0	0	0	P3CLMA0	P2CLMA0	P1CLMA0	P0CLMA0
ICCLMA0H	$FFFF6009_H$	0	0	0	RFCLMA0	0	0	0	0
ICCLMA1L	$FFFF600A_H$	MKCLMA1	0	0	0	P3CLMA1	P2CLMA1	P1CLMA1	P0CLMA1
ICCLMA1H	$FFFF600B_H$	0	0	0	RFCLMA1	0	0	0	0
ICCLMA2L	$FFFF600C_H$	MKCLMA2	0	0	0	P3CLMA2	P2CLMA2	P1CLMA2	P0CLMA2
ICCLMA2H	$FFFF600D_H$	0	0	0	RFCLMA2	0	0	0	0
ICECCCFECL	$FFFF600E_H$	MKECCCFEC	0	0	0	P3ECCCFEC	P2ECCCFEC	P1ECCCFEC	P0ECCCFEC
ICECCCFECH	$FFFF600F_H$	0	0	0	RFECCCFEC	0	0	0	0
ICECCLREDL	$FFFF6010_H$	MKECCLRED	0	0	0	P3ECCLRED	P2ECCLRED	P1ECCLRED	P0ECCLRED
ICECCLREDH	$FFFF6011_H$	0	0	0	RFECCREDD	0	0	0	0
ICECCLRECL	$FFFF6012_H$	MKECCLREC	0	0	0	P3ECCLREC	P2ECCREC	P1ECCLREC	P0ECCLREC
ICECCLRECH	$FFFF6013_H$	0	0	0	RFECCLREC	0	0	0	0
ICECCCDRED2L	$FFFF6014_H$	MKECCCDRED2	0	0	0	P3ECCCDRED2	P2ECCCDRED2	P1ECCCDRED2	P0ECCCDRED2
ICECCCDRED2H	$FFFF6015_H$	0	0	0	RFECCCDRED2	0	0	0	0
ICECCCTRED2L	$FFFF6016_H$	MKECCCTRED2	0	0	0	P3ECCCTRED2	P2ECCCTRED2	P1ECCCTRED2	P0ECCCTRED2
ICECCCTRED2H	$FFFF6017_H$	0	0	0	RFECCCTRED2	0	0	0	0
ICECCCDRED1L	$FFFF6018_H$	MKECCCDRED1	0	0	0	P3ECCCDRED1	P2ECCCDRED1	P1ECCCDRED1	P0ECCCDRED1
ICECCCDRED1H	$FFFF6019_H$	0	0	0	RFECCCDRED1	0	0	0	0
ICECCCTRED1L	$FFFF601A_H$	MKECCCTRED1	0	0	0	P3ECCCTRED1	P2ECCCTRED1	P1ECCCTRED1	P0ECCCTRED1
ICECCCTRED1H	$FFFF601B_H$	0	0	0	RFECCCTRED1	0	0	0	0
ICECCFREDL	$FFFF601C_H$	MKECCFRED	0	0	0	P3ECCFRED	P2ECCFRED	P1ECCFRED	P0ECCFRED
ICECCFREDH	$FFFF601D_H$	0	0	0	RFECCFRED	0	0	0	0
ICECCCNEDL	$FFFF601E_H$	MKECCCNED	0	0	0	P3ECCCNED	P2ECCCNED	P1ECCCNED	P0ECCCNED
ICECCCNEDH	$FFFF601F_H$	0	0	0	RFECCCNED	0	0	0	0
ICLVI0L	$FFFF6020_H$	MKLVI0	0	0	0	P3LVI0	P2LVI0	P1LVI0	P0LVI0
ICLVI0H	$FFFF6021_H$	0	0	0	RFLVI0	0	0	0	0
ICP0L	$FFFF6022_H$	MKP0	0	0	0	P3P0	P2P0	P1P0	P0P0
ICP0H	$FFFF6023_H$	0	0	0	RFP0	0	0	0	0
ICP1L	$FFFF6024_H$	MKP1	0	0	0	P3P1	P2P1	P1P1	P0P1
ICP1H	$FFFF6025_H$	0	0	0	RFP1	0	0	0	0
ICP2L	$FFFF6026_H$	MKP2	0	0	0	P3P2	P2P2	P1P2	P0P2
ICP2H	$FFFF6027_H$	0	0	0	RFP2	0	0	0	0
ICP3L	$FFFF6028_H$	MKP3	0	0	0	P3P3	P2P3	P1P3	P0P3
ICP3H	$FFFF6029_H$	0	0	0	RFP3	0	0	0	0
ICP4L	$FFFF602A_H$	MKP4	0	0	0	P3P4	P2P4	P1P4	P0P4
ICP4H	$FFFF602B_H$	0	0	0	RFP4	0	0	0	0
ICP5L	$FFFF602C_H$	MKP5	0	0	0	P3P5	P2P5	P1P5	P0P5
ICP5H	$FFFF602D_H$	0	0	0	RFP5	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (2/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICP6L	FFFF602E$_H$	MKP6	0	0	0	P3P6	P2P6	P1P6	P0P6
ICP6H	FFFF602F$_H$	0	0	0	RFP6	0	0	0	0
ICP7L	FFFF6030$_H$	MKP7	0	0	0	P3P7	P2P7	P1P7	P0P7
ICP7H	FFFF6031$_H$	0	0	0	RFP7	0	0	0	0
ICP8L	FFFF6032$_H$	MKP8	0	0	0	P3P8	P2P8	P1P8	P0P8
ICP8H	FFFF6033$_H$	0	0	0	RFP8	0	0	0	0
ICP9L	FFFF6034$_H$	MKP9	0	0	0	P3P9	P2P9	P1P9	P0P9
ICP9H	FFFF6035$_H$	0	0	0	RFP9	0	0	0	0
ICP10L	FFFF6036$_H$	MKP10	0	0	0	P3P10	P2P10	P1P10	P0P10
ICP10H	FFFF6037$_H$	0	0	0	RFP10	0	0	0	0
ICDMA0L	FFFF6038$_H$	MKDMA0	0	0	0	P3DMA0	P2DMA0	P1DMA0	P0DMA0
ICDMA0H	FFFF6039$_H$	0	0	0	RFDMA0	0	0	0	0
ICDMACT0L	FFFF603A$_H$	MKDMACT0	0	0	0	P3DMACT0	P2DMACT0	P1DMACT0	P0DMACT0
ICDMACT0H	FFFF603B$_H$	0	0	0	RFDMACT0	0	0	0	0
ICDMA1L	FFFF603C$_H$	MKDMA1	0	0	0	P3DMA1	P2DMA1	P1DMA1	P0DMA1
ICDMA1H	FFFF603D$_H$	0	0	0	RFDMA1	0	0	0	0
ICDMACT1L	FFFF603E$_H$	MKDMACT1	0	0	0	P3DMACT1	P2DMACT1	P1DMACT1	P0DMACT1
ICDMACT1H	FFFF603F$_H$	0	0	0	RFDMACT1	0	0	0	0
ICDMA2L	FFFF6040$_H$	MKDMA2	0	0	0	P3DMA2	P2DMA2	P1DMA2	P0DMA2
ICDMA2H	FFFF6041$_H$	0	0	0	RFDMA2	0	0	0	0
ICDMACT2L	FFFF6042$_H$	MKDMACT2	0	0	0	P3DMACT2	P2DMACT2	P1DMACT2	P0DMACT2
ICDMACT2H	FFFF6043$_H$	0	0	0	RFDMACT2	0	0	0	0
ICDMA3L	FFFF6044$_H$	MKDMA3	0	0	0	P3DMA3	P2DMA3	P1DMA3	P0DMA3
ICDMA3H	FFFF6045$_H$	0	0	0	RFDMA3	0	0	0	0
ICDMACT3L	FFFF6046$_H$	MKDMACT3	0	0	0	P3DMACT3	P2DMACT3	P1DMACT3	P0DMACT3
ICDMACT3H	FFFF6047$_H$	0	0	0	RFDMACT3	0	0	0	0
ICDMA4L	FFFF6048$_H$	MKDMA4	0	0	0	P3DMA4	P2DMA4	P1DMA4	P0DMA4
ICDMA4H	FFFF6049$_H$	0	0	0	RFDMA4	0	0	0	0
ICDMACT4L	FFFF604A$_H$	MKDMACT4	0	0	0	P3DMACT4	P2DMACT4	P1DMACT4	P0DMACT4
ICDMACT4H	FFFF604B$_H$	0	0	0	RFDMACT4	0	0	0	0
ICDMA5L	FFFF604C$_H$	MKDMA5	0	0	0	P3DMA5	P2DMA5	P1DMA5	P0DMA5
ICDMA5H	FFFF604D$_H$	0	0	0	RFDMA5	0	0	0	0
ICDMACT5L	FFFF604E$_H$	MKDMACT5	0	0	0	P3DMACT5	P2DMACT5	P1DMACT5	P0DMACT5
ICDMACT5H	FFFF604F$_H$	0	0	0	RFDMACT5	0	0	0	0
ICDMA6L	FFFF6050$_H$	MKDMA6	0	0	0	P3DMA6	P2DMA6	P1DMA6	P0DMA6
ICDMA6H	FFFF6051$_H$	0	0	0	RFDMA6	0	0	0	0
ICDMACT6L	FFFF6052$_H$	MKDMACT6	0	0	0	P3DMACT6	P2DMACT6	P1DMACT6	P0DMACT6
ICDMACT6H	FFFF6053$_H$	0	0	0	RFDMACT6	0	0	0	0
ICDMA7L	FFFF6054$_H$	MKDMA7	0	0	0	P3DMA7	P2DMA7	P1DMA7	P0DMA7
ICDMA7H	FFFF6055$_H$	0	0	0	RFDMA7	0	0	0	0
ICDMACT7L	FFFF6056$_H$	MKDMACT7	0	0	0	P3DMACT7	P2DMACT7	P1DMACT7	P0DMACT7
ICDMACT7H	FFFF6057$_H$	0	0	0	RFDMACT7	0	0	0	0
ICOSTM0L	FFFF6058$_H$	MKOSTM0	0	0	0	P3OSTM0	P2OSTM0	P1OSTM0	P0OSTM0
ICOSTM0H	FFFF6059$_H$	0	0	0	RFOSTM0	0	0	0	0
ICOSTM1L	FFFF605A$_H$	MKOSTM1	0	0	0	P3OSTM1	P2OSTM1	P1OSTM1	P0OSTM1
ICOSTM1H	FFFF605B$_H$	0	0	0	RFOSTM1	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (3/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICTAUJ0I0L	FFFF605C$_H$	MKTAUJ0I0	0	0	0	P3TAUJ0I0	P2TAUJ0I0	P1TAUJ0I0	P0TAUJ0I0
ICTAUJ0I0H	FFFF605D$_H$	0	0	0	RFTAUJ0I0	0	0	0	0
ICTAUJ0I1L	FFFF605E$_H$	MKTAUJ0I1	0	0	0	P3TAUJ0I1	P2TAUJ0I1	P1TAUJ0I1	P0TAUJ0I1
ICTAUJ0I1H	FFFF605F$_H$	0	0	0	RFTAUJ0I1	0	0	0	0
ICTAUJ0I2L	FFFF6060$_H$	MKTAUJ0I2	0	0	0	P3TAUJ0I2	P2TAUJ0I2	P1TAUJ0I2	P0TAUJ0I2
ICTAUJ0I2H	FFFF6061$_H$	0	0	0	RFTAUJ0I2	0	0	0	0
ICTAUJ0I3L	FFFF6062$_H$	MKTAUJ0I3	0	0	0	P3TAUJ0I3	P2TAUJ0I3	P1TAUJ0I3	P0TAUJ0I3
ICTAUJ0I3H	FFFF6063$_H$	0	0	0	RFTAUJ0I3	0	0	0	0
ICTAUJ1I0L	FFFF6064$_H$	MKTAUJ1I0	0	0	0	P3TAUJ1I0	P2TAUJ1I0	P1TAUJ1I0	P0TAUJ1I0
ICTAUJ1I0H	FFFF6065$_H$	0	0	0	RFTAUJ1I0	0	0	0	0
ICTAUJ1I1L	FFFF6066$_H$	MKTAUJ1I1	0	0	0	P3TAUJ1I1	P2TAUJ1I1	P1TAUJ1I1	P0TAUJ1I1
ICTAUJ1I1H	FFFF6067$_H$	0	0	0	RFTAUJ1I1	0	0	0	0
ICTAUJ1I2L	FFFF6068$_H$	MKTAUJ1I2	0	0	0	P3TAUJ1I2	P2TAUJ1I2	P1TAUJ1I2	P0TAUJ1I2
ICTAUJ1I2H	FFFF6069$_H$	0	0	0	RFTAUJ1I2	0	0	0	0
ICTAUJ1I3L	FFFF606A$_H$	MKTAUJ1I3	0	0	0	P3TAUJ1I3	P2TAUJ1I3	P1TAUJ1I3	P0TAUJ1I3
ICTAUJ1I3H	FFFF606B$_H$	0	0	0	RFTAUJ1I3	0	0	0	0
ICTAUA0I0L	FFFF606C$_H$	MKTAUA0I0	0	0	0	P3TAUA0I0	P2TAUA0I0	P1TAUA0I0	P0TAUA0I0
ICTAUA0I0H	FFFF606D$_H$	0	0	0	RFTAUA0I0	0	0	0	0
ICTAUA0I1L	FFFF606E$_H$	MKTAUA0I1	0	0	0	P3TAUA0I1	P2TAUA0I1	P1TAUA0I1	P0TAUA0I1
ICTAUA0I1H	FFFF606F$_H$	0	0	0	RFTAUA0I1	0	0	0	0
ICTAUA0I2L	FFFF6070$_H$	MKTAUA0I2	0	0	0	P3TAUA0I2	P2TAUA0I2	P1TAUA0I2	P0TAUA0I2
ICTAUA0I2H	FFFF6071$_H$	0	0	0	RFTAUA0I2	0	0	0	0
ICTAUA0I3L	FFFF6072$_H$	MKTAUA0I3	0	0	0	P3TAUA0I3	P2TAUA0I3	P1TAUA0I3	P0TAUA0I3
ICTAUA0I3H	FFFF6073$_H$	0	0	0	RFTAUA0I3	0	0	0	0
ICTAUA0I4L	FFFF6074$_H$	MKTAUA0I4	0	0	0	P3TAUA0I4	P2TAUA0I4	P1TAUA0I4	P0TAUA0I4
ICTAUA0I4H	FFFF6075$_H$	0	0	0	RFTAUA0I4	0	0	0	0
ICTAUA0I5L	FFFF6076$_H$	MKTAUA0I5	0	0	0	P3TAUA0I5	P2TAUA0I5	P1TAUA0I5	P0TAUA0I5
ICTAUA0I5H	FFFF6077$_H$	0	0	0	RFTAUA0I5	0	0	0	0
ICTAUA0I6L	FFFF6078$_H$	MKTAUA0I6	0	0	0	P3TAUA0I6	P2TAUA0I6	P1TAUA0I6	P0TAUA0I6
ICTAUA0I6H	FFFF6079$_H$	0	0	0	RFTAUA0I6	0	0	0	0
ICTAUA0I7L	FFFF607A$_H$	MKTAUA0I7	0	0	0	P3TAUA0I7	P2TAUA0I7	P1TAUA0I7	P0TAUA0I7
ICTAUA0I7H	FFFF607B$_H$	0	0	0	RFTAUA0I7	0	0	0	0
ICTAUA0I8L	FFFF607C$_H$	MKTAUA0I8	0	0	0	P3TAUA0I8	P2TAUA0I8	P1TAUA0I8	P0TAUA0I8
ICTAUA0I8H	FFFF607D$_H$	0	0	0	RFTAUA0I8	0	0	0	0
ICTAUA0I9L	FFFF607E$_H$	MKTAUA0I9	0	0	0	P3TAUA0I9	P2TAUA0I9	P1TAUA0I9	P0TAUA0I9
ICTAUA0I9H	FFFF607F$_H$	0	0	0	RFTAUA0I9	0	0	0	0
ICTAUA0I10L	FFFF6080$_H$	MKTAUA0I10	0	0	0	P3TAUA0I10	P2TAUA0I10	P1TAUA0I10	P0TAUA0I10
ICTAUA0I10H	FFFF6081$_H$	0	0	0	RFTAUA0I10	0	0	0	0
ICTAUA0I11L	FFFF6082$_H$	MKTAUA0I11	0	0	0	P3TAUA0I11	P2TAUA0I11	P1TAUA0I11	P0TAUA0I11
ICTAUA0I11H	FFFF6083$_H$	0	0	0	RFTAUA0I11	0	0	0	0
ICTAUA0I12L	FFFF6084$_H$	MKTAUA0I12	0	0	0	P3TAUA0I12	P2TAUA0I12	P1TAUA0I12	P0TAUA0I12
ICTAUA0I12H	FFFF6085$_H$	0	0	0	RFTAUA0I12	0	0	0	0
ICTAUA0I13L	FFFF6086$_H$	MKTAUA0I13	0	0	0	P3TAUA0I13	P2TAUA0I13	P1TAUA0I13	P0TAUA0I13
ICTAUA0I13H	FFFF6087$_H$	0	0	0	RFTAUA0I13	0	0	0	0
ICTAUA0I14L	FFFF6088$_H$	MKTAUA0I14	0	0	0	P3TAUA0I14	P2TAUA0I14	P1TAUA0I14	P0TAUA0I14
ICTAUA0I14H	FFFF6089$_H$	0	0	0	RFTAUA0I14	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (4/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICTAUA0I15L	FFFF608A$_H$	MKTAUA0I15	0	0	0	P3TAUA0I15	P2TAUA0I15	P1TAUA0I15	P0TAUA0I15
ICTAUA0I15H	FFFF608B$_H$	0	0	0	RFTAUA0I15	0	0	0	0
ICTAUA1I0L	FFFF608C$_H$	MKTAUA1I0	0	0	0	P3TAUA1I0	P2TAUA1I0	P1TAUA1I0	P0TAUA1I0
ICTAUA1I0H	FFFF608D$_H$	0	0	0	RFTAUA1I0	0	0	0	0
ICTAUA1I1L	FFFF608E$_H$	MKTAUA1I1	0	0	0	P3TAUA1I1	P2TAUA1I1	P1TAUA1I1	P0TAUA1I1
ICTAUA1I1H	FFFF608F$_H$	0	0	0	RFTAUA1I1	0	0	0	0
ICTAUA1I2L	FFFF6090$_H$	MKTAUA1I2	0	0	0	P3TAUA1I2	P2TAUA1I2	P1TAUA1I2	P0TAUA1I2
ICTAUA1I2H	FFFF6091$_H$	0	0	0	RFTAUA1I2	0	0	0	0
ICTAUA1I3L	FFFF6092$_H$	MKTAUA1I3	0	0	0	P3TAUA1I3	P2TAUA1I3	P1TAUA1I3	P0TAUA1I3
ICTAUA1I3H	FFFF6093$_H$	0	0	0	RFTAUA1I3	0	0	0	0
ICTAUA1I4L	FFFF6094$_H$	MKTAUA1I4	0	0	0	P3TAUA1I4	P2TAUA1I4	P1TAUA1I4	P0TAUA1I4
ICTAUA1I4H	FFFF6095$_H$	0	0	0	RFTAUA1I4	0	0	0	0
ICTAUA1I5L	FFFF6096$_H$	MKTAUA1I5	0	0	0	P3TAUA1I5	P2TAUA1I5	P1TAUA1I5	P0TAUA1I5
ICTAUA1I5H	FFFF6097$_H$	0	0	0	RFTAUA1I5	0	0	0	0
ICTAUA1I6L	FFFF6098$_H$	MKTAUA1I6	0	0	0	P3TAUA1I6	P2TAUA1I6	P1TAUA1I6	P0TAUA1I6
ICTAUA1I6H	FFFF6099$_H$	0	0	0	RFTAUA1I6	0	0	0	0
ICTAUA1I7L	FFFF609A$_H$	MKTAUA1I7	0	0	0	P3TAUA1I7	P2TAUA1I7	P1TAUA1I7	P0TAUA1I7
ICTAUA1I7H	FFFF609B$_H$	0	0	0	RFTAUA1I7	0	0	0	0
ICTAUA1I8L	FFFF609C$_H$	MKTAUA1I8	0	0	0	P3TAUA1I8	P2TAUA1I8	P1TAUA1I8	P0TAUA1I8
ICTAUA1I8H	FFFF609D$_H$	0	0	0	RFTAUA1I8	0	0	0	0
ICTAUA1I9L	FFFF609E$_H$	MKTAUA1I9	0	0	0	P3TAUA1I9	P2TAUA1I9	P1TAUA1I9	P0TAUA1I9
ICTAUA1I9H	FFFF609F$_H$	0	0	0	RFTAUA1I9	0	0	0	0
ICTAUA1I10L	FFFF60A0$_H$	MKTAUA1I10	0	0	0	P3TAUA1I10	P2TAUA1I10	P1TAUA1I10	P0TAUA1I10
ICTAUA1I10H	FFFF60A1$_H$	0	0	0	RFTAUA1I10	0	0	0	0
ICTAUA1I11L	FFFF60A2$_H$	MKTAUA1I11	0	0	0	P3TAUA1I11	P2TAUA1I11	P1TAUA1I11	P0TAUA1I11
ICTAUA1I11H	FFFF60A3$_H$	0	0	0	RFTAUA1I11	0	0	0	0
ICTAUA1I12L	FFFF60A4$_H$	MKTAUA1I12	0	0	0	P3TAUA1I12	P2TAUA1I12	P1TAUA1I12	P0TAUA1I12
ICTAUA1I12H	FFFF60A5$_H$	0	0	0	RFTAUA1I12	0	0	0	0
ICTAUA1I13L	FFFF60A6$_H$	MKTAUA1I13	0	0	0	P3TAUA1I13	P2TAUA1I13	P1TAUA1I13	P0TAUA1I13
ICTAUA1I13H	FFFF60A7$_H$	0	0	0	RFTAUA1I13	0	0	0	0
ICTAUA1I14L	FFFF60A8$_H$	MKTAUA1I14	0	0	0	P3TAUA1I14	P2TAUA1I14	P1TAUA1I14	P0TAUA1I14
ICTAUA1I14H	FFFF60A9$_H$	0	0	0	RFTAUA1I14	0	0	0	0
ICTAUA1I15L	FFFF60AA$_H$	MKTAUA1I15	0	0	0	P3TAUA1I15	P2TAUA1I15	P1TAUA1I15	P0TAUA1I15
ICTAUA1I15H	FFFF60AB$_H$	0	0	0	RFTAUA1I15	0	0	0	0
ICADCA0ERRL	FFFF60AC$_H$	MKADCA0ERR	0	0	0	P3ADCA0ERR	P2ADCA0ERR	P1ADCA0ERR	P0ADCA0ERR
ICADCA0ERRH	FFFF60AD$_H$	0	0	0	RFADCA0ERR	0	0	0	0
ICADCA0I0L	FFFF60AE$_H$	MKADCA0I0	0	0	0	P3ADCA0I0	P2ADCA0I0	P1ADCA0I0	P0ADCA0I0
ICADCA0I0H	FFFF60AF$_H$	0	0	0	RFADCA0I0	0	0	0	0
ICADCA0I1L	FFFF60B0$_H$	MKADCA0I1	0	0	0	P3ADCA0I1	P2ADCA0I1	P1ADCA0I1	P0ADCA0I1
ICADCA0I1H	FFFF60B1$_H$	0	0	0	RFADCA0I1	0	0	0	0
ICADCA0I2L	FFFF60B2$_H$	MKADCA0I2	0	0	0	P3ADCA0I2	P2ADCA0I2	P1ADCA0I2	P0ADCA0I2
ICADCA0I2H	FFFF60B3$_H$	0	0	0	RFADCA0I2	0	0	0	0
ICADCA0LLTL	FFFF60B4$_H$	MKADCA0LLT	0	0	0	P3ADCA0LLT	P2ADCA0LLT	P1ADCA0LLT	P0ADCA0LLT
ICADCA0LLTH	FFFF60B5$_H$	0	0	0	RFADCA0LLT	0	0	0	0
ICCSIH0IREL	FFFF60B6$_H$	MKCSIH0IRE	0	0	0	P3CSIH0IRE	P2CSIH0IRE	P1CSIH0IRE	P0CSIH0IRE
ICCSIH0IREH	FFFF60B7$_H$	0	0	0	RFCSIH0IRE	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (5/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICCSIH0IRL	FFFF60B8$_H$	MKCSIH0IR	0	0	0	P3CSIH0IR	P2CSIH0IR	P1CSIH0IR	P0CSIH0IR
ICCSIH0IRH	FFFF60B9$_H$	0	0	0	RFCSIH0IR	0	0	0	0
ICCSIH0ICL	FFFF60BA$_H$	MKCSIH0IC	0	0	0	P3CSIH0IC	P2CSIH0IC	P1CSIH0IC	P0CSIH0IC
ICCSIH0ICH	FFFF60BB$_H$	0	0	0	RFCSIH0IC	0	0	0	0
ICCSIH0IJCL	FFFF60BC$_H$	MKCSIH0IJC	0	0	0	P3CSIH0IJC	P2CSIH0IJC	P1CSIH0IJC	P0CSIH0IJC
ICCSIH0IJCH	FFFF60BD$_H$	0	0	0	RFCSIH0IJC	0	0	0	0
ICCSIH1IREL	FFFF60BE$_H$	MKCSIH1IRE	0	0	0	P3CSIH1IRE	P2CSIH1IRE	P1CSIH1IRE	P0CSIH1IRE
ICCSIH1IREH	FFFF60BF$_H$	0	0	0	RFCSIH1IRE	0	0	0	0
ICCSIH1IRL	FFFF60C0$_H$	MKCSIH1IR	0	0	0	P3CSIH1IR	P2CSIH1IR	P1CSIH1IR	P0CSIH1IR
ICCSIH1IRH	FFFF60C1$_H$	0	0	0	RFCSIH1IR	0	0	0	0
ICCSIH1ICL	FFFF60C2$_H$	MKCSIH1IC	0	0	0	P3CSIH1IC	P2CSIH1IC	P1CSIH1IC	P0CSIH1IC
ICCSIH1ICH	FFFF60C3$_H$	0	0	0	RFCSIH1IC	0	0	0	0
ICCSIH1IJCL	FFFF60C4$_H$	MKCSIH1IJC	0	0	0	P3CSIH1IJC	P2CSIH1IJC	P1CSIH1IJC	P0CSIH1IJC
ICCSIH1IJCH	FFFF60C5$_H$	0	0	0	RFCSIH1IJC	0	0	0	0
ICCSIG0IREL	FFFF60C6$_H$	MKCSIG0IRE	0	0	0	P3CSIG0IRE	P2CSIG0IRE	P1CSIG0IRE	P0CSIG0IRE
ICCSIG0IREH	FFFF60C7$_H$	0	0	0	RFCSIG0IRE	0	0	0	0
ICCSIG0IRL	FFFF60C8$_H$	MKCSIG0IR	0	0	0	P3CSIG0IR	P2CSIG0IR	P1CSIG0IR	P0CSIG0IR
ICCSIG0IRH	FFFF60C9$_H$	0	0	0	RFCSIG0IR	0	0	0	0
ICCSIG0ICL	FFFF60CA$_H$	MKCSIG0IC	0	0	0	P3CSIG0IC	P2CSIG0IC	P1CSIG0IC	P0CSIG0IC
ICCSIG0ICH	FFFF60CB$_H$	0	0	0	RFCSIG0IC	0	0	0	0
ICCSIG1IREL	FFFF60CC$_H$	MKCSIG1IRE	0	0	0	P3CSIG1IRE	P2CSIG1IRE	P1CSIG1IRE	P0CSIG1IRE
ICCSIG1IREH	FFFF60CD$_H$	0	0	0	RFCSIG1IRE	0	0	0	0
ICCSIG1IRL	FFFF60CE$_H$	MKCSIG1IR	0	0	0	P3CSIG1IR	P2CSIG1IR	P1CSIG1IR	P0CSIG1IR
ICCSIG1IRH	FFFF60CF$_H$	0	0	0	RFCSIG1IR	0	0	0	0
ICCSIG1ICL	FFFF60D0$_H$	MKCSIG1IC	0	0	0	P3CSIG1IC	P2CSIG1IC	P1CSIG1IC	P0CSIG1IC
ICCSIG1ICH	FFFF60D1$_H$	0	0	0	RFCSIG1IC	0	0	0	0
ICCSIG2IREL	FFFF60D2$_H$	MKCSIG2IRE	0	0	0	P3CSIG2IRE	P2CSIG2IRE	P1CSIG2IRE	P0CSIG2IRE
ICCSIG2IREH	FFFF60D3$_H$	0	0	0	RFCSIG2IRE	0	0	0	0
ICCSIG2IRL	FFFF60D4$_H$	MKCSIG2IR	0	0	0	P3CSIG2IR	P2CSIG2IR	P1CSIG2IR	P0CSIG2IR
ICCSIG2IRH	FFFF60D5$_H$	0	0	0	RFCSIG2IR	0	0	0	0
ICCSIG2ICL	FFFF60D6$_H$	MKCSIG2IC	0	0	0	P3CSIG2IC	P2CSIG2IC	P1CSIG2IC	P0CSIG2IC
ICCSIG2ICH	FFFF60D7$_H$	0	0	0	RFCSIG2IC	0	0	0	0
ICURTH0ISL	FFFF60D8$_H$	MKURTH0IS	0	0	0	P3URTH0IS	P2URTH0IS	P1URTH0IS	P0URTH0IS
ICURTH0ISH	FFFF60D9$_H$	0	0	0	RFURTH0IS	0	0	0	0
ICURTH0IRL	FFFF60DA$_H$	MKURTH0IR	0	0	0	P3URTH0IR	P2URTH0IR	P1URTH0IR	P0URTH0IR
ICURTH0IRH	FFFF60DB$_H$	0	0	0	RFURTH0IR	0	0	0	0
ICURTH0ITL	FFFF60DC$_H$	MKURTH0IT	0	0	0	P3URTH0IT	P2URTH0IT	P1URTH0IT	P0URTH0IT
ICURTH0ITH	FFFF60DD$_H$	0	0	0	RFURTH0IT	0	0	0	0
ICURTH1ISL	FFFF60DE$_H$	MKURTH1IS	0	0	0	P3URTH1IS	P2URTH1IS	P1URTH1IS	P0URTH1IS
ICURTH1ISH	FFFF60DF$_H$	0	0	0	RFURTH1IS	0	0	0	0
ICURTH1IRL	FFFF60E0$_H$	MKURTH1IR	0	0	0	P3URTH1IR	P2URTH1IR	P1URTH1IR	P0URTH1IR
ICURTH1IRH	FFFF60E1$_H$	0	0	0	RFURTH1IR	0	0	0	0
ICURTH1ITL	FFFF60E2$_H$	MKURTH1IT	0	0	0	P3URTH1IT	P2URTH1IT	P1URTH1IT	P0URTH1IT
ICURTH1ITH	FFFF60E3$_H$	0	0	0	RFURTH1IT	0	0	0	0
ICURTH2ISL	FFFF60E4$_H$	MKURTH2IS	0	0	0	P3URTH2IS	P2URTH2IS	P1URTH2IS	P0URTH2IS
ICURTH2ISH	FFFF60E5$_H$	0	0	0	RFURTH2IS	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (6/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICURTH2IRL	FFFF60E6$_H$	MKURTH2IR	0	0	0	P3URTH2IR	P2URTH2IR	P1URTH2IR	P0URTH2IR
ICURTH2IRH	FFFF60E7$_H$	0	0	0	RFURTH2IR	0	0	0	0
ICURTH2ITL	FFFF60E8$_H$	MKURTH2IT	0	0	0	P3URTH2IT	P2URTH2IT	P1URTH2IT	P0URTH2IT
ICURTH2ITH	FFFF60E9$_H$	0	0	0	RFURTH2IT	0	0	0	0
ICFCN0ERRL	FFFF60EA$_H$	MKFCN0ERR	0	0	0	P3FCN0ERR	P2FCN0ERR	P1FCN0ERR	P0FCN0ERR
ICFCN0ERRH	FFFF60EB$_H$	0	0	0	RFFCN0ERR	0	0	0	0
ICFCN0WUPL	FFFF60EC$_H$	MKFCN0WUP	0	0	0	P3FCN0WUP	P2FCN0WUP	P1FCN0WUP	P0FCN0WUP
ICFCN0WUPH	FFFF60ED$_H$	0	0	0	RFFCN0WUP	0	0	0	0
ICFCN0RECL	FFFF60EE$_H$	MKFCN0REC	0	0	0	P3FCN0REC	P2FCN0REC	P1FCN0REC	P0FCN0REC
ICFCN0RECH	FFFF60EF$_H$	0	0	0	RFFCN0REC	0	0	0	0
ICFCN0TRXL	FFFF60F0$_H$	MKFCN0TRX	0	0	0	P3FCN0TRX	P2FCN0TRX	P1FCN0TRX	P0FCN0TRX
ICFCN0TRXH	FFFF60F1$_H$	0	0	0	RFFCN0TRX	0	0	0	0
ICFCN1ERRL	FFFF60F2$_H$	MKFCN1ERR	0	0	0	P3FCN1ERR	P2FCN1ERR	P1FCN1ERR	P0FCN1ERR
ICFCN1ERRH	FFFF60F3$_H$	0	0	0	RFFCN1ERR	0	0	0	0
ICFCN1WUPL	FFFF60F4$_H$	MKFCN1WUP	0	0	0	P3FCN1WUP	P2FCN1WUP	P1FCN1WUP	P0FCN1WUP
ICFCN1WUPH	FFFF60F5$_H$	0	0	0	RFFCN1WUP	0	0	0	0
ICFCN1RECL	FFFF60F6$_H$	MKFCN1REC	0	0	0	P3FCN1REC	P2FCN1REC	P1FCN1REC	P0FCN1REC
ICFCN1RECH	FFFF60F7$_H$	0	0	0	RFFCN1REC	0	0	0	0
ICFCN1TRXL	FFFF60F8$_H$	MKFCN1TRX	0	0	0	P3FCN1TRX	P2FCN1TRX	P1FCN1TRX	P0FCN1TRX
ICFCN1TRXH	FFFF60F9$_H$	0	0	0	RFFCN1TRX	0	0	0	0
ICBRG0L	FFFF60FA$_H$	MKBRG0	0	0	0	P3BRG0	P2BRG0	P1BRG0	P0BRG0
ICBRG0H	FFFF60FB$_H$	0	0	0	RFBRG0	0	0	0	0
ICFLL	FFFF60FE$_H$	MKFL	0	0	0	P3FL	P2FL	P1FL	P0FL
ICFLH	FFFF60FF$_H$	0	0	0	RFFL	0	0	0	0
ICSWNL	FFFF6100$_H$	MKSWN	0	0	0	P3SWN	P2SWN	P1SWN	P0SWN
ICSWNH	FFFF6101$_H$	0	0	0	RFSWN	0	0	0	0
ICTSG20I00L	FFFF6102$_H$	MKTSG20I00	0	0	0	P3TSG20I00	P2TSG20I00	P1TSG20I00	P0TSG20I00
ICTSG20I00H	FFFF6103$_H$	0	0	0	RFTSG20I00	0	0	0	0
ICTSG20I01L	FFFF6104$_H$	MKTSG20I01	0	0	0	P3TSG20I01	P2TSG20I01	P1TSG20I01	P0TSG20I01
ICTSG20I01H	FFFF6105$_H$	0	0	0	RFTSG20I01	0	0	0	0
ICTSG20I02L	FFFF6106$_H$	MKTSG20I02	0	0	0	P3TSG20I02	P2TSG20I02	P1TSG20I02	P0TSG20I02
ICTSG20I02H	FFFF6107$_H$	0	0	0	RFTSG20I02	0	0	0	0
ICTSG20I03L	FFFF6108$_H$	MKTSG20I03	0	0	0	P3TSG20I03	P2TSG20I03	P1TSG20I03	P0TSG20I03
ICTSG20I03H	FFFF6109$_H$	0	0	0	RFTSG20I03	0	0	0	0
ICTSG20I04L	FFFF610A$_H$	MKTSG20I04	0	0	0	P3TSG20I04	P2TSG20I04	P1TSG20I04	P0TSG20I04
ICTSG20I04H	FFFF610B$_H$	0	0	0	RFTSG20I04	0	0	0	0
ICTSG20I05L	FFFF610C$_H$	MKTSG20I05	0	0	0	P3TSG20I05	P2TSG20I05	P1TSG20I05	P0TSG20I05
ICTSG20I05H	FFFF610D$_H$	0	0	0	RFTSG20I05	0	0	0	0
ICTSG20I06L	FFFF610E$_H$	MKTSG20I06	0	0	0	P3TSG20I06	P2TSG20I06	P1TSG20I06	P0TSG20I06
ICTSG20I06H	FFFF610F$_H$	0	0	0	RFTSG20I06	0	0	0	0
ICTSG20I07L	FFFF6110$_H$	MKTSG20I07	0	0	0	P3TSG20I07	P2TSG20I07	P1TSG20I07	P0TSG20I07
ICTSG20I07H	FFFF6111$_H$	0	0	0	RFTSG20I07	0	0	0	0
ICTSG20I08L	FFFF6112$_H$	MKTSG20I08	0	0	0	P3TSG20I08	P2TSG20I08	P1TSG20I08	P0TSG20I08
ICTSG20I08H	FFFF6113$_H$	0	0	0	RFTSG20I08	0	0	0	0
ICTSG20I09L	FFFF6114$_H$	MKTSG20I09	0	0	0	P3TSG20I09	P2TSG20I09	P1TSG20I09	P0TSG20I09
ICTSG20I09H	FFFF6115$_H$	0	0	0	RFTSG20I09	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (7/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICTSG20I10L	$FFFF6116_H$	MKTSG20I10	0	0	0	P3TSG20I10	P2TSG20I10	P1TSG20I10	P0TSG20I10
ICTSG20I10H	$FFFF6117_H$	0	0	0	RFTSG20I10	0	0	0	0
ICTSG20I11L	$FFFF6118_H$	MKTSG20I11	0	0	0	P3TSG20I11	P2TSG20I11	P1TSG20I11	P0TSG20I11
ICTSG20I11H	$FFFF6119_H$	0	0	0	RFTSG20I11	0	0	0	0
ICTSG20I12L	$FFFF611A_H$	MKTSG20I12	0	0	0	P3TSG20I12	P2TSG20I12	P1TSG20I12	P0TSG20I12
ICTSG20I12H	$FFFF611B_H$	0	0	0	RFTSG20I12	0	0	0	0
ICTSG20IPEKL	$FFFF611C_H$	MKTSG20IPEK	0	0	0	P3TSG20IPEK	P2TSG20IPEK	P1TSG20IPEK	P0TSG20IPEK
ICTSG20IPEKH	$FFFF611D_H$	0	0	0	RFTSG20IPEK	0	0	0	0
ICTSG20IVLYL	$FFFF611E_H$	MKTSG20IVLY	0	0	0	P3TSG20IVLY	P2TSG20IVLY	P1TSG20IVLY	P0TSG20IVLY
ICTSG20IVLYH	$FFFF611F_H$	0	0	0	RFTSG20IVLY	0	0	0	0
ICTSG20IERL	$FFFF6120_H$	MKTSG20IER	0	0	0	P3TSG20IER	P2TSG20IER	P1TSG20IER	P0TSG20IER
ICTSG20IERH	$FFFF6121_H$	0	0	0	RFTSG20IER	0	0	0	0
ICTSG20IWNL	$FFFF6122_H$	MKTSG20IWN	0	0	0	P3TSG20IWN	P2TSG20IWN	P1TSG20IWN	P0TSG20IWN
ICTSG20IWNH	$FFFF6123_H$	0	0	0	RFTSG20IWN	0	0	0	0
ICTSG21I00L	$FFFF6124_H$	MKTSG21I00	0	0	0	P3TSG21I00	P2TSG21I00	P1TSG21I00	P0TSG21I00
ICTSG21I00H	$FFFF6125_H$	0	0	0	RFTSG21I00	0	0	0	0
ICTSG21I01L	$FFFF6126_H$	MKTSG21I01	0	0	0	P3TSG21I01	P2TSG21I01	P1TSG21I01	P0TSG21I01
ICTSG21I01H	$FFFF6127_H$	0	0	0	RFTSG21I01	0	0	0	0
ICTSG21I02L	$FFFF6128_H$	MKTSG21I02	0	0	0	P3TSG21I02	P2TSG21I02	P1TSG21I02	P0TSG21I02
ICTSG21I02H	$FFFF6129_H$	0	0	0	RFTSG21I02	0	0	0	0
ICTSG21I03L	$FFFF612A_H$	MKTSG21I03	0	0	0	P3TSG21I03	P2TSG21I03	P1TSG21I03	P0TSG21I03
ICTSG21I03H	$FFFF612B_H$	0	0	0	RFTSG21I03	0	0	0	0
ICTSG21I04L	$FFFF612C_H$	MKTSG21I04	0	0	0	P3TSG21I04	P2TSG21I04	P1TSG21I04	P0TSG21I04
ICTSG21I04H	$FFFF612D_H$	0	0	0	RFTSG21I04	0	0	0	0
ICTSG21I05L	$FFFF612E_H$	MKTSG21I05	0	0	0	P3TSG21I05	P2TSG21I05	P1TSG21I05	P0TSG21I05
ICTSG21I05H	$FFFF612F_H$	0	0	0	RFTSG21I05	0	0	0	0
ICTSG21I06L	$FFFF6130_H$	MKTSG21I06	0	0	0	P3TSG21I06	P2TSG21I06	P1TSG21I06	P0TSG21I06
ICTSG21I06H	$FFFF6131_H$	0	0	0	RFTSG21I06	0	0	0	0
ICTSG21I07L	$FFFF6132_H$	MKTSG21I07	0	0	0	P3TSG21I07	P2TSG21I07	P1TSG21I07	P0TSG21I07
ICTSG21I07H	$FFFF6133_H$	0	0	0	RFTSG21I07	0	0	0	0
ICTSG21I08L	$FFFF6134_H$	MKTSG21I08	0	0	0	P3TSG21I08	P2TSG21I08	P1TSG21I08	P0TSG21I08
ICTSG21I08H	$FFFF6135_H$	0	0	0	RFTSG21I08	0	0	0	0
ICTSG21I09L	$FFFF6136_H$	MKTSG21I09	0	0	0	P3TSG21I09	P2TSG21I09	P1TSG21I09	P0TSG21I09
ICTSG21I09H	$FFFF6137_H$	0	0	0	RFTSG21I09	0	0	0	0
ICTSG21I10L	$FFFF6138_H$	MKTSG21I10	0	0	0	P3TSG21I10	P2TSG21I10	P1TSG21I10	P0TSG21I10
ICTSG21I10H	$FFFF6139_H$	0	0	0	RFTSG21I10	0	0	0	0
ICTSG21I11L	$FFFF613A_H$	MKTSG21I11	0	0	0	P3TSG21I11	P2TSG21I11	P1TSG21I11	P0TSG21I11
ICTSG21I11H	$FFFF613B_H$	0	0	0	RFTSG21I11	0	0	0	0
ICTSG21I12L	$FFFF613C_H$	MKTSG21I12	0	0	0	P3TSG21I12	P2TSG21I12	P1TSG21I12	P0TSG21I12
ICTSG21I12H	$FFFF613D_H$	0	0	0	RFTSG21I12	0	0	0	0
ICTSG21IPEKL	$FFFF613E_H$	MKTSG21IPEK	0	0	0	P3TSG21IPEK	P2TSG21IPEK	P1TSG21IPEK	P0TSG21IPEK
ICTSG21IPEKH	$FFFF613F_H$	0	0	0	RFTSG21IPEK	0	0	0	0
ICTSG21IVLYL	$FFFF6140_H$	MKTSG21IVLY	0	0	0	P3TSG21IVLY	P2TSG21IVLY	P1TSG21IVLY	P0TSG21IVLY
ICTSG21IVLYH	$FFFF6141_H$	0	0	0	RFTSG21IVLY	0	0	0	0
ICTSG21IERL	$FFFF6142_H$	MKTSG21IER	0	0	0	P3TSG21IER	P2TSG21IER	P1TSG21IER	P0TSG21IER
ICTSG21IERH	$FFFF6143_H$	0	0	0	RFTSG21IER	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (8/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICTSG21IWNL	FFFF6144$_H$	MKTSG21IWN	0	0	0	P3TSG21IWN	P2TSG21IWN	P1TSG21IWN	P0TSG21IWN
ICTSG21IWNH	FFFF6145$_H$	0	0	0	RFTSG21IWN	0	0	0	0
ICENCA0IOVL	FFFF6146$_H$	MKENCA0IOV	0	0	0	P3ENCA0IOV	P2ENCA0IOV	P1ENCA0IOV	P0ENCA0IOV
ICENCA0IOVH	FFFF6147$_H$	0	0	0	RFENCA0IOV	0	0	0	0
ICENCA0I0L	FFFF6148$_H$	MKENCA0I0	0	0	0	P3ENCA0I0	P2ENCA0I0	P1ENCA0I0	P0ENCA0I0
ICENCA0I0H	FFFF6149$_H$	0	0	0	RFENCA0I0	0	0	0	0
ICENCA0I1L	FFFF614A$_H$	MKENCA0I1	0	0	0	P3ENCA0I1	P2ENCA0I1	P1ENCA0I1	P0ENCA0I1
ICENCA0I1H	FFFF614B$_H$	0	0	0	RFENCA0I1	0	0	0	0
ICENCA0IUDL	FFFF614C$_H$	MKENCA0IUD	0	0	0	P3ENCA0IUD	P2ENCA0IUD	P1ENCA0IUD	P0ENCA0IUD
ICENCA0IUDH	FFFF614D$_H$	0	0	0	RFENCA0IUD	0	0	0	0
ICENCA0IECL	FFFF614E$_H$	MKENCA0IEC	0	0	0	P3ENCA0IEC	P2ENCA0IEC	P1ENCA0IEC	P0ENCA0IEC
ICENCA0IECH	FFFF614F$_H$	0	0	0	RFENCA0IEC	0	0	0	0
ICENCA1IOVL	FFFF6150$_H$	MKENCA1IOV	0	0	0	P3ENCA1IOV	P2ENCA1IOV	P1ENCA1IOV	P0ENCA1IOV
ICENCA1IOVH	FFFF6151$_H$	0	0	0	RFENCA1IOV	0	0	0	0
ICENCA1I0L	FFFF6152$_H$	MKENCA1I0	0	0	0	P3ENCA1I0	P2ENCA1I0	P1ENCA1I0	P0ENCA1I0
ICENCA1I0H	FFFF6153$_H$	0	0	0	RFENCA1I0	0	0	0	0
ICENCA1I1L	FFFF6154$_H$	MKENCA1I1	0	0	0	P3ENCA1I1	P2ENCA1I1	P1ENCA1I1	P0ENCA1I1
ICENCA1I1H	FFFF6155$_H$	0	0	0	RFENCA1I1	0	0	0	0
ICENCA1IUDL	FFFF6156$_H$	MKENCA1IUD	0	0	0	P3ENCA1IUD	P2ENCA1IUD	P1ENCA1IUD	P0ENCA1IUD
ICENCA1IUDH	FFFF6157$_H$	0	0	0	RFENCA1IUD	0	0	0	0
ICENCA1IECL	FFFF6158$_H$	MKENCA1IEC	0	0	0	P3ENCA1IEC	P2ENCA1IEC	P1ENCA1IEC	P0ENCA1IEC
ICENCA1IECH	FFFF6159$_H$	0	0	0	RFENCA1IEC	0	0	0	0
ICTPBA0IPRDL	FFFF615A$_H$	MKTPBA0IPRD	0	0	0	P3TPBA0IPRD	P2TPBA0IPRD	P1TPBA0IPRD	P0TPBA0IPRD
ICTPBA0IPRDH	FFFF615B$_H$	0	0	0	RFTPBA0IPRD	0	0	0	0
ICTPBA0IDTYL	FFFF615C$_H$	MKTPBA0IDTY	0	0	0	P3TPBA0IDTY	P2TPBA0IDTY	P1TPBA0IDTY	P0TPBA0IDTY
ICTPBA0IDTYH	FFFF615D$_H$	0	0	0	RFTPBA0IDTY	0	0	0	0
ICTPBA0IPATL	FFFF615E$_H$	MKTPBA0IPAT	0	0	0	P3TPBA0IPAT	P2TPBA0IPAT	P1TPBA0IPAT	P0TPBA0IPAT
ICTPBA0IPATH	FFFF615F$_H$	0	0	0	RFTPBA0IPAT	0	0	0	0
ICTPBA1IPRDL	FFFF6160$_H$	MKTPBA1IPRD	0	0	0	P3TPBA1IPRD	P2TPBA1IPRD	P1TPBA1IPRD	P0TPBA1IPRD
ICTPBA1IPRDH	FFFF6161$_H$	0	0	0	RFTPBA1IPRD	0	0	0	0
ICTPBA1IDTYL	FFFF6162$_H$	MKTPBA1IDTY	0	0	0	P3TPBA1IDTY	P2TPBA1IDTY	P1TPBA1IDTY	P0TPBA1IDTY
ICTPBA1IDTYH	FFFF6163$_H$	0	0	0	RFTPBA1IDTY	0	0	0	0
ICTPBA1IPATL	FFFF6164$_H$	MKTPBA1IPAT	0	0	0	P3TPBA1IPAT	P2TPBA1IPAT	P1TPBA1IPAT	P0TPBA1IPAT
ICTPBA1IPATH	FFFF6165$_H$	0	0	0	RFTPBA1IPAT	0	0	0	0
ICTAPA0IPEK0L	FFFF616A$_H$	MKTAPA0IPEK0	0	0	0	P3TAPA0IPEK0	P2TAPA0IPEK0	P1TAPA0IPEK0	P0TAPA0IPEK0
ICTAPA0IPEK0H	FFFF616B$_H$	0	0	0	RFTAPA0IPEK0	0	0	0	0
ICTAPA0IVLY0L	FFFF616C$_H$	MKTAPA0IVLY0	0	0	0	P3TAPA0IVLY0	P2TAPA0IVLY0	P1TAPA0IVLY0	P0TAPA0IVLY0
ICTAPA0IVLY0H	FFFF616D$_H$	0	0	0	RFTAPA0IVLY0	0	0	0	0
ICTAPA1IPEK0L	FFFF616E$_H$	MKTAPA1IPEK0	0	0	0	P3TAPA1IPEK0	P2TAPA1IPEK0	P1TAPA1IPEK0	P0TAPA1IPEK0
ICTAPA1IPEK0H	FFFF616F$_H$	0	0	0	RFTAPA1IPEK0	0	0	0	0
ICTAPA1IVLY0L	FFFF6170$_H$	MKTAPA1IVLY0	0	0	0	P3TAPA1IVLY0	P2TAPA1IVLY0	P1TAPA1IVLY0	P0TAPA1IVLY0
ICTAPA1IVLY0H	FFFF6171$_H$	0	0	0	RFTAPA1IVLY0	0	0	0	0
ICFLX0I0L	FFFF6172$_H$	MKFLX0I0	0	0	0	P3FLX0I0	P2FLX0I0	P1FLX0I0	P0FLX0I0
ICFLX0I0H	FFFF6173$_H$	0	0	0	RFFLX0I0	0	0	0	0
ICFLX0I1L	FFFF6174$_H$	MKFLX0I1	0	0	0	P3FLX0I1	P2FLX0I1	P1FLX0I1	P0FLX0I1
ICFLX0I1H	FFFF6175$_H$	0	0	0	RFFLX0I1	0	0	0	0

Table 4-3 Addresses and Bits of the Interrupt Control Registers (9/9)

Register	Address	Bit							
		7	6	5	4	3	2	1	0
ICFLX0I2L	FFFF6176$_H$	MKFLX0I2	0	0	0	P3FLX0I2	P2FLX0I2	P1FLX0I2	P0FLX0I2
ICFLX0I2H	FFFF6177$_H$	0	0	0	RFFLX0I2	0	0	0	0
ICFLX0I3L	FFFF6178$_H$	MKFLX0I3	0	0	0	P3FLX0I3	P2FLX0I3	P1FLX0I3	P0FLX0I3
ICFLX0I3H	FFFF6179$_H$	0	0	0	RFFLX0I3	0	0	0	0
ICFLX0SIFL08L	FFFF617A$_H$	MKFLX0SIFL08	0	0	0	P3FLX0SIFL08	P2FLX0SIFL08	P1FLX0SIFL08	P0FLX0SIFL08
ICFLX0SIFL08H	FFFF617B$_H$	0	0	0	RFFLX0SIFL08	0	0	0	0
ICFLX0SIFL09L	FFFF617C$_H$	MKFLX0SIFL09	0	0	0	P3FLX0SIFL09	P2FLX0SIFL09	P1FLX0SIFL09	P0FLX0SIFL09
ICFLX0SIFL09H	FFFF617D$_H$	0	0	0	RFFLX0SIFL09	0	0	0	0
ICFLX0SIFL02L	FFFF617E$_H$	MKFLX0SIFL02	0	0	0	P3FLX0SIFL02	P2FLX0SIFL02	P1FLX0SIFL02	P0FLX0SIFL02
ICFLX0SIFL02H	FFFF617F$_H$	0	0	0	RFFLX0SIFL02	0	0	0	0
ICFLX0SIFL07L	FFFF6180$_H$	MKFLX0SIFL07	0	0	0	P3FLX0SIFL07	P2FLX0SIFL07	P1FLX0SIFL07	P0FLX0SIFL07
ICFLX0SIFL07H	FFFF6181$_H$		0	0	RFFLX0SIFL07	0	0	0	0
ICSW0L	FFFF6182$_H$	MKSW0	0	0	0	P3SW0	P2SW0	P1SW0	P0SW0
ICSW0H	FFFF6183$_H$		0	0	RFSW0	0	0	0	0
ICSW1L	FFFF6184$_H$	MKSW1	0	0	0	P3SW1	P2SW1	P1SW1	P0SW1
ICSW1H	FFFF6185$_H$		0	0	RFSW1	0	0	0	0
ICSW2L	FFFF6186$_H$	MKSW2	0	0	0	P3SW2	P2SW2	P1SW2	P0SW2
ICSW2H	FFFF6187$_H$		0	0	RFSW2	0	0	0	0
ICSW3L	FFFF6188$_H$	MKSW3	0	0	0	P3SW3	P2SW3	P1SW3	P0SW3
ICSW3H	FFFF6189$_H$		0	0	RFSW3	0	0	0	0
ICSW4L	FFFF618A$_H$	MKSW4	0	0	0	P3SW4	P2SW4	P1SW4	P0SW4
ICSW4H	FFFF618B$_H$		0	0	RFSW4	0	0	0	0
ICSW5L	FFFF618C$_H$	MKSW5	0	0	0	P3SW5	P2SW5	P1SW5	P0SW5
ICSW5H	FFFF618D$_H$		0	0	RFSW5	0	0	0	0
ICSW6L	FFFF618E$_H$	MKSW6	0	0	0	P3SW6	P2SW6	P1SW6	P0SW6
ICSW6H	FFFF618F$_H$		0	0	RFSW6	0	0	0	0
ICSW7L	FFFF6190$_H$	MKSW7	0	0	0	P3SW7	P2SW7	P1SW7	P0SW7
ICSW7H	FFFF6191$_H$		0	0	RFSW7	0	0	0	0
ICSW8L	FFFF6192$_H$	MKSW8	0	0	0	P3SW8	P2SW8	P1SW8	P0SW8
ICSW8H	FFFF6193$_H$		0	0	RFSW8	0	0	0	0
ICSW9L	FFFF6194$_H$	MKSW9	0	0	0	P3SW9	P2SW9	P1SW9	P0SW9
ICSW9H	FFFF6195$_H$		0	0	RFSW9	0	0	0	0

4.3.2 IMRm (m = 0 to 12): EI Level Interrupt Mask Register

These registers are for aggregating the ICxx.MKxx bits. Bits of the IMRm register reflect settings of the corresponding ICxx.MKxx bits. Corresponding ICxx.MKxx bits also reflect the settings of IMRm registers.

Caution The IMRm.IMRmEIMKn bits for interrupt sources not listed in table 4-2 must always be set to 1.

(1) EI Level Interrupt Mask Register 0 (IMR0)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR0EIMK15 to IMR0EIMK0 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6400$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR0

	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR0H)	IMR0EIMK15	IMR0EIMK14	IMR0EIMK13	IMR0EIMK12	IMR0EIMK11	IMR0EIMK10	IMR0EIMK9	IMR0EIMK8
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR0L)	IMR0EIMK7	IMR0EIMK6	IMR0EIMK5	IMR0EIMK4	IMR0EIMK3	IMR0EIMK2	IMR0EIMK1	IMR0EIMK0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR0EIMK15 to IMR0EIMK0	These are mask bits for EI level maskable interrupt (EIINT) channels 0 to 15 (IMR0EIMK15 to IMR0EIMK0 correspond to EIINT15 to EIINT0). 0: Enables interrupt servicing 1: Disables interrupt servicing

(2) EI Level Interrupt Mask Register 1 (IMR1)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR1EIMK31 to IMR1EIMK16 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6402$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR1

	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR1H)	IMR1EIMK31	IMR1EIMK30	IMR1EIMK29	IMR1EIMK28	IMR1EIMK27	IMR1EIMK26	IMR1EIMK25	IMR1EIMK24
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR1L)	IMR1EIMK23	IMR1EIMK22	IMR1EIMK21	IMR1EIMK20	IMR1EIMK19	IMR1EIMK18	IMR1EIMK17	IMR1EIMK16
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR1EIMK31 to IMR1EIMK16	These are mask bits for EI level maskable interrupt (EIINT) channels 16 to 31 (IMR1EIMK31 to IMR1EIMK16 correspond to EIINT31 to EIINT16). 0: Enables interrupt servicing 1: Disables interrupt servicing

(3) EI Level Interrupt Mask Register 2 (IMR2)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR2EIMK47 to IMR2EIMK32 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6404$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR2

	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR2H)	IMR2EIMK47	IMR2EIMK46	IMR2EIMK45	IMR2EIMK44	IMR2EIMK43	IMR2EIMK42	IMR2EIMK41	IMR2EIMK40
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR2L)	IMR2EIMK39	IMR2EIMK38	IMR2EIMK37	IMR2EIMK36	IMR2EIMK35	IMR2EIMK34	IMR2EIMK33	IMR2EIMK32
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR2EIMK47 to IMR2EIMK32	These are mask bits for EI level maskable interrupt (EIINT) channels 32 to 47 (IMR2EIMK47 to IMR2EIMK32 correspond to EIINT47 to EIINT32). 0: Enables interrupt servicing 1: Disables interrupt servicing

(4) EI Level Interrupt Mask Register 3 (IMR3)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR3EIMK63 to IMR3EIMK48 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6406$_H$

Initial value FFFF$_H$

This register is initialized by any reset.

IMR3	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR3H)	IMR3EIMK63	IMR3EIMK62	IMR3EIMK61	IMR3EIMK60	IMR3EIMK59	IMR3EIMK58	IMR3EIMK57	IMR3EIMK56
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR3L)	IMR3EIMK55	IMR3EIMK54	IMR3EIMK53	IMR3EIMK52	IMR3EIMK51	IMR3EIMK50	IMR3EIMK49	IMR3EIMK48
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR3EIMK63 to IMR3EIMK48	These are mask bits for EI level maskable interrupt (EIINT) channels 48 to 63 (IMR3EIMK63 to IMR3EIMK48 correspond to EIINT63 to EIINT48). 0: Enables interrupt servicing 1: Disables interrupt servicing

(5) EI Level Interrupt Mask Register 4 (IMR4)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR4EIMK79 to IMR4EIMK64 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6408$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR4	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR4H)	IMR4EIMK79	IMR4EIMK78	IMR4EIMK77	IMR4EIMK76	IMR4EIMK75	IMR4EIMK74	IMR4EIMK73	IMR4EIMK72
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR4L)	IMR4EIMK71	IMR4EIMK70	IMR4EIMK69	IMR4EIMK68	IMR4EIMK67	IMR4EIMK66	IMR4EIMK65	IMR4EIMK64
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR4EIMK79 to IMR4EIMK64	These are mask bits for EI level maskable interrupt (EIINT) channels 64 to 79 (IMR4EIMK79 to IMR4EIMK64 correspond to EIINT79 to EIINT64). 0: Enables interrupt servicing 1: Disables interrupt servicing

(6) EI Level Interrupt Mask Register 5 (IMR5)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR5EIMK95 to IMR5EIMK80 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 640A$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR5

	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR5H)	IMR5EIMK95	IMR5EIMK94	IMR5EIMK93	IMR5EIMK92	IMR5EIMK91	IMR5EIMK90	IMR5EIMK89	IMR5EIMK88
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR5L)	IMR5EIMK87	IMR5EIMK86	IMR5EIMK85	IMR5EIMK84	IMR5EIMK83	IMR5EIMK82	IMR5EIMK81	IMR5EIMK80
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR5EIMK95 to IMR5EIMK80	These are mask bits for EI level maskable interrupt (EIINT) channels 80 to 95 (IMR5EIMK95 to IMR5EIMK80 correspond to EIINT95 to EIINT80). 0: Enables interrupt servicing 1: Disables interrupt servicing

(7) EI Level Interrupt Mask Register 6 (IMR6)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR6EIMK111 to IMR6EIMK96 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 640C$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR6

	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR6H)	IMR6EIMK111	IMR6EIMK110	IMR6EIMK109	IMR6EIMK108	IMR6EIMK107	IMR6EIMK106	IMR6EIMK105	IMR6EIMK104
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR6L)	IMR6EIMK103	IMR6EIMK102	IMR6EIMK101	IMR6EIMK100	IMR6EIMK99	IMR6EIMK98	IMR6EIMK97	IMR6EIMK96
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR6EIMK111 to IMR6EIMK96	These are mask bits for EI level maskable interrupt (EIINT) channels 96 to 111 (IMR6EIMK111 to IMR6EIMK96 correspond to EIINT111 to EIINT96). 0: Enables interrupt servicing 1: Disables interrupt servicing

(8) EI Level Interrupt Mask Register 7 (IMR7)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR7EIMK127 to IMR7EIMK112 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 640E$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR7	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR7H)	IMR7EIMK127	1	IMR7EIMK125	IMR7EIMK124	IMR7EIMK123	IMR7EIMK122	IMR7EIMK121	IMR7EIMK120
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR7L)	IMR7EIMK119	IMR7EIMK118	IMR7EIMK117	IMR7EIMK116	IMR7EIMK115	IMR7EIMK114	IMR7EIMK113	IMR7EIMK112
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR7EIMK127 to IMR7EIMK112	These are mask bits for EI level maskable interrupt (EIINT) channels 112 to 127 (IMR7EIMK127 to IMR7EIMK112 correspond to EIINT127 to EIINT112). 0: Enables interrupt servicing 1: Disables interrupt servicing

(9) EI Level Interrupt Mask Register 8 (IMR8)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR8EIMK143 to IMR8EIMK128 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6410$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR8	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR8H)	IMR8EIMK143	IMR8EIMK142	IMR8EIMK141	IMR8EIMK140	IMR8EIMK139	IMR8EIMK138	IMR8EIMK137	IMR8EIMK136
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR8L)	IMR8EIMK135	IMR8EIMK134	IMR8EIMK133	IMR8EIMK132	IMR8EIMK131	IMR8EIMK130	IMR8EIMK129	IMR8EIMK128
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR8EIMK143 to IMR8EIMK128	These are mask bits for EI level maskable interrupt (EIINT) channels 128 to 143 (IMR8EIMK143 to IMR8EIMK128 correspond to EIINT143 to EIINT128). 0: Enables interrupt servicing 1: Disables interrupt servicing

(10) EI Level Interrupt Mask Register 9 (IMR9)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR9EIMK159 to IMR9EIMK144 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6412$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR9	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR9H)	IMR9EIMK159	IMR9EIMK158	IMR9EIMK157	IMR9EIMK156	IMR9EIMK155	IMR9EIMK154	IMR9EIMK153	IMR9EIMK152
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR9L)	IMR9EIMK151	IMR9EIMK150	IMR9EIMK149	IMR9EIMK148	IMR9EIMK147	IMR9EIMK146	IMR9EIMK145	IMR9EIMK144
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR9EIMK159 to IMR9EIMK144	These are mask bits for EI level maskable interrupt (EIINT) channels 144 to 159 (IMR9EIMK159 to IMR9EIMK144 correspond to EIINT159 to EIINT144). 0: Enables interrupt servicing 1: Disables interrupt servicing

(11) EI Level Interrupt Mask Register 10 (IMR10)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR10EIMK175 to IMR10EIMK160 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6414$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR10	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR10H)	IMR10EIMK175	IMR10EIMK174	IMR10EIMK173	IMR10EIMK172	IMR10EIMK171	IMR10EIMK170	IMR10EIMK169	IMR10EIMK168
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR10L)	IMR10EIMK167	IMR10EIMK166	IMR10EIMK165	IMR10EIMK164	IMR10EIMK163	IMR10EIMK162	IMR10EIMK161	IMR10EIMK160
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR10EIMK175 to IMR10EIMK160	These are mask bits for EI level maskable interrupt (EIINT) channels 160 to 175 (IMR10EIMK175 to IMR10EIMK160 correspond to EIINT175 to EIINT160). 0: Enables interrupt servicing 1: Disables interrupt servicing

(12) EI Level Interrupt Mask Register 11 (IMR11)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR11EIMK191 to IMR11EIMK176 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6416$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR11	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR11H)	IMR11EIMK191	IMR11EIMK190	IMR11EIMK189	IMR11EIMK188	IMR11EIMK187	IMR11EIMK186	IMR11EIMK185	IMR11EIMK184
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR11L)	IMR11EIMK183	IMR11EIMK182	IMR11EIMK181	1	1	IMR11EIMK178	IMR11EIMK177	IMR11EIMK176
	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	IMR11EIMK191 to IMR11EIMK176	These are mask bits for EI level maskable interrupt (EIINT) channels 176 to 191 (IMR11EIMK191 to IMR11EIMK176 correspond to EIINT191 to EIINT176). 0: Enables interrupt servicing 1: Disables interrupt servicing

(13) EI Level Interrupt Mask Register 12 (IMR12)

Access Readable/writable in 1-, 8-, or 16-bit units. All of the corresponding IMR12EIMK202 to IMR12EIMK192 bits are updated simultaneously in response to writing in 8- or 16-bit units.

Address FFFF 6418$_H$

Initial value FFFF$_H$

A reset from any source will initialize the bits.

IMR12	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(IMR12H)	1	1	1	1	1	IMR12EIMK202	IMR12EIMK201	IMR12EIMK200
	R	R	R	R	R	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(IMR12L)	IMR12EIMK199	IMR12EIMK198	IMR12EIMK197	IMR12EIMK196	IMR12EIMK195	IMR12EIMK194	IMR12EIMK193	IMR12EIMK192
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
10 to 0	IMR12EIMK202 to IMR12EIMK192	These are mask bits for EI level maskable interrupt (EIINT) channels 192 to 202 (IMR12EIMK202 to IMR12EIMK192 correspond to EIINT202 to EIINT192). 0: Enables interrupt servicing 1: Disables interrupt servicing

4.3.3 ISPR – In-Service Priority Register

This register holds the interrupt priority of EI level maskable interrupt (EIINT) that is being processed by the CPU. When this register receives a response to interrupt request reception from the CPU core, the bit corresponding to the interrupt priority of that interrupt request is set. When this register receives a notification that interrupt servicing is complete, the highest priority bit from among the set bits is automatically cleared. The bits are not cleared on recovery from an FE level interrupt. When multiple EI level maskable interrupts (EIINT) are generated, this register sets the bits in sequence corresponding to the received priority levels and thus holds a history of the interrupt priority levels.

Access This register can be read* in 8- or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 6440_H

Initial value 0000_H

A reset from any source will initialize the bits.

ISPR	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(ISPRH)	ISPR15	ISPR14	ISPR13	ISPR12	ISPR11	ISPR10	ISPR9	ISPR8
	R*	R*	R*	R*	R*	R*	R*	R*

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(ISPRL)	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
	R*	R*	R*	R*	R*	R*	R*	R*

Bit Position	Bit Name	Function
15 to 0	ISPR15 to ISPR0	These bits indicate the priority of the interrupt being acknowledged. 0: Interrupt request of the priority corresponding to the bit position is not acknowledged. 1: Interrupt request of the priority corresponding to the bit position is being processed by the CPU core.

Note All of the bits in ISPR can be cleared by simultaneously writing 1 to all bits of register ISPC and then simultaneously writing 0 to all bits of ISPR (i.e. by using 16-bit operations). Clearing and setting individual bits as required by software writing to this register is not possible. Once the bits are cleared, the original values are not retrievable. For details on ISPC register, refer to 4.3.5, ISPC – In-Service Priority Clear Register.

4.3.4 PMR – Priority Mask Register

This register specifies an interrupt priority by which an interrupt request flag of EI level maskable interrupt (EIINT) is to be masked. It disables all at once the interrupt requests from the EIINT channel for which the interrupt priority specified by this register is set.

The position of each bit of this register corresponds to an interrupt priority. For example, if 1 is set to bit 0, channel of interrupt priority 0 can be masked.

Access Readable/writable in 1-, 8-, or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 6448_H

Initial value 0000_H

A reset from any source will initialize the bits.

PMR	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(PMRH)	PMR15	PMR14	PMR13	PMR12	PMR11	PMR10	PMR9	PMR8
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(PMRL)	PMR7	PMR6	PMR5	PMR4	PMR3	PMR2	PMR1	PMR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PMR15 to PMR0	These bits specify an interrupt priority by which an interrupt request flag is masked: 0 : Enables interrupt servicing of the priority corresponding to a specified bit position (initial value). 1 : Disables interrupt servicing of the priority corresponding to a specified bit position.

4.3.5 ISPC – In-Service Priority Clear Register

This register is for the clearing of settings for interrupt priority and for the control of operations.

Follow the procedure below to clear ISPR.

1. Write $FFFF_H$ to ISPC.ISCPC[15:0] (as a 16-bit unit).
2. Write 0000_H to ISPR.ISPR[15:0] (as a 16-bit unit).

At the same time, the bits indicating processing of an FE level NMI, FE level-maskable interrupt (FEINT), and EI level-maskable interrupt (EIINT) are cleared in ICSR. This clears all interrupt processing mode registers within the interrupt controller that control the processing in progress of interrupt requests by the CPU core. Once the bits have been cleared (to 0), their original values are not retrievable by software.

When the ISPR is cleared by writing 0 to it, the value in the ISPC is also automatically cleared to 0. When register ISPC is read, the value 1 is read from all bits after 1 has been written to all bits, and the value 0 is read from all bits after a reset or clearing of the ISPR. Writing other than writing 1 to all bits or 0 to all bits leaves the register's value unchanged. Furthermore, although writing 0 to all bits while all bits currently have the value 1 clears all bits of register ISPC to 0 but leaves the value in ISPR unchanged.

Access Readable/writable in 16-bit units only.

Address FFFF 6450_H

Initial value 0000_H

A reset from any source will initialize the bits.

ISPC

15	14	13	12	11	10	9	8
ISPC15	ISPC14	ISPC13	ISPC12	ISPC11	ISPC10	ISPC9	ISPC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
ISPC7	ISPC6	ISPC5	ISPC4	ISPC3	ISPC2	ISPC1	ISPC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	ISPC15 to ISPC0	Controls clearing of bits of the in-service priory register (ISPR.ISPR[15:0]). When the register is read, either all bits are read as 1 or all bits are read as 0. When 1 is read from all of the bits, writing 0 to all bits of the ISPR will clear the ISPR.

4.3.6 SCR – Selected Channel Hold Register

This register holds the channel number of the EI level maskable interrupt (EIINT). The value of this register is updated when an interrupt vector is reported to the CPU core.

Access This register is read-only and is read in 8- or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 6458_H

Initial value 0000_H

A reset from any source will initialize the bits.

SCR	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R

(SCRL)	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
	SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0
	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
7 to 0	SCR7 to SCR0	Holds the channel number of the maskable interrupt that has been acknowledged by the CPU. Caution 1. It is overwritten when multiple interrupts of EI level maskable interrupt (EIINT) are acknowledged. Caution 2. These bits are not updated when an FE level interrupt is acknowledged.

4.3.7 ICSR – Interrupt Controller Status Register

This register indicates the operation status of the interrupt controller.

Access This register is read-only and is read in 1-, 8-, or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 645A$_H$

Initial value 0000$_H$

A reset from any source will initialize the bits.

ICSR	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(ICSRH)	0	0	0	0	0	0	0	ICSRPMF
	R	R	R	R	R	R	R	R

	7(7)	6(6)	5(5)	4(4)	3(3)	2(2)	1(1)	0(0)
(ICSRL)	0	ICSRFNR	ICSRFIR	ICSREIR	0	ICSRFNE	ICSRFIE	ICSREIE
	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
8	ICSRPMF	Indicates 1 if the request flag of a channel of EI level maskable interrupt (EIINT) that has the interrupt priority prohibited by the setting of PMR from being serviced is set.
6	ICSRFNR	Indicates 1 if an FE level non-maskable interrupt (FENMI) has been issued to the CPU.
5	ICSRFIR	Indicates 1 if an FE level maskable interrupt (FEINT) has been issued to the CPU.
4	ICSREIR	Indicates 1 if an EI level maskable interrupt (EIINT) has been issued to the CPU.
2	ICSRFNE	Indicates 1 if an CPU is processing the FE level non-maskable interrupt (FENMI).
1	ICSRFIE	Indicates 1 if an CPU is processing the FE level maskable interrupt (FEINT).
0	ICSREIE	Indicates 1 if an CPU is processing the EI level maskable interrupt (EIINT).

4.3.8 FNC – FE Level NMI Status Register

This register is a flag register that indicates the state of control for FE level non-maskable interrupts (FENMI).

Access This register is read-only and is read in 1-, 8-, or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 645C$_H$

Initial value 0000$_H$

A reset from any source will initialize the bits.

FNC	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(FNCH)	0	0	0	FNRF	0	0	0	0
	R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
12	FNRF	Interrupt request flag 0: No interrupt request (initial value) 1: Interrupt request occurred

4.3.9 FIC – FE Level INT Status Register

This register is a flag register that indicates the state of control for FE level maskable interrupts (FEINTs).

Access This register is read-only and is read in 1-, 8-, or 16-bit units.

Either the eight higher-order bits [15:8] or lower-order bits [7:0] may be accessed by reading in 8-bit units.

Address FFFF 645E$_H$

Initial value 0000$_H$

A reset from any source will initialize the bits.

FIC	15(7)	14(6)	13(5)	12(4)	11(3)	10(2)	9(1)	8(0)
(FICH)	0	0	0	FIRF	0	0	0	0
	R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
12	FIRF	Interrupt request flag 0: No interrupt request (initial value) 1: Interrupt request occurred

4.3.10 INTCFGB – FE level Interrupt Switch Register

This register specifies the destination for output from an interrupt source that is switchable between FE-level and EI-level operation.

Once a value has been written to this register, the value is locked into the register until a reset.

Access This register is readable and writable in 1- and 8-bit units. Do not write 1 to any bit for which no function is indicated.

Bits 6 to 0 are always returned as 0 when this register is read.

Address FF83 A000$_H$

Initial value 00$_H$

A reset from any source will initialize the bits.

	7	6	5	4	3	2	1	0
INTCFGB	INTCFGSL0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
7	INTCFGSL0	0: The INTCME, INTWDTA0NMI and INTCLMA0-2 interrupts generate an FENMI. The INTISG interrupt generates a FEINT. 1: The INTCME interrupt is output as the EIINT0 interrupt, the INTISG interrupt is output as the EIINT1 interrupt, the INTWDTA0NMI interrupt is output as the EIINT2 interrupt, the INTCLMA0 interrupt is output as the EIINT4 interrupt, the INTCLMA1 interrupt is output as the EIINT5 interrupt, and the INTCLMA2 interrupt is output as the EIINT6 interrupt.

Procedure for settings

The procedure for switching an FE level interrupt to an EI level interrupt is given below.

1. Release from reset.
2. Disable output of interrupt signals from the safety guardian (INTISG signals); output is disabled with the initial value.
3. Select EI-level operation by writing to this register (INTCFGB).
4. Enable operation of WDTA0, CLMA0 to CLMA2, and output of interrupt signals from the safety guardian.

Caution When changing the setting of this register, only do so when the state is such that interrupts from none of the following sources will be generated (i.e. before their operation is enabled).

- INTISG
- INTWDTA0NMI
- INTCLMA0-2

4.3.11 INTSTR0B – Interrupt Request Flag Storage Register

This register identifies whether an interrupt request is generated or not from an interrupt source that is switchable between FE-level and EI-level operation.

An individual bit is set by either the generation of an interrupt from the corresponding source or writing to the corresponding bit in the INTSTS0B register.

An individual bit is cleared by either writing 1 to the corresponding bit in the interrupt request flag clearing register(INTSTC0B) or a reset.

Access This register is read-only and may be read in 1- or 8-bit units.

Address FF83 A004$_H$

Initial value 00$_H$

A reset from any source will initialize the bits.

	7	6	5	4	3	2	1	0
INTSTR0B	0	0	INTISTF5	INTISTF4	INTISTF3	INTISTF2	INTISTF1	INTISTF0
	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
5	INTISTF5	0: The INTCLMA2 interrupt is not requested. 1: The INTCLMA2 interrupt is requested.
4	INTISTF4	0: The INTCLMA1 interrupt is not requested. 1: The INTCLMA1 interrupt is requested.
3	INTISTF3	0: The INTCLMA0 interrupt is not requested. 1: The INTCLMA0 interrupt is requested.
2	INTISTF2	0: The INTWDTA0NMI interrupt is not requested. 1: The INTWDTA0NMI interrupt is requested.
1	INTISTF1	0: The INTISG interrupt is not requested. 1: The INTISG interrupt is requested. The INTISG interrupt signal is masked.
0	INTISTF0	0: The INTCME interrupt is not requested. 1: The INTCME interrupt is requested. The INTCME interrupt signal is masked.

Caution 1. When the INTSTR0B.INTISTF1 or 0 bit is 1, the given interrupt source is masked to avoid the generation of multiple interrupts. With INTCLMA2-0 and INTWDTA0NMI, further interrupts will not be generated once one interrupt has been generated and until a reset is input. Clear the bit if the generation of multiple interrupts or a new interrupt is required.

Caution 2. In the µPD70F3505A, µPD70F3506, and µPD70F3507, when HEAPCLK is not at the same frequency as PCLK, when a CPU comparison error occurs, an SGA status register (specifically the SGAmESSTR0.SGAmSSE005 bit) must be read to determine whether the error has occurred because in some cases the error indicator will not be set in INTISTF0 bit).

4.3.12 INTSTC0B – Interrupt Request Flag Clearing Register

This register is used to clear error interrupt source flags.

Writing 1 to a bit corresponding to an interrupt request flag that you wish to clear causes clearing of the corresponding flag in the interrupt request flag storage register(INTSTR0B) and the value of the bit in this register reverts to 0.

Access This register is write-only with writing in 1- or 8-bit units. Do not write 1 to any bit for which no function is indicated.

This register always returns 0 when read.

Address FF83 A008$_H$

Initial value 00$_H$

A reset from any source will initialize the bits.

	7	6	5	4	3	2	1	0
INTSTC0B	0	0	INTISTC5	INTISTC4	INTISTC3	INTISTC2	INTISTC1	INTISTC0
	W	W	W	W	W	W	W	W

Bit Position	Bit Name	Function
5	INTISTC5	0: No processing. This bit is always read as 0. 1: Clears the INTISTF5 flag.
4	INTISTC4	0: No processing. This bit is always read as 0. 1: Clears the INTISTF4 flag.
3	INTISTC3	0: No processing. This bit is always read as 0. 1: Clears the INTISTF3 flag.
2	INTISTC2	0: No processing. This bit is always read as 0. 1: Clears the INTISTF2 flag.
1	INTISTC1	0: No processing. This bit is always read as 0. 1: Clears the INTISTF1 flag.
0	INTISTC0	0: No processing. This bit is always read as 0. 1: Clears the INTISTF0 flag.

4.3.13 INTSTS0B – Interrupt Request Flag Setting Register

This register sets an interrupt error source flag.

Writing 1 to a bit corresponding to an interrupt request flag that you wish to set causes setting of the corresponding flag in the interrupt error source storage register to 1 and the value of the bit in this register reverts to 0.

An interrupt will not be generated when a bit in this register is set to 1.

Access This register is write-only with writing in 1- or 8-bit units. Do not write 1 to any bit for which no function is indicated. This register always returns 0 when read.

Address FF83 A00C$_H$

Initial value 00$_H$

A reset from any source will initialize the bits.

INTSTS0B	7	6	5	4	3	2	1	0
	0	0	INTISTS5	INTISTS4	INTISTS3	INTISTS2	INTISTS1	INTISTS0
	W	W	W	W	W	W	W	W

Bit Position	Bit Name	Function
5	INTISTS5	0: No processing. This bit is always read as 0. 1: Sets INTISTF5 flag.
4	INTISTS4	0: No processing. This bit is always read as 0. 1: Sets INTISTF4 flag.
3	INTISTS3	0: No processing. This bit is always read as 0. 1: Sets INTISTF3 flag.
2	INTISTS2	0: No processing. This bit is always read as 0. 1: Sets INTISTF2 flag.
1	INTISTS2	0: No processing. This bit is always read as 0. 1: Sets INTISTF1 flag.
0	INTISTS0	0: No processing. This bit is always read as 0. 1: Sets INTISTF0 flag.

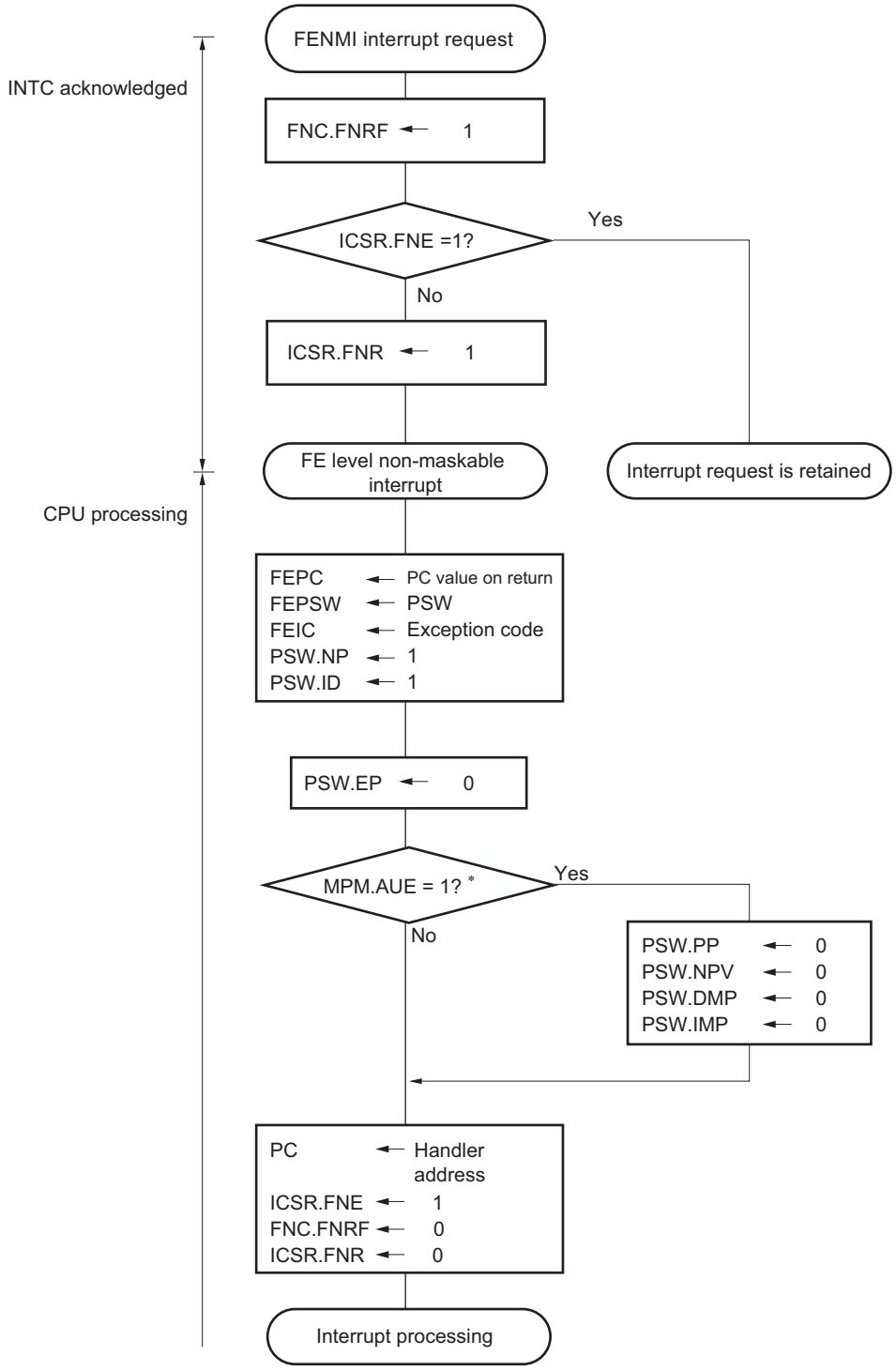
4.4 Interrupt Acknowledgment and Restoring

This section describes the operation during interrupt acknowledgment and restoring from interrupt servicing.

4.4.1 FE Level Non-Maskable Interrupt Caused by FENMI Interrupt Request

When an FENMI interrupt is requested, an FE level non-maskable interrupt is generated in CPU. This FE level non-maskable interrupt is used when a fatal system error occurs.

Caution Upon acknowledgment of the FENMI interrupt, generation of the next FENMI, FEINT, or EIINT interrupt is pended until the FERET instruction is executed (interrupt request is acknowledged and held.)
FENMI can be acknowledged even when the NP bit is set to 1. Therefore, if the FENMI interrupt occurs during the processing of an FEINT exception, PPI exception, or other FE level exceptions, the save address is lost and cannot be restored. After a FENMI interrupt is requested and the required processing has been completed, execute a system reset.



Note: If a processor protection exception (MDP or MIP) occurs, the PSW.PP, NPV, DMP, and IMP bits are always cleared (0), regardless of the status of MPM.AUE.

Figure 4-1 Processing upon Occurrence of FENMI Interrupt Request

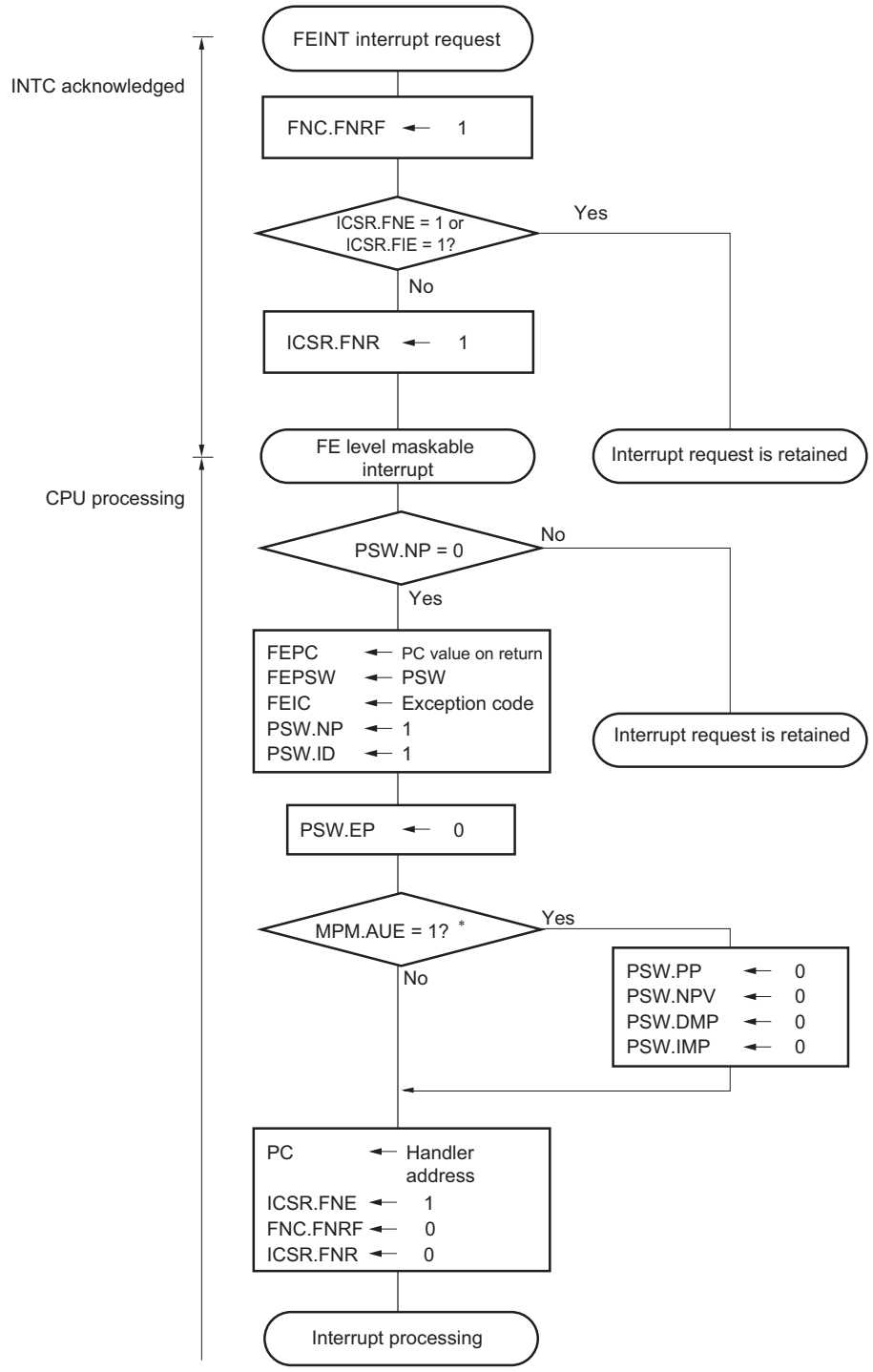
4.4.2 Restore from FE Level Non-Maskable Interrupt (FENMI)

An FE level non-maskable interrupt (FENMI) cannot be restored since it is an interrupt used in cases such as when a fatal system error occurs. Execute a system reset after exception processing.

4.4.3 FE Level Maskable Interrupt Caused by FEINT Interrupt Request

When an FEINT interrupt is requested by the FEINT pin, an FE level maskable interrupt is generated. This interrupt is a recoverable FE level interrupt.

Upon acknowledgment of the FEINT interrupt, generation of the next FEINT or EIINT interrupt is pended until the FERET instruction is executed. Interrupt request is acknowledged and held.



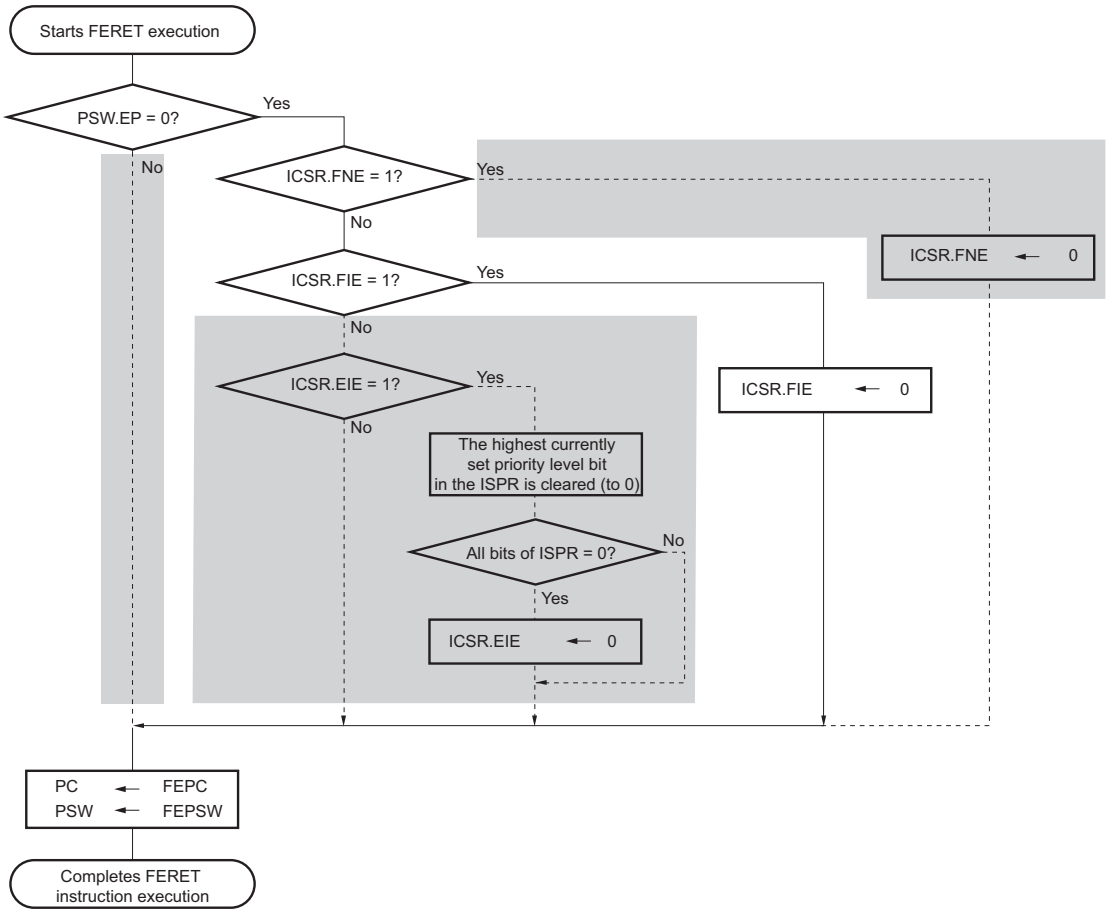
Note: If a processor protection exception (MDP or MIP) occurs, the PSW.PP, NPV, DMP, and IMP bits are always cleared (0), regardless of the status of MPM.AUE.

Figure 4-2 Processing upon Occurrence of FEINT Interrupt Request

4.4.4 Restore from FE Level Maskable Interrupt (FEINT) Servicing

Restore from FE level maskable interrupt (FEINT) servicing is performed using the FERET instructions. Execution of the FERET instruction while the PSW.EP bit status is cleared (0) causes restore processing from the FE level maskable interrupt (FEINT). Completely restoring from interrupt servicing when the PSW.EP bit is "1" is not possible (clearing of the ICSR, ISPR, and other registers is not performed). For return from FE level maskable interrupts (FEINT), execute the FERET instruction with the PSW.EP bit always cleared (0).

Caution Although this CPU core incorporates an RETI instruction, this is only provided for backward compatibility with the V850E1 and V850E2 architectures and its use is, in principle, prohibited.
Replace all RETI instructions other than existing programs that cannot be modified with EIRET or FERET instructions.



Note The shaded portions indicate processing which does not branch during return from servicing of an FE level maskable interrupt (FEINT).

Figure 4-3 Restore from FE Level Maskable Interrupt (FEINT) Servicing

4.4.5 EI Level Maskable Interrupt Caused by EIINT Interrupt Request

An EI level-maskable interrupt leads to an EIINT interrupt request for the CPU: the transition to the interrupt handler is in accord with the setting of the IMR register in the interrupt controller (INTC). This interrupt is a recoverable EI level interrupt.

In the case of an EIINT interrupt, the channel number where the interrupt input occurred is set to the SCR register. As a result, the channel number can be easily known when wishing to share the same interrupt vectors among several channels.

Caution Upon acknowledgment of the EI level interrupt, the priority level of the currently acknowledged interrupt is registered to the ISPR register (in-service priority register). Then, until execution of the EIRET instruction, interrupt with a priority level lower than that of this ISPR register are not generated. Interrupt request is acknowledged and held.
Registration of the priority level of the currently acknowledged interrupt and deletion of the priority level of the interrupt during EIRET to/from the ISPR register are automatically performed by the hardware. Write to the ISPR register cannot be performed by software. Write operations are ignored.

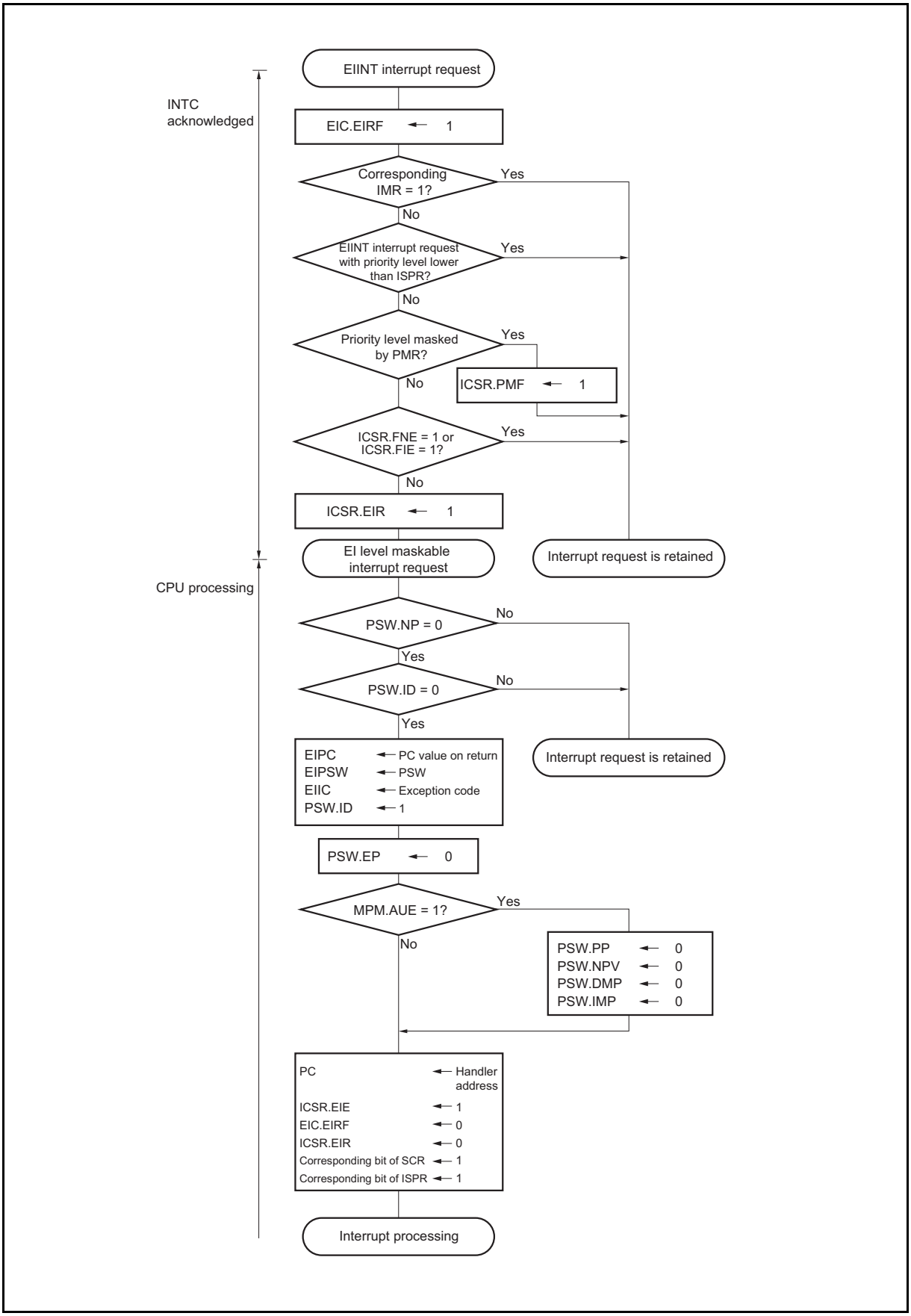
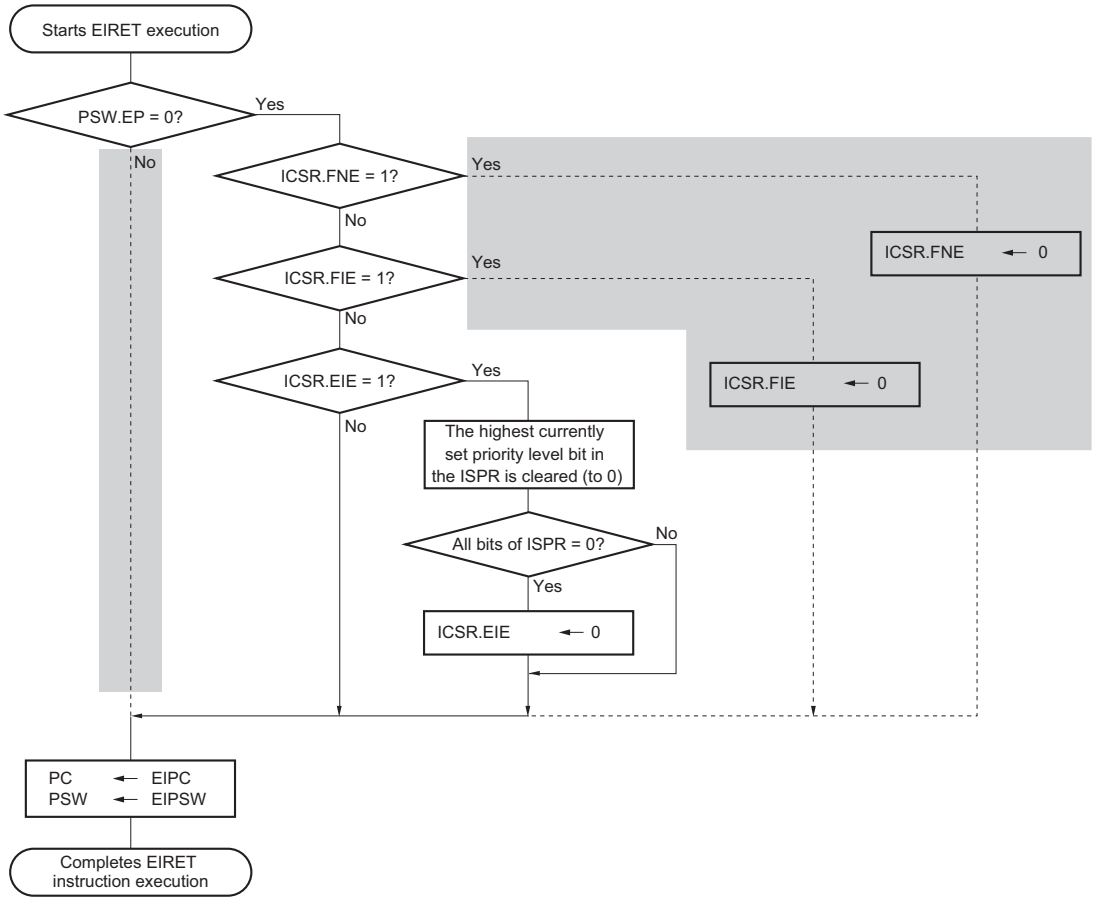


Figure 4-4 Processing upon Occurrence of EIINT Interrupt Request

4.4.6 Restore from EI Level Maskable Interrupt (EIINT)

Restore from EI level maskable interrupt (EIINT) is performed using the EIRET instruction. Execution of the EIRET instruction while the PSW.EP bit status is cleared (0) causes restore processing from the interrupt. Completely restoring from interrupt servicing when the PSW.EP bit is "1" is not possible (clearing of the ICSR, ISPR, and other registers is not performed). For return from EI level maskable interrupt, execute the EIRET instruction with the PSW.EP bit always cleared (0).

Caution Although this CPU core incorporates an RETI instruction, this is only provided for backward compatibility with the V850E1 and V850E2 architectures and its use is, in principle, prohibited.
Replace all RETI instructions other than existing programs that cannot be modified with EIRET or FERET instructions.



Note The shaded portions indicate processing which does not branch during return from servicing of an EI level maskable interrupt (EIINT).

Figure 4-5 Restore from EI Level Maskable Interrupt (EIINT)

4.5 Interrupt Operation

4.5.1 Mask Function of EI Level Maskable Interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by doing the following register settings.

ICxx.MKxx	Operation
1	Masks interrupt.
0	Enables interrupt.

An ICxx.MKxx bit works in conjunction with the corresponding IMRm.IMRmEIMKn (n = 0 to 202) bit, so values can be read from and written to the IMRm (m = 0 to 12) registers.

1. Writing 1 to an IMRm.IMRmEIMKn bit prohibits interrupts on the corresponding channel.
2. When an ICxx.MKxx bit is read, it is read as 1.

Caution For ICxx.MKxx and IMRm.IMRmEIMKn bits, processing subsequent to retention of the interrupt is masked. Acceptance and holding of interrupt requests still proceeds if the ICxx.MKxx and IMRm.IMRmEIMKn bits are set to 1. Furthermore, when an interrupt is blocked by setting of the ICxx.MKxx or IMRm.IMRmEIMKn bit, an interrupt will not be generated even if software requests an interrupt. Moreover, when the value 0 is restored to an ICxx.MKxx and IMRm.IMRmEIMKn bit that is blocking an interrupt request while the request is held pending, the interrupt will be generated at that time. To delete a pending interrupt request that is currently held pending, clear the corresponding ICxx.RFxx bit.

4.5.2 Interrupt Priority Level Judgment

When an FE level non-maskable interrupt (FENMI), FE level maskable interrupt (FEINT), or EI level maskable interrupt (EIINT) is input, the priority levels of other exceptions are taken into account in determining which is to proceed, and the request becomes that for the exception with the highest priority (including the interrupt).

Exceptions requested at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

FENMI > FEINT > EIINT

(For other exceptions, see table 4-2, List of Interrupt Sources, and V850E2M Architecture Manual (R01US0001E).)

Table 4-4 Example of EIINT Interrupt Priority Level Settings and Priority Levels

EIINT	P3xx to P0xx Setting	Priority Level During Operation
EIINT0	3	10
EIINT1	4	11
EIINT2	0	1
EIINT3	0	2
EIINT4	1	3
EIINT5	2	6
EIINT6	2	7
EIINT7	1	4
EIINT8	1	5
EIINT9	2	8
EIINT10	2	9

During interrupt servicing, the interrupt controller also processes multiple interrupts acknowledging other interrupts. When multiple EIINT interrupts are requested at the same time, the interrupt to be acknowledged is determined with the following procedure.

(1) Comparison with the Priority Level as the Interrupt Currently Being Serviced

Interrupts with the same or lower priority level as the interrupt currently being serviced are held.

The priority level of the interrupt currently being serviced is held in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being serviced go on to the next priority judgment stage.

(2) Masking through Priority Mask Register (PMR)

Only interrupts enabled by the PMR register go on to the next priority judgment stage.

(3) Of the Requested Interrupt Sources, that with the Highest Priority Level is Selected

In the case of simultaneous interrupt requests at the highest priority level from multiple sources, that with the lowest interrupt channel number is selected.

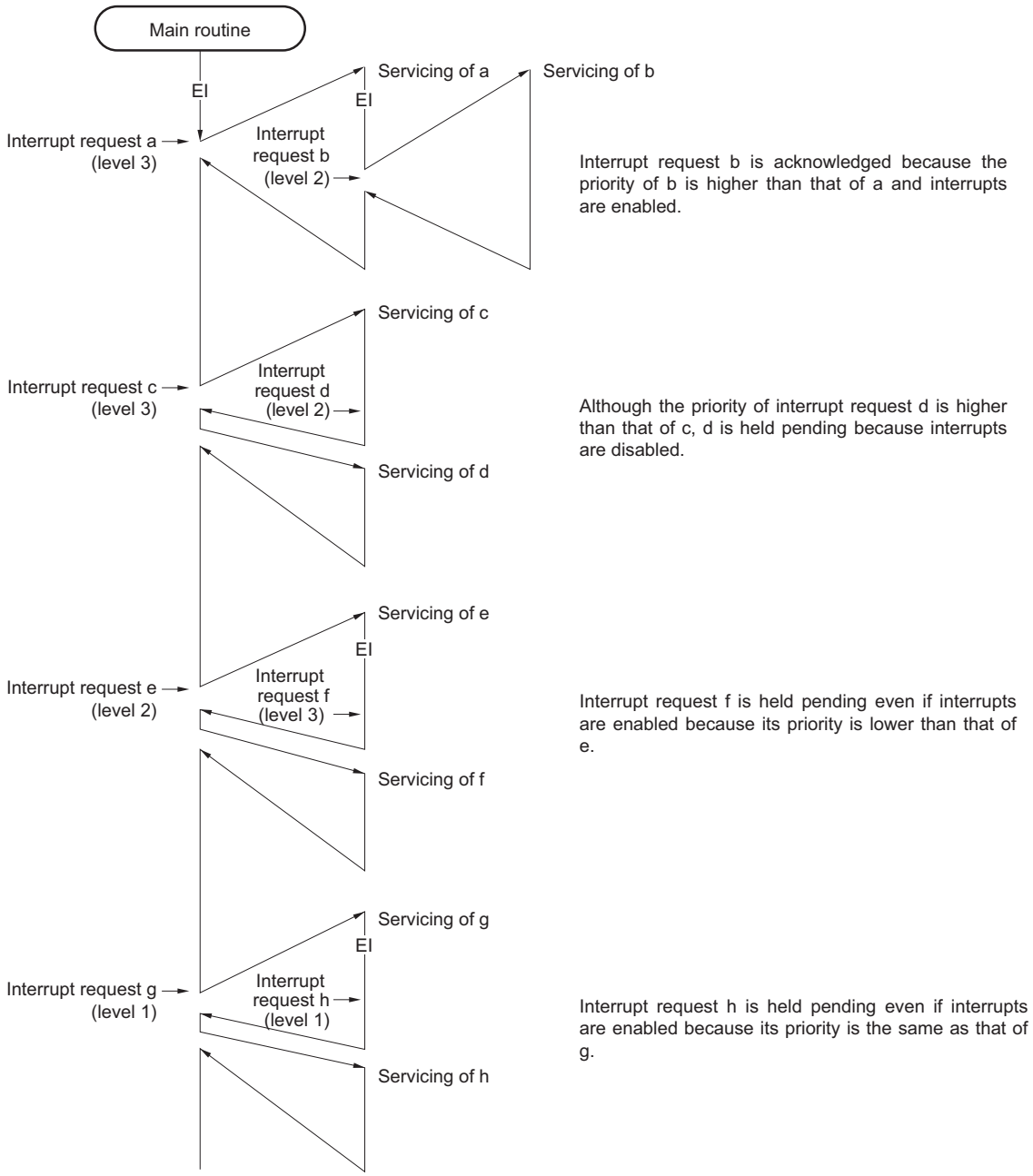
(4) Interrupt Hold by CPU

Interrupt acknowledgment is pended according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT interrupts, FEINT interrupts, and FENMI interrupts is performed even while interrupt acknowledgment is pended, and the interrupt with the highest priority is selected upon realization of the acknowledgment condition.

Example When a priority level 5 EIINT interrupt has already been requested and interrupt generation is pended because the value of the PSW.ID bit is "1", a subsequent priority level 3 EIINT interrupt is requested. Then, if the PSW.ID bit is cleared (0), the priority level 3 EIINT interrupt is generated.

Multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced is shown in figure 4-6.

When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

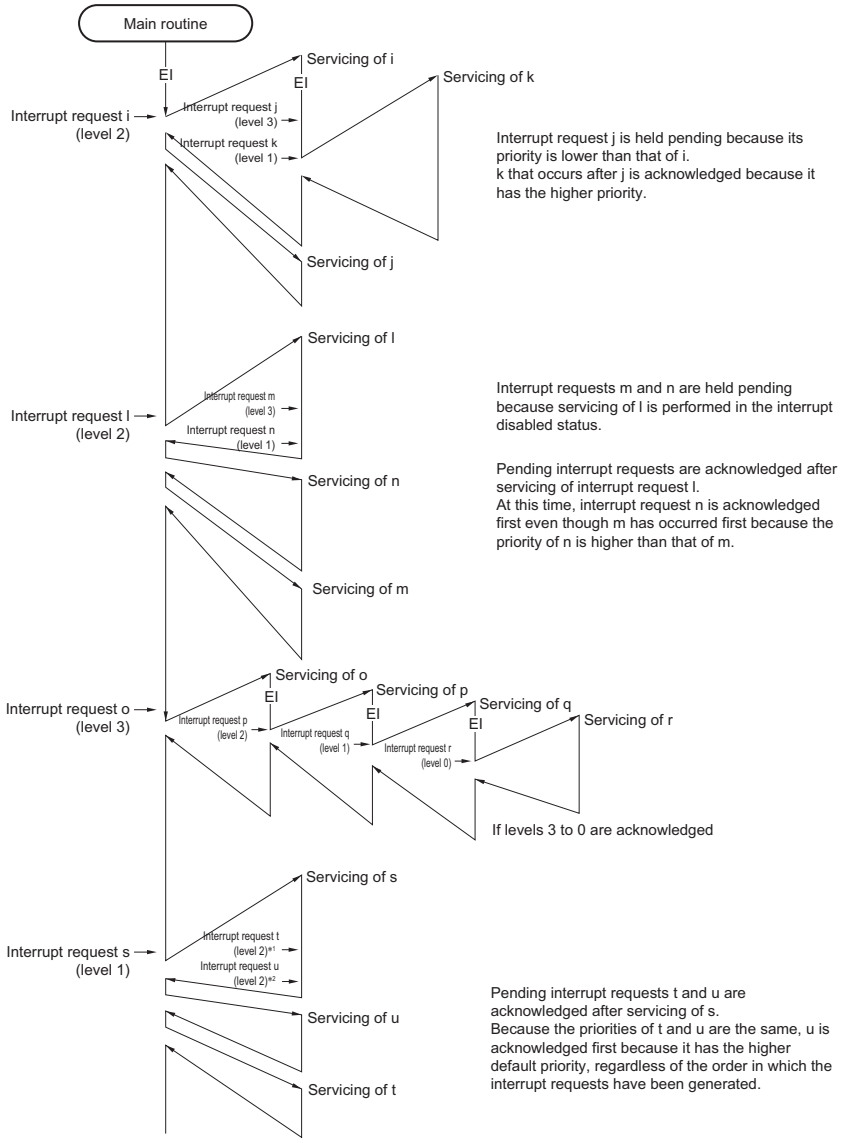


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Note 1. The letters a to h in the figure are simply labels for convenient reference to the interrupt requests in the explanation.

Note 2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 4-6 Example of Processing in which Another Interrupt Request Signal is Issued while an Interrupt is being Serviced (1/2)



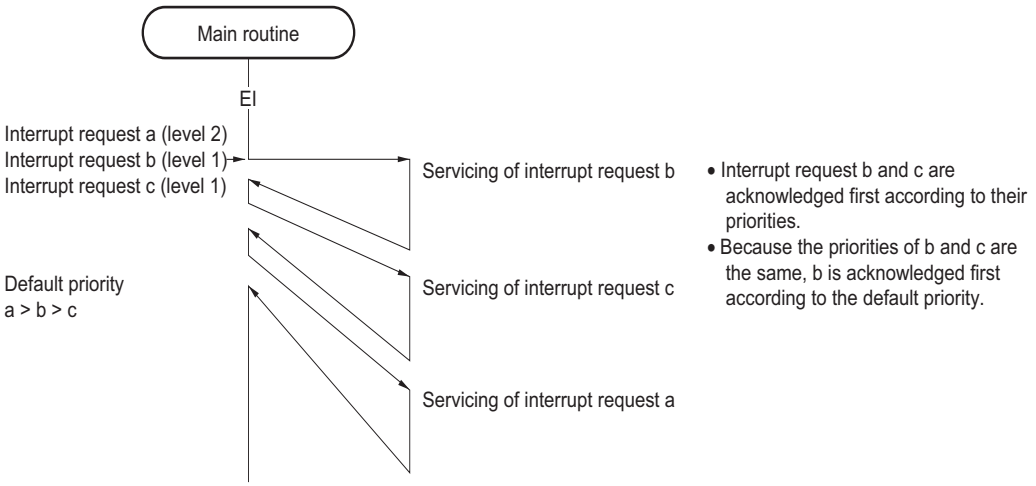
Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

*1. Lower default priority.
*2. Higher default priority.

Note 1. The letters i to u in the figure are simply labels for convenient reference to the interrupt requests in the explanation.

Note 2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 4-6 Example of Processing in which Another Interrupt Request Signal is Issued while an Interrupt is being Serviced (2/2)



Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Note 1. The letters a to c in the figure are simply labels for convenient reference to the interrupt requests in the explanation.

Note 2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 4-7 Example of Servicing Interrupt Request Signals Simultaneously Generated

4.5.3 Priority Mask Function

The priority mask function prohibits in batch EIINT interrupts of the specified interrupt priority level.

The interrupt masking priority level is specified with the PMR register. Masking and acknowledgment can be set for each priority level.

The following operations are possible using this function.

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

PMR.PMRm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note m = 0 to 15

The PMR register prohibits interrupt occurrence. Interrupt request is acknowledged and held even while the interrupt occurrence is prohibited.

The presence of EIINT interrupts held pending with this function can be checked with 4.5.4, Pending Interrupt Report Function.

4.5.4 Pending Interrupt Report Function

The state of the currently pending interrupt can be checked with the pending interrupt report function.

This function allows checking of the following states.

- When interrupts that are masked only by the priority mask function (PMR) exist

  The ICSR.PMF bit is set to 1.

  The only case where the ICSR.PMF bit is not set to 1 is when the setting in the ISPR register is masking interrupts at the given priority level or the interrupt is masked by settings of the ICxx.MKxx and IMRm.IMRmEIMKn bits. Thus, the existence of priority requests pended through the priority mask function can be checked while interrupts are prohibited through priority masking.

- When EI level maskable interrupt request is not output to the CPU

  The ICSR.EIR bit is set to 1.

  By looking at the ICSR.EIR bit in the interval during which PSW.ID = 1, it is possible to check whether an EIINT interrupt request exists.

- When FE level maskable interrupt request is not output to the CPU

  ICSR.FIR bit is set to 1.

  By looking at the ICSR.FIR bit in the interval during which PSW.NP = 1, it is possible to check whether a FEINT interrupt request exists.

4.5.5 In-Service Priority Clear Function

This function initializes the internal status of the interrupt controller. It operates when the ISPC register is accessed. The following operations are possible using this function.

- Clear all contents of ISPR register
- Clear ICSR.EIE, FIE, and FNE bits

All the bits of ISPR register can be cleared to 0 by writing “1” to all bits of this register and then writing “0” to all bits of ISPR. Moreover, the ICSR.EIE, FIE, and FNE bits, which all indicate state in which an interrupt request is being processed by the CPU core, are all cleared.

The value of the ISPC register is automatically cleared to 0 by writing 0 to all the bits of ISPR. The values of the bits of ISPR remain unchanged when the same value is not written to all of the bits.

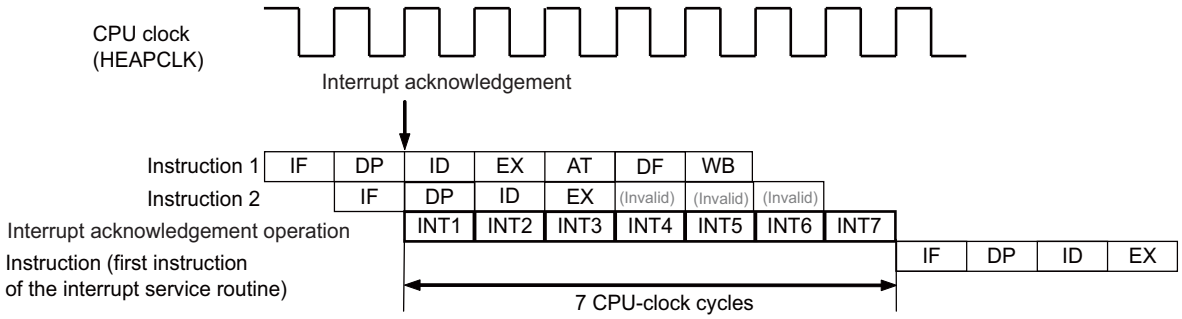
4.6 Exception Handler Address Switching Function

Interrupt handler addresses can be switched by software.

For details, refer to V850E2M Architecture Manual (R01US0001E).

4.7 Interrupt Response Times

Response times from the generation of an interrupt request until activation of interrupt servicing are described below.



Interrupt Response Times		
	Minimum	**Maximum**
Internal interrupt	7 HEAPCLK	12 HEAPCLK
External interrupt	7HEAPCLK + (digital noise cancellation time + 1HEAPCLK)	12HEAPCLK + (digital noise cancellation time + 1HEAPCKL)

Note 1. In addition to the times described above, the internal interrupt requires from a minimum of 1HEAPCLK to a maximum of 3HEAPCLK for the interval between generation and acknowledgement of the interrupt request.

Note 2. INT1 to INT7 in the figure indicate interrupt acknowledgement.

Note 3. The following cases are excluded:
- When successive interrupt non-sample instructions are being executed
- Access to peripheral input/output registers
- When the timing supervision function is in use

Note 4. For the digital noise cancellation time, refer to 2.16, Cancelling Noise on Pins.

Figure 4-8 Pipeline Behavior on acknowledgement of an Interrupt Request (Outline)

Section 5 DMA/DTS Module

The direct memory access (DMA) and data transfer service (DTS) module is described in this section.

5.1 DMA/DTS in Overview

DMA channels: This product has 8 DMA channels.

DTS channels: This product has a channel-free DTS (112 channels altogether)

Instances index n, m, l, k, and j

n, m, l, k, and j are used as shown below:

- For the DMA function, n indicates the channel numbers: n = 0 to 7.
- For the DMA function, m indicates the DMA trigger-source numbers: m = 0 to 107.
- For the DMA function: l indicates the highest number for a DMA trigger source: l = 107
- For the DTS function, k indicates the DTS trigger-source numbers: k = 0 to 111.
- For the DTS function, j indicates the highest number for a DTS trigger source: j = 111.

(1) DMA Trigger Sources

Assignment of DMA trigger sources to DMA channel n is set by the DTFRn.IFC[6:0] bits.

All DMA trigger sources and the settings in the trigger source register (DTFRn) are listed in the following table.

Table 5-1 List of DMA Trigger Sources (1/3)

DTFRn.IFCn[6:0]									
6	5	4	3	2	1	0		Register	DMA Trigger Source
0	0	0	0	0	0	0	0	INTIN000	INTP0
0	0	0	0	0	0	1	1	INTIN001	INTP1
0	0	0	0	0	1	0	2	INTIN002	INTP2
0	0	0	0	0	1	1	3	INTIN003	INTP3
0	0	0	0	1	0	0	4	INTIN004	INTP4
0	0	0	0	1	0	1	5	INTIN005	INTP5
0	0	0	0	1	1	0	6	INTIN006	INTP6
0	0	0	0	1	1	1	7	INTIN007	INTP7
0	0	0	1	0	0	0	8	INTIN008	INTP8
0	0	0	1	0	0	1	9	INTIN009	INTP9
0	0	0	1	0	1	0	10	INTIN010	INTP10
0	0	0	1	0	1	1	11	INTIN011	INTDMA0
0	0	0	1	1	0	0	12	INTIN012	INTDMA1
0	0	0	1	1	0	1	13	INTIN013	INTDMA2
0	0	0	1	1	1	0	14	INTIN014	INTDMA3
0	0	0	1	1	1	1	15	INTIN015	INTDMA4
0	0	1	0	0	0	0	16	INTIN016	INTDMA5
0	0	1	0	0	0	1	17	INTIN017	INTDMA6
0	0	1	0	0	1	0	18	INTIN018	INTDMA7
0	0	1	0	0	1	1	19	INTIN019	INTTAUJ0I0
0	0	1	0	1	0	0	20	INTIN020	INTTAUJ0I1
0	0	1	0	1	0	1	21	INTIN021	INTTAUJ0I2
0	0	1	0	1	1	0	22	INTIN022	INTTAUJ0I3
0	0	1	0	1	1	1	23	INTIN023	INTTAUJ1I0
0	0	1	1	0	0	0	24	INTIN024	INTTAUJ1I1
0	0	1	1	0	0	1	25	INTIN025	INTTAUJ1I2
0	0	1	1	0	1	0	26	INTIN026	INTTAUJ1I3
0	0	1	1	0	1	1	27	INTIN027	INTTAUA0I0
0	0	1	1	1	0	0	28	INTIN028	INTTAUA0I1
0	0	1	1	1	0	1	29	INTIN029	INTTAUA0I2
0	0	1	1	1	1	0	30	INTIN030	INTTAUA0I3
0	0	1	1	1	1	1	31	INTIN031	INTTAUA0I4
0	1	0	0	0	0	0	32	INTIN032	INTTAUA0I5
0	1	0	0	0	0	1	33	INTIN033	INTTAUA0I6
0	1	0	0	0	1	0	34	INTIN034	INTTAUA0I7
0	1	0	0	0	1	1	35	INTIN035	INTTAUA0I8
0	1	0	0	1	0	0	36	INTIN036	INTTAUA0I9
0	1	0	0	1	0	1	37	INTIN037	INTTAUA0I10
0	1	0	0	1	1	0	38	INTIN038	INTTAUA0I11
0	1	0	0	1	1	1	39	INTIN039	INTTAUA0I12

Table 5-1 List of DMA Trigger Sources (2/3)

DTFRn.IFCn[6:0]									
6	5	4	3	2	1	0		Register	DMA Trigger Source
0	1	0	1	0	0	0	40	INTIN040	INTTAUA0I13
0	1	0	1	0	0	1	41	INTIN041	INTTAUA0I14
0	1	0	1	0	1	0	42	INTIN042	INTTAUA0I15
0	1	0	1	0	1	1	43	INTIN043	INTTAUA1I0
0	1	0	1	1	0	0	44	INTIN044	INTTAUA1I1
0	1	0	1	1	0	1	45	INTIN045	INTTAUA1I2
0	1	0	1	1	1	0	46	INTIN046	INTTAUA1I3
0	1	0	1	1	1	1	47	INTIN047	INTTAUA1I4
0	1	1	0	0	0	0	48	INTIN048	INTTAUA1I5
0	1	1	0	0	0	1	49	INTIN049	INTTAUA1I6
0	1	1	0	0	1	0	50	INTIN050	INTTAUA1I7
0	1	1	0	0	1	1	51	INTIN051	INTTAUA1I8
0	1	1	0	1	0	0	52	INTIN052	INTTAUA1I9
0	1	1	0	1	0	1	53	INTIN053	INTTAUA1I10
0	1	1	0	1	1	0	54	INTIN054	INTTAUA1I11
0	1	1	0	1	1	1	55	INTIN055	INTTAUA1I12
0	1	1	1	0	0	0	56	INTIN056	INTTAUA1I13
0	1	1	1	0	0	1	57	INTIN057	INTTAUA1I14
0	1	1	1	0	1	0	58	INTIN058	INTTAUA1I15
0	1	1	1	0	1	1	59	INTIN059	INTADCA0ERR
0	1	1	1	1	0	0	60	INTIN060	INTADCA0I0
0	1	1	1	1	0	1	61	INTIN061	INTADCA0I1
0	1	1	1	1	1	0	62	INTIN062	INTADCA0I2
0	1	1	1	1	1	1	63	INTIN063	INTCSIH0IR
1	0	0	0	0	0	0	64	INTIN064	INTCSIH0IC
1	0	0	0	0	0	1	65	INTIN065	INTCSIH0IJC
1	0	0	0	0	1	0	66	INTIN066	INTCSIH1IR
1	0	0	0	0	1	1	67	INTIN067	INTCSIH1IC
1	0	0	0	1	0	0	68	INTIN068	INTCSIH1IJC
1	0	0	0	1	0	1	69	INTIN069	INTCSIG0IR
1	0	0	0	1	1	0	70	INTIN070	INTCSIG0IC
1	0	0	0	1	1	1	71	INTIN071	INTCSIG1IR
1	0	0	1	0	0	0	72	INTIN072	INTCSIG1IC
1	0	0	1	0	0	1	73	INTIN073	INTCSIG2IR
1	0	0	1	0	1	0	74	INTIN074	INTCSIG2IC
1	0	0	1	0	1	1	75	INTIN075	INTURTH0IS
1	0	0	1	1	0	0	76	INTIN076	INTURTH0IR
1	0	0	1	1	0	1	77	INTIN077	INTURTH0IT
1	0	0	1	1	1	0	78	INTIN078	INTURTH1IS
1	0	0	1	1	1	1	79	INTIN079	INTURTH1IR

Table 5-1 List of DMA Trigger Sources (3/3)

DTFRn.IFCn[6:0]									
6	5	4	3	2	1	0		Register	DMA Trigger Source
1	0	1	0	0	0	0	80	INTIN080	INTURTH1IT
1	0	1	0	0	0	1	81	INTIN081	INTURTH2IS
1	0	1	0	0	1	0	82	INTIN082	INTURTH2IR
1	0	1	0	0	1	1	83	INTIN083	INTURTH2IT
1	0	1	0	1	0	0	84	INTIN084	INTTSG20I11
1	0	1	0	1	0	1	85	INTIN085	INTTSG20I12
1	0	1	0	1	1	0	86	INTIN086	INTTSG20IPEK
1	0	1	0	1	1	1	87	INTIN087	INTTSG20IVLY
1	0	1	1	0	0	0	88	INTIN088	INTTSG21I11
1	0	1	1	0	0	1	89	INTIN089	INTTSG21I12
1	0	1	1	0	1	0	90	INTIN090	INTTSG21IPEK
1	0	1	1	0	1	1	91	INTIN091	INTTSG21IVLY
1	0	1	1	1	0	0	92	INTIN092	INTENCA0IOV
1	0	1	1	1	0	1	93	INTIN093	INTENCA0I0
1	0	1	1	1	1	0	94	INTIN094	INTENCA0I1
1	0	1	1	1	1	1	95	INTIN095	INTENCA0IUD
1	1	0	0	0	0	0	96	INTIN096	INTENCA0IEC
1	1	0	0	0	0	1	97	INTIN097	INTENCA1IOV
1	1	0	0	0	1	0	98	INTIN098	INTENCA1I0
1	1	0	0	0	1	1	99	INTIN099	INTENCA1I1
1	1	0	0	1	0	0	100	INTIN100	INTENCA1IUD
1	1	0	0	1	0	1	101	INTIN101	INTENCA1IEC
1	1	0	0	1	1	0	102	INTIN102	INTTPBA0IPRD
1	1	0	0	1	1	1	103	INTIN103	INTTPBA0IDTY
1	1	0	1	0	0	0	104	INTIN104	INTTPBA0IPAT
1	1	0	1	0	0	1	105	INTIN105	INTTPBA1IPRD
1	1	0	1	0	1	0	106	INTIN106	INTTPBA1IDTY
1	1	0	1	0	1	1	107	INTIN107	INTTPBA1IPAT

(2) DTS Trigger Sources

Table 5-2 List of DTS Trigger Sources

	Enable Register	DTS Trigger Source
0	DTSEN000	INTP0
1	DTSEN001	INTP1
2	DTSEN002	INTP2
3	DTSEN003	INTP3
4	DTSEN004	INTP4
5	DTSEN005	INTP5
6	DTSEN006	INTP6
7	DTSEN007	INTP7
8	DTSEN008	INTP8
9	DTSEN009	INTP9
10	DTSEN010	INTP10
11	DTSEN011	INTDMA0
12	DTSEN012	INTDMA1
13	DTSEN013	INTDMA2
14	DTSEN014	INTDMA3
15	DTSEN015	INTDMA4
16	DTSEN016	INTDMA5
17	DTSEN017	INTDMA6
18	DTSEN018	INTDMA7
19	DTSEN019	INTTAUJ0I0
20	DTSEN020	INTTAUJ0I1
21	DTSEN021	INTTAUJ0I2
22	DTSEN022	INTTAUJ0I3
23	DTSEN023	INTTAUJ1I0
24	DTSEN024	INTTAUJ1I1
25	DTSEN025	INTTAUJ1I2
26	DTSEN026	INTTAUJ1I3
27	DTSEN027	INTTAUA0I0
28	DTSEN028	INTTAUA0I1
29	DTSEN029	INTTAUA0I2
30	DTSEN030	INTTAUA0I3
31	DTSEN031	INTTAUA0I4
32	DTSEN032	INTTAUA0I5
33	DTSEN033	INTTAUA0I6
34	DTSEN034	INTTAUA0I7
35	DTSEN035	INTTAUA0I8
36	DTSEN036	INTTAUA0I9
37	DTSEN037	INTTAUA0I10
38	DTSEN038	INTTAUA0I11
39	DTSEN039	INTTAUA0I12
40	DTSEN040	INTTAUA0I13
41	DTSEN041	INTTAUA0I14
42	DTSEN042	INTTAUA0I15
43	DTSEN043	INTTAUA1I0
44	DTSEN044	INTTAUA1I1
45	DTSEN045	INTTAUA1I2
46	DTSEN046	INTTAUA1I3
47	DTSEN047	INTTAUA1I4
48	DTSEN048	INTTAUA1I5
49	DTSEN049	INTTAUA1I6
50	DTSEN050	INTTAUA1I7
51	DTSEN051	INTTAUA1I8
52	DTSEN052	INTTAUA1I9
53	DTSEN053	INTTAUA1I10
54	DTSEN054	INTTAUA1I11
55	DTSEN055	INTTAUA1I12
56	DTSEN056	INTTAUA1I13
57	DTSEN057	INTTAUA1I14
58	DTSEN058	INTTAUA1I15
59	DTSEN059	INTADCA0ERR
60	DTSEN060	INTADCA0I0
61	DTSEN061	INTADCA0I1
62	DTSEN062	INTADCA0I2
63	DTSEN063	INTCSIH0IR
64	DTSEN064	INTCSIH0IC
65	DTSEN065	INTCSIH0IJC
66	DTSEN066	INTCSIH1IR
67	DTSEN067	INTCSIH1IC
68	DTSEN068	INTCSIH1IJC
69	DTSEN069	INTCSIG0IR
70	DTSEN070	INTCSIG0IC
71	DTSEN071	INTCSIG1IR
72	DTSEN072	INTCSIG1IC
73	DTSEN073	INTCSIG2IR
74	DTSEN074	INTCSIG2IC
75	DTSEN075	INTURTH0IS
76	DTSEN076	INTURTH0IR
77	DTSEN077	INTURTH0IT
78	DTSEN078	INTURTH1IS
79	DTSEN079	INTURTH1IR
80	DTSEN080	INTURTH1IT
81	DTSEN081	INTURTH2IS
82	DTSEN082	INTURTH2IR
83	DTSEN083	INTURTH2IT
84	DTSEN084	INTTSG20I11
85	DTSEN085	INTTSG20I12
86	DTSEN086	INTTSG20IPEK
87	DTSEN087	INTTSG20IVLY
88	DTSEN088	INTTSG21I11
89	DTSEN089	INTTSG21TI12
90	DTSEN090	INTTSG21IPEK
91	DTSEN091	INTTSG21IVLY
92	DTSEN092	INTENCA0IOV
93	DTSEN093	INTENCA0I0
94	DTSEN094	INTENCA0I1
95	DTSEN095	INTENCA0IUD
96	DTSEN096	INTENCA0IEC
97	DTSEN097	INTENCA1IOV
98	DTSEN098	INTENCA1I0
99	DTSEN099	INTENCA1I1
100	DTSEN100	INTENCA1IUD
101	DTSEN101	INTENCA1IEC
102	DTSEN102	INTTPBA0IPRD
103	DTSEN103	INTTPBA0IDTY
104	DTSEN104	INTTPBA0IPAT
105	DTSEN105	INTTPBA1IPRD
106	DTSEN106	INTTPBA1IDTY
107	DTSEN107	INTTPBA1IPAT
108	DTSEN108	INTFLX0I0
109	DTSEN109	INTFLX0I1
110	DTSEN110	INTFLX0SIFL08
111	DTSEN111	INTFLX0SIFL09

5.2 Definitions

The following terms are defined for use in this section,.

Table 5-3 List of Term Definitions

Terms	Meaning
DMA transfer	The period from the start of a DMA cycle until assertion of INTDMA
DMA cycle	Period taken to transfer a single transfer unit once, from the start of the cycle of reading by the on-chip system bus to completion of the write cycle. In the case of 128-bit transfer, completion takes 4 cycles of reading or 4 cycles of writing.
Hardware DMA transfer request	DMA transfer request in the form of any interrupt signal
Software DMA transfer request	DMA transfer request through an internal register (the DTSnSR bit in DTS register)
DMA transfer request	Hardware DMA transfer request or software DMA transfer request
Transfer information (TI)	Transfer-control information, i.e. the information required for DMA transfer, including the transfer addresses, amount of data for transfer, and the number of transfers. Transfer-control information for the DTS in particular is termed TI (transfer information).
TI fetch	The DTS reads transfer information from on-chip RAM
TI write back	The DTS writes transfer information back to on-chip RAM
TI fetch cycle	Period over which the DTS is reading transfer information from on-chip RAM
DTS cycle	Period over which the DTS is transferring information in response to TI
TI write-back cycle	Period over which the DTS is writing transfer information back to on-chip RAM
Single transfer	The DMAC executes a single DMA cycle per transfer request. One cycle of DTS operation is executed per transfer request and consists of the TI fetch cycle, DTS cycle and TI write-back cycle.
Single-step transfer	Only the DMAC has this function. The number of transfers specified by the transfer count setting register (DTCn) is executed per software DMA transfer request. Since the bus is released after each transfer, the CPU is able to generate interrupts. When a transfer request with higher priority is received during single-step transfer, single-step transfer is stopped and execution of the higher-priority request proceeds.
Block transfer	Only the DTS has this function. A single TI fetch cycle, the number of DTS cycles specified in the transfer count setting register (bits[31:16] in TI-A), and a single TI write-back cycle are executed per transfer request. Since the bus is released after each transfer, the CPU is able to generate interrupts. When a transfer request with higher priority is received during block transfer, the DTS ignores the request until the block transfer is completed.

5.3 Overview

Direct memory access (DMA) and the data transfer service (DTS) are used to access data without going through the CPU.

Internally, the subsystem of this product consists of five units: the DTFR (DMA trigger factor register), DTSFSL (DTS factor selector), DMAC, DTS, and DTSSEL (data-transfer selector).

The DMAC is capable of transferring data at high speeds while the DTS transfers data at low speeds and includes a chaining function.

Access to data by the DMA/DTS controller is via the internal system bus. The DTFR, DTSFSL, and DTSSEL have the function of selecting DMA and DTS transfer sources from interrupt requests.

5.3.1 Functions of the DMA Controller (DMAC)

- The DMAC includes registers for storing transfer information (transfer address, transfer unit size, etc.) and for control.
- When a DMA transfer request is accepted, a transfer request is output to the DMAT, according to the transfer information it contains.
- Hardware DMA transfer requests, DMA acknowledge signals, and DMA transfer completion interrupts are input and output.
- Write-back information is written back to the registers.

5.3.2 Function of the DMA Trigger Factor Register (DTFR)

- This register selects DMA transfer sources from among the interrupt signals (triggers for the 8 channels are selected from among the m =108 interrupt signals).

5.3.3 Functions of the Data Transfer Service (DTS)

- The DTS has a set of control registers (the transfer information is placed in the on-chip RAM).
- When a DTS transfer request is accepted, the transfer information is read from the on-chip RAM and a transfer request is output to the DMAT, according to the transfer information.
- Hardware DTS transfer requests, DTS acknowledge signals, and DTS transfer completion interrupts are input and output.
- Write-back information (the next set of transfer information) is written back to the on-chip RAM.

5.3.4 Function of the DTS Factor Selector (DTSFSL)

- Selects sources for activating DTS transfer from among interrupt signals (k = 112 are available as activating sources).
- Four levels of priority can be set.

5.3.5 Function of the DTS Selector (DTSSEL)

- Selects sources for activating DTS transfer from among interrupt signals (k = 112 are available as activating sources).
- Interrupt sources from the DTS and interrupt signals from peripheral devices are selectable.

Table 5-4 Target Spaces for Transfer

Destination / Source	Peripheral I/O (PBUS)	On-chip RAM	On-Chip Code Flash	On-Chip Data Flash
Peripheral I/O (PBUS)	OK	OK	NG	NG
On-chip RAM	OK	OK	NG	NG
On-chip code flash	OK	OK	NG	NG
On-chip data flash	OK	OK	NG	NG

Note For the addresses of the target areas for transfer, refer to *figure5-2 Memory Map as Seen from the DMA/DTS (for the µPD70F3507)*.

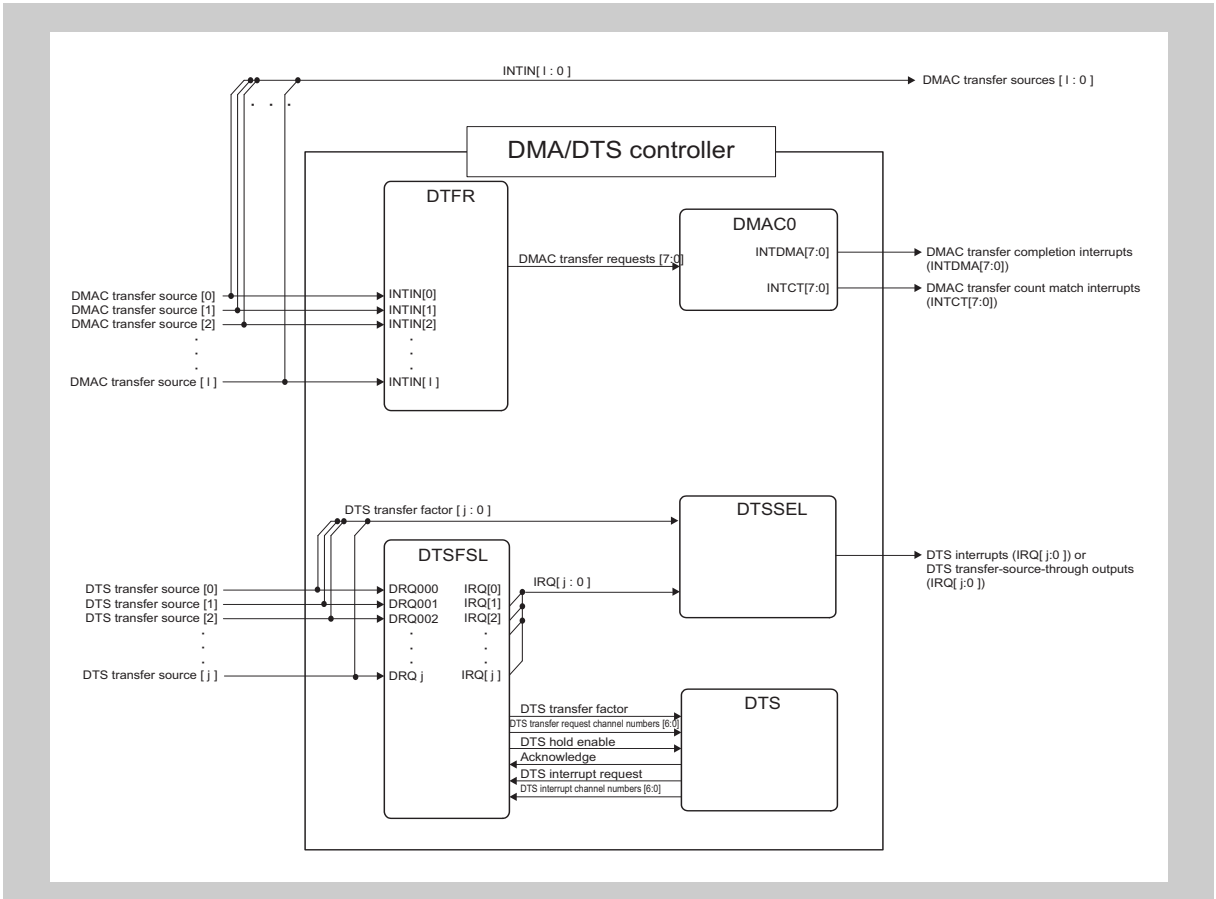


Figure 5-1 Connection of Interrupt Signals

5.3.6 Memory Map for DMA/DTS Access

A memory map of the areas accessible by the DMA/DTS is given below.

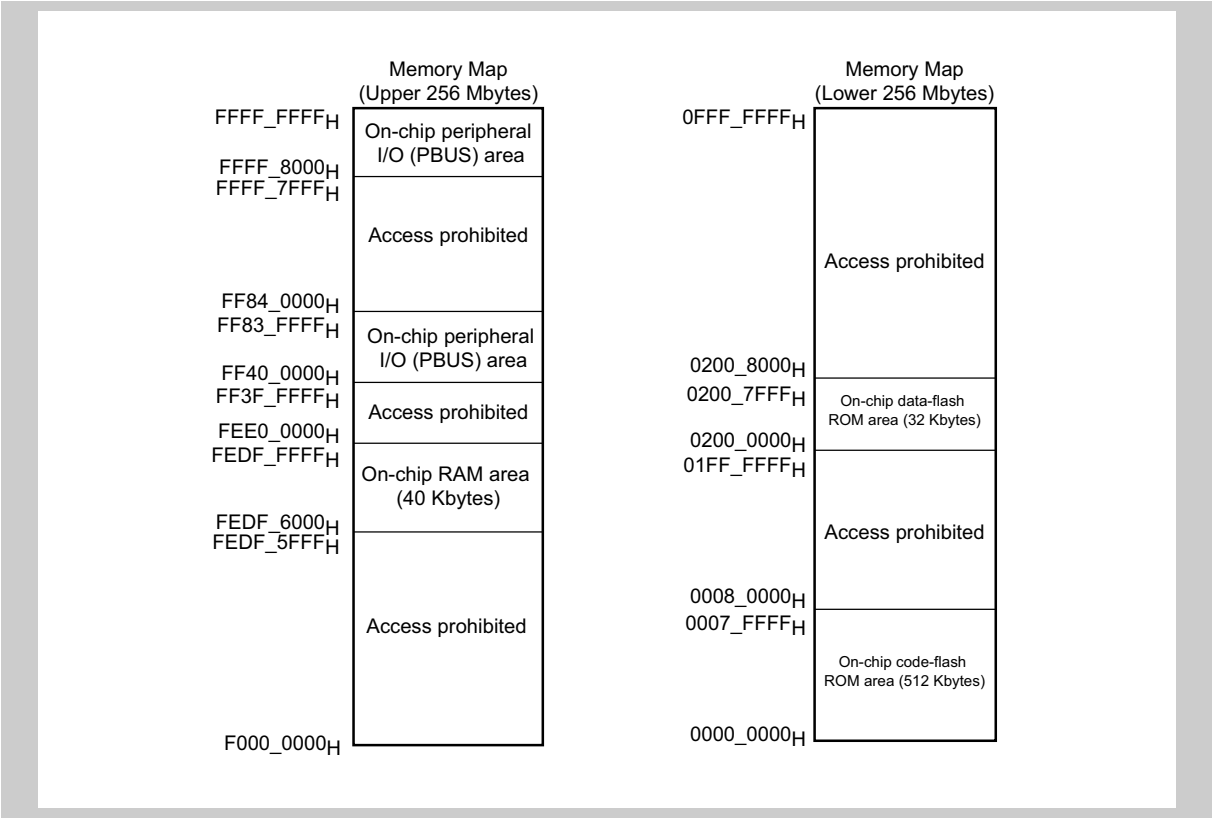


Figure 5-2 Memory Map as Seen from the DMA/DTS (for the μPD70F3507)

5.3.7 Channel Priority

The priority of transfer channels of the DMA/DTS controller is determined in two steps. Orders of priority for the individual blocks (i.e. the DMAC0 or DTS) are determined first.

The priority for DMAC0 is fixed and follows this relation: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. Accordingly, CH0 has the top priority.

For the DTS, settings for four levels of priority are available. Therefore, the priority is determined by these settings and by taking the default priority “CH000 > CH001 > CH002… > CH111” into consideration where this is required.

5.3.8 Arbitration of Transfer Requests

The DMA subsystem has two transfer controllers, DMAC0 and DTS, and arbitrates transfer requests from these controllers. Arbitration is based on this priority relation: DMAC0 > DTS. DMAC0 has the top priority.

When one transfer request is complete, the DMAC0 or DTS cancels the request. Consequently, arbitration leads to the following operation.

Even if requests for transfer on CH0 are continually generated, transfer by DMAC0 will be suspended after each unit transfer to give precedence to transfer in response to a DTS request. Consequently, a request from the DTS takes precedence. After base-table reading (BTR), the DTS continues to output requests for TI-A fetching. When the arbitration is executed, DMAC0 has the higher priority and thus takes precedence over the DTS. After each transfer by DMAC0 is complete, DMAC0 withdraws the request so that transfer from the request for the DTS can be executed.

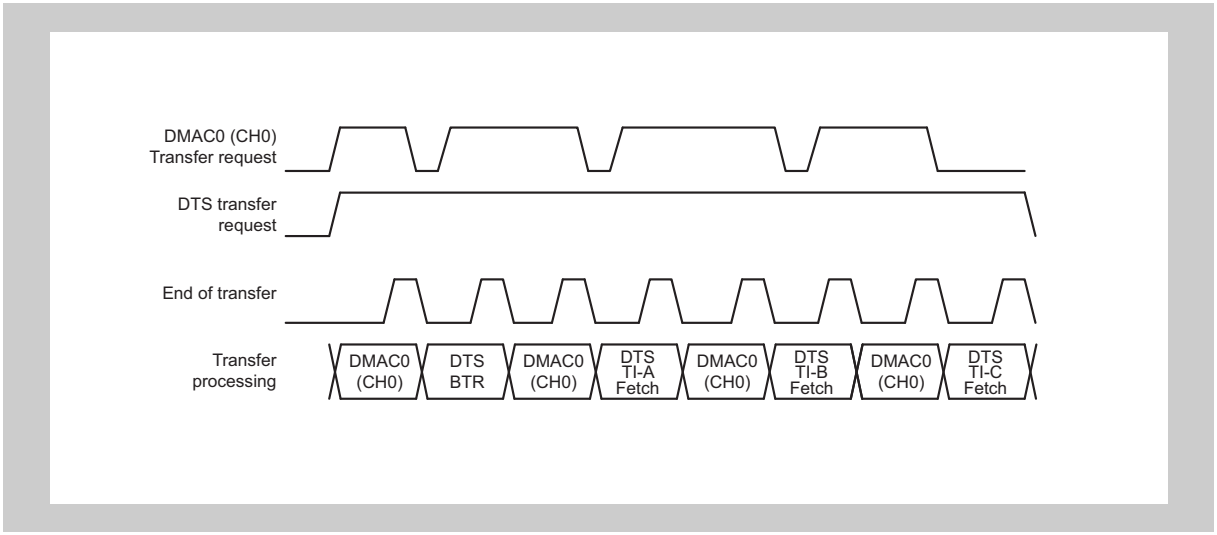


Figure 5-3 Changeover between DMAC0 and DTS

5.4 DMAC Function

5.4.1 Characteristics

Channels

8 (0 channel to 7 channel)

Unit of data transfer

8 bits

16 bits

32 bits

128 bits

Caution When data-flash memory is selected as the source for transfer, specify the unit of data transfer as 32 or 128 bits.

Transfer data

Fixed to little endian

Misaligned data are not supported.

Maximum transfer count

32768(2^{15}) times (the highest order bit of the16-bit register is used for next address function)

Channel priority control

Fixed priority (highest priority (CH0) → lowest priority (CH7))

Targets for transfer

Code-flash

On-chip RAM

Data-flash

Peripheral I/O area

Transfer type

Two-cycle transfer (dual-address transfer)

The addresses at both the transfer source and destination are accessed. Two bus cycles are required to execute transfer once (read cycles + write cycles). The bus is not locked between the read cycles and write cycles, so processing can be interrupted to insert a different cycle of transfer. Also, 128-bit access requires the execution of 4 bus cycles of reading followed by 4 bus cycles of writing. The bus is not locked between the cycles of reading and the cycles of writing, so processing for this transfer may be interrupted at this time.

Transfer mode	• Single transfer mode (when a hardware DNA transfer request is generated) When a hardware DMA transfer request is generated, the controller acquires bus mastership and then releases the bus after one cycle of transfer. If another hardware DMA transfer request is subsequently generated, transfer is executed again. This operation is repeated until transfer has been executed the number of times specified in the transfer count register (DTCn). • Single-step transfer mode (when a software DMA transfer request is generated) When a software DMA transfer request is generated, the bus mastership is acquired, and the bus is released per unit transfer. Once a software DMA transfer request has been acknowledged, this operation is repeated until unit transfers have been executed the number of times specified by the transfer count register (DTCn).
Transfer address control	Incrementation Decrementation Fixed
Transfer error support	When the data from the transfer source contains an error or an error occurs at the transfer destination, DMA transfer is stopped and a SysError exception is issued to notify the CPU.
DMA transfer request	A hardware DMA transfer request or software DMA transfer request can be selected for each channel (by setting the DTRSn register). A software DMA transfer request can be set by software (by setting the DTS register). This register also has a status bit (DTS register) to indicate when a hardware DMA transfer request has been generated.
Interrupt output upon a match of transfer count	This function has a transfer count compare register (DTCCn) for each channel and outputs an interrupt signal (INTDMACT7 to INTDMACT0) when the value in this register matches that in the transfer count register (DTCn) for any channel.
Interrupt output on completion of transfer	This function outputs a transfer completion interrupt signal (INTDMA7 to INTDMA0) when DMA unit transfer on any channel has been completed the number of times specified in the transfer count register (DTCn).
Next address setting	Setting of the next address is handled by equipping each channel with two sets of registers: one for setting the transfer address and transfer count (current transfer registers) of the DMA transfer currently being executed, and the other for setting the transfer address and transfer count (next transfer registers) for the next DMA transfer, i.e. that will follow completion of the DMA transfer that is currently in progress. The registers also have a bit for setting whether to copy values from the “next” to the “current” registers on completion of DMA transfer.
DMA transfer suspension	This function supports DMA transfer suspension by software.

5.4.2 Setting Registers

Table 5-5 DMAC Setting Registers (1/5)

Address	Symbol	Register Name	R/W	Bit Width for Operations				Initial Value
				1	8	16	32	
FFFF7300$_H$	DTRC0	DMA transfer request control register 0	R/W	√	√			00$_H$
FFFF7310$_H$	DTRS0	DMA transfer request select register CH0				√		0000$_H$
FFFF7314$_H$	DSA0	DMA source address register CH0					√	00000000$_H$
FFFF7314$_H$	DSA0L	DMA source address register LCH0				√		0000$_H$
FFFF7316$_H$	DSA0H	DMA source address register HCH0				√		0000$_H$
FFFF7318$_H$	DSC0	DMA source chip select register CH0				√		0001$_H$
FFFF731C$_H$	DNSA0	DMA next source address register CH0					√	00000000$_H$
FFFF731C$_H$	DNSA0L	DMA next source address register LCH0				√		0000$_H$
FFFF731E$_H$	DNSA0H	DMA next source address register HCH0				√		0000$_H$
FFFF7320$_H$	DNSC0	DMA next source chip select register CH0				√		0001$_H$
FFFF7324$_H$	DDA0	DMA destination address register CH0					√	00000000$_H$
FFFF7324$_H$	DDA0L	DMA destination address register LCH0				√		0000$_H$
FFFF7326$_H$	DDA0H	DMA destination address register HCH0				√		0000$_H$
FFFF7328$_H$	DDC0	DMA destination chip select register CH0				√		0001$_H$
FFFF732C$_H$	DNDA0	DMA next destination address register CH0					√	00000000$_H$
FFFF732C$_H$	DNDA0L	DMA next destination address register LCH0				√		0000$_H$
FFFF732E$_H$	DNDA0H	DMA next destination address register HCH0				√		0000$_H$
FFFF7330$_H$	DNDC0	DMA next destination chip select register CH0				√		0001$_H$
FFFF7332$_H$	DTC0	DMA transfer count register CH0				√		0000$_H$
FFFF7334$_H$	DNTC0	DMA next transfer count register CH0				√		0000$_H$
FFFF7336$_H$	DTCC0	DMA transfer count compare register CH0				√		0000$_H$
FFFF7338$_H$	DTCT0	DMA transfer control register CH0				√		0000$_H$
FFFF733A$_H$	DTS0	DMA transfer status register CH0		√	√			00$_H$
FFFF7340$_H$	DTRS1	DMA transfer request select register CH1				√		0000$_H$
FFFF7344$_H$	DSA1	DMA source address register CH1					√	00000000$_H$
FFFF7344$_H$	DSA1L	DMA source address register LCH1				√		0000$_H$
FFFF7346$_H$	DSA1H	DMA source address register HCH1				√		0000$_H$
FFFF7348$_H$	DSC1	DMA source chip select register CH1				√		0001$_H$
FFFF734C$_H$	DNSA1	DMA next source address register CH1					√	00000000$_H$
FFFF734C$_H$	DNSA1L	DMA next source address register LCH1				√		0000$_H$
FFFF734E$_H$	DNSA1H	DMA next source address register HCH1				√		0000$_H$
FFFF7350$_H$	DNSC1	DMA next source chip select register CH1				√		0001$_H$
FFFF7354$_H$	DDA1	DMA destination address register CH1					√	00000000$_H$
FFFF7354$_H$	DDA1L	DMA destination address register LCH1				√		0000$_H$
FFFF7356$_H$	DDA1H	DMA destination address register HCH1				√		0000$_H$
FFFF7358$_H$	DDC1	DMA destination chip select register CH1				√		0001$_H$

Table 5-5 DMAC Setting Registers (2/5)

Address	Symbol	Register Name	R/W	Bit Width for Operations				Initial Value
				1	8	16	32	
FFFF735C$_H$	DNDA1	DMA next destination address register CH1	R/W				√	00000000$_H$
FFFF735C$_H$	DNDA1L	DMA next destination address register LCH1				√		0000$_H$
FFFF735E$_H$	DNDA1H	DMA next destination address register HCH1				√		0000$_H$
FFFF7360$_H$	DNDC1	DMA next destination chip select register CH1				√		0001$_H$
FFFF7362$_H$	DTC1	DMA transfer count register CH1				√		0000$_H$
FFFF7364$_H$	DNTC1	DMA next transfer count register CH1				√		0000$_H$
FFFF7366$_H$	DTCC1	DMA transfer count compare register CH1				√		0000$_H$
FFFF7368$_H$	DTCT1	DMA transfer control register CH1				√		0000$_H$
FFFF736A$_H$	DTS1	DMA transfer status register CH1		√	√			00$_H$
FFFF7370$_H$	DTRS2	DMA transfer request select register CH2				√		0000$_H$
FFFF7374$_H$	DSA2	DMA source address register CH2					√	00000000$_H$
FFFF7374$_H$	DSA2L	DMA source address register LCH2				√		0000$_H$
FFFF7376$_H$	DSA2H	DMA source address register HCH2				√		0000$_H$
FFFF7378$_H$	DSC2	DMA source chip select register CH2				√		0001$_H$
FFFF737C$_H$	DNSA2	DMA next source address register CH2					√	00000000$_H$
FFFF737C$_H$	DNSA2L	DMA next source address register LCH2				√		0000$_H$
FFFF737E$_H$	DNSA2H	DMA next source address register HCH2				√		0000$_H$
FFFF7380$_H$	DNSC2	DMA next source chip select register CH2				√		0001$_H$
FFFF7384$_H$	DDA2	DMA destination address register CH2					√	00000000$_H$
FFFF7384$_H$	DDA2L	DMA destination address register LCH2				√		0000$_H$
FFFF7386$_H$	DDA2H	DMA destination address register HCH2				√		0000$_H$
FFFF7388$_H$	DDC2	DMA destination chip select register CH2				√		0001$_H$
FFFF738C$_H$	DNDA2	DMA next destination address register CH2					√	00000000$_H$
FFFF738C$_H$	DNDA2L	DMA next destination address register LCH2				√		0000$_H$
FFFF738E$_H$	DNDA2H	DMA next destination address register HCH2				√		0000$_H$
FFFF7390$_H$	DNDC2	DMA next destination chip select register CH2				√		0001$_H$
FFFF7392$_H$	DTC2	DMA transfer count register CH2				√		0000$_H$
FFFF7394$_H$	DNTC2	DMA next transfer count register CH2				√		0000$_H$
FFFF7396$_H$	DTCC2	DMA transfer count compare register CH2				√		0000$_H$
FFFF7398$_H$	DTCT2	DMA transfer control register CH2				√		0000$_H$
FFFF739A$_H$	DTS2	DMA transfer status register CH2		√	√			00$_H$
FFFF73A0$_H$	DTRS3	DMA transfer request select register CH3				√		0000$_H$
FFFF73A4$_H$	DSA3	DMA source address register CH3					√	00000000$_H$
FFFF73A4$_H$	DSA3L	DMA source address register LCH3				√		0000$_H$
FFFF73A6$_H$	DSA3H	DMA source address register HCH3				√		0000$_H$
FFFF73A8$_H$	DSC3	DMA source chip select register CH3				√		0001$_H$

Table 5-5 DMAC Setting Registers (3/5)

Address	Symbol	Register Name	R/W	Bit Width for Operations				Initial Value
				1	8	16	32	
FFFF73AC$_H$	DNSA3	DMA next source address register CH3	R/W				√	00000000$_H$
FFFF73AC$_H$	DNSA3L	DMA next source address register LCH3				√		0000$_H$
FFFF73AE$_H$	DNSA3H	DMA next source address register HCH3				√		0000$_H$
FFFF73B0$_H$	DNSC3	DMA next source chip select register CH3				√		0001$_H$
FFFF73B4$_H$	DDA3	DMA destination address register CH3					√	00000000$_H$
FFFF73B4$_H$	DDA3L	DMA destination address register LCH3				√		0000$_H$
FFFF73B6$_H$	DDA3H	DMA destination address register HCH3				√		0000$_H$
FFFF73B8$_H$	DDC3	DMA destination chip select register CH3				√		0001$_H$
FFFF73BC$_H$	DNDA3	DMA next destination address register CH3					√	00000000$_H$
FFFF73BC$_H$	DNDA3L	DMA next destination address register LCH3				√		0000$_H$
FFFF73BE$_H$	DNDA3H	DMA next destination address register HCH3				√		0000$_H$
FFFF73C0$_H$	DNDC3	DMA next destination chip select register CH3				√		0001$_H$
FFFF73C2$_H$	DTC3	DMA transfer count register CH3				√		0000$_H$
FFFF73C4$_H$	DNTC3	DMA next transfer count register CH3				√		0000$_H$
FFFF73C6$_H$	DTCC3	DMA transfer count compare register CH3				√		0000$_H$
FFFF73C8$_H$	DTCT3	DMA transfer control register CH3				√		0000$_H$
FFFF73CA$_H$	DTS3	DMA transfer status register CH3		√	√			00$_H$
FFFF73D0$_H$	DTRS4	DMA transfer request select register CH4				√		0000$_H$
FFFF73D4$_H$	DSA4	DMA source address register CH4					√	00000000$_H$
FFFF73D4$_H$	DSA4L	DMA source address register LCH4				√		0000$_H$
FFFF73D6$_H$	DSA4H	DMA source address register HCH4				√		0000$_H$
FFFF73D8$_H$	DSC4	DMA source chip select register CH4				√		0001$_H$
FFFF73DC$_H$	DNSA4	DMA next source address register CH4					√	00000000$_H$
FFFF73DC$_H$	DNSA4L	DMA next source address register LCH4				√		0000$_H$
FFFF73DE$_H$	DNSA4H	DMA next source address register HCH4				√		0000$_H$
FFFF73E0$_H$	DNSC4	DMA next source chip select register CH4				√		0001$_H$
FFFF73E4$_H$	DDA4	DMA destination address register CH4					√	00000000$_H$
FFFF73E4$_H$	DDA4L	DMA destination address register LCH4				√		0000$_H$
FFFF73E6$_H$	DDA4H	DMA destination address register HCH4				√		0000$_H$
FFFF73E8$_H$	DDC4	DMA destination chip select register CH4				√		0001$_H$
FFFF73EC$_H$	DNDA4	DMA next destination address register CH4					√	00000000$_H$
FFFF73EC$_H$	DNDA4L	DMA next destination address register LCH4				√		0000$_H$
FFFF73EE$_H$	DNDA4H	DMA next destination address register HCH4				√		0000$_H$
FFFF73F0$_H$	DNDC4	DMA next destination chip select register CH4				√		0001$_H$
FFFF73F2$_H$	DTC4	DMA transfer count register CH4				√		0000$_H$
FFFF73F4$_H$	DNTC4	DMA next transfer count register CH4				√		0000$_H$

Table 5-5 DMAC Setting Registers (4/5)

Address	Symbol	Register Name	R/W	Bit Width for Operations				Initial Value
				1	8	16	32	
FFFF73F6$_H$	DTCC4	DMA transfer count compare register CH4	R/W			√		0000$_H$
FFFF73F8$_H$	DTCT4	DMA transfer control register CH4				√		0000$_H$
FFFF73FA$_H$	DTS4	DMA transfer status register CH4		√	√			00$_H$
FFFF7400$_H$	DTRS5	DMA transfer request select register CH5				√		0000$_H$
FFFF7404$_H$	DSA5	DMA source address register CH5					√	00000000$_H$
FFFF7404$_H$	DSA5L	DMA source address register LCH5				√		0000$_H$
FFFF7406$_H$	DSA5H	DMA source address register HCH5				√		0000$_H$
FFFF7408$_H$	DSC5	DMA source chip select register CH5				√		0001$_H$
FFFF740C$_H$	DNSA5	DMA next source address register CH5					√	00000000$_H$
FFFF740C$_H$	DNSA5L	DMA next source address register LCH5				√		0000$_H$
FFFF740E$_H$	DNSA5H	DMA next source address register HCH5				√		0000$_H$
FFFF7410$_H$	DNSC5	DMA next source chip select register CH5				√		0001$_H$
FFFF7414$_H$	DDA5	DMA destination address register CH5					√	00000000$_H$
FFFF7414$_H$	DDA5L	DMA destination address register LCH5				√		0000$_H$
FFFF7416$_H$	DDA5H	DMA destination address register HCH5				√		0000$_H$
FFFF7418$_H$	DDC5	DMA destination chip select register CH5				√		0001$_H$
FFFF741C$_H$	DNDA5	DMA next destination address register CH5					√	00000000$_H$
FFFF741C$_H$	DNDA5L	DMA next destination address register LCH5				√		0000$_H$
FFFF741E$_H$	DNDA5H	DMA next destination address register HCH5				√		0000$_H$
FFFF7420$_H$	DNDC5	DMA next destination chip select register CH5				√		0001$_H$
FFFF7422$_H$	DTC5	DMA transfer count register CH5				√		0000$_H$
FFFF7424$_H$	DNTC5	DMA next transfer count register CH5				√		0000$_H$
FFFF7426$_H$	DTCC5	DMA transfer count compare register CH5				√		0000$_H$
FFFF7428$_H$	DTCT5	DMA transfer control register CH5				√		0000$_H$
FFFF742A$_H$	DTS5	DMA transfer status register CH5		√	√			00$_H$
FFFF7430$_H$	DTRS6	DMA transfer request select register CH6				√		0000$_H$
FFFF7434$_H$	DSA6	DMA source address register CH6					√	00000000$_H$
FFFF7434$_H$	DSA6L	DMA source address register LCH6				√		0000$_H$
FFFF7436$_H$	DSA6H	DMA source address register HCH6				√		0000$_H$
FFFF7438$_H$	DSC6	DMA source chip select register CH6				√		0001$_H$
FFFF743C$_H$	DNSA6	DMA next source address register CH6					√	00000000$_H$
FFFF743C$_H$	DNSA6L	DMA next source address register LCH6				√		0000$_H$
FFFF743E$_H$	DNSA6H	DMA next source address register HCH6				√		0000$_H$
FFFF7440$_H$	DNSC6	DMA next source chip select register CH6				√		0001$_H$
FFFF7444$_H$	DDA6	DMA destination address register CH6					√	00000000$_H$
FFFF7444$_H$	DDA6L	DMA destination address register LCH6				√		0000$_H$
FFFF7446$_H$	DDA6H	DMA destination address register HCH6				√		0000$_H$

Table 5-5 DMAC Setting Registers (5/5)

Address	Symbol	Register Name	R/W	Bit Width for Operations				Initial Value
				1	8	16	32	
$FFFF7448_H$	DDC6	DMA destination chip select register CH6	R/W			√		0001_H
$FFFF744C_H$	DNDA6	DMA next destination address register CH6					√	00000000_H
$FFFF744C_H$	DNDA6L	DMA next destination address register LCH6				√		0000_H
$FFFF744E_H$	DNDA6H	DMA next destination address register HCH6				√		0000_H
$FFFF7450_H$	DNDC6	DMA next destination chip select register CH6				√		0001_H
$FFFF7452_H$	DTC6	DMA transfer count register CH6				√		0000_H
$FFFF7454_H$	DNTC6	DMA next transfer count register CH6				√		0000_H
$FFFF7456_H$	DTCC6	DMA transfer count compare register CH6				√		0000_H
$FFFF7458_H$	DTCT6	DMA transfer control register CH6				√		0000_H
$FFFF745A_H$	DTS6	DMA transfer status register CH6		√	√			00_H
$FFFF7460_H$	DTRS7	DMA transfer request select register CH7				√		0000_H
$FFFF7464_H$	DSA7	DMA source address register CH7					√	00000000_H
$FFFF7464_H$	DSA7L	DMA source address register LCH7				√		0000_H
$FFFF7466_H$	DSA7H	DMA source address register LCH7				√		0000_H
$FFFF7468_H$	DSC7	DMA source chip select register CH7				√		0001_H
$FFFF746C_H$	DNSA7	DMA next source address register CH7					√	00000000_H
$FFFF746C_H$	DNSA7L	DMA next source address register LCH7				√		0000_H
$FFFF746E_H$	DNSA7H	DMA next source address register HCH7				√		0000_H
$FFFF7470_H$	DNSC7	DMA next source chip select register CH7				√		0001_H
$FFFF7474_H$	DDA7	DMA destination address register CH7					√	00000000_H
$FFFF7474_H$	DDA7L	DMA destination address register LCH7				√		0000_H
$FFFF7476_H$	DDA7H	DMA destination address register HCH7				√		0000_H
$FFFF7478_H$	DDC7	DMA destination chip select register CH7				√		0001_H
$FFFF747C_H$	DNDA7	DMA next destination address register CH7					√	00000000_H
$FFFF747C_H$	DNDA7L	DMA next destination address register LCH7				√		0000_H
$FFFF747E_H$	DNDA7H	DMA next destination address register HCH7				√		0000_H
$FFFF7480_H$	DNDC7	DMA next destination chip select register CH7				√		0001_H
$FFFF7482_H$	DTC7	DMA transfer count register CH7				√		0000_H
$FFFF7484_H$	DNTC7	DMA next transfer count register CH7				√		0000_H
$FFFF7486_H$	DTCC7	DMA transfer count compare register CH7				√		0000_H
$FFFF7488_H$	DTCT7	DMA transfer control register CH7				√		0000_H
$FFFF748A_H$	DTS7	DMA transfer status register CH7		√	√			00_H

Caution If an unmapped address is accessed, a write access is ignored and 0 is returned in response to a read access.

5.4.3 Availability of Writing to Control Registers

Control registers can be classified into the two groups shown in the table below: those which are always writable and those for which writing is prohibited while DMA transfer is enabled. However, all of the registers are always readable.

Table 5-6 Availability of Writing to Control Registers

Availability of Writing	Register Name
Always writable	DTRC0, DNSAnL, DNSAnH, DNSCn, DNDAnL, DNDAnH, DNDCn, DNTCn, DTSn
Writing prohibited while DMA transfer is enabled (DTSnDTE = 1) Operation is not guaranteed if writing to these registers is attempted during while DMA transfer.	DTRSn, DSAnL, DSAnH, DSCn, DDAnL, DDAnH, DDCn, DTCn, DTCCn, DTCTn

Note n = 0 to 7

5.5 DMAC Control Registers

5.5.1 DTRC0: DMA Transfer Request Control Register 0

This eight-bit register includes an error flag and controls the suspension of DMA transfer that is currently in progress.

Access This register is readable/writable in 8- or 1-bit units.

Address FFFF 7300_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
DTRC0 ERR	0	0	0	0	0	0	DTRC0 ADS
R/W	R	R	R	R	R	R	R/W

Bit Position	Bit Name	Function
7	DTRC0ERR	DMA transfer error status This bit indicates that an error response has been received from a target for transfer during DMA transfer. To clear this bit, write 0 to it. 0: No DMA transfer error 1: DMA transfer error Note: If an error response is received, the DTRC0ERR and DTRC0ADS bits are set and a SysError exception is generated for the CPU.
0	DTRC0ADS	DMA transfer suspended This bit indicates that DMA transfer has been suspended by a transfer stop request. In addition, writing 1 to this bit suspends DMA transfer that is currently in progress. 0: DMA transfer is not suspended. 1: DMA transfer is suspended (reading) /DMA transfer suspension request (writing)

5.5.2 DTRSn (n = 0 to 7): DMA Transfer Request Select Register

This 16-bit register selects requests from the DMA transfer software and hardware.

Access This register is readable/writable in 16-bit units.

Address DTRS7: FFFF 7460$_H$, DTRS6: FFFF 7430$_H$, DTRS5: FFFF 7400$_H$, DTRS4: FFFF 73D0$_H$, DTRS3: FFFF 73A0$_H$, DTRS2: FFFF 7370$_H$, DTRS1: FFFF 7340$_H$, DTRS0: FFFF 7310$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	DTRSn DTR3	DTRSn DTR2	DTRSn DTR1	DTRSn DTR0
R	R	R	R	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
3 to 0	DTRSnDTR3 to DTRSnDTR0	DMA transfer request assignment These bits specify the assignment of a DMA transfer request to channel n.

DTRSn DTR3	DTRSn DTR2	DTRSn DTR1	DTRSn DTR0	DAM Transfer Request
0	0	0	0	Software DMA transfer request
0	0	0	1	Hardware DMA transfer request
Other than the above				Setting prohibited

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Operation is also not guaranteed if a prohibited setting is made in DTRSnDTR[3:0].

5.5.3 DSAnL (n = 0 to 7): DMA Source Address Register L

This 16-bit register forms the 16 lower-order bits of a 32-bit register used to set a source for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DSA7L: FFFF 7464_H, DSA6L: FFFF 7434_H, DSA5L: FFFF 7404_H, DSA4L: FFFF $73D4_H$, DSA3L: FFFF $73A4_H$, DSA2L: FFFF 7374_H, DSA1L: FFFF 7344_H, DSA0L: FFFF 7314_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DSAn SA15	DSAn SA14	DSAn SA13	DSAn SA12	DSAn SA11	DSAn SA10	DSAn SA9	DSAn SA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DSAn SA7	DSAn SA6	DSAn SA5	DSAn SA4	DSAn SA3	DSAn SA2	DSAn SA1	DSAn SA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	DSAnSA15 to DSAnSA0	DMA source address The 16 lower-order bits of the source address for transfer on channel n are set in these bits. If this register is read during DMA transfer, the value read is the corresponding part of the next address from which data is to be transferred. We recommend access to this register in combination with DSAnH as a 32-bit unit. If the DNSAnNSAV bit of the DNSAnH register is not set (to 1) on completion of DMA transfer, these bits revert to their previous values (those when DMA transfer was started).

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set addresses by access in 32-bit units while the DTSnDTE bit is 0 because data transfer is not possible from an address for which setting is in progress.

Caution 3. DMA transfer of misaligned data is not supported. The 4 lower-order bits of addresses corresponding to the transfer data size are as follows (x indicates any value).

Operation is not guaranteed if settings other than the following are made.

Data Size	DSAnSA3	DSAnSA2	DSAnSA1	DSAnSA0
8 bits	x	x	x	x
16 bits	x	x	x	0
32 bits	x	x	0	0
128 bits	0	0	0	0

5.5.4 DSAnH (n = 0 to 7): DMA Source Address Register H

This 16-bit register forms the 16 higher-order bits of a 32-bit register used to set a source for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DSA7H: FFFF 7466_H, DSA6H: FFFF 7436_H, DSA5H: FFFF 7406_H, DSA4H: FFFF $73D6_H$, DSA3H: FFFF $73A6_H$, DSA2H: FFFF 7376_H, DSA1H: FFFF 7346_H, DSA0H: FFFF 7316_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	DSAn SA28	DSAn SA27	DSAn SA26	DSAn SA25	DSAn SA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DSAn SA23	DSAn SA22	DSAn SA21	DSAn SA20	DSAn SA19	DSAn SA18	DSAn SA17	DSAn SA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
12 to 0	DSAnSA28 to DSAnSA16	DMA source address The 13 higher-order bits of the source address for transfer on channel n are set in these bits. If this register is read during DMA transfer, the value read is the corresponding part of the next address from which data is to be transferred. We recommend access to this register in combination with DSAnL as a 32-bit unit. If the DNSAnNSAV bit of the DNSAnH register is not set (to 1) on completion of DMA transfer, these bits revert to their previous values (those when DMA transfer was started).

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set addresses by access in 32-bit units while the DTSnDTE bit is 0 because data transfer is not possible from an address for which setting is in progress.

5.5.5 DSCn (n = 0 to 7): DMA Source Chip Select Register

This 16-bit register selects the target chip-select area containing the transfer source.

Access This register is readable/writable in 16-bit units.

Address DSC7: FFFF 7468$_H$, DSC6: FFFF 7438$_H$, DSC5: FFFF 7408$_H$, DSC4: FFFF 73D8$_H$, DSC3: FFFF 73A8$_H$, DSC2: FFFF 7378$_H$, DSC1: FFFF 7348$_H$, DSC0: FFFF 7318$_H$

Initial value 0001$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DSCn SCS1	DSCn SCS0	DSCn SCSE
R	R	R	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
2 1 0	DSCnSCS1 DSCnSCS0 DSCnSCSE	DMA source chip select These bits specify an area to be selected as the source for transfer on channel n.

DSCnSCS1	DSCnSCS0	DSCnSCSE	Selected Area
0	0	1	Data-flash, peripheral I/O area
0	1	0	Code-flash, on-chip RAM
Other than the above			Setting prohibited

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set the DSCnSCS0 and DSCnSCSE bits so that only one of them is 1. Operation is not guaranteed if both bits are set to 1.

Caution 3. Be sure to set the DSCnSCS1 bit to 0.

5.5.6 DNSAnL (n = 0 to 7): DMA Next Source Address Register L

This 16-bit register forms the 16 lower-order bits of a 32-bit register used to set the next source for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DNSA7L: FFFF 746C$_H$, DNSA6L: FFFF 743C$_H$, DNSA5L: FFFF 740C$_H$, DNSA4L: FFFF 73DC$_H$, DNSA3L: FFFF 73AC$_H$, DNSA2L: FFFF 737C$_H$, DNSA1L: FFFF 734C$_H$, DNSA0L: FFFF 731C$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNSAn NSA15	DNSAn NSA14	DNSAn NSA13	DNSAn NSA12	DNSAn NSA11	DNSAn NSA10	DNSAn NSA9	DNSAn NSA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNSAn NSA7	DNSAn NSA6	DNSAn NSA5	DNSAn NSA4	DNSAn NSA3	DNSAn NSA2	DNSAn NSA1	DNSAn NSA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	DNSAnNSA15 to DNSAnNSA0	DMA next source address The 16 lower-order bits of the source address for the next transfer on channel n are set in these bits.

Caution DMA transfer of misaligned data is not supported. The 4 lower-order bits of addresses corresponding to the transfer data size are as follows (x indicates any value).

Operation is not guaranteed if settings other than the following are made.

Data Size	DNSAnNSA3	DNSAnNSA2	DNSAnNSA1	DNSAnNSA0
8 bits	x	x	x	x
16 bits	x	x	x	0
32 bits	x	x	0	0
128 bits	0	0	0	0

5.5.7 DNSAnH (n = 0 to 7): DMA Next Source Address Register H

This 16-bit register forms the 16 higher-order bits of a 32-bit register used to set the next source for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DNSA7H: FFFF 746E$_H$, DNSA6H: FFFF 743E$_H$, DNSA5H: FFFF 740E$_H$, DNSA4H: FFFF 73DE$_H$, DNSA3H: FFFF 73AE$_H$, DNSA2H: FFFF 737E$_H$, DNSA1H: FFFF 734E$_H$, DNSA0H: FFFF 731E$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNSAn NSAV	0	0	DNSAn NSA28	DNSAn NSA27	DNSAn NSA26	DNSAn NSA25	DNSAn NSA24
R/W	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNSAn NSA23	DNSAn NSA22	DNSAn NSA21	DNSAn NSA20	DNSAn NSA19	DNSAn NSA18	DNSAn NSA17	DNSAn NSA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DNSAnNSAV	DMA next source address valid This bit controls whether to copy the address from the DMA next source address register to the DMA source address register on completion of DMA transfer. It is cleared once the address has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
12 to 0	DNSAnNSA28 to DNSAnNSA16	DMA next source address These bits specify the 13 higher-order bits of the source address for the next transfer on channel n.

5.5.8 DNSCn (n = 0 to 7): DMA Next Source Chip Select Register

This 16-bit register selects the target area containing the source for the next transfer.

Access This register is readable/writable in 16-bit units.

Address DNSC7: FFFF 7470$_H$, DNSC6: FFFF 7440$_H$, DNSC5: FFFF 7410$_H$, DNSC4: FFFF 73E0$_H$, DNSC3: FFFF 73B0$_H$, DNSC2: FFFF 7380$_H$, DNSC1: FFFF 7350$_H$, DNSC0: FFFF 7320$_H$

Initial value 0001$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNSCn NSCV	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DNSCn NSCS1	DNSCn NSCS0	DNSCn NSCSE
R	R	R	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DNSCnNSCV	DMA next source address select valid This bit controls whether to copy the chip select signal from the DMA next source chip select register to the DMA source chip select register on completion of DMA. It is cleared once the chip select signal has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
2 1 0	DNSCnNSCS1 DNSCnNSCS0 DNSCnNSCSE	DMA next source chip select These bits specify the area to be selected as the transfer source for the next transfer on channel n.

DNSCn NSCS1	DNSCn NSCS0	DNSCn NSCSE	Selected Area
0	0	1	Data-flash, peripheral I/O area
0	1	0	Code-flash, on-chip RAM
Other than the above			Setting prohibited

Caution 1. Set the DNSCnNSCS0 and DNSCnNSCSE bits so that only one of them is 1. Operation is not guaranteed if both bits are set to 1.

Caution 2. Be sure to set the DNSCnNSCS1 bit to 0.

5.5.9 DDAnL (n = 0 to 7): DMA Destination Address Register L

This 16-bit register forms the 16 lower-order bits of a 32-bit register used to set a destination for transfer over a DMA channel.

Access This register is readable/writable in 16-bit units.

Address DDA7L: FFFF 7474$_H$, DDA6L: FFFF 7444$_H$, DDA5L: FFFF 7414$_H$, DDA4L: FFFF 73E4$_H$, DDA3L: FFFF 73B4$_H$, DDA2L: FFFF 7384$_H$, DDA1L: FFFF 7354$_H$, DDA0L: FFFF 7324$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DDAn DA15	DDAn DA14	DDAn DA13	DDAn DA12	DDAn DA11	DDAn DA10	DDAn DA9	DDAn DA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DDAn DA7	DDAn DA6	DDAn DA5	DDAn DA4	DDAn DA3	DDAn DA2	DDAn DA1	DDAn DA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	DDAnDA15 to DDAnDA0	DMA destination address The 16 lower-order bits of the destination address for transfer on channel n are set in these bits. If this register is read during DMA transfer, the value read is the corresponding part of the next address from which data is to be transferred. We recommend access to this register in combination with DDAnH as a 32-bit unit. If the DNDAnNDAV bit in the DNDAnH register is not set (to 1) on completion of DMA transfer, these bits revert to their previous values (those when DMA transfer was started).

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set addresses by access in 32-bit units while the DTSnDTE bit is 0 because data transfer is not possible from an address for which setting is in progress.

Caution 3. If an error occurs in the target for transfer in a cycle of reading for DMA transfer, the corresponding write cycle is not executed but the destination address is still updated.

Caution 4. DMA transfer of misaligned data is not supported. The 4 lower-order bits of addresses corresponding to the transfer data size are as follows (x indicates any value).

Operation is not guaranteed if settings other than the following are made.

Data Size	DDAnDA3	DDAnDA2	DDAnDA1	DDAnDA0
8 bits	x	x	x	x
16 bits	x	x	x	0
32 bits	x	x	0	0
128 bits	0	0	0	0

5.5.10 DDAnH (n = 0 to 7): DMA Destination Address Register H

This 16-bit register forms the 16 higher-order bits of a 32-bit register used to set a destination for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DDA7H: FFFF 7476_H, DDA6H: FFFF 7446_H, DDA5H: FFFF 7416_H, DDA4H: FFFF $73E6_H$, DDA3H: FFFF $73B6_H$, DDA2H: FFFF 7386_H, DDA1H: FFFF 7356_H, DDA0H: FFFF 7326_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	DDAn DA28	DDAn DA27	DDAn DA26	DDAn DA25	DDAn DA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DDAn DA23	DDAn DA22	DDAn DA21	DDAn DA20	DDAn DA19	DDAn DA18	DDAn DA17	DDAn DA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
12 to 0	DDAnDA28 to DDAnDA16	DMA destination address The 13 higher-order bits of the destination address for transfer on channel n are set in these bits. If this register is read during DMA transfer, the value read is the corresponding part of the next address from which data is to be transferred. We recommend access to this register in combination with DDAnL as a 32-bit unit. If the DNDAnNDAV bit in the DNDAnH register is not set (to 1) on completion of DMA transfer, these bits revert to their previous values (those when DMA transfer was started).

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set addresses by access in 32-bit units while the DTSnDTE bit is 0 because data transfer is not possible from an address for which setting is in progress.

Caution 3. If an error occurs in the target for transfer in a cycle of reading for DMA transfer, the corresponding write cycle is not executed but the destination address is still updated.

5.5.11 DDCn (n = 0 to 7): DMA Destination Chip Select Register

This 16-bit register selects the target area containing the transfer destination.

Access This register is readable/writable in 16-bit units.

Address DDC7: FFFF 7478$_H$, DDC6: FFFF 7448$_H$, DDC5: FFFF 7418$_H$, DDC4: FFFF 73E8$_H$, DDC3: FFFF73B8$_H$, DDC2: FFFF 7388$_H$, DDC1: FFFF 7358$_H$, DDC0: FFFF 7328$_H$

Initial value 0001$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DDCn DCS1	DDCn DCS0	DDCn DCSE
R	R	R	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
2 1 0	DDCnDCS1 DDCnDCS0 DDCnDCSE	DMA destination chip select These bits specify an area to be selected as the transfer destination of channel n.

DDCn DCS1	DDCn DCS0	DDCn DCSE	Selected Area
0	0	1	Data-flash, peripheral I/O area
0	1	0	Code-flash, on-chip RAM
Other than the above			Setting prohibited

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Set the DDCnDCS0 and DDCnDCSE bits so that only one of them is 1. Operation is not guaranteed if both bits are set to 1.

Caution 3. Be sure to set the DDCnDCS1 bit to 0.

5.5.12 DNDAnL (n = 0 to 7): DMA Next Destination Address Register L

This 16-bit register forms the 16 lower-order bits of a 32-bit register used to set the next destination for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DNDA7L: FFFF 747C$_H$, DNDA6L: FFFF 744C$_H$, DNDA5L: FFFF 741C$_H$, DNDA4L: FFFF 73EC$_H$, DNDA3L: FFFF 73BC$_H$, DNDA2L: FFFF 738C$_H$, DNDA1L: FFFF 735C$_H$, DNDA0L: FFFF 732C$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNDAn NDA15	DNDAn NDA14	DNDAn NDA13	DNDAn NDA12	DNDAn NDA11	DNDAn NDA10	DNDAn NDA9	DNDAn NDA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNDAn NDA7	DNDAn NDA6	DNDAn NDA5	DNDAn NDA4	DNDAn NDA3	DNDAn NDA2	DNDAn NDA1	DNDAn NDA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	DNDAnNDA15 to DNDAnNDA0	DMA next destination address These bits specify the 16 lower-order bits of the destination address for the next transfer on channel n.

Caution DMA transfer of misaligned data is not supported. The four lower-order bits of addresses corresponding to the transfer data size are as follows (x indicates any value).

Operation is not guaranteed if settings other than the following are made.

Data Size	DNDAnNDA3	DNDAnNDA2	DNDAnNDA1	DNDAnNDA0
8 bits	x	x	x	x
16 bits	x	x	x	0
32 bits	x	x	0	0
128 bits	0	0	0	0

5.5.13 DNDAnH (n = 0 to 7): DMA Next Destination Address Register H

This 16-bit register forms the 16 higher-order bits of a 32-bit register used to set the next destination for transfer over the corresponding DMA channel.

Access This register is readable/writable in 16-bit units.

Address DNDA7H: FFFF 747E$_H$, DNDA6H: FFFF 744E$_H$, DNDA5H: FFFF 741E$_H$, DNDA4H: FFFF 73EE$_H$, DNDA3H: FFFF 73BE$_H$, DNDA2H: FFFF 738E$_H$, DNDA1H: FFFF 735E$_H$, DNDA0H: FFFF 732E$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNDAn NDAV	0	0	DNDAn NDA28	DNDAn NDA27	DNDAn NDA26	DNDAn NDA25	DNDAn NDA24
R/W	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNDAn NDA23	DNDAn NDA22	DNDAn NDA21	DNDAn NDA20	DNDAn NDA19	DNDAn NDA18	DNDAn NDA17	DNDAn NDA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DNDAnNDAV	DMA next destination address valid This bit controls whether to copy the address from the DMA next destination address register to the DMA destination address register on completion of DMA transfer. It is cleared once the address has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
12 to 0	DNDAnNDA28 to DNDAnNDA16	DMA next destination address These bits specify the 13 higher-order bits of the destination address for the next transfer on channel n.

5.5.14 DNDCn (n = 0 to 7): DMA Next Destination Chip Select Register

This 16-bit register selects the target area containing the next destination for transfer.

Access This register is readable/writable in 16-bit units.

Address DNDC7: FFFF 7480$_H$, DNDC6: FFFF 7450$_H$, DNDC5: FFFF 7420$_H$, DNDC4: FFFF 73F0$_H$, DNDC3: FFFF 73C0$_H$, DNDC2: FFFF 7390$_H$, DNDC1: FFFF 7360$_H$, DNDC0: FFFF 7330$_H$

Initial value 0001$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNDCn NDCV	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DNDCn NDCS1	DNDCn NDCS0	DNDCn NDCSE
R	R	R	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DNDCnNDCV	DMA next destination chip select valid This bit controls whether to copy the chip select signal from the DMA next destination chip select register to the DMA destination chip select register on completion of DMA. It is cleared once the chip select signal has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
2 1 0	DNDCnNDCS1 DNDCnNDCS0 DNDCnNDCSE	DMA next destination chip select These bits specify the area to be selected as the transfer destination for the next transfer on channel n.

DNDCn NDCS1	DNDCn NDCS0	DNDCn NDCSE	Selected Area
0	0	1	Data-flash, peripheral I/O area
0	1	0	Code-flash, on-chip RAM
Other than above			Setting prohibited

Caution 1. Set the DNDCnNDCS0 and DNDCnNDCSE bits so that only one of them is 1. Operation is not guaranteed if both bits are set to 1.

Caution 2. Be sure to set the DNDCnNDCS1 bit to 0.

5.5.15 DTCn (n = 0 to 7): DMA Transfer Count Register

This 16-bit register sets the number of times unit transfer on the DMA channel is to proceed.

Access This register is readable/writable in 16-bit units.

Address DCT7: FFFF 7482_H, DCT6: FFFF 7452_H, DCT5: FFFF 7422_H, DCT4: FFFF $73F2_H$, DCT3: FFFF $73C2_H$, DCT2: FFFF 7392_H, DCT1: FFFF 7362_H, DCT0: FFFF 7332_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	DTCnDTC14	DTCnDTC13	DTCnDTC12	DTCnDTC11	DTCnDTC10	DTCnDTC9	DTCnDTC8
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTCnDTC7	DTCnDTC6	DTCnDTC5	DTCnDTC4	DTCnDTC3	DTCnDTC2	DTCnDTC1	DTCnDTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
14 to 0	DTCnDTC14 to DTCnDTC0	DMA transfer count These bits specify the number of times DMA transfer (DMA transfer count) is to proceed on channel n. Reading this register during DMA transfer returns the remaining number of times DMA transfer is to be executed. If the DNTCnNTCV bit of the DNTCn register is not set (to 1), these bits hold the value when DMA transfer has been completed (0000H). If DMA transfer is completed with the DTCTn.DTCTnMLE = 1, next transfer is executed 32, 768 times upon a DMA transfer request.

DTCnDTC[14:0]	Operation
0000_H	Transfer executed 32, 768 times or until completion of transfer
0001_H	Transfer executed once or transfer to be executed once
:	:
$7FFF_H$	Transfer executed 32, 767 times or 32, 767 times of transfer to be executed

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. If an error occurs in the target for transfer in a cycle of reading for DMA transfer, the corresponding write cycle is not executed but the destination address is still updated.

5.5.16 DNTCn (n = 0 to 7): DMA Next Transfer Count Register

This 16-bit register sets the number of times transfer on the DMA channel is to proceed.

Access This register is readable/writable in 16-bit units.

Address DNCT7: FFFF 7484$_H$, DNCT6: FFFF 7454$_H$, DNCT5: FFFF 7424$_H$, DNCT4: FFFF 73F4$_H$, DNCT3: FFFF 73C4$_H$, DNCT2: FFFF 7394$_H$, DNCT1: FFFF 7364$_H$, DNCT0: FFFF 7334$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DNTCn NTCV	DNTCn NDTC14	DNTCn NDTC13	DNTCn NDTC12	DNTCn NDTC11	DNTCn NDTC10	DNTCn NDTC9	DNTCn NDTC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNTCn NDTC7	DNTCn NDTC6	DNTCn NDTC5	DNTCn NDTC4	DNTCn NDTC3	DNTCn NDTC2	DNTCn NDTC1	DNTCn NDTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DNTCnNTCV	DMA next transfer count valid This bit controls whether or not copying of the number of DMA unit transfers (DMA transfer count) from the DMA next transfer count register to the DMA count register is to proceed on completion of DMA transfer. It is cleared once the DMA transfer count has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
14 to 0	DNTCnNDTC14 to DNTCnNDTC0	DMA next transfer count These bits specify the number of times unit DMA transfer is to proceed on channel n.

DNTCnNDTC[14:0]	Operation
0000$_H$	Transfer executed 32, 768 times
0001$_H$	Transfer executed once
:	:
7FFF$_H$	Transfer executed 32, 767 times

5.5.17 DTCCn (n =0 to 7): DMA Transfer Count Compare Register

The value in this 16-bit register is compared with that in the DMA transfer counter and interrupts are controlled accordingly.

Access This register is readable/writable in 16-bit units.

Address DTCC7: FFFF 7486$_H$, DTCC6: FFFF 7456$_H$, DTCC5: FFFF 7426$_H$, DTCC4: FFFF 73F6$_H$, DTCC3: FFFF 73C6$_H$, DTCC2: FFFF 7396$_H$, DTCC1: FFFF 7366$_H$, DTCC0: FFFF 7336$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	DTCCn DTCC14	DTCCn DTCC13	DTCCn DTCC12	DTCCn DTCC11	DTCCn DTCC10	DTCCn DTCC9	DTCCn DTCC8
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTCCn DTCC7	DTCCn DTCC6	DTCCn DTCC5	DTCCn DTCC4	DTCCn DTCC3	DTCCn DTCC2	DTCCn DTCC1	DTCCn DTCC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
14 to 0	DTCCnDTCC 14 to DTCCnDTCC0	DMA transfer count comparison DTCC[14:0] specifies a number of times transfer is to proceed for comparison with the value in the DMA transfer count register for channel n, and an interrupt is generated when the two (DTCn and DTCCn) match. DTCCn can be used as a trigger for setting of the next address. More precisely, an interrupt is generated when unit DMA transfer has been completed the same number of times as the value in the DTCCn register.

DTCCnDTCC[14:0]	Operation
0000$_H$	Not compared
0001$_H$	Interrupt is generated when DTC = 0001$_H$
:	:
7FFF$_H$	Interrupt is generated when DTC = 7FFF$_H$

Caution Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

5.5.18 DTCTn (n = 0 to 7): DMA Transfer Control Register

This 16-bit register sets transfer data size and direction (up or down) for counting of addresses.

Access This register is readable/writable in 16-bit units.

Address DTCT7: FFFF 7488$_H$, DTCT6: FFFF 7458$_H$, DTCT5: FFFF 7428$_H$, DTCT4: FFFF 73F8$_H$, DTCT3: FFFF 73C8$_H$, DTCT2: FFFF 7398$_H$, DTCT1: FFFF 7368$_H$, DTCT0: FFFF 7338$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	DTCTnDS1	DTCTnDS0	DTCTnMLE	0	0	0	0
R	R/W	R/W	R/W	R	R	R	R

7	6	5	4	3	2	1	0
DTCTnSACM1	DTCTnSACM0	DTCTnDACM1	DTCTnDACM0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

Bit Position	Bit Name	Function
14 13	DTCTnDS1 DTCTnDS0	DMA transfer data size These bits specify the DMA transfer data size of channel n.
12	DTCTnMLE	Multi-link enable This bit specifies whether to acknowledge the next DMA transfer request without clearing the DTSnTC bit (to 0) after DMA transfer has been completed. If this bit is set (to 1), the DTSnDTE bit is not cleared upon completion of DMA transfer. Even if the DTSnTC bit is not cleared, DMA transfer is executed if a DMA transfer request is issued. 0: The DTSnDTE bit is cleared upon completion of DMA transfer. 1: The DTSnDTE bit is not cleared upon completion of DMA transfer.

DTCTnDS1	DTCTnDS0	Transfer Data Size
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	128 bits

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7
6</td><td>DTCTnSACM1
DTCTnSACM0</td><td>DMA transfer source address counting direction
These bits specify the direction (up or down) in which counting from the transfer source address for channel n is to proceed.
<table>
<tr><th>DTCTnSACM1</th><th>DTCTnSACM0</th><th>Counting Direction</th></tr>
<tr><td>0</td><td>0</td><td>Incremented</td></tr>
<tr><td>0</td><td>1</td><td>Decremented</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>5
4</td><td>DTCTnDACM1
DTCTnDACM0</td><td>DMA transfer destination address counting direction
These bits specify the direction (up or down) in which counting from the transfer destination address for channel n is to proceed.
<table>
<tr><th>DTCTnDACM1</th><th>DTCTnDACM0</th><th>Counting Direction</th></tr>
<tr><td>0</td><td>0</td><td>Incremented</td></tr>
<tr><td>0</td><td>1</td><td>Decremented</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Caution 1. Writing to these bits is prohibited while DMA transfer is enabled (DTSnDTE bit = 1). Operation is not guaranteed if this is attempted.

Caution 2. Operation is not guaranteed if the DTCTnSACM[1:0] and DTCTnDACM[1:0] bits are set to a prohibited state.

Caution 3. Be sure to set bits 11 and 0 of the DTCTn register to 0.

5.5.19 DTSn (n = 0 to 7): DMA Transfer Status Register

This 8-bit register is for checking the state of DMA transfer control.

Access This register is readable and writable in 1- and 8-bit units.

Address DTS7: FFFF 748A$_H$, DTS6: FFFF 745A$_H$, DTS5: FFFF 742A$_H$,
DTS4: FFFF 73FA$_H$, DTS3: FFFF 73CA$_H$, DTS2: FFFF 739A$_H$,
DTS1: FFFF 736A$_H$, DTS0: FFFF 733A$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
DTSnTC	DTSnDT	0	0	DTSnER	DTSnDR	DTSnSR	DTSnDTE
R/W	R/W	R	R	R	R	R/W	R/W

Bit Position	Bit Name	Function
7	DTSnTC	DMA transfer end status This bit indicates that DMA transfer has been completed. To clear this bit, write 0 to it after reading it as 1. We recommend using bit-manipulation instructions such as CLR1 when writing to this bit. 0: DMA transfer not completed 1: DMA transfer completed
6	DTSnDT	DMA transfer status This bit indicates that a DMA transfer request has been acknowledged and that DMA transfer is in progress. It is not set (to 1) when a DMA transfer request is simply issued. This bit is cleared (to 0) on completion of DMA transfer. If the DTSnDTE bit is 0, this bit can be cleared by the user (and writing to this bit can proceed at the same time as writing to the DTSnDTE bit). 0: DMA transfer request acknowledged 1: DMA transfer in progress
3	DTSnER	DMA transfer error flag This bit indicates that a DMA transfer error has occurred in channel n. It is cleared (to 0) when the DTRC0ERR bit of the DTRC0 register is cleared. Note that this bit is read-only. 0: No DMA transfer error 1: DMA transfer error
2	DTSnDR	Hardware DMA transfer request flag This bit indicates the presence or absence of a hardware DMA transfer request for channel n. The bit is cleared (to 0) when the hardware DMA transfer request is negated. The state of the DTSnDTE bit does not affect the operation of this bit. It is not set (to 1) by a software DMA transfer request, or by a hardware DMA transfer request when a software DMA transfer request is currently selected in the DMA transfer request selection register. Note that this bit is read-only. 0: No hardware DMA transfer request 1: Hardware DMA transfer request
1	DTSnSR	Software DMA transfer request This bit selects a software DMA transfer request. If the software DMA transfer request is currently selected in the DMA transfer request select register, writing 1 to this bit and to the DTSnDTE bit starts DMA transfer. This bit is automatically cleared (to 0) when DMA transfer has been completed. Writing 0 to this bit suspends DMA transfer. 0: No software DMA transfer request 1: Software DMA transfer request

Bit Position	Bit Name	Function
0	DTSnDTE	DMA transfer enable This bit enables or disables DMA transfer. After a 1 is written to this bit, DMA transfer is executed once a DMA transfer request is issued. This bit is automatically cleared (to 0) on completion of DMA transfer if the DTCTnMLE bit is 0. DMA transfer is suspended if 0 is written to this bit during DMA transfer. 0: DMA transfer is disabled. 1: DMA transfer is enabled.

5.6 DMAC Function Details

5.6.1 DMAC Transfer Setting Flow

The following figure shows the flow of DMAC transfer settings.

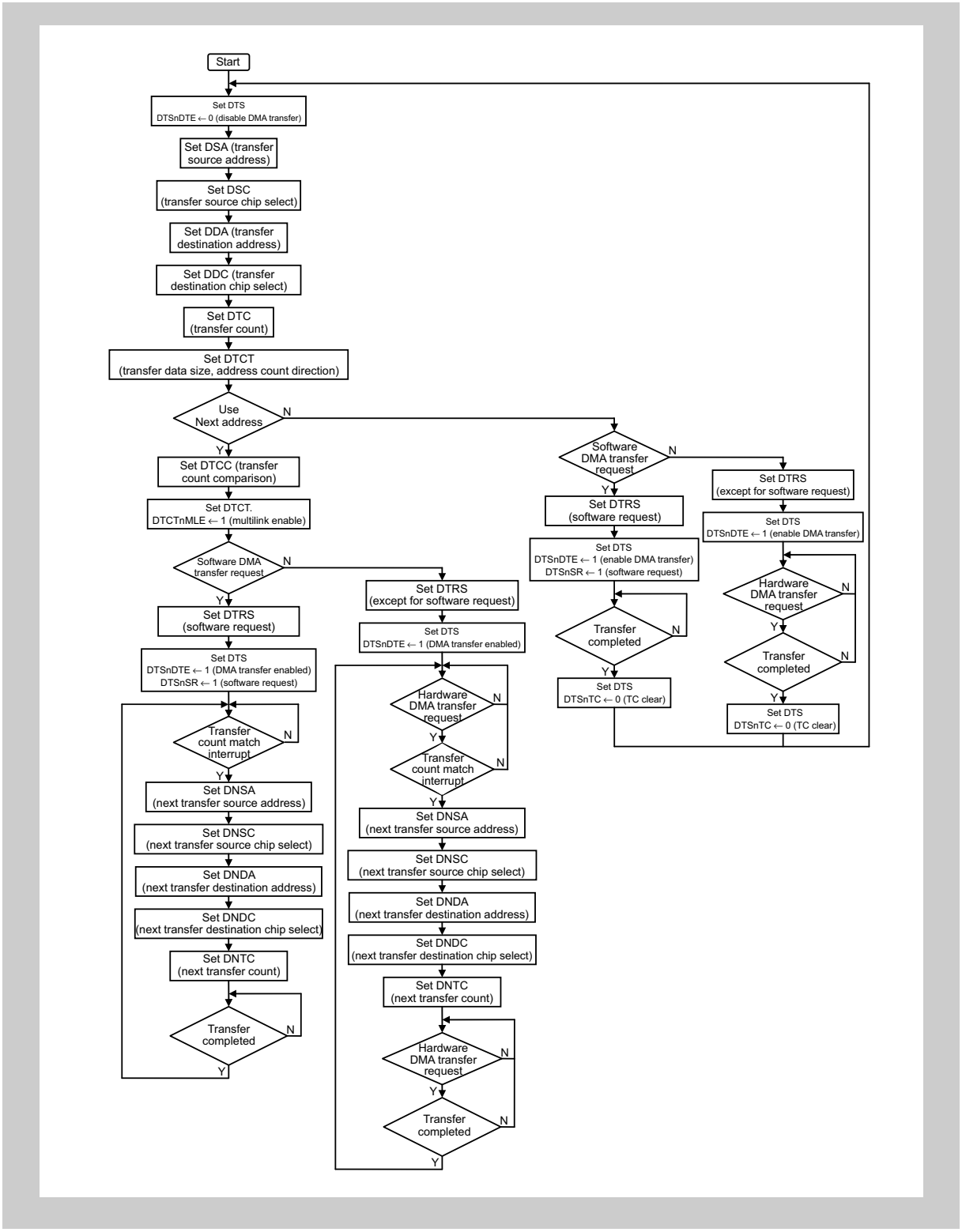


Figure 5-4 DMAC Transfer Setting Flow

5.6.2 DMAC Transfer Modes

The DMAC supports a single-transfer mode and single-step transfer mode as transfer modes.

In either mode, transfer is executed in 2 cycles (dual address transfer) and therefore, a read cycle and a write cycle are generated each time transfer is executed. In the case of 128-bit transfer, the read cycle is generated 4 times and the write cycle is generated 4 times, in that order.

Note that the bus is not locked. Consequently, a CPU-access cycle may interrupt between the read and write cycles, and between the 4 read cycles and 4 write cycles during 128-bit transfer.

(1) Single Transfer Mode (when Hardware DMA Transfer Request Is Generated)

When a hardware DMA transfer request is acknowledged, the amount of data specified as the unit of data transfer (8, 16, 32, or 128 bits) is transferred. Each time transfer of this amount is executed, the bus is released and the DMA controller waits for a DMA transfer request. The acknowledge n signal (n = 7 to 0), which indicates that a hardware DMA transfer request has been acknowledged, is also output at this time.

Transfer is executed once for each acknowledgement of a hardware DMA transfer request. This operation is repeated the number of times specified in DMA transfer count register n (DTCn) (n = 7 to 0).

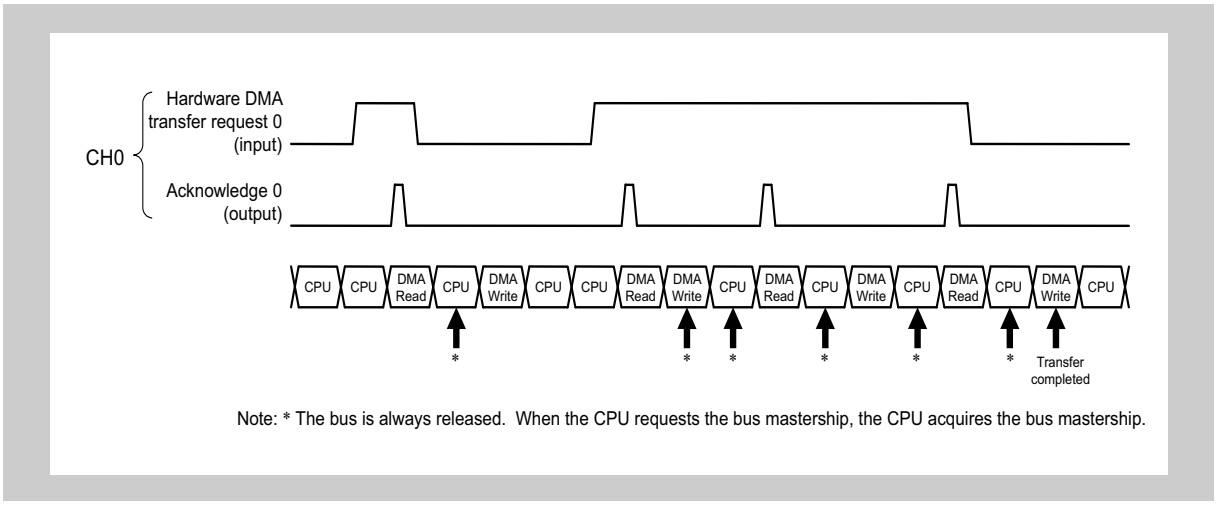


Figure 5-5 Example of Single Transfer (8/16/32 bits)

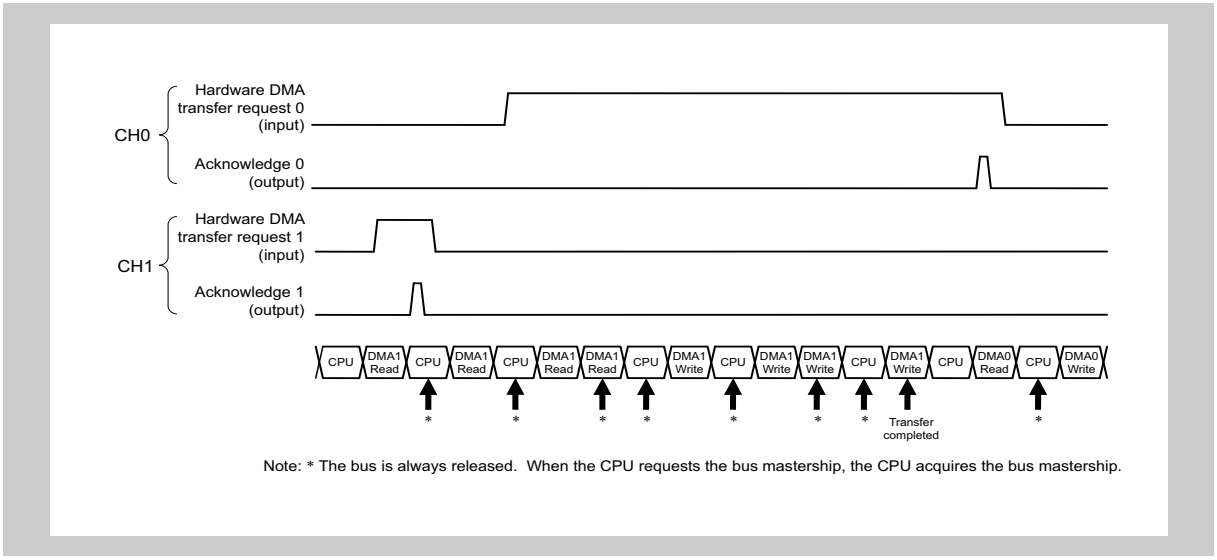


Figure 5-6 Example of Single Transfer (128 Bits, DMA Channel Priority: CH0 (High) > CH1 (Low))

(2) Single-Step Transfer Mode (when Software DMA Transfer Request Is Generated)

When a software DMA transfer request is acknowledged, the amount of data specified as the unit of data transfer (8, 16, 32, or 128 bits) is transferred. Each time transfer of this amount is executed, the bus is released and the DMA controller waits for a DMA transfer request. The acknowledge n signal (n = 7 to 0), which indicates that a hardware DMA transfer request has been acknowledged is not output at this time.

Once a software DMA transfer request has been acknowledged, this operation is repeated the number of times specified in DMA transfer count register n (DTCn) (n = 7 to 0). Priority is determined each time transfer is executed, so a DMA cycle for a channel having a higher priority may interrupt transfer in response to a software DMA request.

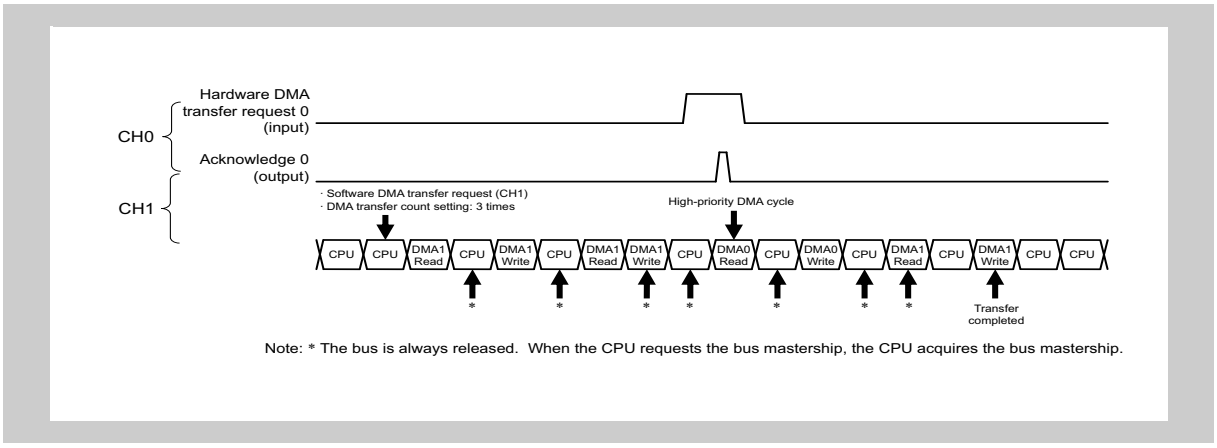


Figure 5-7 Example of Single-Step Transfer (8/16/32 Bits, DMA Channel Priority: CH0 (High) > CH1 (Low))

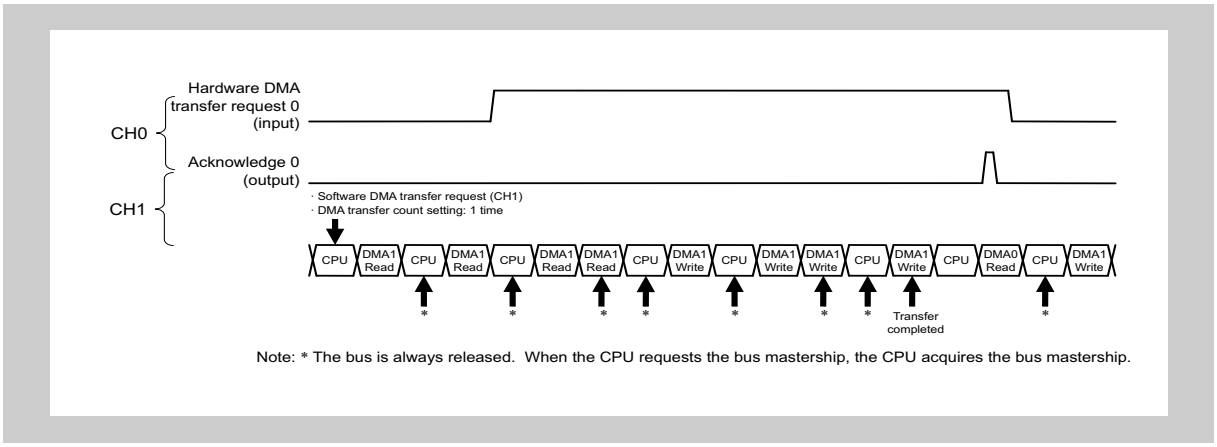


Figure 5-8 Example of Single-Step Transfer (128 Bits, DMA Channel Priority: CH0 (High) > CH1 (Low))

5.6.3 DMAC Channel Priority Control

The priority of each DMAC0 channel is fixed and the order of priority is as follows:

CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7.

If another DMA transfer request with a high priority is generated while transfer is in progress, the request with the higher priority always takes precedence. This is also the case for software DMA requests, since the bus is also released after every DMA cycle in response to a software DMA transfer request.

The following figure shows an example where a next DMA transfer request with higher priority is generated while DMA transfer is in progress.

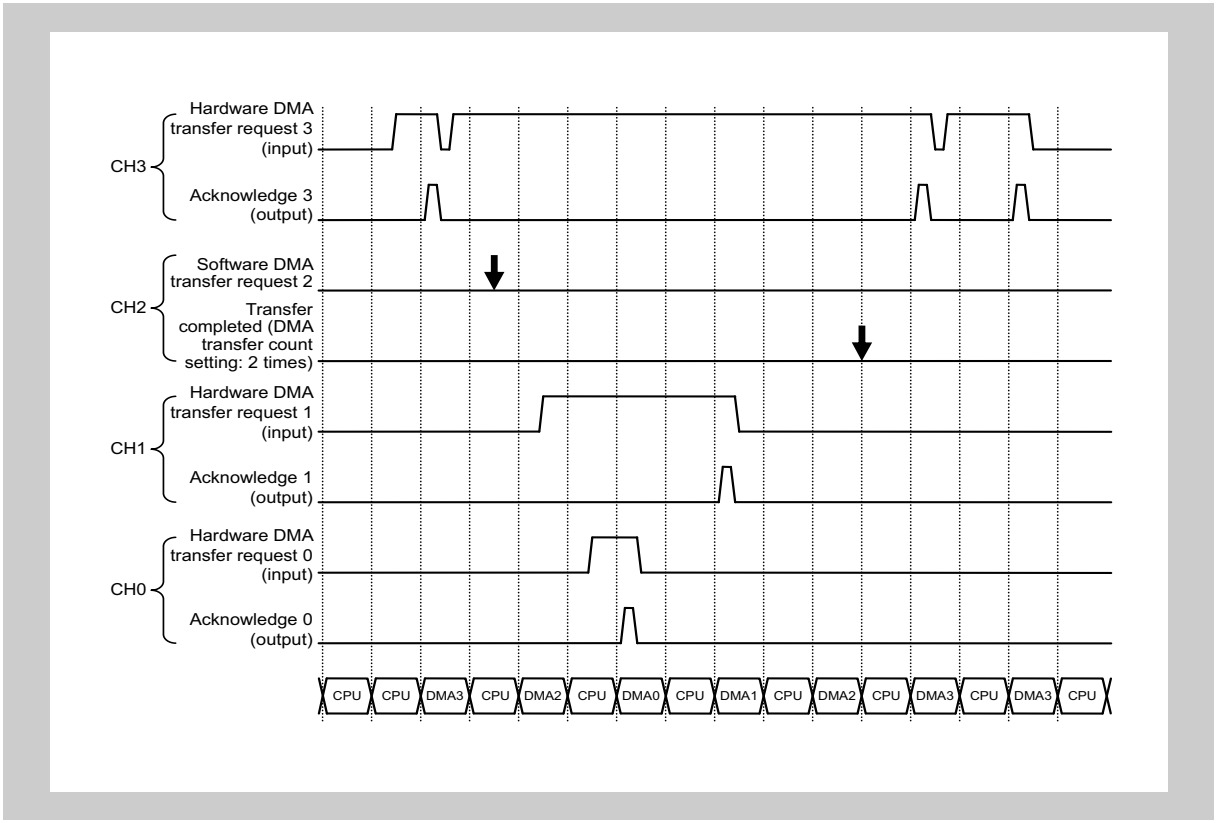


Figure 5-9 Example of Priority Control

5.6.4 Conditions for Validity of DMA Transfer Requests

Whether a request for DMA transfer on channel n is or is not acknowledged depends on the setting of the DTRC0ERR and DTRC0ADS bits of the DMA transfer request control register (DTRC0), the DTCTnMLE bit of the DMA transfer control register (DTCTn), and the DTSnTC and DTSnDTE bits of the DMA transfer status register (DTSn). The following table lists the relationship between the settings of these bits and whether a DMA transfer request is or is not acknowledged.

Table 5-7 Conditions for Validity of Requests for DMA Transfer on Channel n

Register.Bit Name	DTSn. DTSnDTE	DTSn. DTSnTC	DTCTn. DTCTnMLE	DTRC0. DTRC0ERR	DTRC0. DTRC0ADS	DMA Transfer Request
When DMA transfer is disabled	0	X	X	X	X	Invalid
When DMA transfer error occurs	X	X	X	1	X	Invalid
When DMA transfer is suspended	X	X	X	X	1	Invalid
When DMA transfer is completed (multilink disabled)	X	1	0	X	X	Invalid
When DMA transfer is completed/not completed (multilink enabled)	1	X	1	0	0	Valid
When DMA transfer is enabled	1	0	0	0	0	Valid

Note n = 0 to 7

5.6.5 Next Address Function

(1) Next Address Setting Registers

These registers are used to set the transfer information for the next transfer in advance while DMA transfer is in progress. This information is copied to the corresponding registers at the start of the last DMA cycle for the current transfer. The registers are as follows.

- DMA next source address register (DNSAnH/DNSAnL)
- DMA next source chip select register (DNSCn)
- DMA next destination address register (DNDAnH/DNDAnL)
- DMA next destination chip select register (DNDCn)
- DMA next transfer count register (DNTCn)

The most significant bit of each register for higher-order address bits is a “valid” bit that can be used to select whether or not to copy the transfer information to the “current” registers at the start of the last DMA cycle for the current transfer. Once the transfer information to be transferred next is copied to the current register, the “valid” bit is cleared.

(2) Processing upon DMA Transfer Completion when Using Next Address Function

Normally, upon completion of DMA transfer, the DMA transfer enable bit (DTSnDTE) is cleared at the same time the DMA transfer completion status bit (DTSnTC) of the DMA transfer status register (DTSn) is set, and subsequent DMA transfer requests are no longer acknowledged. However, if the multilink enable bit (DTCTnMLE) is set, DTSnDTE is not cleared and DMA transfer requests can be acknowledged even if DTSnTC is set.

Therefore, setting DTCTnMLE eliminates the steps of clearing DTSnTC and setting DTSnDTE upon completion of DMA transfer when the next address function is in use.

(3) Timing of Next-Address Settings

A new value can be written to a next address setting register at any time. However, to prevent a conflict between copying to the “current” register and writing by the user, make sure that setting of the next address setting register is completed before the start of the last DMA cycle for the current transfer.

We recommend that you use the DMA transfer count match interrupt as the trigger for setting the next address setting register. In this case, set the DMA transfer count compare register (DTCCn) so as to secure the time required for setting in the next address setting register.

5.6.6 Suspending/Resuming DMA Transfer

(1) Suspending or Resuming DMA Transfer for All Channels through Software

Subsequent DMA transfer can be suspended by setting the DMA transfer abort bit (DTRC0ADS) of the DMA transfer request control register (DTRC0).

When this is done during a DMA cycle, DMA transfer is suspended on completion of the ongoing DMA cycle. Note that the DMA transfer enable bit (DTSnDTE) and the software DMA transfer request bit (DTSnSR) of the DMA transfer status register (DTSn) are not cleared.

To resume a DMA transfer suspended in this way, clear the DTRC0ADS bit. To end DMA transfer, clear the DMA transfer request while the DTSnDTE bit is clear.

(2) Suspending or Resuming DMA Transfer by Using DMA Transfer Enable Bit (DTSnDTE)

Subsequent DMA transfer can be suspended by clearing the DMA transfer enable bit (DTSnDTE) of the DMA transfer status register (DTSn).

When this is done during a DMA cycle, DMA transfer is suspended on completion of the ongoing DMA cycle. Note that the software DMA transfer request bit (DTSnSR) for the DTS is not cleared.

To resume a DMA transfer suspended in this way, set the DTSnDTE bit. To end DMA transfer, clear the DMA transfer request while the DTSnDTE bit is clear.

(3) Suspending or Resuming DMA Transfer by Using Software DMA Transfer Request Bit (DTSnSR)

Subsequent DMA transfer can be suspended by clearing the software DMA transfer request bit (DTSnSR) of the DMA transfer status register (DTSn).

When this is done during a DMA cycle, DMA transfer is suspended on completion of the ongoing DMA cycle.

To resume a DMA transfer suspended in this way, set the DTSnSR bit.

5.6.7 Error Responses

(1) Error Response Leading to Suspension of DMA Transfer

When an error occurs at the DMA transfer source or transfer destination, DMAC sets the DMA transfer abort bit (DTRC0ADS) of the DMA transfer request control register (DTRC0) to suspend subsequent DMA transfer. At the same time, the DMA transfer error status bit (DTRC0ERR) is set and a SysError exception is generated for the CPU. Once the user has confirmed that DTRC0ERR has been set, the DMA transfer error flag (DTSnER) of the DMA transfer status register (DTSn) can be used to identify the channel where the error occurred.

In this case, note that if the error response is acknowledged within the read cycle, the write cycle does not proceed but the transfer address and transfer count are updated.

(2) Canceling Transfer Suspension Due to Error Response

DMA transfer suspension can be reversed by clearing the DMA transfer abort bit (DTRC0ADS) and DMA transfer error status bit (DTRC0ERR) of the DMA transfer request control register (DTRC0).

Clear the DMA transfer enable bit (DTSnDTE) of the DMA transfer status register (DTSn) in advance, so that DMA transfer is not resumed after reversal of suspension. In the case of a software DMA transfer request, also clear the software DMA transfer request bit (DTSnSR).

5.7 DTFR Functions

Each DMA trigger factor register (DTFR) is used to select DMA trigger sources from among interrupt signals and to request DMA transfer by the DMAC. A DTFRn register (n = 7 to 0) is included to select the signals to be used for DMA transfer requests for each channel from among the m = 108 input interrupt signals.

Caution If transfer is enabled (DTSn.DTE = 1), writing to a DTFR might lead to the simultaneous start of transfer. To avoid this, use the DMA request clearing register (DRQCLR) to clear the transfer activating source flag just before transfer is enabled.

5.7.1 Features

Number of transfer sources DMA transfer requests (for 8 channels) are selected from among the m = 108 interrupt signals.

DMAC interface The DMA transfer request signal n (n = 7 to 0) is output.

The DMA transfer request signal n is cleared by an acknowledge signal from DMA.

CPU interface The last transfer signal from DMA is output as a CPU interrupt signal.

Clearing of transfer requests Register access to clear transfer request signals being sent to the DMA block is possible.

Confirming transfer requests Register access to check transfer request signals sent to the DMA block is possible.

5.8 DTFR Control Registers

5.8.1 DTFRn (n = 0 to 7): DMA Trigger Factor Register

This 16-bit register selects an activating source to control the start of DMA transfer operations on the corresponding channel.

Access This register is readable/writable in 16-bit units.

Address DTFR0: FFFF 7B00$_H$, DTFR1: FFFF 7B02$_H$, DTFR2: FFFF 7B04$_H$, DTFR3: FFFF 7B06$_H$, DTFR4: FFFF 7B08$_H$, DTFR5: FFFF 7B0A$_H$, DTFR6: FFFF 7B0C$_H$, DTFR7: FFFF 7B0E$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DTFRn REQEN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	IFCn[6:0]						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	DTFRnREQEN	This bit enables or disables operation of the DMA activating source selector for channel n. 1: The source selector operates. 0: The source selector is stopped and corresponding DMA transfer request (DMARQ) is not issued. The settings of IFC6 to IFC0 are valid. Requests are always sampled.
6 to 0	IFCn6 to IFCn0	These bits select the transfer source. The set values are shown in table 5-1, List of DMA Trigger Sources.

5.8.2 DRQCLR: DMA Request Clear Register

This 16-bit register is used to clear activating sources for DAM transfer.

Access This register is readable/writable in 16-bit units.

Address FFFF 7B40$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
RQCR7	RQCR6	RQCR5	RQCR4	RQCR3	RQCR2	RQCR1	RQCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
7	RQCR7	This is a control bit for clearing the DMA request check register (DRQSTR.RQST7 bit). Setting 1 in this bit clears a transfer request held for channel 7 to 0.
6	RQCR6	This is a control bit for clearing the DMA request check register (DRQSTR.RQST6 bit). Setting 1 in this bit clears a transfer request held for channel 6 to 0.
5	RQCR5	This is a control bit for clearing the DMA request check register (DRQSTR.RQST5 bit). Setting 1 in this bit clears a transfer request held for channel 5 to 0.
4	RQCR4	This is a control bit for clearing the DMA request check register (DRQSTR.RQST4 bit). Setting 1 in this bit clears a transfer request held for channel 4 to 0.
3	RQCR3	This is a control bit for clearing the DMA request check register (DRQSTR.RQST3 bit). Setting 1 in this bit clears a transfer request held for channel 3 to 0.
2	RQCR2	This is a control bit for clearing the DMA request check register (DRQSTR.RQST2 bit). Setting 1 in this bit clears a transfer request held for channel 2 to 0.
1	RQCR1	This is a control bit for clearing the DMA request check register (DRQSTR.RQST1 bit). Setting 1 in this bit clears a transfer request held for channel 1 to 0.
0	RQCR0	This is a control bit for clearing the DMA request check register (DRQSTR.RQST0 bit). Setting 1 in this bit clears a transfer request held for channel 0 to 0.

Note Writing 0 to bits 15 to 0 is ignored.

5.8.3 DRQSTR: DMA Request Check Register

This 16-bit register is used to check the state of DMA transfer requests.

Access Only reading is possible and this must be in 16-bit units.

Address FFFF 7B44$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
RQST7	RQST6	RQST5	RQST4	RQST3	RQST2	RQST1	RQST0
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
7	RQST7	1: Request issued (DMA transfer request signal 7 is 1) 0: No request (DMA transfer request signal 7 is 0)
6	RQST6	1: Request issued (DMA transfer request signal 6 is 1) 0: No request (DMA transfer request signal 6 is 0)
5	RQST5	1: Request issued (DMA transfer request signal 5 is 1) 0: No request (DMA transfer request signal 5 is 0)
4	RQST4	1: Request issued (DMA transfer request signal 4 is 1) 0: No request (DMA transfer request signal 4 is 0)
3	RQST3	1: Request issued (DMA transfer request signal 3 is 1) 0: No request (DMA transfer request signal 3 is 0)
2	RQST2	1: Request issued (DMA transfer request signal 2 is 1) 0: No request (DMA transfer request signal 2 is 0)
1	RQST1	1: Request issued (DMA transfer request signal 1 is 1) 0: No request (DMA transfer request signal 1 is 0)
0	RQST0	1: Request issued (DMA transfer request signal 0 is 1) 0: No request (DMA transfer request signal 0 is 0)

5.9 DTS Function

The data transfer service (DTS) does not include registers for setting transfer information (TI). Consequently, each time a transfer request is generated, the TI corresponding to the request number is fetched (read from on-chip RAM), and the transfer request for the TI is then sent to the DMAT, which executes data transfer, etc.

5.9.1 Features

Number of channels

All channels are freely available (112 channels).

Units of transfer

8 bits

16 bits

32 bits

Data for transfer

Fixed to little endian

Misaligned data are not supported.

Maximum transfer count

When the DTS transfer count register is used as a 16-bit register: 65535 times

When the DTS transfer count register is split into two 8-bit registers: 255 × n: (n indicates no limitation)

Therefore, the number of transfers is not limited.

Channel priority control

No priority control function is provided (control must be done outside of DTS).

Targets for transfer

Code-flash

On-chip RAM

Data-flash

Peripheral I/O area

Chain function

With chaining, a single request is able to start the DTS multiple times. After writing back the TI, different TIs are continuously fetched so that data transfer (and other DTS functions if selected) can proceed.

Transfer type

- Two-cycle transfer (dual-address transfer) function

  After the TI is fetched, data are read from the transfer source and then written to the transfer destination. After that, the updated TI is written back. Operation is thus the same as in DMA transfer, except for fetching and writing back of the TI. Since the bus remains unlocked during all transfer cycles for the DTS, operation can be interrupted to give bus access to the CPU.

- Comparison

  After the TI is fetched, data are read from the comparison source and the comparison destination and then compared. After that, the updated TI is written back. The result of comparison (true or false) can then be used to select whether to output an interrupt to the CPU or to continue using the chain function for further transfer.

- Flag checking (bit checking)

  This function is used to check whether the values of data bits are “1” or “0” as required.
  After the TI is fetched, the mask for use in flag checking and the data containing the flags to be checked are read, after which the values of the non-masked bits are determined. After that, the updated TI is written back.
  The results of flag checking (true or false) can then be used to select whether to output an interrupt to the CPU or to continue using the chain function for further transfer.

Transfer count register split mode

- Split

  In this mode, the 16-bit transfer count register is split into higher- and lower-order eight-bit counters. The lower-order byte is used as a transfer counter and the higher-order byte is used as a transfer-count holding register. Although the maximum number of transfers is 255, values from the higher-order byte can be copied to the lower-order byte when the value of the latter reaches 0, so continuous transfer can proceed without the need to remake the transfer counter settings.

- Combined

  All 16 bits of the transfer counter are used as a single counter. The maximum number of unit transfers is 65,535.

Transfer mode

- Single-transfer mode

  In this case, each DTS transfer request corresponds to one DTS cycle, a further single DTS cycle is performed in response to the next DTS transfer request, and so on. This operation is repeated until the value in the transfer counter reaches 0.

- Block transfer mode

  When a DTS transfer request is generated, the number of DTS cycles specified in the transfer count register is performed.
  Other DTS transfer requests are not accepted until the number of DTS cycles specified in the transfer count register is completed. However, because the bus is not locked, a CPU-access cycle may interrupt.

Transfer address control

Incrementation

Decrementation

Fixed

Transfer error support	When the system bus generates an error response, the current DTS cycle is aborted and an error flag is set. Also, a SysError exception is output for the CPU.
Interrupt output	An interrupt source signal is output on completion of the number of DTS transfers specified in the bus-cycle counter, or depending on the results of flag checking or comparison (this is handled by the DTSFSL block).
Suspending DTS transfer	This function supports suspending DTS transfer by software but not hardware.

5.10 DTS Control Registers

The following registers are used to set up DTS operations.

Registers of the DTS include those that are accessible from the CPU (register group A) and internal registers: these are not accessible by the CPU (register group B). After transfer information is read from the on-chip RAM, it is stored in the internal registers.

Register group B is comprised of DTS0SAR, DTS0DAR, DTS0TCEA, DTS0CIR, DTS0SCS, DTS0DCS, DTS0ESCS, and DTS0EDCS.

Table 5-8 List of DTS Control Registers

	Address*	Symbol	Register Name	R/W	Bit Width for Operations			Initial Value
					8	16	32	
Register Group A	$FFFF7C00_H$	DTS0TSR	DTS transfer status register	R	√	–	–	00_H
	$FFFF7C04_H$	DTS0TRC	DTS transfer request control register	R/W	√	–	–	00_H
	$FFFF7C08_H$	DTS0ICR	DTS initialization control register	R/W	–	√	–	0000_H
	$FFFF7C08_H$	DTS0ICH	DTS initialization channel number register	R/W	√	–	–	00_H
	$FFFF7C09_H$	DTS0ITR	DTS initialization channel trigger register	R/W	√	–	–	00_H
	$FFFF7C0C_H$	DTS0BTR	DTS base table register	R/W	–	–	√	00000000_H
	$FFFF7C10_H$	DTS0BVR	DTS base vector register	R/W	–	–	√	00000000_H
	$FFFF7C14_H$	DTS0ACR	DTS active channel register	R	–	√	–	0000_H
	$FFFF7C18_H$	DTS0HST	DTS TI hold status register	R	√	–	–	00_H
	$FFFF7C20_H$	DTS0HC	DTS TI hold channel number register	R	–	–	√	00000000_H
	$FFFF7C20_H$	DTS0HC0	DTS hold channel register 0	R	√	–	–	00_H
	$FFFF7C21_H$	DTS0HC1	DTS hold channel register 1	R	√	–	–	00_H
	$FFFF7C22_H$	DTS0HC2	DTS hold channel register 2	R	√	–	–	00_H
	$FFFF7C23_H$	DTS0HC3	DTS hold channel register 3	R	√	–	–	00_H
Register Group B	–	DTS0SAR	DTS source address register	–	–	–	–	00000000_H
	–	DTS0DAR	DTS destination address register	–	–	–	–	00000000_H
	–	DTS0TCEA	DTS transfer counter or else address register	–	–	–	–	0000_H
	–	DTS0CIR	DTS control information register	–	–	–	–	0000_H
	–	DTS0SCS	DTS source address count size register	–	–	–	–	00_H
	–	DTS0DCS	DTS destination address count size register	–	–	–	–	00_H
	–	DTS0ECSRA	DTS extension address count size/ repeat address register	–	–	–	–	00000000_H

Note * In access to unmapped addresses by the DTS, writing is ignored and 0 is returned in response to reading.

5.10.1 DTS0TSR: DTS Transfer Status Register [Register Group A]

This register is used to check the state of DTS transfer.

Access Only reading is possible and this must be in 8-bit units.

Address FFFF 7C00$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	DTS0STPE	DTS0STPU
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
1	DTS0STPE	DTS stopped by error response (DTS transfer stopped by an error response) This bit indicates that DTS transfer has been stopped by an error response received from a target for transfer. When an error response is received, this bit is set to 1 and a SysError exception is issued by the CPU to stop the DTS transfer. When this bit is 1, requests for transfer are not accepted. This is a read-only bit. To clear this bit, write 1 to the DTS0ECL bit in the DTS0TRC register.
0	DTS0STPU	DTS stopped by user request (DTS transfer suspended by user request) The setting 1 for this bit indicates that DTS transfer has been suspended by user request. This bit is set to 1 by writing 1 to the DTS0UST bit in the DTS0TRC register, which suspends transfer. When this bit is 1, requests for transfer are not accepted. This is a read-only bit. To clear this bit, write 1 to the DTS0UCL bit in the DTS0TRC register.

5.10.2 DTS0TRC: DTS Transfer Request Control Register [Register Group A]

This 8-bit register is used to stop DTS transfer and to clear the DTS0STPE and DTS0STPU bits.

Access This register is readable/writable in 8-bit units.

Address FFFF 7C04$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
DTS0UST	0	0	0	0	0	DTS0ECL	DTS0UCL
W	R	R	R	R	R	W	W

Bit Position	Bit Name	Function
7	DTS0UST	DTS user request stop trigger (DTS stop request trigger bit) This is the transfer-stop request trigger for all DTS channels. When 1 is written to this bit, the DTS0STPU bit in DTS0TSR is set to 1 and DTS transfer is suspended. This bit is always read as 0, and writing 0 to this bit is ignored. When this bit and DTS0UCL are set to 1 at the same time, the setting of this bit is effective.
1	DTS0ECL	DTS dts0stpe clearing trigger (DTS0STPE clearing request trigger bit) This is the clearing request trigger for the DTS0STPE bit in DTS0TSR. When this bit is set to 1, the DTS0STPE bit in DTS0TSR is cleared to 0 so that requests for DTS transfer can be accepted. This bit is always read as 0, and writing 0 to this bit is ignored.
0	DTS0UCL	DTS dts0stpu clearing trigger (DTS0STPU clearing request trigger bit) This is the clearing request trigger for the DTS0STPU bit in DTS0TSR. When this bit is set to 1, the DTS0STPU bit in DTS0TSR is cleared to 0 so that requests for DTS transfer can be accepted. If transfer in progress has been suspended, the transfer is restarted. This bit is always read as 0, and writing 0 to this bit is ignored.

5.10.3 DTS0ICR: DTS Initialization Control Register [Register Group A]

This register controls initialization of DTS transfer.

Access This register is readable/writable in 16-bit units.

However, when the higher- and lower-order 8-bit halves of the DTS0ICR register are in use as the DTS0ITR register and DTS0ICH register, respectively, this register is readable and writable in 8-bit units.

Address DTS0ICR: FFFF 7C08$_H$

DTS0ITR: FFFF 7C09$_H$

DTS0ICH: FFFF 7C08$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

- DTS0ICR

15	14	13	12	11	10	9	8
0	0	0	0	0	DTS0ICS	DTS0HIT	DTS0TIT
R	R	R	R	R	R	W	W

7	6	5	4	3	2	1	0
0	DTS0ICH[6:0]						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DTS0ITR

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0ICS	DTS0HIT	DTS0TIT
R	R	R	R	R	R	W	W

- DTS0ICH

7	6	5	4	3	2	1	0
0	DTS0ICH[6:0]						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
10	DTS0ICS	DTS initialization continuous status (transfer stop status flag) This status flag indicates that the DTS transfer is currently stopped. When this flag is 1, writing to this register is prohibited (attempts at writing to this register are ignored). When DTS0HIT or DTS0TIT in this register is set to 1, this bit is set to 1. This bit is cleared on completion of processing for transfer-stop requested by using DTS0HIT or DTS0TIT in this register. 0: Normal state 1: Transfer is currently stopped.
9	DTS0HIT	DTS hold initialization trigger (trigger for transfer suspension and TI hold-buffer initialization) This is the trigger bit used to request suspension of the current DTS transfer and clearing of all TIs held by the DTS. When 1 is written to this bit, all held TIs are cleared once the current DTS cycle is complete. While the control circuit is waiting for this, the DTS0ICS bit is set to 1. This bit is always read as 0.
8	DTS0TIT	DTS transfer initialization trigger (trigger for transfer suspension) This is the trigger bit used to request suspension of DTS transfer on the channel specified by the DTS0ICH bits in this register. When 1 is written to this bit, the channel number on which DTS transfer is currently proceeding is compared with the DTS0ICH bits once the current DTS cycle is complete. If the two match, all subsequent transfer is stopped. While the control circuit is waiting for completion of the DTS cycle, the DTS0ICS bit is set to 1. If the TI-hold bit for the specified channel is set, the TI-hold bit is also cleared. This bit is always read as 0.
6 to 0	DTS0ICH6 to DTS0ICH0	DTS initialization channel (initialization stop request channel) This bit specifies the channel for which initialization is to be aborted by the DTS0TIT bit. A 1 can be written to these bits at the same time as the writing of 1 to the DTS0HIT or DTS0TIT bits in this register.

5.10.4 DTS0BTR: DTS Base Table Register [Register Group A]

This 32-bit register is used to set the address where the table used for DTS transfer starts.

Access This register is readable/writable in 32-bit units.

Address $FFFF7C0C_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
0	0	0	DTS0BTR[28:24]				
R	R	R	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
DTS0BTR[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
DTS0BTR[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTS0BTR[7:1]							0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit Position	Bit Name	Function
28 to 1	–	This register is for setting the address where the table that is used by the DTS to detect the locations of TIs starts. Select a location in the on-chip RAM area. Bits 31 to 29 and 0 are fixed to 0. The DTS uses DTS0BVR and this register to calculate the TI locations corresponding to each of the channels.

5.10.5 DTS0BVR: DTS Base Vector Register [Register Group A]

This 32-bit register is used to set the address where the vectors for use in DTS transfer start.

Access This register is readable/writable in 32-bit units.

Address FFFF 7C10$_H$

Initial value 00000000$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
0	0	0	DTS0BVR[28:24]				
R	R	R	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
DTS0BVR[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
DTS0BVR[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTS0BVR[7:2]						0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit Position	Bit Name	Function
28 to 2	–	This register is for setting the address where the vectors that are used by the DTS to detect the locations of TIs starts. Select a location in the on-chip RAM area. Bits 31 to 29, 1, and 0 are fixed to 0. When a DTS request is generated, a 16-bit vector address is read from the address in the table calculated from DTS0BTR + channel no. × 2. The TI's address is determined by adding the read value to the value in this register.

5.10.6 DTS0ACR: DTS Active Channel Register [Register Group A]

This 16-bit register indicates the state of DTS transfer.

Access Only reading is possible and this must be in 16-bit units.

Address FFFF 7C14$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	DTS0ACT
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	DTS0CH						
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
8	DTS0ACT	DTS active (the state of execution by the DTS) This bit indicates the state of execution by the DTS. 0: The DTS is not operating. 1: The DTS is operating.
6 to 0	DTS0CH	Channel (active channel) These bits indicate the active channel when the DTS0ACT bit is 1. When the DTS0ACT bit is 0, the value is the number of the channel on which the immediately prior transfer was executed.

5.10.7 DTS0HST: DTS TI Hold Status Register [Register Group A]

This 8-bit register indicates the state of hold buffer.

Access Only reading is possible and this must be in 8-bit units.

Address FFFF 7C18$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	DTS0TON3	DTS0TON2	DTS0TON1	DTS0TON0
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
3	DTS0TON3	DTS TI-hold ON 3 (TI hold 3 ON) This bit checks whether TI hold buffer 3 currently holds TI. 0: TI is not currently held. 1: TI is currently held.
2	DTS0TON2	DTS TI-hold ON 2 (TI hold 2 ON) This bit checks whether TI hold buffer 2 currently holds TI. 0: TI is not currently held. 1: TI is currently held.
1	DTS0TON1	DTS TI-hold ON 1 (TI hold 1 ON) This bit checks whether TI hold buffer 1 currently holds TI. 0: TI is not currently held. 1: TI is currently held.
0	DTS0TON0	DTS Ti-hold ON 0 (TI hold 0 ON) This bit checks whether TI hold buffer 0 currently holds TI. 0: TI is not currently held. 1: TI is currently held.

5.10.8 DTS0HC: DTS TI Hold Channel Number Register [Register Group A]

This 32-bit register indicates the number of a channel for which the TI is held.

Access This register is read-only, and is read as a 32-bit unit.
However, when bits 30 to 24, 22 to 16, 14 to 8, and 6 to 0 in the DTS0HC register are in use as the DTS0HC3, DTS0HC2, DTS0HC1, and DTS0HC0 registers respectively, these registers are still read-only, but are readable in 8-bit units.

Address DTS0HC: FFFF 7C20$_H$
DTS0HC0: FFFF 7C20$_H$, DTS0HC1: FFFF 7C21$_H$, DTS0HC2: FFFF 7C22$_H$, DTS0HC3: FFFF 7C23$_H$

Initial value 00000000$_H$

This register is initialized by a reset from any source.

- DTS0HC

31	30	29	28	27	26	25	24
0	DTS0HC[30:24]						
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	DTS0HC[22:16]						
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	DTS0HC[14:8]						
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	DTS0HC[6:0]						
R	R	R	R	R	R	R	R

- DTS0HC3

7	6	5	4	3	2	1	0
0	DTS0HC[30:24]						
R	R	R	R	R	R	R	R

- DTS0HC2

7	6	5	4	3	2	1	0
0	DTS0HC[22:16]						
R	R	R	R	R	R	R	R

- DTS0HC1

7	6	5	4	3	2	1	0
0	DTS0HC[14:8]						
R	R	R	R	R	R	R	R

- DTS0HC0

7	6	5	4	3	2	1	0
0	DTS0HC[6:0]						
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
30 to 24	DTS0HC30 to DTS0HC24	These bits store the channel number corresponding to the TI being held in TI hold buffer 3.
22 to 16	DTS0HC22 to DTS0HC16	These bits store the channel number corresponding to the TI being held in TI hold buffer 2.
14 to 8	DTS0HC14 to DTS0HC8	These bits store the channel number corresponding to the TI being held in TI hold buffer 1.
6 to 0	DTS0HC6 to DTS0HC0	These bits store the channel number corresponding to the TI being held in TI hold buffer 0.

5.10.9 DTS0SAR: DTS Source Address Register [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU.

Access —

Address —

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
IMM31	IMM30	IMM29	DTS0SAR[28:24] (IMM28-24)				

23	22	21	20	19	18	17	16
DTS0SAR[23:16] (IMM23-16)							

15	14	13	12	11	10	9	8
DTS0SAR[15:8] (IMM15-8)							

7	6	5	4	3	2	1	0
DTS0SAR[7:0] (IMM7-0)							

Bit Position	Bit Name	Function
31 to 29	IMM31 to IMM29	Immediate (immediate bits) When the immediate function is in use, these bits and the DTS0SAR bits are used in combination as a 32-bit register.
28 to 0	DTS0SAR28 to DTS0SAR0	DTS source address register (source address register) These bits hold the source address. When the immediate function is in use, however, they function as IMM28 to IMM0.

Caution DTS transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).

Data Size	DTS0SAR1	DTS0SAR0
8 bits	x	x
16 bits	x	0
32 bits	0	0

5.10.10 DTS0DAR: DTS Destination Address Register [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU.

Access —

Address —

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
0	0	0	DTS0DAR[28:24]				

23	22	21	20	19	18	17	16
DTS0DAR[23:16]							

15	14	13	12	11	10	9	8
DTS0DAR[15:8]							

7	6	5	4	3	2	1	0
DTS0DAR[7:0]							

Bit Position	Bit Name	Function
28 to 0	DTS0DAR28 to DTS0DAR0	Destination Address Register (destination address register) These bits hold the destination address. Bits 31 to 29 are fixed to 0.

Caution DTS transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).

Operation is not guaranteed if a setting other than the following is made.

Data Size	DTS0DAR1	DTS0DAR0
8 bits	x	x
16 bits	x	0
32 bits	0	0

5.10.11 DTS0TCEA: DTS Transfer Counter or Else Address Register [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU.

Access —

Address —

Initial value 0000_H

This register is initialized by a reset from any source.

(1) Transfer Count 255 Mode

15	14	13	12	11	10	9	8
DTS0STR[7:0]							

7	6	5	4	3	2	1	0
DTS0STC[7:0]							

Bit Position	Bit Name	Function
15 to 8	DTS0STR7 to DTS0STR0	Short transfer number register (transfer count hold register) During operation in transfer-count 255 mode, this register holds the transfer count. If a DTS request is generated while the DTS0STC bits are 0, the contents of this register are copied to the DTS0STC bits.
7 to 0	DTS0STC7 to DTS0STC0	Short transfer counter (short transfer count register) During operations in transfer-count 255 mode, this bit field counts the number of transfers. The count is decremented by one per DTS cycle. Setting the value to 1 leads to a single DTS transfer, and setting the value to FF_H leads to 255 cycles of DTS transfer. An interrupt is generated when the value of this register reaches 0 (or the chain function can be selected). Continued execution of transfer can be selected. If a DTS request is generated while the value of this bit field is 0, the contents of the DTS0STR field are copied to this field, and decrementation starts again.

(2) Transfer Count 65535 Mode

15	14	13	12	11	10	9	8
DTS0LTC[15:8]							

7	6	5	4	3	2	1	0
DTS0LTC[7:0]							

Bit Position	Bit Name	Function
15 to 0	DTS0LTC15 to DTS0LTC0	Long transfer counter (long transfer count register) During operations in transfer-count 65,535 mode, this bit field counts the number of transfers. The count is decremented by one per DTS cycle. Setting the value to 1 leads to a single DTS transfer, and setting the value to $FFFF_H$ leads to 65,535 cycles of DTS transfer. An interrupt is generated when the value of this register reaches 0 (or the chain function can be selected).

(3) Extended Chain Mode

15	14	13	12	11	10	9	8
DTS0EA[15:8]							

7	6	5	4	3	2	1	0
DTS0EA[7:0]							

Bit Position	Bit Name	Function
15 to 0	DTS0EA15 to DTS0EA0	Else address (ELSE address) During operation in extended chain mode, this register specifies the position of the TI to be read if the result of the condition test is false (the ELSE case). To specify the TI position, use this register to set an offset from the current position (the TI-A address where extended chain operation is specified). The value is specified as a signed 16-bit value from -32,768 to 32,760 (the three lower-order bits are 3'b000).

5.10.12 DTS0SCS: DTS Source Address Count Size Register [Register Group B]

This is a DTS-internal register and is neither readable nor writable from the CPU.

When the value of the DTS0IMM bits in DTS0CIR is 1, the following operations are performed.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 32 bits

  DTS0SCS becomes 0.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 16 bits

  DTS0SAR (IMM15 to IMM0) is incremented or decremented by DTS0SCS.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 8 bits

  DTS0SAR (IMM7 to IMM0) is incremented or decremented by DTS0SCS.

However, the settings in this register are ignored in SG-mode operation.

Access —

Address —

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0SCS2	DTS0SCS1	DTS0SCS0

Bit Position	Bit Name	Function
2 1 0	DTS0SCS2 DTS0SCS1 DTS0SCS0	Specifies the increment for counting of source addresses. 000: 0 (FIX) 001: +1 010: +2 011: +4 101: –1 110: –2 111: –4

5.10.13 DTS0DCS: DTS Destination Address Count Size Register [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU.

The setting of this register is ignored during operation in SG mode.

Access —

Address —

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0DCS2	DTS0DCS1	DTS0DCS0

Bit Position	Bit Name	Function
2 1 0	DTS0DCS2 DTS0DCS1 DTS0DCS0	Specifies the increment for counting of destination addresses. 000: 0 (FIX) 001: +1 010: +2 011: +4 101: –1 110: –2 111: –4

5.10.14 DTS0CIR: DTS Control Information Register [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU

Access —

Address —

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
DTS0CAF	DTS0DPE1	DTS0SPE1	DTS0DISI	DTS0CM1	DTS0CM0	DTS0DS1	DTS0DS0

7	6	5	4	3	2	1	0
DTS0BEN	DTS0IMM	DTS0TTYP5	DTS0TTYP4	DTS0TTYP3	DTS0TTYP2	DTS0TTYP1	DTS0TTYP0

Bit Position	Bit Name	Function
15	DTS0CAF	Condition accumulative flag This is an accumulative flag indicating the results of flag checking and comparison. When the result of flag checking or comparison is false, this bit is set to 1 and the value is written back to TI-A. If this bit is 1 when TI-A is read, then 1 must be written back here. Condition for setting (to 1): • Result of comparison is false. • Result of flag checking is false. • Reading of 1 from this bit (bit 15 was 1 in TI-A fetching). In other words, when bit 15 of TI-A is set to 1, the value of this bit remains 1. Condition for clearing (to 0): • Reading of this bit as 0 (bit 15 was 0 in TI-A fetching).
14	DTS0DPE1	Destination select PE1 (destination PE1 select) This bit specifies the address range on the destination side. 0: Selects data-flash and the peripheral I/O area 1: Selects code-flash and on-chip RAM
13	DTS0SPE1	Source select PE1 (source PE1 select) This bit specifies the address range on the source side. 0: Selects data-flash and peripheral I/O area 1: Selects code-flash and on-chip RAM
12	DTS0DISI	Disable interrupts (interrupt output disabled) This bit prohibits the generation of interrupt requests for the DTSFSL block. 0: Interrupts enabled 1: Interrupts disabled
11 10	DTS0CM1 DTS0CM0	Chain mode 00: Transfers are not chained. [if (counter==0) then interrupt else no operation] 01: Transfers are chained when counter = 0. [if (counter==0) then chain & interrupt else no operation] 10: Transfers are chained if condition is true. [if (true) then chain else interrupt] 11: Transfers are always chained.

Bit Position	Bit Name	Function
9 8	DTS0DS1 DTS0DS0	Data size These bits set the data size for transfer, comparison, and flag checking. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
7	DTS0BEN	Block enable This is the block enable signal. 0: Single mode 1: Block mode
6	DTS0IMM	Immediate enable (immediate function) This bit enables/disables the immediate function. When the immediate function is enabled, the DTS0SAR register functions as the immediate register. 0: Immediate function is disabled. 1: Immediate function is enabled.
5 4 3 2 1 0	DTS0TTYP5 DTS0TTYP4 DTS0TTYP3 DTS0TTYP2 DTS0TTYP1 DTS0TTYP0	Transfer type (transfer type select) These bits select the transfer type. When a transfer type is selected, a special function and transfer count are selected at the same time. See section 5.11.5, Transfer Types, for the list of transfer types.

5.10.15 DTS0ECSRA: DTS Extension Address Count Size/Repeat Address Registers [Register Group B]

This is a DTS internal register and is neither readable nor writable from the CPU.

Access —

Address —

Initial value 00000000_H

This register is initialized by a reset from any source.

(1) SG (Scatter and Gather) Mode

31	30	29	28	27	26	25	24
DTS0EDCS[15:8]							

23	22	21	20	19	18	17	16
DTS0EDCS[7:0]							

15	14	13	12	11	10	9	8
DTS0ESCS[15:8]							

7	6	5	4	3	2	1	0
DTS0ESCS[7:0]							

Bit Position	Bit Name	Function
31 to 16	DTS0EDCS15 to DTS0EDCS0	These bits specify the number to count for destination addresses. Any signed 16-bit value from -32,768 to 32,767 can be specified. This register is only valid when SG mode is selected as a special function of the DTS. When operation is not in SG mode, DTS0DCS determines the number for counting.
15 to 0	DTS0ESCS15 to DTS0ESCS0	These bits specify the number to count for source addresses. Any signed 16-bit value from -32,768 to 32,767 can be specified. This register is only valid when SG mode is selected as a special function of the DTS. When operation is not in SG mode, DTS0SCS determines the number for counting.

(2) SREP (Source Repeat) Mode

31	30	29	28	27	26	25	24
DTS0SRA[31:24]							

23	22	21	20	19	18	17	16
DTS0SRA[23:16]							

15	14	13	12	11	10	9	8
DTS0SRA[15:8]							

7	6	5	4	3	2	1	0
DTS0SRA[7:0]							

Bit Position	Bit Name	Function
31 to 0	DTS0SRA31 to DTS0SRA0	These bits indicate the location where the repeat address is held. Source repetition only becomes effective when SREP is selected as a special function of the DTS.

(3) DREP (Destination Repeat) Mode

31	30	29	28	27	26	25	24
DTS0DRA[31:24]							

23	22	21	20	19	18	17	16
DTS0DRA[23:16]							

15	14	13	12	11	10	9	8
DTS0DRA[15:8]							

7	6	5	4	3	2	1	0
DTS0DRA[7:0]							

Bit Position	Bit Name	Function
31 to 0	DTS0DRA31 to DTS0DRA0	These bits indicates the location where the repeat address is held. Destination repetition only becomes effective when DREP is selected as a special function of the DTS.

5.11 DTS Function Details

5.11.1 Transfer information (TI)

DTS internal registers (register group B) cannot be operated directly from an external source. The DTS reads transfer information from on-chip RAM, and then sets values to the internal registers. This transfer information is called TI.

TI is configured of either 96 bits (TI-A to TI-C) or 128 bits (TI-A to TI-D), and the setting in DTS0TTYP5 to DTS0TTYP0 determines which type is used.

(1) Configuration of TI

Each TI should be 32-bit aligned (the lower 2 bits of the address are 00).

The 128 bits from TI-A to TI-D should be allocated to on-chip RAM as one set.

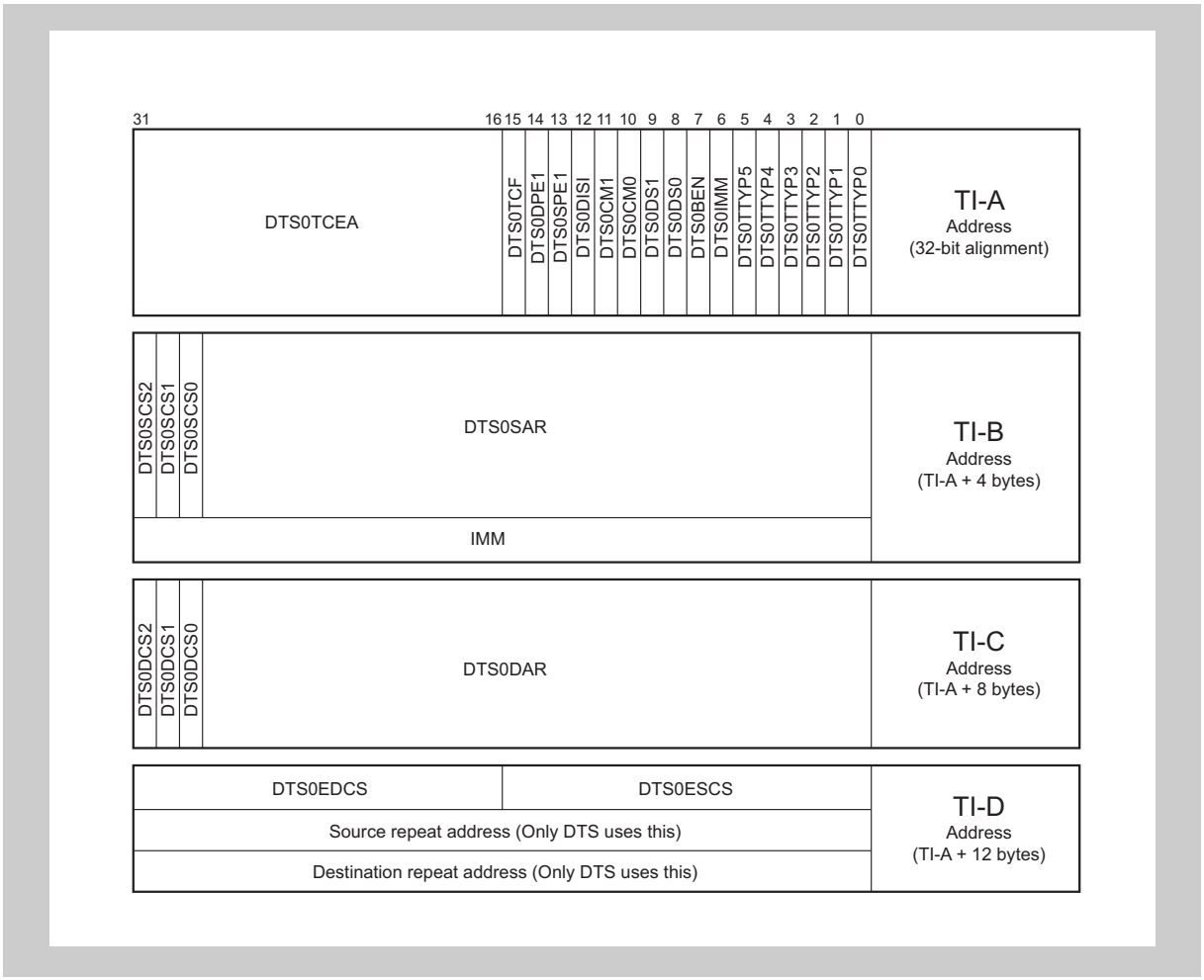


Figure 5-10 Configuration of TI

(2) Address where TI Is Allocated

The DTS uses DTS0BTR and DTS0BVR to determine the address of the TI for a channel.

A TI can be freely placed anywhere in the range from the address specified by DTS0BVR to the address specified by DTS0BVR plus $FFFC_H$.

The method for determining the TI addresses for the DTS is as follows.

1. A 16-bit offset address is read from the table address [table address = DTS0BTR + channel number × 2] (the 2 lower-order bits of the offset address are ignored).
2. The next TI in sequence is read from the vector address [TI address = DTS0BVR + offset address]. In this case as well, the 2 lower-order bits are fixed to 00.

As for the request for transfer on channel 3 in the following figure, for example, the offset to the address within the table ($1EC0_1000_H$ + 4 × 2) is 0320_H and the vector address is DTS0BVR + 0320_H.

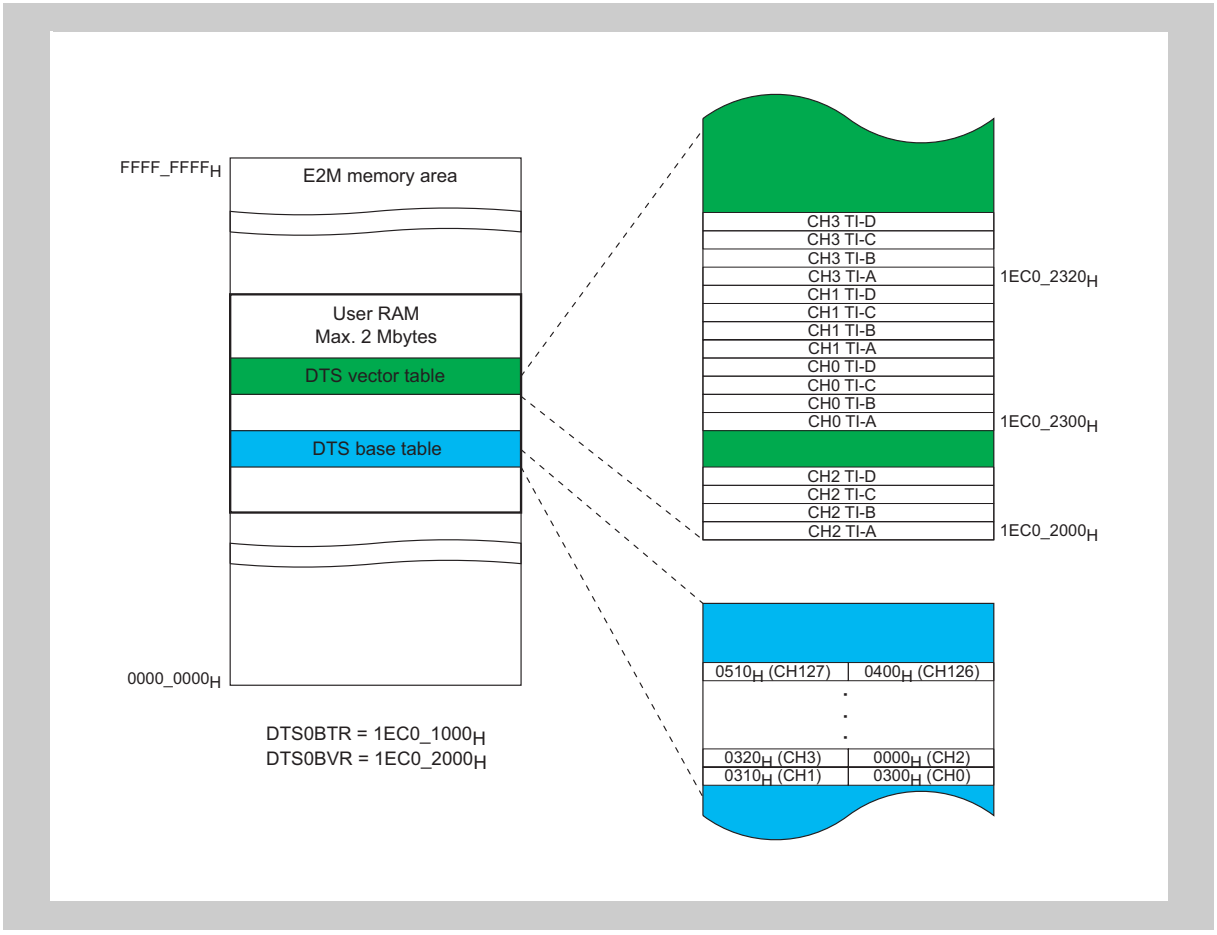


Figure 5-11 Example of Address where TI Is Allocated

(3) Base Table

The DTS refers to a base table to find the TI's position. This base table must be created by the user. The address set by DTS0BTR is the base table's start address, and 16 bits are allocated for each address, starting from channel 0.

Data set to the table is offset from the DTS0BVR value (the offset range is 0 to 65,524 when using TI-A, TI-B, and TI-C, or 0 to 65,520 when using TI-A, TI-B, TI-C, and TI-D).

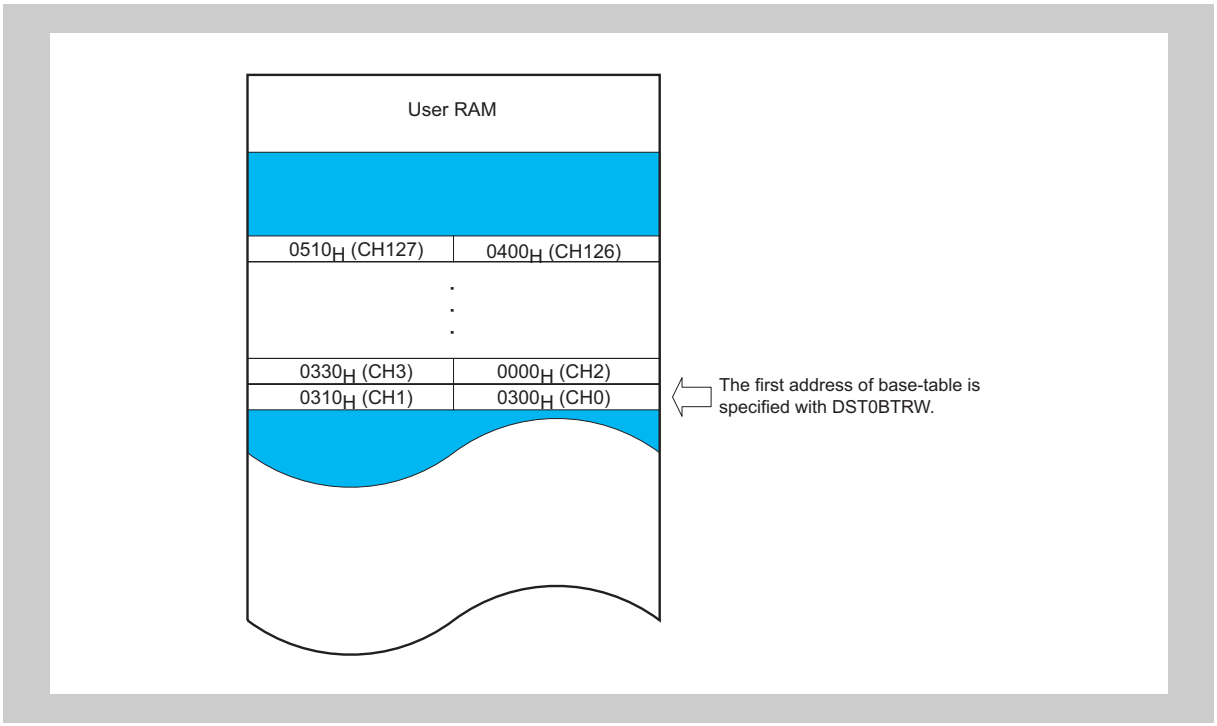


Figure 5-12 Description of Base Table

5.11.2 DTS Transfer Setting Flow

The following figure shows the flow for setting up DTS transfer by hardware when DTSEN is set to “enabled” after the TI is set.

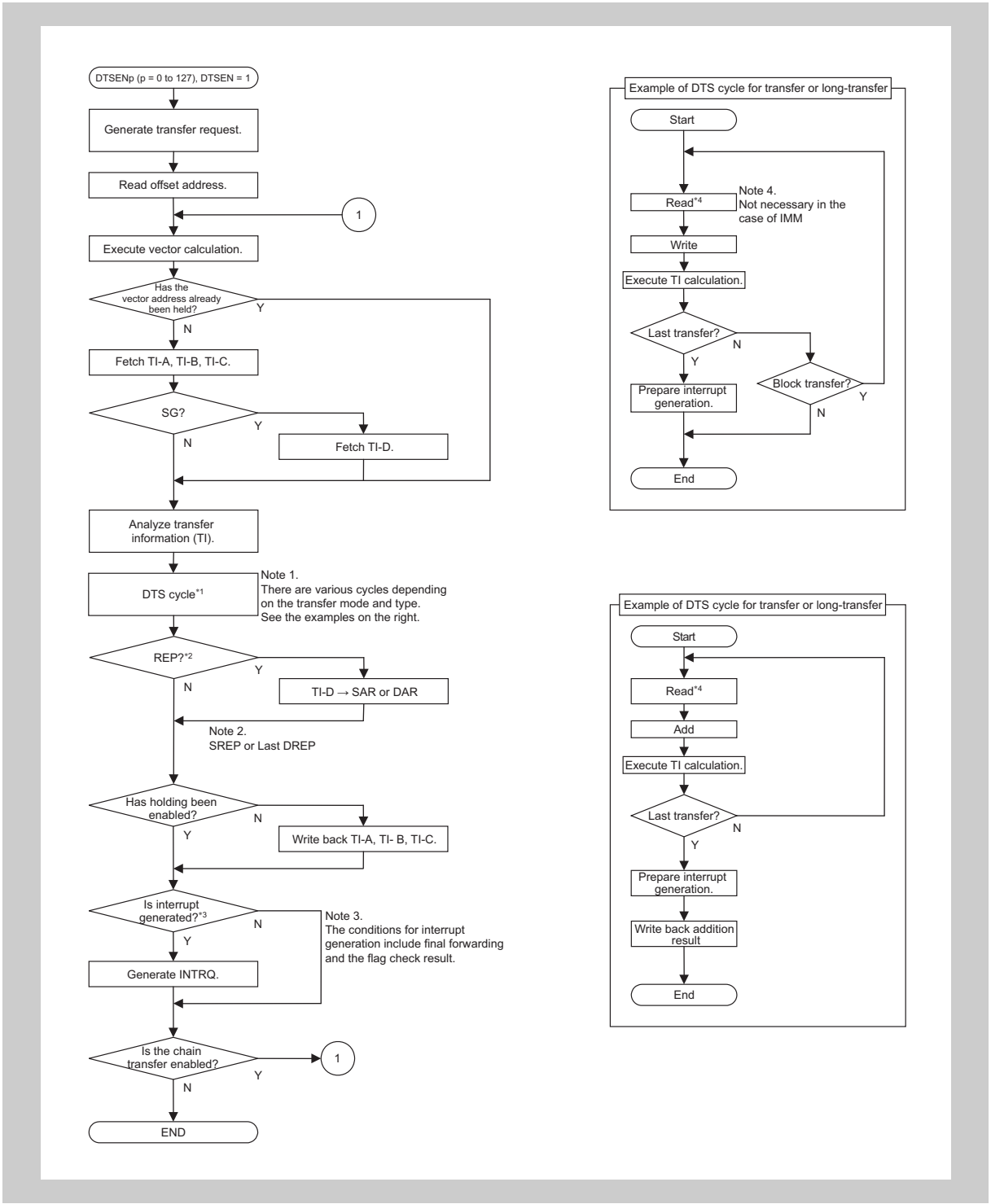


Figure 5-13 DTS Transfer Setting Flow

5.11.3 Basic Operations of DTS

The DTS fetches the information required for transfer (TI) from on-chip RAM, then executes transfer in accord with the TI, after which it updates the TI and writes the updated TI back to on-chip RAM.

(1) TI Fetching and Write Back

For each DTS request, one round of "TI fetch cycle → DTS cycle → TI write back cycle" proceeds. Since the bus is not locked, a request for a CPU-access cycle may interrupt transfer.

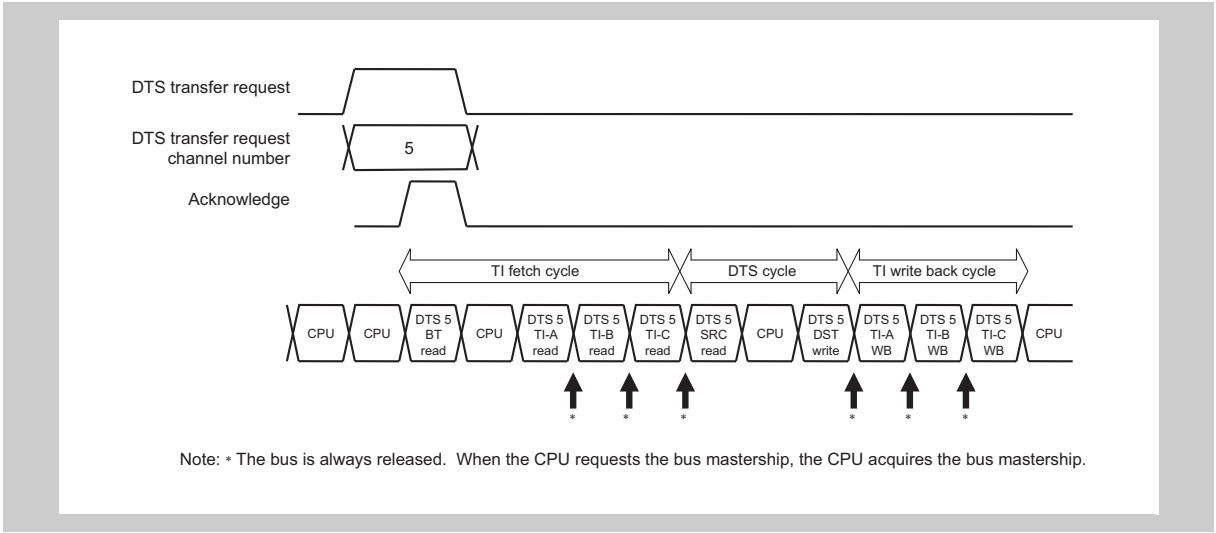


Figure 5-14 Example of TI Fetch and Write Back

5.11.4 Transfer Modes

As transfer modes, the DTS supports a single-transfer mode and block-transfer mode.

(1) Single Transfer Mode

When a DTS request is generated in single-transfer mode, one round of “TI fetch → DTS cycle → TI write back” proceeds, after which the transfer-request waiting status flag is set. When the next DTS request is generated, the same sequence is executed again from fetching of the TI. The sequence is repeated until the value of DTS0STC (or DTS0LTC if operation is in 65535 mode) becomes 0. Since the bus is not locked, a request for a CPU-access cycle may interrupt transfer.

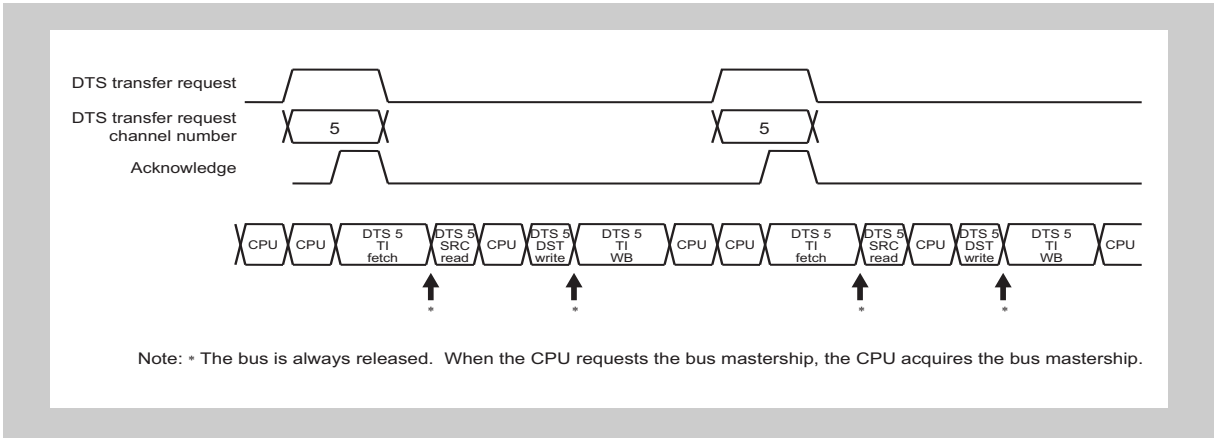


Figure 5-15 Example of Single Transfer

(2) Block Transfer Mode

When a DTS request is generated and operation is in block-transfer mode, the TI is fetched and DTS cycles are repeated until the value of DTS0STC (or DTS0LTC if operation is in 65535 mode) becomes 0, after which the TI is written back. Since the bus is not locked, a request for a CPU-access cycle may interrupt transfer.

Other DTS requests cannot be received during block transfer (they are put on hold in the DTSFSL).

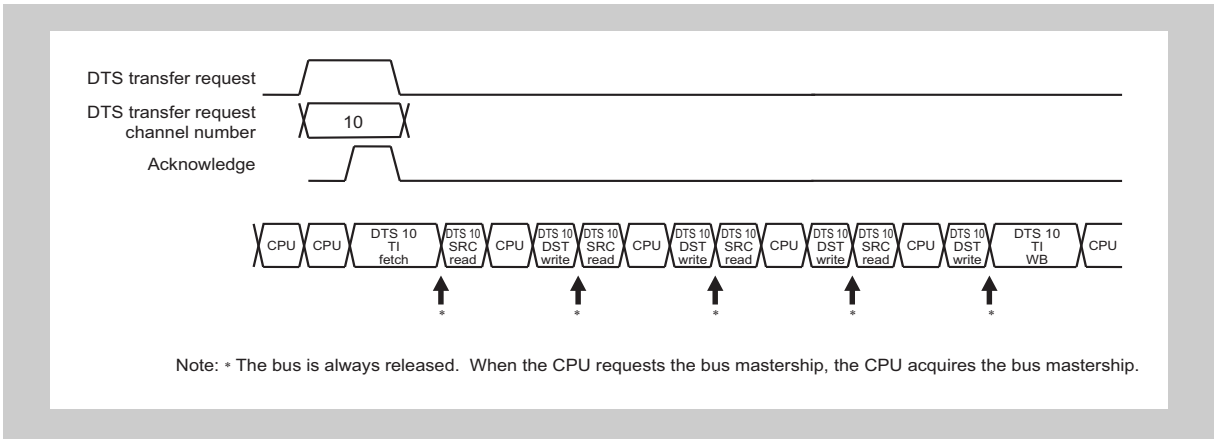


Figure 5-16 Example of Block Transfer

5.11.5 Transfer Types

Transfer (two-cycle transfer), flag checking, comparison, and addition are supported as transfer types. Although the transfer type is specified by the DTS0TTYP5 to DTS0TTYP0 bits in DTS0CIR, the transfer count and special function* must already have been set at the time.

The DTS transfer types are listed below. Operation for each transfer type varies with the setting of the DTS0IMM bits in DTS0CIR.

Note * For a description of the special function, see section 5.11.6, Special Functions.

Table 5-9 List of DTS Transfer Types (1/2)

Settings of Bits TTYPE5 to TTYPE0			Maximum Transfer Count	Transfer Type	Special Function	TI-D
0	000	00	255	Transfer	None	Not Used
		01			SREP	Used
		10			DREP	Used
		11			SG	Used
0	001	00	255	Addition	None	Not Used
		01			SREP	Used
		10			DREP	Used
		11			SG	Used
0	010	00	255	flag[0]	None	Not Used
		01		flag[0]	SREP	Used
		10		flag[0]	DREP	Used
		11		flag[0]	(SG)	Used
0	011	00	255	flag[1]	None	Not Used
		01		flag[1]	SREP	Used
		10		flag[1]	DREP	Used
		11		flag[1]	(SG)	Used
0	100	00	255	comp[=]	None	Not Used
		01		comp[=]	SREP	Used
		10		comp[=]	DREP	Used
		11		comp[=]	(SG)	Used
0	101	00	255	comp[!=]	None	Not Used
		01		comp[!=]	SREP	Used
		10		comp[!=]	DREP	Used
		11		comp[!=]	(SG)	Used
0	110	00	255	comp[<]	None	Not Used
		01		comp[<]	SREP	Used
		10		comp[<]	DREP	Used
		11		comp[<]	(SG)	Used

Table 5-9 List of DTS Transfer Types (2/2)

Settings of Bits TTYPE5 to TTYPE0			Maximum Transfer Count	Transfer Type	Special Function	TI-D
0	111	00	255	comp[>]	None	Not Used
		01		comp[>]	SREP	Used
		10		comp[>]	DREP	Used
		11		comp[>]	(SG)	Used
1	000	00	65536	Long transfer	None	Not Used
		01			(SREP)	Used
		10			(DREP)	Used
		11			SG	Used
1	001	00	65536	Long addition	None	Not Used
		01			(SREP)	Used
		10			(DREP)	Used
		11			SG	Used
1	010		—	flag[0]	Extended chain	Not Used
	011			flag[1]		
	100			comp[=]		
	101			comp[!=]		
	110			comp[<]		
	111			comp[>]		

(1) Transfer: TTYP5 to TTYP2 Are 0000 or 1000

- When TTYP5 to TTYP2 are 0000 and DTS0IMM in DTS0CIR is 0.

  When the transfer type is set to “transfer”, the DTS cycles are only used for data transfer. DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC. When the value of DTS0STC becomes 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  The available chain-mode specifications are 00 (no chaining), 01 (chain with transfer count 0 chain), or 11 (transfer is always chained).

  SREP (source repeat), DREP (destination repeat), or SG (scatter and gather) can be selected as the special function.

Operation Data read from address SAR are written to address DAR.

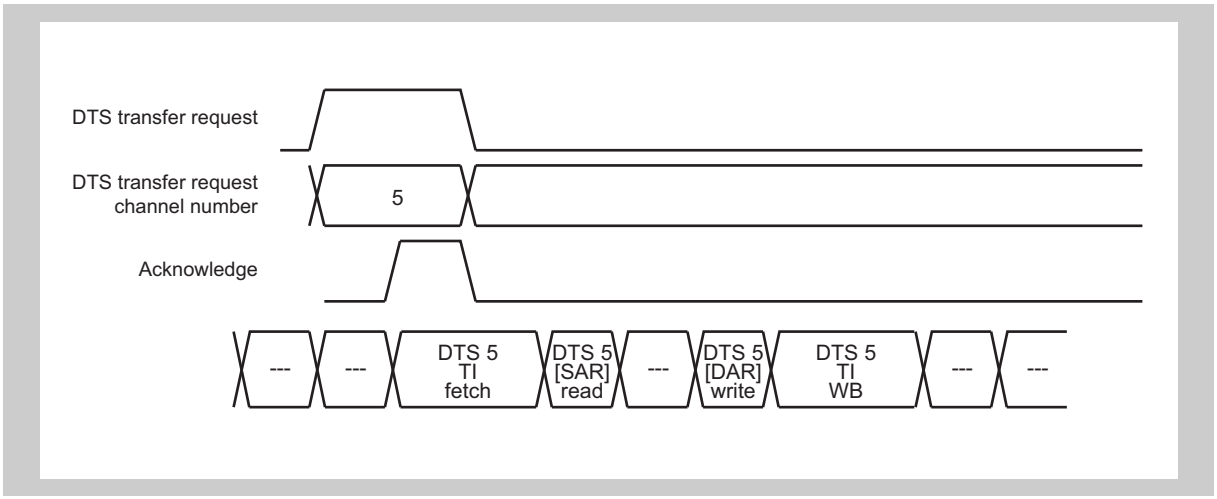


Figure 5-17 When TTYP5 to TTYP2 Are 0000 and DTS0IMM in DTS0CIR Is 0

- When TTYP5 to TTYP2 are 0000 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM and the DTS0SAR value is written to the address indicated by DTS0DAR.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC. When the value of DTS0STC becomes 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  The chain-mode specification can be 00, 01, or 11, and DREP or SG can be selected as the special function.

Operation The SAR data are written to address DAR without modification.

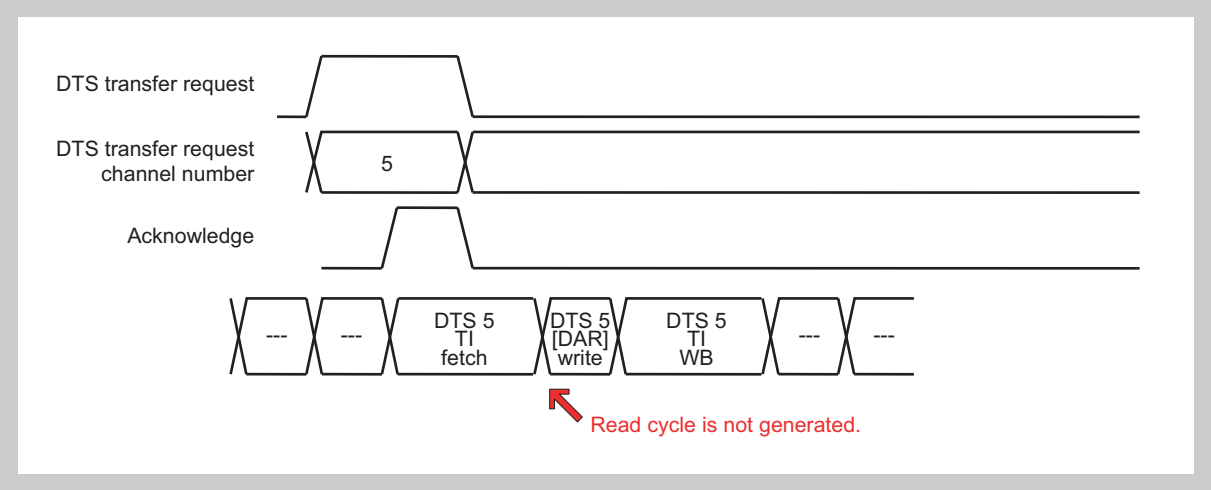


Figure 5-18 When TTYP5 to TTYP2 Are 0000 and Immediate Function Is Enabled (DTS0IMM in DTS0CIR Is 1)

- When TTYP5 to TTYP2 are 1000 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “long transfer”, the DTS cycle is only used for data transfer. DTS0TCEA becomes DTS0LTC and a value up to 65,535 is specified as the transfer count. Once the value of DTS0LTC reaches 0, no more requests for transfer can be accepted.

  Before executing further transfer, set DTS0LTC (in the TI-A's DTS0TCEA) again.

  The chain mode specification can be 00, 01, or 11, and SG can be selected as the special function.

Operation Data read from address SAR are written to address DAR.

The figure is the same as *figure5-17 When TTYP5 to TTYP2 Are 0000 and DTS0IMM in DTS0CIR Is 0*.

- When TTYP5 to TTYP2 are 1000 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM and the DTS0SAR value is written to the address indicated by DTS0DAR. DTS0TCEA becomes DTS0LTC and a value up to 65,535 is specified as the transfer count. When the value of DTS0LTC reaches 0, no more requests for transfer can be accepted. Before executing further transfer, set DTS0LTC (in the TI-A's DTS0TCEA) again.

  The chain mode specification can be 00, 01, or 11, and SG can be selected as the special function.

Operation The SAR data are written to address DAR without modification.

The figure is the same as *figure5-18 When TTYP5 to TTYP2 Are 0000 and Immediate Function Is Enabled (DTS0IMM in DTS0CIR Is 1)*.

(2) Addition: When TTYP5 to TTYP2 Are 0001 or 1001

- When TTYP5 to TTYP2 are 0001 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “addition”, each DTS cycle becomes an accumulation cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  The chain mode specification can be 00 or 01, and SREP, DREP, or SG can be selected as the special function.

  DTS0BEN in DTS0CIR is ignored and the mode is always block transfer.

Operation Data are read from address SAR until DTS0STC reaches 0.
The read data are added each time as unsigned data.
When DTS0STC reaches 0, the addition result is written to address DAR.

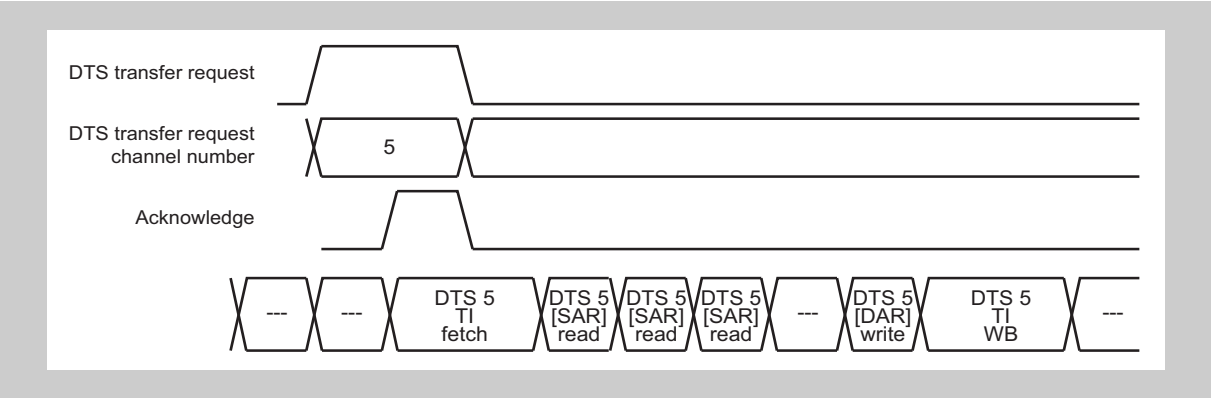


Figure 5-19 When TTYP5 to TTYP2 Are 0001 and DTS0IMM in DTS0CIR Is 0

A cumulative value is stored in a 32-bit register regardless of the data size, and when the transfer count reaches 0, the data are written to the DAR as a 32-bit unit. Any data exceeding 32 bits are discarded.

Accordingly, when the transfer size is 32 bits, the result of calculation will not be correct if an overflow or borrowing occurs. When the transfer size is 8 or 16 bits, an overflow or borrowing will not occur even if the maximum transfer count of 65,535 times is specified.

- When TTYP5 to TTYP2 are 0001 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  DTS0BEN in DTS0CIR is ignored and the mode always becomes block transfer. In other words, the DTS0SAR value is added the number of times specified by DTS0STC, and when the value of DTS0STC reaches 0, the result is written to the address indicated by the DAR.

  The chain-mode specification can be 00 or 01, and DREP or SG can be selected as the special function.

- When TTYP5 to TTYP2 are 1001 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “long addition”, the DTS cycle becomes a cycle of accumulation. DTS0TCEA becomes DTS0LTC and a value up to 65,535 is specified as the transfer count. When the value of DTS0STC reaches 0, further transfer requests cannot be accepted.

  Before executing further transfer, set DTS0LTC (in the TI-A's DTS0TCEA) again.

  The chain mode specification can be 00 or 01, and SG can be selected as the special function.

  DTS0BEN in DTS0CIR is ignored and the mode always becomes block transfer.

- When TTYP5 to TTYP2 are 1001 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM.

  DTS0TCEA becomes DTS0LTC and a value up to 65,535 is specified as the transfer count.

  DTS0BEN in DTS0CIR is ignored and the mode always becomes block transfer. In other words, the DTS0SAR value is added the number of times specified by DTS0LTC, and when the value of DTS0STC reaches 0, the result of addition is written to the address indicated by the DAR.

  The chain mode specification can be 00, or 01, and SG can be selected as the special function.

(3) flag[0]: When TTYP5 to TTYP2 Are 0010 or 1010

- When TTYP5 to TTYP2 are 0010 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “flag [0]”, the DTS cycle becomes a flag-checking [0] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (SREP, DREP, or SG) and any chain mode can be selected.

  The function of flag checking [0] is to check whether a desired bit in the data read from the address indicated by the DAR is 0.

  If at least one bit is 0, the result of checking is “true”. The result can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, the value specifying the bits for checking is read from the address indicated by the SAR, after which the value to be checked is read from the address indicated by the DAR and flag checking is executed.

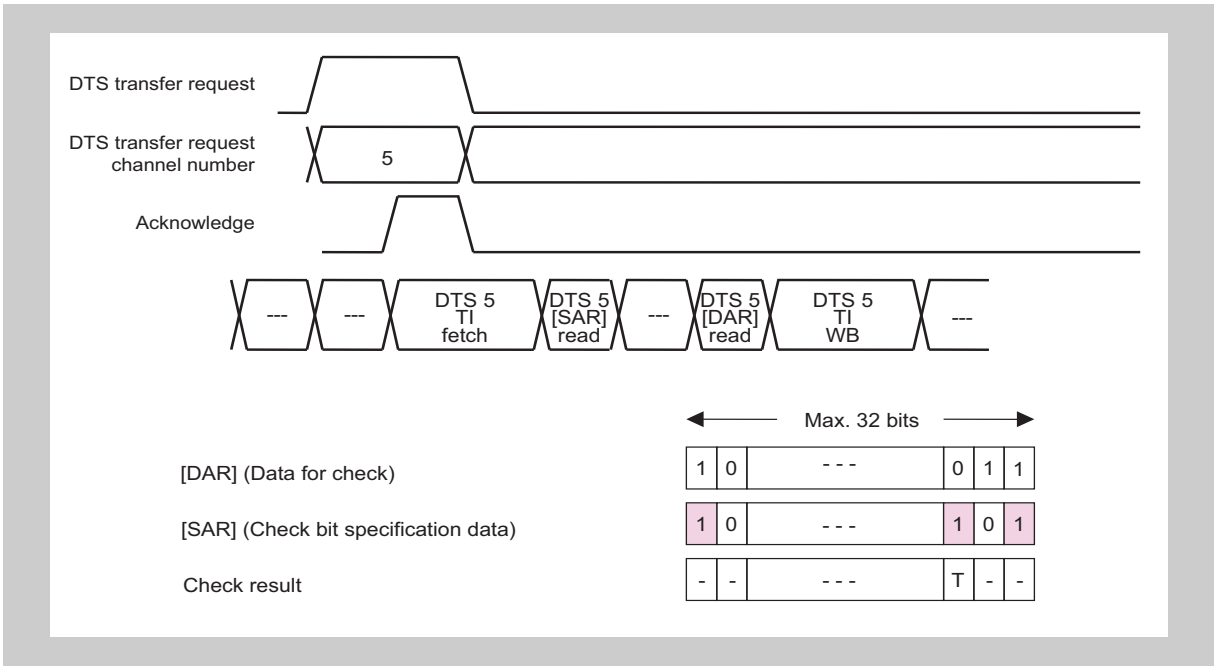


Figure 5-20 When TTYP5 to TTYP2 Are 0010 and DTS0IMM in DTS0CIR Is 0

- When TTYP5 to TTYP2 are 0010 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the bits to be checked.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (DREP or SG) and any chain mode can be selected.

  The function of flag checking [0] is to check whether a desired bit in the data read from the address indicated by the DAR is 0.

  If at least one bit is 0, the result of checking is “true”. The result can be used as a condition for chaining or to produce interrupt output.

Operation The data to be checked are read from the address indicated by the DAR, and the value in the SAR is used to specify the flags to be checked.

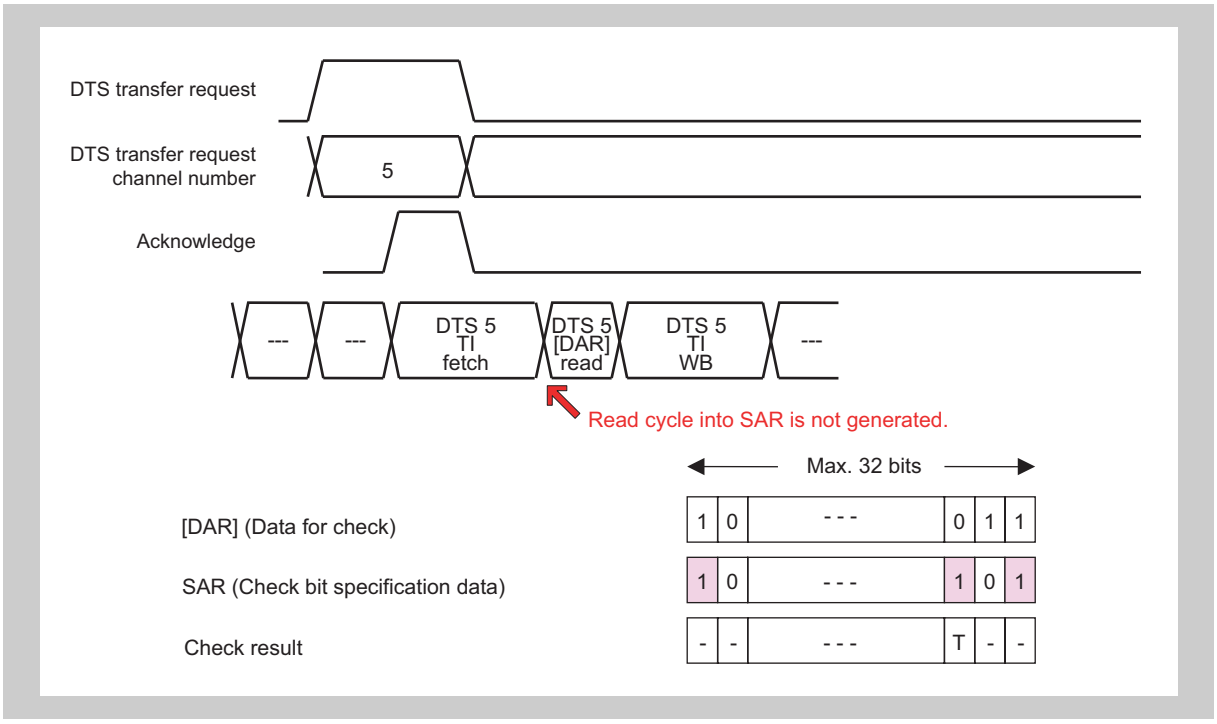


Figure 5-21 When TTYP5 to TTYP2 are 0010 and Immediate Function Is Enabled (DTS0IMM in DTS0CIR Is 1)

- When TTYP5 to TTYP2 are 1010 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended flag[0]”, the DTS cycle becomes a flag-checking [0] cycle.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation Firstly, the value specifying the bits to be checked is read from the address indicated by the SAR, after which the value to be checked is read from the address indicated by DAR, and flag checking proceeds.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1010 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the bits to be checked.

  DTS0TCEA becomes DTS0EA and has the function of specifying an offset address for use when the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfers. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfers.

  Neither a chain mode nor a special function can be selected.

Operation The value to be checked is read from the address indicated by the DAR, and flag checking is executed with the SAR value specifying the bits to be checked.

Transfers are always chained.

(4) flag[1]: When TTYP5 to TTYP2 Are 0011 or 1011

- When TTYP5 to TTYP2 are 0011 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “flag [1]”, the DTS cycle becomes a flag-checking [1] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (SREP, DREP, or SG) and any chain mode can be selected.

  The function of flag checking [1] is to check whether a desired bit in the data read from the address indicated by the DAR is 1.

  If at least one bit is 1, the result of checking is “true”. The result can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, the value specifying the bits for checking is read from the address indicated by the SAR, after which the value to be checked is read from the address indicated by the DAR and flag checking is executed.

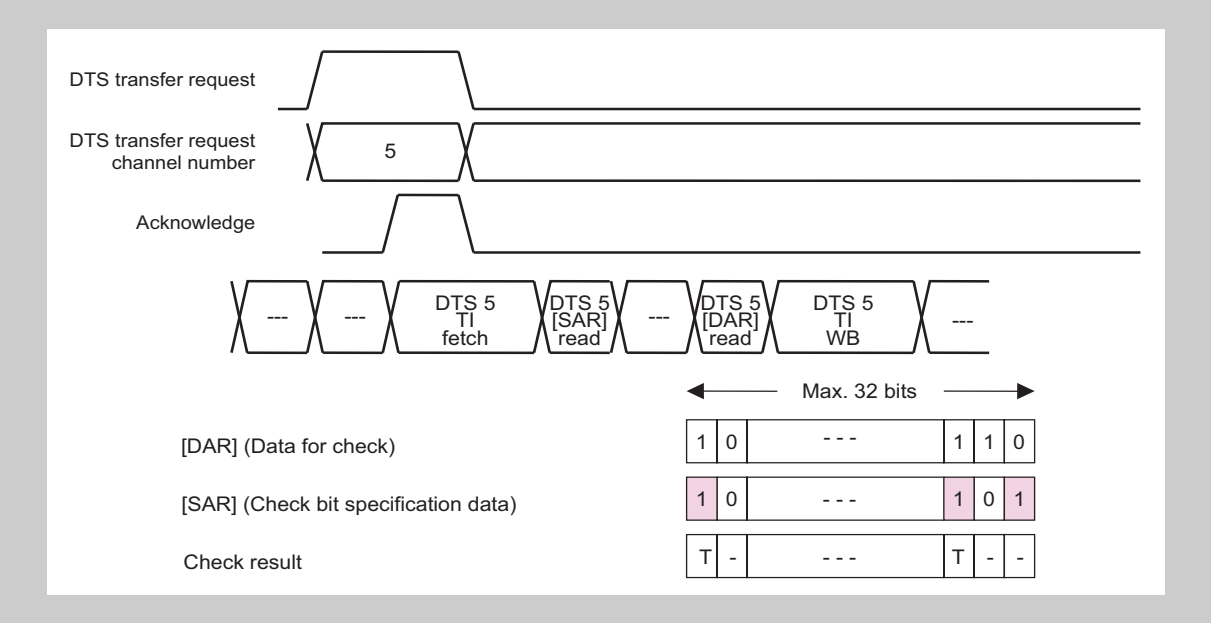


Figure 5-22 When TTYP5 to TTYP2 Are 0011 and DTS0IMM in DTS0CIR Is 0

- When TTYP5 to TTYP2 are 0011 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the bits to be checked.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (DREP or SG) and any chain mode can be selected.

  The function of flag checking [1] is to check whether a desired bit in the data read from the address indicated by the DAR is 1.

  If at least one bit is 1, the result of checking is “true”. The result can be used as a condition for chaining or to produce interrupt output.

Operation The data to be checked are read from the address indicated by the DAR, and the value in the SAR is used to specify the flags to be checked.

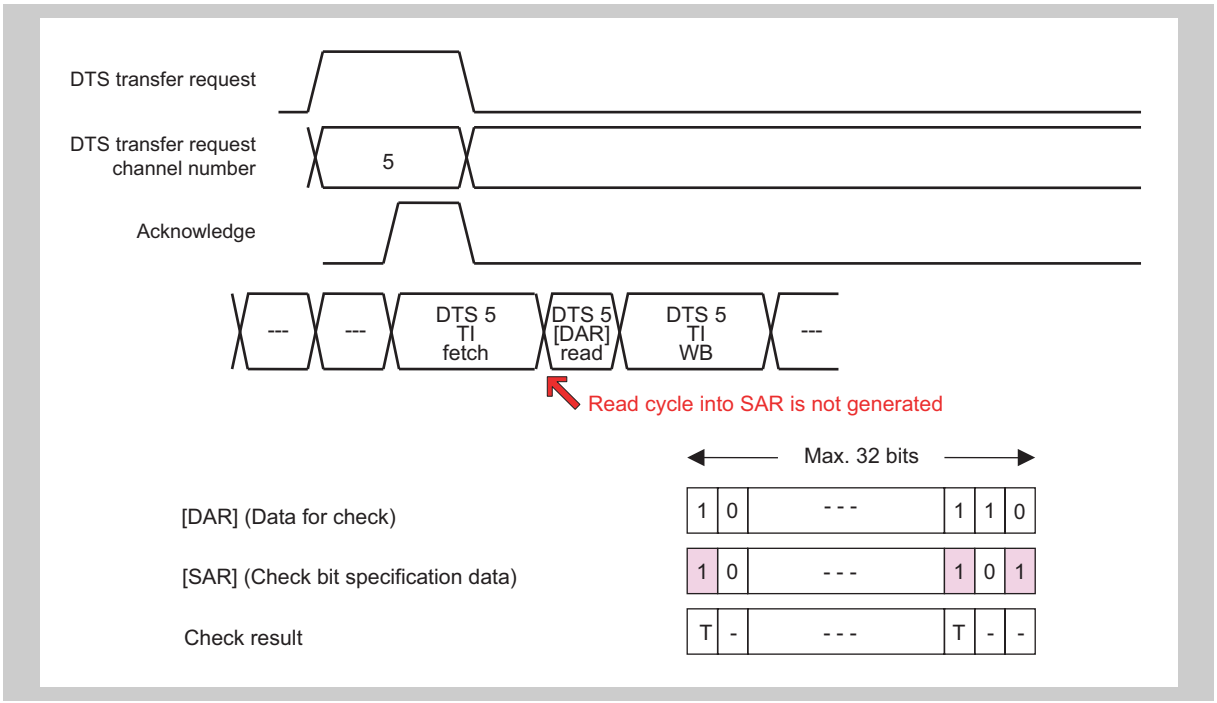


Figure 5-23 When TTYP5 to TTYP2 Are 0011 and Immediate Function Is Enabled (DTS0IMM in DTS0CIR Is 1)

- When TTYP5 to TTYP2 are 1011 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended flag[1]”, the DTS cycle becomes a flag-checking [1] cycle.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, the transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation Firstly, the value specifying the bits for checking is read from the address indicated by the SAR, after which the value to be checked is read from the address indicated by the DAR and flag checking is executed.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1011 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the bits to be checked.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, the transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation The data to be checked are read from the address indicated by the DAR, and the value in the SAR is used to specify the flags to be checked. Transfers are always chained.

(5) comp[=]: When TTYP5 to TTYP2 Are 0100 or 1100

- When TTYP5 to TTYP2 are 0100 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “comp[=]”, the DTS cycle becomes a comparison [=] cycle. DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (SREP, DREP, or SG) and any chain mode can be selected.

  With comp[=], values read from the addresses indicated by the SAR are compared with values read from the addresses indicated by the DAR, and if the data match, the result of comparison is “true”.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, a value is read from the address indicated by the SAR, then a value is read from the address indicated by the DAR.

Next, whether [SAR] == [DAR] is checked.

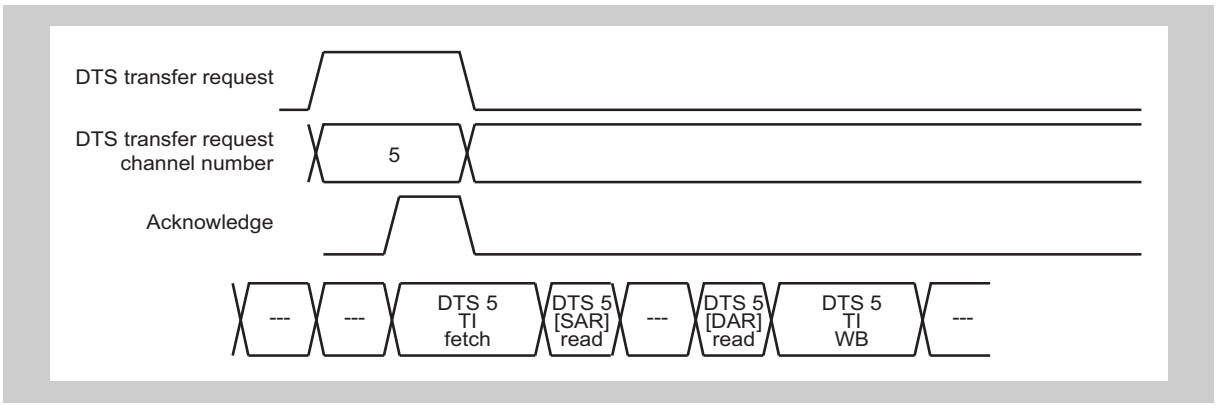


Figure 5-24 When TTYP5 to TTYP2 Are 0100 and DTS0IMM in DTS0CIR Is 0

- When TTYP5 to TTYP2 are 0100 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (DREP or SG) and any chain mode can be selected.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation A value is read from the address indicated by the DAR, and whether SAR == [DAR] is checked.

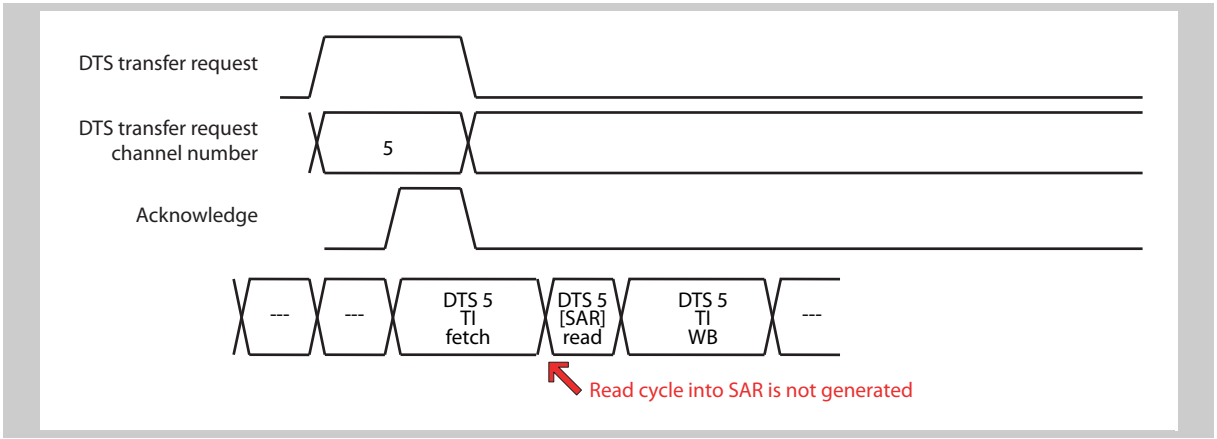


Figure 5-25 When TTYP5 to TTYP2 Are 0100 and Immediate Function Is Enabled (DTS0IMM in DTS0CIR Is 1)

- When TTYP5 to TTYP2 are 1100 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended comp[=]”, the DTS cycle becomes a comparison [=] cycle.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation Firstly, a value is read from the address indicated by the SAR, then a value is read from the address indicated by the DAR.

Next, whether [SAR] == [DAR] is checked.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1100 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation The value for comparison is read from the address indicated by the DAR, and is then compared with the value in the SAR.

Transfers are always chained.

(6) comp[!=]: When TTYP5 to TTYP2 Are 0101 or 1101

- When TTYP5 to TTYP2 are 0101 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “comp[!=]”, the DTS cycle becomes a comparison [!=] cycle. DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (SREP, DREP, or SG) and any chain mode can be selected.

  With comp[!=], values read from the addresses indicated by the SAR are compared with values read from the addresses indicated by the DAR, and if the data do not match, the result of comparison is “true”.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, a value is read from the address indicated by the SAR, then a value is read from the address indicated by the DAR.

Next, whether [SAR] != [DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and DTS0IMM in DTS0CIR is 0.

- When TTYP5 to TTYP2 are 0101 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (DREP or SG) and any chain mode can be selected.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation Values are read from the addresses indicated by the DAR, and whether SAR != [DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and immediate function is enabled (DTS0IMM in DTS0CIR is 1).

- When TTYP5 to TTYP2 are 1101 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended comp[!=]”, the DTS cycle becomes a comparison [!=] cycle.
  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.
  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.
  Neither a chain mode nor a special function can be selected.

Operation Firstly, a value is read from the address indicated by the SAR, then a value is read from the address indicated by the DAR.

Next, whether [SAR] != [DAR] is checked.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1101 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.
  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.
  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.
  Neither a chain mode nor a special function can be selected.

Operation The value for comparison is read from the address indicated by the DAR, and is then compared with the value in the SAR.

Transfers are always chained.

(7) comp[<]: When TTYP5 to TTYP2 Are 0110 or 1110

- When TTYP5 to TTYP2 are 0110 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “comp[<]”, the DTS cycle becomes a comparison [<] cycle.
  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.
  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

A special function (SREP, DREP, or SG) and any chain mode can be selected.

With comp[<], values read from the address indicated by the SAR are compared with values read from the addresses indicated by the DAR, and the result of comparison is “true” if the data read from the address indicated by the SAR is smaller.

The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, a value is read from the address indicated by the SAR, and then a value is read from the address indicated by the DAR.

Next, whether [(unsigned) SAR] < [(unsigned) DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and DTS0IMM in DTS0CIR is 0.

- When TTYP5 to TTYP2 are 0110 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (DREP or SG) and any chain mode can be selected.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation A value is read from the address indicated by the DAR.

Next, whether (unsigned) SAR < [(unsigned) DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and immediate function is enabled (DTS0IMM in DTS0CIR is 1).

- When TTYP5 to TTYP2 are 1110 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended comp[<]”, the DTS cycle becomes a comparison [<] cycle.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation Firstly, a value is read from the address indicated by the SAR, and then a value is read from the address indicated by the DAR.

Next, whether [(unsigned) SAR] < [(unsigned) DAR] is checked.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1110 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation The value for comparison is read from the address indicated by the DAR, and is then compared with the value in the SAR.

Transfers are always chained.

(8) comp[>]: When TTYP5 to TTYP2 are 0111 or 1111

- When TTYP5 to TTYP2 are 0111 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “comp[>]”, the DTS cycle becomes a comparison [>] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

  A special function (SREP, DREP, or SG) and any chain mode can be selected.

  With comp[>], values read from addresses indicated by the SAR are compared with values read from addresses indicated by the DAR, and the result of comparison is “true” if the data read from the address indicated by the SAR is greater.

  The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation Firstly, a value is read from the address indicated by the SAR, and then a value is read from the address indicated by the DAR.

Next, whether [(unsigned) SAR] > [(unsigned) DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and DTS0IMM in DTS0CIR is 0.

- When TTYP5 to TTYP2 are 0111 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and a value up to 255 is specified as the transfer count in DTS0STC.

  When the value of DTS0STC reaches 0, the value of DTS0STR is copied to DTS0STC and further transfer can proceed.

A special function (DREP or SG) and any chain mode can be selected.

The result of checking can be used as a condition for chaining or to produce interrupt output.

Operation A value is read from the address indicated by the DAR.

Next, whether (unsigned) SAR > [(unsigned) DAR] is checked.

The transfer cycle is the same as when TTYP5 to TTYP2 are 0100 and immediate function is enabled (DTS0IMM in DTS0CIR is 1).

- When TTYP5 to TTYP2 are 1111 and DTS0IMM in DTS0CIR is 0

  When the transfer type is set to “extended comp[>]”, the DTS cycle becomes a comparison [>] cycle.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation Firstly, a value is read from the address indicated by the SAR, and then a value is read from the address indicated by the DAR.

Next, whether [(unsigned) SAR] > [(unsigned) DAR] is checked.

Transfers are always chained.

- When TTYP5 to TTYP2 are 1111 and immediate function is enabled (DTS0IMM in DTS0CIR is 1)

  When DTS0IMM is 1, DTS0SAR becomes IMM, and DTS0SAR indicates the data to be compared.

  DTS0TCEA becomes DTS0EA, and has the function of specifying an offset address for use if the result of checking is “false”.

  Accordingly, a transfer count is not specifiable. When the result of checking is “true”, the TI is fetched from the current TI-fetching address plus 10_H for chaining of the transfer. When the result of checking is “false”, the TI is fetched from the current TI-fetching address plus the value in DTS0EA for chaining of the transfer.

  Neither a chain mode nor a special function can be selected.

Operation The value for comparison is read from the address indicated by the DAR, and is then compared with the value in the SAR.

Transfers are always chained.

5.11.6 Special Functions

SREP, DREP, and SG are supported as special functions. The special function is selected by the setting of the DTS0TTYP1 and DTS0TTYP0 bits in DTS0CIR.

(1) SREP (Source Address Repeat)

This function restores the initial value to DTS0SAR once the value of DTS0STC has reached 0. The TI-D area is used.

If you wish to use this function, set a value equal to the DTS0SAR value in advance in the TI-D area.

DTS cycles are executed until the DTS0STC becomes 0. When the last transfer proceeds after that, the value from the TI-D area is read out to DTS0SAR, and writing-back is performed to achieve the SREP.

With SREP, when the value in DTS0STC becomes 0, writing-back to TI-B always proceeds (i.e. this is never skipped).

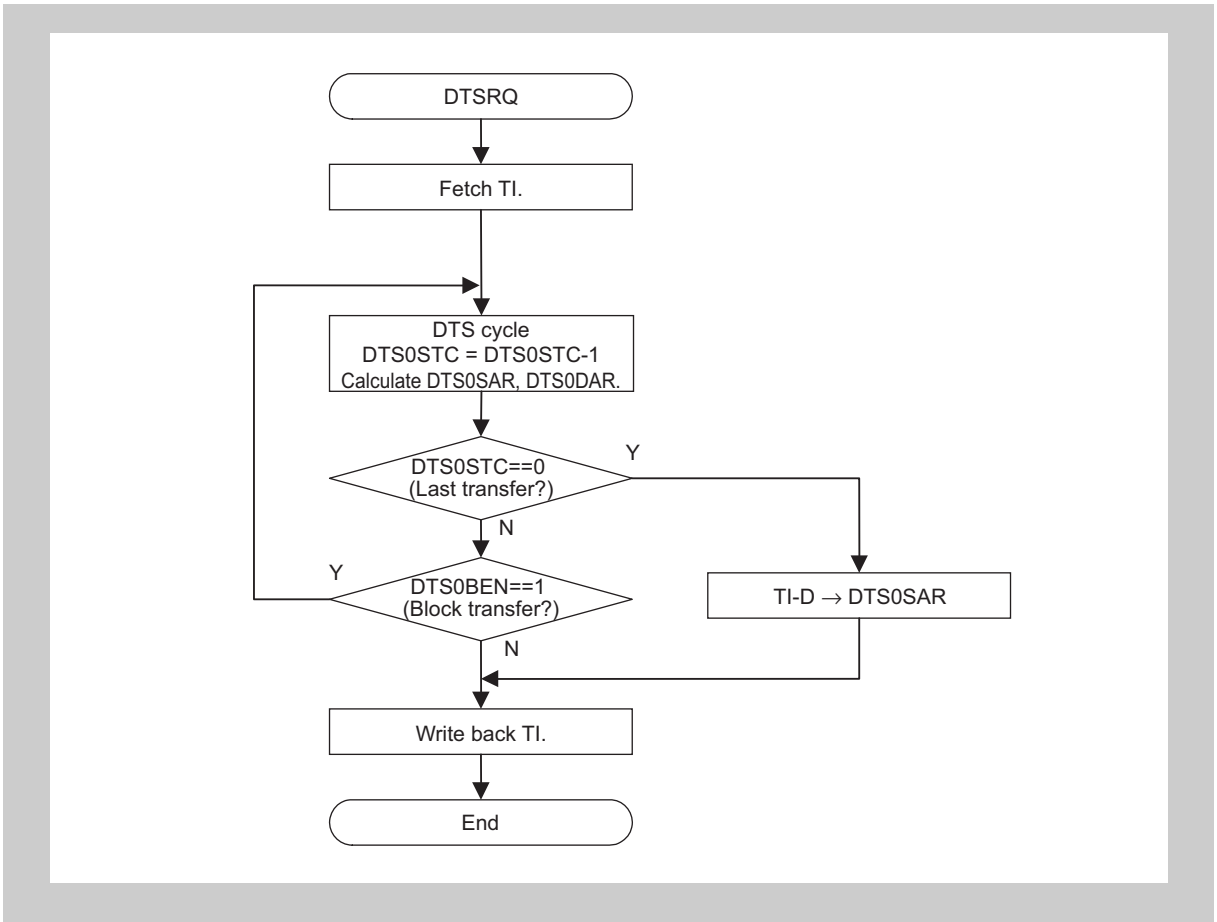


Figure 5-26 SREP Processing Flow

(2) DREP (Destination Address Repeat)

This function restores the initial value to DTS0DAR once the value of DTS0STC has reached 0. The TI-D area is used.

The value is restored in the same way as with SREP, and the TI-D value is written back to DTS0DAR.

With DREP, when the value in DTS0STC becomes 0, writing-back of TI-C always proceeds (i.e. this is never skipped).

Note SREP and DREP cannot be set at the same time (since the TI-D area is used).

(3) SG (Scatter and Gather)

In this mode, the sizes of the increments or decrements to source and destination addresses are specified as signed 16-bit values. The TI-D area is used. In modes other than SG, the increment or decrement for address updating by the DTS size can only be specified as a value in the range from -4 to +4 by using DTS0SCS and DTS0DCS. In SG mode, however, DTS0ESCS and DTS0EDCS are used, substantially increasing the sizes of increments and decrements to addresses (values from the TI-D area are read out to DTS0ESCS and DTS0EDCS, internal registers of the DTS).

5.11.7 Transfer Count

Depending on the type of operation, the transfer count is up to 255 or up to 65,535. In either case, when the specified number of transfers is completed, an interrupt request is output to DTSFSL.

(1) Maximum 255 Times

Although the maximum transfer count is 255, the initial value for the transfer count can be restored when the transfer counter (DTS0STC) reaches 0, so unlimited transfers can in fact be handled in these cases.

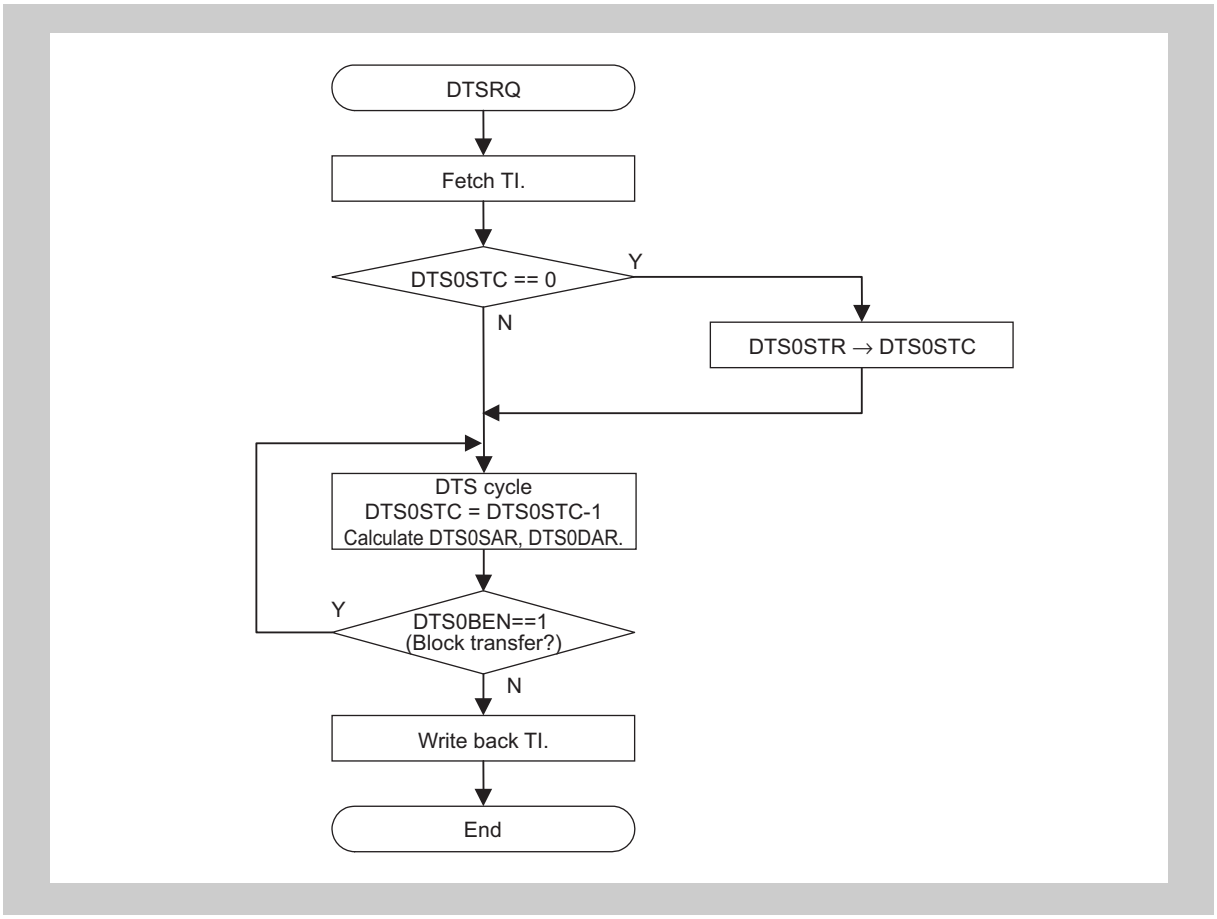


Figure 5-27 Transfer Count Processing Flow

(2) Maximum 65535 Times

When DTS0TCEA is used as a 16-bit counter, a transfer count of up to 65,535 can be specified.

5.11.8 Chain Function

DTS includes a chain function for performing multiple successive data transfers.

A condition for the chain function can be set by using the DTS0CM1 and DTS0CM0 bits in DTS0CIR. Also, the DTS0TTYP4 to DTS0TTYP0 bits in DTS0CIR can be used to select the extended chain function.

When the extended chain function (special flag[0], special flag[1], or special comp[=, !=, <, >]) has been selected, the settings in DTS0CM1 and DTS0CM0 are ignored. On each completion of the sequence TI fetch → DTS cycle → TI write back, the chain function fetches TI and thus achieves the chain of transfers. The address from which the TI is fetched is always TI-D + 4 even when only TI-A to TI-C are in use (consecutive addresses in on-chip RAM).

Table 5-10 List of Chain Functions (1/2)

Transfer Mode	Special Mode	Settings of DTS0CM1 and DTS0CM0 in DTS0CIR	Operation
Transfer	—	00 (No chaining)	Transfers are not chained.
		01 (Last transfer)	Transfers are chained when DTS0STC (or DTS0LTC when the maximum value for counting is 65535) is 0.
		10 (Chain if true)	Transfers are not chained.
		11 (Always chain)	Transfers are chained following each DTS cycle. For block transfer, this setting is the same as “01”.
Addition	—	00 (No chaining)	Transfers are not chained.
		01 (Last transfer)	Transfers are chained when DTS0STC (or DTS0LTC when the maximum value for counting is 65535) is 0.
		10 (Chain if true)	Transfers are not chained.
		11 (Always chain)	Transfers are chained following each DTS cycle. For block transfer, this setting is the same as “01”.
Flag checking	None	00 (No chaining)	Transfers are not chained.
	SREP	01 (Last transfer)	Transfers are chained when the value in DTS0STC reaches 0.
	DREP	10 (Chain if true)	Transfers are only chained if the condition is true. An interrupt is generated if it is false.
		11 (Always chain)	Transfers are chained following each DTS cycle. For block transfer, this setting is the same as “01”.
	Extended chain	Ignored	Transfers are always chained. If the condition is true, TI is fetched from the current address for TI fetching plus 10_H for chaining of the transfer (i.e. in the same manner as in normal chain transfer). If the condition is false, the TI is fetched from the current address for TI fetching plus ±DTS0TCEA for chaining of the transfer.

Table 5-10 List of Chain Functions (2/2)

Transfer Mode	Special Mode	Settings of DTS0CM1 and DTS0CM0 in DTS0CIR	Operation
Comparison	None	00 (No chaining)	Transfers are not chained.
	SREP	01 (Last transfer)	Transfers are chained when the value in DTS0STC reaches 0.
	DREP	10 (Chain if true)	Transfers are only chained if the condition is true. An interrupt is generated if it is false.
		11 (Always chain)	Transfers are chained following each DTS cycle. For block transfer, this setting is the same as “01”.
	Extended chain	Ignored	Transfers are always chained. If the condition is true, TI is fetched from the current address for TI fetching plus 10_H for chaining of the transfer (i.e. in the same manner as in normal chain transfer). If the condition is false, the TI is fetched from the current address for TI fetching plus ±DTS0TCEA for chaining of the transfer.

5.11.9 TI Hold Function

The DTSHENm (m = 0 to 3) register of DTSFSL can be used to enable the TI hold function (i.e. switch this function ON).

(1) Single Transfer when TI Hold Is ON

When the setting of the TI-hold function is “ON” and operation is in single-transfer mode, the sequence “TI fetch → DTS cycle” is executed once per DTS request, after which the transfer request wait status flag is set (the TI is set to the TI-hold state). After this, if a request for DTS transfer on another channel arrives, the sequence “TI fetch → DTS cycle → TI write back” is executed for that channel. If a DTS request arrives for the same channel as that for which the TI is being held, processing goes as far as reading the BT (base table), but since the TI-hold state is already set, TI fetching does not proceed for the DTS cycle.

This operation is repeated until the value of DTS0STC (or DTS0LTC if the maximum value for counting is 65,535) reaches 0.

When DTS0STC (or DTS0LTC) reaches 0, the TI-hold function is set to “OFF” and the TI is written back.

Since the bus is not locked, a request for a CPU-access cycle may interrupt transfer.

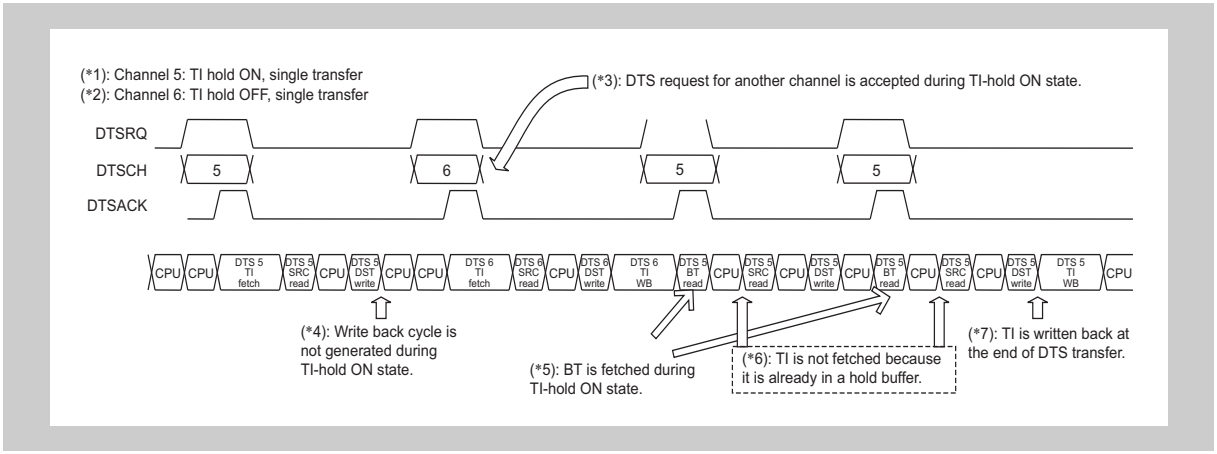


Figure 5-28 TI Hold Function

(2) Number of TI Hold Buffers

Four TI-hold buffers are provided. The hold instruction for the hold buffers is specified in the DTSHENm (m = 3 to 0) register of DTSFSL.

5.11.10 Interrupt Output Function

The DTS is able to output interrupts to the CPU via the DTSFSL block. The DTS sends interrupt source signals to the DTSFSL block, which then decodes the signals and conveys them to the CPU as interrupt signals.

The conditions for enabling the interrupt source signal (DTSINTRQ) involve the transfer count, result of flag checking, and result of comparison.

For the transfer count, DTSINTRQ is asserted upon completion of a single remaining transfer.

With regard to the results of flag checking and comparison, if the result of either operation is false, DTSINTRQ is asserted upon completion of the corresponding transfer.

The interrupt source signal (DTSINTRQ) can be masked by using the DTS0DISI bit in DTS0CIR.

DTSINTRQ is not asserted when the DTS0DISI bit in DTS0CIR is 1.

For details, refer to section 5.12, DTSFSL Function.

5.11.11 Interrupt and Chain of Transfers during Block Transfer

In the case of block transfer, judgments concerning interrupts and chaining of transfers only proceeds once, on completion of the number of transfers specified in DTS0TCEA.

With regard to interrupts, an interrupt is always output when the specified number of transfers has been completed (the counter reaches 0). Note that interrupts are maskable by using DTS0DISI in DTS0CIR.

The chaining of transfers always proceeds when DTS0CM1 and DTS0CM0 in DTS0CIR are 01 (chaining when counter reaches 0) or 11 (always chained). When these bits are 10 (chain if true), transfers are only chained when all results of flag checking and comparison are true (and chaining of transfers does not proceed if any result is false).

The DTS0TCF bit in DTS0CIR becomes 1 if any of the results of flag checking and comparison is false.

5.11.12 TI Write-back Skipping Function

The DTS skips any unnecessary writing back of TIs. The conditions for this are as follows.

Table 5-11 Conditions for Skipping Write-back

	DTS0CIR				Transfer Count	SCS	SAM
	DS	IMM	Type	Special Function			
TI-B write back skip conditions	—	—	—	SG	—	—	0
	—	0	Other than addition	SREP	Pending transfers remain	0	—
	—	0	—	DREP/none	—	0	—
	32 bits	1	Other than addition	SREP	Pending transfers remain	—	—
	8/16 bits	1	Other than addition	SREP	Pending transfers remain	0	—
	32 bits	1	—	DREP/none	—	—	—
	8/16 bits	1	—	DREP/none	—	0	—

	DTS0CIR		Transfer Count	SCS	SAM
	Type	Special Function			
TI-C write back skip conditions	—	SG	—	—	0
	—	SREP	—	0	—
	Other than addition	DREP	Pending transfers remain	0	—
	—	None	—	0	—

5.11.13 DTS Channel Priority Control

The DTSFSL has a priority function that can be used to set up to 4 priority levels.

5.11.14 Valid DTS Transfer Request Conditions

When the DTSEN bit in DTSEN000 to DTSENj of DTSFSL is set to 0, transfer requests for the DTS are not generated for the corresponding channel, but sampling of the DTS transfer sources still proceeds. When sampled, if the DTSEN bit is set to 1, a DTS transfer request is generated.

When the DTSEN bit is 1, if an INTIN is generated while the DTS is transferring data or stopped either by an ELBERR response or by the user, the INTIN is held pending in DTSFSL until the DTS is able to accept it.

Table 5-12 Valid DTS Transfer Request Conditions

DTS	DTS	DTSFSL		Transfer Request by DTS Transfer Sources
DTS0TSR.Bit 1	DTS0TSR.Bit 0	DTSEN	Status of DTS	
x	x	0	—	Invalid (held)
0	0	1	Acceptable	Valid (if a next request arrives during transfer, it is held)
x	1	1	Not acceptable	Held in DTSFSL
1	x	1	Not acceptable	Held in DTSFSL

5.11.15 Suspending and Resuming DTS Transfer

Table 5-13 DTS Transfer Suspension and Resumption Timing

Transfer Mode	Transfer Count	Condition for Suspension
Single transfer with TI hold OFF	Last transfer	Writing back the TI
	Pending transfers remain	
Block transfer with TI hold OFF	Last transfer	Writing back the TI
	Pending transfers remain	DTS cycle
Single transfer with TI hold ON	Last transfer	Writing back the TI
	Pending transfers remain	DTS cycle
Block transfer with TI hold ON	Last transfer	Writing back the TI
	Pending transfers remain	DTS cycle

(1) Using Hardware to Suspend and Resume DTS Transfer on All Channels (NMI)

This product does not support this function.

(2) Using Software to Suspend and Resume DTS Transfer on All Channels

Setting the DTS transfer stop request trigger bit (DTS0UST) in the DTS transfer request control register (DTS0TRC) to 1 suspends subsequent DTS transfer. If this occurs during a DTS cycle, subsequent DTS transfer is suspended in accord with the conditions in the table above.

To resume suspended DTS transfer, set the DTS0UCL bit in the DTS transfer request control register (DTS0TRC) to 1, and then clear the DTS0STPU bit to 0. If a DTS transfer request arrives at this point, the DTS transfer is resumed.

(3) Using Software to Suspend and Resume DTS Transfer on Individual Channels

To suspend DTS transfer, clear the DTS transfer request enable bit (DTSEN) in the DTS transfer enable register (DTSENn) to 0. This leads to the suspension of subsequent DTS transfer. If this occurs during a DTS cycle, subsequent DTS transfer is suspended in accord with the conditions in the table above. To resume suspended DTS transfer, set the DTSENn.DTSEN bit to 1. If a DTS transfer request arrives at this point, the DTS transfer is resumed.

5.11.16 Error Response Support

(1) Suspending DTS Transfer in Response to an Error

When the DMAT receives an error response from the internal bus, the DTS cycle end signal (CYCEND) and the internal bus transfer-error detection signal (ELBERR) are asserted at the same time. If ELBERR is being detected as the high level, the DTS sets the DTS0STPE bit in the DTS transfer status register (DTS0TSR) to 1, which immediately suspends DTS transfer. Subsequent requests for DTS transfer are not accepted. Meanwhile, the DTS transfer error interrupt signal (DTSERR) is asserted for 1 clock cycle. Suspension takes place as soon as an error response is received regardless of the transfer mode and whether holding of TIs is ON or OFF.

The user can learn in which channel the error has occurred, by reading the DTS active channel register, after confirming that DTS0STPE has been set.

Note 1. If an error response is received on a channel with TI holding ON, the TI is not written back.

Note 2. It is not possible to determine the cycle (TI fetching, etc.) which led to an error response.

(2) Canceling Transfer Suspension Caused by Error Response

To release the DTS from suspension caused by an error response, check the DTS active channel register (DTS0ACR), use the DTS0TIT and DTS0ICH bits in the DTS initialization control register (DTS0ICR) to clear the TI for the channel on which the error occurred, and then set the DTS0ECL bit in the DTS transfer request control register (DTS0TRC) to 1, which clears DTS0STPE to 0.

5.12 DTSFSL Function

The DTS factor selector (DTSFSL) block selects DTS trigger sources from among the interrupt signals. Four priority levels can be set for DTS trigger sources.

Sources that are not in use with the DTS can be input to the interrupt controller via DTS transfer-source-through outputs (IRQ[j:0]).

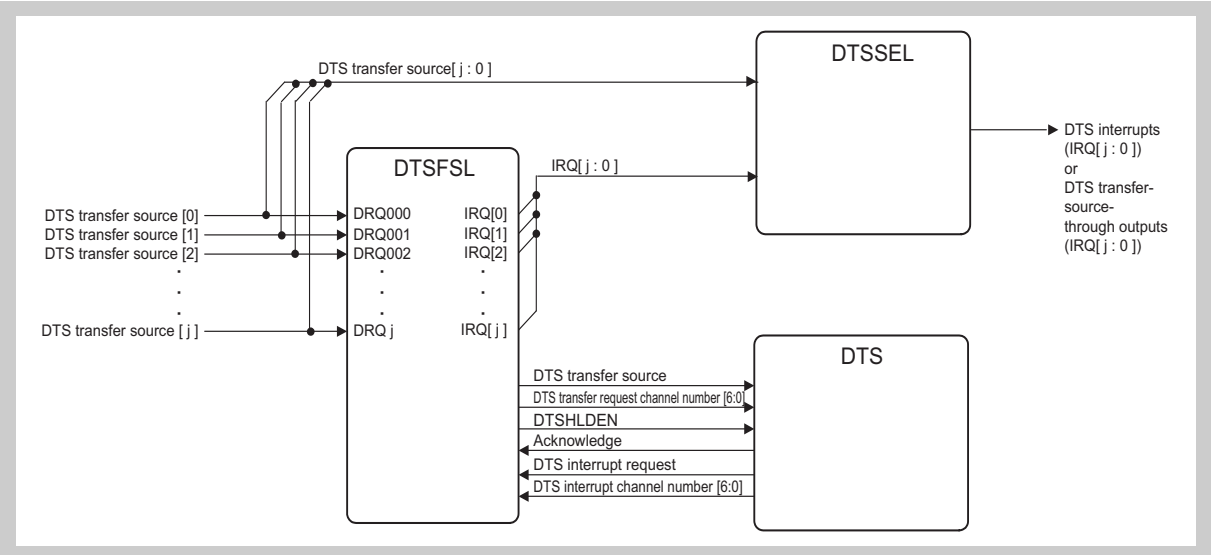


Figure 5-29 Connections between DTSFSL and DTS

5.12.1 Features

"k" (112) sources are assignable as sources for initiating DTS transfer. Any of 4 priority levels can be set for each of these sources.

On generation of an interrupt request source condition in the DTS, the DTS interrupt (IRQ (j: 0)) corresponding to the particular channel is output (shared output) to the CPU.

Note The types of interrupt request sources from the DTS are "full transfer count completed", "result of flag checking is false", and "result of comparison is false".

5.13 DTSFSL Control Registers

5.13.1 DTSENn: DTS Transfer Enable Register

Access This register is readable/writable in 8-bit units.

Address FFFF 7D00$_H$ + 2 × n

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
DRQ	DRQSET	DRQCLR	0	0	DTSEN	DTSPR1	DTSPR0
R	R/W	R/W	R	R	R/W	R/W	R/W

Bit Position	Bit Name	Function
7	DRQ	This bit indicates whether a transfer request to the DTS is pending. This is a read-only bit, and writing to this bit is ignored. 0: No DTS transfer requests are pending. 1: At least one DTS transfer request exists (waiting state). Condition for setting (to 1): Detection of the rising edge of DRQn Condition for clearing (to 0): Writing of 1 to the DRQCLR bit, or acceptance of the pending transfer request by the DTS.
6	DRQSET	Writing 1 to this bit sets the DRQ bit to 1. Writing 0 to this bit is ignored.
5	DRQCLR	Writing 1 to this bit clears the DRQ bit to 0. Writing 0 to this bit is ignored.
2	DTSEN	DTS transfer request enable bit for DRQn 0: DRQn is not in use as a DTS transfer request (DTS requests disabled). 1: DRQn is in use as a DTS transfer request (DTS requests enabled).
1 0	DTSPR1 DTSPR0	These bits specify the priority order. 00 specifies the highest priority level and 11 specifies the lowest priority level.

5.13.2 DTSHENm (m = 0 to 3): DTS Hold Enable Register

Access This register is readable/writable in 16-bit units.

Address DTSHEN0: FFFF 7E00$_H$, DTSHEN1: FFFF 7E02$_H$, DTSHEN2: FFFF 7E04$_H$, DTSHEN3: FFFF 7E06$_H$

Initial value 0000$_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
HEN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	dtsch[6:0]						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15	HEN	This bit enables/disables operation of the DTS hold-enable register. 1: Enables operation of the hold-enable register. 0: Disables operation of the hold-enable register.
6 to 0	dtsch6 to dtsch0	These bits select the target channel for holding. Settings for these bits are only valid when the setting of the HEN bit is 1. The settings are listed in the table below.

Table 5-14 List of Channels Subject to Holding Selected by Setting dtsch6 to dtsch0

dtsch6 to dtsch0	Holding Channel
000_0000	CH000
000_0001	CH001
000_0010	CH002
000_0011	CH003
000_0100	CH004
000_0101	CH005
000_0110	CH006
000_0111	CH007
000_1000	CH008
000_1001	CH009
000_1010	CH010
000_1011	CH011
000_1100	CH012
000_1101	CH013
000_1110	CH014
000_1111	CH015
001_0000	CH016
001_0001	CH017
001_0010	CH018
001_0011	CH019
001_0100	CH020
001_0101	CH021
001_0110	CH022
001_0111	CH023
001_1000	CH024
001_1001	CH025
001_1010	CH026
001_1011	CH027
001_1100	CH028
001_1101	CH029
001_1110	CH030
001_1111	CH031
010_0000	CH032
010_0001	CH033
010_0010	CH034
010_0011	CH035
010_0100	CH036
010_0101	CH037
010_0110	CH038
010_0111	CH039
010_1000	CH040
010_1001	CH041
010_1010	CH042
010_1011	CH043
010_1100	CH044
010_1101	CH045
010_1110	CH046
010_1111	CH047
011_0000	CH048
011_0001	CH049
011_0010	CH050
011_0011	CH051
011_0100	CH052
011_0101	CH053
011_0110	CH054
011_0111	CH055
011_1000	CH056
011_1001	CH057
011_1010	CH058
011_1011	CH059
011_1100	CH060
011_1101	CH061
011_1110	CH062
011_1111	CH063
100_0000	CH064
100_0001	CH065
100_0010	CH066
100_0011	CH067
100_0100	CH068
100_0101	CH069
100_0110	CH070
100_0111	CH071
100_1000	CH072
100_1001	CH073
100_1010	CH074
100_1011	CH075
100_1100	CH076
100_1101	CH077
100_1110	CH078
100_1111	CH079
101_0000	CH080
101_0001	CH081
101_0010	CH082
101_0011	CH083
101_0100	CH084
101_0101	CH085
101_0110	CH086
101_0111	CH087
101_1000	CH088
101_1001	CH089
101_1010	CH090
101_1011	CH091
101_1100	CH092
101_1101	CH093
101_1110	CH094
101_1111	CH095
110_0000	CH096
110_0001	CH097
110_0010	CH098
110_0011	CH099
110_0100	CH100
110_0101	CH101
110_0110	CH102
110_0111	CH103
110_1000	CH104
110_1001	CH105
110_1010	CH106
110_1011	CH107
110_1100	CH108
110_1101	CH109
110_1110	CH110
110_1111	CH111

5.14 DTSSEL Function (DTS Transfer Completion Interrupt)

The DTSSEL (DTS selector) selects the DTS transfer source signal (when DTS transfer is not in use) or DTS transfer completion interrupt signal (when DTS transfer is in use) for output to the CPU subsystem.

When the DTS transfer completion interrupt is selected by the DTSSEL function, an interrupt is generated on completion of DTS transfer triggered by the DTS source interrupt. On the other hand, when an external or internal interrupt is selected, an interrupt is generated by the DTS source through DTSSEL.

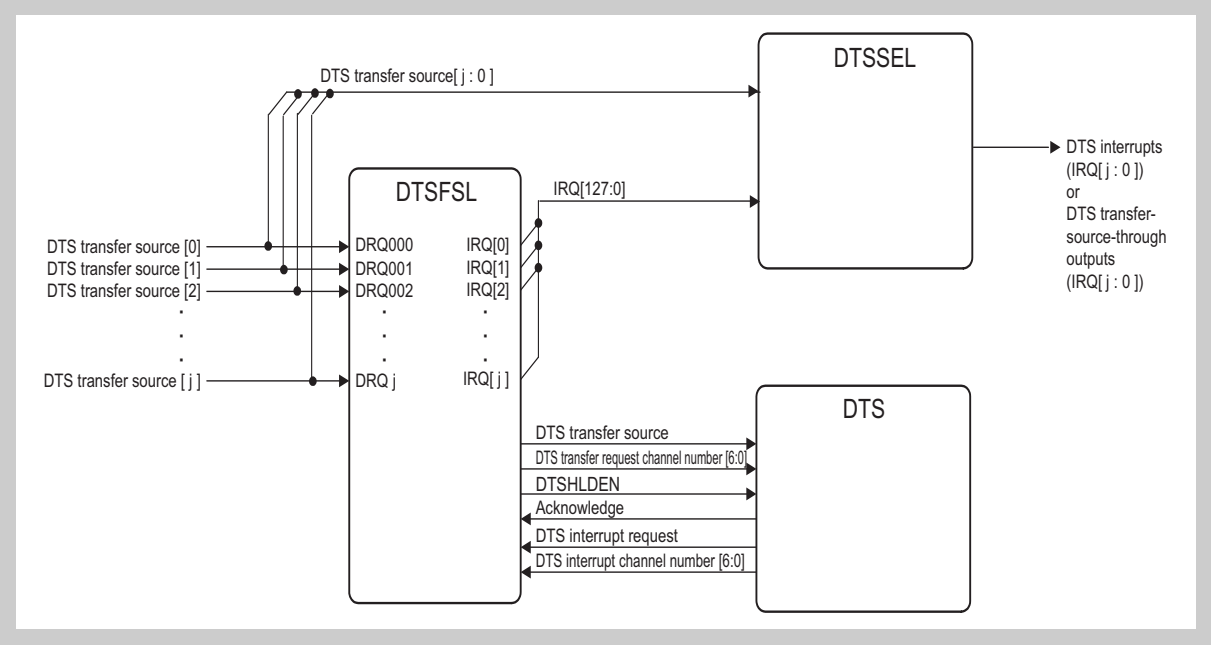


Figure 5-30 Connections between DTSSEL and DTSFSL

5.14.1 Features

Selection between DTS transfer complete interrupt or DTS transfer factor signal is possible. Interrupts are output to the CPU from the interrupt request sources as shown in the diagram above.

The following table is a list of the transfer completion interrupt register addresses.

Table 5-15 List of DTSSEL Registers

Register Name	Symbol	Address
DTSSEL transfer completion interrupt register 0	DTSSEL0	<DTSSEL_base>
DTSSEL transfer completion interrupt register 1	DTSSEL1	<DTSSEL_base> + 04_H
DTSSEL transfer completion interrupt register 2	DTSSEL2	<DTSSEL_base> + 08_H
DTSSEL transfer completion interrupt register 3	DTSSEL3	<DTSSEL_base> + $0C_H$
DTSSEL transfer completion interrupt register 4	DTSSEL4	<DTSSEL_base> + 10_H
DTSSEL transfer completion interrupt register 5	DTSSEL5	<DTSSEL_base> + 14_H
DTSSEL transfer completion interrupt register 6	DTSSEL6	<DTSSEL_base> +18_H
DTSSEL transfer completion interrupt register 7	DTSSEL7	<DTSSEL_base> + $1C_H$
DTSSEL transfer completion interrupt register 8	DTSSEL8	<DTSSEL_base> + 20_H
DTSSEL transfer completion interrupt register 9	DTSSEL9	<DTSSEL_base> + 24_H
DTSSEL transfer completion interrupt register 10	DTSSEL10	<DTSSEL_base> + 28_H
DTSSEL transfer completion interrupt register 11	DTSSEL11	<DTSSEL_base> + $2C_H$
DTSSEL transfer completion interrupt register 12	DTSSEL12	<DTSSEL_base> + 30_H
DTSSEL transfer completion interrupt register 13	DTSSEL13	<DTSSEL_base> + 34_H
DTSSEL transfer completion interrupt register 14	DTSSEL14	<DTSSEL_base> + 38_H

<DTSSEL_base> The following table shows the base address of DTSSEL: <DTSSEL_base>.

Table 5-16 Register Base Address <DTSSEL_base>

DTSSEL	Address of <DTSSEL_base>
DTSSEL	FF81 $F000_H$

5.15 DTSSEL Control Registers

5.15.1 DTSSELn (n = 0 to 14): DTS Transfer Completion Interrupt Register

This 32-bit register selects whether to use transfer completion interrupt.

Access This register is readable/writable in 32-bit units.

Address Refer to table 5-15, List of DTSSEL Registers.

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
DSELh	DSELg	DSELf	DSELe	DSELd	DSELc	DSELb	DSELa
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
7	DSELh	These bits select whether to issue transfer completion interrupts on completion of DTS transfer. 0: DTS transfer is not in use. 1: DTS transfer is in use.
6	DSELg	
5	DSELf	
4	DSELe	
3	DSELd	
2	DSELc	
1	DSELb	
0	DSELa	

Note a = 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104
b = 1, 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105
c = 2, 10, 18, 26, 34, 42, 50, 58, 66, 74, 82, 90, 98, 106
d = 3, 11, 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107
e = 4, 12, 20, 28, 36, 44, 52, 60, 68, 76, 84, 92, 100, 108
f = 5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 93, 101, 109
g = 6, 14, 22, 30, 38, 46, 54, 62, 70, 78, 86, 94, 102, 110
h = 7, 15, 23, 31, 39, 47, 55, 63, 71, 79, 87, 95, 103, 111

DTS transfer setting procedure

Follow the procedure below to enable DTS transfer.

1. Disable the generation of interrupt signals by the DTS transfer source.
2. Make the setting to enable transfer requests in the DTSENn (n = 0 to 127) register of the DTSFSL block.
3. Set the DSEL bit of the DTSSELn (n = 0 to 14) register to select use of the DTS.
4. Enable interrupt generation by the source of the trigger for DTS transfer.

DTS transfer release setting procedure

Follow the procedure below to disable DTS transfer.

1. Disable the generation of interrupt signals by the DTS transfer source.
2. Make the setting to disable transfer requests in the DTSENn (n = 0 to 127) register of the DTSFSL block.
3. Set the DSEL bit of the DTSSELn (n = 0 to 14) register to deselect use of the DTS.
4. Enable interrupt generation by the source of the trigger for DTS transfer.

Section 6 Memory Modules

This section describes the memory modules.

Products of this series are equipped with code flash memory, data flash memory, and RAM. Refer to table 6-1, On-Chip Memory of V850E2/Px4 Products for the memory size per product.

Code flash memory is programmable in two ways: by a flash programmer or by self-programming. Data flash memory is also programmable in two ways: by a flash programmer or by the EEPROM emulation library.

If the dedicated flash programmer is to program on-chip memory, it has to be connected to the target system.

For self-programming, an application program is used instead of the dedicated flash programmer.

Flash memory is commonly used for the following purposes in the development environment and field of application:

- altering software after solder-mounting of the microcontroller on the target system,
- differentiating software in the small-scale production of a variety of models, and
- adjusting data when mass production starts.

Table 6-1 On-Chip Memory of V850E2/Px4 Products

	μPD70F3505A	**μPD70F3506, μPD70F3507**	**μPD70F3508, μPD70F3509**
Code flash	384 Kbytes	512 Kbytes	1 Mbyte
Data flash	32 Kbytes	32 Kbytes	32 Kbytes
On-chip RAM	24 Kbytes	40 Kbytes	80 Kbytes

6.1 Features

- All-block batch erasure or single-block erasure
- Communicating with the dedicated flash programmer via a serial interface
- Voltage for erasure/programming: erasure and programming only require a single power supply
- On-board programming
- Self-programming of the flash memory

6.1.1 Code Flash Memory

- ROM capacity: 384 Kbytes, 512 Kbytes, or 1 Mbyte
- Erasure unit: single block (each block taking up 4 Kbytes)
- ECC function

  Refer to section 9, Safety Functions, for details.
- Methods of erasure and programming
  - Support for Nexus
  - Support for serial interfaces (one-line UART and three-line CSI-HS)
  - Support for self-programming
- Other supported functions
  - Prohibition of erasure and programming (security function)
  - Boot changeover (boot swapping)

Table 6-2 Code Flash Memory of V850E2/Px4 Products

	Size	Address
µPD70F3505A	384 Kbytes	0000 0000_H to 0005 $FFFF_H$
µPD70F3506, µPD70F3507	512 Kbytes	0000 0000_H to 0007 $FFFF_H$
µPD70F3508, µPD70F3509	1 Mbyte	0000 0000_H to 000F $FFFF_H$

6.1.2 Data Flash Memory

- ROM capacity: 32 Kbytes
- Erasure unit: single block (each taking up 2 Kbytes)
- Methods of erasure and programming
  - Dedicated flash programmer
  - EEPROM emulation library

Table 6-3 Data Flash Memory of V850E2/Px4 Products

	Size	Address
μPD70F3505A	32 Kbytes	0200 0000_H to 0200 $7FFF_H$
μPD70F3506, μPD70F3507	32 Kbytes	0200 0000_H to 0200 $7FFF_H$
μPD70F3508, μPD70F3509	32 Kbytes	0200 0000_H to 0200 $7FFF_H$

6.1.3 On-Chip RAM

- RAM capacity: 24 Kbytes, 40 Kbytes, or 80 Kbytes

Table 6-4 On-Chip RAM of V850E2/Px4 Products

	Size	Address
μPD70F3505A	24 Kbytes	FEDF $A000_H$ to FEDF $FFFF_H$
μPD70F3506, μPD70F3507	40 Kbytes	FEDF 6000_H to FEDF $FFFF_H$
μPD70F3508, μPD70F3509	80 Kbytes	FEDE $C000_H$ to FEDF $FFFF_H$

6.2 Programming Environment

Programming, erasure, etc. are handled through a serial interface between the dedicated programmer and this product. Supply the operating clock for this product by mounting oscillators and capacitors to configure an oscillation circuit on the same board as this product.

6.3 Communications Methods

(1) Nexus communications

Transfer rate: 25 MHz (max.)

(2) One-line UART communications

Transfer rate: 1 Mbps (max.)

(3) Three-line CSI-HS communications

Transfer rate: 5 MHz (max.)

6.4 Handling of Pins

For on-board programming, the target system requires a connector for connection to the dedicated flash programmer.

When the transition is made to flash memory programming mode, all of the pins which are not used for flash memory programming are in the same state as just after a reset from normal operating mode (single-chip mode). Consequently, the port pins are all in the high-impedance state. Pin settings must be adjusted if this is not acceptable for an external device.

6.4.1 Power Supply

Supply the same voltage as in the normal operating mode.

Stabilize the power voltage by inserting capacitors between power-supply pins and ground during operations in flash memory programming mode (including flash-self programming).

6.4.2 Pins

Pins to be used for each interface are as follows:

- Nexus: FLMD0, $\overline{\text{RESET}}$, $\overline{\text{DCUTRST}}$, DCUTCK, DCUTMS, DCUTDI, DCUTDO, $\overline{\text{DCUTRDY}}$
- One-line UART: RxD/ TxD, FLMD0, $\overline{\text{RESET}}$
- Three-line CSI-HS: SI, SO, SCK, FLMD0, $\overline{\text{RESET}}$

When connecting pins for interfaces which are already connected to other devices on the board to the dedicated flash programmer, take care to avoid signal conflicts, abnormal operation of other devices, and so on.

6.4.3 Reset Pin

Connecting the reset signal of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin, which in turn is connected to the reset-signal generation circuit on the board, leads to signal conflicts. To avoid these conflicts, isolate the pin's connection with the reset generation circuit.

A reset signal from the user system input during operation in flash memory programming mode can cause abnormal programming operation. Therefore, do not input any signals other than the reset signal from the dedicated flash programmer.

6.4.4 FLMD0 Pin

Input the high level on FLMD0 except for intervals over which pulses are input during operations in flash memory programming mode (programming, erasure, or reading). After the pulse input, keep the pin at the high level until flash-memory operations are completed.

6.4.5 Port Pins

All port pins other than those for communications with the dedicated flash programmer will be in the high-impedance state when the settings for flash-memory programming mode are made. No adjustment is required for these pins.

However, if a pin being in the high-impedance state is not acceptable for an external device to which it is connected, connect the port pin to EVDD or EVSS via a resistor.

6.5 Option-Setting Bytes

The “option bytes” specify product operation. Values of these bytes are programmed by using the dedicated flash programmer or a function for writing to them.

Erasing the chip leads to erasure of the option bytes. The bytes thus have to be programmed as required.

Operation of the product with option bytes erased is not guaranteed.

The option byte settings must produce a frequency in the corresponding range and these settings will differ with the operating frequency settings.

Note When a chip is erased, the setting of the FOP (option byte) is initialized (HEAPCLK = 48 MHz, PCLK = 1/4 HEAPCLK). Therefore, remake the FOP settings so that the serial communications speed is within the corresponding range below.

- One-line UART:
  Baud rate ≤ 750 kbps
- Three-line CSI-HS:
  SCK ≤ 2 MHz

6.5.1 OPBT0 - Option Byte Verification Register

This register is used to check the settings of option bytes.

Access This register can be read in 32-bit units.

Address FF47 000C$_H$

Initial value The corresponding value is set when the source signal of any reset other than the software reset is generated.

FFFF FFC3$_H$ (µPD70F3505A, µPD70F3506, µPD70F3507:
when HEAPCLK = 80 MHz, PCLK = 80 MHz, and FlexRay is disabled)

FFFF FFEA$_H$ (µPD70F3508, µPD70F3509:
when HEAPCLK = 160 MHz, PCLK = 80 MHz, and FlexRay is enabled)

(1) μPD70F3505A, μPD70F3507

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FOP 31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	FOP5	FOP4	FOP3	FOP2	FOP1	1

Table 6-5 Option Byte Verification Register of V850E2/Px4 Products

Bit Position	Bit Name	Function
31	FOP31	Switches the JTAG port 0: Enabled 1: Disabled When connecting the port to a development tool, set this bit to 1 (= disabled).
5	FOP5	Sets enabling or disabling of the FlexRay clock (SCLK2) 0: Disabled 1: Enabled Set the bit to 0 if the product does not have FlexRay.
4, 3	FOP[4:3]	Sets the peripheral clock (PCLK) **FOP4 / FOP3 / PCLK** 0 / 0 / HEAPCLK 0 / 1 / HEAPCLK/2 1 / 0 / Setting prohibited 1 / 1 / Setting prohibited
2, 1	FOP[2:1]	Sets the CPU clock (HEAPCLK) **FOP2 / FOP1 / HEAPCLK** 0 / 0 / 64 MHz 0 / 1 / 80 MHz 1 / 0 / Setting prohibited 1 / 1 / Setting prohibited

(2) µPD70F3506

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FOP 31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	FOP5	FOP4	FOP3	FOP2	FOP1	1

Table 6-6 Option Byte Verification Register of V850E2/Px4 Products

Bit Position	Bit Name	Function
31	FOP31	Switches the JTAG port 0: Enabled 1: Disabled When connecting the port to a development tool, set this bit to 1 (= disabled).
5	FOP5	Sets enabling or disabling of the FlexRay clock (SCLK2) 0: Disabled 1: Enabled
4, 3	FOP[4:3]	Sets the peripheral clock (PCLK)
2, 1	FOP[2:1]	Sets the CPU clock (HEAPCLK)

FOP4	FOP3	PCLK
0	0	HEAPCLK
0	1	HEAPCLK/2
1	0	Setting prohibited
1	1	Setting prohibited

FOP2	FOP1	HEAPCLK
0	0	64 MHz
0	1	80 MHz
1	0	Setting prohibited
1	1	48 MHz

(3) μPD70F3508, μPD70F3509

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FOP 31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	FOP5	FOP4	FOP3	FOP2	FOP1	FOP0

Table 6-7 Option Byte Verification Register of V850E2/Px4 Products

Bit Position	Bit Name	Function
31	FOP31	Switches the JTAG port 0: Enabled 1: Disabled When connecting the port to a development tool, set this bit to 1 (= disabled).
5	FOP5	Sets enabling or disabling of the FlexRay clock (SCLK2) 0: Disabled 1: Enabled
4, 3	FOP[4:3]	Sets the peripheral clock (PCLK)
2 to 0	FOP[2:0]	Sets the CPU clock (HEAPCLK)

FOP4	FOP3	PCLK
0	0	Setting prohibited
0	1	HEAPCLK/2
1	0	HEAPCLK/4*
1	1	HEAPCLK/4*

FOP2	FOP1	FOP0	HEAPCLK
0	1	0	160 MHz
0	1	1	80 MHz
Other than the above			Setting prohibited

Note * The settings for PCLK = HEAPCLK/4 are only available when the HEAPCLK is running at 160 MHz.

6.6 Product Identification

(1) PRDNAME Register

This register provides identifying information for the product.

Access This register can be read in 32-bit units.

Address FF47 0028$_H$

Initial value 0DB11100$_H$ (version 1.0 of the µPD70F3505A)

0DB20100$_H$ (version 1.0 of the µPD70F3506)

0DB30100$_H$ (version 1.0 of the µPD70F3507)

0DB40100$_H$ (version 1.0 of the µPD70F3508)

0DB50100$_H$ (version 1.0 of the µPD70F3509)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	1	1	0	1	1	0	1	1	0	0/1	0/1	1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6-8 PRDNAME Register Contents

Bit Position	Function
31 to 12	**Bits 31 to 12** / **Description** 0DB11$_H$ / µPD70F3505A 0DB20$_H$ / µPD70F3506 0DB30$_H$ / µPD70F3507 0DB40$_H$ / µPD70F3508 0DB50$_H$ / µPD70F3509
11 to 0	The information on the product version is set in these bits.

(2) PRDSELH Register

This register provides identifying information for the product.

Access This register can be read in 32-bit units.

Address FF47 0024$_H$

Initial value FEDFA001$_H$ (µPD70F3505A)

FEDF6002$_H$ (µPD70F3506, µPD70F3507)

FEDEC004$_H$ (µPD70F3508, µPD70F3509)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	0/1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0/1	0/1	0/1	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6-9 PRDSELH Register Contents

Bit Position	Function
31 to 8	**31 to 8 Bits** / **Description** FEDFA0$_H$ — Address where on-chip RAM starts in the µPD70F3505A (FEDF A000$_H$) FEDF60$_H$ — Address where on-chip RAM starts in the µPD70F3506, µPD70F3507 (FEDF 6000$_H$) FEDEC0$_H$ — Address where on-chip RAM starts in the µPD70F3508, µPD70F3509 (FEDE C000$_H$)
7 to 0	**7 to 0 Bits** / **Description** 01$_H$ — The µPD70F3505A has 384 Kbytes of code flash memory. 02$_H$ — The µPD70F3506 and µPD70F3507 have 512 Kbytes of code flash memory. 04$_H$ — The µPD70F3508 and µPD70F3509 have 1 Mbyte of code flash memory.

(3) PRDSELL Register

This register provides identifying information for the product.

Access This register can be read in 32-bit units.

Address FF47 0020_H

Initial value $800020FF_H$ (µPD70F3505A)

$000020FF_H$ (µPD70F3506, µPD70F3507)

$000020FF_H$ (µPD70F3508, µPD70F3509)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0/1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6-10 PRDSELL Register Contents

Bit Position	Function
31 to 24	**31 to 24 Bits** / **Description** 80_H / The µPD70F3505A has 384 Kbytes of code flash memory. 00_H / The µPD70F3506 and µPD70F3507 have 512 Kbytes of code flash memory. 00_H / The µPD70F3508 and µPD70F3509 have 1 Mbyte of code flash memory.
23 to 8	**23 to 8 Bits** / **Description** $00\ 20_H$ / The product has 32 Kbytes of data flash memory.

6.7 Setting the FLMD Pin

Input the high level on the FLMD0 pin when flash-memory operations are to proceed. Use of the on-chip pull-up resistor is selectable by setting a bit in the FLMD control register (the FLMDCNT.FLMDPUP bit) to 1.

Table 6-11 List of FLMD Pin Setting Registers

Register Name	Symbol	Address
FLMD control register	FLMDCNT	FF43 8000$_H$
FLMD protection command register	FLMDPCMD	FF43 8004$_H$
FLMD protection error status register	FLMDPS	FF43 8008$_H$

6.7.1 Registers

(1) FLMDCNT - FLMD Control Register

This eight-bit register specifies pulling the FLMD0 pin up or down. A requirement for writing in a specified sequence protects the register. When programming is not in the specified sequence, the FLMDPREPP bit in the FLMDPS register is set to 1 to indicate the protection error.

For details, refer to section 6.7.2, Setting the FLMDCNT Register.

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

Address FF43 8000$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FLMDPUP
R	R	R	R	R	R	R	R/W

Table 6-12 FLMDCNT Register Contents

Bit Position	Bit Name	Function
0	FLMDPUP	Specifies pulling up or pulling down for the FLMD0 pin 0: FLMD0 pin is pulled down 1: FLMD0 pin is pulled up

(2) FLMDPCMD - FLMD Protection Command Register

This is the protection command register for FLMDCNT register.

For details, refer to section 6.7.2, Setting the FLMDCNT Register.

Access This register can be written in 8-bit units. The value read is always 00_H.

Address $FF43\ 8004_H$

Initial value Undefined

7	6	5	4	3	2	1	0
FLMDPCMD[7:0]							
W	W	W	W	W	W	W	W

Table 6-13 FLMDPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	FLMDPCMD[7:0]	Protection commands to enable writing to FLMDCNT

(3) FLMDPS - FLMD Protection Error Status Register

This register verifies whether writing to the write protection register (FLMDCNT) was successful or not.

For details, refer to section 6.7.2, Setting the FLMDCNT Register.

Access This register can be read in 8-bit units.

Address $FF43\ 8008_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FLMD PRERR
R	R	R	R	R	R	R	R

Table 6-14 FLMDPS Register Contents

Bit Position	Bit Name	Function
0	FLMDPRERR	Indicates whether writing to the write protection register (FLMDCNT) was successful or not 0: Writing succeeded 1: Writing failed

6.7.2 Setting the FLMDCNT Register

The FLMDCNT register is protected against writing. Specifically, writing is only effective when performed in the following sequence.

STEP1 Write the fixed value ($A5_H$) to the protection register (FLMDPCMD).

STEP2 Write the new setting to the corresponding register (FLMDCNT).

STEP3 Write the inverse of the new setting to the corresponding register (FLMDCNT).

STEP4 Write the new setting to the corresponding register (FLMDCNT).

STEP5 Confirm that the setting has been written to the protected register by checking that the FLMDPS.FLMDPRERR bit is 0.

Writing to an FLMD pin setting register (for details, refer to table 6-11, List of FLMD Pin Setting Registers) between steps 1 and 4 of the sequence specified above leads to failure in writing to the protected register (, which is indicated with the FLMDPS.FLMDPRERR set to 1).

In such a case, the sequence has to be restarted from step 1.

Access to registers other than FLMD pin setting registers can proceed during the sequence without preventing its completion.

The protection function operates as follows when the sequence is suspended.

- Suspension of the sequence due to the arrival of an interrupt
  If an interrupt request is accepted while the above specified sequence for writing is in progress and the interrupt service routine does not access any of the FLMD pin setting registers (for details, refer to table 6-11, List of FLMD Pin Setting Registers), the sequence is not obstructed. Writing to the protected register can proceed successfully on return from the interrupt service routine.
- Suspension of the sequence due to halting of the emulator
  If the emulator is halted because it reached a break point, for example, while the above specified sequence for writing is in progress, the sequence of writing to the register is fully suspended until the emulator returns from the halted state to normal operation.
  In other words, even access to an FLMD pin setting register (for details, refer to table 6-11, List of FLMD Pin Setting Registers) while the emulator is halted does not obstruct the sequence of writing. Furthermore, access to an FLMD pin setting register does not lead to setting of the FLMDPS.FLMDPRERR bit.

Section 7 Clock Generation

The clock generator (CG) controls the internal system clocks that are supplied to individual on-board units such as the CPU.

It also monitors the input clock to detect abnormalities and has a baud-rate generator to produce the desired output clock signal on the CLKOUT pin.

7.1 Overview of Clock Generation

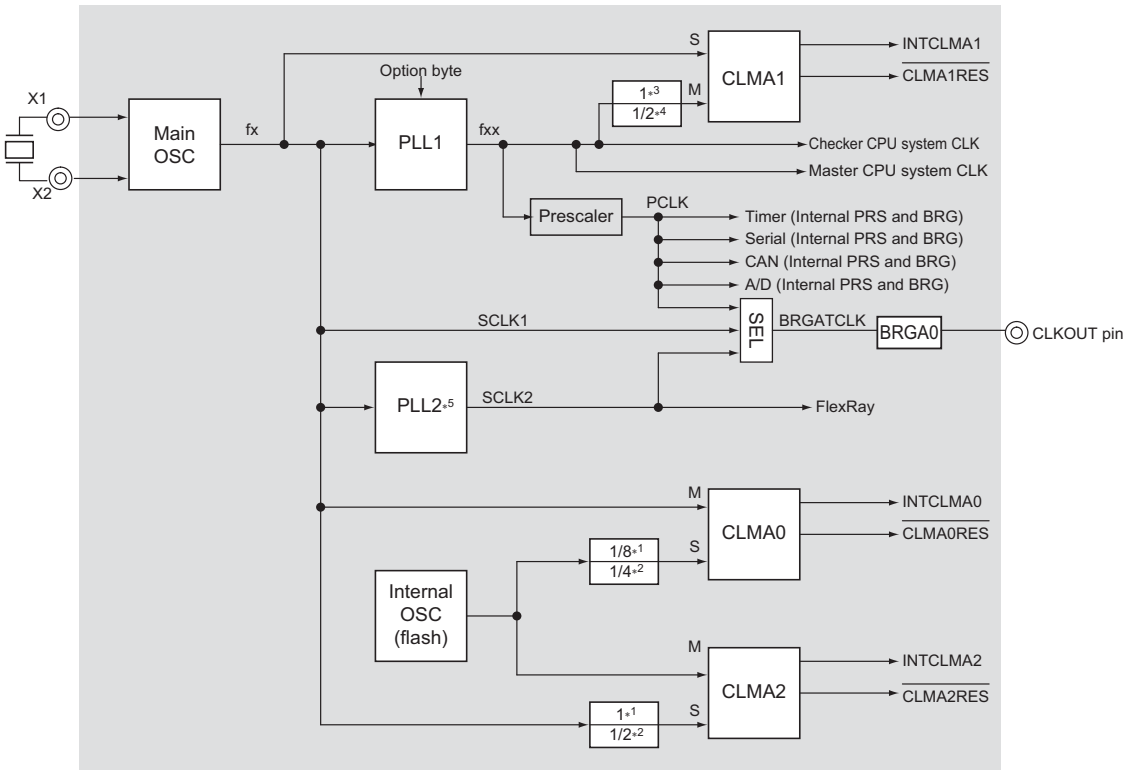
Functional overview

Clock generation involves the following controls and functions.

- Oscillator
  - External resonator: 16 MHz (µPD70F3506, µPD70F3508),
    8 or 16 MHz (µPD70F3507, µPD70F3509),
    8 MHz (µPD70F3505A)
- Clock monitoring (CLMA0 to CLMA2) blocks
  - CLMA0 monitors the main oscillator and generates reset and interrupt-request signals.
  - CLMA1 monitors the internal system clock and generates reset and interrupt-request signals.
  - CLMA2 monitors the internal oscillator and generates reset and interrupt-request signals.
- Clock output: The frequency of the signal output on the CLKOUT pin is adjustable.

Note For the output frequencies, their allowed range, and other parameters, refer to section 6.5, Option-Setting Bytes, and to the Data Sheet.

7.2 Configuration



Note 1. When the frequency of the main oscillator (OSC) is 8 MHz
Note 2. When the frequency of the main oscillator is 16 MHz
Note 3. When the frequency of HEAPCLK is other than 160 MHz
Note 4. When the frequency of HEAPCLK is 160 MHz (for the μPD70F3508 and μPD70F3509)
Note 5. Not supported by the μPD70F3505A.

Figure 7-1 Clock Generating Circuit

7.3 Selecting the Input Clock Signal

The clock generator consists of an oscillator and PLL synthesizers and is capable of, for example, generating internal system clocks at 48, 64, 80, and 160 MHz when an external resonator (crystal oscillator or ceramic oscillator) running at 8 or 16 MHz is connected to the X1 and X2 pins.

For the oscillator to be connected to the X1 and X2 pins, refer to table 7-1, Oscillators Connectable to the X1 and X2 Pins. The 8- or 16-MHz signal from the external resonator is input to PLL1, which produces a frequency-multiplied clock signal for supply that is supplied to the CPU as the operating clock and to peripheral I/O via the prescaler. Moreover, the same resonator can be input to PLL2, which generates an independent clock signal for the FlexRay interface.

Table 7-1 Oscillators Connectable to the X1 and X2 Pins

Oscillators Connectable to the X1 and X2 Pins
8 MHz, 16 MHz*

Note * The only frequency for the µPD70F3505A is 8 MHz.
The only frequency for the µPD70F3506 and µPD70F3508 is 16 MHz.

Table 7-2 Operating Frequency

Internal System Clock	Peripheral Input-Output Clock (PCLK)	Frequency of External Resonator
48/64/80/160 MHz	24/32/40/48/64/80 MHz	8/16 MHz

Note For input frequency and multiplier settings, refer to section 6.5, Option-Setting Bytes.

Caution If you intend to use FlexRay, set PCLK to a frequency no higher than 64 MHz (this is only applicable to the µPD70F3506 and µPD70F3507).

7.4 Clock Generating Circuit

The structure of the clock generating circuit is described below.

(1) Structure of Clock Generating Circuit

The circuit consists of the clock oscillator circuit, PLL1, and PLL2.

- Main oscillator circuit (Main OSC)

  The clock signal f_X produced by the main oscillator circuit provides the clock for the main systems and is input to the PLL. The oscillator circuit requires the connection of an external resonator between X1 and X2.

- PLL1

  The PLL1 circuit generates the clock signal for use in driving the microcontrollers. The clock signal output from PLL1 is divided by one or two, and then supplied to peripheral input and output.

- PLL2

  The PLL2 circuit generates a clock signal for use if the microcontroller is to conduct FlexRay operations.

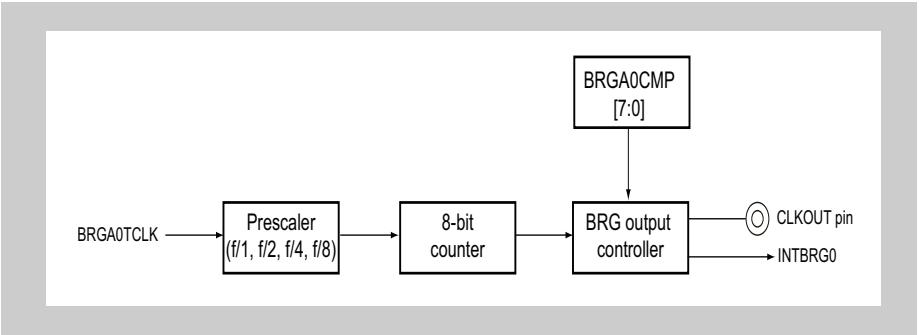
Resetting the clock generating circuit

The clock generator is initialized by a reset from any source other than a software reset. After release from the reset state, operation is restarted in accord with the setting of the option byte.

7.5 Clock Output Function (CLKOUT)

The clock-output function handles the output of a clock signal from CLKOUT pin. The signal for output from the CLKOUT pin can be frequency-divided by the baud-rate generator.

The figure below is a schematic view of the clock output function.



7.5.1 Baud-Rate Generator for CLKOUT Function (BRGA)

The baud-rate generator frequency-divides the clock signal it receives as input and can thus produce clock signals at various frequencies on the CLKOUT pin.

BRGA0 inverts the output signal on CLKOUT when the value of the baud-rate counter matches the setting of the BRGA0CMP[7:0] bits in the BRGA0CMP register. If BRGA0CTL.BRGA0ODIS = 1, the low level is output on CLKOUT.

When the counter reaches the value for comparison set in BRGA0CMP.BRGA0CMP[7:0], the BRGA0 interrupt (INTBRG0) is generated.

Note When BRGA0CMP.BRGA0CMP[7:0] = 00_H, the initial counting value is 01_H, and the sequence of counting is 02_H, 03_H, ... FE_H, FF_H, 00_H. Comparison after the overflow of the counter overflows produces a match.

Calculating the period of the signal from the baud-rate generator

- For BRGA0CMP.BRGA0CMP[7:0] = 00_H, the period of the output clock signal (CLKOUT) from the baud-rate generator is calculated from:

  CLKOUT period = counter clock cycle × 256 × 2

- For BRGA0CMP.BRGA0CMP[7:0] = N = 01_H to FF_H, the period of the output clock signal (CLKOUT) from the baud-rate generator is calculated from:

  CLKOUT period = counter clock cycle × N × 2

Clock supply The baud-rate generator BRGA0 provides the following clock input.

Table 7-3 Clock Source for BRGA0

BRGA0	Clock Source
BRGATCLK	Clock selected in the BRGCKCTL register (SCLK1, SCLK2, or PCLK)

Interrupt The BRGA interrupt is as indicated in the following table.

Table 7-4 BRGA Interrupt

BRGA Signal	Function	Connected to
INTBRG0	BRGA0 interrupt signal	Interrupt controller

Register addresses A list of addresses of the frequency output registers and clock monitor registers is given below.

Table 7-5 List of Registers in the Baud-Rate Generator for CLKOUT

Register Name	Symbol	Address
BRGA0 flag register	BRGA0FLG	FFFF FD00$_H$
BRGA0 control register	BRGA0CTL	FF83 9008$_H$
BRGA0 compare register	BRGA0CMP	FF83 900C$_H$
BRGA0 clock selection register	BRGCKCTL	FF42 0030$_H$

7.5.2 BRGA Registers for CLKOUT Function

(1) BRGA0 Clock Selection Register (BRGCKCTL)

Register BRGCKCTL is an 8-bit register that selects the clock signal for input to the baud-rate generator (BRGA0).

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

Address FF42 0030$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	BRGCKSL1	BRGCKSL0
R	R	R	R	R	R	R/W	R/W

Table 7-6 BRGCKCTL Register Contents

Bit Position	Bit Name	Function
1, 0	BRGCKSL1, BRGCKSL0	This register selects the input clock signal on the BRGATCLK signal line.

BRGCKSL1	BRGCKSL0	Input Clock
0	0	Selects SCLK1 (OSC clock)
0	1	Selects PCLK (peripheral clock)
1	0	Selects SCLK2 (FlexRay clock)
1	1	Setting prohibited

Caution 1. Write to BRGCKCTL (selection of BRGA0 operation clock) while BRGA0 is stopped (the BRGA0CE bit of the BRGA0CTL register = 0).

Caution 2. Do not use the interrupt (INTBRG0) if SCLK2 (FlexRay clock) is selected in register BRGCKCTL.

(2) BRGA0 Control Register (BRGA0CTL)

Register BRGA0CTL is an 8-bit register that enables counting by the counter, sets the prescaler period, and controls the output of BRGA0.

The BRGA0ODIS and BRGA0CCS[1:0] bits can only be modified while the BRGA0CEF bit of BRGA0FLG register is set to 0.

Refer to the BRGA0FLG register for a description of the BRGA0CEF bit.

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

Address $FF839008_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	BRGA0CE	0	BRGA0 ODIS	BRGA0 CCS1	BRGA0 CCS0
R	R	R	R/W	R	R/W	R/W	R/W

Table 7-7 BRGA0CTL Register Contents

Bit Position	Bit Name	Function
4	BRGA0CE	This bit enables or disables operation of the baud-rate counter 0: Baud-rate counter operation is disabled (INTBRG0 interrupts will not be generated). 1: Enable baud-rate counter operation (INTBRG0 interrupts will be generated).
2	BRGA0ODIS	This bit controls output from BRGA0. 0: The signal from the baud-rate generator is output on the CLKOUT pin. 1: The low level (fixed) is output on the CLKOUT pin.
1, 0	BRGA0CCS1, BRGA0CCS0	This bit selects the counter-clock cycle for the baud-rate generator.

BRGA0CCS1	BRGA0CCS0	Counter-clock cycle
0	0	BRGATCLK/1
0	1	BRGATCLK/2
1	0	BRGATCLK/4
1	1	BRGATCLK/8

Caution 1. Only write to the bits BRGA0CTL.BRGA0ODIS, BRGA0CTL.BRGA0CCS[1:0] and BRGA0CMP.BRGA0CMP[7:0] while the baud-rate counter is stopped (BRGA0FLG.BRGA0CEF = 0). Only reading is possible while the baud-rate generator is operating. The level of the CLKOUT signal and timing of INTBRG0 interrupts become undefined if settings are modified during operations.

Caution 2. To restart the baud-rate counter after it has stopped, read the BRGA0FLG.BRGA0CEF bit and confirm that the baud-rate counter is stopped (BRGA0FLG.BRGA0CEF = 0).
The level of the CLKOUT signal and timing of INTBRG0 interrupts become undefined when the setting of the BRGA0CTL.BRGA0CE bit is 0 and 1 is written to BRGA0CTL.BRGA0CE bit while the baud-rate counter is not stopped (BRGA0FLG.BRGA0CEF = 1).

Caution 3. BRGA0CTL.BRGA0CE is synchronized with BRGA0TCLK, so synchronization requires 3 clock cycles of BRGA0TCLK. For this reason, CLKOUT and the INTBRG0 interrupt are initialized (both signals are fixed to the low level) after 3 cycles of BRGA0TCLK by writing 0 to BRGA0CTL.BRGA0CE bit during operations, which stops the baud-rate counter. For the states of operation of the baud-rate counter, refer to the description of the BRGA0FLG.BRGA0CEF bit.

(3) BRGA0 Compare Register (BRGA0CMP)

BRGA0CMP register stores the baud-rate counter comparison value.

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

Address $FF83900C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
BRGA0CMP[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7-8 BRGA0CMP Register Contents

Bit Position	Bit Name	Function
7 to 0	BRGA0CMP [7:0]	Value for comparison

(4) BRGA0 Flag Register (BRGA0FLG)

BRGA0FLG register indicates the current operation status of the baud-rate counter.

Access This register can be read in 8-bit units.

Address FFFF FD00$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	BRGA0 CEF
R	R	R	R	R	R	R	R

Table 7-9 BRGA0FLG Register Contents

Bit Position	Bit Name	Function
0	BRGA0CEF	This bit indicates the status of the baud-rate counter. 0: Baud-rate counter operation is stopped 1: Baud-rate counter is in operation

7.6 Clock Monitor A (CLMAn) Function

This product has three instances of the clock monitor A.

7.6.1 Features of the Clock Monitors (CLMAn; n = 0, 1, 2)

Table 7-10 Instances

Clock Monitor A	
Number of instances	3
Name	CLMAn

Instances index n Throughout this section, the individual instances of a clock monitor A are identified by the index "n" (n = 0 to 2), for example, CLMAnCTL0 for the control register 0 of CLMAn.

<CLMAn_base> The base address <CLMAn_base> of each CLMAn is listed in the table below.

Table 7-11 Register Base Address <CLMAn_base>

CLMAn	<CLMAn_base> Address
CLMA0	$FF80_2000_H$
CLMA1	$FF80_3000_H$
CLMA2	$FF80_4000_H$

Clock supply The monitored and the sampling clocks of all clock monitors A are listed in the following table:

Table 7-12 CLMAn Clock Supply

Clocks for CLMAn	Function	Connected to
CLMA0:		
CLMA0TSMP	Sampling clock for CLMA0	Internal OSC
CLMA0TMON	Clock for monitoring by CLMA0	Main OSC
CLMA1:		
CLMA1TSMP	Sampling clock for CLMA1	Main OSC
CLMA1TMON	Clock for monitoring by CLMA1	Internal system clock
CLMA2:		
CLMA2TSMP	Sampling clock for CLMA2	Main OSC
CLMA2TMON	Clock for monitoring by CLMA2	Internal OSC

Interrupts and reset outputs

The interrupts and reset outputs of the CLMAn are listed in the table below.

Table 7-13 CLMAn Interrupts and Reset Outputs

CLMAn Signal	Function	Connected to
CLMA0:		
$\overline{\text{CLMA0RES}}$	CLMA0 error reset	Reset controller $\overline{\text{CLMA0RES}}$
CLMA0TI	CLMA0 error interrupt request	Interrupt controller INTCLMA0
CLMA1:		
$\overline{\text{CLMA1RES}}$	CLMA1 error reset	Reset controller $\overline{\text{CLMA1RES}}$
CLMA1TI	CLMA1 error interrupt request	Interrupt controller INTCLMA1
CLMA2:		
$\overline{\text{CLMA2RES}}$	CLMA2 error reset	Reset controller $\overline{\text{CLMA2RES}}$
CLMA2TI	CLMA2 error interrupt request	Interrupt controller INTCLMA2

Output signal

CLMAn output signals are listed in the following table.

Table 7-14 Output Signals from CLMAn

CLMAn Signal	Function	Connected to
CLMA0:		
CLMA0TERR	Error output signal from CLMA0	Safety guardian
CLMA0RES	Error reset signal from CLMA0	
CLMA1:		
CLMA1TERR	Error output signal from CLMA1	Safety guardian
CLMA1RES	Error reset signal from CLMA1	
CLMA2:		
CLMA2TERR	Error output signal from CLMA2	Safety guardian
CLMA2RES	Error reset signal from CLMA2	

7.6.2 CLMA Enable and Start-Up Options

(1) CLMA Enable

Monitoring of clock signals by a clock monitor is set up by setting the CLMAnCTL1, CLMAnCMPL, and CLMAnCMPH registers, and starts when the CLMAnCLME bit in the CLMAnCTL0 register is set to 1.

7.6.3 Functional Overview

The clock monitor CLMAn indicates an abnormal frequency of the monitored clock.

Features summary

The clock monitor has the following features:

- monitoring of the frequency of an input clock CLMAnTMON by using a sampling clock CLMAnTSMP
- indication of abnormal clock frequencies by the following means:
  - output of a reset request signal, or
  - output of an error signal in combination with the generation of an error interrupt request

The following figure shows the main components of the clock monitor.

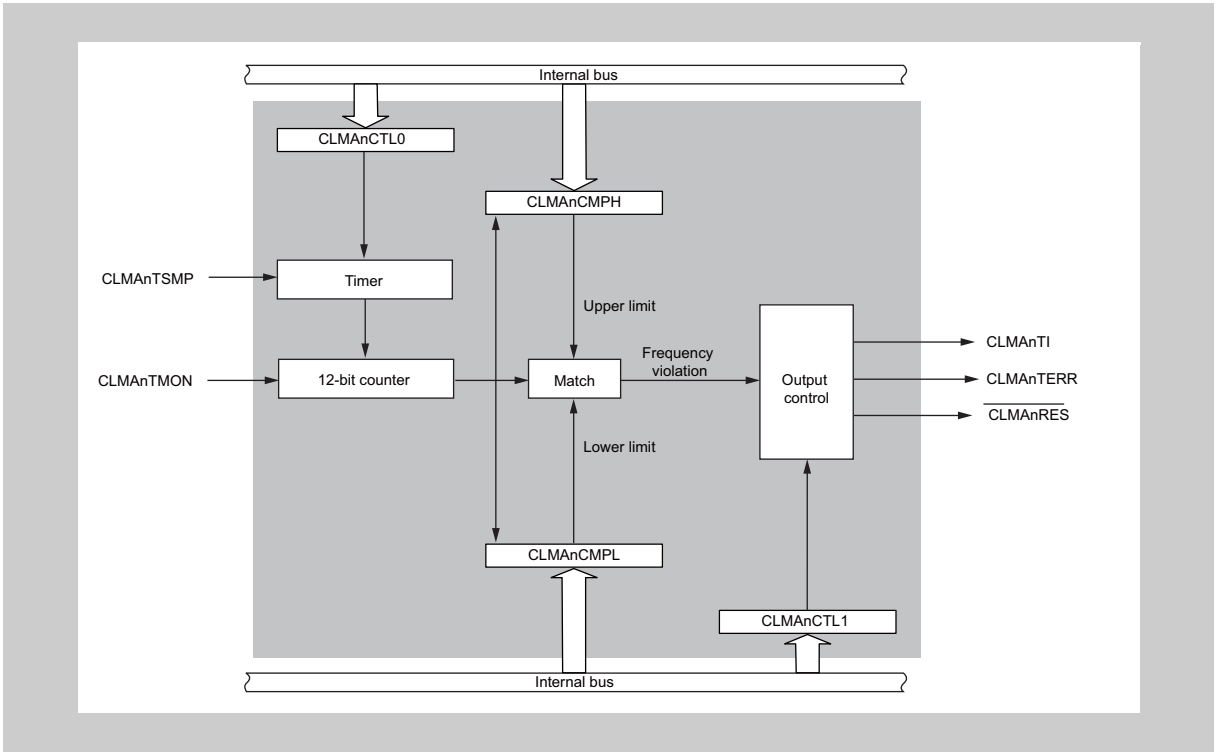


Figure 7-2 Block Diagram of the Clock Monitor A

7.6.4 Functional Description

The clock monitor CLMAn is used to ensure that the frequency of a clock (CLMAnTMON) stays between certain limits.

(1) Detection of Abnormal Clock Frequencies

Method

1. CLMAn counts the rising edges of the monitored clock CLMAnTMON within 16 cycles of the sampling clock CLMAnTSMP and then compares the counter with the configured thresholds:
   - CLMAnCMPL.CLMAnCMPL[11:0] defines the lower threshold.
   - CLMAnCMPH.CLMAnCMPH[11:0] defines the upper threshold.
2. When CLMAnTMON stops or its frequency is too low, the counter falls below CLMAnCMPL.CLMAnCMPL[11:0].
3. When the frequency of CLMAnTMON is too high, the counter exceeds CLMAnCMPH.CLMAnCMPH[11:0].

In both cases, CLMAn indicates an abnormal clock frequency as described in 7.6.4, (2) Indication of Abnormal Clock Frequency.

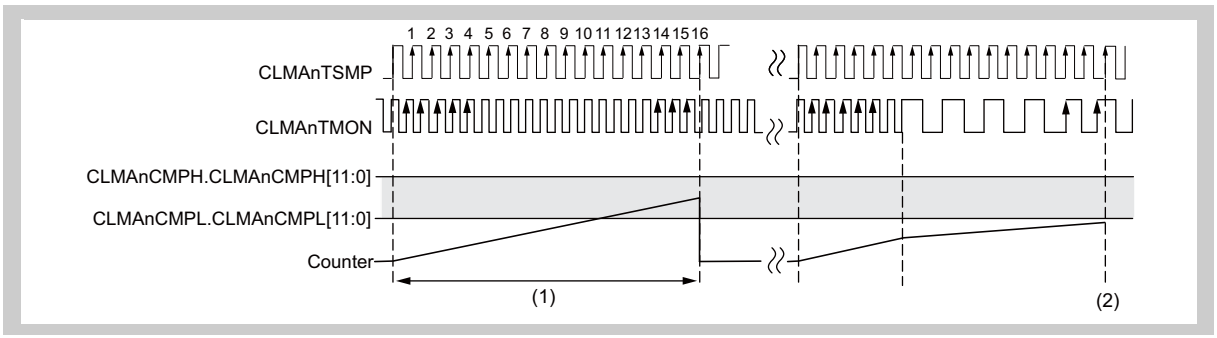


Figure 7-3 Example: $f_{CLMAnTMON}$ is low

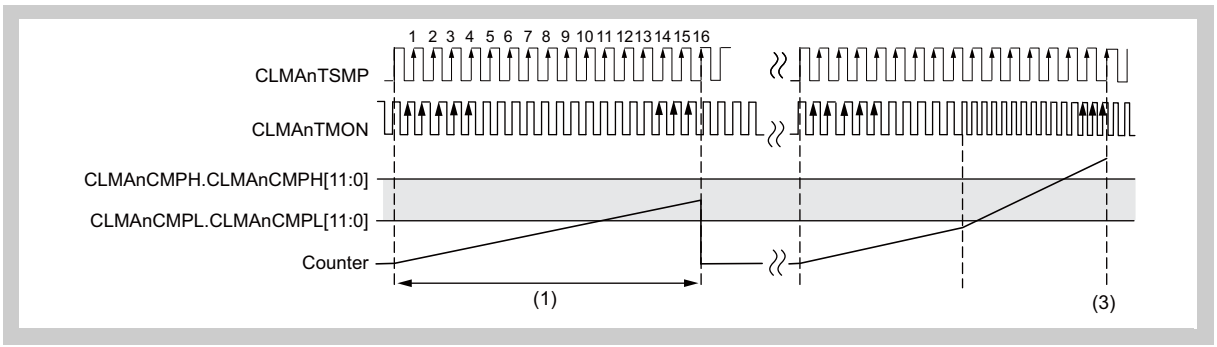


Figure 7-4 Example: $f_{CLMAnTMON}$ is high

Note When $f_{CLMAnTMON}$ changes within the sampling interval (16 cycles of the CLMAnTSMP clock), the counter might be within the valid range even though $f_{CLMAnTMON}$ has actually become too high or low.

The abnormal $f_{CLMAnTMON}$ is detected one sampling interval later.

(a) Calculating the Thresholds of CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]

The compare registers CLMAnCMPL and CLMAnCMPH are configured with the minimum and maximum number of clock cycles of CLMAnTMON that are assumed to be valid within 16 cycles of the sampling clock CLMAnTSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMAnTSMP}} = \frac{N}{f_{CLMAnTMON}}$$

$$N = \frac{f_{CLMAnTMON}}{f_{CLMAnTSMP}} \times 16$$

Considering the allowed frequency deviations of CLMAnTMON and CLMAnTSMP, the threshold values can be calculated by the following formulas:

$$\text{Lower threshold} = N_{min} = \frac{f_{CLMAnTMON(min)}}{f_{CLMAnTSMP(max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{max} = \frac{f_{CLMAnTMON(max)}}{f_{CLMAnTSMP(min)}} \times 16 + 1$$

Note For examples of CLMAnCMPH and CLMAnCMPL registers of this product, refer to table 7-16, Examples of CLMAnCMPH and CLMAnCMPL Register Settings.

Minimum thresholds

The following restrictions must be taken into account:

- CLMAnCMPL $\geq 0001_H$
- CLMAnCMPH $\geq$ CLMAnCMPL + 0003_H

(2) Indication of Abnormal Clock Frequency

$f_{CLMAnTMON}$ too high

$f_{CLMAnTMON}$ exceeding the upper threshold indicates clock abnormalities as follows:

1. The reset request signal $\overline{CLMAnRES}$ (active low) is being output.
2. The peripheral reset signal PERRES is being generated and is resetting CLMAn.

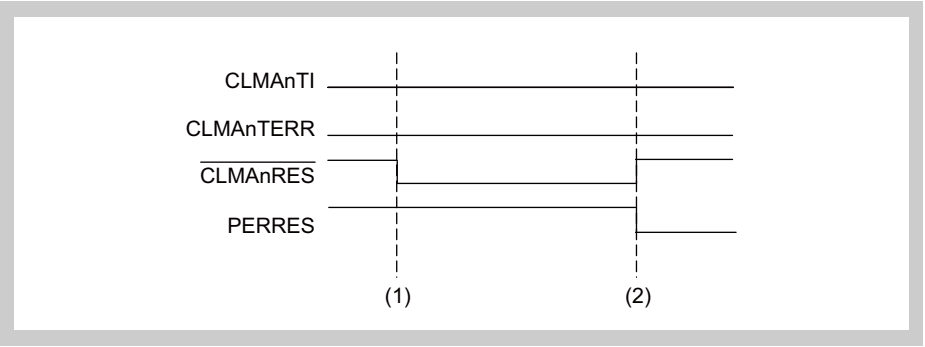


Figure 7-5 Indication of Clock Abnormality when $f_{CLMAnTMON}$ Rises above the Upper Threshold

$f_{CLMAnTMON}$ too low

The response by CLMAn to $f_{CLMAnTMON}$ falling below the lower threshold is described by the entry in the table that corresponds to the current setting of CLMAnCTL1.CLMAnOSEL

Table 7-15 When $f_{CLMAnTMON}$ is too Low

CLMAnCTL1. CLMAnOSEL	Indication of Clock Abnormality when $f_{CLMAnTMON}$ Falls below the Upper Threshold
0	CLMAn outputs a reset-request signal ($\overline{CLMAnRES}$: active low). This is the same response as when $f_{CLMAnTMON}$ is too high; see the description above.
1	CLMAn outputs an error signal (CLMAnTERR: active high) and generates an error-interrupt request (CLMAnTI). • The error signal (CLMAnTERR) is set to the high level and remains there until CLMAn is reset (PERRES). • The error-interrupt request (CLMATI) is only generated once, even if the clock frequency remains too low.

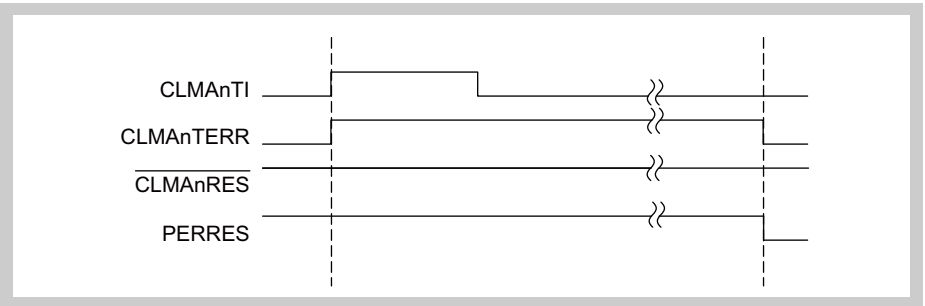


Figure 7-6 Indication of Clock Abnormality when $f_{CLMAnTMON}$ Falls below the Lower Threshold (when CLMAnCTL1.CLMAnOSEL = 1)

(3) Enabling CLMAn (Writing to the CLMAnCTL0 Register)

The control register (CLMAnCTL0) is the write-protection register and is used to enable CLMAn.

Note Once CLMAn is enabled, it cannot be disabled by software; it can only be disabled by a reset.

(a) Initial Value of the CLMAnCTL0 Register

The initial value of the CLMAnCTL0 register is 00_H. Using software to set the CLMAnCLME bit to 1 initiates CLMAn operation.

(b) Procedure for Writing to Enable CLMAn

Execute instructions in the sequence below to set the CLMAnCTL0 register to 01_H.

1. Write $A5_H$ to the CLMAnPCMD register.
2. The following sequence is required for writing to the CLMAnCTL0 register.
   - Write 01_H to enable CLMAn.
   - Write the inverse of that value, i.e. FE_H.
   - Write the intended value (01_H) again.
3. Read the CLMAnCTL0 register.

   If the value of the register is 01_H, CLMAn is enabled.
   If this was not the value, check the settings in the CLMAnCTL0 register and in the write operation status register (CLMAnPS).
   - If the value in the CLMAnPS register is 01_H, the instruction sequence was not executed correctly. Restart the sequence for enabling CLMAn from step 1.
   - If the value in the CLMAnPS register is 00_H, write 00_H to the CLMAnPCMD register and only then re-execute the sequence from step 1.

(4) Examples of CLMAnCMPH and CLMAnCMPL Register Settings

Table 7-16 Examples of CLMAnCMPH and CLMAnCMPL Register Settings

	Operating Frequency	Main OSC	Clock to be Monitored		Sampling Clock		CLMAn CMPH	CLMAn CMPL
				Frequency		Frequency		
CLMA0	48 MHz 64 MHz 80 MHz 160 MHz	8 MHz	Main OSC	8 MHz	Internal OSC/8	2 MHz	$004E_H$	0034_H
		16 MHz	Main OSC	16 MHz	Internal OSC/4	4 MHz	$004E_H$	0034_H
CLMA1	48 MHz	16 MHz	Internal system clock	48 MHz	Main OSC	16 MHz	0032_H	$002E_H$
	64 MHz	8 MHz	Internal system clock	64 MHz	Main OSC	8 MHz	0084_H	$007C_H$
		16 MHz				16 MHz	0043_H	$003D_H$
	80 MHz	8 MHz	Internal system clock	80 MHz	Main OSC	8 MHz	$00A5_H$	$009B_H$
		16 MHz				16 MHz	0053_H	$004D_H$
	160 MHz	8 MHz	Internal system clock/2	80 MHz	Main OSC	8 MHz	$00A5_H$	$009B_H$
		16 MHz				16 MHz	0053_H	$004D_H$
CLMA2	48 MHz 64 MHz 80 MHz 160 MHz	8 MHz	Internal OSC	16 MHz	Main OSC	8 MHz	0028_H	0019_H
		16 MHz	Internal OSC	16 MHz	Main OSC/2	8 MHz	0028_H	0019_H

(5) Behavior on a Reset Due to CLMAn Detecting an Error ($\overline{\text{CLMAnRES}}$)

A reset is generated and conveyed to the SGA to notify it of the error.

If the states of the main oscillator, PLL1 and the internal oscillator become normal at that time, the chip is released from the reset state after the flash reset sequence is finished, and the OSC stabilization time and PLL lock-up time have elapsed.

If any one of main oscillator, PLL1 and the internal oscillator is stopped or in an abnormal state, the chip will not be released from the reset state.

(6) Behavior after a CLMAn Error-Interrupt Request (CLMAnTI)

(a) After a CLMA0 Error Interrupt Request (CLMA0TI)

The error signal is conveyed to the SGA, after which only the CPU and SGA continue to operate with the clock signal generated at the corresponding frequency by the free-running PLL.

Stop operations by peripheral functions, including DMA transfer, since the results cannot be guaranteed (except in the reading and writing of registers). Once abnormal oscillation is detected, a subsequent return to normal oscillation is not detectable.

(b) After a CLMA1 Error Interrupt Request (CLMA1TI)

The error signal is conveyed to the SGA. The CPU will not operate if its system clock is stopped. Operation of the CPU and peripheral functions in this case is not guaranteed.

(c) After a CLMA2 Error Interrupt Request (CLMA2TI)

The error signal is conveyed to the SGA. If the internal oscillator is stopped, even if oscillation of the main oscillator is abnormal, the problem will not be detectable by CLM0 or CLM1 so the operation of the CPU and peripheral functions cannot be guaranteed. If the internal OSC is stopped at that time even after the input of a reset, the flash reset sequence will not proceed, and the chip will not be released from the reset state.

7.6.5 Clock Monitor A Registers

Clock monitor A is controlled and operated by the following registers.

Register addresses

The list of register addresses of clock monitor A is given below.

Table 7-17 List of Clock Monitor Registers

Register Name	Symbol	Address
CLMAn control register 0	CLMAnCTL0	<CLMAn_base> + 00_H
CLMAn control register 1	CLMAnCTL1	<CLMAn_base> + 04_H
CLMAn compare register L	CLMAnCMPL	<CLMAn_base> + 08_H
CLMAn compare register H	CLMAnCMPH	<CLMAn_base> + $0C_H$
CLMAn protection command register	CLMAnPCMD	<CLMAn_base> + 10_H
CLMAn protection status register	CLMAnPS	<CLMAn_base> + 14_H

<CLMAn_base>

The base addresses <CLMAn_base> of the CLMAn are defined in table 7-11, Register Base Address <CLMAn_base>.

(1) CLMAnCTL0 – CLMAn Control Register 0

This register is used to enable the clock monitor A (CLMAn).

Access

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

Writing to this register is protected by a special sequence of instructions. For details, refer to section 7.6.4, (3) Enabling CLMAn (Writing to the CLMAnCTL0 Register).

Address

<CLMAn_base> + 00_H

Initial value

00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CLMAn CLME
R	R	R	R	R	R	R	R/W

Table 7-18 CLMAnCTL0 Register Contents

Bit Position	Bit Name	Function
0	CLMAnCLME	Enables/disables the clock monitor: 0: Disable CLMAn 1: Enable CLMAn This bit can only be cleared by a reset.

(2) CLMAnCTL1 – CLMAn Control Register 1

This register specifies the signal to be output when the frequency of the clock signal being monitored (CLMAnTMON) is beyond the specified range.

For details, refer to section 7.6.4, (2) Indication of Abnormal Clock Frequency.

Access This register can be read/written in 1- or 8-bit units. It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

Address <CLMAn_base> + 04_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CLMAn OSEL
R	R	R	R	R	R	R	R/W

Table 7-19 CLMAnCTL1 Register Contents

Bit Position	Bit Name	Function
0	CLMAnOSEL	Specifies the signal(s) that are output when the frequency of CLMAnTMON falls below the lower threshold: 0: Reset request signal $\overline{\text{CLMAnRES}}$ 1: Error interrupt request CLMAnTI and error signal $\overline{\text{CLMAnTERR}}$

Caution When the internal oscillator is stopped, an internal reset is not generated (this only applies to the µPD70F3508 and µPD70F3509).
Even when CLMA2CTL1.CLMA2OSEL is set to 0 (reset request signal), the SGA detects stopping of the internal oscillator, and the low level is output on the ERROROUT pin.

(3) CLMAnCMPH – CLMAn Compare Register H

This register specifies the upper frequency threshold.

For details, refer to section 7.6.4, (1)-(a), Calculating the Thresholds of CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0].

Access This register can be read/written in 16-bit units. It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

Address <CLMAn_base> + $0C_H$

Initial value $03FF_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CLMAnCMPH[11:0]											
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7-20 CLMAnCMPH Register Contents

Bit Position	Bit Name	Function
11 to 0	CLMAnCMPH[11:0]	Specifies the upper threshold. • The recommended value: Refer to table 7-16, Examples of CLMAnCMPH and CLMAnCMPL Register Settings. • The minimum value: CLMAnCMPL + 0003_H

(4) CLMAnCMPL – CLMAn Compare Register L

This register specifies the lower frequency threshold.

For details, refer to section 7.6.4, (1)-(a), Calculating the Thresholds of CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0].

Access This register can be read/written in 16-bit units. It can be written only when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

Address <CLMAn_base> + 08_H

Initial value 0001_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CLMAnCMPL[11:0]											
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7-21 CLMAnCMPL Register Contents

Bit Position	Bit Name	Function
11 to 0	CLMAnCMPL[11:0]	Specifies the lower threshold. • The recommended value: Refer to table 7-16, Examples of CLMAnCMPH and CLMAnCMPL Register Settings. • The minimum value: 0001_H

(5) CLMAnPCMD – CLMAn Protection Command Register

This is the protection command register for CLMAnCTL0.

For details, refer to section 7.6.4, (3) Enabling CLMAn (Writing to the CLMAnCTL0 Register).

Access This register can be written in 8-bit units.

Address <CLMAn_base> + 10_H

Initial value Undefined

7	6	5	4	3	2	1	0
CLMAnREG[7:0]							
W	W	W	W	W	W	W	W

Table 7-22 CLMAnPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMAnREG[7:0]	Protection command to enable writing to CLMAnCTL0 register.

(6) CLMAnPS – CLMAn Protection Status Register

This register verifies whether or not writing to the write protection register (CLMAnCTL0) was successful.

For details, refer to section 7.6.4, (3) Enabling CLMAn (Writing to the CLMAnCTL0 Register).

Access This register can be read in 8-bit units.

Address <CLMAn_base> + 14_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CLMAn PRERR
R	R	R	R	R	R	R	R

Table 7-23 CLMAnPS Register Contents

Bit Position	Bit Name	Function
0	CLMAnPRERR	Indicates if writing to the write protection register CLMAnCTL0 register was successful. 0: Writing succeeded. 1. Writing failed.

Section 8 Reset Controller

This section describes an overview of the reset controller.

8.1 Functional Overview

This product incorporates a number of system reset functions.

Features summary

A reset can be caused by the following events.

- External reset signal $\overline{\text{RESET}}$

  Noise in the external reset signal is eliminated by an analog filter.
- Overflow of the watchdog timer ($\overline{\text{WDTA0RES}}$)
- Clock monitors reset ($\overline{\text{CLMA0RES}}$ to $\overline{\text{CLMA2RES}}$)
- Low voltage indicator reset ($\overline{\text{LVIRES}}$)
- Software reset (SWRES)
- Safety guardian reset ($\overline{\text{SGARES}}$)
- Self-diagnostic BIST reset ($\overline{\text{BISTRES}}$)
- Debug reset ($\overline{\text{DBRES}}$)

The following block diagram shows the main components of the reset controller.

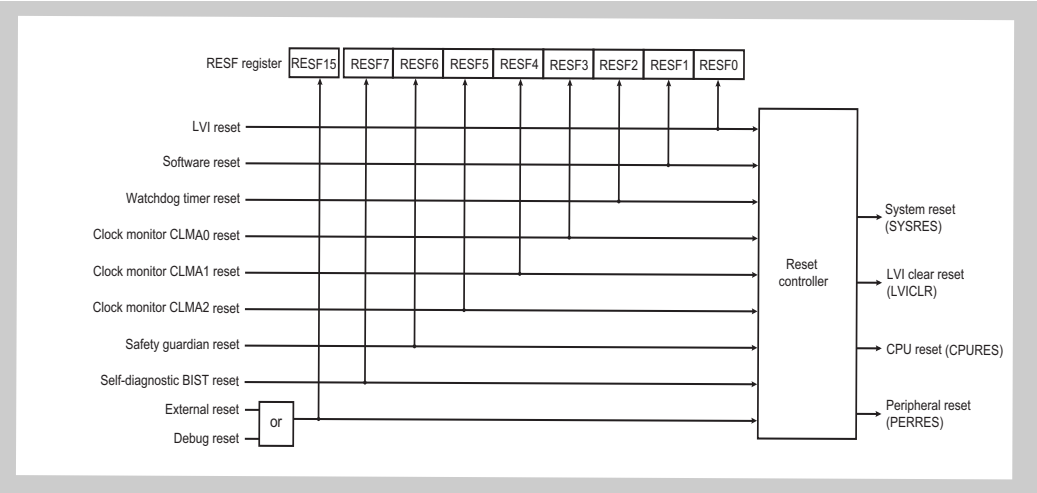


Figure 8-1 Block Diagram of the Reset Controller

Table 8-1 Reset Sources and Timing of Operations

Reset Input Source	Internal Reset Signals*1				Flash Reset Sequence Time*2	PLL Lock-up Time*2	OSC Stabilization Time*2	Self-Diagnostic BIST Execution*3	RESF Register
	SYSRES	LVICLR	CPURES	PERRES					
$\overline{\text{RESET}}$	√	√	√	√	√	√	–	√	RESF15*4
$\overline{\text{DBRES}}$	√	√	√	√	√	√	–	√	RESF15*4
$\overline{\text{BISTRES}}$	–	√	√	√	√	–*5	–	–	RESF7
$\overline{\text{SGARES}}$	–	√	√	√	√	√	–	√	RESF6
$\overline{\text{CLMA2RES}}$	–	√	√	√	√	√	√	√	RESF5
$\overline{\text{CLMA1RES}}$	–	√	√	√	√	√	√	√	RESF4
$\overline{\text{CLMA0RES}}$	–	√	√	√	√	√	√	√	RESF3
$\overline{\text{WDTA0RES}}$	–	√	√	√	√	√	–	√	RESF2
SWRES	–	√	√	√	–	–	–	–	RESF1
$\overline{\text{LVIRES}}$	–	–	√	√	√	√	√	√	RESF0

Note 1. (-) The signal is not available as an internal reset signal. (√) The signal is available as an internal reset signal.

Note 2. (-) The time is not considered. (√) The time is considered by the internal hardware counter.

Note 3. (-) Self-diagnostic BIST is not executed on release from the reset state.
(√) Self-diagnostic BIST is executed on release from the reset state.
Self-diagnostic BIST is not executed for the µPD70F3505A when skipping of self-diagnostic BIST is selected. For details, see section 9.4.1, Self-Diagnostic BIST Skip Function (Only for the µPD70F3505A).

Note 4. All other RESF flags are cleared by SYSRES ($\overline{\text{RESET}}$/$\overline{\text{DBRES}}$).

Note 5. The time is considered with the µPD70F3508 and µPD70F3509.

(1) Internal Reset Signals

The reset controller manages the generation of all internal reset signals upon occurrence of reset requests from various reset sources.

- System reset SYSRES
  The system reset is generated by an external reset or the debug reset. SYSRES is applied to all microcontroller components.
- CPU reset CPURES
  The CPU reset is generated by all reset sources. CPURES is applied to the master and checker CPU sub-systems (such as DMA and INTC).
- LVI clear reset LVICLR
  The LVI clear reset is generated by all reset sources except the $\overline{\text{LVIRES}}$.
- Peripheral reset PERRES
  The peripheral reset is generated by all reset sources.

Caution With the µPD70F3508 and µPD70F3509, an internal reset is not generated if the internal oscillator is stopped. Accordingly, to detect stopping of the internal oscillator, monitoring of the ERROROUT pin by an external circuit is recommended.

- Enable monitoring of the internal oscillator by CLMA2 (CLMA2CTL0.CLMA2CLME = 1).
- Enable CLMA2 source by SGA (SGAEMK0.SGAEMK002 = [initial value]).

8.2 Functional Description

8.2.1 Reset Flags

The reset flag register RESF provides reset flags for each reset source.

If a reset has occurred, the assigned flag is set. This way the source of the reset can be evaluated.

All flags in RESF are only cleared by SYSRES or by software clear request via the RESFC register. Thus, the operation of these registers is cumulative: each reset source sets its own flag, independent of all others.

8.2.2 Low-Voltage Indicator (LVI)

The low-voltage indicator circuit (LVI) permanently compares the power supply voltage EVDD with the LVI internal reference voltage V_{LVI}.

If EVDD falls below the internal reference voltage (EVDD < V_{LVI}), the internal reset signal $\overline{\text{LVIRES}}$ or the interrupt signal INTLVI is generated.

Additionally, the $\overline{\text{LVIRES}}$ flag RESF.RESF0 is set.
RESF.RESF0 is not automatically cleared, if EVDD exceeds V_{LVI}. It is cleared by

- setting RESFC.RESFC0 = 1
- SYSRES

LVI reference voltage

For the specification of the internal reference voltage (V_{LVI}), see Section 28 Electrical Characteristics.

Generation of $\overline{\text{LVIRES}}$

The generation of an interrupt instead of $\overline{\text{LVIRES}}$ is selectable.

The following figure shows the timing of $\overline{\text{LVIRES}}$ and changes to RESF.RESF0.

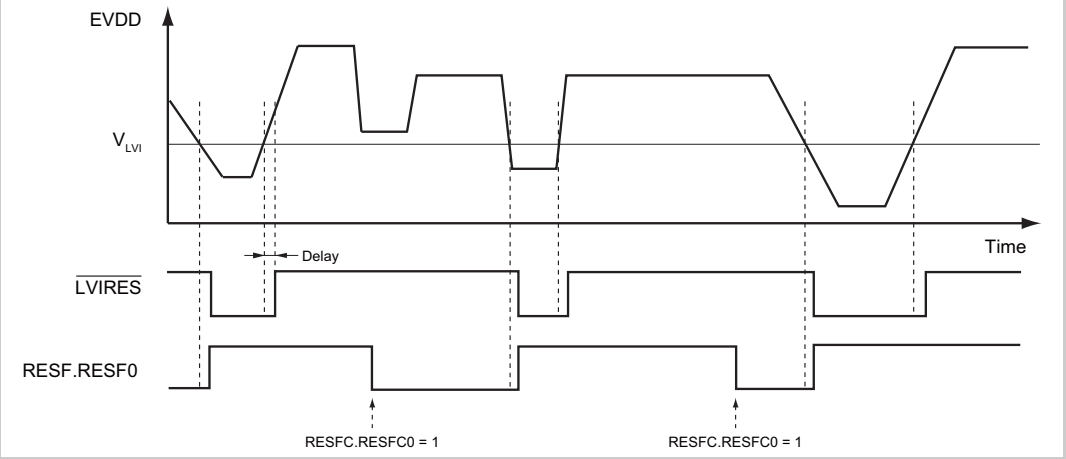


Figure 8-2 LVI Reset Timing

Delay

A delay time is induced between EVDD crossing the V_{LVI} level and assertion of $\overline{\text{LVIRES}}$ setting of RESF.RESF0.

8.2.3 External $\overline{\text{RESET}}$

A system reset is performed when a low level signal is applied to the $\overline{\text{RESET}}$ pin.

SYSRES clears the RESF register to 8000_H. The RESF.RESF15 bit represents the external $\overline{\text{RESET}}$ event.
The RESF.RESF15 bit is not automatically cleared. It is cleared by

- setting RESFC.RESFC15 = 1

The $\overline{\text{RESET}}$ signal is passed through an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when an external reset is performed. It explains the effect of the noise eliminator.

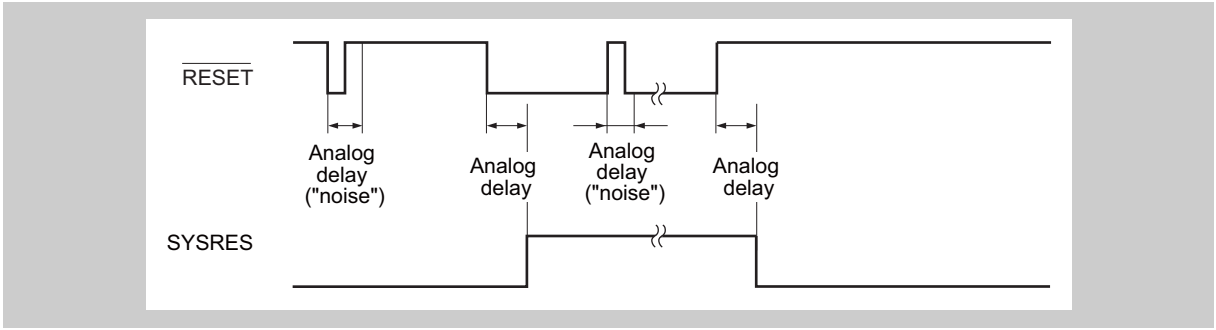


Figure 8-3 External $\overline{\text{RESET}}$ Timing

The analog delay is caused by the analog filter. The filter regards pulses up to a certain width as noise and suppresses them.
For the noise-cancelling intervals, see table 2-56, List of Noise-Cancelling Intervals and Sampling Clock for Noise Cancellation, and for the minimum $\overline{\text{RESET}}$ pulse width, see Section 28 Electrical Characteristics.

8.2.4 Watchdog Timer Reset

The watchdog timer can be configured to generate a reset if the watchdog time expires. After a watchdog reset, RESF.RESF2 for the watchdog timer reset flag (WDTA0RES) is set. RESF.RESF2 is not automatically cleared. It is cleared by

- setting RESFC.RESFC2 = 1
- SYSRES

8.2.5 Software Reset

The software reset SWRES can be asserted by setting SWRESA.SWRESA = 1.

RESF.RESF1 is not automatically cleared. It is cleared by

- setting RESFC.RESFC1 = 1
- SYSRES

8.2.6 Clock Monitor Reset

The clock monitors supervise the various internal clock signals. On detecting deviation of a clock signal from the expected range, the corresponding monitor generates a reset.

- $\overline{\text{CLMA0RES}}$: A failure in the main oscillation circuit is being detected.
- $\overline{\text{CLMA1RES}}$: A failure in PLL1 is being detected.
- $\overline{\text{CLMA2RES}}$: A failure in the internal oscillator is being detected.

Upon a clock monitor reset, the respective reset flags in the RESF register are set.
These flags are not cleared automatically. They are cleared by

- setting RESFC.RESFC3 = 1 for $\overline{\text{CLMA0RES}}$, RESFC.RESFC4 = 1 for $\overline{\text{CLMA1RES}}$, and RESFC.RESFC5 = 1 for $\overline{\text{CLMA2RES}}$, respectively.
- SYSRES

8.2.7 Self-Diagnostic BIST Reset

Self-diagnostic BIST is usually executed after release from any reset source (except SWRES). This kind of reset is generated after self-diagnostic BIST to initialize the circuits that are the target of self-diagnostic BIST.

The generation of BISTRES is indicated by RESF.RESF7.

RESF.RESF7 is not automatically cleared. It is cleared by

- setting RESFC.RESFC7 = 1
- SYSRES

8.2.8 Safety Guardian Reset

The safety guardian (SGA) collects all internal error signals from different monitoring units.

All error inputs connected to the SGA are able to generate the $\overline{\text{SGARES}}$ reset signal.

After a safety guardian reset, the safety guardian reset flag RESF.RESF6 is set.

RESF.RESF6 is not automatically cleared. It is cleared by

- setting RESFC.RESFC6 = 1
- SYSRES

8.2.9 Reset Flag Evaluation

After any CPU reset (CPURES), the software starts at the reset vector which is always located at address 0000 0000_H.

Because each reset source will set its respective reset flag, if the reset flag register RESF is equal to 0000_H, a faulty software is assumed.

The reset flag register (RESF) can be evaluated as described below.

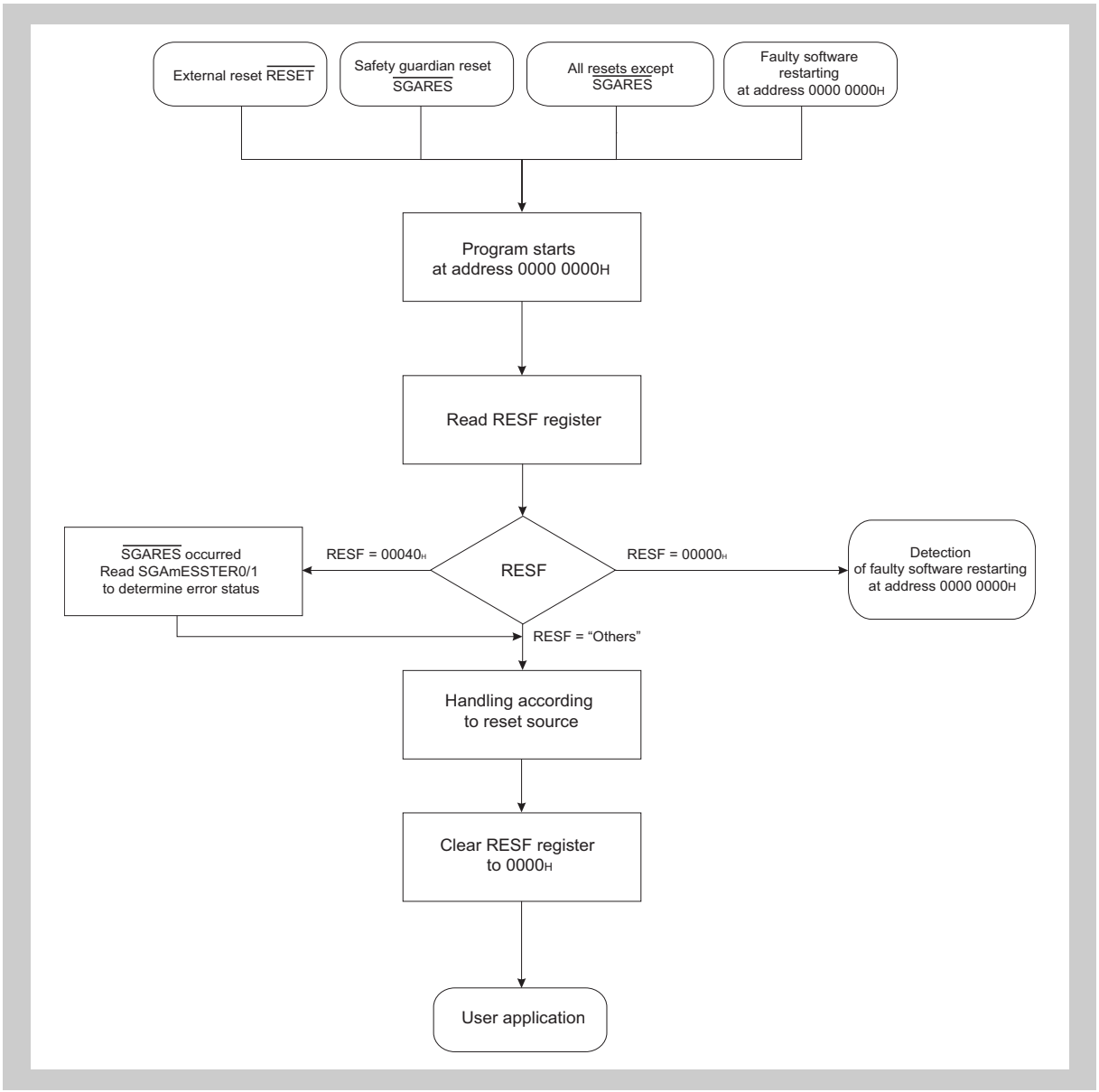


Figure 8-4 Procedure for Evaluating and Handling the Reset Flag

8.2.10 Protection for Registers of the Reset Controller

The write-protected registers are protected from inadvertent write access due to the execution of erroneous program code, etc.

Write access to write-protected registers is only possible within a special sequence of instructions shown below.

1. Write the fixed value $A5_H$ to the protection command register CSCPCMD.
2. Write the desired value to the protected register.
3. Write the bit-wise inversion of desired value to the protected register.
4. Write the desired value to the protected register.
5. Verify successful write of the desired value to the protected register by checking that CSCPS.CSCPRERR = 0.

In case of any access to the reset-related registers between steps 1 to 4 of the above sequence of instructions, the write to the protected register fails (indicated by CSCPS.CSCPRERR = 1) and the entire sequence of instructions has to be restarted from step 1.

Within the special sequence of instructions described above, it is allowed to access other registers, except the reset-related registers, without disrupting the protection sequence.

In case the protection sequence is interrupted, the protection mechanism operates as follows.

- Interrupts during protection sequence
  If an interrupt is acknowledged within the above protection sequence and the interrupt service routine does not access any reset-related registers, the protection sequence is not disrupted. The write to the protected register can be successfully completed after returning from the interrupt service routine.

8.3 Registers

This section contains a description of all registers of the reset controller.

8.3.1 Overview of Reset Controller Registers

The reset controller is controlled and operated by the following registers.

Table 8-2 List of the Reset Controller Registers

Register Name	Symbol	Address
Reset flag registers		
Reset source register	RESF	FF42 0020$_H$
Reset source set register	RESFS	FF42 0024$_H$
Reset source clear register	RESFC	FF42 0028$_H$
Software reset control register		
Software reset register	SWRESA	FF42 002C$_H$
Protection command registers		
Protection command register	CSCPCMD	FF42 0014$_H$
Protection status register	CSCPS	FF42 0018$_H$

Caution The respective registers include bits to which no function is allocated. Do not write values other than 0 to these bits, because operation cannot be guaranteed if this is done.

8.3.2 Details of Reset Controller Registers

(1) RESF - Reset Source Register

This register contains information about which type of resets occurred since the last SYSRES (RESET/DBRES).

Each following reset condition sets the corresponding flag in the register. For example, if a clock monitor CLMA0RES occurs after a watchdog timer reset WDTA0RES, this register reads $000C_H$.

Access This register can be read in 16-bit units.

Address $FF42\ 0020_H$

Initial value 8000_H

This register is initialized by SYSRES (RESET/DBRES).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESF15	0	0	0	0	0	0	0	RESF7	RESF6	RESF5	RESF4	RESF3	RESF2	RESF1	RESF0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8-3 RESF Register Contents

Bit Position	Bit Name	Function
15	RESF15	External reset flag 0: The RESET/DBRES reset is not being generated 1: The RESET/DBRES reset is being generated
7	RESF7	Self-diagnostic BIST reset flag 0: The BISTRES reset is not being generated 1: The BISTRES reset is being generated This flag will always be set after the execution of self-diagnostic BIST. The state of the flag is retained as long as self-diagnostic BIST is not executed again.
6	RESF6	Safety guardian reset flag 0: The SGARES reset is not being generated 1: The SGARES reset is being generated
5	RESF5	Clock monitor CLMA2 reset flag 0: The CLMA2RES reset is not being generated 1: The CLMA2RES reset is being generated
4	RESF4	Clock monitor CLMA1 reset flag 0: The CLMA1RES reset is not being generated 1: The CLMA1RES reset is being generated
3	RESF3	Clock monitor CLMA0 reset flag 0: The CLMA0RES reset is not being generated 1: The CLMA0RES reset is being generated
2	RESF2	Watchdog timer WDTA0 reset flag 0: The WDTA0RES reset is not being generated 1: The WDTA0RES reset is being generated
1	RESF1	Software reset flag 0: The SWRES reset is not being generated 1: The SWRES reset is being generated
0	RESF0	Low voltage indicator reset flag 0: The LVIRES reset is not being generated 1: The LVIRES reset is being generated

Caution 1. If the generation of a reset and writing to the RESFC register are in contention, the setting for generation of the reset takes priority.

Caution 2. In the μPD70F3508 and μPD70F3509, after writing to the RESFS and RESFC registers, actual reflection of the written value in the register takes the number of cycles listed below. If a next value is written within the number of cycles listed below, the value may not be reflected in the register.

HEAPCLK [MHz]	Main OSC [MHz]	Number of cycles (of HEAPCLK)
160	8	120
	16	60
80	8	60
	16	30

(2) RESFC - Reset Source Clear Register

This register clears the reset flags of the RESF register.

Access This register can be read/written in 16-bit units. When read, the value returned is 0000_H.

Address FF42 0028_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESFC15	0	0	0	0	0	0	0	RESFC7	RESFC6	RESFC5	RESFC4	RESFC3	RESFC2	RESFC1	RESFC0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	RW	R/W	R/W	R/W

Table 8-4 RESFC Register Contents (1/2)

Bit Position	Bit Name	Function
15	RESFC15	Controls clearing of the external reset flag (the RESF.RESF15 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF15 is cleared
7	RESFC7	Controls clearing of the self-diagnostic BIST reset flag (the RESF.RESF7 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF7 is cleared
6	RESFC6	Controls clearing of the safety guardian reset flag (the RESF.RESF6 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF6 is cleared
5	RESFC5	Controls clearing of the clock monitor CLMA2 reset flag (the RESF.RESF5 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF5 is cleared
4	RESFC4	Controls clearing of the clock monitor CLMA1 reset flag (the RESF.RESF4 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF4 is cleared

Table 8-4 RESFC Register Contents (2/2)

Bit Position	Bit Name	Function
3	RESFC3	Controls clearing of the clock monitor CLMA0 reset flag (the RESF.RESF3 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF3 is cleared
2	RESFC2	Controls clearing of the watchdog timer WDTA0 reset flag (the RESF.RESF2 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF2 is cleared
1	RESFC1	Controls clearing of the software reset flag (the RESF.RESF1 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF1 is cleared
0	RESFC0	Controls clearing of the low voltage indicator reset flag (the RESF.RESF0 bit). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF0 is cleared

(3) RESFS - Reset Source Set Register

This register is for testing of the RESF register. Even if the flag corresponding to a reset source is set in the RESF register, the corresponding reset will not be generated.

Access This register can be read/written in 16-bit units. When read, the value returned is 0000_H.

Address $FF42\ 0024_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESFS15	0	0	0	0	0	0	0	RESFS7	RESFS6	RESFS5	RESFS4	RESFS3	RESFS2	RESFS1	RESFS0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	RW	R/W	R/W	R/W

Table 8-5 RESFS Register Contents

Bit Position	Bit Name	Function
15	RESFS15	Sets the external reset flag (RESF.RESF15). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF15 is set
7	RESFS7	Sets the self-diagnostic BIST reset flag (RESF.RESF7). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF7 is set
6	RESFS6	Sets the safety guardian reset flag (RESF.RESF6). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF6 is set
5	RESFS5	Sets the clock monitor CLMA2 reset flag (RESF.RESF5). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF5 is set
4	RESFS4	Sets the clock monitor CLMA1 reset flag (RESF.RESF4). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF4 is set
3	RESFS3	Sets the clock monitor CLMA0 reset flag (RESF.RESF3). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF3 is set
2	RESFS2	Sets the watchdog timer WDTA0 reset flag (RESF.RESF2). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF2 is set
1	RESFS1	Sets the software reset flag (RESF.RESF1). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF1 is set
0	RESFS0	Sets the low voltage indicator reset flag (RESF.RESF0). 0: Invalid (Setting this bit to 0 does not affect the flag.) 1: RESF.RESF0 is set

8.3.3 Details of Software Reset Control Registers

(1) SWRESA - Software Reset Register

This register is used to generate a software reset SWRES.

Access This register is write-only with writing in 8-bit units. When read, the value returned is 00_H.

Writing to this register is protected by a special sequence of instructions by using the protection command register CSCPCMD. For details, refer to the section 8.2.10, Protection for Registers of the Reset Controller.

Address FF42 002C$_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SWRESA
R	R	R	R	R	R	R	W

Table 8-6 SWRESA Register Contents

Bit Position	Bit Name	Function
0	SWRESA	Software reset control 0: Invalid (Setting this bit to 0 does not generate a software reset.) 1: A software reset SWRES is generated.

Caution 1. Do not apply a software reset while using the EEPROM emulation library to emulate the operation of data flash memory.

Caution 2. In the μPD70F3508 and μPD70F3509, after 1 is written to this register, actual execution of the reset takes up to the number of cycles indicated below.

HEAPCLK [MHz]	Number of cycles (of HEAPCLK)
160	20
80	10

8.3.4 Details of Protection Command Registers

(1) CSCPCMD - Protection Command Register

This register is the protection command register for the write protected reset controller registers.

Access This register can be written in 8-bit units.

When read, the value returned is 0.

Address FF42 0014$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W

For details, see section 8.2.10, Protection for Registers of the Reset Controller

Table 8-7 CSCPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	–	Protection commands to enable writing to write protected reset controller registers.

(2) CSCPS - Protection Status Register

This register shows the status of the protection sequence operated by the CSCPCMD.

Access This register can be read in 8-bit units.

Writing to this register is ignored.

Address FF42 0018$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CSC PRERR
R	R	R	R	R	R	R	R

Table 8-8 CSCPS Register Contents

Bit Position	Bit Name	Function
0	CSCPRERR	Protected write sequence error monitor 0: No protection error 1: Protection error occurred

8.4 Power-on Flag/Low Voltage Indicator (POF/LVI)

8.4.1 Overview of the POF/LVI Function

POF function The power-on flag (the POF.POF bit) is set to 1 if the voltage generated for the 5-V systems (EVDD) falls below the POF detection voltage.

LVI function The LVI sets the LVI detection flag (LVISF) to 1 and generates an interrupt signal (INTLVI) or a reset signal (LVIRES) if the voltage generated for the 5-V systems (EVDD) falls below the LVI detection voltage.

- The LVIMD bit in the LVICNT register is used to select the output of an interrupt or reset signal in response to detection of a low voltage.
- An internal reset signal is output when the power-supply voltage falls below the detection voltage (this is the case when LVICNT.LVIMD = 1).
- An interrupt request signal is generated when the power-supply voltage falls below the detection voltage (this is the case when LVICNT.LVIMD = 0).
- An interrupt request signal is generated when the power-supply voltage rises above the detection voltage (this is the case when LVICNT.LVIMD = 0).

Table 8-9 LVI Interrupt Request

LVI Signal	Function	Connected to
INTLVI	LVI interrupt signal (for detection of voltage falling below and rising above the given level)	Interrupt controller INTLVI0

8.4.2 Operation

(1) POF Detection

Monitoring of the voltage from the external power supply leads to setting of the power-on flag (the POF.POF bit) if the externally generated voltage falls below the prescribed detection voltage (both when the voltage is supplied and when the voltage is cut off).

Note The state (operating or stopped) is not specifiable by software.

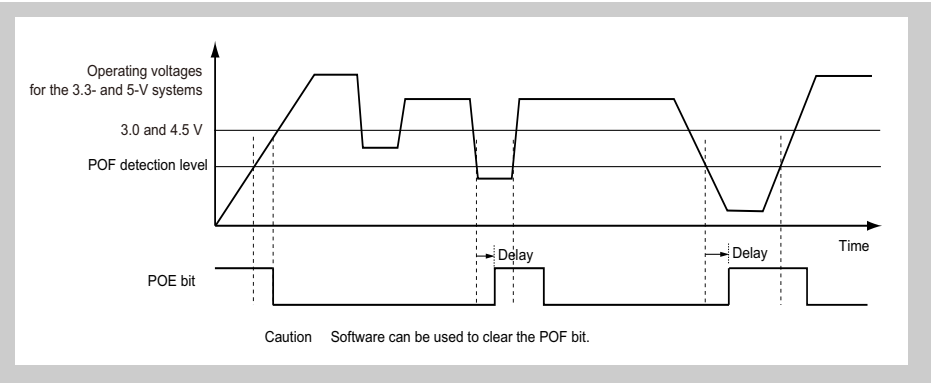


Figure 8-5 Example of POF Operations

(2) LVI Operation

Monitoring of the voltage from the external power supply after setting the LVICNT bit to 1 can be used to generate an interrupt signal (INTLVI) or a reset signal (LVIRES) in accord with the setting of the LVICNT.LVIMD bit when the externally generated voltage falls below the prescribed detection voltage.

(a) Using the LVI for the Output of Reset Signals

<Starting LVI operation>

1. Set the interrupt controller so that the interrupt (INTLVI) is masked.
2. Set the LVICNT.LVICNT bit so that a detection voltage exists.
3. Software handles at least 350 μs of waiting time.
4. Test the value of the LVISF.LVISF bit to confirm that the voltage currently being supplied is above the detection level.
5. Set the LVICNT.LVIMD bit to 1 (selecting the output of a reset signal).

Caution After the LVICNT.LVIMD bit has been set to 1, further changes to the LVICNT register are not possible until the generation of a request for a different type of reset.

<Stopping LVI operation>

Stopping LVI operation is not possible except through the generation of a reset of a different type.

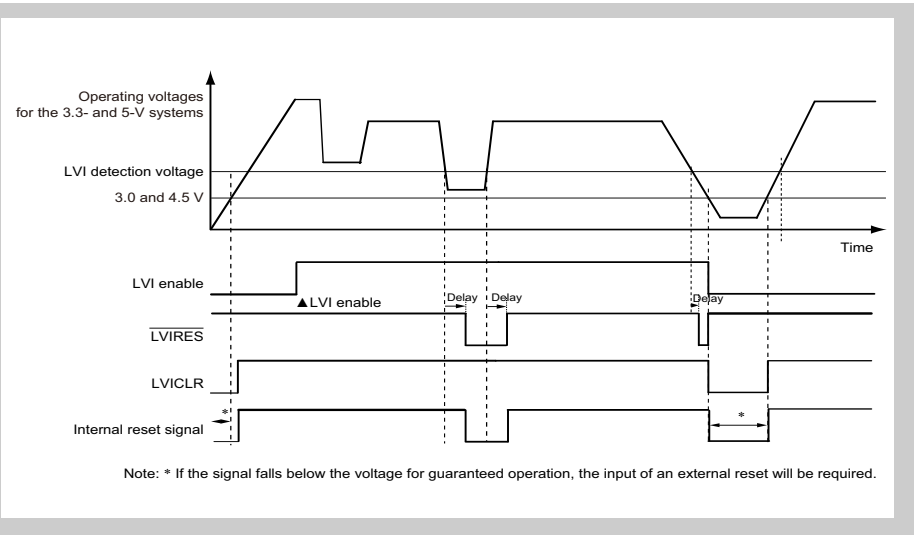


Figure 8-6 Example of LVI Operation (when the LVICNT.LVIMD Bit = 1)

(b) Using the LVI for the Output of Interrupt Request Signals

<Starting LVI operation>

1. Set the interrupt controller so that the interrupt (INTLVI) is masked.
2. Set the LVICNT.LVIMD bit to 0 (selecting the output of an interrupt). Since this is the initial setting, explicitly making this setting is not generally required.
3. Set the LVICNT.LVICNT bit so that a detection voltage exists.
4. Software handles at least 350 µs of waiting time.
5. Test the value of the LVISF.LVISF bit to confirm that the voltage currently being supplied is above the detection level.
6. In the interrupt controller, clear the interrupt request flag for INTLVI.
7. Also in the interrupt controller, release masking of INTLVI.
8. On generation of the interrupt (INTLVI), test the value of the LVISF.LVISF bit to confirm the state of the power-supply voltage.

<Stopping LVI operation>

Set the LVICNT.LVICNT bit to 0.

Caution 1. When an INTLVI is generated, test the value of the LVISF.LVISF bit to confirm the state of the voltage currently being supplied.

Caution 2. To change the detection voltage, first stop the module and then set the voltage by following the same procedure as that for starting LVI operation.

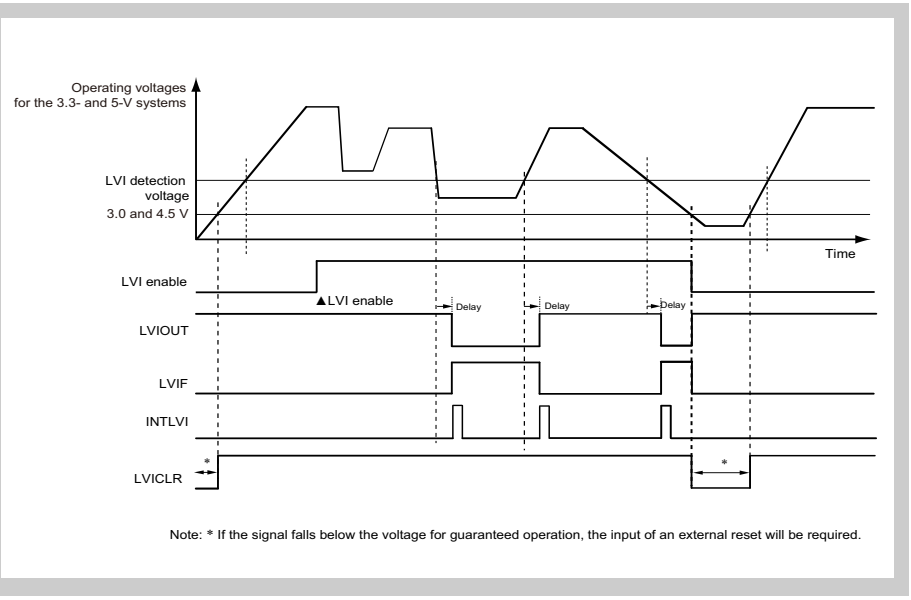


Figure 8-7 Example of LVI Operation (when the LVICNT.LVIMD Bit = 0)

8.4.3 Overview of POF/LVI Registers

Table 8-10 List of POF/LVI Registers

Register Name	Symbol	Address
Power-on flag control registers		
Power-on flag register	POF	FFFF FC00$_H$
Power-on flag clear register	POFC	FFFF FC04$_H$
Power-on flag set register	POFS	FFFF FC08$_H$
LVI control registers		
LVI status flag register	LVISF	FFFF FC10$_H$
LVI control register	LVICNT	FF45 0020$_H$

8.4.4 Details of POF / LVI Control Registers

(1) POF - Power-on Flag Register

This register is used to confirm the state of the power-supply voltage for the 3.3- and 5-V systems (EVDD).

When the voltage has fallen below the detection voltage, the POF bit is set to 1 (i.e. initialized).

Access This register can be read in 8-bit units.

Address FFFF FC00$_H$

Initial value 01$_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POF
R	R	R	R	R	R	R	R

Table 8-11 POF Register Contents

Bit Position	Bit Name	Function
0	POF	This bit indicates the state of detection for the power-on flag. 0: Voltage EVDD is above the detection voltage. 1: Voltage EVDD is below the detection voltage (a low voltage from the power-supply for the 3.3- and 5-V systems has been detected).

Caution Note that the power-on flag function is only effective if the voltage from the power-supply for the 1.2-V systems is within the range for guaranteed operation.

(2) POFC - Power-on Flag Clear Register

This register is used to clear the POF bit in the POF register to 0.

Access This register can be written in 8-bit units.

When read, the value returned is 00_H.

Address FFFF FC04$_H$

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POFC
R	R	R	R	R	R	R	W

Table 8-12 POFC Register Contents

Bit Position	Bit Name	Function
0	POFC	This bit controls clearing of the power-on flag register (the POF.POF bit). Writing 1 to the POFC bit causes clearing of the POF bit to 0. 0: No effect 1: The POF bit is cleared to 0.

(3) POFS - Power-on Flag Set Register

This register is used to set the POF bit in the POF register to 1.

Access This register can be written in 8-bit units.

When read, the value returned is 00_H.

Address FFFF FC04$_H$

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POFS
R	R	R	R	R	R	R	W

Table 8-13 POFS Register Contents

Bit Position	Bit Name	Function
0	POFS	Writing 1 to the POFS bit causes setting of the POF bit to 1. 0: No effect 1: The POF bit is set to 1.

(4) LVICNT - LVI Control Register

This register is used to set the operating mode for the LVI.

Once this register is set with the LVIMD and LVICNT bits set to 1, further changes to the setting are prohibited until the generation of a reset signal by a source other than the LVI (indicated by $\overline{\text{LVICLR}}$ being at the active level).

Input of the reset signal ($\overline{\text{LVICLR}}$) at the active level (indicating a reset generated by a source other than the LVI) initializes this register to 00_H.

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

Address $FF45\ 0020_H$

Initial value 00_H

This register is initialized by a reset generated by a source other than the LVI.

7	6	5	4	3	2	1	0
LVIMD	0	0	0	0	0	LVICNT1	LVICNT0
R/W	R	R	R	R	R	R/W	R/W

Table 8-14 LVICNT Register Contents

Bit Position	Bit Name	Function
7	LVIMD	This bit specifies the output signal in response to the LVI detecting an abnormal voltage. 0: The output is an interrupt signal ($\overline{\text{INTLVI}}$). 1: The output is a reset signal ($\overline{\text{LVIRES}}$).
1,0	LVICNT1, LVICNT0	These bits enables/disables voltage detection by the LVI.

μPD70F3505A, μPD70F3507, μPD70F3509

LVICNT1	LVICNT0	Function
0	0	Voltage detection by the LVI is disabled.
0	1	Voltage detection by the LVI is enabled. The detection voltage is 4.6 ± 0.1 V.
1	0	Voltage detection by the LVI is enabled. The detection voltage is 4.3 ± 0.1 V.
1	1	Setting prohibited

μPD70F3506, μPD70F3508

LVICNT1	LVICNT0	Function
0	0	Voltage detection by the LVI is disabled.
0	1	Setting prohibited
1	0	Setting prohibited
1	1	Voltage detection by the LVI is enabled. The detection voltage is 3.1 ± 0.1 V.

Note For details on the detection voltage, see Section 28 Electrical Characteristics.

(5) LVISF - LVI Status Flag Register

This register indicates the current state of detection by the LVI.

Access This register can be read in 8-bit units.

Address FFFF FC10$_H$

Initial value 00$_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	LVISF
R	R	R	R	R	R	R	R

Table 8-15 LVISF Register Contets

Bit Position	Bit Name	Function
0	LVISF	This is the LVI detection flag and indicates the current state of detection. 0: The power-supply voltage for the 3.3- and 5-V systems (EVDD) is not low (or the LVI is not operating). 1: The power-supply voltage for the 3.3- and 5-V systems (EVDD) is low.

Section 9 Safety Functions

This section describes an overview of the safety functions.

The safety functions are as follows.

- Peripheral bus access error detection
- Memory access protection
- Self-diagnostic BIST
- ECC

9.1 Peripheral Bus Access Error Detection

This product incorporates a detector for errors in peripheral bus access.

This is capable of detecting errors in the form of

- access to peripheral modules,
- access to undefined areas, and
- access timeout.

9.2 Memory Access Protection

This product incorporates memory-access protection functions.

- CPU core protection
  - Memory protection (MPU)
  - Peripheral protection (PPU)

9.2.1 Memory Access Protection

CPU core protection can be applied to memory areas and the address spaces specified in peripheral function registers. Rights to access can be assigned in accord with the state of software (for example, to set up a distinction between rights to access by the application program and drivers on the one hand and the operating system on the other).

For details, refer to the V850E2M Architecture Manual (R01US0001E).

9.3 Registers Related to the Peripheral I/O Bus

Registers related to the peripheral I/O bus are described in this section.

9.3.1 Overview of Registers Related to the Peripheral I/O Bus

Table 9-1 List of Registers Related to the Peripheral I/O Bus

Register Name	Symbol	Address
Peripheral I/O bus PSELG control register	APC	FF45 4000_H
Peripheral I/O bus PSELG error status register	APES	FF45 4004_H
Peripheral I/O bus PSELG error status clear register	APEC	FF45 4008_H
Peripheral I/O bus PSELG error address storage register	APAM	FF45 $400C_H$
Peripheral I/O bus maximum access time set register	MATS	FF45 4010_H

9.3.2 Details of Registers Related to the Peripheral I/O Bus

(1) APC - Peripheral I/O Bus PSELG Control Register

This register is used to specify several forms of checking for errors in access over the peripheral I/O bus.

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

Address FF45 4000$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	SLV*	UAA	0	ATO
R	R	R	R	R/W	R/W	R	R/W

Table 9-2 APC Register Contents

Bit Position	Bit Name	Function
3	SLV*	0/1: Error checking is executed on access to peripheral functions (EDC). This source flag is set when access to a peripheral function (TSG20, TSG21, or ADCA0) leads to an error on the data bus. This error checking always proceeds in case of such access.
2	UAA	0: Checking for access to undefined areas does not proceed. 1: Checking for access to undefined areas proceeds. Undefined area is the area which does not belong to the peripheral I/O area.
0	ATO	0: Checking for access timeout does not proceed. 1: Checking for access timeout proceeds. This source flag is set when the time taken for access to the peripheral I/O bus exceeds the maximum access time specified in the MATS register.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

(2) APES - Peripheral I/O Bus PSELG Error Status Register

This register contains error flags generated during the peripheral I/O bus access.

Access This register can be read in 8-bit units.

Address FF45 4004$_H$

Initial value 00$_H$

This register is initialized by a reset from any source or by writing to the APEC register.

7	6	5	4	3	2	1	0
0	0	0	0	SLV*	UAA	0	ATO
R	R	R	R	R	R	R	R

Table 9-3 Contents of the APES Register

Bit Position	Bit Name	Function
3	SLV*	0: A slave error has not been generated. 1: A slave error has been generated.
2	UAA	0: An undefined-area-access error has not been generated. 1: An undefined-area-access error has been generated.
0	ATO	0: An access-timeout error has not been generated. 1: An access-timeout error has been generated.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

Caution Once any of the ATO, UAA, and SLV bits in this register is set to 1, it remains so until it is cleared by writing to the APEC register or by a reset.

(3) APEC - Peripheral I/O Bus PSELG Error Status Clear Register

This register is used to clear the several error flags which may be generated by access over the peripheral I/O bus.

Access This register can be written in 8-bit units.

When read, the value returned is 00_H.

Address FF45 4008_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	SLV*	UAA	0	ATO
R	R	R	R	R/W	R/W	R	R/W

Table 9-4 Contents of the APEC Register

Bit Position	Bit Name	Function
3	SLV*	0: The slave-error source flag in the APES register is not cleared. 1: The slave-error source flag in the APES register is cleared.
2	UAA	0: The undefined-area-access error source flag in the APES register is not cleared. 1: The undefined-area-access error source flag in the APES register is cleared.
0	ATO	0: The access-timeout error source flag in the APES register is not cleared. 1: The access-timeout error source flag in the APES register is cleared.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

(4) APAM - Peripheral I/O Bus PSELG Error Address Store Register

This register holds the address where access over the peripheral I/O bus led to any of the several errors (the μPD70F3508 and μPD70F3509 do not have this function).

In cases where multiple errors are generated, this register holds the address of the most recent error.

Access This register can be read in 32-bit units.

Address FF45 400C$_H$

Initial value FF00 0000$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	1	1	1	1	1	1	1	A23	A22	A21	A20	A19	A18	A17	A16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-5 Contents of the APAM Register

Bit Position	Bit Name	Function
23 to 0	A[23:0]	These bits hold the address of the most recent error generated due to attempted access.

Caution Since this register holds the address of the most recent error, the stored address might not be that corresponding to the source which led to setting of an error status bit (APES.ATO, APES.UAA, and APES.SLV) to 1.

(5) MATS - Peripheral I/O Bus Maximum Access Time Set Register

This register is used to prevent blocking of access to the peripheral I/O bus by specifying the maximum bus access time (timeout time). Access over the peripheral I/O bus is forcibly ended when the time taken exceeds the time specified here. At this time, the access-timeout error flag (ATO) in the APES register is also set to 1.

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

Address FF45 4010$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	MATS3	MATS2	MATS1	MATS0
R	R	R	R	R/W	R/W	R/W	R/W

Table 9-6 MATS Register Contents

Bit Position	Bit Name	Function
3 to 0	MATS[3:0]	The available settings for maximum bus access time are listed below.

MATS3	MATS2	MATS1	MATS0	Access Time
0	0	0	0	2^9/PCLK
0	0	0	1	2^{10}/PCLK
0	0	1	0	2^{11}/PCLK
0	0	1	1	2^{12}/PCLK
0	1	0	0	2^{13}/PCLK
0	1	0	1	2^{14}/PCLK
0	1	1	0	2^{15}/PCLK
0	1	1	1	2^{16}/PCLK
Other than above				Setting prohibited

9.4 Overview of Self-Diagnostic BIST

This section contains a general description of the self-diagnostic BIST.

Caution The registers listed below are beyond the scope of self-diagnostic BIST. Accordingly, except in the case of BSEQ0STRHBT and BSEQ0CTL, software must run self-diagnosis for these registers (by reading from and writing to the register or setting and clearing its bits).

Although setting and clearing of the bits in BSEQ0STRHBT is not possible, self diagnosis can take the form of checking the difference between master and checker bits in this register. See table 9-10, Contents of the BSEQ0STRHBT Register (for the µPD70F3505A, µPD70F3506, and µPD70F3507), or table 9-11, Contents of the BSEQ0STRHBT Register (for the µPD70F3508 and µPD70F3509).

BSEQ0CTL has a redundant configuration and abnormalities can be judged from the state after writing (refer to table 9-15, Values Read from the BSEQ0CTL Register and the State of Writing).

- POF (power on flag register)
- RESF (reset source flag register)
- SGAMESSTR0-1 (error source status registers 0 and 1, master)
- SGAMESSTR0A-C (error source status registers 0A to 0C, master)
- SGAMESSTR1A-C (error source status registers 1A to 1C, master)
- SGACESSTR0-1 (error source status registers 0 and 1, checker)
- SGACESSTR0A-C (error source status registers 0A to 0C, checker)
- SGACESSTR1A-C (error source status registers 1A to 1C, checker)
- BSEQ0STRHBT (self-diagnostic BIST status register)
- BRAMDAT3-0 (backup RAM3-0 registers)
- BSEQ0CTL (self-diagnostic BIST control register)-only for the µPD70F3505A

9.4.1 Self-Diagnostic BIST Skip Function (Only for the µPD70F3505A)

The µPD70F3505A contains a facility for software to select skipping of the self-diagnostic BIST. Skipping the self-diagnostic BIST shortens the time to boot up from a reset when the power is on.

Caution If a pin reset coincides with RAM access, the value in RAM may be undefined after the reset. Accordingly, stop DMA transfer and place the controller in halted mode before input of the pin-reset signal.

9.5 Self-Diagnostic BIST Related Registers

This section contains a general description of the registers related to self-diagnostic BIST.

9.5.1 Overview of Registers Related to Self-Diagnostic BIST

Table 9-7 List of Registers Related to Self-Diagnostic BIST

Register Name	Symbol	Address
BIST protection command register	BSEQ0 TCRPCMD	FF83 $B000_H$
BIST protection status register	BSEQ0 TCRPESR	FF83 $B004_H$
Self-Diagnostic BIST status register	BSEQ0 STRHBT	FF83 $B008_H$
Self-Diagnostic BIST status clear trigger register	BSEQ0 STCHBT	FF83 $B00C_H$
SGA error pulse control register	SGAEPCTL	FF83 $F020_H$
Backup RAM register 0	BRAMDAT0	FF83 $F030_H$
Backup RAM register 1	BRAMDAT1	FF83 $F034_H$
Backup RAM register 2	BRAMDAT2	FF83 $F038_H$
Backup RAM register 3	BRAMDAT3	FF83 $F03C_H$
Self-Diagnostic BIST control register*	BSEQ0CTL	FF42 $003C_H$

Note * Only for the μPD70F3505A.

9.5.2 Details of Registers Related to Self-Diagnostic BIST

(1) BSEQ0TCRPCMD - BIST Protection Command Register

This register is a command register for the write-protected BIST registers.

Access This register can be written in 32-bit units.

Address FF83 B000$_H$

Initial value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 9-8 BSEQ0TCRPCMD Register Contents

Bit Position	Bit Name	Function
31 to 0		Write enable command for the write-protected BIST registers

(2) BSEQ0TCRPESR - BIST Protection Status Register

This register indicates the state in the sequence of protection executed by the BSEQ0TCRPCMD register.

Access This register can be read in 32-bit units.

Address FF83 B004$_H$

Initial value 0000 0000$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PESR0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-9 BSEQ0TCRPESR Register Contents

Bit Position	Bit Name	Function
0	PESR0	0: The generation of a protection error is not detected by protected-write-sequence error monitoring. 1: The generation of a protection error is detected by protected-write-sequence error monitoring.

(3) BSEQ0STRHBT - Self-Diagnostic BIST Status Register

This register indicates the state of self-diagnostic BIST.

Access This register can be read in 32-bit units.

Address FF83 B008$_H$

Initial value 0000 FF00$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCE	MPE	CPE	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWBS15	HWBS14	HWBS13	HWBS12	HWBS11	HWBS10	HWBS9	HWBS8	HWBS7	HWBS6	HWBS5	HWBS4	HWBS3	HWBS2	HWBS1	HWBS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-10 Contents of the BSEQ0STRHBT Register (for the µPD70F3505A, µPD70F3506, and µPD70F3507) (1/2)

Bit Position	Bit Name	Function
31	BCE	0: BIST sequencer comparison errors are not generated. 1: BIST sequencer comparison errors are generated.
30	MPE	0: Self-diagnostic BIST code-parity errors are not generated (on the master side). 1: Self-diagnostic BIST code-parity errors are generated (on the master side).
29	CPE	0: Self-diagnostic BIST code-parity errors are not generated (on the checker side). 1: Self-diagnostic BIST code-parity errors are generated (on the checker side).
15 to 8	HWBS[15:8]	The state of self-diagnostic BIST by the checker's BIST sequencer 0: The sequence of self-diagnostic BIST by the checker ended normally. 1: The sequence of self-diagnostic BIST by the checker ended abnormally, did not end, or has not started.

HWBS Bit	Memory /Logic	Test Group
15	Memory	When OPBT0 bit 5 is set to 1, the results of testing for the FlexRay RAM are stored.
14		(CSIH RAM and FCN RAM) PBUS1 group
13		Flash cache
12		On-chip RAM
11	Logic	When OPBT0 bit 5 is set to 1, the result of testing for the FlexRay interface is stored.
10		PBUS1 group
9		Checker group
8		Master group

Table 9-10 Contents of the BSEQ0STRHBT Register (for the µPD70F3505A, µPD70F3506, and µPD70F3507) (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 0</td><td>HWBS[7:0]</td><td>The state of self-diagnostic BIST by the master's BIST sequencer
0: The sequence of self-diagnostic BIST by the master ended abnormally, did not end, or has not started.
1: The sequence of self-diagnostic BIST by the master ended normally.
<table>
<tr><th>HWBS Bit</th><th>Memory /Logic</th><th>Test Group</th></tr>
<tr><td>7</td><td rowspan="4">Memory</td><td>When OPBT0 bit 5 is set to 1, the results of testing of the FlexRay RAM are stored.</td></tr>
<tr><td>6</td><td>(CSIH RAM and FCN RAM) PBUS1 group</td></tr>
<tr><td>5</td><td>Flash cache</td></tr>
<tr><td>4</td><td>On-chip RAM</td></tr>
<tr><td>3</td><td rowspan="4">Logic</td><td>When OPBT0 bit 5 is set to 1, the result of testing for the FlexRay interface is stored.</td></tr>
<tr><td>2</td><td>PBUS1 group</td></tr>
<tr><td>1</td><td>Checker group</td></tr>
<tr><td>0</td><td>Master group</td></tr>
</table>
</td></tr>
</table>

Note Master group: Master-side CPU and SGA, on-chip RAM, flash cache
Checker group: Checker-side CPU and SGA, CMP, WDTA, CLMA
PBUS1 group: Timers, serial interfaces, ports, ADC (controller), DCRA, PIC, BRGA, DNF

Caution When using the µPD70F3506 with HEAPCLK running at 48 MHz and OPBT0.FOP5 = 1, mask the HWBS15, HWBS11, HWBS7, and HWBS3 bits related to the FlexRay interface.

Table 9-11 Contents of the BSEQ0STRHBT Register (for the µPD70F3508 and µPD70F3509)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31</td><td>BCE</td><td>0: BIST sequencer comparison errors are not generated.
1: BIST sequencer comparison errors are generated.</td></tr>
<tr><td>30</td><td>MPE</td><td>0: Self-diagnostic BIST code-parity errors are not generated (on the master side).
1: Self-diagnostic BIST code-parity errors are generated (on the master side).</td></tr>
<tr><td>29</td><td>CPE</td><td>0: Self-diagnostic BIST code-parity errors are not generated (on the checker side).
1: Self-diagnostic BIST code-parity errors are generated (on the checker side).</td></tr>
<tr><td>15 to 12</td><td>HWBS[15:12]</td><td>State of self-diagnostic BIST of the checker's BIST sequencer memory
0: Self-diagnostic BIST of the checker ended normally.
1: Self-diagnostic BIST of the checker ended abnormally, did not end, or has not started.
<table>
<tr><th>HWBS Bit</th><th>Test Group</th></tr>
<tr><td>15</td><td>CSIH RAM, flash cache</td></tr>
<tr><td>14</td><td>FlexRay RAM1, FCN RAM1</td></tr>
<tr><td>13</td><td>On-chip RAM</td></tr>
<tr><td>12</td><td>FlexRay RAM2, FCN RAM2</td></tr>
</table></td></tr>
<tr><td>11 to 8</td><td>HWBS[11:8]*</td><td>State of self-diagnostic BIST of the checker's BIST sequencer logic
0: Self-diagnostic BIST of the checker ended normally.
1: Self-diagnostic BIST of the checker ended abnormally, did not end, or has not started.</td></tr>
<tr><td>7 to 4</td><td>HWBS[7:4]</td><td>State of self-diagnostic BIST of the master's BIST sequencer memory
0: The Self-diagnostic BIST of the master ended abnormally, did not end, or has not started.
1: Self-diagnostic BIST of the master ended normally.
<table>
<tr><th>HWBS Bit</th><th>Test Group</th></tr>
<tr><td>7</td><td>CSIH RAM, flash cache</td></tr>
<tr><td>6</td><td>FlexRay RAM1, FCN RAM1</td></tr>
<tr><td>5</td><td>On-chip RAM</td></tr>
<tr><td>4</td><td>FlexRay RAM2, FCN RAM2</td></tr>
</table></td></tr>
<tr><td>3 to 0</td><td>HWBS[3:0]*</td><td>State of self-diagnostic BIST of the master's BIST sequencer logic
0: Self-diagnostic BIST of the master ended abnormally, did not end, or has not started.
1: Self-diagnostic BIST of the master ended normally.</td></tr>
</table>

Note * To maintain software compatibility with the µPD70F3505A, µPD70F3506, and µPD70F3507, the states of self-diagnostic BIST in these bits are set according to the same criteria.

Note FlexRay RAM1: Input buffer, output buffer, transmit buffer

FlexRay RAM2: Message buffer

FCN RAM1: Message buffer

FCN RAM2: Message buffer (with a history list)

(4) BSEQ0STCHBT - Self-Diagnostic BIST Status Clear Trigger Register

This register is used to clear error source flags in the BSEQ0STRHBT register. This register has a designated sequence, so only access in that sequence will be effective for writing. For details, see section 9.5.3, Procedure for Setting the BSEQ0STCHBT Register.

Access This register can be written in 32-bit units.

Address FF83 B00C$_H$

Initial value 0000 0000$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLHWBS31	CLHWBS30	CLHWBS29	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLHWBS15	CLHWBS14	CLHWBS13	CLHWBS12	CLHWBS11	CLHWBS10	CLHWBS9	CLHWBS8	CLHWBS7	CLHWBS6	CLHWBS5	CLHWBS4	CLHWBS3	CLHWBS2	CLHWBS1	CLHWBS0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 9-12 Contents of the BSEQ0STCHBT Register

Bit Position	Bit Name	Function
31	CLHWBS31	This bit clears the HWBS31 bit in the BSEQ0STRHBT register. 0: Invalid (Setting this bit to 0 does not affect the BSEQ0STRHBT register.) 1: Bit HWBS31 is cleared
30	CLHWBS30	This bit clears the HWBS30 bit in the BSEQ0STRHBT register. 0: Invalid (Setting this bit to 0 does not affect the BSEQ0STRHBT register.) 1: Bit HWBS30 is cleared
29	CLHWBS29	This bit clears the HWBS29 bit in the BSEQ0STRHBT register. 0: Invalid (Setting this bit to 0 does not affect the BSEQ0STRHBT register.) 1: Bit HWBS29 is cleared
15 to 8	CLHWBS[15:8]	This bit sets the HWBS[15:8] bits in the BSEQ0STRHBT register. 0: Invalid (Setting this bit to 0 does not affect the BSEQ0STRHBT register.) 1: Bits HWBS[15:8] are cleared
7 to 0	CLHWBS[7:0]	This bit sets the HWBS[7:0] bits in the BSEQ0STRHBT register. 0: Invalid (Setting this bit to 0 does not affect the BSEQ0STRHBT register.) 1: Bits HWBS[7:0] are cleared

Caution With the µPD70F3505A, µPD70F3506, and µPD70F3507, when clearing the status flags, be sure to clear all required bits with a single access. (Write E000FFFF$_H$ to the BSEQ0STCHBT register.)

(5) SGAEPCTL - SGA Error Pulse Control Register

This register is used to select the timer for dynamic-mode operation (for details, see section 10.3.3, Operations for Error Output).

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

Address FF83 F020$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SGATMSL
R	R	R	R	R	R	R	R/W

Table 9-13 Contents of the SGAEPCTL Register

Bit Position	Bit Name	Function
0	SGATMSL	This bit selects the source of the timer for input to the safety guardian. 0: TAUA1 channel15 1: OSTM1

(6) BRAMDATn (n = 0 to 3) - Back-up RAM Register

The contents of on-chip RAM are undefined after self-diagnostic BIST. Since this register may hold values before the execution of self-diagnostic BIST, save the data beforehand. This 128-bit register is composed of four registers.

Access This register can be read/written in 32-bit units.

Address FF83 F030$_H$ (BRAMDAT0)

FF83 F034$_H$ (BRAMDAT1)

FF83 F038$_H$ (BRAMDAT2)

FF83 F03C$_H$ (BRAMDAT3)

Initial value Undefined

127-96	95-64	63-32	31-0
BRAMDAT3	BRAMDAT2	BRAMDAT1	BRAMDAT0
R/W	R/W	R/W	R/W

(7) BSEQ0CTL - Self-Diagnostic BIST Control Register

This register is used to specify whether to skip or execute self-diagnostic BIST (only for the μPD70F3505A). Since this register is write-protected, it is only accessible by following the designated sequence (regarding the designated sequence for the BSEQ0CTL register, refer to section 8.2.10, Protection for Registers of the Reset Controller). This register is initialized when power is supplied and when the 5-V power supply (EVDD) falls below the detection voltage as indicated by the POF.

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

Address FF42 003C$_H$

Initial value 01$_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	HWBISTENM	HWBISTEN
R	R	R	R	R	R	R	R/W

Table 9-14 Contents of the BSEQ0CTL Register

Bit Position	Bit Name	Function
1	HWBISTENM	Status flag When bits 0 and 1 are read, the combination of their values can be used to confirm the state of writing.
0	HWBISTEN	This bit specifies execution or skipping of self-diagnostic BIST. 0: BIST is skipped. 1: BIST is executed.

Note This register contains status flags that indicate the state of writing. That is, values read from the BSEQ0CTL register can be used to check the state of writing.

Table 9-15 Values Read from the BSEQ0CTL Register and the State of Writing

Value Read from the BSEQ0CTL Register	Settings and Corresponding States of Writing
01$_H$	Self-diagnostic BIST is being executed.
02$_H$	Self-diagnostic BIST is being skipped.
03$_H$	Self-diagnostic BIST is being executed but abnormal operation has arisen during writing.

Caution 1. This register is reset when the POF function for monitoring the external power supply detects a low voltage. Accordingly, if you wish to enable the POF function, ensure that the power-supply voltage from the internal power-supply system rises before that from the external power-supply system or that both rise at the same time. Regarding the timing with which the voltages rise, design the systems so that the voltage from the internal power-supply system reaches the reference voltage (1.1 V) before the voltage from the external power-supply system reaches 2.7 V.

Caution 2. Settings in this register can be changed only once after powering on. The reset by specifying skipping of self-diagnostic BIST casues the register to restart. To change the settings is not enabled after the reset (though writing the same value again to the register makes no affect).

9.5.3 Procedure for Setting the BSEQ0STCHBT Register

The BSEQ0STCHBT register is write-protected, and is thus only writable by following the designated sequence below.

Step 1: Write the designated value ($000000A5_H$) to the protection-command register (BSEQ0TCRPCMD).

Step 2: Write the desired setting to the target write-protected register (BSEQ0STCHBT). (Writing is invalid in this step.)

Step 3: Write the inverse of the desired setting to the target write-protected register (BSEQ0STCHBT). (Writing is invalid in this step.)

Step 4: Again write the desired setting to the target write-protected register (BSEQ0STCHBT). (Writing is effective in this step.)

9.6 ECC Related Registers

This section describes the ECC related registers.

9.6.1 Overview of ECC Related Registers

Table 9-16 Contents of the ECC Related Registers

Register Name	Symbol	Address
Code flash ECC		
Code flash ECC error flag register	CECCER	FF43 2000$_H$
Code flash ECC error flag clear register	CECCERC	FF43 2004$_H$
Code flash ECC error correction address register	CECADR	FF43 2008$_H$
Code flash ECC error detection address register	CEDADR	FF43 200C$_H$
On-chip RAM ECC		
On-chip RAM ECC error flag register	LECCER	FF46 8000$_H$
On-chip RAM ECC error flag clear register	LECCERC	FF46 8004$_H$
On-chip RAM ECC error correction address register	LECADR	FF46 8008$_H$
On-chip RAM ECC error detection address register	LEDADR	FF46 800C$_H$
CAN RAM ECC		
CAN0 ECC0 control register	E6A0CTL	FF46 0000$_H$
CAN0 ECC1 control register	E6A1CTL	FF46 0100$_H$
CAN1 ECC0 control register	E6A2CTL	FF46 0800$_H$
CAN1 ECC1 control register	E6A3CTL	FF46 0900$_H$
FlexRay RAM ECC*		
FlexRay ECC0 control register	E7A0CTL	FF46 4000$_H$
FlexRay ECC1 control register	E7A1CTL	FF46 4100$_H$
FlexRay ECC2 control register	E7A2CTL	FF46 4200$_H$
FlexRay ECC3 control register	E7A3CTL	FF46 4300$_H$
FlexRay ECC4 control register	E7A4CTL	FF46 4400$_H$
FlexRay ECC5 control register	E7A5CTL	FF46 4500$_H$
FlexRay ECC6 control register	E7A6CTL	FF46 4600$_H$

Note * The µPD70F3505A does not have this function.

Caution: When an ECC error is generated from the contents or tag bits of the flash cache and flash cache tag, the error is notified but the error address is not retained.

9.6.2 Details of ECC Related Registers

(1) CECCER - Code Flash ECC Error Flag Register

The corresponding bit in this register is set when an ECC error is generated.

Access This register can be read in 8-bit units.

Address FF43 2000$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CECCER0 EDFLG	CECCER0 ECFLG
R	R	R	R	R	R	R	R

Table 9-17 Contents of the CECCER Register

Bit Position	Bit Name	Function
1	CECCER0EDFLG	0: ECC uncorrectable error is not generated. 1: ECC uncorrectable error is generated.
0	CECCER0ECFLG	0: ECC correction does not occur. 1: ECC correction occurs.

(2) CECCERC - Code Flash ECC Error Flag Clear Register

This register is used to clear values set in the CECCER register. Specifically, writing 1 to an effective bit in this register clears the corresponding bit in the CECCER register.

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

When read, the value returned is always 00_H.

Address $FF43\ 2004_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CECCER0 EDCLR	CECCER0 ECCLR
R	R	R	R	R	R	R/W	R/W

Table 9-18 Contents of the CECCERC Register

Bit Position	Bit Name	Function
1	CECCER0 EDCLR	This bit controls clearing of the code-flash ECC error flag (the CECCER.CECCER0EDFLG bit). 0: CECCER.CECCER0EDFLG is not cleared. 1: CECCER.CECCER0EDFLG is cleared.
0	CECCER0 ECCLR	This bit controls clearing of the code-flash ECC error flag (the CECCER.CECCER0ECFLG bit). 0: CECCER.CECCER0ECFLG is not cleared. 1: CECCER.CECCER0ECFLG is cleared.

(3) CECADR - Code Flash ECC Error Correction Address Register

If ECC correction occurs, this register holds the corresponding address. If the CECCER.CECCER0ECFLG is 0, the stored address where the error was encountered is not changed until clearing of the CECCER0ECFLG.

Access This register can be read in 32-bit units.

Address FF43 2008$_H$

Initial value 0000 0000$_H$

This register is initialized by clearing the CECCER.CECCER0ECFLG bit or by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	CECADR0A[21:16]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CECADR0A[15:4]												0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-19 Contents of the ECC Error Correction Address Register

Bit Position	Bit Name	Function
21 to 4	CECADR0A [21:4]	These bits hold the address where ECC correction was applied.

(4) CEDADR - Code Flash ECC Error Detection Address Register

When an ECC error is encountered, this register holds the corresponding address. If CECCER.CECCER0EDFLG is 0, the stored address where the error was encountered is not changed until clearing of CECCER0EDFLG.

Access This register can be read in 32-bit units.

Address FF43 200C$_H$

Initial value 0000 0000$_H$

This register is initialized by clearing the CECCER.CECCER0EDFLG bit or by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	CEDADR0A[21:16]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CEDADR0A[15:4]												0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-20 Contents of the ECC Error Detection Address Register

Bit Position	Bit Name	Function
21 to 4	CEDADR0A [21:4]	These bits hold the address where the ECC error was encountered.

(5) LECCER - On-chip RAM ECC Error Flag Register

This register indicates the occurrence of ECC correctable and uncorrectable errors in the on-chip RAM.

Access This register can be read in 8-bit units.

Address FF46 8000$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	LECCER0 EDFLG	LECCER0 ECFLG
R	R	R	R	R	R	R	R

Table 9-21 Contents of the LECCER Register

Bit Position	Bit Name	Function
1	LECCER0 EDFLG	0: ECC uncorrectable error is not generated in the on-chip RAM. 1: ECC uncorrectable error is generated in the on-chip RAM.
0	LECCER0 ECFLG	0: ECC correction does not occur in the on-chip RAM. 1: ECC correction occurs in the on-chip RAM.

(6) LECCERC - On-chip RAM ECC Error Flag Clear Register

This register is used to clear values set in the LECCER register. Specifically, writing 1 to an effective bit in this register clears the corresponding bit in the LECCER register.

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

When read, the value returned is 00_H.

Address FF46 8004_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	LECCER0EDCLR	LECCER0ECCLR
R	R	R	R	R	R	R/W	R/W

Table 9-22 Contents of the LECCERC Register

Bit Position	Bit Name	Function
1	LECCER0 EDCLR	This bit controls clearing of the on-chip RAM ECC error flag (the LECCER.LECCER0EDFLG bit). 0: LECCER.LECCER0EDFLG is not cleared. 1: LECCER.LECCER0EDFLG is cleared.
0	LECCER0 ECCLR	This bit controls clearing of the on-chip RAM ECC error flag (the LECCER.LECCER0ECFLG bit). 0: LECCER.LECCER0ECFLG is not cleared. 1: LECCER.LECCER0ECFLG is cleared.

(7) LECADR - On-chip RAM Error Correction Address Register

When an ECC error is encountered in access to on-chip RAM, this register holds the corresponding address. If LECCER.LECCER0ECFLG is 0, the stored address where the error was encountered is not changed until clearing of LECCER0ECFLG.

Access This register can be read in 32-bit units.

Address FF46 8008$_H$

Initial value 0000 0000$_H$

This register is initialized by clearing the LECCER.LECCER0ECFLG bit or by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	LECADR0A[20:16]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LECADR0A[15:2]														0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-23 Contents of the On-chip RAM Error Correction Address Register

Bit Position	Bit Name	Function
20 to 2	LECADR0A [20:2]	These bits hold the address where ECC was applied to on-chip RAM.

(8) LEDADR - On-chip RAM ECC Error Detection Address Register

When an ECC error is encountered in access to on-chip RAM, this register holds the corresponding address. If LECCER.LECCER0EDFLG is 0, the stored address where the error was encountered is not changed until clearing of LECCER0EDFLG.

Access This register can be read in 32-bit units.

Address FF46 800C$_H$

Initial value 0000 0000$_H$

This register is initialized by clearing the LECCER.LECCER0EDFLG bit or by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	LEDADR0A[20:16]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEDADR0A[15:2]														0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9-24 Contents of the On-chip RAM ECC Error Detection Address Register

Bit Position	Bit Name	Function
20 to 2	LEDADR0A [20:2]	These bits hold the address where an ECC error was encountered in access to on-chip RAM.

(9) E6AxCTL (x = 0, 1) - CAN0 ECC0/1 Control Register

When an ECC error is generated in the CAN RAM, the corresponding bit in this register is set to 1 or the current setting is cleared. Writing 1 to a bit in this register clears the bit.

Access This register can be read/written in 32-bit units.

The value read from E6AxCTL.E6AxCTLER2C (x = 0, 1) is always 0.

Address FF46 0000_H (E6A0CTL)

FF46 0100_H (E6A1CTL)

Initial value Undefined

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	E6AxCTLER2C	0	0	0	0	0	0	0	E6AxCTLECCER2	0/1	0/1
R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

Table 9-25 Contents of the E6AxCTL Register (x = 0, 1)

Bit Position	Bit Name	Function
10	E6AxCTLER2C (x = 0, 1)	0: The E6AxCTL.E6AxCTLECCER2 (x = 0, 1) bits are not cleared. 1: The E6AxCTL.E6AxCTLECCER2 (x = 0, 1) bits are cleared.
2	E6AxCTLECCER2 (x = 0, 1)	0: An ECC error in the CAN RAM has not been generated. 1: An ECC error in the CAN RAM has been generated.

(10) E6AxCTL (x = 2, 3) - CAN1 ECC0/1 Control Register

When an ECC error is generated in the CAN RAM, the corresponding bit in this register is set to 1 or the current setting is cleared. Writing 1 to a bit in this register clears the bit.

Access This register can be read/written in 32-bit units.

The value read from E6AxCTL.E6AxCTLER2C (x = 2, 3) is always 0.

Address FF46 0800$_H$ (E6A2CTL)

FF46 0900$_H$ (E6A3CTL)

Initial value Undefined

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	E6AxCTLER2C	0	0	0	0	0	0	0	E6AxCTLECCER2	0/1	0/1
R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

Table 9-26 Contents of the E6AxCTL Register (x = 2, 3)

Bit Position	Bit Name	Function
10	E6AxCTLER2C (x = 2, 3)	0: The E6AxCTL.E6AxCTLECCER2 (x = 2, 3) bits are not cleared. 1: The E6AxCTL.E6AxCTLECCER2 (x = 2, 3) bits are cleared.
2	E6AxCTLECCER2 (x = 2, 3)	0: An ECC error in the CAN RAM has not been generated. 1: An ECC error in the CAN RAM has been generated.

(11) E7AxCTL (x = 0-6) - FlexRay ECC0/1/2/3/4/5/6 Control Register

When an ECC error is generated in the FlexRay RAM, the corresponding bit in this register is set to 1 or the current setting is cleared. Writing 1 to a bit in this register clears the bit.

Access This register can be read/written in 32-bit units.

The value read from E7AxCTL.E7AxCTLER2C (x = 0-6) is always 0.

Address FF46 4000$_H$ (E7A0CTL)

FF46 4100$_H$ (E7A1CTL)

FF46 4200$_H$ (E7A2CTL)

FF46 4300$_H$ (E7A3CTL)

FF46 4400$_H$ (E7A4CTL)

FF46 4500$_H$ (E7A5CTL)

FF46 4600$_H$ (E7A6CTL)

Initial value Undefined

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	E7AxCTLER2C	0	0	0	0	0	0	0	E7AxCTLCCER2	0/1	0/1
R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

Table 9-27 Contents of the E7AxCTL Register (x = 0 to 6)

Bit Position	Bit Name	Function
10	E7AxCTLER2C (x = 0 to 6)	0: The E7AxCTL.E7AxCTLECCER2 (x = 0 to 6) bits are not cleared. 1: The E7AxCTL.E7AxCTLECCER2 (x = 0 to 6) bits are cleared.
2	E7AxCTLCCER2 (x = 0 to 6)	0: An ECC error in the FlexRay RAM has not been generated. 1: An ECC error in the FlexRay RAM has been generated.

9.7 Resources Required for Initialization

9.7.1 On-chip RAM

Initial values in on-chip RAM are undefined, so reading it without initialization raises the possibility of an ECC error. Accordingly, we recommend initializing the whole on-chip RAM (with any desired values).

Caution After an uninitialized address is written to, if the same address is read following the write, an ECC error may occur.

9.7.2 Registers

Initial values for the registers listed below are undefined, so reading them without initialization raises the possibility of a CPU comparison error. Accordingly, we recommend initializing all of these registers (with any desired values).

When initializing the specified registers only according to the conditions, evaluate them enough before using.

Table 9-28 List of Registers to be Initialized (Program Registers)

Name	Function
r1	Assembler-reserved register
r2	Address- and data-variables register
r3	Stack pointer (SP)
r4	Global pointer (GP)
r5	Text pointer (TP)
r6 to r29	Address- and data-variables register
r30	Element pointer (EP)
r31	Link pointer (LP)

Caution In some cases these registers will be implicitly used by the C compiler, etc., so use the start-up routine to initialize these registers.

Table 9-29 List of Registers to be Initialized (System Registers)

Group Name	Bank Name	Number of System Register	Name	Function
CPU group	Base bank	0	EIPC	This register preserves the state on acceptance of an EI level exception.
		2	FEPC	This register preserves the state on acceptance of an FE level exception.
		11	SCCFG	Settings for SYSCAL operations
		12	SCBP	Base pointer for SYSCALL
		16	CTPC	This register preserves the state on execution of a CALLT instruction.
		20	CTBP	Base pointer for CALLT
		28*	EIWR	Working register for use with EI levels
		29*	FEWR	Working register for use with FE levels
FPU group	FPU status bank	6	FPSR	Settings and states for floating-point calculations
		7	FPEPC	Program counter at the time of a floating-point exception
		8	FPST	Floating-point state
		9	FPCC	Result of comparison in floating-point calculation
CPU core protection group	CPU core protection violations	1	VSTID	Task identifier for system-register protection violations
		2	VSADR	Address of system-register protection violations
		5	VMTID	Task identifier for memory-protection violations
		6	VMADR	Address of memory-protection violations
		24	MCA	Address for checking of memory-protection settings
		25	MCS	Size for checking of memory-protection settings

Note * The above system registers are always accessible regardless of the state of the BSELL register.

Table 9-30 List of Registers to be Initialized (CPU core Protection Related Registers)

Address	Name	Function
$FFFF5110_H$	VPNECR	Peripheral device protection NT state violation cause register
$FFFF5110_H$	VPNECRL	Peripheral device protection NT state violation cause register L
$FFFF5110_H$	VPNECRLL	Peripheral device protection NT state violation cause register LL
$FFFF5111_H$	VPNECRLH	Peripheral device protection NT state violation cause register LH
$FFFF5114_H$	VPNADR	Peripheral device protection NT state violation address register
$FFFF5114_H$	VPNADRL	Peripheral device protection NT state violation address register L
$FFFF5114_H$	VPNADRLL	Peripheral device protection NT state violation address register LL
$FFFF5115_H$	VPNADRLH	Peripheral device protection NT state violation address register LH
$FFFF5116_H$	VPNADRH	Peripheral device protection NT state violation address register H
$FFFF5116_H$	VPNADRHL	Peripheral device protection NT state violation address register HL
$FFFF5117_H$	VPNADRHH	Peripheral device protection NT state violation address register HH
$FFFF5118_H$	VPNTID	Peripheral device protection NT state violation task ID register
$FFFF5118_H$	VPNTIDL	Peripheral device protection NT state violation task ID register L
$FFFF5118_H$	VPNTIDLL	Peripheral device protection NT state violation task ID register LL
$FFFF5119_H$	VPNTIDLH	Peripheral device protection NT state violation task ID register LH
$FFFF511A_H$	VPNTIDH	Peripheral device protection NT state violation task ID register H
$FFFF511A_H$	VPNTIDHL	Peripheral device protection NT state violation task ID register HL
$FFFF511B_H$	VPNTIDHH	Peripheral device protection NT state violation task ID register HH
$FFFF5120_H$	VPTECR	Peripheral device protection T state violation cause register
$FFFF5120_H$	VPTECRL	Peripheral device protection T state violation cause register L
$FFFF5120_H$	VPTECRLL	Peripheral device protection T state violation cause register LL
$FFFF5121_H$	VPTECRLH	Peripheral device protection T state violation cause register LH
$FFFF5124_H$	VPTADR	Peripheral device protection T state violation address register
$FFFF5124_H$	VPTADRL	Peripheral device protection T state violation address register L
$FFFF5124_H$	VPTADRLL	Peripheral device protection T state violation address register LL
$FFFF5125_H$	VPTADRLH	Peripheral device protection T state violation address register LH
$FFFF5126_H$	VPTADRH	Peripheral device protection T state violation address register H
$FFFF5126_H$	VPTADRHL	Peripheral device protection T state violation address register HL
$FFFF5127_H$	VPTADRHH	Peripheral device protection T state violation address register HH
$FFFF5128_H$	VPTTID	Peripheral device protection T state violation task ID register
$FFFF5128_H$	VPTTIDL	Peripheral device protection T state violation task ID register L
$FFFF5128_H$	VPTTIDLL	Peripheral device protection T state violation task ID register LL
$FFFF5129_H$	VPTTIDLH	Peripheral device protection T state violation task ID register LH
$FFFF512A_H$	VPTTIDH	Peripheral device protection T state violation task ID register H
$FFFF512A_H$	VPTTIDHL	Peripheral device protection T state violation task ID register HL
$FFFF512B_H$	VPTTIDHH	Peripheral device protection T state violation task ID register HH

Table 9-31 List of Registers to be Initialized (Registers Related to the Timing Supervisor)

Address	Name	Function
$FFFF5024_H$	TSCCNT0	Count value of timing supervision counter n register 0
$FFFF5028_H$	TSCCMP0	Comparison value of timing supervision counter n register 0
$FFFF502C_H$	TSCRLD0	Reload value of timing supervision counter n register 0
$FFFF5034_H$	TSCCNT1	Count value of timing supervision counter n register 1
$FFFF5038_H$	TSCCMP1	Comparison value of timing supervision counter n register 1
$FFFF503C_H$	TSCRLD1	Reload value of timing supervision counter n register 1
$FFFF5044_H$	TSCCNT2	Count value of timing supervision counter n register 2
$FFFF5048_H$	TSCCMP2	Comparison value of timing supervision counter n register 2
$FFFF504C_H$	TSCRLD2	Reload value of timing supervision counter n register 2
$FFFF5054_H$	TSCCNT3	Count value of timing supervision counter n register 3
$FFFF5058_H$	TSCCMP3	Comparison value of timing supervision counter n register 3
$FFFF505C_H$	TSCRLD3	Reload value of timing supervision counter n register 3
$FFFF5064_H$	TSCCNT4	Count value of timing supervision counter n register 4
$FFFF5068_H$	TSCCMP4	Comparison value of timing supervision counter n register 4
$FFFF506C_H$	TSCRLD4	Reload value of timing supervision counter n register 4
$FFFF5074_H$	TSCCNT5	Count value of timing supervision counter n register 5
$FFFF5078_H$	TSCCMP5	Comparison value of timing supervision counter n register 5
$FFFF507C_H$	TSCRLD5	Reload value of timing supervision counter n register 5

Section 10 Safety Guardian (SGA)

This section gives an overview of the SGA.

10.1 SGA Features

Number of instances

This product has a single SGA.

Table 10-1 Instances of SGA

SGA	
Instances	1
Names	SGA (common area), SGAM, SGAC

Meaning of m in names

Throughout this section, the SGAM and SGAC are collectively referred to by attaching the index "m" (m = M and C). The individual values (M and C) identify the respective areas. That is,

- SGAmESET indicates both SGAMESET and SGACESET.

Meaning of xx in signal names

Throughout this section, the index "xx" (xx = 0 to 47) attached to the common part of the signal name collectively indicates SGATERRIN0 to SGATERRIN47. The individual values (0 to 47) identify the respective signals.

Register addresses

All SGA register addresses are given as address offsets from the individual base addresses <SGA_base>, <SGAM_base>, or <SGAC_base>.
The addresses of each area are listed in the following table.

Table 10-2 SGA Register Base Addresses

SGA Register	Address Label	Address
SGAM	<SGAM_base>	FF83 $C000_H$
SGAC	<SGAC_base>	FF83 $D000_H$
SGA	<SGA_base>	FF83 $E000_H$

Clock supply All SGA provide one clock input.

Table 10-3 SGA Clock Supply

SGA Area	SGA Clock	Connected to
SGA, SGAM, SGAC	PCLK	Clock controller

Interrupts The SGA can generate the following interrupt requests.

Table 10-4 SGA Interrupt Requests

SGA Signal	Function	Connected to
SGATI	SGA interrupt	Interrupt controller INTISG

I/O signals The error input signals of the SGA are listed in the following table.

Table 10-5 SGA Error Input Signals (1/3)

SGA Signal	Function	Connected to
SGATERRIN0	CLMA0 error input	OSC_ERR
SGATERRIN1	CLMA1 error input	PLL_ERR
SGATERRIN2	CLMA2 error input	ROSC_ERR
SGATERRIN3	WDTA error input	WDTA_ERR
SGATERRIN4	LVI reset input	LVI_RES
SGATERRIN5	Compare unit error input	CMP_ERR
SGATERRIN6	Flash memory ECC error detection	FLASH_ED
SGATERRIN7	Flash memory ECC error detection (and correction)	FLASH_EC
SGATERRIN8	RAM ECC error detection	IRAM_ED
SGATERRIN9	RAM ECC error detection (and correction)	IRAM_EC
SGATERRIN10	Flash cache RAM ECC error detection	FCACHE_RAM_ED2
SGATERRIN11	Flash cache RAM ECC error detection	FCACHE_RAM_ED1
SGATERRIN12	Flash cache TAG ECC error detection	FCACHE_TAG_ED2
SGATERRIN13	Flash cache TAG ECC error detection	FCACHE_TAG_ED1
SGATERRIN14	CAN ECC error detection	CAN_ED
SGATERRIN15*[1]	FlexRay ECC error detection	FR_ED
SGATERRIN16*[2]	EDC-based detection of errors in data on the peripheral bus EDCs (error-detection codes) are used in monitoring data transfer to and from the A/D converter and TSG2 module.	APB_DATA_EDC
SGATERRIN17	Reserved	Reserved

Table 10-5 SGA Error Input Signals (2/3)

SGA Signal	Function	Connected to
SGATERRIN18	Unintentional enabling of self-diagnostic BIST (for the target self-diagnostic BIST groups indicated below) µPD70F3505A, µPD70F3506, µPD70F3507: Master group µPD70F3508, µPD70F3509: All BIST group	µPD70F3505A, µPD70F3506, µPD70F3507: ET_MASTER_ENABLE µPD70F3508, µPD70F3509: ET_BIST_ENABLE*3
SGATERRIN19	Unintentional enabling of self-diagnostic BIST (for the target self-diagnostic BIST groups indicated below) µPD70F3505A, µPD70F3506, µPD70F3507: Checker group µPD70F3508, µPD70F3509: All BIST group	µPD70F3505A, µPD70F3506, µPD70F3507: ET_CHECKER_ENABLE µPD70F3508, µPD70F3509: ET_BIST_ENABLE*3
SGATERRIN [20...31]	Reserved	Reserved
SGATERRIN32	Unintentional activation of code flash programming mode	FRB_PRG_ENABLE
SGATERRIN33	Unintentional prohibition of CLMA0	CLM0_DISABLE
SGATERRIN34	Unintentional prohibition of CLMA1	CLM1_DISABLE
SGATERRIN35	Unintentional prohibition of CLMA2	CLM2_DISABLE
SGATERRIN36	Unintentional enabling of self-diagnostic BIST (for the target self-diagnostic BIST groups indicated below) µPD70F3505A, µPD70F3506, µPD70F3507: PBUS1 group µPD70F3508, µPD70F3509: All BIST groups	µPD70F3505A, µPD70F3506, µPD70F3507: ET_APB1_ENABLE µPD70F3508, µPD70F3509: ET_BIST_ENABLE*3
SGATERRIN37*1	Unintentional enabling of self-diagnostic BIST (for the target self-diagnostic BIST groups indicated below) µPD70F3506, µPD70F3507: FlexRay µPD70F3508, µPD70F3509: All BIST group	µPD70F3506, µPD70F3507: ET_APB2_ENABLE µPD70F3508, µPD70F3509: ET_BIST_ENABLE*3
SGATERRIN38	Unintentional enabling of production test mode	TEST_ENABLE
SGATERRIN39	CPU mode error: This error is generated on detection of an unintentional change to the CPU operating mode.	CPU_MODE_ERR
SGATERRIN40	SGA compare error, internal loop back of the SGA error output. Inconsistency of the redundant SGA is indicated by this error.	SG_CMP_ERR

Table 10-5 SGA Error Input Signals (3/3)

SGA Signal	Function	Connected to
SGATERRIN41	Unintentional masking of an error signal: Since the BIST block toggles error signals, error generation during BIST execution is masked in accord with this signal. This signal also indicates whether or not error generation is masked beyond BIST execution.	SG_HW_BIST_ERR_EN
SGATERRIN [42...47]	Reserved	Reserved
SGATERRSWS	This signal indicates the state of writing to SGAmESET.	SGA internal connection
$\overline{\text{SGATERRLB}}$	Loop-back signal from the signal on the ERROROUT pin (i.e. this signal indicates the level on the ERROROUT pin).	ERROROUT pin

Note 1. The μPD70F3505A does not have this signal.
Note 2. The μPD70F3508 and μPD70F3509 do not have this signal.
Note 3. In the μPD70F3508 and μPD70F3509, the error signal from unintentionally enabling self-diagnostic BIST is connected in common to these SGA error inputs: SGATERRIN18, 19, 36, and 37.

I/O signals The I/O signals of the SGA are listed in the following table.

Table 10-6 SGA Input Signals

SGA Signal	Function	Connected to
SGATERROUTZ	SGA error output used for signaling of errors to an external watchdog.	ERROROUT pin
$\overline{\text{SGARES}}$	SGA internal reset request can force an internal reset of the device.	Reset controller
$\overline{\text{SGATRERRO}}$	SGA error output for Hi-Z control of TAPA. In case of an error, signaled by SGATERROUTZ, a Hi-Z state can automatically be enabled for the motor control timer outputs.	PIC (ERROROUT)
SGATTIN	SGA timer input. The dynamic behavior of the SGATERROUTZ can be configured by this timer input.	TAUA1 ch15 OSTM1

10.2 SGA Functional Overview

The safety guardian (SGA) collects all the internal error signals coming from the different error sources and monitoring circuits. These errors are for instance generated by the clock monitor, ECC circuit, compare unit, and other monitoring units.

Features summary

The SGA has the following features.

- Error collection, status indication, and individual treatment
  - Error generation via SGATERROUTZ
  - Interrupt generation via SGATI
  - Internal reset generation via $\overline{\text{SGARES}}$

The SGA is implemented redundant. Some of the redundant registers are accessed simultaneously by one write operation and some registers have to be accessed individually. This allows an individual functional test by software of the two safety guardian instances.

- SGA (safety guardian) register area represents the common register area where the redundant registers are accessed simultaneously by one write operation.
- SGAM (safety guardian master) register area represents a register area where the SGA master registers are accessed individually.
- SGAC (safety guardian checker) register area represents a register area where the SGA checker registers are accessed individually.

The following block diagram shows the mapping of the SGA register areas.

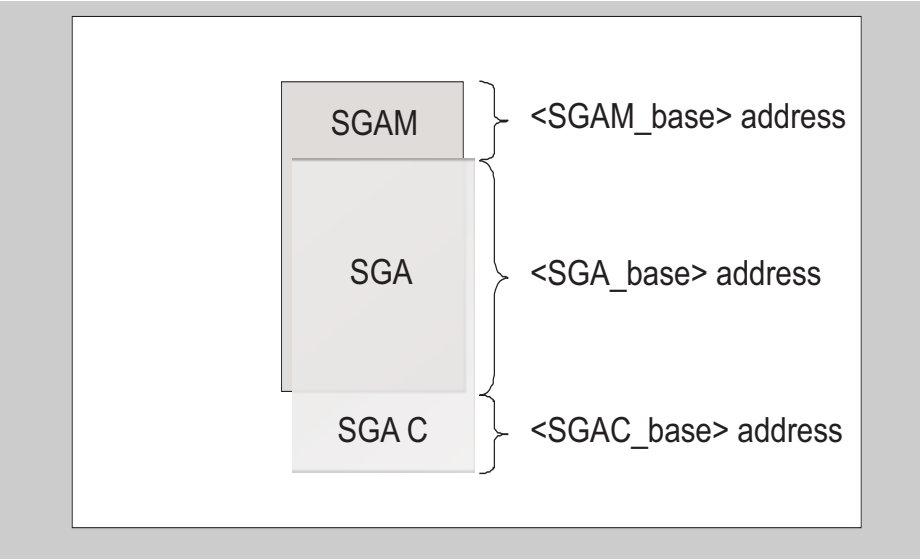


Figure 10-1 Register Areas of the Redundant SGA

The following block diagram shows the main components of the SGA.

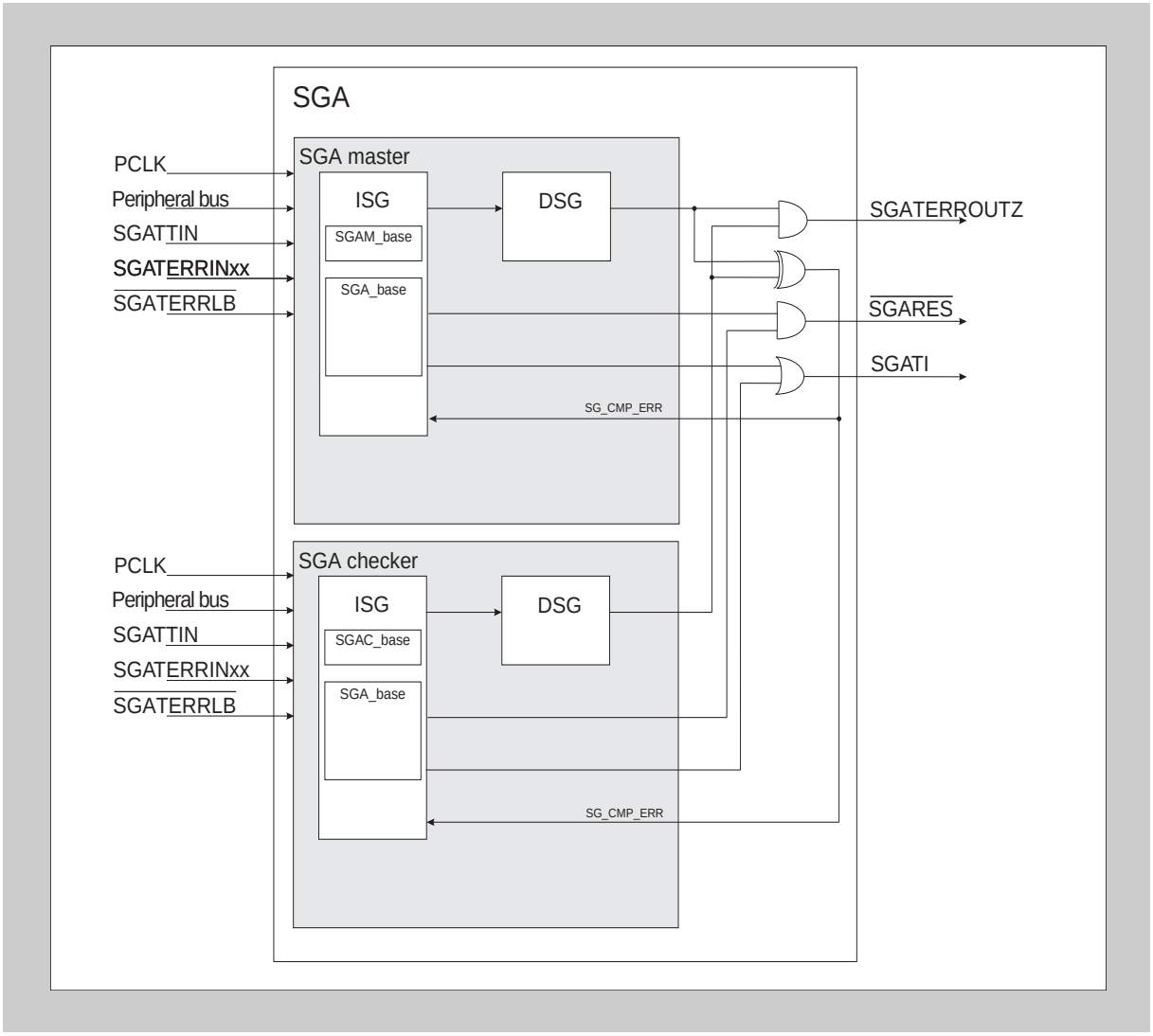


Figure 10-2 Block Diagram of the Redundant SGA

The SGA is divided into the indirect safety guardian (ISG) and direct safety guardian (DSG).

- ISG is used for static configuration of the error sources.
- DSG consists of combinatorial logic only. Therefore, it is robust against clock faults and able to indicate an error towards SGATERROUTZ even in case of clock faults.

The SGA has the following external connections.

- Inputs
  - Error inputs SGATERRINxx from all internal error sources and monitoring units. The SGA master and SGA checker have identical error input signals.
  - The SGATTIN timer input connection is for generating dynamic error operation.
  - Error loop-back $\overline{\text{SGATERRLB}}$ is used to read back the error output level from the ERROROUT pin of the device.
- Outputs
  - Interrupt output SGATI
  - Internal reset output $\overline{\text{SGARES}}$
  - Error output SGATERROUTZ

10.3 Functional Description

10.3.1 SGA Operating States

The SGA operation can be sub-divided into four main phases.

- Reset: Hardware initialization of the device.
- Start-up test: Execution/Evaluation of self-diagnostic BIST and software self-test
- Application: Configuration of SGA and execution of user application
- Error: Error handling according to SGA configuration

The following block diagram shows an example of the different phases of operation.

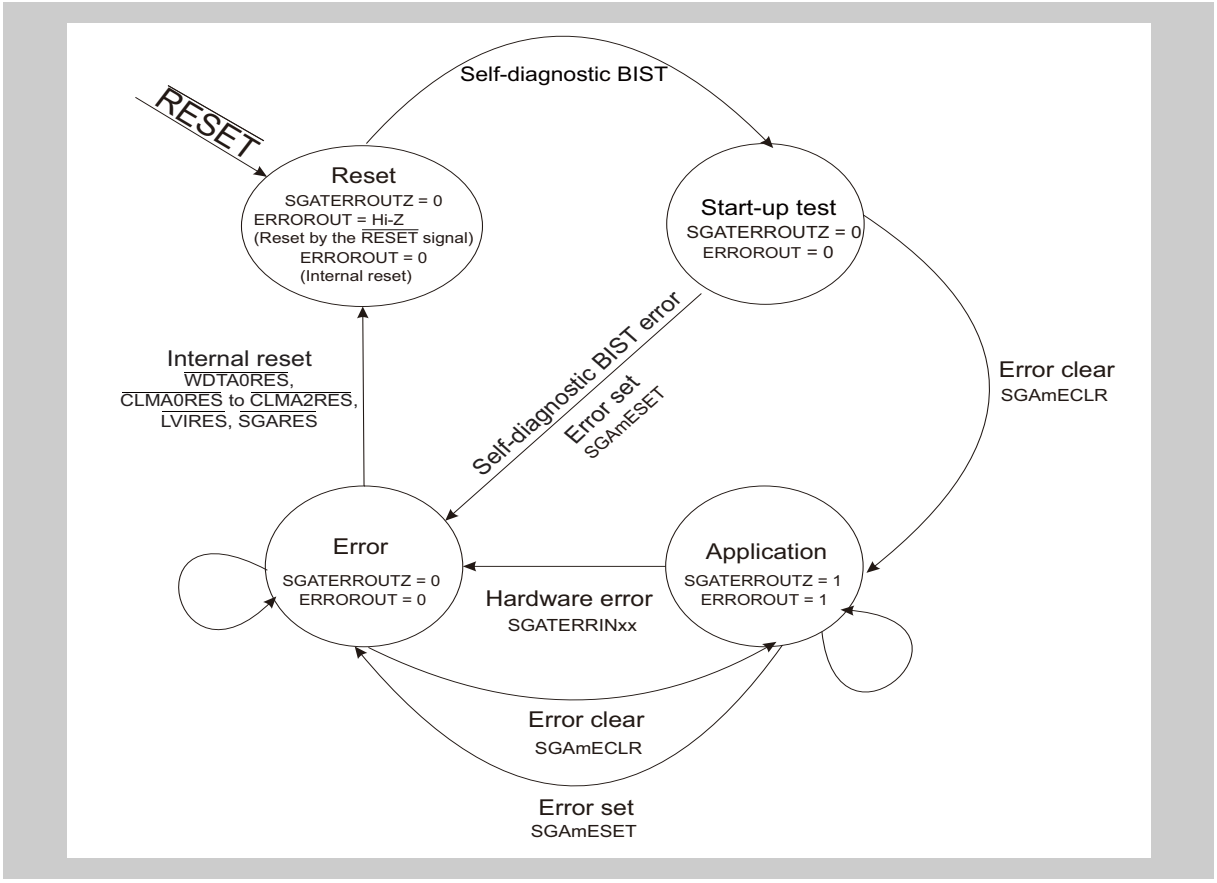


Figure 10-3 SGA Error States and Phases

10.3.2 SGA Configuration Overview

The SGA supports the following functions.

- Error set/clear register
  - The error set/clear register is used to enter or release the active or inactive error state. Where active (error set) means that the error output SGATERROUTZ is low independent from the error inputs SGATERRINxx. Inactive (error clear) means that the error output SGATERROUTZ is high (or toggled by timer input) until an error is indicated by SGATERRINxx.
- Error source status registers
  - Represent the status of the individual error source input SGATERRINxx.
- Error source status clear registers
  - Are used to clear the individual error status.
- Error mask registers
  - Each individual error input SGATERRINxx can be masked towards the error output SGATERROUZ.
- Internal reset configuration registers
  - Each individual error input SGATERRINxx can be configured to generate an internal reset or not.
- Interrupt configuration register
  - Interrupt generation by SGATI can be configured for some error inputs SGATERRIN41 to SGATERRIN32.
- Pseudo error registers
  - This function allows functional tests of the error path by pseudo error injection.
- Error pulse configuration register
  - The SGATERROUTZ error-output signal can be configured for dynamic or non-dynamic operation. In dynamic mode, the error-output signal operates in accord with the level on SGATTIN.

The register and signal dependency is shown below.

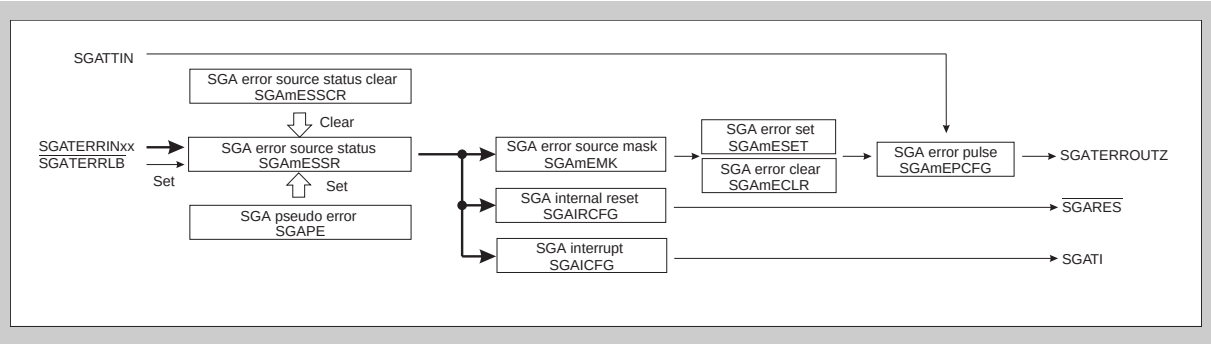


Figure 10-4 SGA Register and Signal Dependency

10.3.3 Operations for Error Output

The error output SGATERROUZ is connected to the ERROROUT pin.

The error output can be configured for two different modes of operation, non-dynamic or dynamic.

Table 10-7 Error Output Operating Modes

Error Status SGAmESSTRy.SGAmSSE019-000/ SGAmSSE131-100 (y = 0, 1)	Operating Mode SGASL0 Bit	Error Output Operating Mode	Error Output Level	Error Status
0	0	Non-dynamic	High	No error
	1	Dynamic	Toggles (according to SGATTIN timer configuration)	No error
1	0	Non-dynamic	Low	Error
	1	Dynamic	Low	Error

Caution The following programming procedure should be considered.

(1) Dynamic Mode Enable

1) Set the SGAEMK0 and SGAEMK1 registers.
2) Initialize the related timer (connected to SGATTIN).
3) Drive the error output to high level by setting the SGAmECLR.SGAmECT bit to 1. Refer to the note in section 10.4.2 (2), SGAmECLR - SGAm Error Clear Trigger Register.
4) Set the SGAEPCFG.SGASL0 bit to 1 for dynamic mode.
5) Start the timer.

Note Determine the setting for the timer cycle according to the characteristics of the external device to be connected.

(2) Dynamic Mode Disable

1) Set the SGAEMK0 and SGAEMK1 registers.
2) Drive the error output to low level by setting the SGAmESET.SGAmEST bit to 1. Refer to the note in section 10.4.2 (1), SGAmESET SGAm Error Set Trigger Register.
3) Stop the timer.
4) Set the SGAEPCFG.SGASL0 bit to 0 for non-dynamic mode.

10.3.4 Loop-Back Function

The SGA supports a loop-back function which is used to check the error path from the SGA to the ERROROUT pin. The SGATERRLB is connected to the ERROROUT pin and the level of the output is reflected to the SGAmSSE131 bit, shown in table 10-14, SGAmESSTR1 Register Contents.

If a loop-back test is to be run, do so while all error flags that are not masked by settings in SGAEMK0/1 are clear.

(1) Example of a Loop-Back Test after Reset

1) After any reset, the error output has a “low level”.
2) Read the SGAMESSTR1.SGAMSSE131 and SGACESSTR1.SGACSSE131 bits and check for “0”.
3) Clear all internal error sources by the SGAESSTC0 and SGAESSTC1 registers.
4) Mask SG_CMP_ERR by setting the SGAEMK1.SGAEMK108 bit.
5) Set error output to “high level” by the SGAMECLR and SGACECLR registers.
6) Read the SGAMESSTR1.SGAMSSE131 and SGACESSTR1.SGACSSE131 bits and check for “1”.
7) Set error output to “low level” by the SGAMESET register.
8) Read the SGAMESSTR1.SGAMSSE131 and SGACESSTR1.SGACSSE131 bits and check for “0”
9) Clear the error by the SGAESSTC1.SGACLSSE130 bit.
10) Set error output to “high level” by the SGAMECLR register.
11) Read the SGAMESSTR1.SGAMSSE131 and SGACESSTR1.SGACSSE131 bits and check for “1”.
12) Set error output to “low level” by the SGACESET register.
13) Read the SGAMESSTR1.SGAMSSE131 and SGACESSTR1.SGACSSE131 bits and check for “0”.
14) Set error output to “low level” by the SGAMESET and SGACESET registers.
15) Clear the error by the SGAESSTC1.SGACLSSE108 and SGAESSTC1.SGACLSSE130 bits.
16) Unmask SG_CMP_ERR by clearing the SGAEMK1.SGAEMK108 bit.

Refer to the note in section 10.4.2 (1), SGAmESET SGAm Error Set Trigger Register.

10.3.5 Pseudo Error Generation

The SGA supports a pseudo error injection function to “emulate” an error for test or debug purposes. The operation of the SGA during injection of pseudo errors is identical to the occurrence of real errors. This means that all configurations for error masks, interrupt, or internal reset apply in the same way.

The pseudo error generation can be used:

- for debug purposes to generate errors and check for instance the operation of the software for
  - SGA interrupts (SGATI)
  - Internal reset ($\overline{\text{SGARES}}$)
  - error output (SGATERROUTZ)
- for test purposes after reset to perform a functional test of the error source status registers SGAMESSTR0 and SGAMESSTR1. This test can be combined with the loop-back function.

A pseudo-error can be generated by setting the error source bit (SGAPE100 to SGAPE108 or SGAPE000 to SGAPE019) for the given source in the SGAPE0 or SGAPE1 register to 1.

10.3.6 Error Status

The error status is indicated by the SGAmESSTR0 and SGAmESSTR1 registers. The error status is only cleared after external reset by RESET. In case of an internal reset, the status is kept and the error source can be evaluated afterwards by reading the SGAmESSTR0 and SGAmESSTR1 registers. Additionally, these registers are excluded from the self-diagnostic BIST in order to not destroy their content. As a result, the content of these registers is generated by a majority voter based on the content of the SGAmESSTR0n and SGAmESSTR1n registers.

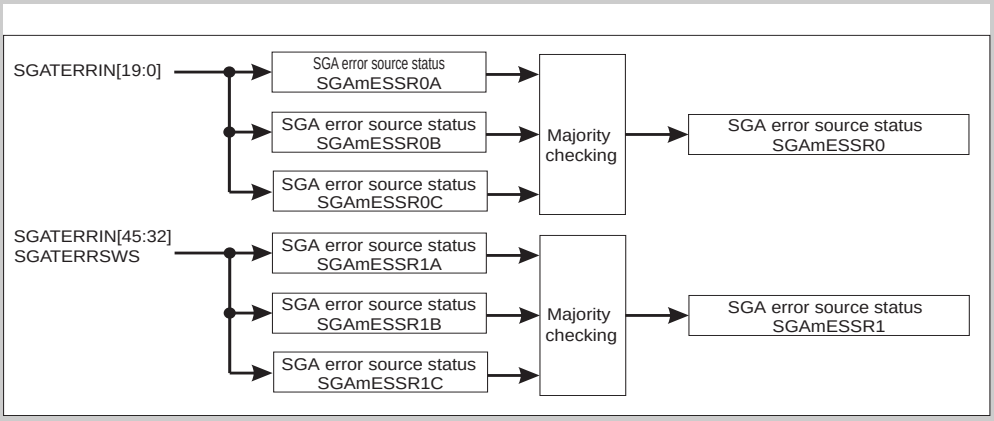


Figure 10-5 SGA Majority Checking Registers

10.3.7 Writing to Protected Registers

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc. An overview of the related registers is shown in table 10-10, Overview of SGA Registers.

(1) Protection Unlock Sequence

Write access to a write protected register is only possible within a special protection unlock sequence.

1. Write the fixed value 0000 $00A5_H$ to the protection command registers SGAPCMD1 and SGAmPCMD0.
2. Write the desired value to the protected registers of SGA and SGAm.
3. Write the bit-wise inversion of desired value to the protected registers of SGA and SGAm.
4. Write the desired value to the protected registers of SGA and SGAm.
5. Check successful write of the desired value to the protected register by checking that SGAPS.SGAPRERR = 0.

In case of any access to another register between step 1 to step 4 of the above sequence, the protection mechanism behaves as follows.

- If the second register belongs to the SGA, the write to the protected register fails (indicated by SGAPS.SGAPRERR = 1). The entire sequence has to be restarted at step 1.
- If the second register does not belong to the SGA, the protection unlock sequence is not disrupted and the write to the first register can be completed successfully.

Note For sequences of SGAPCMD1 and SGAmPCMD0 registers, the status of the protection unlock sequence is commonly indicated by the SGAPS.SGAPRERR flag. Therefore, it is recommended not to intermix protection unlock sequences of SGAPCMD1 and SGAmPCMD0.

In case the protection unlock sequence is interrupted, the protection mechanism behaves as follows.

- Interrupts during protection unlock sequence
  If an interrupt is acknowledged within the above protection unlock sequence and the interrupt service routine does not access any register of the SGA, the protection unlock sequence is not disrupted and the write to the protected register can be successfully completed after returning from the interrupt service routine.

Caution All protected registers as well as the SGA protection command registers SGAPCMD1 and SGAmPCMD0 must be accessed in 32-bit units.

10.4 Registers

This section contains a description of all registers of the SGA.

10.4.1 Overview of SGA Registers

The SGA is controlled and operated by the following registers.

Table 10-8 Overview of SGA Master Registers

Register Name	Symbol	Write Protected by Special Sequence	Address
SGA master error set trigger register	SGAMESET	Enabled	<SGAM_base>
SGA master error clear trigger register	SGAMECLR	Enabled	<SGAM_base> + 4_H
SGA master error source status register 0	SGAMESSTR0	Disabled	<SGAM_base> + 8_H
SGA master error source status register 1	SGAMESSTR1	Disabled	<SGAM_base> + C_H
SGA master protection command register	SGAMPCMD0	Not applicable	<SGAM_base> + 10_H
SGA master error source status register 0A (majority A)	SGAMESSTR0A	Disabled	<SGAM_base> + 20_H
SGA master error source status register 1A (majority A)	SGAMESSTR1A	Disabled	<SGAM_base> + 24_H
SGA master error source status register 0B (majority B)	SGAMESSTR0B	Disabled	<SGAM_base> + 28_H
SGA master error source status register 1B (majority B)	SGAMESSTR1B	Disabled	<SGAM_base> + $2C_H$
SGA master error source status register 0C (majority C)	SGAMESSTR0C	Disabled	<SGAM_base> + 30_H
SGA master error source status register 1C (majority C)	SGAMESSTR1C	Disabled	<SGAM_base> + 34_H

Table 10-9 Overview of SGA Checker Registers

Register Name	Symbol	Write Protected by Special Sequence	Address
SGA checker error set trigger register	SGACESET	Enabled	<SGAC_base>
SGA checker error clear trigger register	SGACECLR	Enabled	<SGAC_base> + 4_H
SGA checker error source status register 0	SGACESSTR0	Disabled	<SGAC_base> + 8_H
SGA checker error source status register 1	SGACESSTR1	Disabled	<SGAC_base> + C_H
SGA checker protection command register	SGACPCMD0	Not applicable	<SGAC_base> + 10_H
SGA checker error source status register 0A (majority A)	SGACESSTR0A	Disabled	<SGAC_base> + 20_H
SGA checker error source status register 1A (majority A)	SGACESSTR1A	Disabled	<SGAC_base> + 24_H
SGA checker error source status register 0B (majority B)	SGACESSTR0B	Disabled	<SGAC_base> + 28_H
SGA checker error source status register 1B (majority B)	SGACESSTR1B	Disabled	<SGAC_base> + $2C_H$
SGA checker error source status register 0C (majority C)	SGACESSTR0C	Disabled	<SGAC_base> + 30_H
SGA checker error source status register 1C (majority C)	SGACESSTR1C	Disabled	<SGAC_base> + 34_H

Table 10-10 Overview of SGA Registers

Register Name	Symbol	Write Protected by Special Sequence	Address
SGA error pulse configuration register	SGAEPCFG	Enabled	<SGA_base>
SGA interrupt configuration register	SGAICFG1	Enabled	<SGA_base> + 4_H
SGA internal reset configuration register 0	SGAIRCFG0	Enabled	<SGA_base> + 8_H
SGA internal reset configuration register 1	SGAIRCFG1	Enabled	<SGA_base> + C_H
SGA SGATERROUTZ mask register 0	SGAEMK0	Enabled	<SGA_base> + 10_H
SGA SGATERROUTZ mask register 1	SGAEMK1	Enabled	<SGA_base> + 14_H
SGA error source status clear register 0	SGAESSTC0	Enabled	<SGA_base> + 18_H
SGA error source status clear register 1	SGAESSTC1	Enabled	<SGA_base> + $1C_H$
SGA protection command register	SGAPCMD1	Not applicable	<SGA_base> + 20_H
SGA protection status register	SGAPS	Not applicable	<SGA_base> + 24_H
SGA pseudo error trigger register 0	SGAPE0	Enabled	<SGA_base> + 28_H
SGA pseudo error trigger register 1	SGAPE1	Enabled	<SGA_base> + $2C_H$
SGA error pulse output control register	SGAEPCTL	Not applicable	FF83 $F020_H$

10.4.2 SGA Registers Details

(1) SGAmESET SGAm Error Set Trigger Register

This register is used to set error output signals to the low (active) level.

The given error output is immediately set to the active level after a value is written to this register. This response is not maskable.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGAm_base>

Initial value 0000 0000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SGAmEST
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-11 SGAmEST Register Contents

Bit position	Bit Name	Function
0	SGAmEST	SGA error set trigger bit 0: Writing 0 to this bit is ignored. 1: Initiate an active state of the SGATERROUTZ (low level).

Caution Setting the error output via the SGAmESET register will generate the error (in the SGAmESSTR1.SGAmSSE108 and SGAmESSTR1.SGAmSSE130 bits).

Therefore, the following has to be set in advance.

1) Set the SGAEMK1.SGAEMK108 bit to “masked”.
2) Prevent the generation of SGATI interrupts by setting the SGAICFG1.SGAIE108 bit to “prohibited”.
3) Prevent generation of the $\overline{\text{SGARES}}$ reset by setting the SGAIRCFG1.SGAIRE108 bit to “prohibited”.
4) Set the error output bit in the SGAmESET register.
5) Clear error flags by setting the SGAESSTC1.SGACLSSE108 and SGAESSTC1.SGACLSSE130 bits.
6) Make the following settings in accord with the condition of usage for the SGATERRIN40 error.
   - If an SGATERROUTZ error is being output, set the SGAEMK1.SGAEMK108 bit to “not masked”.
   - If an SGATI interrupt is being generated, set the SGAICFG1.SGAIE108 bit to “enabled”.
   - If a $\overline{\text{SGARES}}$ reset is being generated, set the SGAIRCFG1.SGAIRE108 bit to “enabled”.

(2) SGAmECLR - SGAm Error Clear Trigger Register

This register is used to deactivate error output (by placing the signal at the high level). If no further errors are pending, the setting is immediately effective for the signal on the error pin.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGAm_base> + 4_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SGAmECT
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-12 SGAmECLR Register Contents

Bit position	Bit Name	Function
0	SGAmECT	SGA error clear trigger bit 0: Writing 0 to this bit is ignored. 1: Initiate an inactive state of the SGATERROUTZ (high level).

Caution 1. Clearing of SGATERROUTZ is only possible if all errors, not masked by SGAEMK0/1, are cleared beforehand.

Caution 2. Clearing the error output via the SGAmECLR register will generate the error (in the SGAmESSTR1.SGAmSSE108 bit).

Therefore, the following has to be set in advance.

1) Set the SGAEMK1.SGAEMK108 bit to “masked”.
2) Prevent the generation of SGATI interrupts by setting the SGAICFG1.SGAIE108 bit to “prohibited”.
3) Prevent generation of the $\overline{\text{SGARES}}$ reset by setting the SGAIRCFG1.SGAIRE108 bit to “prohibited”.
4) Clear the error output bit in the SGAmECLR register.
5) Clear error flags by setting the SGAESSTC1.SGACLSSE108 bit.
6) Make the following settings in accord with the condition of usage for the SGATERRIN40 error.
   - If an SGATERROUTZ error is being output, set the SGAEMK1.SGAEMK108 bit to “not masked”.
   - If an SGATI interrupt is being generated, set the SGAICFG1.SGAIE108 bit to “enabled”.
   - If a $\overline{\text{SGARES}}$ reset is being generated, set the SGAIRCFG1.SGAIRE108 bit to “enabled”.

Caution 3. Once the setting for dynamic mode has been made, we recommend not switching back to non-dynamic mode. This is because doing so raises the possibility of glitches at times of error output. For details, see section 10.3.3, Operations for Error Output.

(3) SGAmESSTR0 - SGAm Error Source Status Register 0

This register shows the status of each error source. The status is indicated regardless of mask settings.

Access This register can be read in 32-bit units.

Address <SGAm_base> + 8_H

Initial value $0000\ 0000_H$

This register is initialized by $\overline{\text{RESET}}$ only.

Note This register is not tested by the self-diagnostic BIST. The content of this register is generated by a majority checking process based on the content of the description in (6) SGAmESSTR0n - SGAm Error Source Status Register 0n.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGAmSSE019	SGAmSSE018	0	SGAmSSE016*
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGAmSSE015	SGAmSSE014	SGAmSSE013	SGAmSSE012	SGAmSSE011	SGAmSSE010	SGAmSSE009	SGAmSSE008	SGAmSSE007	SGAmSSE006	SGAmSSE005	SGAmSSE004	SGAmSSE003	SGAmSSE002	SGAmSSE001	SGAmSSE000
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10-13 SGAmESSTR0 Register Contents

Bit position	Bit Name	Function
19 to 18, 16 to 0	SGAmSSE019 to SGAmSSE018, SGAmSSE016 to SGAmSSE000	SGAmSSE019 to SGAmSSE000 correspond to SGATERRIN19 to SGATERRIN00. 0: SGATERRINxx error not occurred 1: SGATERRINxx error occurred

Note * The µPD70F3508 and µPD70F3509 do not have this function.

Caution 1. The self-diagnostic BIST leads to setting (to 1) of the SGAmESSTR0.SGAmSSE018 and SGAmESSTR0.SGAmSSE019 bits. They should be cleared by software.

Caution 2. The generation of an internal reset may lead to setting of the SGAmESSTR0.SGAmSSE006 bit (indicating detection of an ECC error in flash memory) and the SGAmESSTR0.SGAmSSE007 bit (indicating detection of an ECC error in flash memory, including when the error has been corrected). For reliable detection of ECC errors, set the SGAIRCFG0.SGAIRE006 and SGAIRCFG0.SGAIRE007 bits to suppress internal resetting by the SGA (i.e. setting the bits to 0) and test the error source flags in register CECCER.

(4) SGAmESSTR1 - SGAm Error Source Status Register 1

This register shows the status of each error source. The status is indicated regardless of mask settings.

Access This register can be read in 32-bit units.

Address <SGAm_base> + C_H

Initial value 0000 0000_H

This register is initialized by $\overline{\text{RESET}}$ only.

However, with the μPD70F3505A, μPD70F3506, and μPD70F3507, the SGAmSSE130 bit is initialized by a reset from any source.

Note This register is not tested by the self-diagnostic BIST. The content of this register is generated by a majority checking process based on the content of the description in (7) SGAmESSTR1n - SGAm Error Source Status Register 1n.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SGAmSSE131	SGAmSSE130	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGAmSSE109	SGAmSSE108	SGAmSSE107	SGAmSSE106	SGAmSSE105	SGAmSSE104	SGAmSSE103	SGAmSSE102	SGAmSSE101	SGAmSSE100
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10-14 SGAmESSTR1 Register Contents

Bit position	Bit Name	Function
31	SGAmSSE131	Indicates the error output loop-back state ($\overline{\text{SGATERRLB}}$). 0: Error output (the ERROROUT pin) is low level 1: Error output (the ERROROUT pin) is high level
30	SGAmSSE130	Indicates the SGAmESET write status (SGATERRSWS). 0: No error 1: Error is set by SGAmESET.SGAmEST
9 to 0	SGAmSSE109 to SGAmSSE100	SGAmSSE109 to SGAmSSE100 correspond to SGATERRIN41 to SGATERRIN32. 0: SGATERRINxx error not occurred 1: SGATERRINxx error occurred

Caution Since self-diagnostic BIST leads to setting of the SGAmESSTR1.SGATmSSE104, SGAmESSTR1.SGATmSSE105, and SGAmESSTR1.SGATmSSE109 bits (to 1), software should clear these bits to 0.

(5) SGAmPCMD0 - SGAm (Master/Checker) Protection Command Register

This register is the protection command register for the SGAm (master/checker) register. For a list of protected registers, refer to table 10-10, Overview of SGA Registers.

For details on the register write protection sequence, refer to section 10.3.7, Writing to Protected Registers. The status of the protected write sequence is indicated in the description in (17) SGAPS - SGA Protection Status Register.

Access This register can be written in 32-bit units.

Address <SGAm_base> + 10_H

Initial value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SGAm0REG[7:0]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-15 SGAPmCMD0 Register Contents

Bit position	Bit Name	Function
7 to 0	SGAm0REG[7:0]	Protection commands to enable writing to SGAm registers

(6) SGAmESSTR0n - SGAm Error Source Status Register 0n

The error source status shown in the description (3) SGAmESSTR0 - SGAm Error Source Status Register 0, is generated by a majority checking circuit. The SGAmESSTR0n registers consist of a set of three registers.

The index n stands for one instance of these registers. (n = A, B, or C).

These registers can be used for consistency checking to determine whether majority checking is in effect. The bit position in the SGAmESSTR0n registers is identical to that in the SGAmESSTR0.

Access This register can be read in 32-bit units.

Address <SGAm_base> + 20_H

<SGAm_base> + 28_H

<SGAm_base> + 30_H

Initial value 0000 0000_H

This register is initialized by $\overline{\text{RESET}}$ only.

Note This register is not tested by the self-diagnostic BIST.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGAmSSE019n	SGAmSSE018n	0	SGAmSSE016n*
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGAmSSE015n	SGAmSSE014n	SGAmSSE013n	SGAmSSE012n	SGAmSSE011n	SGAmSSE010n	SGAmSSE009n	SGAmSSE008n	SGAmSSE007n	SGAmSSE006n	SGAmSSE005n	SGAmSSE004n	SGAmSSE003n	SGAmSSE002n	SGAmSSE001n	SGAmSSE000n
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10-16 SGAmESSTR0n Register Contents

Bit position	Bit Name	Function
19 to 18, and 16 to 0	SGAmSSE019n to SGAmSSE018n, and SGAmSSE016n to SGAmSSE000n	SGAmSSE019n to SGAmSSE000n correspond to SGATERRIN19 to SGATERRIN00. 0: SGATERRINxxn error not occurred 1: SGATERRINxxn error occurred

Note * The µPD70F3508 and µPD70F3509 do not have this function.

(7) SGAmESSTR1n - SGAm Error Source Status Register 1n

The error source status shown in the description (4) SGAmESSTR1 - SGAm Error Source Status Register 1, is generated by a majority checking circuit. The SGAmESSTR1n registers consist of a set of three registers.

The index n stands for one instance of these registers. (n = A, B, or C).

These registers can be used for consistency checking to determine whether majority checking is in effect. The bit position in the SGAmESSTR1n registers is identical to that in the SGAmESSTR1.

Access This register can be read in 32-bit units.

Address <SGAm_base> + 24_H

<SGAm_base> + $2C_H$

<SGAm_base> + 34_H

Initial value $0000\ 0000_H$

This register is initialized by $\overline{\text{RESET}}$ only.

Note This register is not tested by the self-diagnostic BIST.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	SGAmSSE130n	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGAmSSE109n	SGAmSSE108n	SGAmSSE107n	SGAmSSE106n	SGAmSSE105n	SGAmSSE104n	SGAmSSE103n	SGAmSSE102n	SGAmSSE101n	SGAmSSE100n
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10-17 SGAmESSTR1n Register Contents

Bit position	Bit Name	Function
30	SGAmSSE130n	Indicates the error set trigger state (SGATERRSWS). 0: No error 1: Error is set by SGAmESET.SGAmEST
9 to 0	SGAmSSE109n to SGAmSSE100n	SGAmSSE109n to SGAmSSE100n correspond to SGATERRIN41 to SGATERRIN32. 0: SGATERRINxx error not occurred 1: SGATERRINxx error occurred

(8) SGAEPCFG - SGA Error Pulse Configuration Register

This register is used to specify the operating mode for the error output signal.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base>

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SGASL0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 10-18 SGAEPCFG Register Contents

Bit position	Bit Name	Function
0	SGASL0	Error output operation setting for SGATERROUTZ 0: Non-dynamic mode 1: Dynamic mode. Dynamic error signals are generated from the timer input signal connected to SGATTIN.

Caution After setting the dynamic mode, it is recommended not to change to non-dynamic mode again, because there is a possibility of a glitch at the error output. For details, see section 10.3.3, Operations for Error Output.

(9) SGAICFG1 - SGA Interrupt Configuration Register 1

This register is used to set generation of the SGATI interrupt. Specifically, the generation of interrupts in response to errors (SGATERRINxx) is selectable.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 4_H

Initial value 0000 0000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGAIE109	SGAIE108	SGAIE107	SGAIE106	SGAIE105	SGAIE104	SGAIE103	SGAIE102	SGAIE101	SGAIE100
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-19 SGAICFG1 Register Contents

Bit position	Bit Name	Function
9 to 0	SGAIE109 to SGAIE100	SGAIE109 to SGAIE100 correspond to the error input signals SGATERRIN41 to SGATERRIN32. SGATI interrupt disable/enable 0: Disabled. SGATI is not generated by the occurrence of SGATERRINxx. 1: Enabled. SGATI is generated by the occurrence of SGATERRINxx.

Caution Setting SGAIE109 to 1 does not generate an interrupt for SGATERRIN41.

(10) SGAIRCFG0 - SGA Internal Reset Configuration Register 0

This register is used to set the generation of internal resets in response to internal errors.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 8_H

Initial value 0000 0000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGAIRE019	SGAIRE018	0	SGAIRE016*
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGAIRE015	SGAIRE014	SGAIRE013	SGAIRE012	SGAIRE011	SGAIRE010	SGAIRE009	SGAIRE008	SGAIRE007	SGAIRE006	SGAIRE005	SGAIRE004	SGAIRE003	SGAIRE002	SGAIRE001	SGAIRE000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-20 SGAIRCFG0 Register Contents

Bit position	Bit Name	Function
19, 18, and 16 to 0	SGAIRE019, SGAIRE018, and SGAIRE016 to SGAIRE000	SGAIRE019 to SGAIRE000 correspond to the error input signals $\overline{\text{SGATERRIN}}$19 to $\overline{\text{SGATERRIN}}$00. $\overline{\text{SGARES}}$ internal reset disable/enable 0: Disabled. $\overline{\text{SGARES}}$ is not generated by the occurrence of $\overline{\text{SGATERRIN}}$xx. 1: Enabled. $\overline{\text{SGARES}}$ is generated by the occurrence of $\overline{\text{SGATERRIN}}$xx.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

Caution The generation of an internal reset may lead to the SGAmESSTR0.SGAmSSE006 bit (flash memory ECC error detection) and the SGAmESSTR0.SGAmSSE007 bit (flash memory ECC error detection, including when the error has been corrected) being set to 1. Accordingly, for the reliable detection of ECC errors, set the SGAIRCFG0.SGAIRE006 and SGAIRCFG0.SGAIRE007 bits to suppress internal resetting by the SGA (i.e. setting the bits to 0) and test the error source flags in register CECCER.

(11) SGAIRCFG1 - SGA Internal Reset Configuration Register 1

This register is used to set the generation of internal resets in response to internal errors.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + C_H

Initial value 0000 0000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGAIRE109	SGAIRE108	SGAIRE107	SGAIRE106	SGAIRE105	SGAIRE104	SGAIRE103	SGAIRE102	SGAIRE101	SGAIRE100
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-21 SGAIRCFG1 Register Contents

Bit position	Bit Name	Function
9 to 0	SGAIRE109 to SGAIRE100	SGAIRE109 to SGAIRE100 correspond to the error input signals SGATERRIN41 to SGATERRIN32. SGARES internal reset disable/enable 0: Disabled. SGARES is not generated by the occurrence of SGATERRINxx. 1: Enabled. SGARES is generated by the occurrence of SGATERRINxx.

(12) SGAEMK0 - SGA Error Mask Register 0

This register is used to mask individual error sources (SGATERRINxx) for the error output signal SGATERROUTZ.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 10_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGAEMK019	SGAEMK018	0	SGAEMK016*
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGAEMK015	SGAEMK014	SGAEMK013	SGAEMK012	SGAEMK011	SGAEMK010	SGAEMK009	SGAEMK008	SGAEMK007	SGAEMK006	SGAEMK005	SGAEMK004	SGAEMK003	SGAEMK002	SGAEMK001	SGAEMK000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-22 SGAEMK0 Register Contents

Bit position	Bit Name	Function
19, 18, and 16 to 0	SGAEMK019, SGAEMK018, and SGAEMK016 to SGAEMK000	SGAEMK019 to SGAEMK000 correspond to the error input signals SGATERRIN19 to SGATERRIN00. SGATERROUTZ error output signal mask setting 0: Not maksed. SGATERROUTZ enters the active or error state by occurrence of SGATERRINxx. 1: Masked. SGATERROUTZ does not enter the active or error state by occurrence of SGATERRINxx.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

(13) SGAEMK1 - SGA Error Mask Register 1

This register is used to mask individual error sources (SGATERRINxx) for the error output signal SGATERROUTZ.

Access This register can be read/written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 14_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGAEMK109	SGAEMK108	SGAEMK107	SGAEMK106	SGAEMK105	SGAEMK104	SGAEMK103	SGAEMK102	SGAEMK101	SGAEMK100
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-23 SGAEMK1 Register Contents

Bit position	Bit Name	Function
9 to 0	SGAEMK109 to SGAEMK100	SGAEMK109 to SGAEMK100 correspond to the error input signals SGATERRIN41 to SGATERRIN32. SGATERROUTZ error output signal mask setting 0: Not maksed. SGATERROUTZ is generated by the occurrence of SGATERRINxx. 1: Masked. SGATERROUTZ is not generated by the occurrence of SGATERRINxx.

Caution Setting SGAEMK109 to 1 does not mask SGATERRIN41.

(14) SGAESSTC0 - SGA Error Source Status Clear Register 0

This register is used to clear the individual error source status of the SGAESSTR0 register. The error status of both SGAM and SGAC is cleared simultaneously.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 18_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGACLSSE019	SGACLSSE018	0	SGACLSSE016*
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGACLSSE015	SGACLSSE014	SGACLSSE013	SGACLSSE012	SGACLSSE011	SGACLSSE010	SGACLSSE009	SGACLSSE008	SGACLSSE007	SGACLSSE006	SGACLSSE005	SGACLSSE004	SGACLSSE003	SGACLSSE002	SGACLSSE001	SGACLSSE000
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-24 SGAmESSTC0 Register Contents

Bit position	Bit Name	Function
19,18, and 16 to 0	SGACLSSE019, SGACLSSE018, and SGACLSSE016 to SGACLSSE000	Controls the status clearing of the SGAmESSTR0.SGASSE019 to SGASSE000 bits. 0: No action. Writing to this bit is ignored. 1: The related error status SGASSE0xx is cleared.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

(15) SGAESSTC1 - SGA Error Source Status Clear Register 1

This register is used to clear the individual error source status of the SGAESSTR1 register. The error status of both SGAM and SGAC is cleared simultaneously.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + $1C_H$

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	SGACLSSE130	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	SGACLSSE109	SGACLSSE108	SGACLSSE107	SGACLSSE106	SGACLSSE105	SGACLSSE104	SGACLSSE103	SGACLSSE102	SGACLSSE101	SGACLSSE100
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-25 SGAmESSTC1 Register Contents

Bit position	Bit Name	Function
30	SGACLSSE130	Controls the status clearing of the SGAESSTR1.SGASSE130 bit. 0: No action. Writing to this bit is ignored. 1: The related error status SGASSE130 is cleared.
9 to 0	SGACLSSE109 to SGACLSSE100	Controls the status clearing of the SGAmESSTR1.SGASSE109 to SGASSE100 bits 0: No action. Writing to this bit is ignored. 1: The related error status SGASSE1 is cleared.

(16) SGAPCMD1 - SGA Protection Command Register

This register is the protection command register for the SGA register. For a list of protected registers, refer to table 10-10, Overview of SGA Registers.

For details about the register write protection sequence, refer to section 10.3.7, Writing to Protected Registers.

Access This register can be written in 32-bit units.

Address <SGA_base> + 20_H

Initial value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SGA1REG[7:0]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-26 SGAPCMD1 Register Contents

Bit position	Bit Name	Function
7 to 0	SGA1REG [7:0]	Protection commands to enable writing to SGA registers

(17) SGAPS - SGA Protection Status Register

This register is used to verify whether the write protected register has been written successfully or not.

For details, refer to section 10.3.7, Writing to Protected Registers.

Access This register can be read in 32-bit units.

Address <SGA_base> + 24_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SGAPRERR
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10-27 SGAPS Register Contents

Bit position	Bit Name	Function
0	SGAPRERR	Indicates whether writing to a write protected register failed or was successful. 0: Writing was successfully completed. 1: Writing failed.

(18) SGAPE0 - SGA Pseudo Error Register 0

This register is used to generate a pseudo error for test purposes. The SGA operation in response to the generation of a pseudo error is identical to that in response to a real error signal on SGATERRINxx.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + 28_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	SGAPE019	SGAPE018	0	SGAPE016*
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGAPE015	SGAPE014	SGAPE013	SGAPE012	SGAPE011	SGAPE010	SGAPE009	SGAPE008	SGAPE007	SGAPE006	SGAPE005	SGAPE004	SGAPE003	SGAPE002	SGAPE001	SGAPE000
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-28 SGAPE0 Register Contents

Bit position	Bit Name	Function
19, 18, and 16 to 0	SGAPE019, SGAPE018, and SGAPE016 to SGAPE000	SGAPE019 to SGAPE000 correspond to SGATERRIN19 to SGATERRIN00. 0: No action. The SGATERRINxx error is not generated. 1: A pseudo error is generated in the same way as SGATERRINxx.

Note * The µPD70F3508 and µPD70F3509 do not have this function.

Caution SGATERRIN18, SGATERRIN19, SGATERRIN36, and SGATERRIN37 cannot be generated simultaneously with other pseudo errors.

(19) SGAPE1 - SGA Pseudo Error Register 1

This register is used to generate a pseudo error for test purposes. The SGA operation in response to the generation of a pseudo error is identical to that in response to a real error signal on SGATERRINxx.

Access This register can be written in 32-bit units. Writing to this register is protected by a sequence of instructions.

Address <SGA_base> + $2C_H$

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SGAPE108	SGAPE107	SGAPE106	SGAPE105	SGAPE104	SGAPE103	SGAPE102	SGAPE101	SGAPE100
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 10-29 SGAPE1 Register Contents

Bit position	Bit Name	Function
8 to 0	SGAPE108 to SGAPE100	SGAPE108 to SGAPE100 correspond to SGATERRIN40 to SGATERRIN32. 0: No action. The SGATERRINxx error is not generated. 1: A pseudo error is generated in the same way as SGATERRINxx.

Caution SGATERRIN18, SGATERRIN19, SGATERRIN36, and SGATERRIN37 cannot be generated simultaneously with other pseudo errors.

(20) SGAEPCTL - SGA Error Pulse Output Control Register

This register is used to select the timer for the output of error signals while the SGA is in dynamic mode.

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

Address FF83 F020$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SGATMSL
R	R	R	R	R	R	R	R/W

Table 10-30 SGAEPCTL Register Contents

Bit position	Bit Name	Function
0	SGATMSL	This bit selects the timer for the output of error signals while the SGA is in dynamic mode. 0: TAUA1 channel 15 is selected. 1: OST1 is selected.

Section 11 Data CRC Function A (DCRA)

This section contains a generic description of the data CRC function A (DCRA).

11.1 DCRA Features

Instances

This product has the following number of instances of the data CRC function A.

Table 11-1 Instances of DCRA

Data CRC Function A	
Instance	2
Name	DCRAn

Instances index n

Throughout this section, the individual instances of a data CRC function A is identified by the index "n" (n = 0, 1), for example, DCRAnCTL for the DCRAn control register.

Register addresses

All DCRAn register addresses are given as address offsets to the individual base address <DCRAn_base0> or <DCRAn_base1>.
The base addresses <DCRAn_base0> and <DCRAn_base1> of each DCRAn are listed in the following table.

Table 11-2 Register Base Addresses <DCRAn_base0> and <DCRAn_base1>

DCRAn	<DCRAn_base0> Address	<DCRAn_base1> Address
DCRA0	FFFF FA00$_H$	FF83 6000$_H$
DCRA1	FFFF FB00$_H$	FF83 7000$_H$

Clock supply

All data CRC function A provide one clock input.

Table 11-3 DCRAn Clock Supply

DCRAn Instance	DCRAn Clock	Connected to
DCRAn	PCLK	Clock generator

11.2 Functional Overview

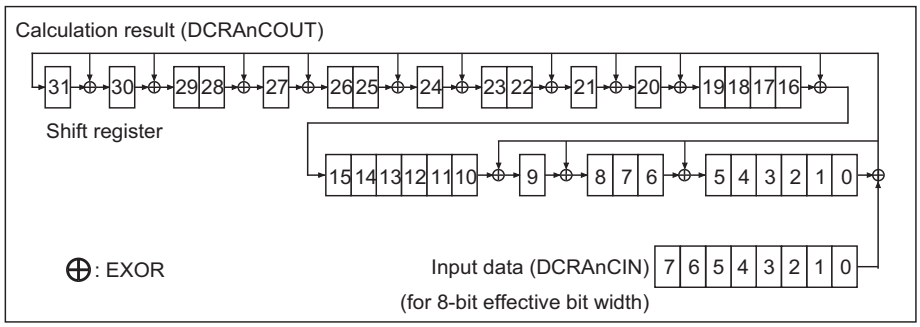
Features summary

The data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC ($04C11DB7_H$)

  $(X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1)$

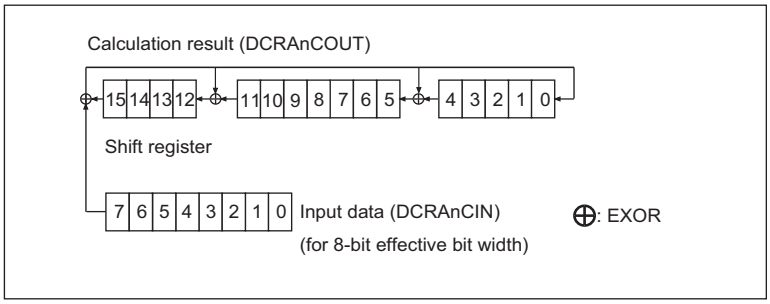
  XOR of the result for the generated CRC code and $FFFFFFFF_H$



- 16-bit CCITT CRC (1021_H)

  $(X^{16} + X^{12} + X^5 + 1)$

  XOR of the result for the generated CRC code and 0000_H



- CRC generation to an arbitrary data block length
- After initialization of the CRC input register, every write access to the CRC input register generates a new CRC according to the chosen polynomial and the result is stored in the CRC data register.

The following figure shows the block diagram of the data CRC function A.

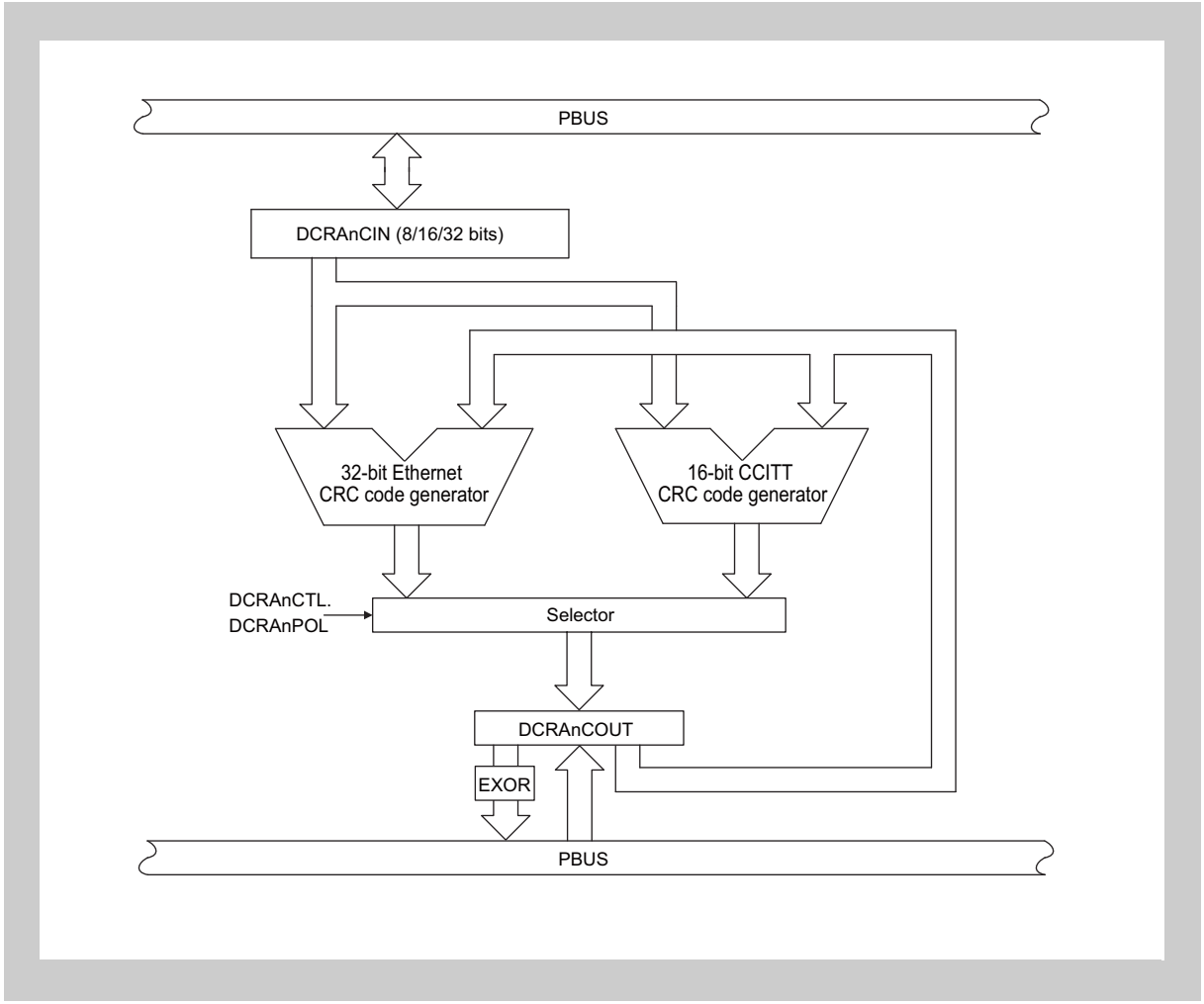


Figure 11-1 Block Diagram of Data CRC Function A

11.3 Functional Description

The data CRC function A generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the data CRC function in 8-, 16-, or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT. The initial starting value must be set at the DCRAnCOUT register before the first write access to the CRC input register (DCRAnCIN) is performed.

The flow chart below shows the CRC generating procedure.

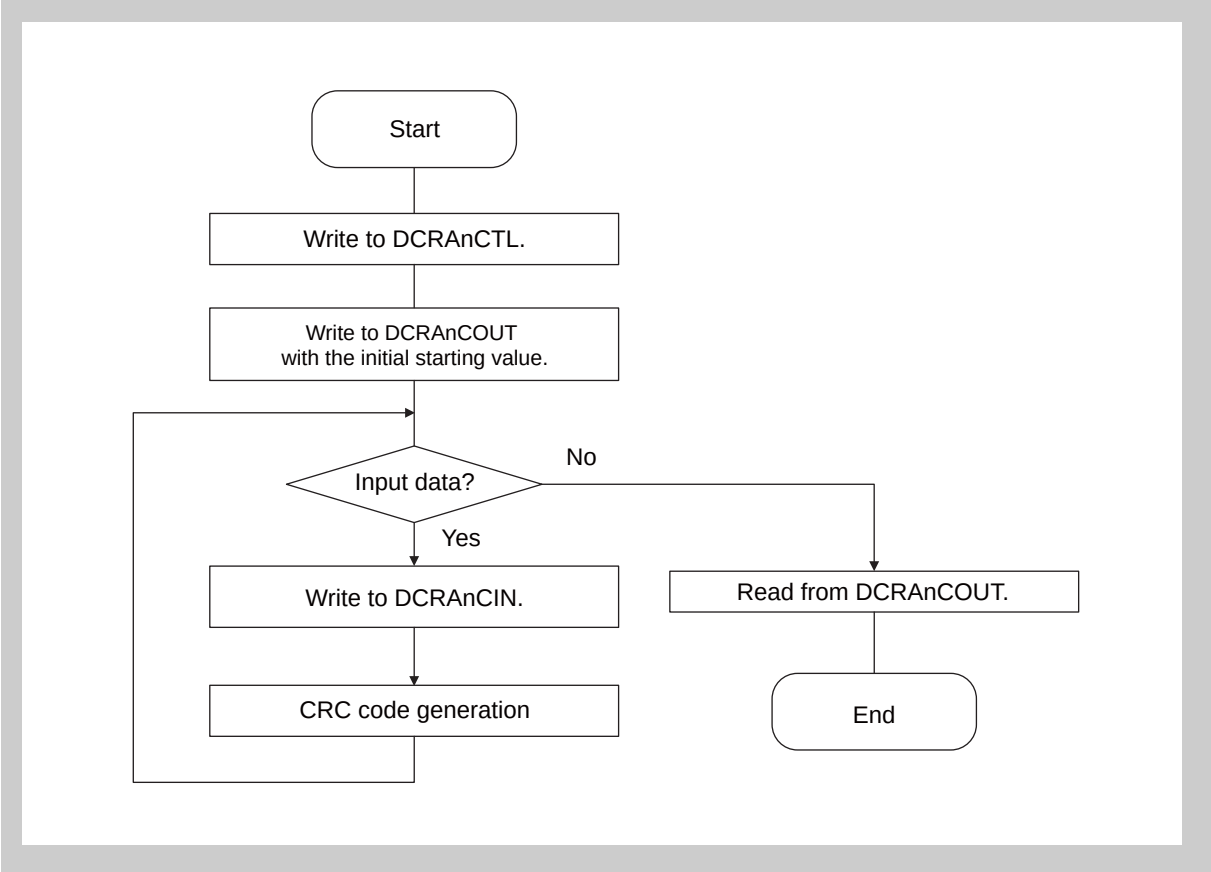


Figure 11-2 Data CRC Function A Flow Diagram

Note For the settings of the individual registers and points for caution regarding the settings, see section 11.4.2, DCRA Registers Details.

11.4 Registers

This section contains a description of all registers of the DCRA.

11.4.1 DCRA Registers Overview

The DCRA is controlled and operated by the following registers.

Table 11-4 DCRA Registers Overview

Register Name	Symbol	Address
DATA-DCRA input register n	DCRAnCIN	<DCRAn_base0> + 00_H
DATA-DCRA data register n	DCRAnCOUT	<DCRAn_base0> + 04_H
DCRA control register n	DCRAnCTL	<DCRAn_base1> + 20_H

11.4.2 DCRA Registers Details

(1) DCRAnCIN — CRC Input Register

This register holds the input data for the CRC calculation. The effective bit width used for CRC calculation must be set by DCRAnCTL.DCRAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAnCIN register is written. The DCRAnCOUT register must be initialized, with the initial starting value, before the first data of the data block is written to the DCRAnCIN register.

Byte order The byte order in DCRAnCIN depends on the selected CRC generating function.

- 32-bit Ethernet CRC polynomial generation (DCRAnCTL.DCRAnPOL = 0)

  The byte order is LSB (least significant byte) first, means LSB at bit position 7...0 of the DCRAnCIN register.

- 16-bit CCITT CRC polynomial generation (DCRAnCTL.DCRAnPOL = 1)

  The byte order is MSB (most significant byte) first, means MSB at bit position 7...0 of the DCRAnCIN register.

Access This register can be read/written in 32-bit units.

Address <DCRAn_base0> + 00_H

Initial value $0000\ 0000_H$

This register is initialized by any reset sources.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCRAnCIN[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCRAnCIN[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-5 DCRAnCIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCIN[31:0]	Input data for CRC calculation. The valid bits are: • For 32-bit effective bit width: DCRAnCIN[31:0] • For 16-bit effective bit width: DCRAnCIN[15:0] • For 8-bit effective bit width: DCRAnCIN[7:0]

(2) DCRAnCOUT — CRC Data Register

This register stores the result of the CRC code generated by the 32-bit Ethernet or 16-bit CCITT polynomial.

Access This register can be read/written in 32-bit units.

Address <DCRAn_base0> + 04_H

Initial value $FFFF\ FFFF_H$

This register is initialized by any reset sources.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCRAnCOUT[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCRAnCOUT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-6 DCRAnCOUT Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCOUT[31:0]	Result of the CRC code generation. When the 16-bit CCITT polynomial is enabled, bits 15 to 0 show the CRC result. The bits 31 to 16 are undefined. On reading these bits, the value read out is the XOR of the value described below and the DCRAnCOUT.DCRAnCOUT bits (this does not affect the DCRAnCOUT.DCRAnCOUT bits). • 32-bit Ethernet polynomial: $FFFF\ FFFF_H$ • 16-bit CCITT CRC polynomial: 0000_H Therefore, for the 32-bit Ethernet polynomial, $0000\ 0000_H$ is read from this register even in the initial state ($FFFF\ FFFF_H$).

Caution This register must be initialized with the starting value before the first data of the data block is written to the DCRAnCIN register.

(3) DCRAnCTL — CRC Control Register

This register controls the CRC generation process.

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

Address <DCRAn_base1> + 20_H

Initial value 00_H

This register is initialized by any reset sources.

7	6	5	4	3	2	1	0
0	0	0	0	0	DCRAnISZ[1:0]		DCRAn POL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-7 DCRAnCTL Register Contents

Bit Position	Bit Name	Function
2, 1	DCRAnISZ[1:0]	Specifies the CRC input bit width. 00: 32 bits (DCRAnCIN[31:0]) 01: 16 bits (DCRAnCIN[15:0]) 10: 8 bits (DCRAnCIN[7:0]) 11: Setting prohibited
0	DCRAnPOL	Specifies the CRC generating function. 0: 32-bit Ethernet CRC polynomial generation The byte order of the DCRAnCIN register is LSB (least significant byte) first, means LSB at bit position 7...0 of the DCRAnCIN register. 1: 16-bit CCITT CRC polynomial generation The byte order of the DCRAnCIN register is MSB (most significant byte) first, means MSB at bit position 7...0 of the DCRAnCIN register.

Note After changing the CRC generating function (DCRAnCTL.DCRAnPOL), the DCRAnCOUT register must be initialized.

Caution The CRC bit width (DCRAnCTL.DCRAnISZn) must be set according to the data block bit width. Switching the CRC bit width is not allowed during processing of a data block (a data block consists of N bytes, half words, or words). After the final CRC result is read from the DCRAnCOUT register, the bit width can be changed. In that case, the register should be set again according to *figure11-2 Data CRC Function A Flow Diagram*.

Section 12 Window Watchdog Timer A (WDTA)

This section contains a generic description of the window watchdog timer A (WDTA).

12.1 WDTA Features

Instances

This product has one instance of the window watchdog timer A.

Table 12-1 Instances of WDTA

Window Watchdog Timer A	
Instances	1
Names	WDTAn

Instances index n

Throughout this section, the individual instances of a window watchdog timer A are identified by the index “n” (n = 0), for example, WDTAnWDTE for the WDTAn enable register.

Register addresses

All WDTAn register addresses are given as address offsets from the individual base address <WDTAn_base>.

The register base address of each WDTAn is listed in the following table.

Table 12-2 Register Base Addresses of WDTAn

WDTAn	<WDTAn_base> Addresses
WDTA0	FF80 6000_H

Clock supply

The window watchdog timer A provides WDTATCKI as the clock input. WDTATCKI is connected to the clock controller.

Table 12-3 Clock Supply of Window Watchdog Timer A

WDTAn	Clock Input Signal: WDTATCKI
WDTA0	Clock controller: SCLK1

Interrupts and reset outputs

The interrupts and reset outputs of the WDTAn are listed in the table below.

Table 12-4 WDTA Interrupts and Reset Outputs

WDTAn Signal	Function	Connected to
WDTA0TRES	WDTA0 error reset	Reset controller: $\overline{\text{WDTA0RES}}$ Safety guardian: SGATERRIN3
WDTA0TNMI	WDTA0 error NMI	Interrupt controller: INTWDTA0NMI Safety guardian: SGATERRIN3
WDTA0TIT	WDTA0 75% interrupt	Interrupt controller: INTWDTA0

12.2 Functional Overview

Features summary

The WDTA has the following functions.

- Operation in response to error detection is selectable.
  - Generation of NMI request (WDTAnTNMI) on error detection
  - Generation of reset (WDTAnTRES) on error detection
- Interrupt request generation at 75% of the counter overflow value
- Window function

The following figure shows the main components of the WDTA.

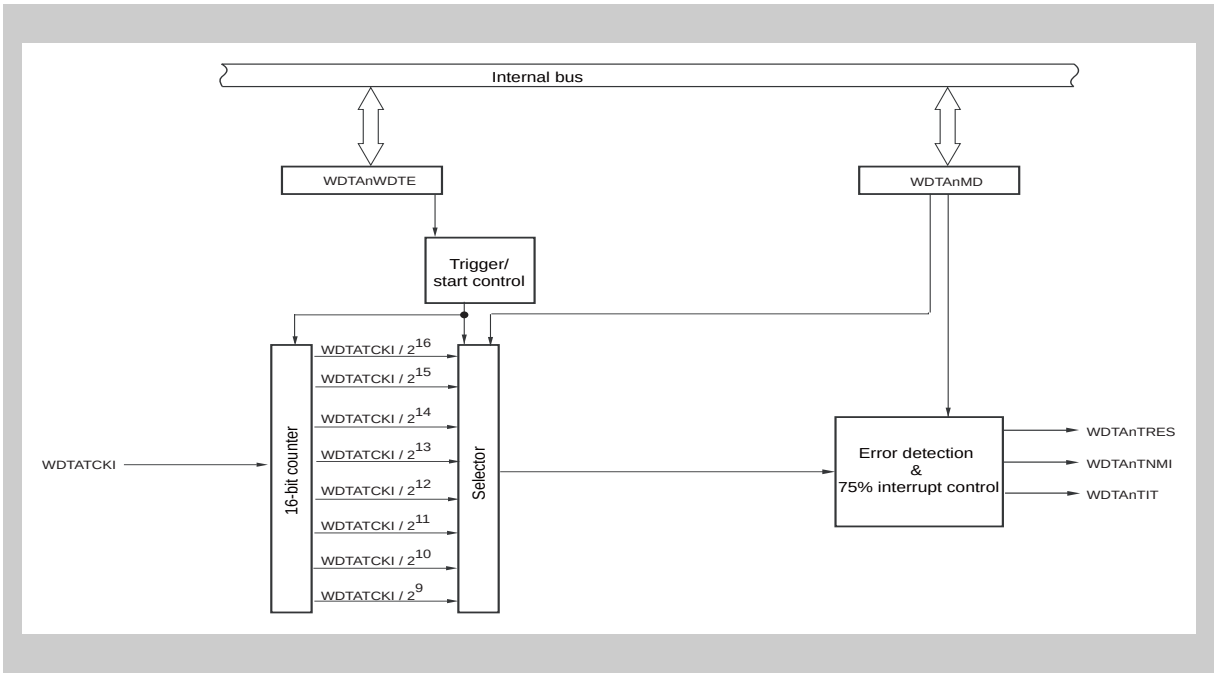


Figure 12-1 Block Diagram of the WDTA

12.3 Functional Description

The WDTA generates a reset or a non-maskable interrupt if the 16-bit counter overflows or if any other error condition is fulfilled. For a description of all error conditions, refer to section 12.3.3, Error Detection.

The counter is cleared and restarted every time a WDTA trigger occurs in the window-open period. Refer to section 12.3.2, WDTA Trigger, and section 12.3.5, Window Function, for details.

At 75% of the counter overflow value, the WDTA can generate an interrupt request (WDTAnTIT). Refer to section section 12.3.4, 75% Interrupt Output, for details.

12.3.1 WDTA after Reset Release

The counter value remains 0000_H after reset release.

The counter is started with the first WDTA trigger.

(1) WDTA Settings after Reset Release

The WDTA settings are as follows after reset release.

Function	Setting	Remarks
Overflow time	2^{16}/WDTATCKI	
75% interrupt mode	Enabled	
Error mode	Reset mode	Any error condition before the first trigger generates a reset.
Window-open period	100%	

Change WDTA settings

After the first trigger, the WDTA continues according to the settings of the watchdog timer mode register (WDTAnMD).

To change the WDTA settings, WDTAnMD must be written before the first trigger. Changing the value of WDTAnMD after the first trigger leads to an error condition.

If the value in WDTAnMD is not changed before the first trigger, the WDTA will operate with the initial value of WDTAnMD.

The settings of WDTAnMD apply after the first trigger.

WDTA start timing The WDTA start timing and the changes to the WDTA settings are illustrated in the following figure.

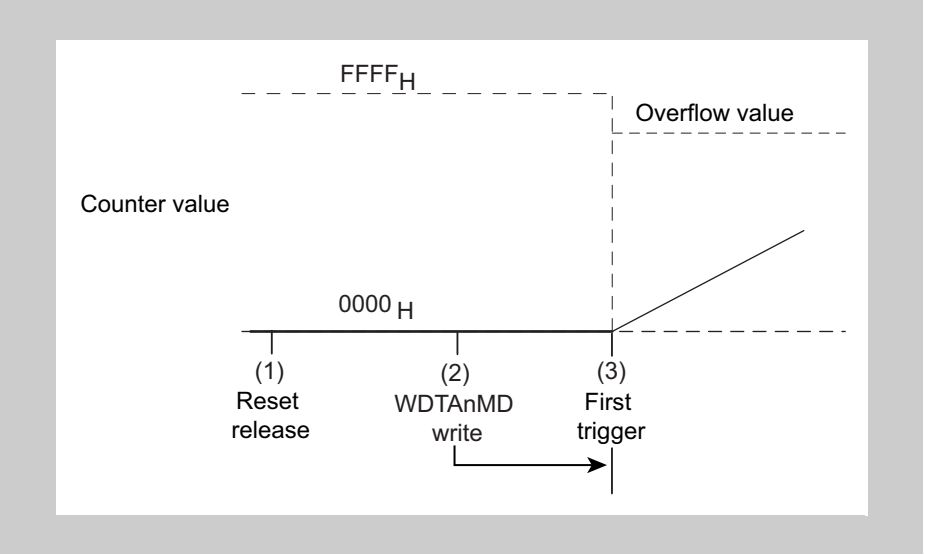


Figure 12-2 Timing Diagram of WDTA Start

The timing diagram above shows the following.

1. After reset release, the counter value remains 0000_H until the first trigger is generated.
2. WDTAnMD should be written before the first trigger is generated. However, the settings are not applied immediately.
3. The counter starts at the first trigger.

   The overflow value and other settings specified in WDTAnMD are applied.

12.3.2 WDTA Trigger

The WDTA trigger has the following functions.

- Starting the WDTA by the first trigger after release from reset (i.e. the first software trigger to start counting).
- Counter re-start trigger to keep the counter from overflowing.

The WDTA can be triggered by writing a fixed activation code to the trigger register (see table 12-5, Trigger Register and Activation Code).

Table 12-5 Trigger Register and Activation Code

Type of Activation Code	Trigger Register	Activation Code
Fixed	WDTAnWDTE	AC_H

12.3.3 Error Detection

The conditions for error detection are:

- Overflow time is exceeded (counter overflow)
- Wrong activation code is written to the trigger register
- Writing to the trigger register outside the open window.
- Illegal update of the watchdog timer mode register (WDTAnMD)
  - Writing a new value to WDTAnMD after the first trigger leads to an error detection.
  - Writing the same value to WDTAnMD after the first trigger does not lead to an error detection.

Error mode

When an error is detected, either an NMI request (WDTAnTNMI) or a reset (WDTAnTRES) is generated.

WDTAnMD.WDTAnERM selects the error mode:

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: Reset mode

The following figure shows the reset or NMI request generation when the counter overflows.

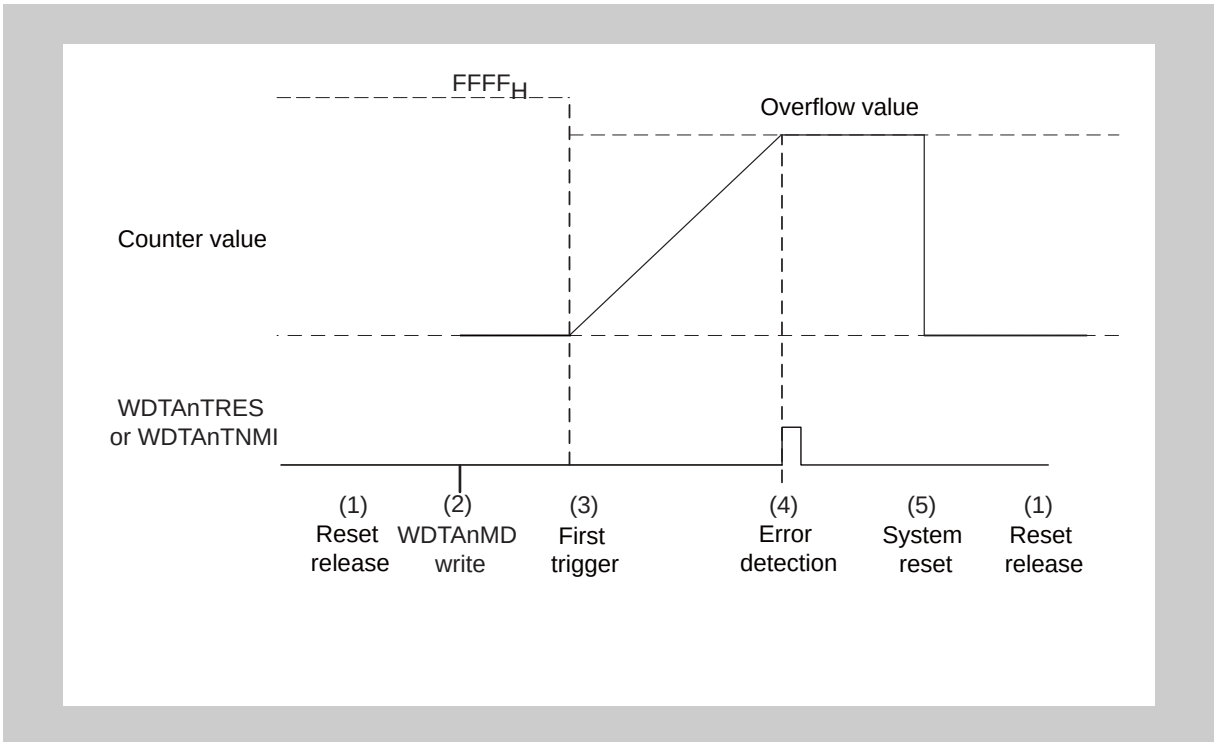


Figure 12-3 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram above shows the following.

1. After reset release, the counter value remains 0000_H until the first trigger is generated.
2. WDTAnMD should be written before the first trigger is generated. However, the settings are not applied immediately.
3. The counter starts at the first trigger.
   The overflow value and other settings specified in WDTAnMD are applied.
4. When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.

   The counter value remains until the system reset is performed.
5. When the system is reset, the counter is cleared.

12.3.4 75% Interrupt Output

When the counter reaches 75% of the overflow value, the interrupt request WDTAnTIT is generated.

The WDTAnMD.WDTAnWIE register enables/disables this function.

The following figure shows the 75% interrupt request generation with the following conditions:

- Count clock changes after the first trigger

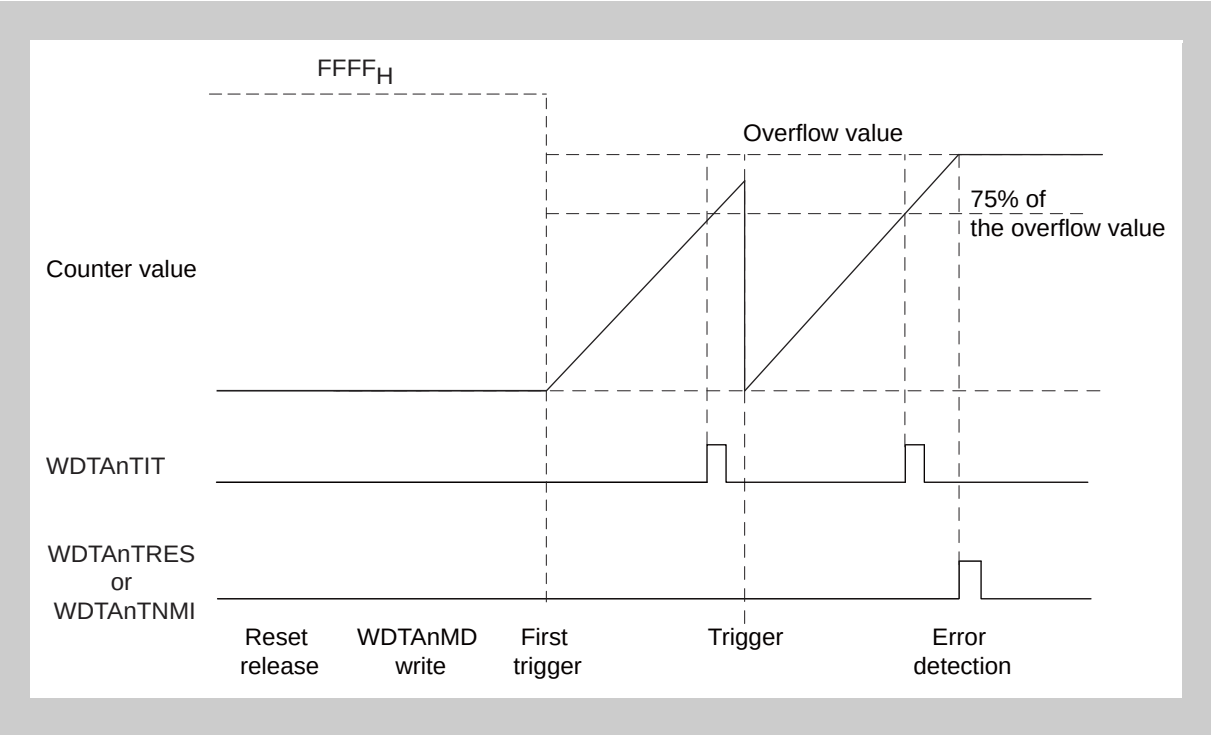


Figure 12-4 Timing Diagram of WDTA 75% Interrupt Output

12.3.5 Window Function

When the window-open period is set to less than 100%, an error is detected if the trigger occurs outside the open window.

The definition of the window-open period differs before and after the first trigger.

- After reset release, the window-open period is 100%.

  Bits WDTAnMD.WDTAnWS[1:0] are not applied.

- After the first trigger, the window-open period is specified by bits WDTAnMD.WDTAnWS[1:0].

The following figure shows WDTA operation with a window-open period of 25%.

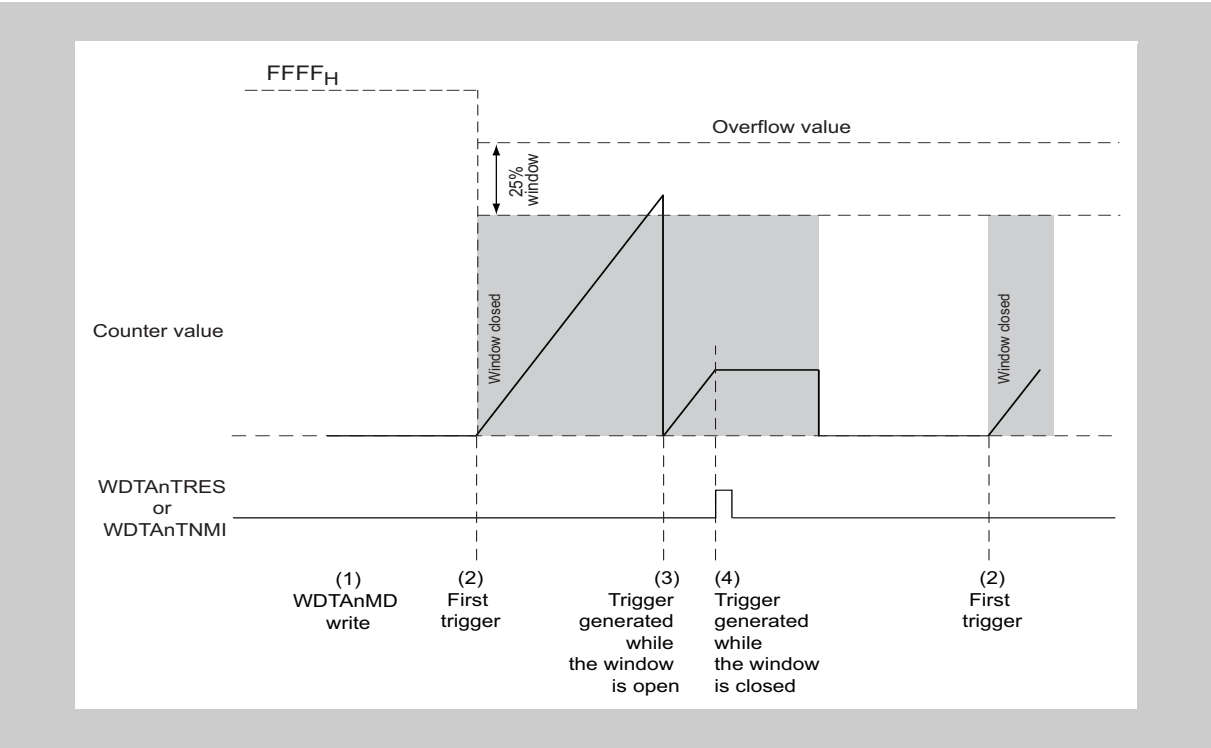


Figure 12-5 Timing Diagram of WDTA Window Function

The timing diagram above shows the following.

1. The operating-mode settings in WDTAnMD should be made before the first trigger is generated. However, the settings are not applied immediately.
2. The window-open period is fixed to 100% for the first trigger. After the first trigger, the overflow value and other settings specified in WDTAnMD are applied.
3. A trigger that occurs in the window-open period does not generate an error.
4. A trigger that occurs in the closed window generates a WDTAnTNMI request or a WDTAnTRES reset, depending on the selected operating mode.

12.4 Registers

This section contains a description of all registers of the WDTA.

12.4.1 WDTA Registers Overview

The WDTA is controlled and operated by the following registers.

Table 12-6 WDTA Register Overview

Register Name	Symbol	Address
WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000_H
WDTA mode register	WDTAnMD	<WDTAn_base> + $000C_H$

12.4.2 WDTA Registers Details

(1) WDTA Enable Register (WDTAnWDTE)

This register is the WDTA start control and trigger register.

WDTA trigger Writing AC_H to this register restarts the counter. Refer to section 12.3.2, WDTA Trigger, for details.

Do not write values other than AC_H to this register. An error will be detected if any other value is written here.

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

Address <WDTAn_base> + 0000_H

Initial value $2C_H$

This register is initialized by any reset sources.

7	6	5	4	3	2	1	0
WDTAn RUN	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-7 WDTAnWDTE Register Contents

Bit Position	Bit Name	Function
7	WDTAnRUN	Enables/disables the WDTAn. 0: WDTAn disable 1: WDTAn enabled Since the WDTA cannot be stopped once it was started, this bit can only be cleared by a reset.

(2) WDTA Mode Register (WDTAnMD)

This register specifies the overflow time, the 75% interrupt output mode, the error mode, and the window-open period.

It can be updated only once after reset release and before the first trigger. The updated value is effective after the next WDTA trigger.

Updating this register after the WDTA has been started leads to error detection, but the read value of this register can be written without generating an error.

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

Address <WDTAn_base> + $000C_H$

Initial value $7F_H$

This register is initialized by any reset sources.

7	6	5	4	3	2	1	0
0	WDTAnOVF[2:0]			WDTAnWIE	WDTAnERM	WDTAnWS[1:0]	
R*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

* Writing to this bit is ignored, reading returns 0.

Table 12-8 WDTAnMD Register Contents

Bit Position	Bit Name	Function
6 to 4	WDTAnOVF[2:0]	Selects the overflow time.
3	WDTAnWIE	Enables/disables the 75% interrupt request WDTAnTIT. 0: WDTAnTIT disabled 1: WDTAnTIT enabled
2	WDTAnERM	Specifies the error mode. 0: NMI request mode 1: Reset mode
1, 0	WDTAnWS[1:0]	Selects the period over which the window is open.

WDTAn OVF2	WDTAn OVF1	WDTAn OVF0	Overflow Time
0	0	0	2^9/WDTATCKI
0	0	1	2^{10}/WDTATCKI
0	1	0	2^{11}/WDTATCKI
0	1	1	2^{12}/WDTATCKI
1	0	0	2^{13}/WDTATCKI
1	0	1	2^{14}/WDTATCKI
1	1	0	2^{15}/WDTATCKI
1	1	1	2^{16}/WDTATCKI

WDTAnWS1	WDTAnWS0	Window-Open Period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Section 13 Timer Array Unit A (TAUA)

This section contains a generic description of the timer array unit A (TAUA).

Caution TAUA may have unavailable channels depending on a channel combination or port I/O selection.

13.1 TAUA Features

Instances This product has the following number of instances of TAUA.

Table 13-1 Instances of TAUA

	TAUA
Instance	2
Name	TAUA0, TAUA1

Instances index n Throughout this section, each of the TAUA instances is identified by "n" (n = 0 or 1), such as TAUAn channel output mode register (TAUAnTOM).

Channels index m TAUA has 16 channels. Throughout this section, each channel is dentified by "m" (m = 0 to 15). Thus, a certain channel is denoted as CHm.
The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even.

The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

Register addresses All TAUAn register addresses are given as address offsets to the individual base addresses <TAUAn_base0>, <TAUAn_base1>.
The base address of each TAUAn is listed in the following table.

Table 13-2 Register Base Addresses <TAUAn_basen>

TAUAn	<TAUAn_base0> Address	<TAUAn_base1> Address
TAUA0	FF80 8000_H	FFFF $C400_H$
TAUA1	FF80 9000_H	FFFF $C800_H$

Clock supply All timer array units provide one clock input.

Table 13-3 TAUAn Clock Supply

TAUAn	Clock	Connected to
TAUA0	PCLK	Clock controller
TAUA1	PCLK	Clock controller

Interrupt and DMA/DTS The timer array unit A can generate the following interrupts and DMA/DTS requests.

Table 13-4 TAUAn Interrupts and DMA/DTS Requests

TAUAn Signal	Function	Connected to
TAUA0:		
INTTAUA0I0 to INTTAUA0I15	Channel 0 to 15 interrupt	Interrupt controllers INTTAUA0I0 to INTTAUA0I15 DMA controller triggers 27 to 42 DTS controller triggers 27 to 42
TAUA1:		
INTTAUA1I0 to INTTAUA1I15	Channel 0 to 15 interrupt	Interrupt controllers INTTAUA1I0 to INTTAUA1I15 DMA controller triggers 43 to 58 DTS controller triggers 43 to 58

I/O signals The I/O signals of the timer array unit A are listed in the table below.

Table 13-5 TAUAn I/O Signals

TAUA Signal	Function	Connected to
TAUAnTTINm	Channel m input	Port TAUAnIm UARTHn (baud rate detection function for LIN communication slave operation)
TAUAnTTOUTm	Channel m output	Port TAUAnOm
TAUA1TTOUT15	Channel 15 output	Safety guardian (SGATTIN input)
TAUAnTSSTm	Simultaneous start trigger input	PIC Timer simultaneous start trigger function

TAUAn interrupts and I/O signals are illustrated below.

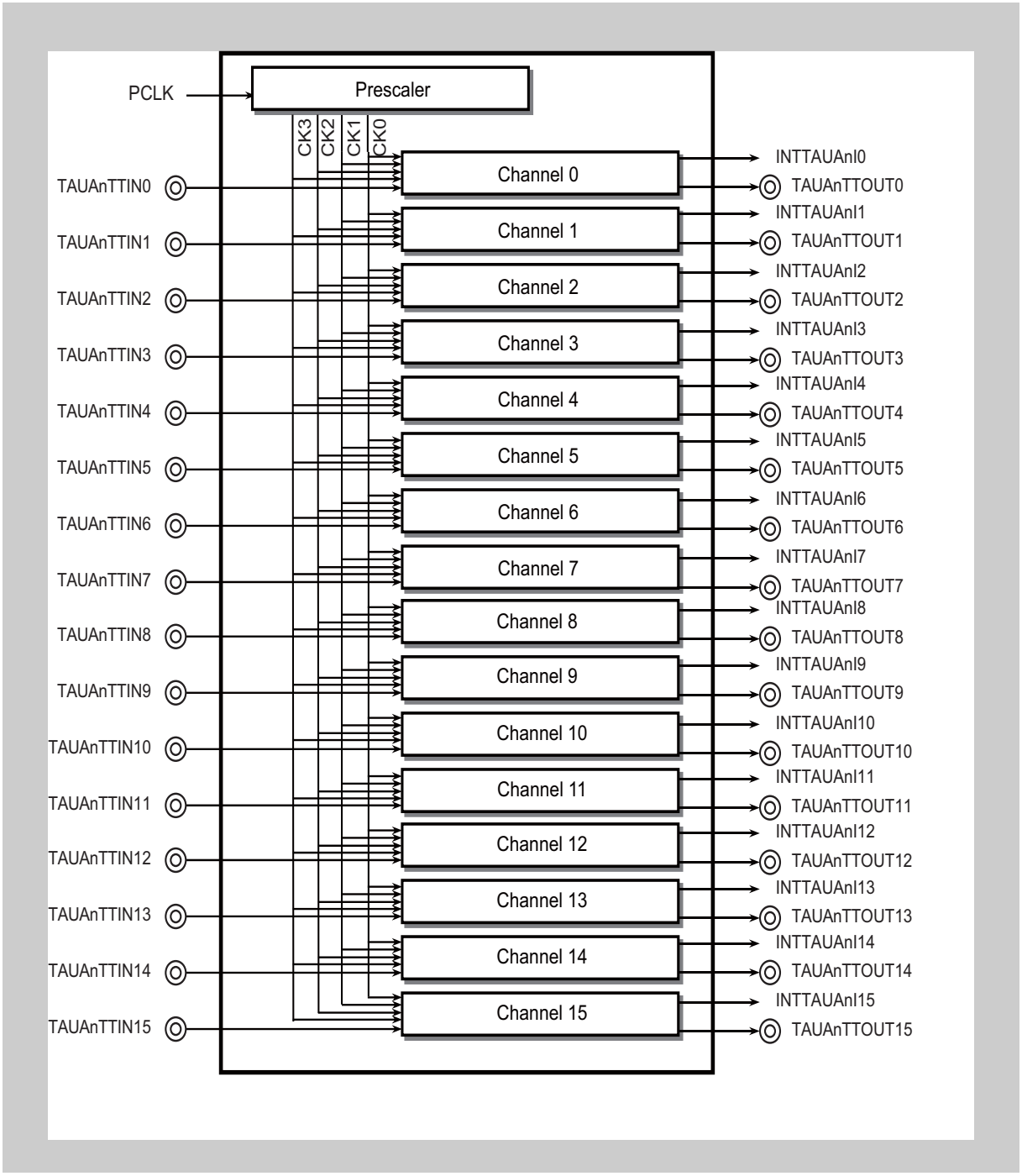


Figure 13-1 TAUA I/O and Interrupt Signals

13.2 Functional Overview

Features summary TAUA has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operations)
- Generation of different types of output signals
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

Caution The timing chart described in this section shows an operating timing image. A delay time is added to a timer input. For details, see section 13.11, TAUAnTTINm Edge Detection.

The following figure shows the main components of the TAUA.

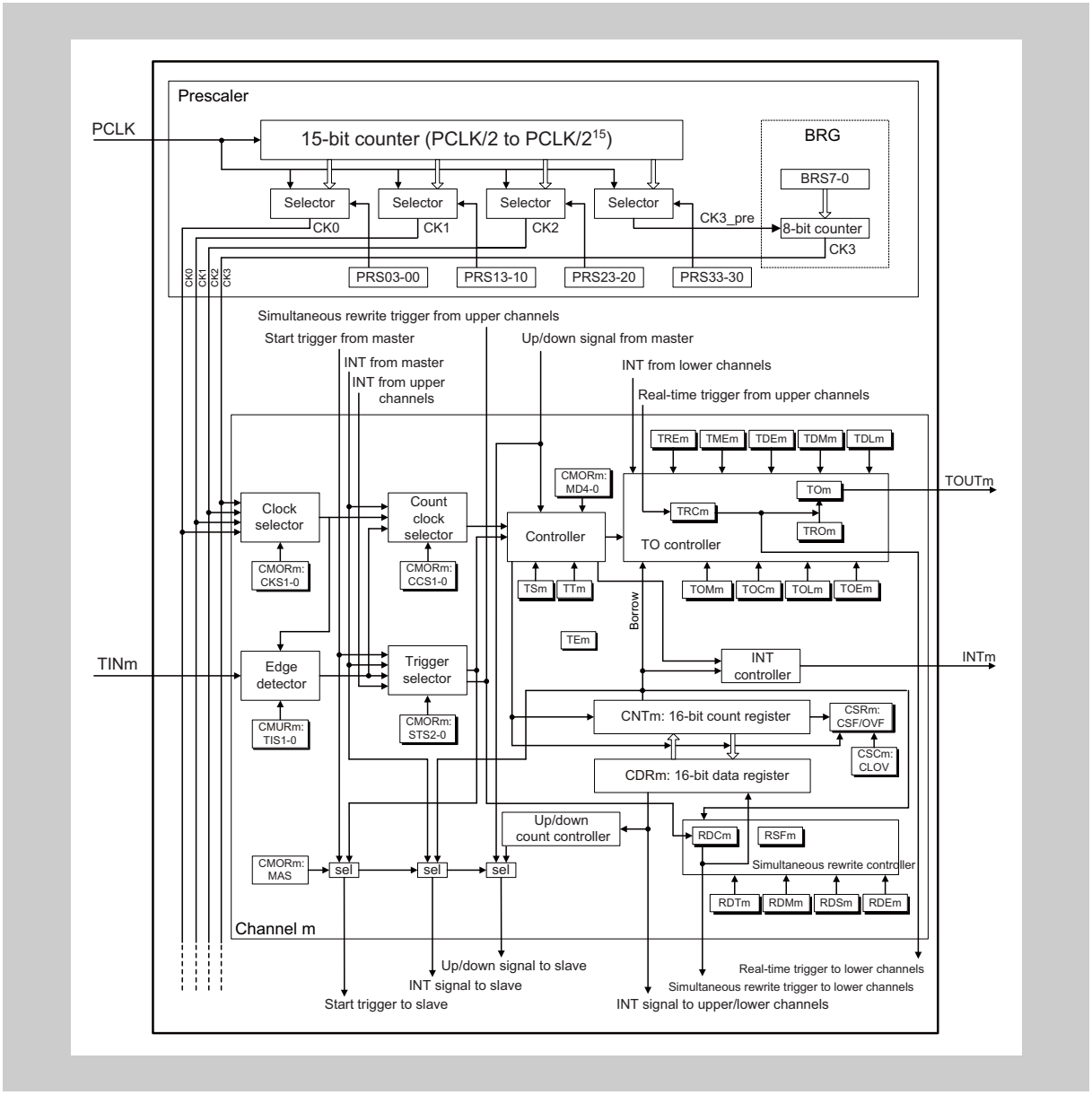


Figure 13-2 TAUA Block Diagram

The prefix "TAUAn" has been omitted from the register names for the sake of brevity in the above figure.

13.2.1 Terms

In this section, the following terms are used.

- Independent/synchronous channel operation

  Independent or synchronous channel operation describes the dependency of channels on each other:

  – If a channel operates independent of all other channels, this is called independent channel operation.

  – If a channel operates depending on other channels, this is called synchronous channel operation.

- Channel group

  In synchronous channel operation, all channels that depend on each other are referred to as a "channel group".

  A channel group has one master channel and one or more slave channels.

- Operating mode

  An operating mode can be selected for every channel m. The operating mode defines the basic operation and features of a channel.

  In synchronous channel operation, every channel in a channel group can operate in a different operating mode.

  Examples are "capture mode", "event count mode", and "interval timer mode".

- Channel output mode

  The channel output mode defines the operation of TAUAnTTOUTm:

  – of a single channel (independent output operation) or

  – of all channels in a channel group (synchronous output operation).

  Examples are "independent channel output mode 1" and "synchronous channel output mode 2 with dead time output".

- Channel operation function

  The channel operation function defines the complete function and all features:

  – of a single channel (independent channel operation) or

  – of all channels in a channel group (synchronous channel operation).

- Upper and lower channels

  Upper and lower channels are defined below.

  – Upper channel: A channel with a lower channel number

  – Lower channel: A channel with a higher channel number

  Example: For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

13.3 Functional Description

The timer array unit A performs various count or timer operations and outputs a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16 bit counter TAUAnCNTm and a 16-bit data register TAUAnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operating modes, either independently or in combination with other channels (synchronously); for example, in a combination of one master and one or more slave channels, slave channels depend on the master channel.
When a channel is operated independently, its operating mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, one channel can be used to control the count start timing or reset timing of others.

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.
Plescaler outputs CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.The fourth prescaler output CK3 can be adjusted more precisely by an additional division factor that is not a power of 2.

Count clock selection

For every channel, the count clock selector selects which of the following is used as a clock source:

- One of the prescaler outputs CK0 to CK3 (selected by the clock selector)
- INTTAUAnIm from master channel
- Valid edge of TAUAnTTINm input signal

Controller

The controller controls the main operations of the counter:

- Operating mode (selected by TAUAnCMORm.TAUAnMD[4:0] bits)
- Counter start enable (TAUAnTS.TAUAnTSm) and counter stop (TAUAnTT.TAUAnTTm)
  When counter start is enabled, status flag TAUAnTE.TAUAnTEm is set.
- Count direction (can be controlled by master channel)

Trigger selector

Depending on the selected operating mode, the counter starts automatically when it is enabled (TAUAnTE.TAUAnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as a start trigger.

- Synchronous channel start trigger input TAUAnTSSTm
  For details on how to make a simultaneous start between the units, see section 26.4.1, Simultaneous Start Trigger Function.
- Valid edge of TAUAnTTINm input
- INTTAUAnIm from master or any upper channel
- Up/down output trigger signal of master channel
- Dead-time output signal of TAUAnTTOUTm generation unit

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operating modes. The data registers of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUAnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals or triangular wave signals.

13.3.1 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 13-6 Functional List of TAUA Operations

Function of Independent Operation	Function of Synchronous Operation
Independent channel operation functions	Synchronous channel operation functions
Interval timer function	PWM output function
TAUAnTTINm input interval timer function	Delay pulse output function
Delay count function	AD conversion trigger output function type1
One-pulse output function	Synchronous PWM signal functions triggered by an external signal
Independent channel signal measurement functions	One-shot pulse output function
TAUAnTTINm input pulse interval measurement function	Offset trigger output function
TAUAnTTINm input signal width measurement function	Synchronous triangle PWM output functions
TAUAnTTINm input period count detection function	Triangle PWM output function
TAUAnTTINm input pulse interval judgment function	Triangle PWM output function with dead time
TAUAnTTINm input signal width judgment function	AD conversion trigger output function type 2
Independent channel real-time functions	Synchronous non-complementary modulation output function and synchronous complementary output function
Real-time output function type 1	Non-complementary modulation output function type 1
Real-time output function type 2	Non-complementary modulation output function type 2
Independent channel simultaneous rewrite function	Complementary modulation output function
Simultaneous rewrite trigger generation function type 1	Other synchronous channel functions
Independent channel one-phase PWM function	Interrupt sampling function
One-phase PWM output function	
Other independent channel functions	
External event count function	
Clock divide function	
TAUAnTTINm input position detection function	

13.4 General Operating Procedures

The following lists the general operation procedures for the TAUAn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. The control register of TAUAnTTOUTm is also initialized and outputs a low level.

1. Set the TAUAnTPS and TAUAnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUAn function:
   – Set the operating mode
   – Set the channel output mode
   – Set any other control bits
3. Enable the counter by setting the TAUAnTS.TAUAnTSm bit to 1.

   The counter starts to count immediately, or when an appropriate trigger is detected, according to bit settings.
4. If desired during counting, and if possible for the configured function, stop the counter or perform a forced restart operation.
5. To stop the function, set the TAUAnTT.TAUAnTTm bit to 1.

Note For details on necessary control bits and the operations of individual functions, see section 13.13, Independent Channel Operation Functions, and section 13.14, Synchronous Channel Operation Functions.

13.5 Operating Modes

The TAUA contains 12 operating modes.

One operating mode can be set for each channel, which is specified using the TAUAnCMORm.TAUAnMD[4:0] bits.

Note Some of the registers and bits are fixed and some are user-selectable depending on the operation function.

For details on the registers and bit settings, see the corresponding sections of operational functions.

13.6 Concepts of Synchronous Channel Operation

In synchronous channel operation, multiple channels depend on each other, or are affected by changes in another channel. Therefore, several rules apply for the use of synchronous channel functions. These rules are detailed in section 13.6.1, Rules.

Two special features for synchronous channel operation are detailed in the following sections:

- Section 13.6.2, Simultaneous Start and Stop of Synchronous Channel Counters
- Section 13.7, Simultaneous Rewrite

13.6.1 Rules

Number of masters and slaves

- Only even channels (CH0, CH2, CH4,...) can be set as master channels. Any channel apart from CH0 can be set as a slave channel
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, channels (CH3, CH4, CH5,...) following CH3 can be set as slave channels.
- If two master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Count clock

- The same count clock should be set for master and slave channels. This is achieved by setting the TAUAnCMORm.CKS[1:0] bits of master and slave channels to the same value.

The basic concepts of master and slave usage and the count clocks are illustrated in the following figure.

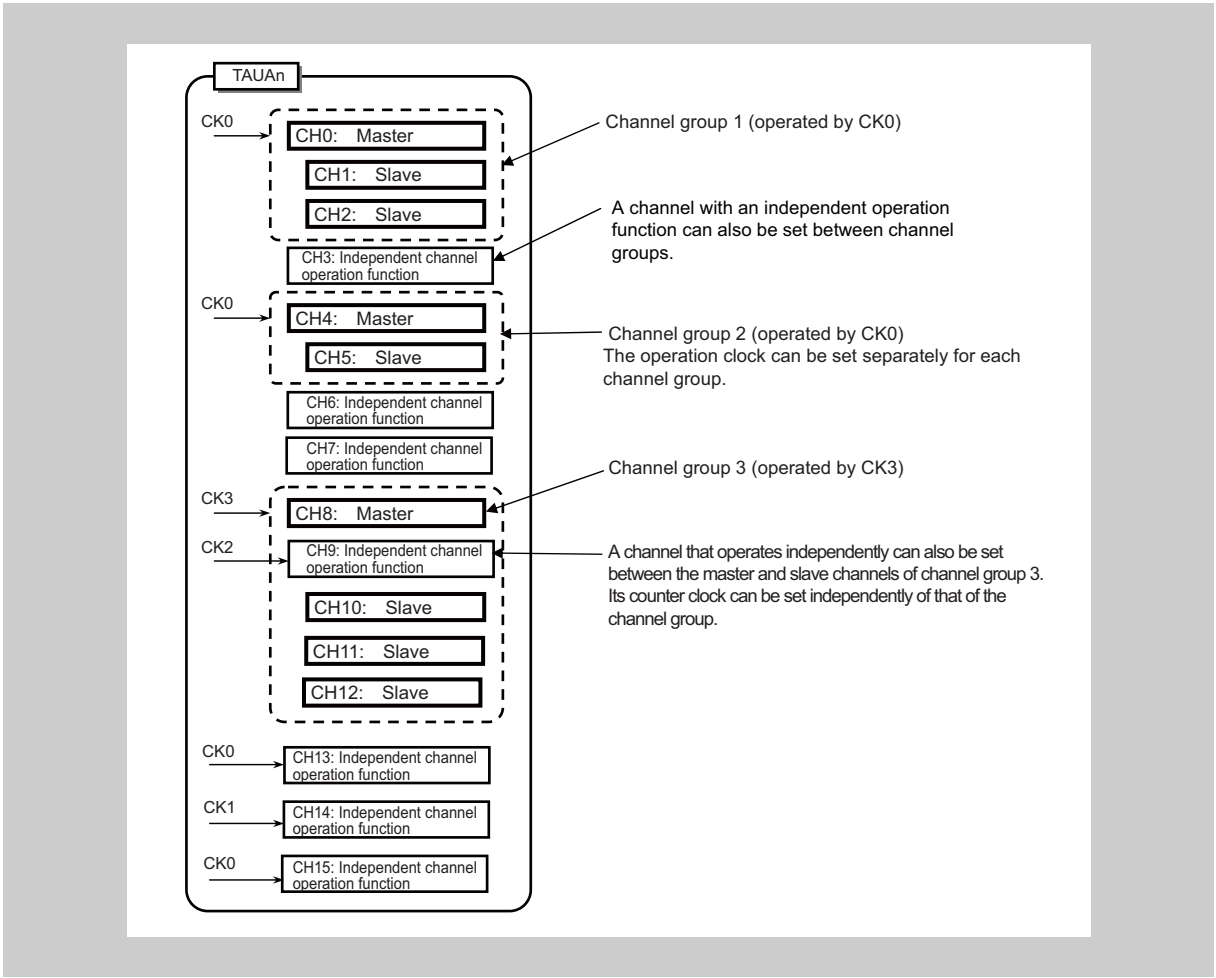


Figure 13-3 Grouping of Channels and Assignment of Count Clocks

Control trigger signal for master/ salve channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals for their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

13.6.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

(1) Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUAnTS.TAUAnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUAnTT.TAUAnTTm bits of the channels should be set at the same time.

Setting to the TAUAnTS.TAUAnTSm bits to 1 also sets the corresponding TAUAnTE.TAUAnTEm bits to 1, enabling counting. The count start timing depends on operating mode.

(2) Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal. For details on how to make a simultaneous start between the units, see section 26.4.1, Simultaneous Start Trigger Function.

13.7 Simultaneous Rewrite

13.7.1 Overview

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data registers and control registers (TAUAnCDRm and TAUAnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operating mode) reaching a certain value
- INTTAUAnIm being issued on the upper channel specified by TAUAnRDC.TAUAnRDCm

There are three methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 13-7 Simultaneous Rewrite Methods and Trigger Timing

Method	Simultaneous Rewrite Trigger Timing	TAUAn RDE. TAUAn RDEm	TAUAn RDS. TAUAn RDSm	TAUAn RDM. TAUAn RDMm
-	No simultaneous rewrite	0	0	0
A	The master channel starts/restarts counting.	1	0	0
B	The slave channel starts counting down at the upper peak of a triangular cycle.	1	0	1
C1	INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.RDCm	1	1	0

The following table lists which of these three methods is available for each channel operation function. For details on the individual channel operation functions, see section 13.13, Independent Channel Operation Functions, and section 13.14, Synchronous Channel Operation Functions.

Table 13-8 Simultaneous Rewrite and Trigger Timing

Function	-	A	B	C1
Simultaneous rewrite trigger generating function type 1				X
PWM output function		X		X
One-shot pulse output function		X		
Offset trigger output function	X			
Delay pulse output function		X		
Triangle PWM output function			X	X
Triangle PWM output function with dead time			X	X
Interrupt sampling function		X	X	X
AD conversion trigger output function type 1		X		X
AD conversion trigger output function type 2			X	X
Non-complementary modulation output function type 1		X		X
Non-complementary modulation output function type 2			X	X
Complementary modulation output function			X	X

Note X: Available Blank: Not available

13.7.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

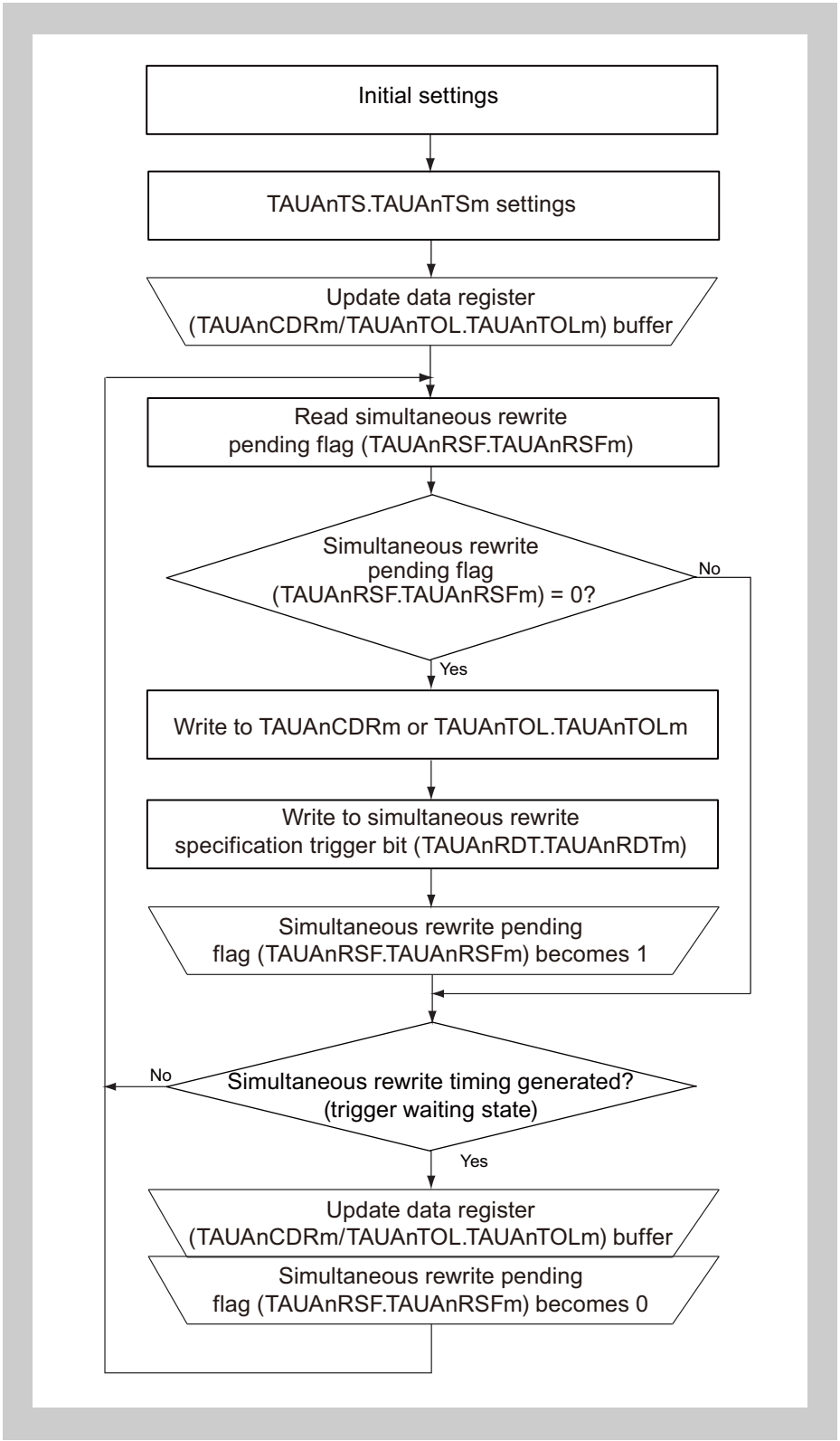


Figure 13-4 General Procedure for Simultaneous Rewrite

(1) Initial Settings

- To enable simultaneous rewrite in channel m, set TAUAnRDE.TAUAnRDEm = 1
- To select the type of simultaneous rewrite, set TAUAnRDM.TAUAnRDMm and TAUAnRDS.TAUAnRDSm according to the values listed in table 13-7, Simultaneous Rewrite Methods and Trigger Timing.
- To select which upper channel is monitored for simultaneous rewrite triggers, use TAUAnRDC.TAUAnRDCm (prerequisite: TAUAnRDS.TAUAnRDSm is set in upper channel.)

(2) Start Counter and Count Operation

- To start all the TAUAnCNTm counters of the channel group, set the corresponding TAUAnTS.TAUAnTSm bits to1. The values of TAUAnTOL.TAUAnTOLm and the data registers (TAUAnCDRm) are loaded into the corresponding TAUAnTOL.TAUAnTOLm buffer (TAUAnTOL.TAUAnTOLm buf) and data buffer registers (TAUAnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUAnRDT.TAUAnRDTm) to 1 sets the reload flag (TAUAnRSF.TAUAnRSFm) to 1, enabling simultaneous rewrite. TAUAnRSF.TAUAnRSFm remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the TAUAnRSF.TAUAnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUAnRSF.TAUAnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

(3) Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUAnRSF.TAUAnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- The TAUAnRSF.TAUAnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

13.7.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUAnRDE.TAUAnRDEm, TAUAnRDS.TAUAnRDSm, TAUAnRDM.TAUAnRDMm, and TAUAnRDC.TAUAnRDCm cannot be changed while the counter is in operation (TAUAnTE.TAUAnTEm = 1).
- TAUAnTOL.TAUAnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUAnTOL.TAUAnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUAnTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUAnRDS.TAUAnRDSm = 1), the TAUAnRDC.TAUAnRDCm bit controls all the lower channels. This means that if the TAUAnRDC.TAUAnRDCm bits of CH2 and CH7 are set to 1 and the TAUAnRDC.TAUAnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger (TAUAnRDE.TAUAnRDEm and TAUAnRDS.TAUAnRDSm = 1) but no upper channel is set (TAUAnRDC.TAUAnRDC[15:0] = 0), simultaneous rewrite cannot take place.

13.7.4 Type of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

(1) Simultaneous Rewrite when the Master Channel Starts/Restarts to Count (Method A)

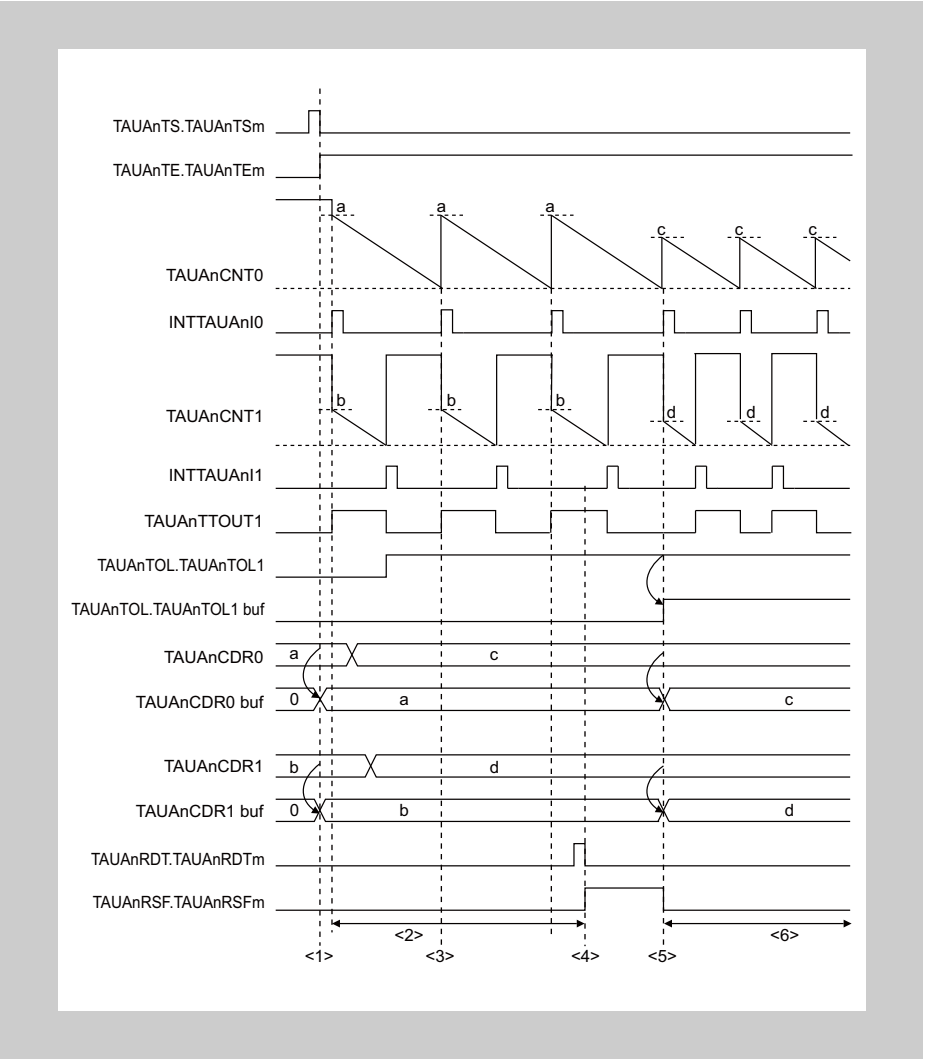


Figure 13-5 Simultaneous Rewrite when the Master Channel Starts/ Restarts Counting

Setting

- CH0 is the master channel which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

1. When TAUAnTS.TAUAnTSm = 1 is set, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer and the value of TAUAnTOL.TAUAnTOLm is copied to the TAUAnTOL.TAUAnTOLm buffer.
2. The TAUAnCDRm and TAUAnTOL.TAUAnTOLm registers can be written at any time.
3. CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUAnRSF.TAUAnRSFm = 0)
4. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
5. Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUAnCDRm value is loaded into the TAUAnCDRm buffer and the TAUAnTOL.TAUAnTOLm value is loaded into the TAUAnTOL.TAUAnTOLm buffer.
6. The counters count down and await the next simultaneous rewrite trigger. The values of TAUAnCDRm and TAUAnTOL.TAUAnTOLm can be changed again.

(2) Simultaneous Rewrite at the Peak of a Triangular Cycle of Slave Channel (Method B)

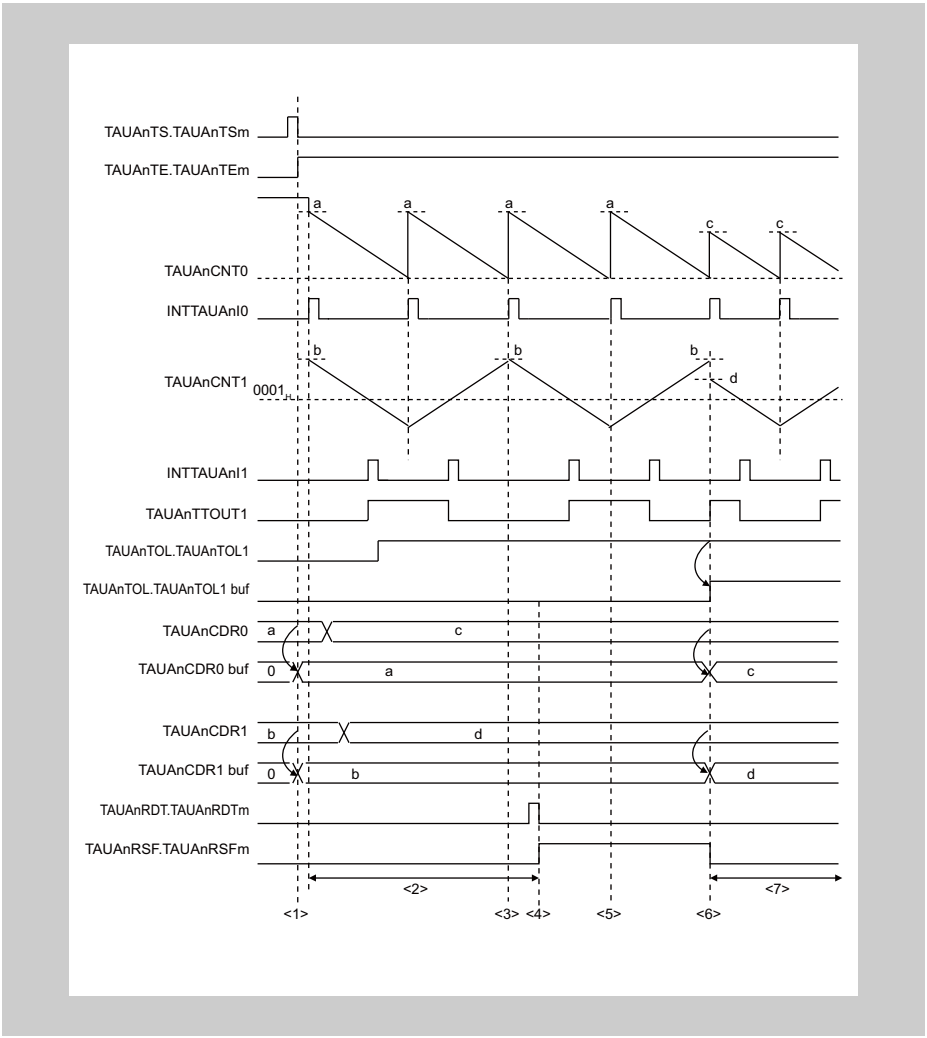


Figure 13-6 Simultaneous Rewrite at the Peak of a Triangular Cycle of Slave Channel

Setting

- CH0 is the master channel which performs counting up and down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

1. When TAUATS.TAUAnTSm = 1 is set, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm and TAUAnTOL registers can be written at any time.
3. Simultaneous rewrite does not occur because it is disabled (TAUAnRSF.TAUAnRSFm = 0).
4. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
5. Simultaneous rewrite does not take place at the bottom of the triangular cycle.
6. Simultaneous rewrite takes place at the top of the triangular cycle. The TAUAnCDRm value is loaded into the TAUAnCDRm buffer, the TAUAnTOL.TAUAnTOLm value is loaded into the TAUAnTOL.TAUAnTOLm buffer.
7. The counters count down and await the next simultaneous rewrite trigger. The values of TAUAnCDRm and TAUAnTOL.TAUAnTOLm can be changed again.

(3) Simultaneous Rewrite when INTTAUAnIm is Generated on an Upper Channel Specified by TAUAnRDC.TAUAnRDCm (Method C1)

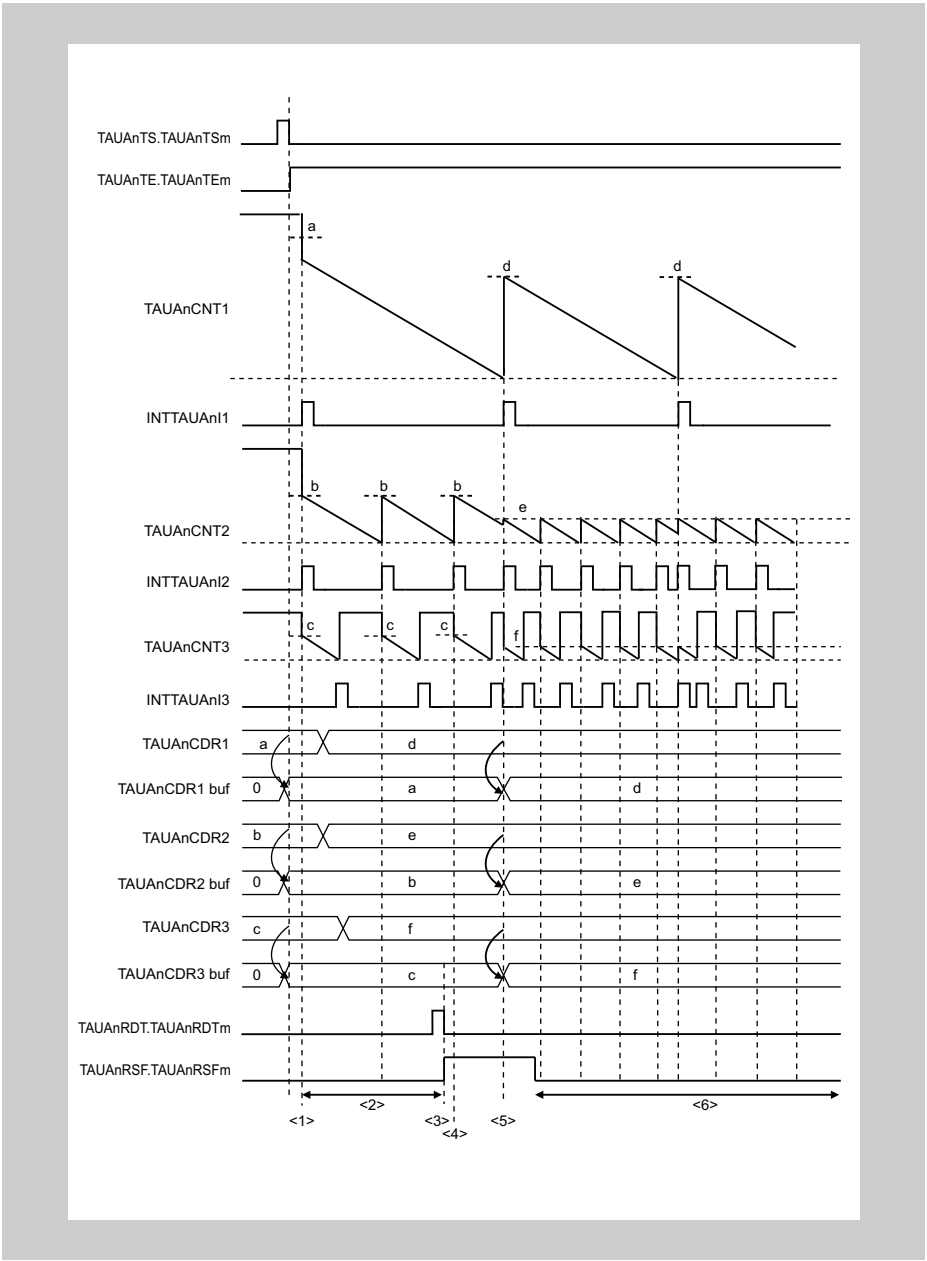


Figure 13-7 Simultaneous Rewrite when INTTAUAnIm is Generated on an Upper Channel Specified by TAUAnRDC.TAUAnRDCm

Setting

- CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUAnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, TAUAnCDRm value is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm register is always ready to write.
3. By setting the reload data trigger bit (TAUAnRDT.TAUAnRDTm) to 1, the status flag is set (TAUAnRSF.TAUAnRSFm = 1) to enable simultaneous rewrite.
4. Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
5. Simultaneous rewrite is triggered by INT1 which is generated when counter1 reaches 0000_H. The TAUAnCDRm values are loaded into the corresponding TAUAnCDRm buffers.
6. The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUAnCDRm registers can be rechanged.

13.8 Channel Output Modes

The output of the TAUAnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUAnTOE.TAUAnTOEm = 0)

  When controlled by software, the value written in the output register bit (TAUAnTO.TAUAnTOm) is sent out of the output pin (TAUAnTTOUTm).

- By TAUA signals (TAUAnTOE.TAUAnTOEm = 1)

  When controlled by TAUA signals, the output level of TAUAnTTOUTm is set or reset or toggled by internal signals. The value of TAUAnTO.TAUAnTOm is updated accordingly to reflect the value of TAUAnTTOUTm

  – Independently (TAUAnTOM.TAUAnTOMm = 0)

    In case of independent operation, the output of the TAUAnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUAnTOM.TAUAnTOMm = 0).

  – Synchronously (TAUAnTOM.TAUAnTOMm = 1)

    In case of synchronous operation, the output of the TAUAnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUAnTOM.TAUAnTOMm = 1).

The TAUAnTO.TAUAnTOm bit can always be read to determine the current value of TAUAnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in table 13-9, Channel Output Modes.

The channel output modes are described in details below.

- Section 13.8.2, Channel Output Modes Controlled Independently by TAUAn Signals
- Section 13.8.3, Channel Output Modes Controlled Synchronously by TAUAn Signals

Batch operation of TAUAnTOm bit

Whether a set value is reflected to the TAUAnTOm bit or not is controlled by the TAUAnTOE.TAUAnTOEm bit.

The TAUAnTOm setting is written only to the bit (channel) set with TAUAnTOE.TAUAnTOEm bit = 0 when a write to the TAUAnTO register is attempted. No TAUAnTOm setting is reflected to the bit (channel) set with TAUAnTOE.TAUAnTOEm bit = 1.

Note TAUAnTO.TAUAnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or inverted logic of the output is specified by control bit TAUAnTOL.TAUAnTOLm.

The value of TAUAnTOL.TAUAnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUAnTOL.TAUAnTOLm result in an invalid TAUAnTTOUTm signal output.

See section 13.7, Simultaneous Rewrite.

The various channel output modes and the channel output control bits are listed in the following table.

Table 13-9 Channel Output Modes

Channel Output Mode	TAUAn TOE. TAUAn TOEm	TAUAn TOM. TAUAn TOMm	TAUAn TOC. TAUAn TOCm	TAUAn TDE. TAUAn TDEm	TAUAn TRE. TAUAn TREm	TAUAn TME. TAUAn TMEm	TAUAn TDM. TAUAn TDMm
By software							
Independent channel output mode controlled by software	0	X					
By TAUA signals, independently							
Independent channel output mode 1	1	0	0	0	0	0	0
with real-time output					1		
Independent channel output mode 2			1		0		
By TAUA signals, synchronously							
Synchronous channel output mode 1	1	1	0	0	0	0	0
with non-complementary modulation output					1	1	
Synchronous channel output mode 2			1	0	0	0	0
with dead time output				1			
with one-phase PWM output							1
with complementary modulation output					1	1	0
with non-complementary modulation output			1	0			

Caution 1. Any combinations not described in the above table are prohibited.

Caution 2. The bit marked with "x" can be set to any value.

Caution 3. The following bits cannot be changed during count operation (TAUAnTE.TAUAnTE = 1):

- TAUAnTOE.TAUAnTOEm
- TAUAnTOM.TAUAnTOMm
- TAUAnTOC.TAUAnTOCm
- TAUAnTDE.TAUAnTDEm
- TAUAnTRE.TAUAnTREm
- TAUAnTDM.TAUAnTDMm

Caution 4. The following bits should remain unchanged during count operation (TAUAnTE.TAUAnTEm = 1) except in channel output modes with modulation output:

- TAUAnTME.TAUAnTMEm
- TAUAnTDL.TAUAnTDLm

13.8.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUAnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUAnTOE.TAUAnTOEm = 0).

1. Set TAUAnTO.TOm to specify the initial level of the TAUAnTTOUTm output.
2. Set channel output mode according to table 13-9, Channel Output Modes, and the output logic using the TAUAnTOL.TAUAnTOLm bit.
3. Start the counter (TAUAnTS.TAUAnTSm = 1).

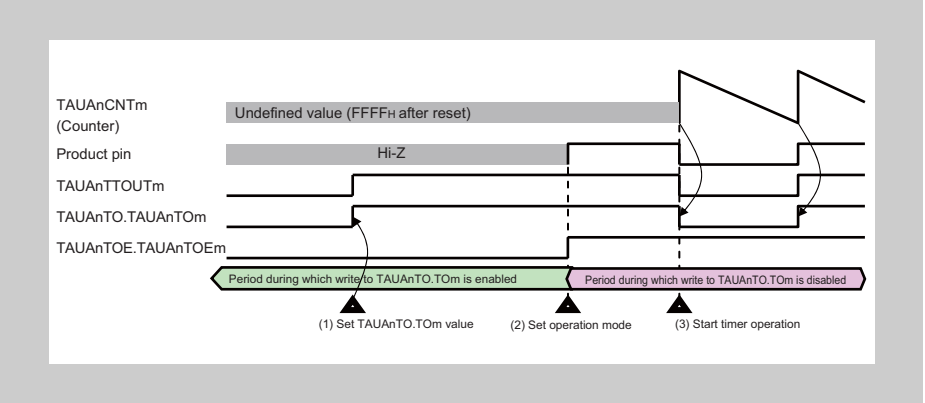


Figure 13-8 General Procedure for Specifying a TAUAnTTOUTm Channel Output Mode

13.8.2 Channel Output Modes Controlled Independently by TAUAn Signals

This section lists the channel output modes that are controlled independently by TAUAn signals. The control bits used to specify a mode are listed in table 13-9, Channel Output Modes.

(1) Independent Channel Output Mode 1

Set/reset conditions In this output mode, TAUAnTTOUTm toggles when INTTAUAnIm is detected. The value of TAUAnTOL.TAUAnTOLm is ignored.

Prerequisites There are no prerequisites other than those shown in table 13-9, Channel Output Modes.

(2) Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUAnTRO.TAUAnTROm bit of the trigger channel is output to TAUAnTTOUTm. The trigger channel is specified by setting the corresponding TAUAnTRC.TRCm bit to 1. It controls all lower channels for which TAUAnTRC.TRCm = 0.

Set/reset conditions The value of TAUAnTRO.TAUAnTROm bit is sent to TAUAnTTOUTm only when an INTTAUAnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUAnTTINm input edge/counter start

The type of trigger is set using the TAUAnCMORm.TAUAnMD[4:1] bits.

Prerequisites Both master and slave channels can be set as a trigger generation channel. A channel for which TAUAnTRC.TRCm is set to 1 serves as a trigger generation channel even if TAUAnTRE.TAUAnTREm is set to 0. If there is no channel for which TAUAnTRE.TAUAnTREm is set to 1 or if TAUAnTRC.TAUAnTRCm = 0, real-time output cannot take place.

This can be seen in the following figure.

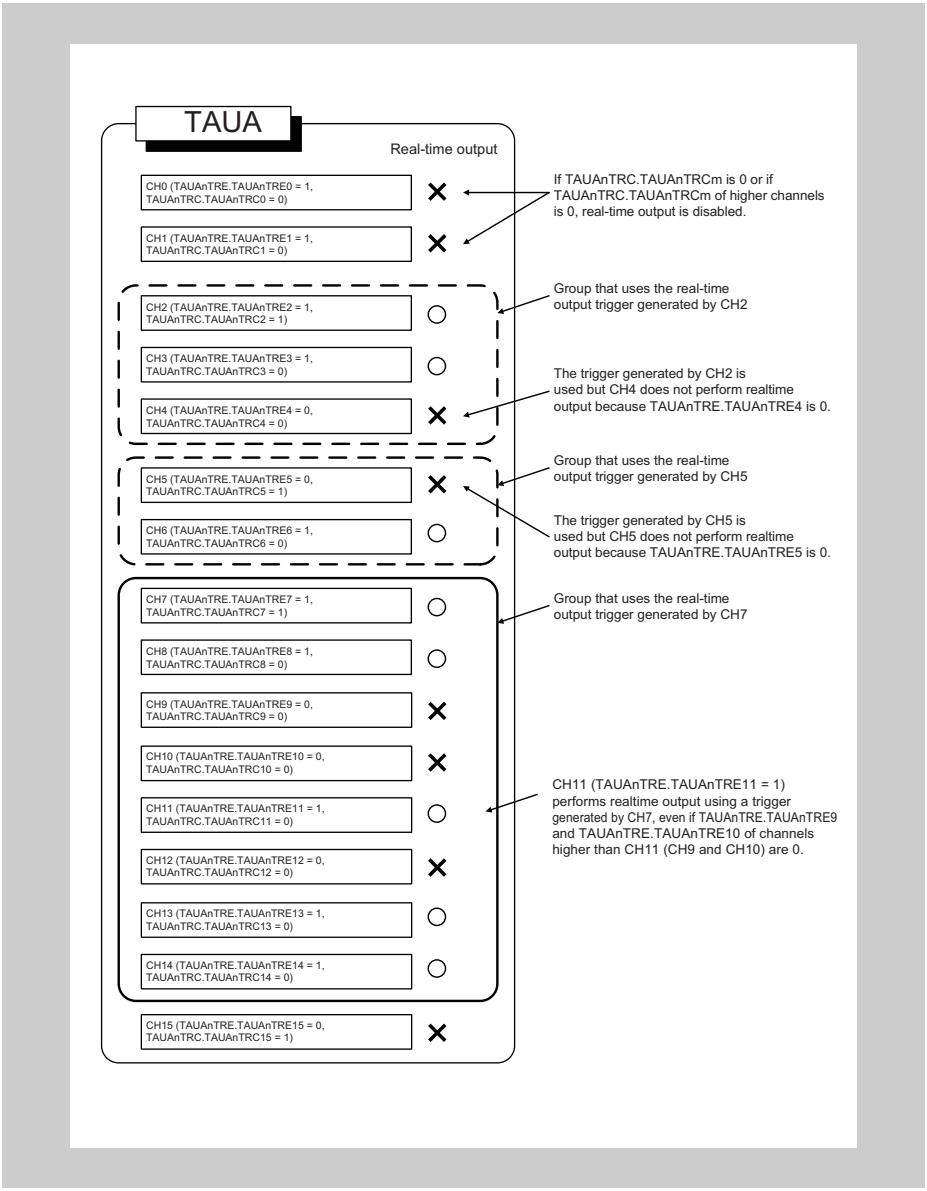


Figure 13-9 Real-Time Output

(3) Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUAnTTOUTm is set when INTTAUAnIm occurs at the time of count start, and reset when INTTAUAnIm occurs due to a match between TAUAnCNTm and TAUAnCDRm.

Prerequisites

There are no prerequisites other than those shown in table 13-9, Channel Output Modes.

13.8.3 Channel Output Modes Controlled Synchronously by TAUAn Signals

This section lists the channel output modes that are controlled synchronously by TAUAn signals. The control bits used to specify a mode are listed in table 13-9, Channel Output Modes.

(1) Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUAnIm of master channel serves as a set signal and INTTAUAnIm of the slave channel as a reset signal. If INTTAUAnIm of master channel and INTTAUAnIm of the slave channel are generated at the same time, INTTAUAnIm of the slave channel (reset signal) has priority over INTTAUAnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in table 13-9, Channel Output Modes.

(2) Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUAnTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUAnTRO.TAUAnTROm) of a channel.

The phase period to which the dead time is added is specified using the TAUAnTDL.TDLm bit; for positive phase set TAUAnTDL.TAUAnTDLm = 0 and for negative phase set TAUAnTDL.TAUAnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm can only be changed during count operation.

- If TAUAnTME.TAUAnTMEm is changed, its new value is applied upon detection of INTTAUAnIm on the specified channel.
- If TAUAnTME.TAUAnTMEm and TAUAnTDL.TAUAnTDLm are changed, their new values are applied upon detection of INTTAUAnIm on the master channel.

(3) Synchronous Channel Output Mode

In this output mode, the operating mode should be set to up/down count mode. The result is a triangle PWM wave at TAUAnTTOUTm. For details, see section 13.14.6, Triangle PWM Output Function.

Set/reset conditions TAUAnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUAnTTOUTm to toggle.

Prerequisites A set of two channels is required to generate the triangle PWM output. TAUAnTTOUTm should be set to 0 before the function starts.

(4) Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUAnTTOUTm. The set/reset conditions are shown in the following figure.

Set/reset conditions

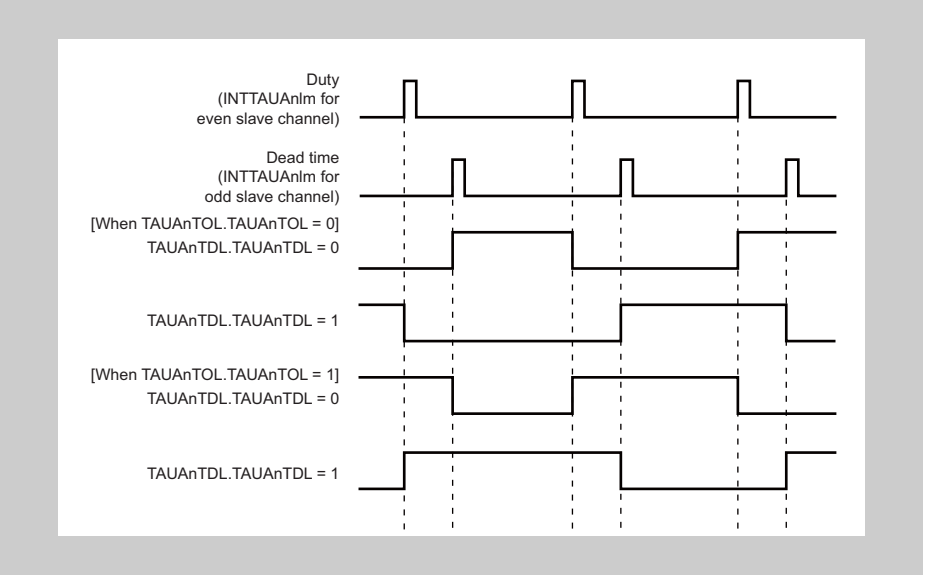


Figure 13-10 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUAnTDL.TAUAnTDLm = 0 for rising edges and TAUAnTDL.TAUAnTDLm = 1 for falling edges.

Prerequisites Dead time control requires a set of three channels, each operating in the following modes:

- One master channel

  The master channel should be set to interval timer mode.

- One even slave channel

  The even slave channel should be set up/down count mode.

- One odd slave channel (even channel + 1)

  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUAnTOE.TAUAnTOEm
- TAUAnTME.TAUAnTMEm
- TAUAnTRE.TAUAnTREm
- TAUAnTOM.TAUAnTOMm
- TAUAnTOC.TAUAnTOCm
- TAUAnTDE.TAUAnTDEm
- TAUAnTDM.TAUAnTDMm

(5) Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUAnTTOUTm. The set/reset conditions are shown in the following figure.

Set/reset conditions

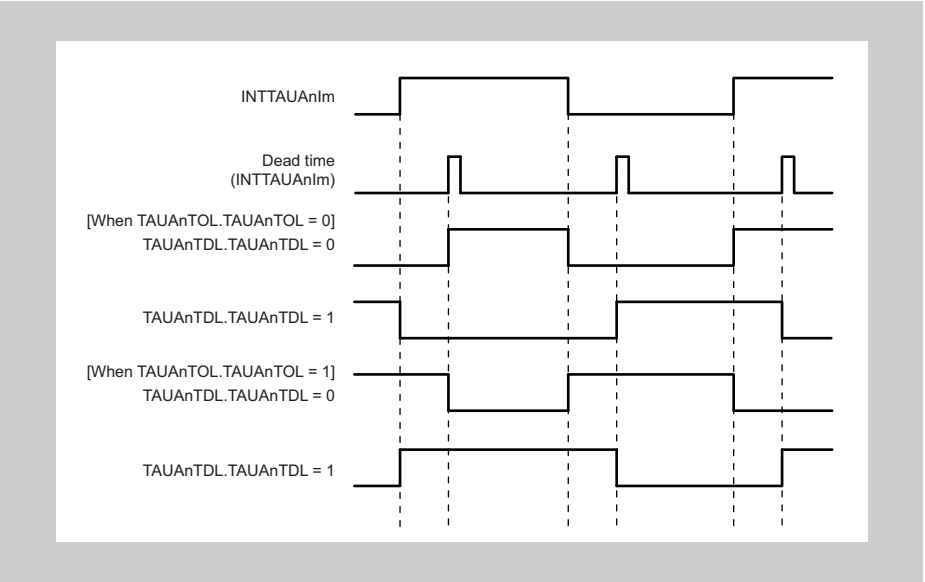


Figure 13-11 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUAnTDL.TAUAnTDLm = 0 for rising edges and TAUAnTDL.TAUAnTDLm = 1 for falling edges.

Prerequisites One-phase PWM output control requires a set of two channels:

- One even slave channel
- One odd slave channel (even channel + 1)

  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUAnTOE.TAUAnTOEm
- TAUAnTME.TAUAnTMEm
- TAUAnTRE.TAUAnTREm
- TAUAnTOM.TAUAnTOMm
- TAUAnTOC.TAUAnTOCm
- TAUAnTDE.TAUAnTDEm
- TAUAnTDM.TAUAnTDMm

(6) Synchronous Channel Output Mode 2 with Complementary Modulation Output

Set/reset conditions

In this output mode, TAUAnTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUAnTRO.TAUAnTROm), the modulation output bit (TAUAnTME.TAUAnTMEm), and the output level bit (TAUAnTOL.TAUAnTOLm) of a pair of slave channels.

For details, see section 13.14.11, Complementary Modulation Output Function.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, 4 more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm can only be changed during count operation.

- If TAUAnTME.TAUAnTMEm is changed during operation, its new value is applied upon detection of INTTAUAnIm at the specified channel.
- If TAUAnTME.TAUAnTMEm and TAUAnTDL.TAUAnTDLm are changed, their new values are applied upon detection of INTTAUAnIm on an even slave channel

(7) Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a rectangular wave while mode 2 has a triangular wave.

13.9 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUAnTS.TAUAnTSm is set to 1 in each operating mode.

In all modes, the value of data register (TAUAnCDRm register) and whether or not an interrupt occurs depends on mode and register settings.

Caution The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

13.9.1 Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode

The counter starts operating with the next count clock after TAUAnTS.TAUAnTSm is set to 1. The value of data register is also loaded when the counter starts.

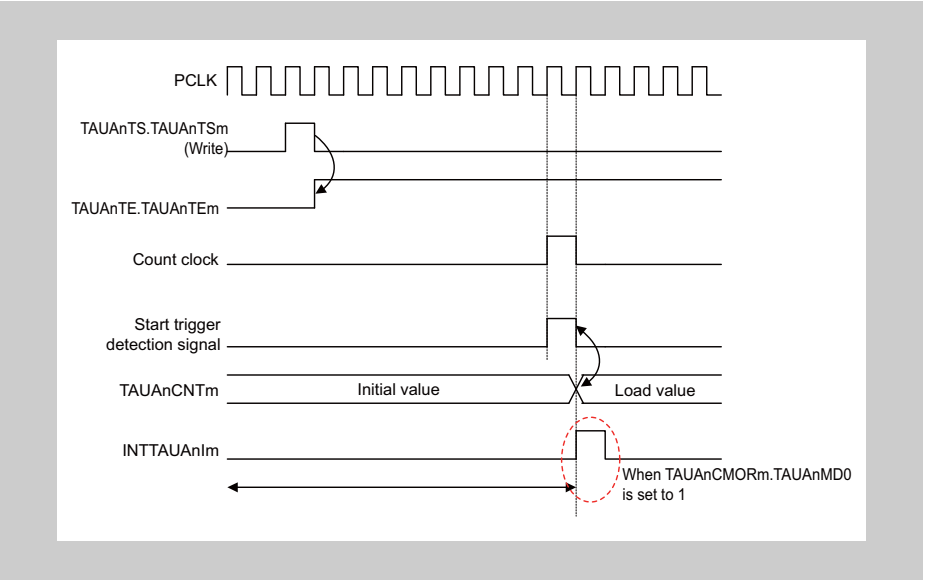


Figure 13-12 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode

13.9.2 Event Mode

The value of data register (TAUAnCDRm register) is loaded as soon as TAUAnTS.TAUAnTSm is set to 1. The counter also starts immediately. The value of data register (TAUAnCDRm register) increments with subsequent count clocks.

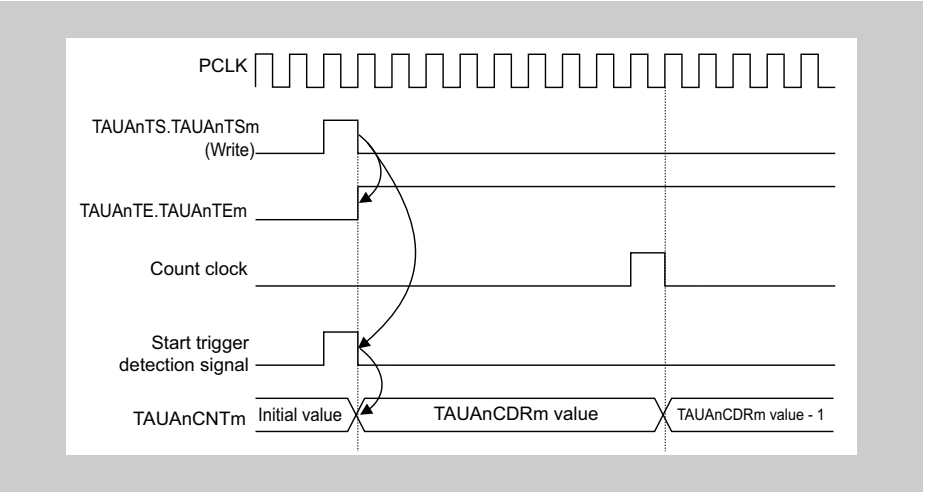


Figure 13-13 Start Timing in Event Mode

13.9.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUAnTTINm. Once the counter starts, the value of data register (TAUAnCDRm register) is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

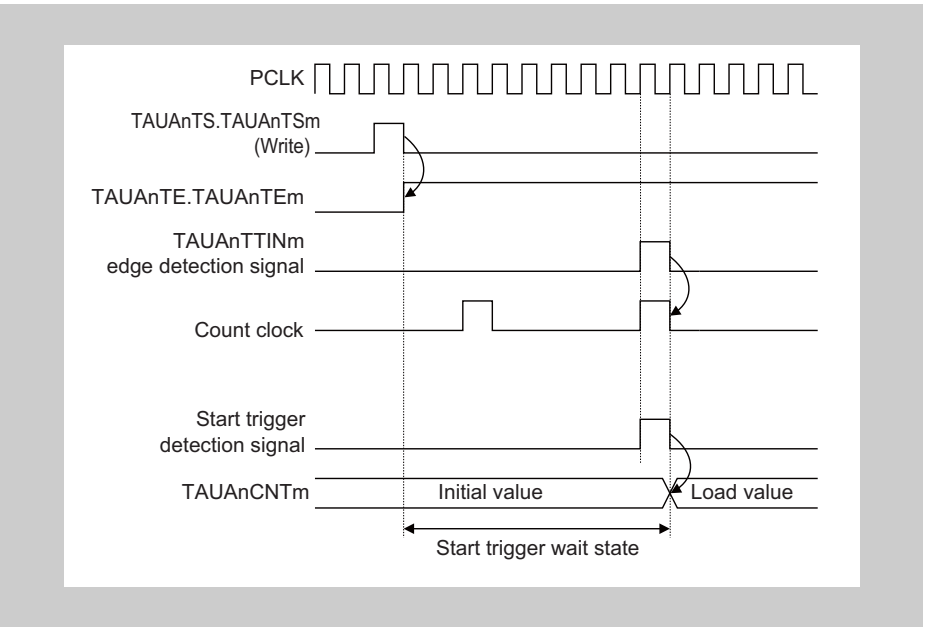


Figure 13-14 Start Timing in Other Operating Modes

13.10 TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUAnIm is to be generated using the TAUAnCMORm.TAUAnMD0 bit. The effect of the bit depends on the selected mode, as shown in the following table. The effects of INTTAUAnIm on TAUAnTTOUTm depend on the selected channel operation function.

Table 13-10 Effect of TAUAnCMORm.TAUAnMD0 Bit on Generation of INTTAUAnIm When Counter Is Triggered

Mode	TAUAnCMOR.MD0 Bit	INTTAUAnIm Generated When Counter Starts
Interval timer mode Capture mode Count capture mode	0	Not generated
	1	Generated
Capture and one-count mode Capture and gate count mode Event count mode Up/down count mode	0	Not generated
One-count mode Gate count mode	0/1	Not generated, regardless of setting of TAUAnCMORm.TAUAnMD0 bit
Pulse one-count mode		Generated, regardless of setting of TAUAnCMORm.TAUAnMD0 bit

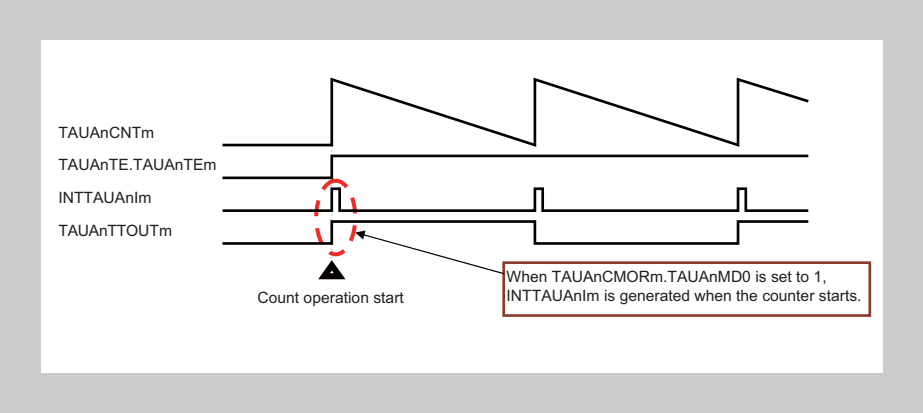


Figure 13-15 INTTAUAnIm Generated when Counter Starts

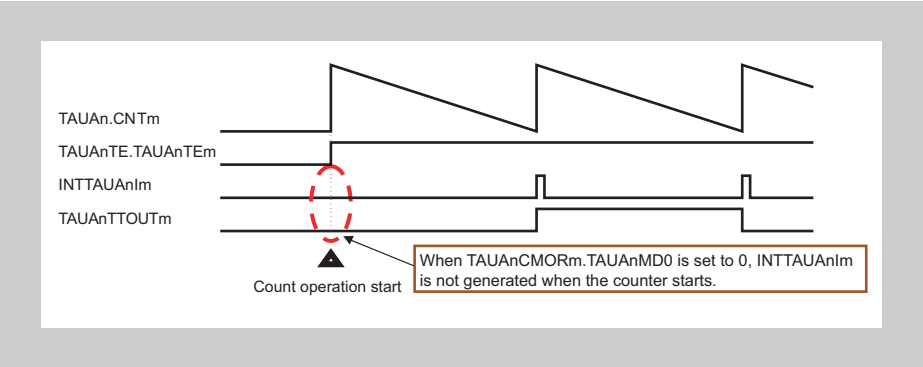


Figure 13-16 INTTAUAnIm Not Generated when Counter Starts

13.11 TAUAnTTINm Edge Detection

TAUAnTTINm edge is detected at the rising edge of a sampling clock. A maximum delay of one sampling clock cycle may occur.

The following figure shows an image of edge detection timing.

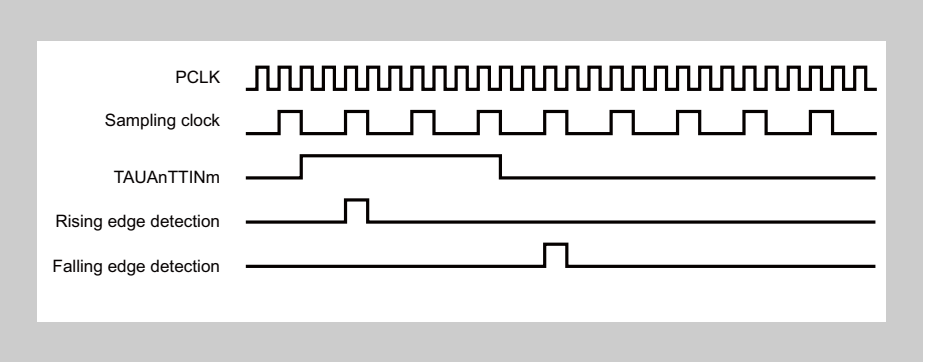


Figure 13-17 Basic Edge Detection Timing

Caution *figure13-17 Basic Edge Detection Timing,* shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUAnIm pin and TAUAn causes a delay time.

- When using a noise filter

  Noise filter delay time + edge detection delay time (maximum one sampling clock)
- When not using a noise filter

  Synchronization circuit delay time (maximum two PCLK) + edge detection delay time (maximum one sampling clock)

13.12 Assigning DMA Window Addresses

DMA (direct memory access) can be used to store the current values of TAUAnCDRm and TAUAnTOL registers in the TAUAnDWR registers.

The following figure shows how to rewrite the period register, duty registers, and TAUAnTOL register to the TAUAnDWR registers.

1. The addresses of the selected registers (TAUAnCDR0, TAUAnCDR2, TAUAnCDR4, TAUAnCDR6, TAUAnCDR9, TAUAnCDR13, TAUAnTOL.TAUAnTOLm, and TAUAnRDT.TAUAnRDTm) are loaded into the TAUAnDAS registers.
2. Then, the TAUAnDMA window address function loads the values of the selected registers into the corresponding TAUAnDWR registers.

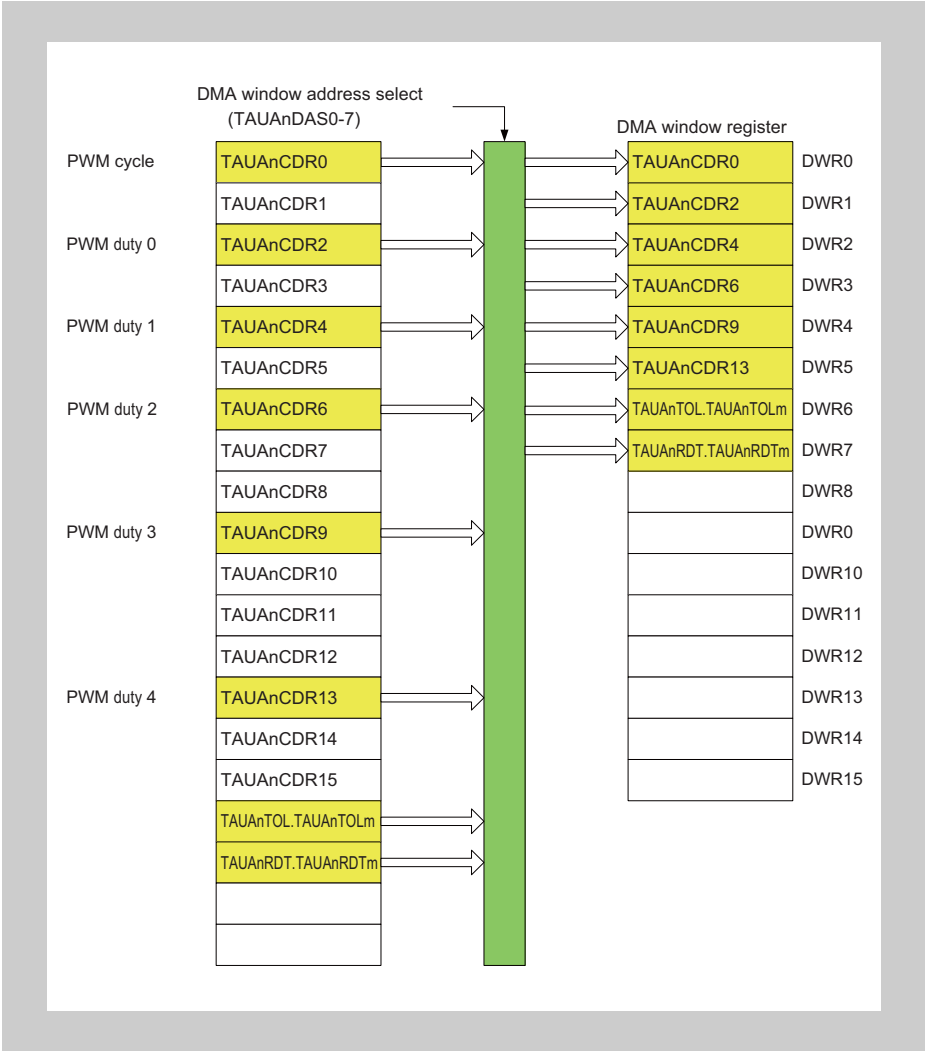


Figure 13-18 Assigning DMA Window Addresses

Note When using the simultaneous rewrite function for the data register (TAUAnCDRm register), it is recommended to allocate the TAUAnRDT register at the end of the specified area.

13.13 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUA. For a general overview of independent channel operation, see section 13.3, Functional Description.

13.13.1 Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUAnIm) at regular intervals. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites
- The operating mode should be set to interval timer mode. (See table 13-11, TAUAnCMORm Settings for Interval Timer Function.)
- The channel output mode should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes.)

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. The TAUAnCDRm value is loaded in TAUAnCNTm and subsequently operation continues.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm stop but retain their values. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm to 1 during operation.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUAnTTOUTm does not toggle. This results in an inverted TAUAnTTOUTm signal compared to when TAUAnCMORm.TAUAnMD0 is set to 1. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

(2) Equations

INTTAUAnIm cycle = count clock cycle × (TAUAnCDRm + 1)

TAUAnTTOUTm square wave cycle = count clock cycle ×

(TAUAnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

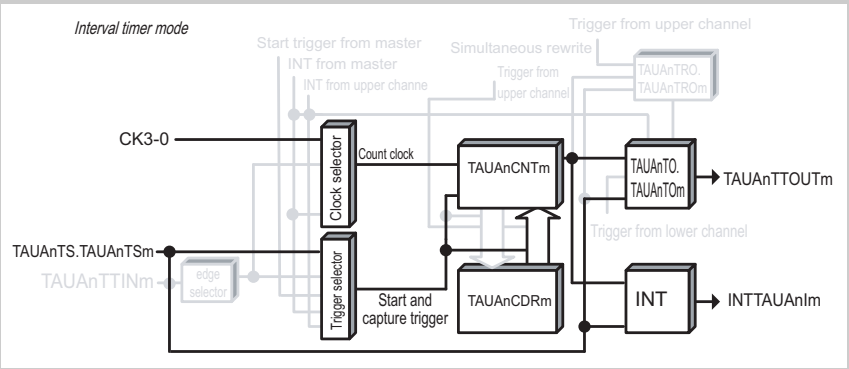


Figure 13-19 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 1)

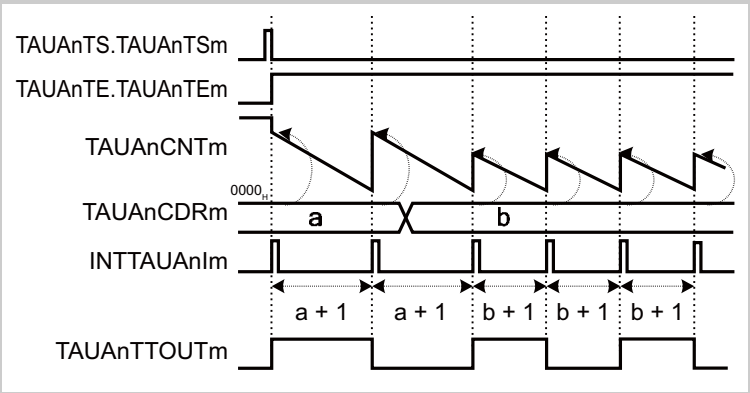


Figure 13-20 General Timing Diagram of Interval Timer Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-11 TAUAnCMORm Settings for Interval Timer Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated to toggle TAUAnTTOUTm at the beginning of an operation. 1: INTTAUAnIm is generated to toggle TAUAnTTOUTm at the beginning of an operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-12 TAUAnCMURm Settings for Interval Timer Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode

Table 13-13 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables independent channel output mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode if TAUAnTOM.TOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details, see section 13.8, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

Table 13-14 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for Interval Timer Function

Table 13-15 Operating Procedure for Interval Timer Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-11, TAUAnCMORm Settings for Interval Timer Function, and table 13-12, TAUAnCMURm Settings for Interval Timer Function. Set the value of TAUAnCDRm register. Set channel output mode by setting the control bits as described in table 13-13, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. The TAUAnCDRm value is loaded in TAUAnCNTm. When TAUAnCMORm.MD0 = 1, INTTAUAnIm is generated and TAUAnTTOUTm toggles.
During Operation	The TAUAnCDRm register value can be changed at any time. The TAUAnCNTm register can be read at all times.	TAUAnCNTm counts down. When the counter reaches 0000_H. • The TAUAnCDRm value is loaded in TAUAnCNTm again and count operation continues. • INTTAUAnIm is generated and TAUAnTTOUTm toggles.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(6) Specific Timing Diagrams

(a) TAUAnCDRm = 0000_H, Count Clock = PCLK/2

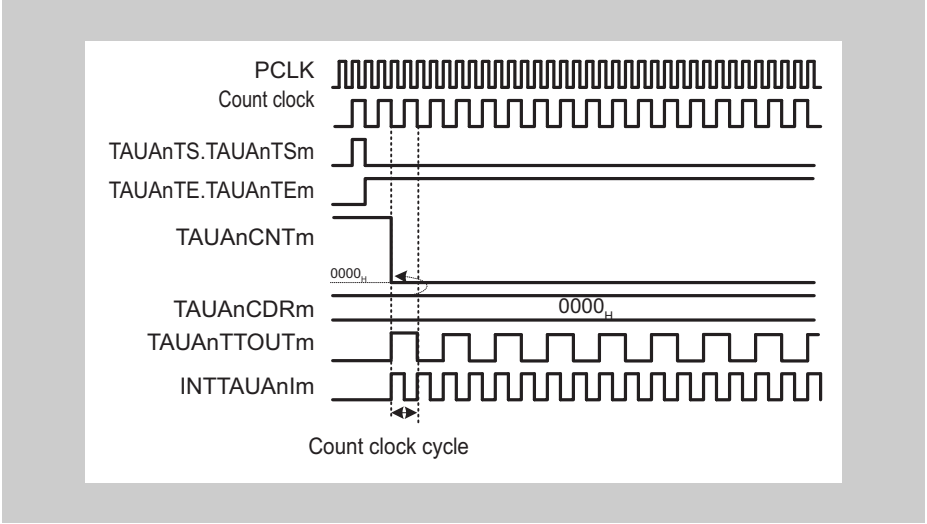


Figure 13-21 TAUAnCDRm = 0000_H, Count Clock = PCLK/2

- If TAUAnCDRm = 0000_H and the count clock = PCLK/2^1, the TAUAnCDRm value is loaded into TAUAnCNTm every count clock, meaning that TAUAnCNTm is always 0000_H.
- INTTAUAnIm is generated every count clock, resulting in TAUAnTTOUTm toggling every count clock.

(b) TAUAnCDRm = 0000_H, Count Clock = PCLK

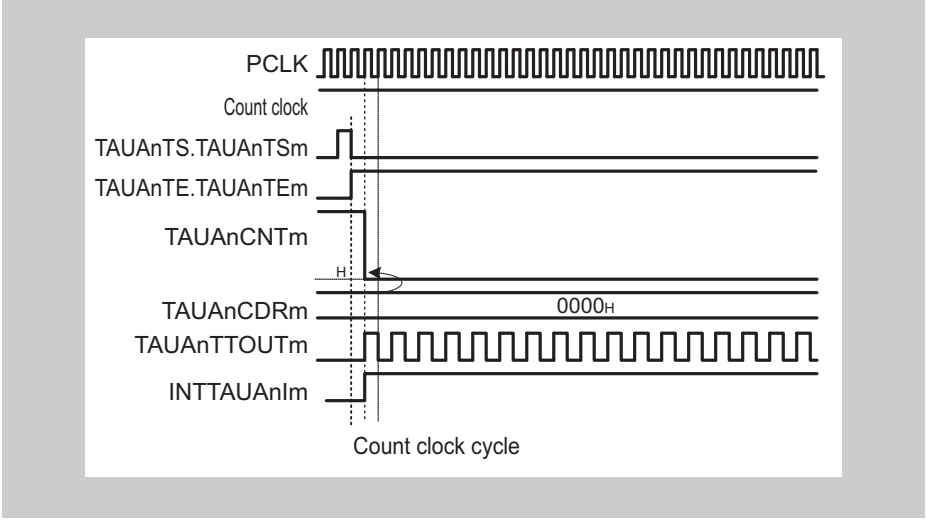


Figure 13-22 TAUAnCDRm = 0000_H, Count Clock = PCLK

- If TAUAnCDRm = 0000_H and the count clock = PCLK, the TAUAnCDRm value is loaded into TAUAnCNTm every PCLK clock, meaning that TAUAnCNTm is always 0000_H.
- INTTAUAnIm is generated continuously, resulting in TAUAnTTOUTm toggling every PCLK clock.

(c) Operation Stop and Restart

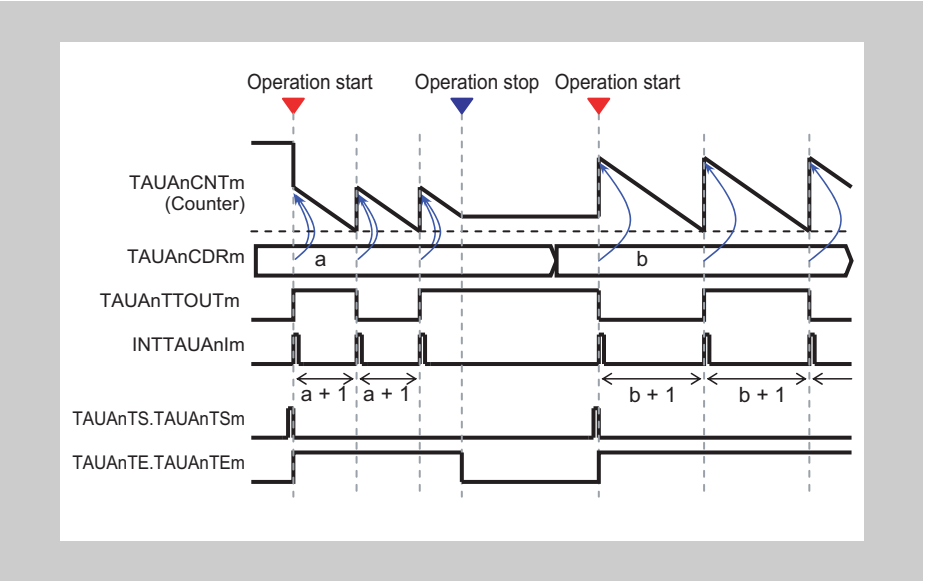


Figure 13-23 Operation Stop and Restart (TAUAnCMORm.TAUAnMD0 = 1)

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm and TAUAnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

(d) Forced Restart

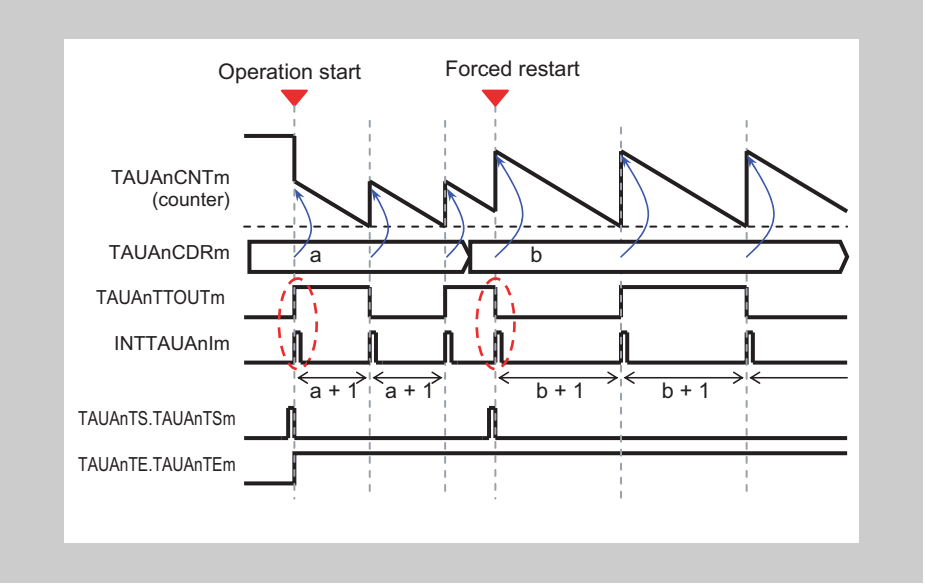


Figure 13-24 Forced Restart Operation (TAUAnCMORm.TAUAnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm to 1 during operation.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 1, the first interrupt after a start or restart is generated.

13.13.2 TAUAnTTINm Input Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUAnIm) at regular intervals or when a valid TAUAnTTINm input edge is detected. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to interval timer mode. See table 13-16, TAUAnCMORm Settings for TAUAnTTINm Input Interval Timer Function.
- The channel output mode should be set to independent channel output mode 1. See section 13.8, Channel Output Modes.

Description This function operates in an identical manner to the interval timer function (see section 13.13.1, Interval Timer Function) except that this function is restarted by a valid TAUAnTTINm input edge. The type of edge used as a trigger is specified using the TAUAnCMURm.TAUAnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

(2) Equations

INTTAUAnIm cycle = count clock cycle × (TAUAnCDRm + 1)

TAUAnTTOUTm square wave cycle = count clock cycle × (TAUAnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

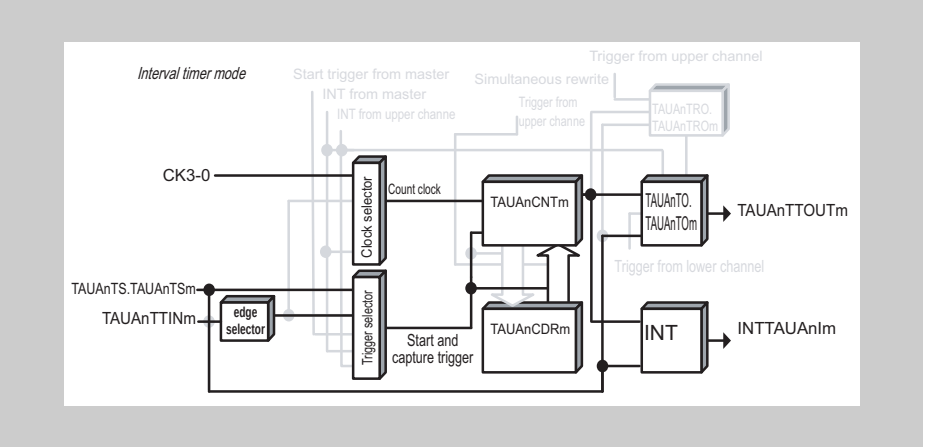


Figure 13-25 Block Diagram of TAUAnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 1)
- Rising edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

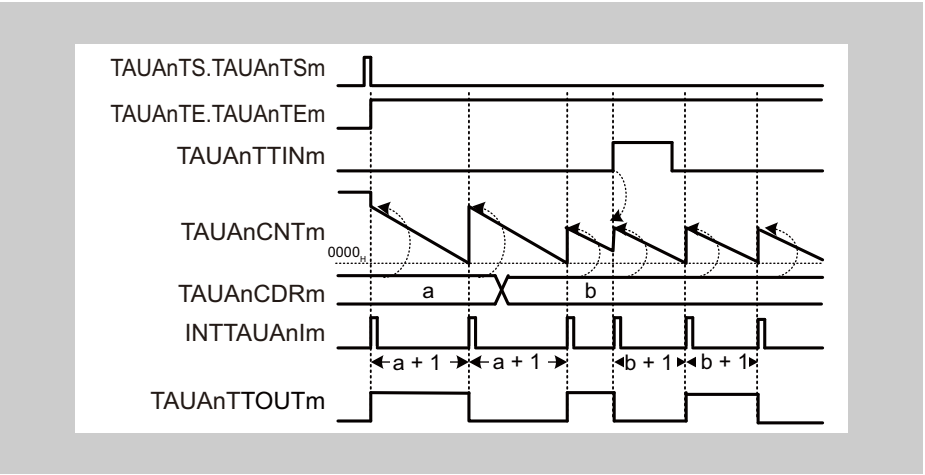


Figure 13-26 General Timing Diagram of TAUAnTTINm Input Interval Timer Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-16 TAUAnCMORm Settings for TAUAnTTINm Input Interval Timer Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated to toggle TAUAnTTOUTm at the beginning of an operation. 1: INTTAUAnIm is generated to toggle TAUAnTTOUTm at the beginning of an operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-17 TAUAnCMURm Settings for TAUAnTTINm Input Interval Timer Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges

(c) Channel Output Mode

Table 13-18 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables independent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode = 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details, see section 13.8, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 13-19 Simultaneous Rewrite Settings for TAUAnTTINm Input Interval Timer Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for TAUAnTTINm Input Interval Timer Function

Table 13-20 Operating Procedure for TAUAnTTINm Input Interval Timer Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-16, TAUAnCMORm Settings for TAUAnTTINm Input Interval Timer Function, and table 13-17, TAUAnCMURm Settings for TAUAnTTINm Input Interval Timer Function. Set the value of TAUAnCDRm register. Set channel output mode by setting the control bits as described in table 13-18, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. The TAUAnCDRm value is loaded in TAUAnCNTm. When TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated and TAUAnTTOUTm toggles.
During Operation	The values of the TAUAnCMURm.TAUAnTIS[1:0] and the TAUAnCDRm register are changeable at any time. The TAUAnCNTm register are readable at any time. Detection of TAUAnTTINm edge	TAUAnCNTm counts down. When the counter reaches 0000_H. • The TAUAnCDRm value is loaded in TAUAnCNTm again and count operation continues. • INTTAUAnIm is generated and TAUAnTTOUTm toggles. When a TAUAnTTINm input valid edge is detected during count operation, the TAUAnCDRm value is loaded in TAUAnCNTm and count operation continues. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(6) Specific Timing Diagram

The timing diagrams in section 13.13.1, Interval Timer Function. The counter can also be restarted by a valid TAUAnTTINm input edge without using this function.

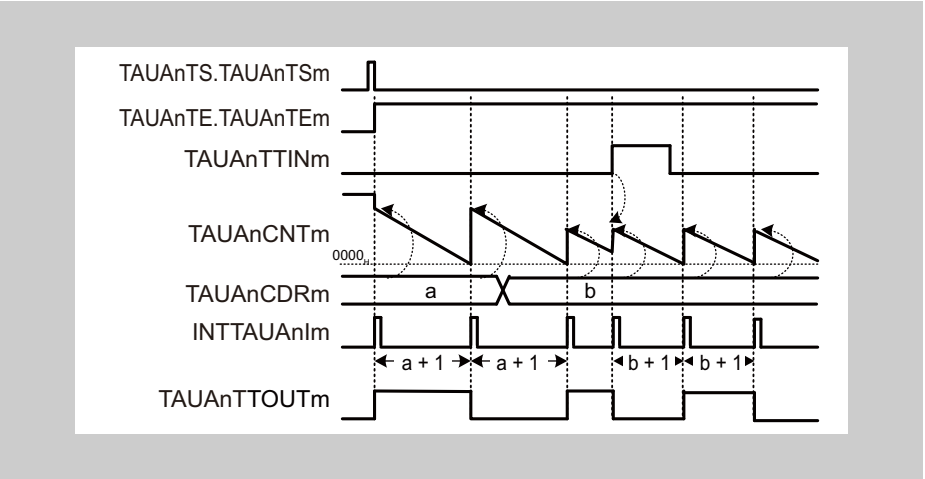


Figure 13-27 Counter Triggered by Rising TAUAnTTINm Input Edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B), TAUAnCMORm.TAUAnMD0 = 1

- If a valid TAUAnTTINm input edge is detected, an interrupt is generated which causes TAUAnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B).

13.13.3 Delay Count Function

(1) Overview

Summary This function generates interrupts (INTTAUAnIm), which have a defined delay to the TAUAnTTINm input signal. TAUAnTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites
- The operating mode should be set to one-count mode. See table 13-21, TAUAnCMORm Settings for Delay Count Function.
- TAUAnTTOUTm is not used with this function.
- Start trigger should be disabled during counting (TAUAnCMORm.TAUAnMD0 = 0).

Functional description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. The value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value.

When the counter reaches 0000_H, an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUAnTTINm input edge.

When the counter is counting down, further TAUAnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions The type of edge used as a trigger is specified by the TAUAnCMURm.TIS[1:0] bits:
- If TAUAnCMURm.TAUAnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

(2) Equations

Delay between TAUAnTTINm and INTTAUAnIm = count clock cycle × (TAUAnCDRm + 1)

(3) Block Diagram and General Timing Diagram

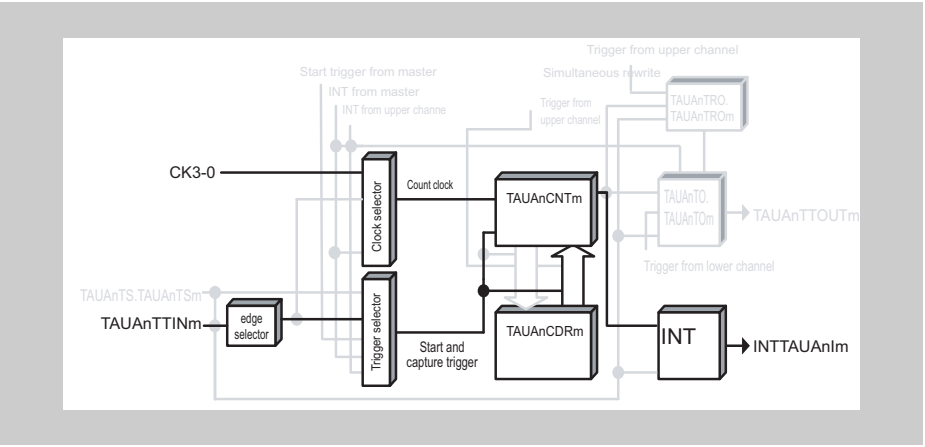


Figure 13-28 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0]= 00_B)

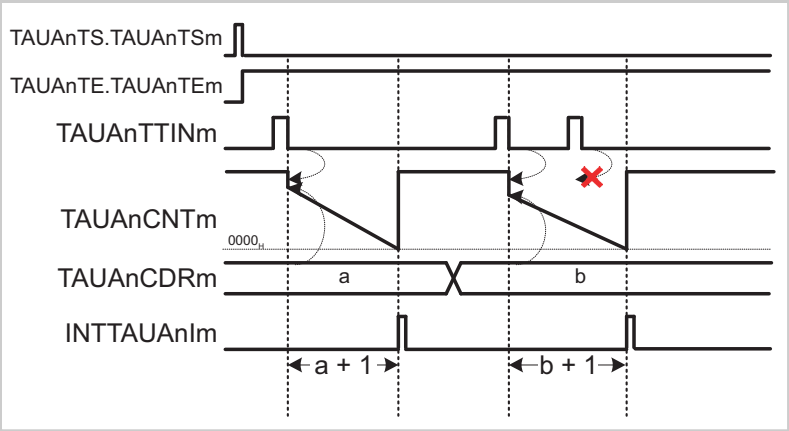


Figure 13-29 General Timing Diagram of Delay Count Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-21 TAUAnCMORm Settings for Delay Count Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	0: Disables a start trigger during operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-22 TAUAnCMURm Settings for Delay Count Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-23 Simultaneous Rewrite Settings for Delay Count Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for Delay Count Function

Table 13-24 Operating Procedure for Delay Count Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-21, TAUAnCMORm Settings for Delay Count Function, and table 13-22, TAUAnCMURm Settings for Delay Count Function. Set the value of TAUAnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, the TAUAnCDRm value is loaded in TAUAnCNTm.
During Operation	The values set in the TAUAnCDRm register are changeable at any time. The TAUAnCNTm register is readable at any time.	TAUAnCNTm counts down. When the counter reaches 0000_H: INTTAUAnIm is generated. TAUAnCNTm stops counting, returns $FFFF_H$, and waits for a trigger. If a trigger occurs while TAUAnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its value.

Restart

13.13.4 One-Pulse Output Function

(1) Overview

Summary

This function generates an interrupt (INTTAUAnIm) when a valid TAUAnTTINm input edge is detected and at a defined interval afterward. TAUAnTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to pulse one-count mode. (See table 13-25, TAUAnCMORm Settings for One-Pulse Output Function.*)*
- The channel output mode should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes.)
- Start trigger should be disabled during counting (TAUAnCMORm.TAUAnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input edge is detected. The value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value. An interrupt is generated and TAUAnTTOUTm toggles.

When the counter reaches 0001_H, an interrupt is generated and TAUAnTTOUTm is set to the inactive level. The counter stops at 0000_H and awaits the next valid TAUAnTTINm input edge.

When the counter is counting down, further TAUAnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:

- If TAUAnCMURm.TAUAnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

(2) Equations

Interval between TAUAnTTINm and INTTAUAnIm = TAUAnTTOUTm (timer output) width = count clock cycle × TAUAnCDRm

(3) Block Diagram and General Timing Diagram

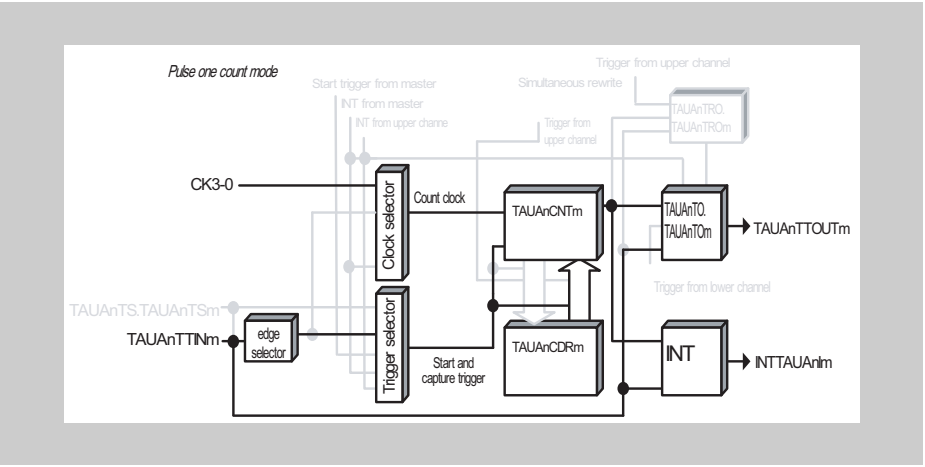


Figure 13-30 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

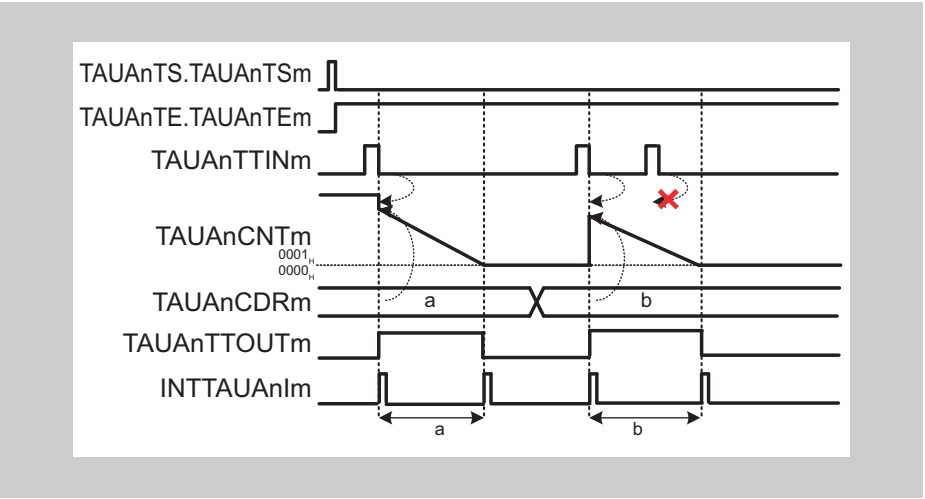


Figure 13-31 General Timing Diagram of One-Pulse Output Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-25 TAUAnCMORm Settings for One-Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1010: Pulse one-count mode
TAUAnMD0	0: Disables a start trigger during operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-26 TAUAnCMURm Settings for One-Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

Table 13-27 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables independent channel output mode controlled by software.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Set/reset mode
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Note The channel output mode can also be set to channel output mode controlled by software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details, see table 13-9, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

Table 13-28 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for One-Pulse Output Function

Table 13-29 Operating Procedure for One-Pulse Output Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-25, TAUAnCMORm Settings for One-Pulse Output Function, and table 13-26, TAUAnCMURm Settings for One-Pulse Output Function. Set the value of TAUAnCDRm register. Set channel output mode by setting the control bits as described in table 13-27, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, TAUAnCNTm loads the TAUAnCDRm value.
During Operation	The value of TAUAnCDRm is changeable at any time. The TAUAnCNTm register are readable at any time.	INTTAUAnIm is generated when TAUAnCNTm starts and TAUAnTTOUTm is set to its active level. TAUAnCNTm counts down. When the counter reaches 0001_H: • INTTAUAnIm is generated. • TAUAnTTOUTm is set to its inactive level. TAUAnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUAnCNTm is counting, the trigger is ignored.
Stop Operation	Set TAUAnTT.TTm to 1. TAUAnTT.TTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values

Restart

13.13.5 TAUAnTTINm Input Pulse Interval Measurement Function

(1) Overview

Summary This function captures the count value and uses this value and the overflow bit TAUAnCSRm.TAUAnOVF to measure the interval of the TAUAnTTINm input signal

Prerequisites
- The operating mode should be set to capture mode. See table 13-31, TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Measurement Function.
- TAUAnTTOUTm is not used with this function.

Functional description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter TAUAnCNTm starts to count up from 0000_H. When a valid TAUAnTTINm edge is detected, the value of TAUAnCNTm is captured, transferred to TAUAnCDRm, and an interrupt INTTAUAnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUAnTTINm edge is detected, it overflow. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUAnCDRm and TAUAnCSRm.TAUAnOVF respectively depend on the values of bits TAUAnCMORm.TAUAnCOS[1:0].

Table 13-30 Effects of Overflow

TAUAnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUAnTTINm Input is Detected	
	TAUAnCDRm	TAUAnCSRm.TAUAnOVF	TAUAnCDRm, TAUAnCNTm	TAUAnCSRm.TAUAnOVF
00	Unchanged	0	TAUAnCNTm loaded into TAUAnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUAnCNTm set to 0, TAUAnCDRm unchanged	0
11		1		

When TAUAnCMORm.TAUAnCOS[0] = 1, the overflow bit (TAUAnCSRm.TAUAnOVF) can be cleared only by setting TAUAnCSCm.TAUAnCLOV = 1.

The combination of the value of TAUAnCDRm and TAUAnCSRm.TAUAnOVF can be used to deduce the interval of the TAUAnTTINm signal. However, if an overflow occurs multiple times before a valid TAUAnTTINm input is detected, the overflow bit TAUAnCSRm.TAUAnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUAnTT.TAUAnTTm = 1. This sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm stops but retains its value. While the function is stopped, valid TAUAnTTINm input edge detection and TAUAnCNTm capture are not performed.

The counter can be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm = 1 during counting.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

Note When TAUAnCMORm.TAUAnCOS[1] = 1, the value of TAUAnCNTm is not loaded into TAUAnCDRm when the first valid TAUAnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUAnTTINm input pulse interval = count clock cycle × [(TAUAnCSRm.TAUAnOVF × ($FFFF_H$+1)) + TAUAnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

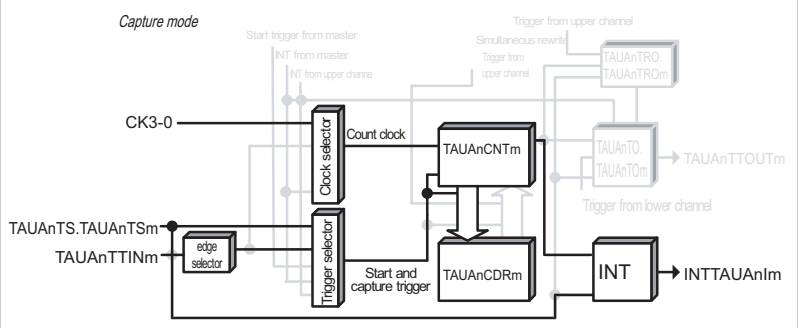


Figure 13-32 Block Diagram of TAUAnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUAnIm not generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 0)
- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)
- When a valid TAUAnTTINm input is detected after an overflow, TAUAnCDRm is changed and TAUAnCSRm.TAUAnOVF is set to 1 (TAUAnCMORm.TAUAnCOS[1:0]= 00_B)

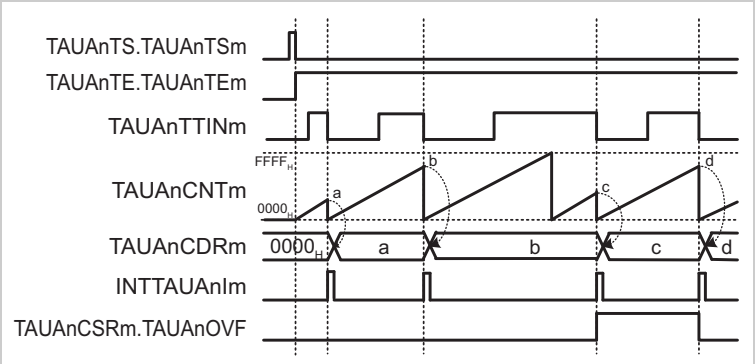


Figure 13-33 General Timing Diagram of TAUAnTTINm Input Pulse Interval Measurement Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-31 TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is the external capture trigger.
TAUAnCOS[1:0]	See table 13-30, Effects of Overflow.
TAUAnMD[4:1]	0010: Capture mode
TAUAnMD0	0: INTTAUAnIm not generated at the beginning of operation. 1: INTTAUAnIm generated at the beginning of operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-32 TAUAnCMURm Settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

Table 13-33 Simultaneous Rewrite Settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disableing simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for TAUAnTTINm Input Pulse Interval Measurement Function

Table 13-34 Operating Procedure for TAUAnTTINm Input Pulse Interval Measurement Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-31, TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Measurement Function, and table 13-32, TAUAnCMURm Settings for TAUAnTTINm Input Pulse Interval Measurement Function. TAUAnCDRm register operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm is cleared to 0000_H. INTTAUAnIm is generated when TAUAnCMORm.MD0 is set to 1.
During Operation	Detection of TAUAnTTINm edges The TAUAnCMURm.TAUAnTIS[1:0] bits can be changed at any time. The TAUAnCDRm and TAUAnCSRm registers can be read at any time. TAUAnCSCm.TAUAnCLOV can be written to 1. (TAUAnCSRm.TAUAnOVF bit is cleared to 0.)	TAUAnCNTm starts to count up from 0000_H. When a TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm, and returns to 0000_H. • INTTAUAnIm is then generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TTm to 1. TAUAnTT.TTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific Timing Diagrams: Overflow Operation

(a) TAUAnCMORm.TAUAnCOS[1:0] = 00_B

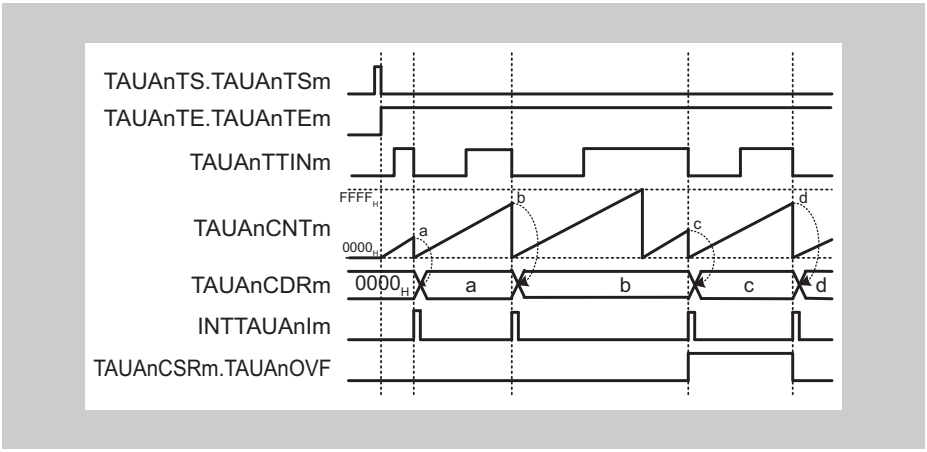


Figure 13-34 TAUAnCMORm.TAUAnCOS[1:0] = 00_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded into TAUAnCDRm and TAUAnCSRm.TAUAnOVF is set to1.
- Upon detection of the next valid TAUAnTTINm input edge with no overflow occurring, TAUAnCSRm.TAUAnOVF is cleared to 0.

(b) TAUAnCMORm.TAUAnCOS[1:0] = 01_B

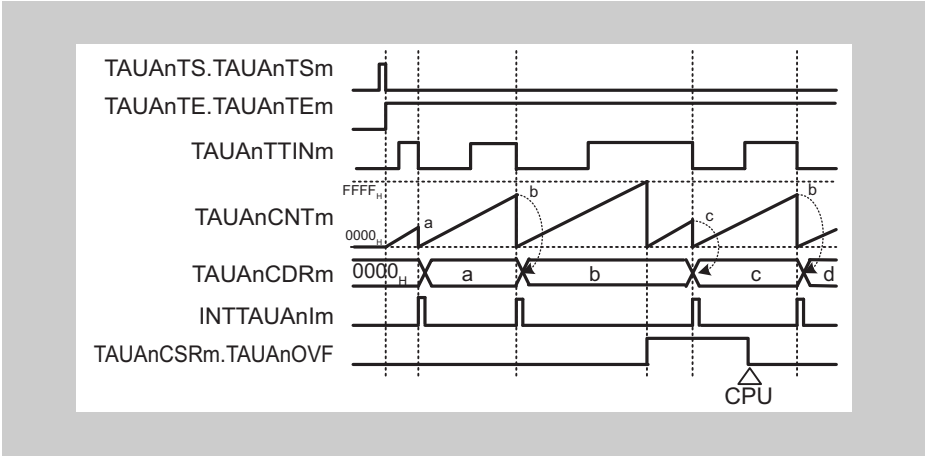


Figure 13-35 TAUAnCMORm.TAUAnCOS[1:0] = 01_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF is set to1.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded into TAUAnCDRm.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV bit to 1.

(c) TAUAnCMORm.TAUAnCOS[1:0] = 10_B

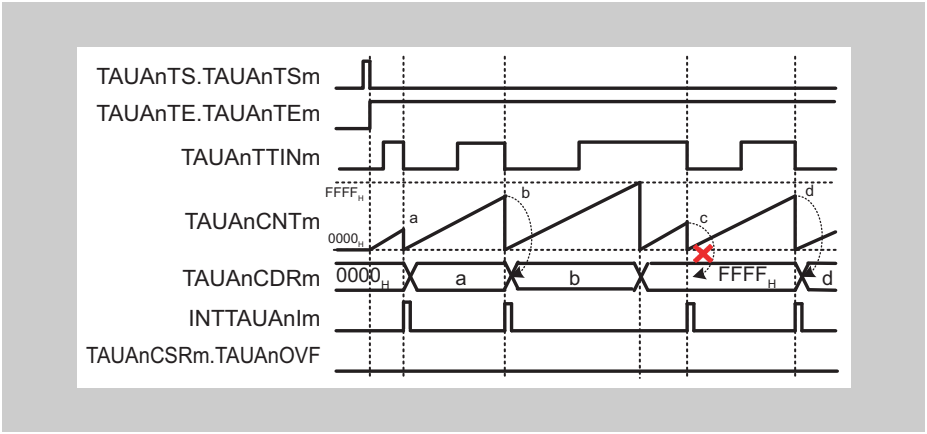


Figure 13-36 TAUAnCMORm.TAUAnCOS[1:0] = 10_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next valid TAUAnTTINm input edge after the overflow is ignored.

(d) TAUAnCMORm.TAUAnCOS[1:0] = 11_B

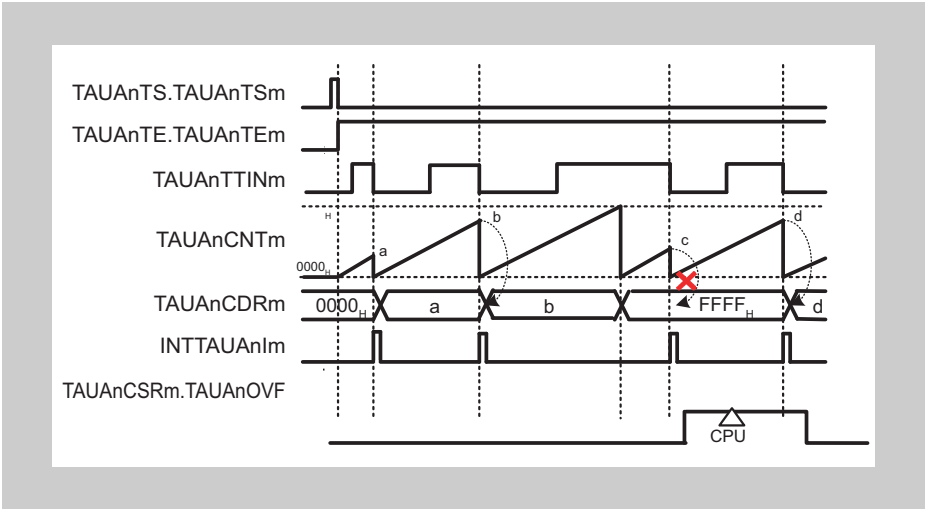


Figure 13-37 TAUAnCMORm.TAUAnCOS[1:0] = 11_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next valid TAUAnTTINm input edge after the overflow is ignored.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV to 1.

13.13.6 TAUAnTTINm Input Signal Width Measurement Function

(1) Overview

Summary This function measures the width of a TAUAnTTINm input signal.

Prerequisites
- The operating mode should be set to capture and one-count mode. See table 13-36, TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Measurement Function.
- TAUAnTTOUTm is not used with this function.
- TAUAnCMORm.TAUAnMD0 should be set to 0.

Functional description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When a valid TAUAnTTINm start edge is detected, the counter TAUAnCNTm starts to count up from 0000_H. When a valid TAUAnTTINm stop edge is detected, the value of TAUAnCNTm is captured, transferred to TAUAnCDRm, and an interrupt INTTAUAnIm is generated. The counter retains its value and awaits the next valid TAUAnTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUAnTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUAnCDRm and TAUAnCSRm.TAUAnOVF respectively depend on the values of bits TAUAnCMORm.TAUAnCOS[1:0].

Table 13-35 Effects or Overflow

TAUAnCMORm. COS[1:0]	When Overflow Occurs		When a Valid TAUAnTTINm Input Stop Edge is Detected	
	TAUAnCDRm	TAUAnCSRm. TAUAnOVF	TAUAnCDRm, TAUAnCNTm	TAUAnCSRm. TAUAnOVF
00	Unchanged	0	TAUAnCNTm loaded into TAUAnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUAnCNTm stops counting TAUAnCDRm unchanged	0
11		1		

When TAUAnCMORm.TAUAnCOS[0] = 1, overflow bit TAUAnCSRm.TAUAnOVF can be cleared only by setting TAUAnCSCm.TAUAnCLOV to 1.

The combination of the value of TAUAnCDRm and TAUAnCSRm.TAUAnOVF can be used to deduce the width of the TAUAnTTINm signal. However, if an overflow occurs multiple times before a valid TAUAnTTINm input is detected, overflow bit TAUAnCSRm.TAUAnOVF cannot indicate the occurrence of multiple overflows.

Caution This function cannot be forcibly restarted.

(2) Equations

TAUAnTTINm input signal width = count clock cycle × [(TAUAnCSRm.TAUAnOVF × ($FFFF_H$ + 1)) + TAUAnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

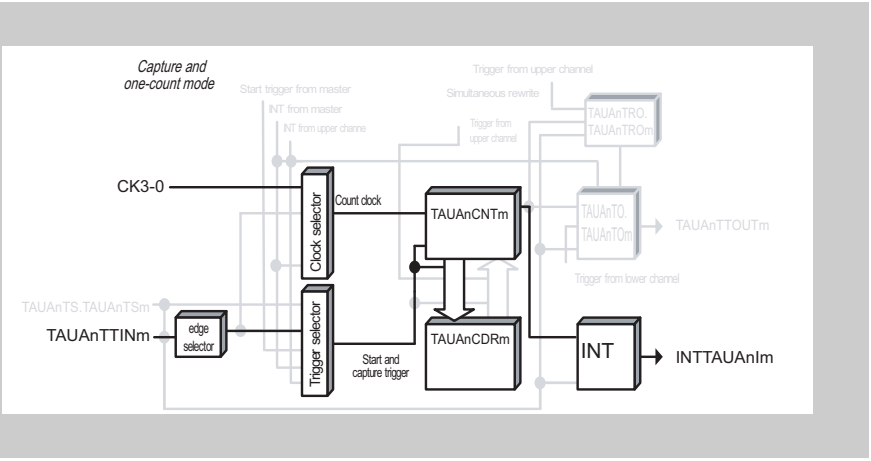


Figure 13-38 Block Diagram of TAUAnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Detection of falling and rising edges = High width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)
- When a valid TAUAnTTINm input is detected after an overflow, TAUAnCDRm is changed and TAUAnCSRm.TAUAnOVF is set to 1. (TAUAnCMORm.TAUAnCOS[1:0] = 00_B)

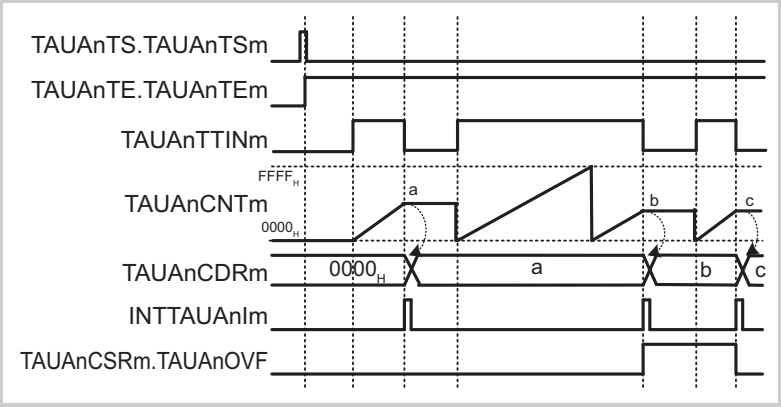


Figure 13-39 General Timing Diagram of TAUAnTTINm Input Signal Width Measurement Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-36 TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
TAUAnCOS[1:0]	See table 13-35, Effects or Overflow.
TAUAnMD[4:1]	0110: Capture and one-count mode
TAUAnMD0	0: Disables the start trigger during operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-37 TAUAnCMURm Settings for TAUAnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUAnTIS[1:0]	10: Detection of falling and rising edges (Low width measurement) 11: Detection of falling and rising edges (High width measurement)

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm input signal width measurement function. Therefore, these registers should be set to 0.

Table 13-38 Simultaneous Rewrite Settings for TAUAnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0:Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for TAUAnTTINm Input Signal Width Measurement Function

Table 13-39 Operating Procedure for TAUAnTTINm Input Signal Width Measurement Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-36, TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Measurement Function, and table 13-37, TAUAnCMURm Settings for TAUAnTTINm Input Signal Width Measurement Function. TAUAnCDRm register operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of TAUAnTTINm start edge. When a TAUAnTTINm start edge is detected, TAUAnCNTm starts to count up.
During Operation	Detection of TAUAnTTINm edges TAUAnCDRm, TAUAnCNTm, and TAUAnCSRm registers can be read at any time. TAUAnCSC.TAUAnCLOV bit can be set to 1.	TAUAnCNTm starts to count up from 0000_H. When TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm, and retains its value. • INTTAUAnIm is then generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific Timing Diagrams: Overflow Operation

(a) TAUAnCMORm.TAUAnCOS[1:0] = 00_B

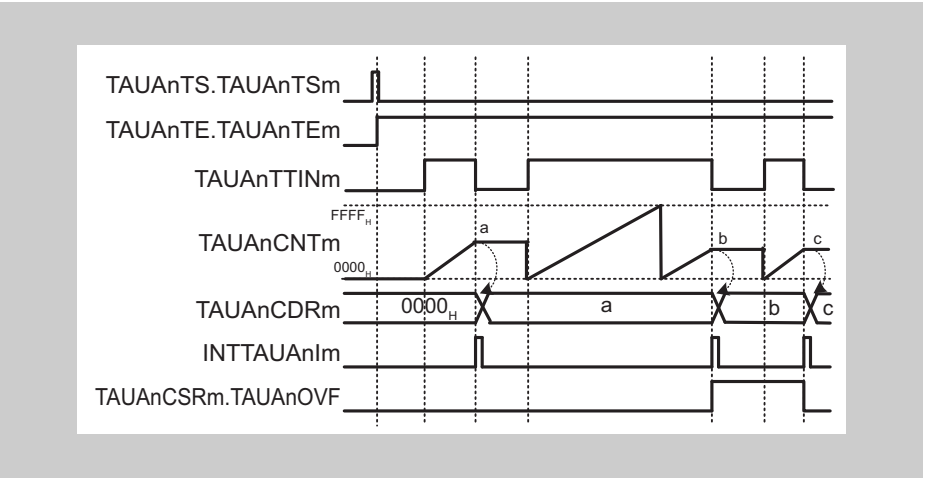


Figure 13-40 TAUAnCMORm.TAUAnCOS[1:0] = 00_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded into TAUAnCDRm and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge with no overflow occurring, TAUAnCSRm.TAUAnOVF is cleared to 0.

(b) TAUAnCMORm.TAUAnCOS[1:0] = 01_B

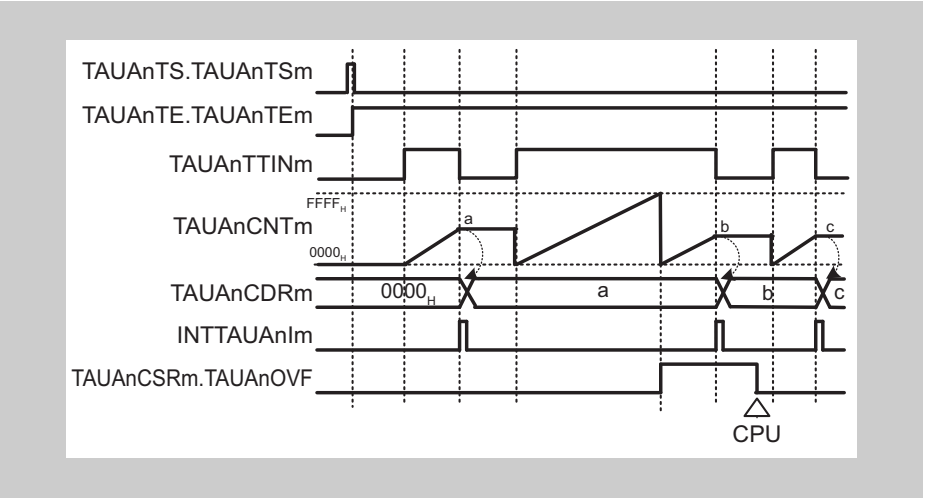


Figure 13-41 TAUAnCMORm.TAUAnCOS[1:0] = 01_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded into TAUAnCDRm.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV bit to 1.

(c) TAUAnCMORm.TAUAnCOS[1:0] = 10_B

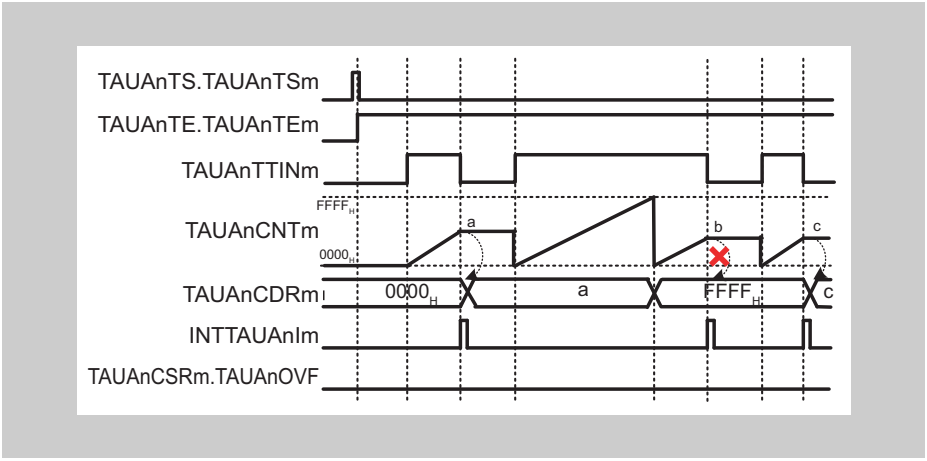


Figure 13-42 TAUAnCMORm.TAUAnCOS[1:0] = 10_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next TAUAnTTINm input valid edge after the overflow is ignored.

(d) TAUAnCMORm.TAUAnCOS[1:0] = 11_B

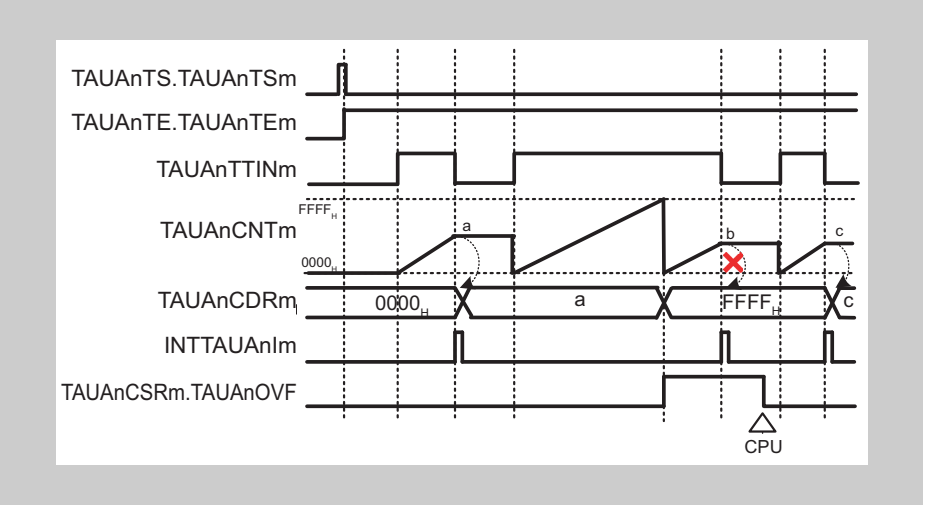


Figure 13-43 TAUAnCMORm.TAUAnCOS[1:0] = 11_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next valid TAUAnTTINm input edge after the overflow is ignored.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV to 1.

13.13.7 TAUAnTTINm Input Period Count Detection Function

(1) Overview

Summary This function measures the cumulative width of a TAUAnTTINm input signal.

Prerequisites
- The operating mode should be set to capture and gate count mode. (See table 13-40, TAUAnCMORm Settings for TAUAnTTINm Input Period Count Detection Function.)
- TAUAnTTOUTm is not used with this function.

Functional description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter awaits a valid TAUAnTTINm input edge.

When a valid TAUAnTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUAnTTINm input stop edge is detected, the current TAUAnCNTm value is loaded into TAUAnCDRm and an interrupt (INTTAUAnIm) is generated. The counter stops and retains its value until the next valid TAUAnTTINm input start edge is detected.

When the next valid TAUAnTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches $FFFF_H$, TAUAnCSRm.TAUAnOVF bit is set to 1 and the counter restarts to count from 0000_H.

TAUAnCSRm.TAUAnOVF value is cleared by setting TAUAnCSCm.TAUAnCLOV to 1.

Note TAUAnTTINm input signal is sampled at the frequency of a sampling clock set by the TAUAnCMORm.TAUAnCKS[1:0] bits.

Conditions The valid start and stop edges are specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, the TAUAnTTINm input low period is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUAnCMURm.TAUAnTIS[1:0] = 11_B, the TAUAnTTINm input high period is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

(2) Equations

Cumulative TAUAnTTINm input width = count clock cycle × [(TAUAnCSRm.TAUAnOVF × ($FFFF_H$ + 1)) + (TAUAnCDRm capture value + 1)]

(3) Block Diagram and General Timing Diagram

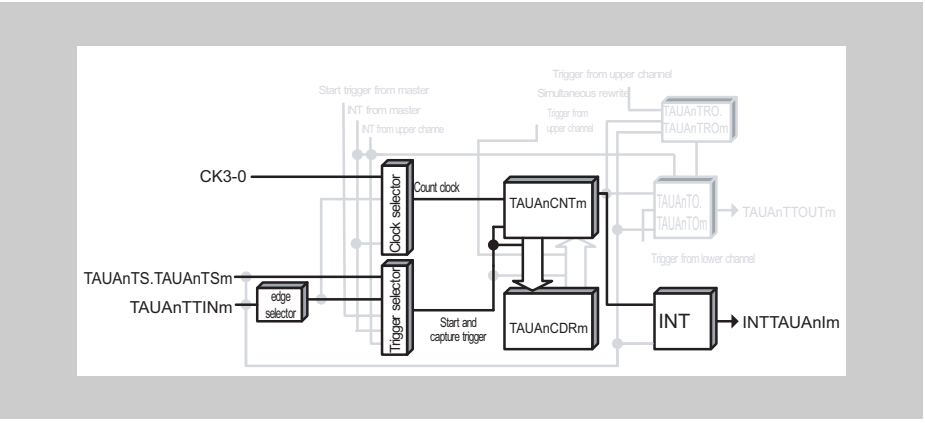


Figure 13-44 Block Diagram of TAUAnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of falling and rising edges = High width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

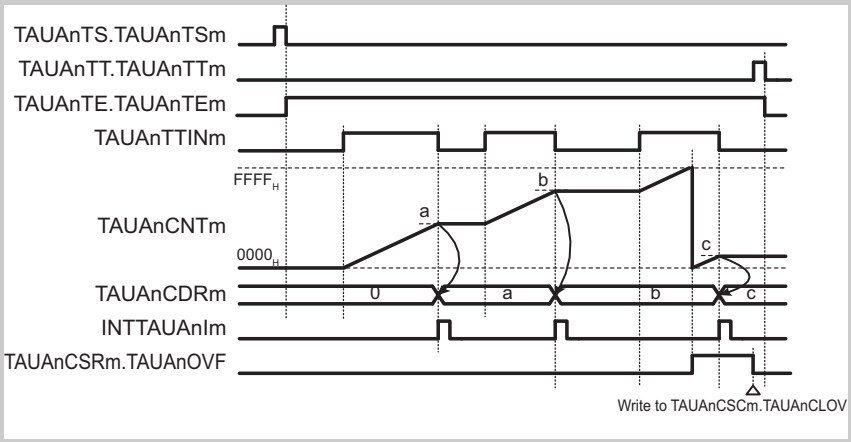


Figure 13-45 General Timing Diagram of TAUAnTTINm Input Period Count Detection Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-40 TAUAnCMORm Settings for TAUAnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigge.
TAUAnCOS[1:0]	01: Overflow (TAUAnCSRm.TAUAnOVF) is set when the counter overflows. This is cleared by setting TAUAnCSCm.TAUAnCLOV to 1.
TAUAnMD[4:1]	1101: Capture and gate count mode
TAUAnMD0	0: Disables a start trigger during operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-41 TAUAnCMURm Settings for TAUAnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUAnTIS[1:0]	10: Detection of falling and rising edges (Low width measurement) 11: Detection of falling and rising edges (High width measurement)

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-42 Simultaneous Rewrite Settings for TAUAnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for TAUAnTTINm Input Period Count Detection Function

Table 13-43 Operating Procedure for TAUAnTTINm Input Period Count Detection Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-40, TAUAnCMORm Settings for TAUAnTTINm Input Period Count Detection Function, and table 13-41, TAUAnCMURm Settings for TAUAnTTINm Input Period Count Detection Function. TAUAnCDRm register operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, TAUAnCNTm is cleared to 0000_H and starts counting up.
During Operation	Detection of TAUAnTTINm edges The TAUAnCDRm, TAUAnCNTm, and TAUAnCSRm registers can be read at any time. The TAUAnCSC.TAUAnCLOV bit can be set to 1.	When a TAUAnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUAnCNTm starts counting up from the stop value. When TAUAnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUAnCDRm and INTTAUAnIm is generated. Counting stops at the "value transferred to TAUAnCDRm + 1" and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When TAUAnCNTm reaches $FFFF_H$, the counter overflows and TAUAnCSRm.TAUAnOVF is set to 1. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. TAUAnCNTm and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific Timing Diagrams

(a) Operation Stop and Restart

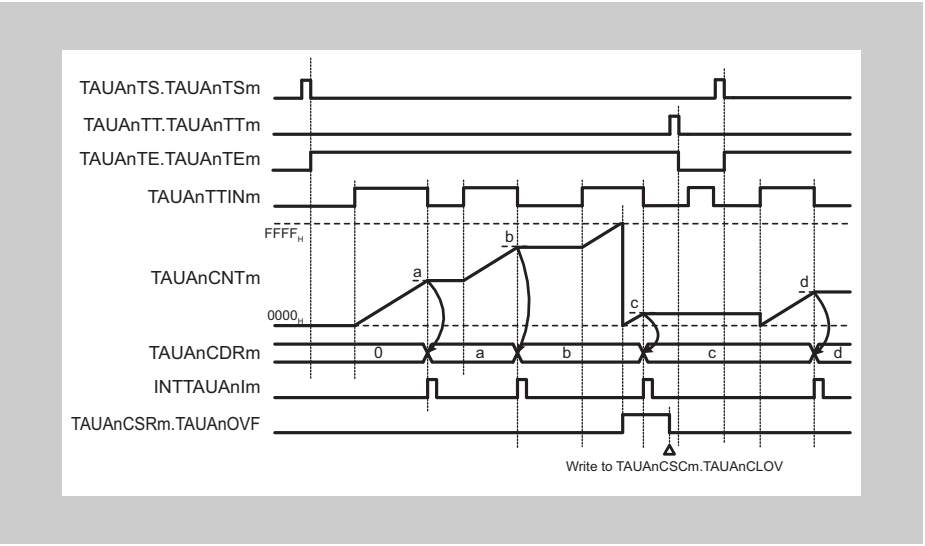


Figure 13-46 Operation Stop and Restart (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUAnTTINm input edges are ignored.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm restarts to count from 0000_H.

13.13.8 TAUAnTTINm Input Pulse Interval Judgment Function

(1) Overview

Summary

This function outputs the result of a comparison between the count value (TAUAnCNTm) and the value in the channel data register (TAUAnCDRm) when a TAUAnTTINm input pulse occurs. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge mode. See table 13-44, TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Judgment Function.
- TAUAnTTOUTm is not used with this function.

Functional description

The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from this value.

When a TAUAnTTINm valid edge is detected or TAUAnTS.TAUAnTSm is set to 1, the function compares the current values of TAUAnCNTm and TAUAnCDRm. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true. TAUAnCNTm reloads the value of TAUAnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUAnTTINm valid edge is detected, TAUAnCNTm underflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions

The TAUAnCMORm.TAUAnMD0 bit specifies the type of comparison:

- If TAUAnCMORm.TAUAnMD0 = 0, INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm.
- If TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm

(2) Block Diagram and General Timing Diagram

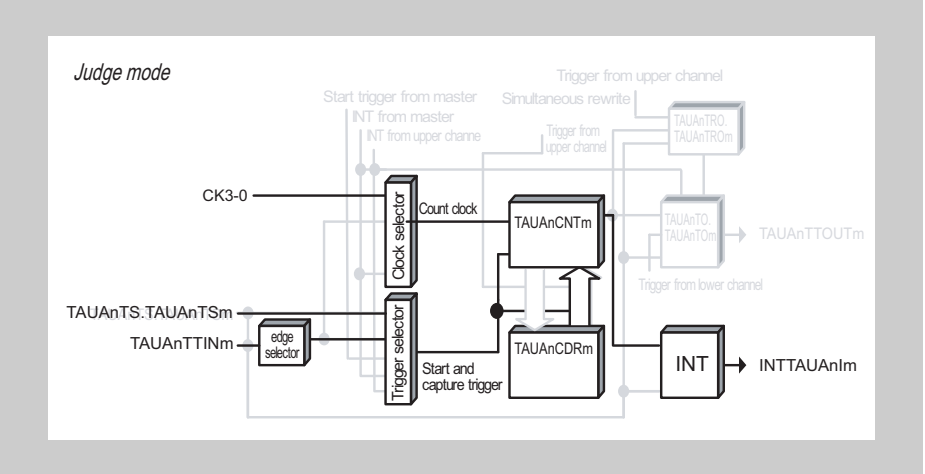


Figure 13-47 Block Diagram of TAUAnTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

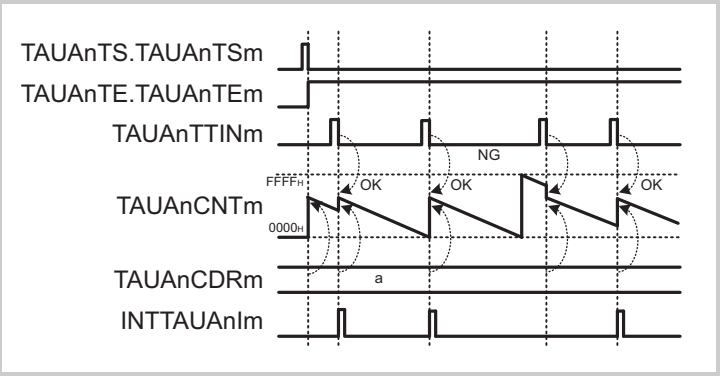


Figure 13-48 General Timing Diagram of TAUAnTTINm Input Pulse Interval Judgment Function

(3) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-44 TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0001: Judge mode
TAUAnMD0	0: INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm 1: INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-45 TAUAnCMURm Settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-46 Simultaneous Rewrite Settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Operating Procedure for TAUAnTTINm Input Pulse Interval Judgment Function

Table 13-47 Operating Procedure for TAUAnTTINm Input Pulse Interval Judgment Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-44, TAUAnCMORm Settings for TAUAnTTINm Input Pulse Interval Judgment Function, and table 13-45, TAUAnCMURm Settings for TAUAnTTINm Input Pulse Interval Judgment Function. Set the value of TAUAnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCDRm value is loaded into TAUAnCNTm.
During Operation	Detection of TAUAnTTINm edge The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When a TAUAnTTINm input edge is detected: • TAUAnCNTm reloads TAUAnCDRm value and continues count operation. • TAUAnCNTm compares the values and judges the condition according to the TAUAnCMORm.TAUAnMD0 setting. • If the condition is satisfied, INTTAUAnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

13.13.9 TAUAnTTINm Input Signal Width Judgment Function

(1) Overview

Summary

This function outputs the result of a comparison between the count value (TAUAnCNTm) and the value in the channel data register (TAUAnCDRm) when a valid stop edge of a TAUAnTTINm input signal is detected. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge and one-count mode. (See table 13-48, TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Judgment Function.)
- TAUAnTTOUTm is not used with this function.

Functional description

The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When a valid TAUAnTTINm input start edge is detected, the current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from this value.

When a TAUAnTTINm valid stop edge is detected, the function compares the current values of TAUAnCNTm and TAUAnCDRm. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true. The counter TAUAnCNTm retains its value until the next valid TAUAnTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUAnTTINm stop edge is detected, TAUAnCNTm underflows and is set to $FFFF_H$. The counter then continues to count down.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions

- The TAUAnCMORm.TAUAnMD0 bit specifies the type of comparison:
  - If TAUAnCMORm.TAUAnMD0 = 0, INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm.
  - If TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm.
- The TAUAnCMURm.TAUAnTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B), TAUAnTTINm rising edge is used as a start edge and TAUAnTTINm falling edge as a stop edge.
  - For low width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 10_B), TAUAnTTINm falling edge is used as a start edge and TAUAnTTINm rising edge as a stop edge.
- This function cannot make a forced restart.

(2) Block Diagram and General Timing Diagram

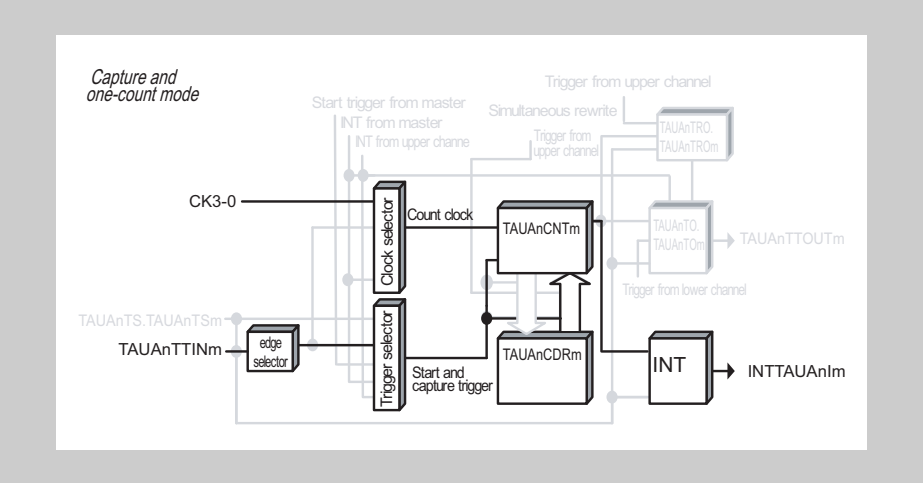


Figure 13-49 Block Diagram of TAUAnTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm (TAUAnCMORm.TAUAnMD0 = 0)
- TAUAnTTINm valid start edge = rising edge, TAUAnTTINm valid stop edge = falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

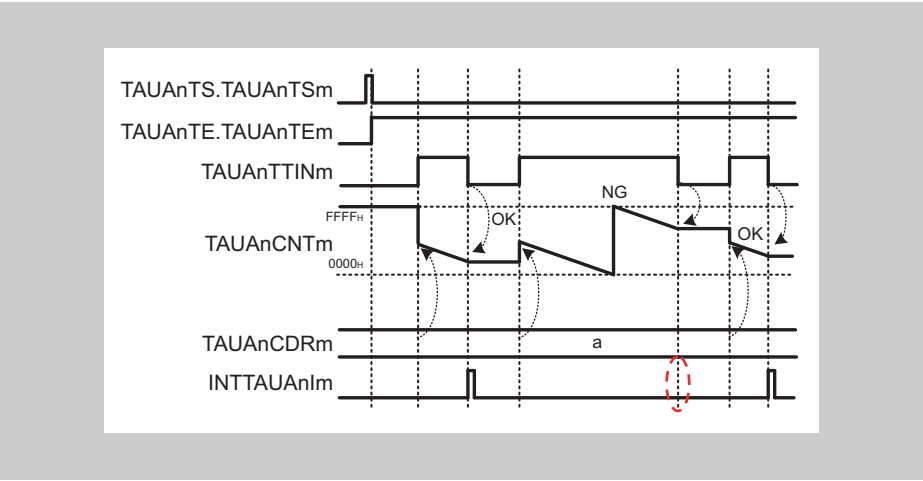


Figure 13-50 General Timing Diagram of TAUAnTTINm Input Signal Width Judgment Function

(3) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS[1:0]		TAUAnCCS[1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS[1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-48 TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is used as an external start trigger and the reverse edge as a stop triggler.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0111: Judge and one-count mode
TAUAnMD0	0: INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm 1: INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-49 TAUAnCMURm Settings for TAUAnTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUAnTIS[1:0]	10: Detection of falling and rising edges (Low width measurement) 11: Detection of falling and rising edges (High width measurement)

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-50 Simultaneous Rewrite Settings for TAUAnTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Operating Procedure for TAUAnTTINm Input Signal Width Judgment Function

Table 13-51 Operating Procedure for TAUAnTTINm Input Signal Width Judgment Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-48, TAUAnCMORm Settings for TAUAnTTINm Input Signal Width Judgment Function, and table 13-49, TAUAnCMURm Settings for TAUAnTTINm Input Signal Width Judgment Function. Set the value of TAUAnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of TAUAnTTINm start edge. When a TAUAnTTINm start edge is detected, TAUAnCDRm value is loaded into TAUAnCNTm.
During Operation	Detection of TAUAnTTINm edge The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When TAUAnTTINm stop edge is detected: • TAUAnCNTm stops and waits for detection of TAUAnTTINm start edge. • TAUAnCNTm compares the values and judges the condition according to the TAUAnCMORm.TAUAnMD0 setting. • If the condition is satisfied, INTTAUAnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

13.13.10 Real-Time Output Function Type 1

(1) Overview

Summary

This function outputs a value of the TAUAnTRO.TAUAnTROm bit from TAUAnTTOUTm when a specified channel generates an interrupt (INTTAUAnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel (TAUAnTRC.TAUAnTRCm = 1) which generates a real-time output trigger, and the lower channel is a channel (TAUAnTRC.TAUAnTRCm = 0) which makes a real-time output in response to the upper channel trigger.

Prerequisites

- Channels should use the TAUAnTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See table 13-52, TAUAnCMORm Settings for Real-Time Output Function Type 1.*)*
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See section 13.8, Channel Output Modes.)
- Real-time output should be enabled for the upper channel (TAUAnTRE.TAUAnTREm = 1).

Functional description

The counter of the upper channel is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The current value of the data register of the upper channel (TAUAnCDRm) is loaded into the counter (TAUAnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUAnIm is generated and TAUAnTTOUTm outputs the current value of the real-time output bit (TAUAnTRO.TAUAnTROm) of every channel (only channels with TAUAnTRE.TAUAnTREm = 1). TAUAnCNTm then reloads the TAUAnCDRm value to continue operation subsequently.

The TAUAnTTOUTm signal changes only when an interrupt is generated, and when its value is different to current value of TAUAnTRO.TAUAnTROm at the moment that the interrupt is generated.

Conditions

- The channel which is monitored for INTTAUAnIm occurrence is specified by setting TAUAnTRC.TAUAnTRCm to 1 for the corresponding channel. The TAUAnTRC.TAUAnTRCm bit should be 0 for all other channels.
- If real-time output of a lower channel is disabled (TAUAnTRE.TAUAnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUAnTRC.TAUAnTRCm = 1), the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUAnTRE.TAUAnTREm = 1) and TAUAnTRC.TAUAnTRCm = 0, the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated in the upper channel.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

(2) Equations

INTTAUAnIm generation cycle = count clock cycle × (TAUAnCDRm value + 1)

(3) Block Diagram and General Timing Diagram

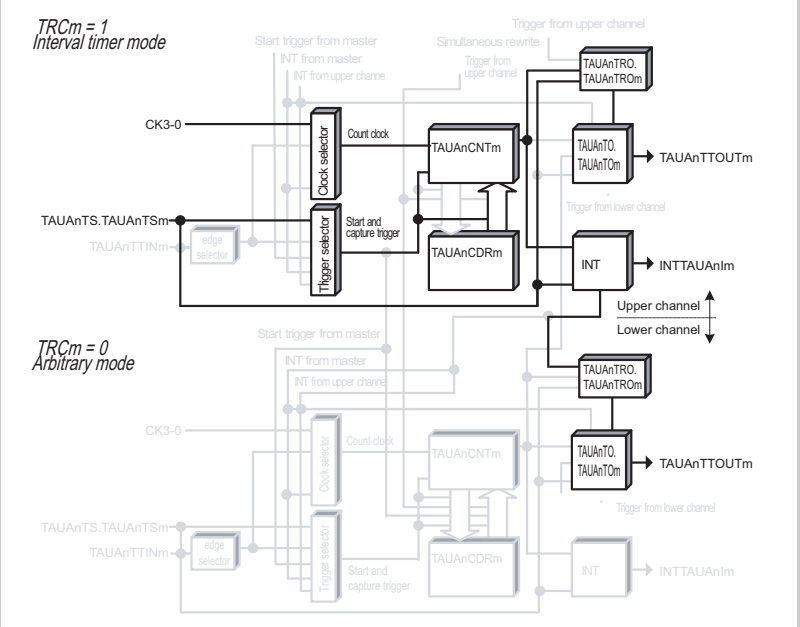


Figure 13-51 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)

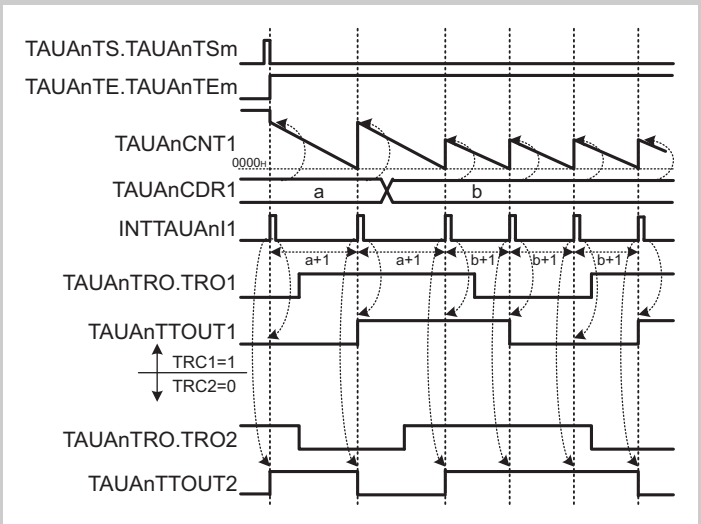


Figure 13-52 General Timing Diagram of Real-Time Output Function Type 1

(4) Register Settings for Upper Channels

(a) TAUAnCMORm for Upper Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-52 TAUAnCMORm Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	000: Trigger the counter using software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation. 1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Upper Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-53 TAUAnCMURm Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Upper Channels

Table 13-54 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables independent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	1: Channel m generates a unique real-time trigger.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

Table 13-55 Simultaneous Rewrite Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Register Settings for Lower Channels

(a) TAUAnCMORm for Lower Channels

The TAUAnCMORm register for lower channels is available for any setting.

(b) TAUAnCMURm for Lower Channels

The TAUAnCMURm register for lower channels is available for any setting.

(c) Channel Output Mode for Lower Channels

Table 13-56 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables independent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set this bit to 0.
TAUAnTDL.TAUAnTDLm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set this bit to 0.
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

(6) Operating Procedure for Real-Time Output Function Type 1

Table 13-57 Operating Procedure for Real-Time Output Function Type 1

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers for upper channels as described in table 13-52, TAUAnCMORm Settings for Real-Time Output Function Type 1, and table 13-53, TAUAnCMURm Settings for Real-Time Output Function Type 1. Set TAUAnCMORm and TAUAnCMURm registers for lower channels as described in (5) Register Settings for Lower Channels. Set the value of TAUAnCDRm register (only channels with TAUAnTRC.TAUAnTRCm = 1) Set channel output mode for the upper channel by setting the control bits as described in table 13-54, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output. Set channel output mode for the lower channel by setting the control bits as described in table 13-56, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm = 1 on the channel with TAUAnTRC.TAUAnTRCm set to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUAnTRC.TAUAnTRCm set to 1] TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCDRm value is loaded into TAUAnCNTm. If TAUAnCMORm.TAUAnMD0 is 1, INTTAUAnIm occurs.
During Operation	TAUAnCDRm and TAUAnTRO.TAUAnTROm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When the counter reaches 0000_H: • The value of TAUAnCDRm is reloaded into TAUAnCNTm to continue count operation. • INTTAUAnIm is generated. • TAUAnTTOUTm outputs the current value of the real-time output bit TAUAnTRO.TAUAnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. Both TAUAnCNTm and TAUAnTTOUTm retain their current values.

Restart

(7) Specific Timing Diagrams

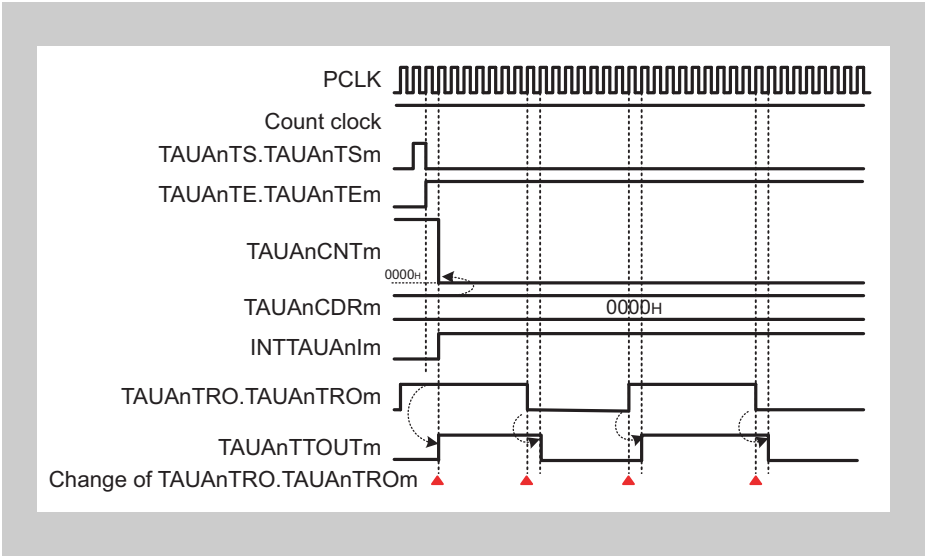


Figure 13-53 TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1

- When PCLK cycle is delayed, the value of TAUAnTTOUTm changes according to the setting of TAUAnTRO.TAUAnTROm.

13.13.11 Real-Time Output Function Type 2

(1) Overview

Summary

This function outputs the value of TAUAnTRO.TAUAnTROm bit from TAUAnTTOUTm when a specified channel generates an interrupt (INTTAUAnIm). In this function, the interrupt is generated when a valid TAUAnTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUAnTRC.TAUAnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUAnTRC.TAUAnTRCm = 0).

Prerequisites

- Channels should use the TAUAnTTOUTm control of the other channels
- The operating mode for the upper channel should be set to interval timer mode. (See table 13-58, TAUAnCMORm Settings for Real-Time Output Function Type 2.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See section 13.8, Channel Output Modes.)
- Real-time output should be enabled for the upper channel (TAUAnTRE.TREm = 1).

Functional description

The counter for upper channels is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm to 1, enabling count operation. The counter starts to count up.

When a valid TAUAnTTNm input edge is generated on one of upper channels, an interrupt occurs and TAUAnTTOUTm outputs the current value of the real-time output bit (TAUAnTRO.TAUAnTROm) of every channel (only channels with TAUAnTRE.TAUAnTREm = 1).

The TAUAnTTOUTm signal changes only when an interrupt is generated, and when TAUAnTTOUTm value is different to the current value of TAUAnTRO.TAUAnTROm during the occurrence of the interrupt.

Conditions

- The channel which is monitored for INTTAUAnIm occurrence is specified by setting TAUAnTRC.TRCm to 1 for the corresponding channel. The TAUAnTRC.TRCm bit should be 0 for all other channels.
- If real-time output of a lower channel is disabled (TAUAnTRE.TAUAnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUAnTRC.TAUAnTRCm = 1), the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUAnTRE.TAUAnTREm = 1) and TAUAnTRC.TAUAnTRCm = 0, the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated in the upper channel.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

(2) Block Diagram and General Timing Diagram

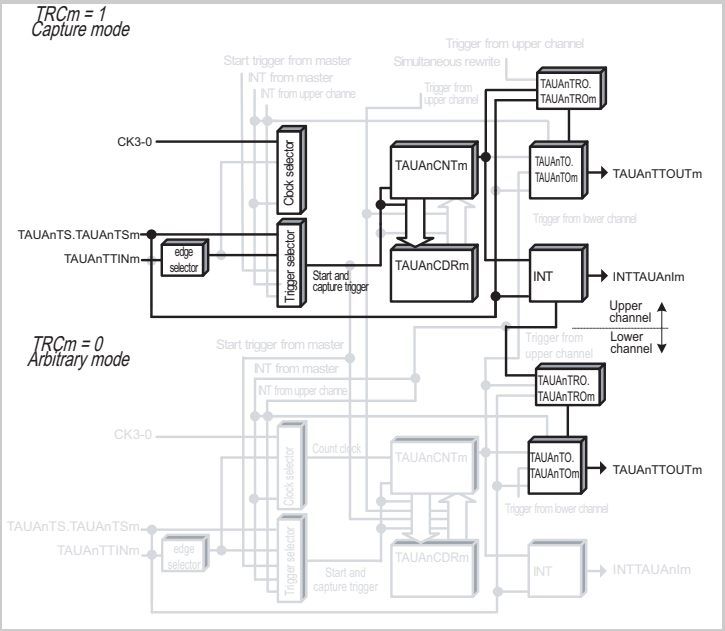


Figure 13-54 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUAnIm is not generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 0)

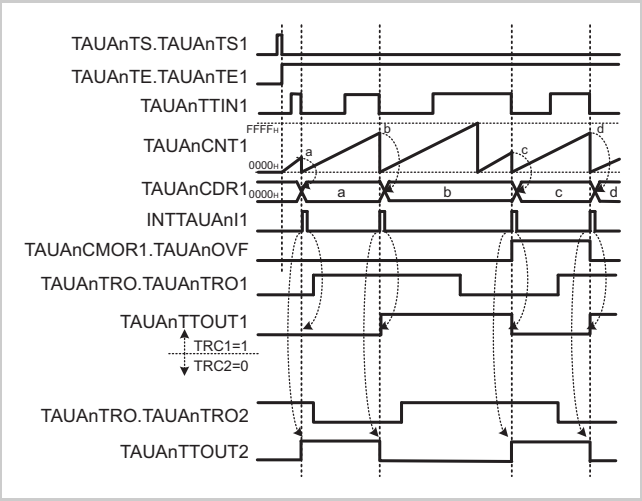


Figure 13-55 General Timing Diagram of Real-Time Output Function Type 2

(3) Register Settings for Upper Channels

(a) TAUAnCMORm for Upper Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-58 TAUAnCMORm Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0010: Capture mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation. 1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Upper Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-59 TAUAnCMURm Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode for Upper Channels

Table 13-60 Control Bit Settings in Independent Channel Output Mode 2 with Real-Time Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	1: Channel m generates a unique real-time trigger.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

Table 13-61 Simultaneous Rewrite Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Register Settings for Lower Channels

(a) TAUAnCMORm for Lower Channels

TAUAnCMORm register for lower channels can be set arbitrarily.

(b) TAUAnCMURm for Lower Channels

TAUAnCMURm register for lower channels can be set arbitrarily.

(c) Channel Output Mode for Lower Channels

Table 13-62 Control Bit Settings for Lower Channels in Independent Channel Output Mode 2 with Real-Time Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output. 1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: Upper channel generates a real-time trigger for channel m. 1: Channel m generates its own real-time trigger.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

(5) Operating Procedure for Real-Time Output Function Type 2

Table 13-63 Operating Procedure for Real-Time Output Function Type 2

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers for upper channels as described in table 13-58, TAUAnCMORm Settings for Real-Time Output Function Type 2, and table 13-59, TAUAnCMURm Settings for Real-Time Output Function Type 2. Set TAUAnCMORm and TAUAnCMURm registers for the lower channel as described in (5) Register Settings for Lower Channels. Set channel output mode for the upper channel by setting the control bits as described in table 13-60, Control Bit Settings in Independent Channel Output Mode 2 with Real-Time Output. Set channel output mode for the lower channel by setting the control bits as described in table 13-62, Control Bit Settings for Lower Channels in Independent Channel Output Mode 2 with Real-Time Output.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm = 1 on the channel with TAUAnTRC.TAUAnTRCm set to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUAnTRC.TAUAnTRCm set to 1] TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm is cleared to 0000_H. If TAUAnCMORm.TAUAnMD0 is 1, INTTAUAnIm occurs.
During Operation	TAUAnTRO.TAUAnTROm can be changed at any time.	TAUAnCNTm starts to count up from 0000_H. When a valid TAUAnTTINm input edge is detected: • INTTAUAnIm is generated. TAUAnTTOUTm outputs the current value of real-time output bit TAUAnTRO.TAUAnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. TAUAnCNTm, TAUAnCSRm.TAUAnOVF, and TAUAnTTOUTm retain their current values.

Restart

13.13.12 Simultaneous Rewrite Trigger Generation Function Type 1

(1) Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUAnRDC.TAUAnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUAnRDC.TAUAnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUAnRDE.RDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See table 13-64, TAUAnCMORm Settings for Simultaneous Rewrite Trigger Generation Function Type 1.)
- For the operating mode that can be set for lower channels, see table 13-8, Simultaneous Rewrite and Trigger Timing.
- TAUAnTTOUTm is not used for any channel in this function.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUAnTS.TAUAnTSm) for upper and lower channels to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUAnCDRm buf) is loaded into the counter (TAUAnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000_H, an interrupt occurs on the channel. The current value of the corresponding TAUAnCDRm buffer is loaded into TAUAnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUAnRDC.RDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUAnRSF.RSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUAnIm occurrence is specified by setting TAUAnRDC.TAUAnRDCm = 1 for the corresponding channel. The TAUAnRDC.TAUAnRDCm bit should be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

(2) Equations

Simultaneous rewrite trigger generation cycle = count clock cycle x (TAUAnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUAnCDRm = [(value of TAUAnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] - 1

[Triangle PWM]

TAUAnCDRm = [(value of TAUAnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] - 1

That is, the ratio of TAUAnCDRm + 1 and TAUAnCDRm_master + 1 should be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

(3) Block Diagram and General Timing Diagram

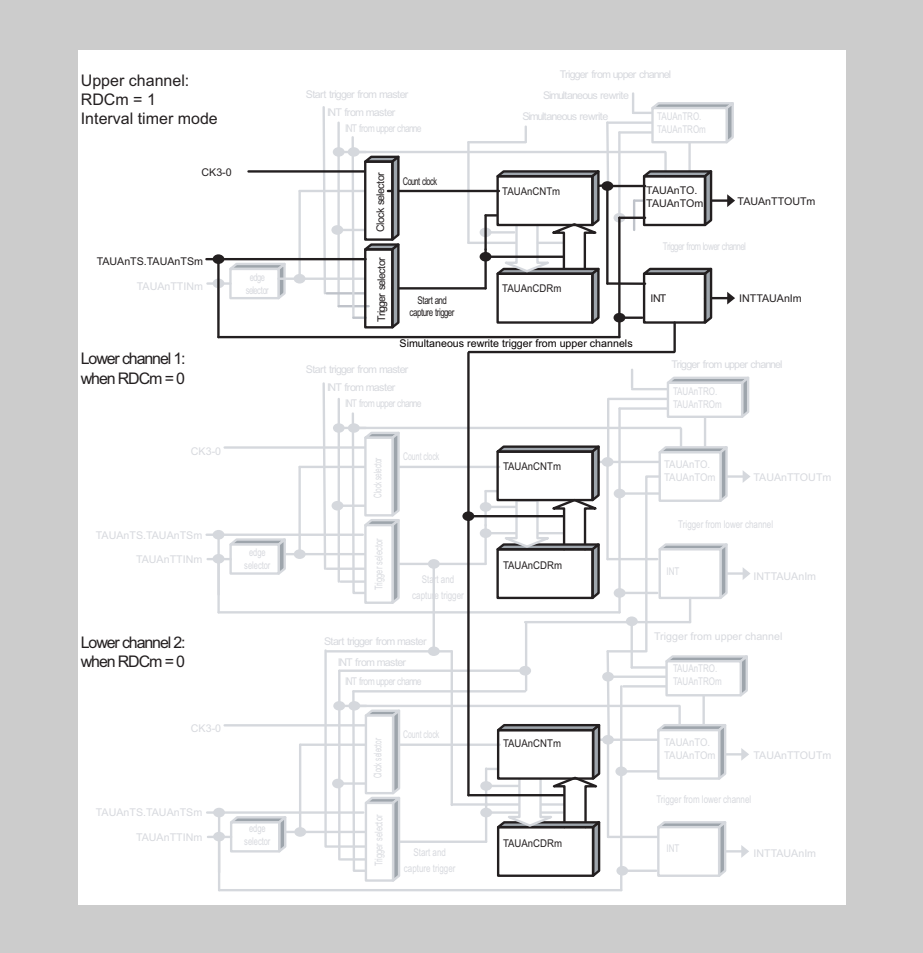


Figure 13-56 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)

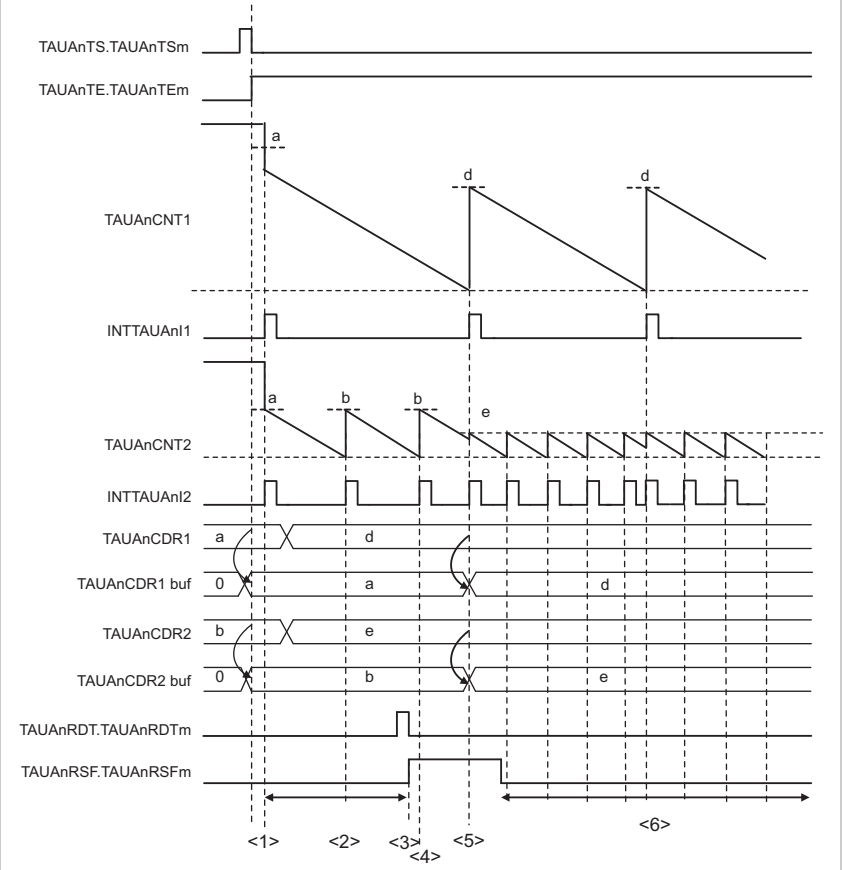


Figure 13-57 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to TAUAnCDRm buffer.
2. TAUAnCDRm register is always ready to write.
3. Status flag is set (TAUAnRSF.TAUAnRSFm = 1) by setting the reload data trigger bit (TAUAnRDT.TAUAnRDTm) to 1, enabling simultaneous rewrite.
4. Simultaneous rewrite is not performed even if enabled, because it is triggered only by a CH1 interrupt.
5. Simultaneous rewrite is triggered by INTTAUAnI1 which is generated when the TAUAnCNT1 reaches 0000_H. TAUAnCDRm value is loaded into the corresponding TAUAnCDRm buffer.
6. The counter starts counting down and waits for the next simultaneous rewrite trigger. The value of TAUAnCDRm register can be changed again.

(4) Register Settings for Upper Channels

(a) TAUAnCMORm for Upper Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-64 TAUAnCMORm Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation. 1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Upper Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-65 TAUAnCMURm Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Upper Channels

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Upper Channels

Table 13-66 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUAnRDC.TAUAnRDCm	1: Monitors INTTAUAnIm signal which triggers a simultaneous rewrite on the channel.

(5) Register Settings for Lower Channels

(a) TAUAnCMORm for Lower Channels

TAUAnCMORm register for lower channels must follow the TAUAnCMORm register settings in the operating mode which can be set. See table 13-8, Simultaneous Rewrite and Trigger Timing.

(b) TAUAnCMURm for Lower Channels

TAUAnCMURm register for lower channels must follow the TAUAnCMURm register settings in the operating mode which can be set. See table 13-8, Simultaneous Rewrite and Trigger Timing.

(c) Channel Output Mode for Lower Channels

Output can be made according to the operating mode setting (master/slave) for lower channels.

(d) Simultaneous Rewrite for Lower Channels

Table 13-67 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: Not monitor INTTAUAnIm signal which triggers a simultaneous rewrite on the channel.

(6) Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 13-68 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers for the upper channel as described in table 13-64, TAUAnCMORm Settings for Simultaneous Rewrite Trigger Generation Function Type 1, and table 13-65, TAUAnCMURm Settings for Simultaneous Rewrite Trigger Generation Function Type 1. Set TAUAnCMORm and TAUAnCMURm registers for lower channels as described in (5) Register Settings for Lower Channels. Set the value of TAUAnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCDRm value is loaded into TAUAnCNTm. If TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm occurs.
During Operation	TAUAnRDT.TAUAnRDTm and TAUAnCDR.CDRm is changeable. TAUAnRSF.TAUAnRSFm can be always read.	TAUAnCNTm counts down. When the counter reaches 0000_H: • The TAUAnCDRm value is reloaded into TAUAnCNTm to continue count operation. • INTTAUAnIm occurs. If INTAUAnIm occurs on the channel where TAUAnRDC.TAUAnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

13.13.13 One-Phase PWM Output Function

(1) Overview

Summary

This function adds dead time to a TAUAnTTINm input signal. The resulting PWM signal is output via TAUAnTTOUTm of the channel and TAUAnTTOUTm of upper channels.

Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUAnTDE.TDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See table 13-69, TAUAnCMORm Settings for One-Phase PWM Output Function.)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWN output. (See section 13.8, Channel Output Modes.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. The value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value.

When the counter reaches 0000_H, an interrupt occurs. The counter is reset to $FFFF_H$ and waits for the next valid TAUAnTTINm input start edge.

Conditions

- TAUAnCMURm.TIS[1:0] bits specify the type of width measurement:
  - TAUAnCMURm.TAUAnTIS[1:0] = 10_B: Uses both edges as valid edges for detection (Low width measurement).
  - TAUAnCMURm.TAUAnTIS[1:0] = 11_B: Uses both edges as valid edges for detection (High width measurement).
- The TAUAnTDL.TAUAnTDLm bit specifies the operation of TAUAnTTOUTm for each channel when an interrupt or valid TAUAnTTINm edge is detected on the lower channel:
  - If TAUAnTDL.TAUAnTDLm = 0, an interrupt is used as a TAUAnTTOUTm set trigger and a valid TAUAnTTINm edge as a TAUAnTTOUTm reset trigger.
  - If TAUAnTDL.TAUAnTDLm = 1, a valid TAUAnTTINm edge is used as a TAUAnTTOUTm set trigger and an interrupt as a TAUAnTTOUTm reset trigger.
- This function cannot make a forced restart.

(2) Block Diagram and General Timing Diagram

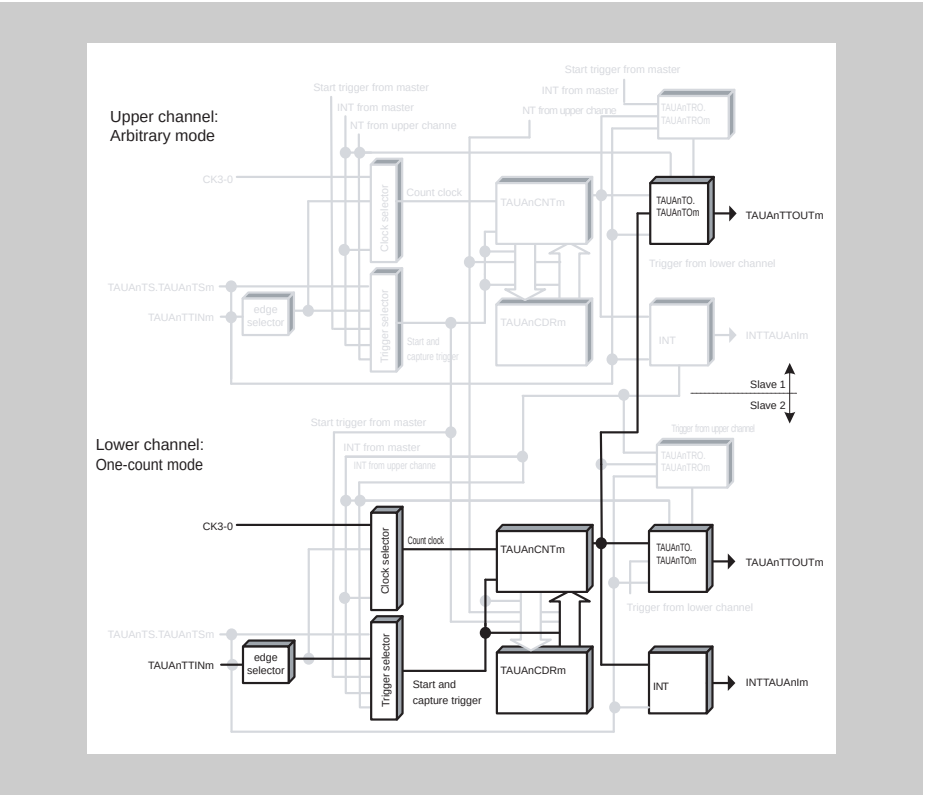


Figure 13-58 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of falling and rising edges = High width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

This setting considers a duty as a high active.

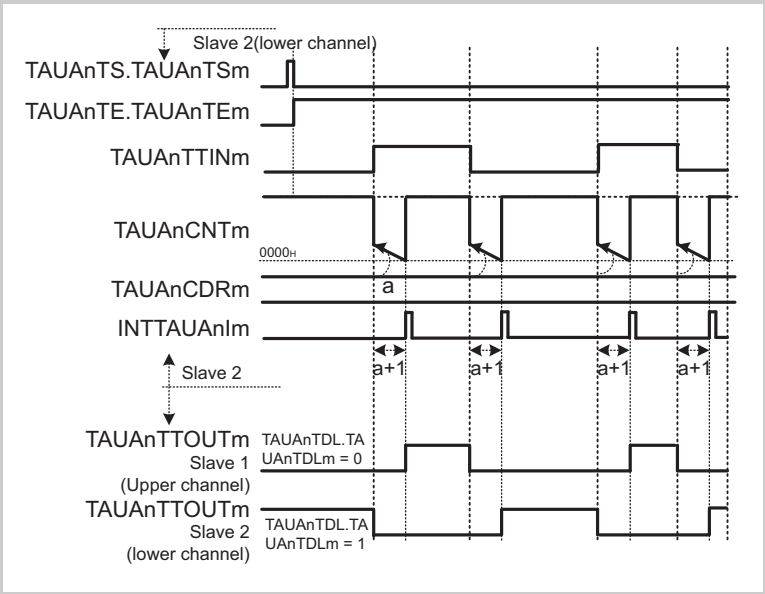


Figure 13-59 General Timing Diagram of One-Phase PWM Output Function

(3) Register Settings for Lower Channels

(a) TAUAnCMORm for Lower Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-69 TAUAnCMORm Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is used as an external start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Enables start trigger detection while counting.

(b) TAUAnCMURm for Lower Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-70 TAUAnCMURm Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUAnTIS[1:0]	10: Detection of falling and rising edges (Low width measurement) 11: Detection of falling and rising edges (High width measurement)

(c) Channel Output Mode for Lower Channels

Table 13-71 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation.
TAUAnTDM.TAUAnTDMm	1: Adds dead time upon detection of a TAUAnTTINm input edge on a lower odd channel.
TAUAnTDL.TAUAnTDLm	0: Uses an interrupt as a TAUAnTTOUTm set trigger and a valid TAUAnTTINm edge as a TAUAnTTOUTm reset trigger. 1: Uses a valid TAUAnTTINm edge as a TAUAnTTOUTm set trigger and an interrupt as a TAUAnTTOUTm reset trigger.
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Caution Set TAUAnTDL.TAUAnTDLm to upper channels exclusively.

(d) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-72 Simultaneous Rewrite Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Register Settings for Upper Channels

(a) TAUAnCMORm for Upper Channels

TAUAnCMORm register for upper channels can be set arbitrarily.

(b) TAUAnCMURm for Upper Channels

TAUAnCMURm register for upper channels can be set arbitrarily.

(c) Channel Output Mode for Upper Channels

Table 13-73 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation.
TAUAnTDM.TAUAnTDMm	1: Adds dead time upon detection of a TAUAnTTINm input edge on a lower odd channel.
TAUAnTDL.TAUAnTDLm	0: Uses a lower channel interrupt as a TAUAnTTOUTm set trigger and a valid lower channel TAUAnTTINm edge as a TAUAnTTOUTm reset trigger. 1: Uses a valid lower channel TAUAnTTINm edge as a TAUAnTTOUTm set trigger and a lower channel interrupt as a TAUAnTTOUTm reset trigger.
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Caution Set TAUAnTDL.TAUAnTDLm to lower channels exclusively.

(d) Simultaneous Rewrite for Upper Channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

(5) Operating Procedure for One-Phase PWM Output Function

Table 13-74 Operating Procedure for One-phase PWM Output Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers for the lower channel as described in table 13-69, TAUAnCMORm Settings for One-Phase PWM Output Function, and table 13-70, TAUAnCMURm Settings for One-Phase PWM Output Function. Set TAUAnCMORm and TAUAnCMURm registers for the upper channel as described in (4) Register Settings for Upper Channels. Set the value of TAUAnCDRm register. Set channel output mode for the upper and lower channels by setting the control bits as described in table 13-71, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output.	Channel operation is stopped.
Start Operation	Set TAUAnTOE.TAUAnTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUAnTS.TAUAnTSm = 1 for slave channel 2. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 (slave channel 2) and TAUAnCNTm waits for detection of TAUAnTTINm start edge. TAUAnCNTm loads TAUAnCDRm value.
During Operation	TAUAnCDRm register value is changeable at any time. The TAUAnCNTm register are readable at any time.	TAUAnCNTm of slave channel 2 counts down. When the counter reaches 0000_H: • INTTAUAnIm occurs. • TAUAnCNTm stops counting. TAUAnTTOUTm is changed by a TAUAnTTINm edge detection signal and slave channel 2 INTTAUAnIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm = 1 for slave channel 2. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. TAUAnCNTm and TAUAnTTOUTm retain their current values.

Restart

13.13.14 External Event Count Function

(1) Overview

Summary This function is used as an event timer, which generates an interrupt (INTTAUAnIm) when a specific number of TAUAnTTINm input pulses has occurred.

Prerequisites
- The operating mode should be set to the event count mode. (See table 13-75, TAUAnCMORm Settings for External Event Count Function.)
- TAUAnTTOUTm is not used with this function.

Functional description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When the counter starts, the current value of TAUAnCDRm is loaded into TAUAnCNTm.

When a valid TAUAnTTINm input edge is detected, the value of TAUAnCNTm decrements by 1. TAUAnCNTm retains this value until a valid TAUAnTTINm input edge is detected or the counter is restarted.

When the counter reaches 0000_H, INTTAUAnIm occurs. Then, TAUAnCDRm value is loaded into TAUAnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm stops and retains its value. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUAnTS.TAUAnTSm to 1 during operation.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions An edge type used as a trigger is specified by TAUAnCMURm.TAUAnTIS[1:0] bits.
- When TAUAnCMURm.TAUAnTIS[1:0] = 00_B, falling edges are counted.
- When TAUAnCMURm.TAUAnTIS[1:0] = 01_B, rising edges are counted.
- When TAUAnCMURm.TAUAnTIS[1:0] = 10_B, both edges are counted.

(2) Equations

Number of valid edges detected before INTTAUAnIm generation = TAUAnCDRm + 1

(3) Block Diagram and General Timing Diagram

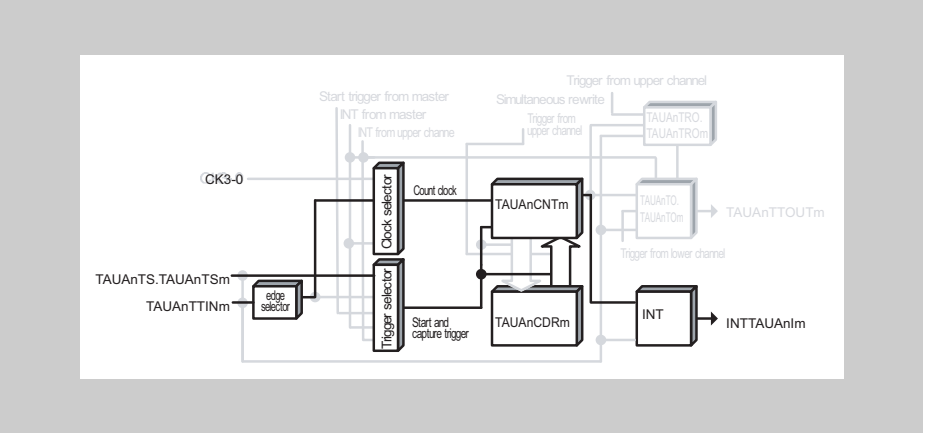


Figure 13-60 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

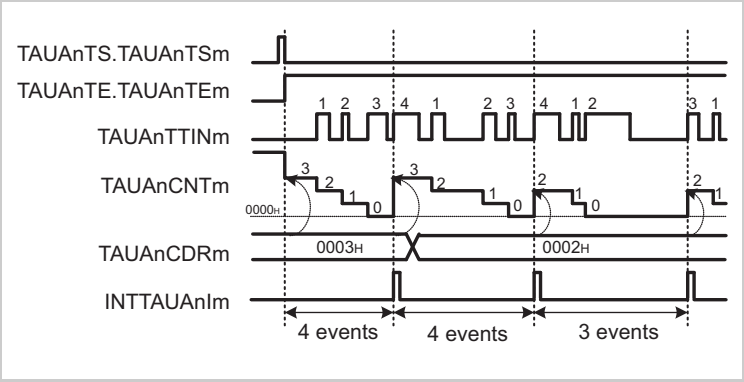


Figure 13-61 General Timing Diagram of External Event Count Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-75 TAUAnCMORm Settings for External Event Count Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	01: Valid TAUAnTTINm input edge is used as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	000: Triggers the counter using software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0011: Event count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-76 TAUAnCMURm Settings for External Event Count Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Falling edge 01: Rising edge 10: Falling and rising edges

(c) Channel Output Mode

The channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-77 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for External Event Count Function

Table 13-78 Operating Procedure for External Event Count Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-75, TAUAnCMORm Settings for External Event Count Function, and table 13-76, TAUAnCMURm Settings for External Event Count Function. Set the value of TAUAnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads TAUAnCDRm value and waits for TAUAnTTINm input edge detection.
During Operation	Detection of TAUAnTTINm edge The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down each time TAUAnTTINm input edge is detected. When the counter reaches 0000_H: • TAUAnCDRm value is loaded in TAUAnCNTm and count operation continues. • INTTAUAnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

(6) Specific Timing Diagrams

(a) TAUAnCDRm = 0000_H

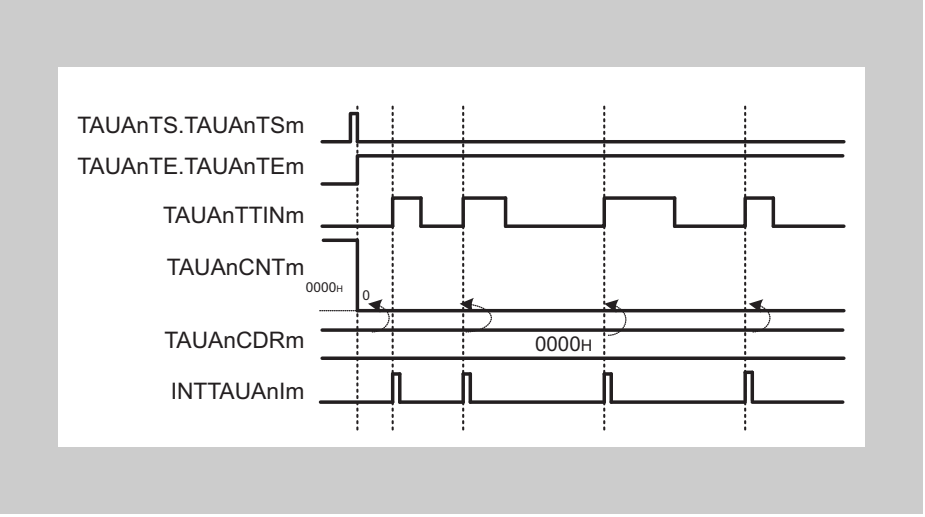


Figure 13-62 TAUAnCDRm = 0000_H, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

- If 0000_H = TAUAnCDRm, 0000_H is loaded into TAUAnCNTm each time a valid TAUAnTTINm input edge is detected.

  In other words, INTTAUAnIm occurs each time a valid TAUAnTTINm input edge is detected.

(b) Operation Stop and Restart

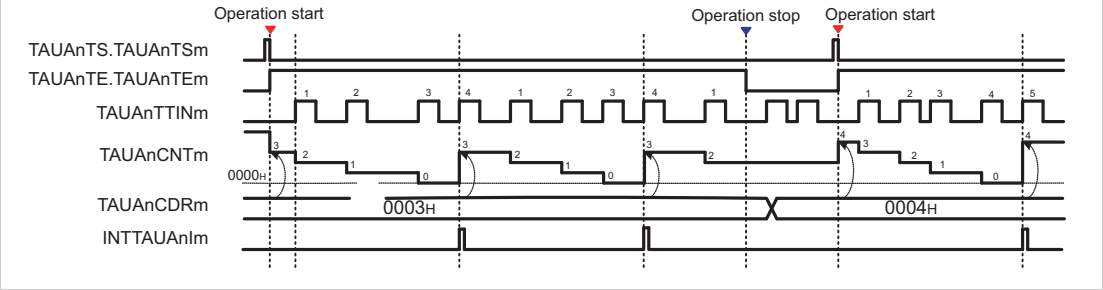


Figure 13-63 Operation Stop and Restart (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and retains its current value. TAUAnTTINm continues and TAUAnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm loads the TAUAnCDRm value and restarts count operation.

(c) Forced Restart

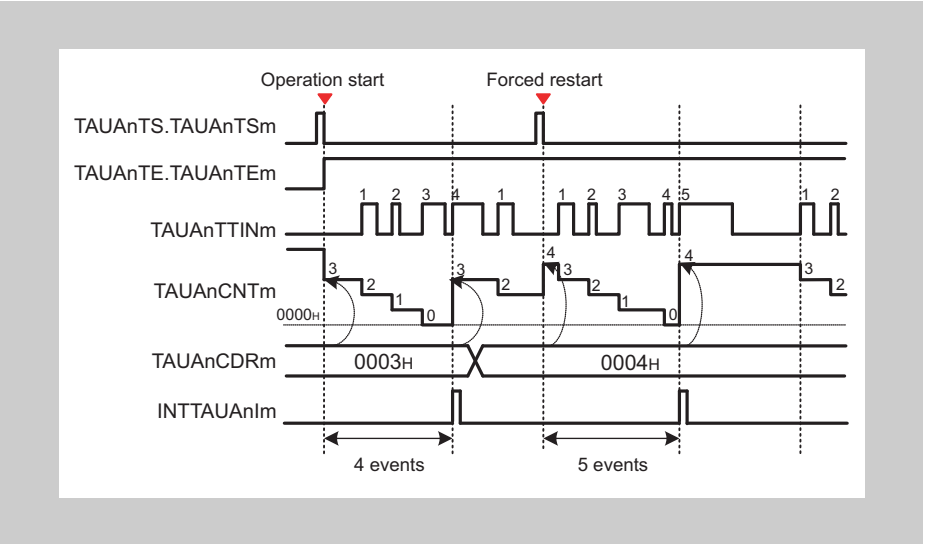


Figure 13-64 Forced Restart (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

Once a forced restart is made, the changed point is applied to TAUAnCDRm immediately.

- The counter can be restarted without making a stop by setting TAUAnTS.TAUAnTSm to 1 during operation.
- The value of TAUAnCDRm is loaded into TAUAnCNTm and the counter awaits the next valid TAUAnTTINm input edge.

13.13.15 Clock Divide Function

(1) Overview

Summary This function is used as a frequency divider. The frequency of the input signal TAUAnTTINm is divided by a factor related to TAUAnCDRm, and the resulting signal is output to TAUAnTTOUTm.

Prerequisites
- TAUAnTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See table 13-79, TAUAnCMORm Settings for Clock Divide Function.)
- The channel output mode should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes.)

Functional description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from this value, using TAUAnTTINm as a count clock.

When the counter value reaches 0000_H, INTTAUAnIm occurs and TAUAnTTOUTm signal is toggled. Then, TAUAnCDRm value is loaded into TAUAnCNTm to continue operation subsequently.

The value of TAUAnCDRm can be rewritten at any time. The changed value of TAUAnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUAnTT.TAUAnTTm = 1. This sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm and TAUAnTTOUTm stop but retain their values. The function can be restarted by setting TAUAnTS.TAUAnTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUAnTS.TAUAnTSm = 1 during operation (forced restart).

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUAnTTOUTm does not toggle. This results in an inverted TAUAnTTOUTm signal compared to when TAUAnCMORm.TAUAnMD0 is set to 1. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

Note TAUAnTTINm input signals are sampled at the frequency of a sampling clock set by TAUAnCMORm.TAUAnCKS[1:0] bits. Therefore, the TAUAnTTOUTm output clock cycle has an error of ± 1 sampling clock cycle.

(2) Equations

- When rising edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency/ [(TAUAnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency/ [(TAUAnCDRm + 1) × 2]
- When falling and rising edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency/ (TAUAnCDRm + 1)

(3) Block Diagram and General Timing Diagram

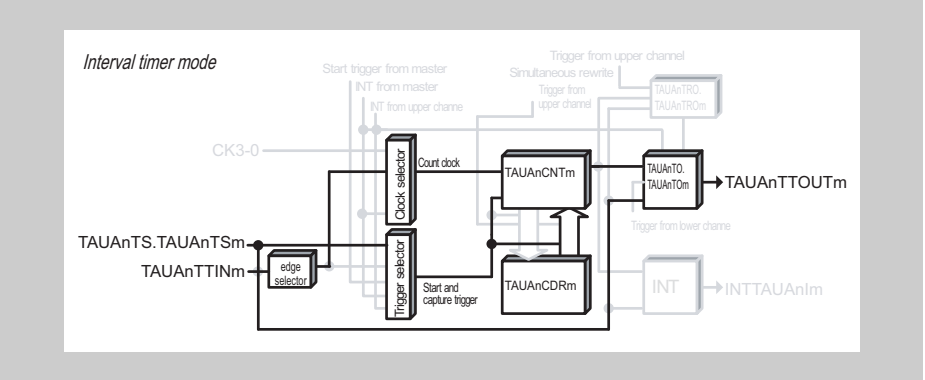


Figure 13-65 Block Diagram of Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)
- Detection of rising edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

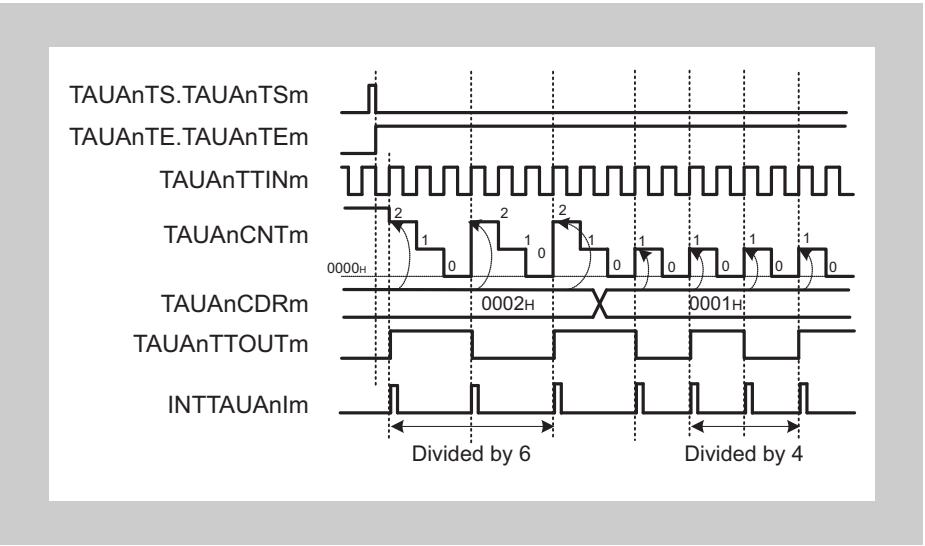


Figure 13-66 General Timing Diagram of Clock Divide Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS[1:0]		TAUAnCCS[1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS[1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-79 TAUAnCMORm Settings for Clock Divide Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	01: Valid TAUAnTTINm input edge is used as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated and TAUAnTTOUTm is not toggled at the beginning of operation. 1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-80 TAUAnCMURm Settings for Clock Divide Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

Table 13-81 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When dissabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-82 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for Clock Divide Function

Table 13-83 Operating Procedure for Clock Divide Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-79, TAUAnCMORm Settings for Clock Divide Function, and table 13-80, TAUAnCMURm Settings for Clock Divide Function. Set the value of TAUAnCDRm register. Set the channel output mode by setting the control bits as described in table 13-81, Control Bit Settings in Independent Channel Output Mode 1.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads TAUAnCDRm value. If TAUAnCMORm.TAUAnMD0 is set to 1, INTTAUAnIm occurs and TAUAnTTOUTm is toggled.
During Operation	The value of TAUAnCDRm can be changed at any time. TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down each time TAUAnTTINm input edge is detected. When the counter reaches 0000_H: • TAUAnCDRm value is loaded in TAUAnCNTm and count operation continues. • INTTAUAnIm is generated. • TAUAnTTOUTm is toggled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. TAUAnCNTm and TAUAnTTOUTm retain their current values.

Restart

(6) Specific Timing Diagrams

(a) TAUAnCDRm = 0000_H

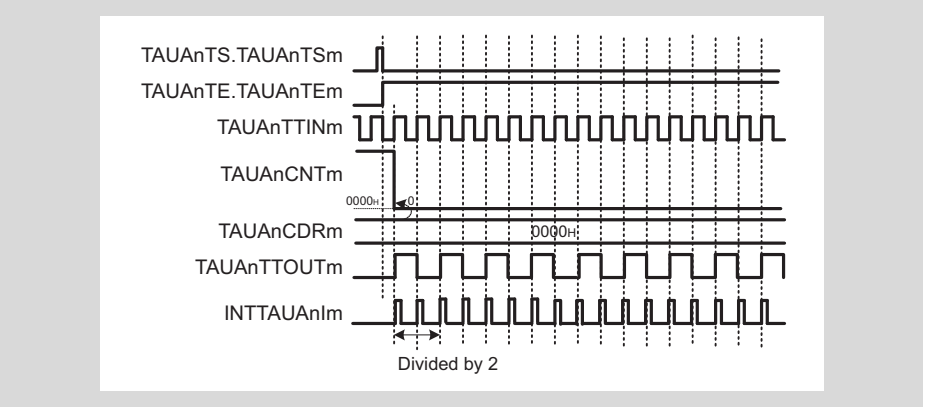


Figure 13-67 TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

- If TAUAnCDRm is 0000_H, TAUAnCNTm is always 0000_H.
- INTTAUAnIm is generated every count clock, resulting in TAUAnTTOUTm toggling every count clock

figure13-67 TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B, shows an operation timing example. Actually, there is a delay from TAUAnTTINm detection until TAUAnTTOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUAnIm pin and TAUAn.

(b) Restart

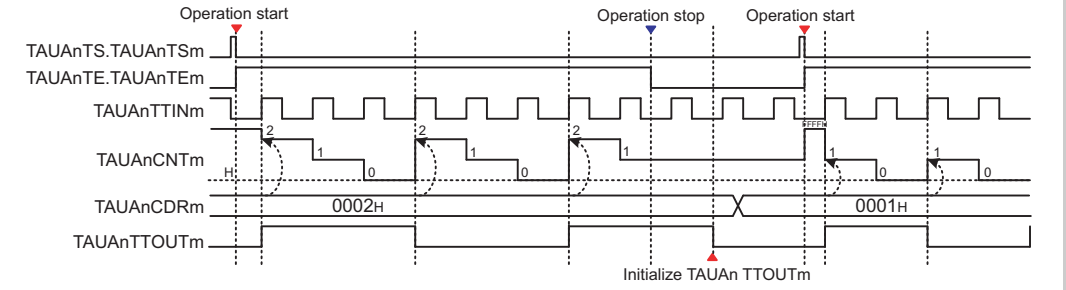


Figure 13-68 Restart (TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

To reset the value of TAUAnTTOUTm:

- Set TAUAnTOE.TAUAnTOEm = 0 when the counter is stopped (TAUAnTE.TAUAnTEm = 0).
- Then, write either 0 or 1 TAUAnTO.TAUAnTOm to set the new start value of TAUAnTTOUTm.

13.13.16 TAUAnTTINm Input Position Detection Function

(1) Overview

Summary This function measures the duration between a count start timing and a TAUAnTTINm input signal.

Prerequisites
- The operating mode should be set to count capture mode. (See table 13-84, TAUAnCMORm Settings for TAUAnTTINm Input Position Detection Function.*)*
- TAUAnTTOUTm is not used with this function.

Functional description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter starts counting from 0000_H. When a valid TAUAnTTINm input edge is detected, the current value of TAUAnCNTm is loaded into TAUAnCDRm and an interrupt (INTTAUAnlm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the TAUAnCSRm.TAUAnOVF bit is set to 1 and the counter restarts to count from 0000_H. TAUAnCSRm.TAUAnOVF is cleared by writing 1 to TAUAnCSCm.TAUAnCLOV.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see section 13.10, TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts.

(2) Equations

Functional duration at a TAUAnTTINm input pulse =

count clock cycle × [(TAUAnCSRm.TAUAnOVF × ($FFFF_H$ + 1)) + (TAUAnCDRm capture value + 1)]

(3) Block Diagram and General Timing Diagram

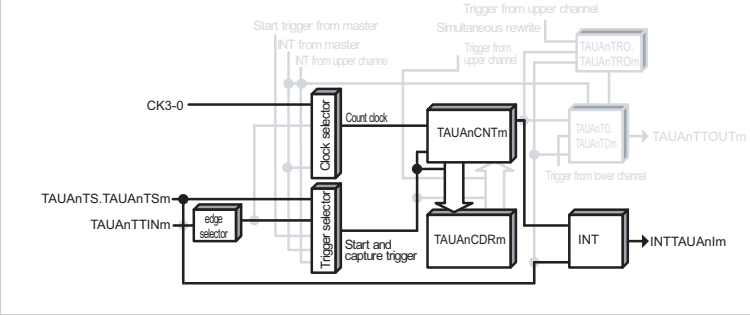


Figure 13-69 Block Diagram of TAUAnTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUAnIm is not generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 0)
- Detection of falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

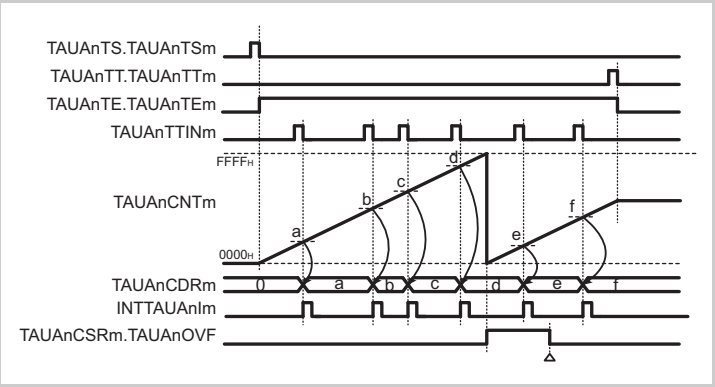


Figure 13-70 General Timing Diagram of TAUAnTTINm Input Position Detection Function

(4) Register Settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-84 TAUAnCMORm Settings for TAUAnTTINm Input Position Detection Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Unused. Set to 0.
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as an external capture trigger.
TAUAnCOS[1:0]	01: Overflow (TAUAnCSRm.TAUAnOVF) is set when the counter overflows. This is cleared by setting TAUAnCSCm.TAUAnCLOV = 1.
TAUAnMD[4:1]	1011: Count capture mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation. 1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-85 TAUAnCMURm Settings for TAUAnTTINm Input Position Detection Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges

(c) Channel Output Mode

The channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-86 Simultaneous Rewrite Settings for TAUAnTTINm Input Position Detection Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating Procedure for TAUAnTTINm Input Position Detection Function

Table 13-87 Operating Procedure for TAUAnTTINm Input Position Detection Function

	Operation	TAUAn Status
Initial Channel Setting	Set TAUAnCMORm and TAUAnCMURm registers as described in table 13-84, TAUAnCMORm Settings for TAUAnTTINm Input Position Detection Function, and table 13-85, TAUAnCMURm Settings for TAUAnTTINm Input Position Detection Function. TAUAnCDRm register operates as a capture register.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. If TAUAnCMORm.TAUAnMD0 is 1, INTTAUAnIm occurs.
During Operation	The TAUAnCMURm.TIS[1:0] bits can be changed at any time. The TAUAnCDRm and TAUAnCSRm registers can be read at any time. TAUAnCSC.CLOV bit can be set to 1.	TAUAnCNTm starts to count up from 0000_H. When a valid TAUAnTTINm edge is detected: • TAUAnCNTm transfers (captures) its own value to TAUAnCDRm. • INTTAUAnIm occurs. • The counter is not cleared to 0000_H and TAUAnCNTm continues counting. Afterwards, this procedure is repeated.
Stop Operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops. TAUAnCNTm and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific Timing Diagrams

(a) Operation Stop and Restart

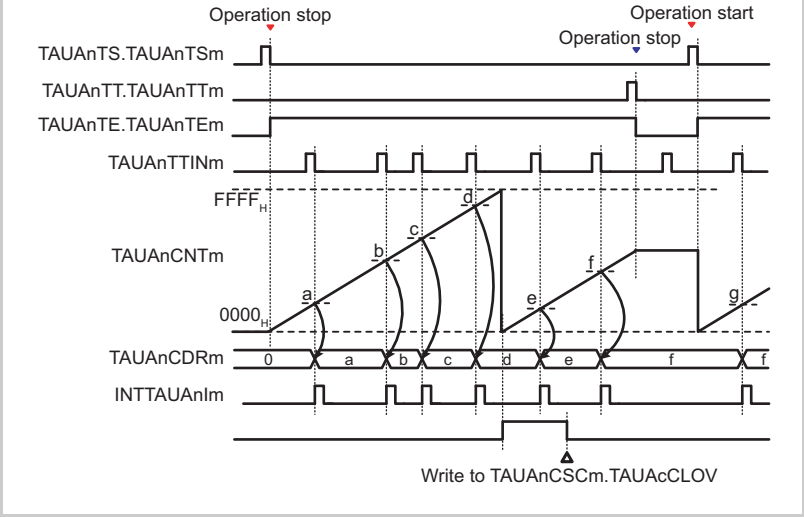


Figure 13-71 Operation Stop and Restart (TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

- The counter can stop operating by setting TAUAnTT.TAUAnTTm to 1. This sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUAnTTINm input edges are ignored.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm restarts to count from 0000_H.

13.14 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit A. For a general overview of synchronous channel operation, see section 13.3, Functional Description.

13.14.1 PWM Output Function

(1) Overview

Summary This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUAnTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See table 13-88, TAUAnCMORm Settings for Master Channels of PWM Output Function.)
- The operating mode for the slave channels should be set to one count mode. (See table 13-91, TAUAnCMORm Settings for Slave Channels of PWM Output Function.)
- TAUAnTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1. (See section 13.8, Channel Output Modes.)

Functional description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded into TAUAmCNT, and the counter starts counting down from the TAUAnCDRm value. If an INTTAUAnlm occurs on the master channel and TAUAnTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:

  When the master channel counter reaches 0000_H and the pulse cycle time has passed, INTTAUAnIm occurs. The counter loads TAUAnCDRm value into TAUAnCNTm and counts down.

- Slave channels:

  When INTTAUAnIm occurs on the master channel, the counter operation of the slave channel is triggered. The current value of TAUAnCDRm (slave) is loaded into TAUAnCNTm (slave) and the counter starts counting down from the TAUAnCDRm value. TAUAnTTOUTm signal is set to the active level. When the counter reaches to 0000_H (duty time has elapsed), INTTAUAnIm occurs and a TAUAnTTOUTm signal is set to an inactive level. The counter is reset to $FFFF_H$ and waits for the next INTTAUAnlm (start of the next pulse cycle) of the master channel.

The counter can stop operating by setting the TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. See section 13.7, Simultaneous Rewrite.

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUAnCDRm (slave)/(TAUAnCDRm (master) + 1) × 100

– Duty cycle = 0%

  TAUAnCDRm (slave) = 0000_H

– Duty cycle = 100%

  TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

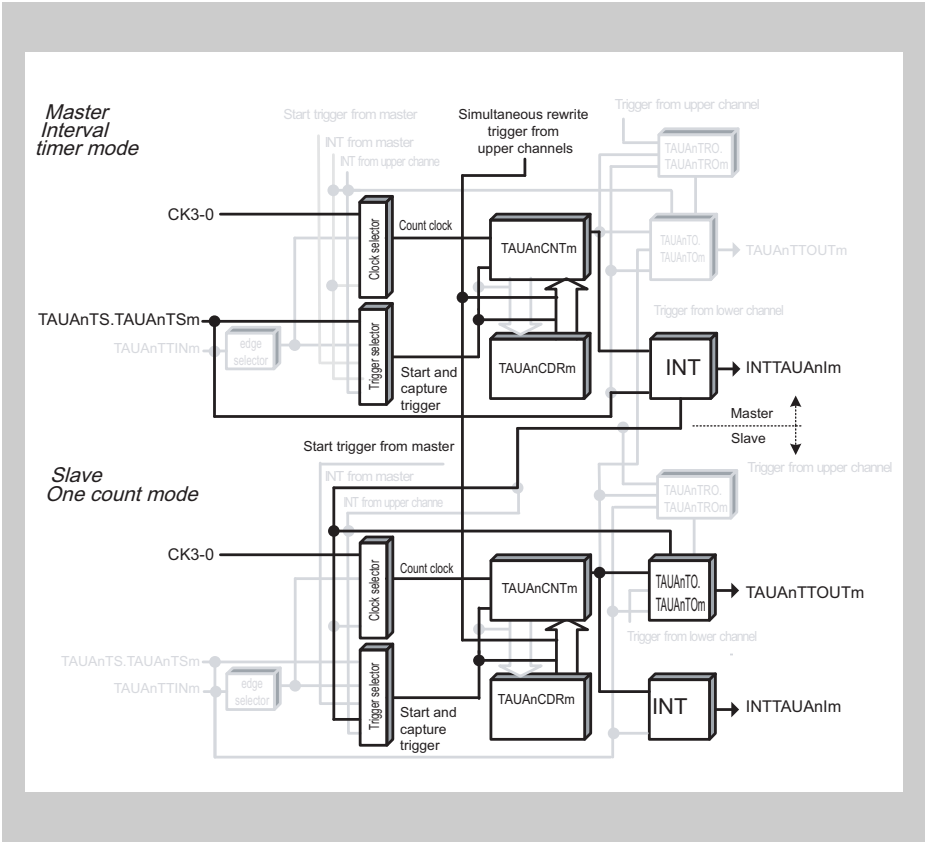


Figure 13-72 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channels: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

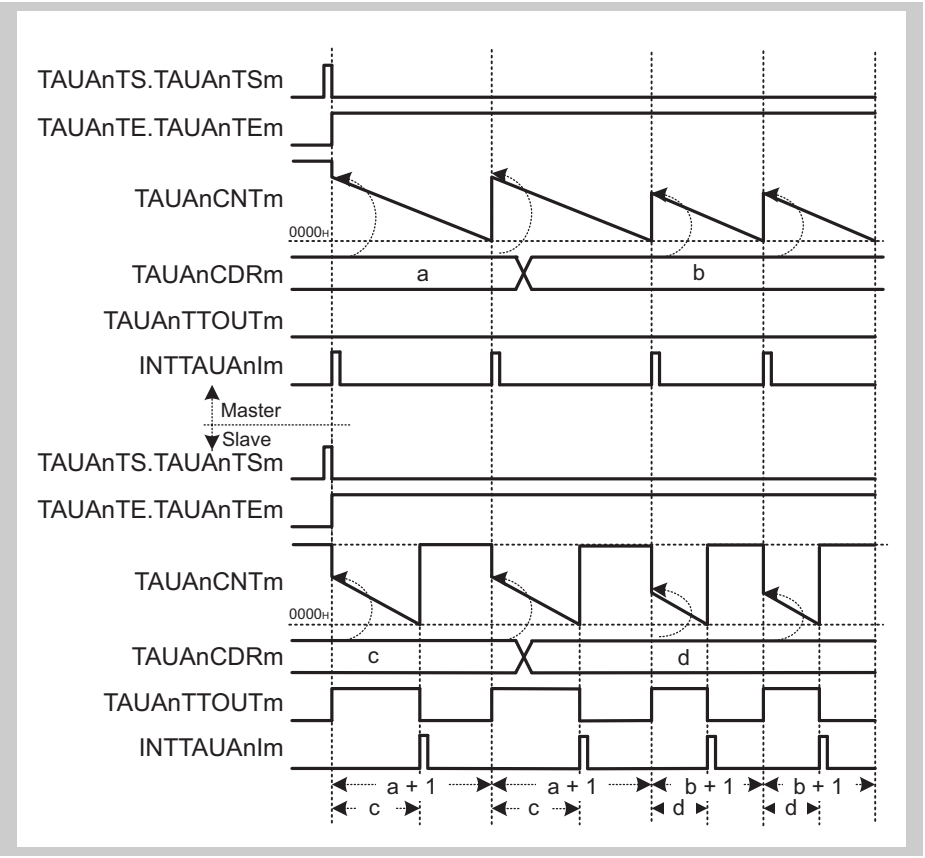


Figure 13-73 General Timing Diagram of PWM Output Function

Note The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUAnCDRm, whereas for the master channel the interval is the value of the corresponding TAUAnCDRm + 1.

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS[1:0]		TAUAnCCS[1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS[1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-88 TAUAnCMORm Settings for Master Channels of PWM Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-89 TAUAnCMURm Settings for Master Channels of PWM Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

The channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-90 Simultaneous Rewrite Settings for Master Channels of the PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note Use with TAUAnRDS.TAUAnRDSm bit = 1 requires an upper channel higher than the master channel that operates with the "simultaneous rewrite trigger output function type 1."

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel:
  TAUAnRDCm = 1, TAUAnRDS = 1
  TAUAnCDR settings for this channel are as follows:
  = ((TAUAnCDR setting for the master channel targeted for simultaneous rewrite + 1) × interrupt count) - 1
- Master channels: TAUAnRDCm = 0, TAUAnRDS = 1
- Slave channels: TAUAnRDCm = 0, TAUAnRDS = 1

If TAUAnCDRm (slave) setting > TAUAnCDRm (master) setting + 1, 100% duty output should be provided through aggregation.

(5) Register Settings for Slave Channels

(a) TAUAnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-91 TAUAnCMORm Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Start trigger during operation is valid.

(b) TAUAnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-92 TAUAnCMURm Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channels

Table 13-93 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	0: Operating mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-94 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Operating Procedure for PWM Output Function

Table 13-95 Operating Procedure for PWM Output Function

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (4) Register Settings for Master Channels. Slave channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channels. Set the value of TAUAnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (slave) is set.
During Operation	TAUAnCDRm can be changed at any time.TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of master channel loads TAUAnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUAnIm (master) occurs. • TAUAnCDRm value is loaded into TAUAnCNTm (master) to continue count operation. • TAUAnCDRm value is loaded into TAUAnCNTm (slave) to perform counting down. • TAUAnTTOUTm (slave) is set to the active level. If TAUAnCNTm (slave) reaches 0000_H: • INTTAUAnIm (slave) occurs. • TAUAnTTOUTm (slave) is set to an inactive level.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific Timing Diagrams

(a) Duty Cycle = 0%

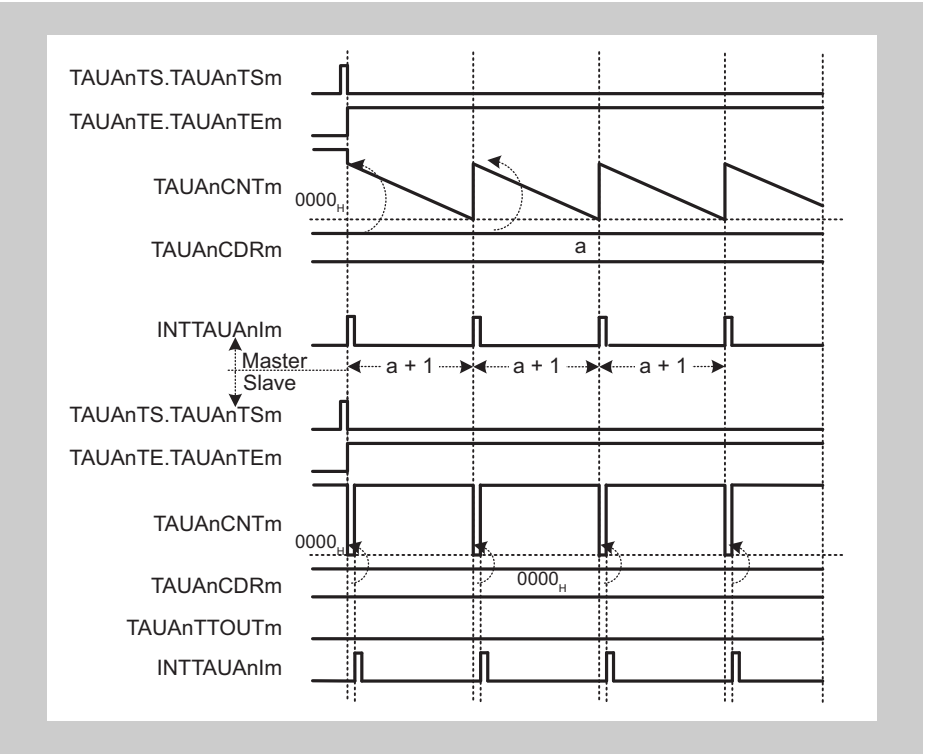


Figure 13-74 TAUAnCDRm (Slave) = 0000_H, Positive Logic (TAUAnTOL.TAUAnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUAnIm), 0000_H is loaded in to TAUAnCNTm (slave). Therefore, TAUAnCNTm (slave) cannot start to count and TAUAnTTOUTm remains inactive.
- TAUAnCDRm value is loaded into TAUAnCNTm (slave) to generate an interrupt.

(b) Duty Cycle = 100%

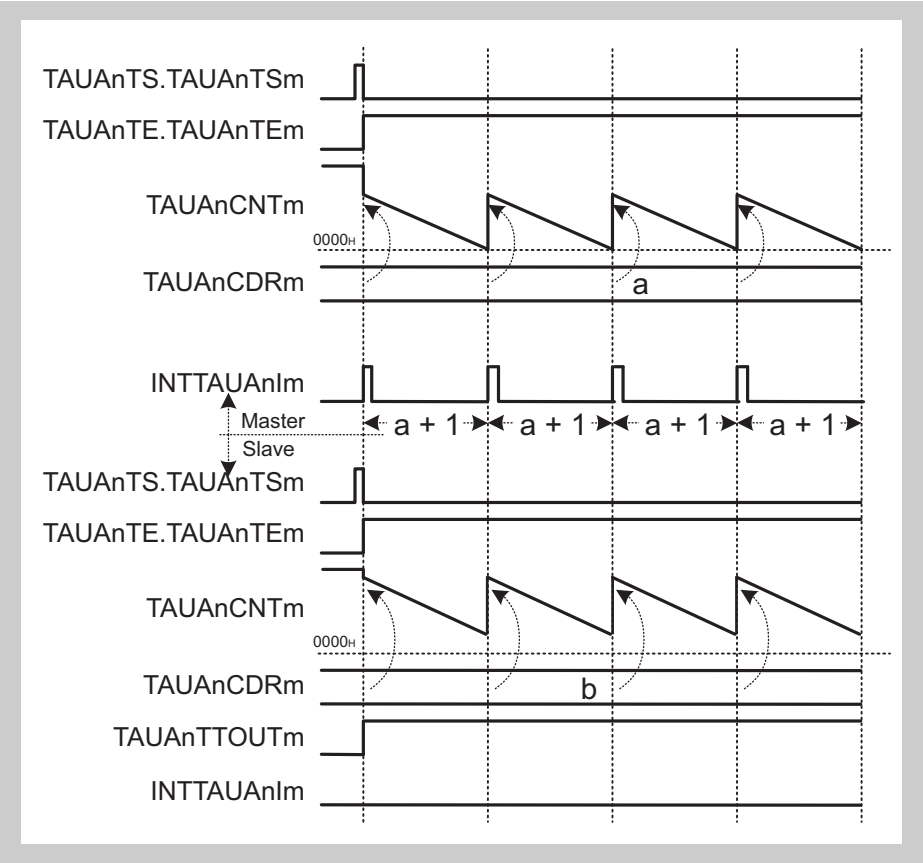


Figure 13-75 TAUAnCDRm (Slave) ≥ TAUAnCDRm (Master) + 1 Positive Logic (TAUAnTOL.TAUAnTOLm (Slave) = 0)

- If TAUAnCDRm (slave) value is greater than TAUAnCDRm (master) value, the slave channel counter does not reach 0000_H and consequently, no interrupt occurs. TAUAnTTOUTm remains active.

(c) Operation Stop and Restart

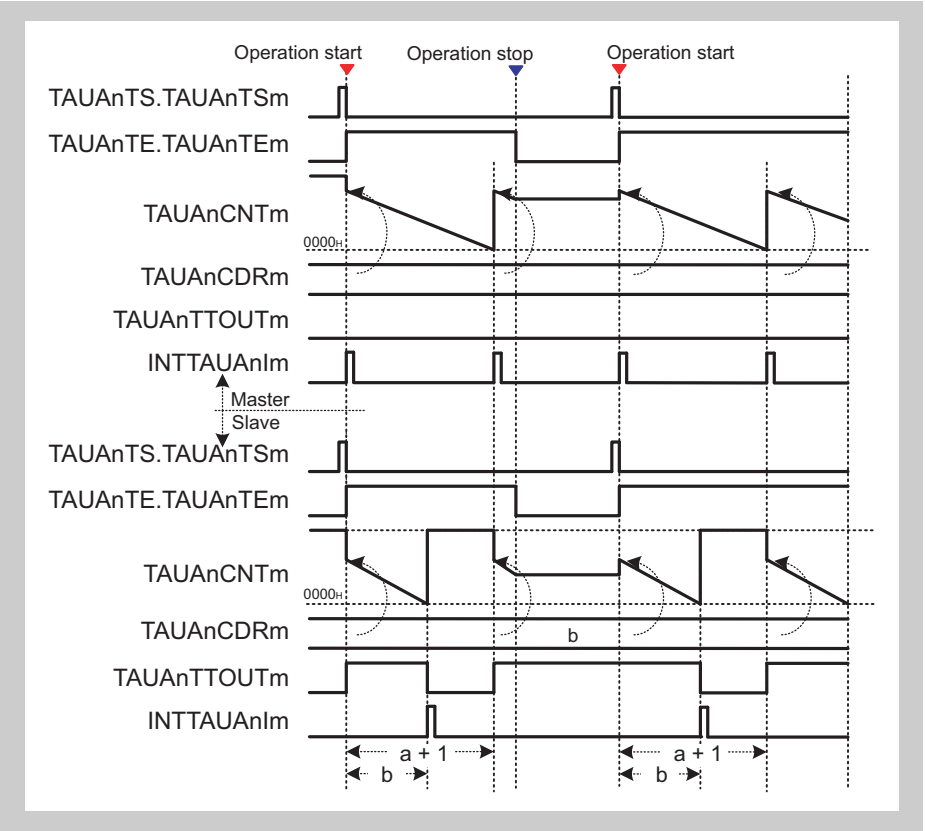


Figure 13-76 Operation Stop and Restart
Positive Logic (TAUAnTOL.TAUAnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm and TAUAnTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm of master and slave channels to 1. TAUAnCNTm of master and slave channels reload the current values of TAUAnCDRm and start to count down from these values.

13.14.2 Delay Pulse Output Function

(1) Overview

Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified on slave channel 2.

Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See table 13-96, TAUAnCMORm Settings for Master Channels of the Delay Pulse Output Function.)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See table 13-99, TAUAnCMORm Settings for Slave Channel 1 of Delay Pulse Output Function.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See table 13-103, TAUAnCMORm Settings for Slave Channel 2 of Delay Pulse Output Function.)
- TAUAnTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See section 13.8, Channel Output Modes.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See section 13.8, Channel Output Modes.)

Functional description

The counters of the channel group are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm to 1, enabling count operation.

- Master channel:

  The current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from this value. INTTAUAnIm is generated on the master channel.

  When the counter value of master channel reaches 0000_H and pulse cycle time has elapsed, INTTAUAnIm is generated. The TAUAnCDRm value is reloaded into the counter to perform count down.

- Slave channels 1 and 2:

  Slave channels 1 and 2 start to count down from the current TAUAnCDRm value when detecting an interrupt from the master channel. TAUAnTTOUTm signal (slave 1) is set.

  – Slave channel 1:

    When the counter of slave channel 1 reaches 0000_H (duty time has elapsed), INTTAUAnIm is generated and TAUAnTTOUTm signal is reset. The counter is reset to $FFFF_H$ and waits form the next INTTAUAnIm of master channel.

– Slave channel 2:

When the counter of slave channel 2 reaches 0000_H and delay time has elapsed, INTTAUAnIm is generated. The counter is reset to $FFFF_H$ and waits for the next INTTAUAnlm of master channel.

Generating INTTAUAnIm (slave channel 2) triggers the counter of slave channel 3.

- Slave channel 3:

When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUAnCDRm. INTTAUAnIm is generated and the TAUAnTTOUTm signal (slave 3) is set.
When the counter of slave channel 3 reaches 0001_H, INTTAUAnIn is generated and the TAUAnTTOUTm signal is reset.
The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. See section 13.7, Simultaneous Rewrite.

Equations Pulse cycle = (TAUAnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUAnCDRm (slave 1)) × count clock cycle

Delay width = (TAUAnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUAnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:
$0000_H \leq$ TAUAnCDRm (slave 2) < TAUAnCDRm (master)

Note 1. The waveform of TAUAnTTOUTm (slave 3) becomes the waveform made by delaying the waveform of TAUAnTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.

Note 2. If INTTAUAnIm of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the waveform of TAUAnTTOUTm (slave 3) is retained on the active level. In this case, TAUAnTTOUTm (slave 3) cannot output the waveform generated by delaying the basic pulse of TAUAnTTOUTm (slave 1).

(2) Block Diagram and General Timing Diagram

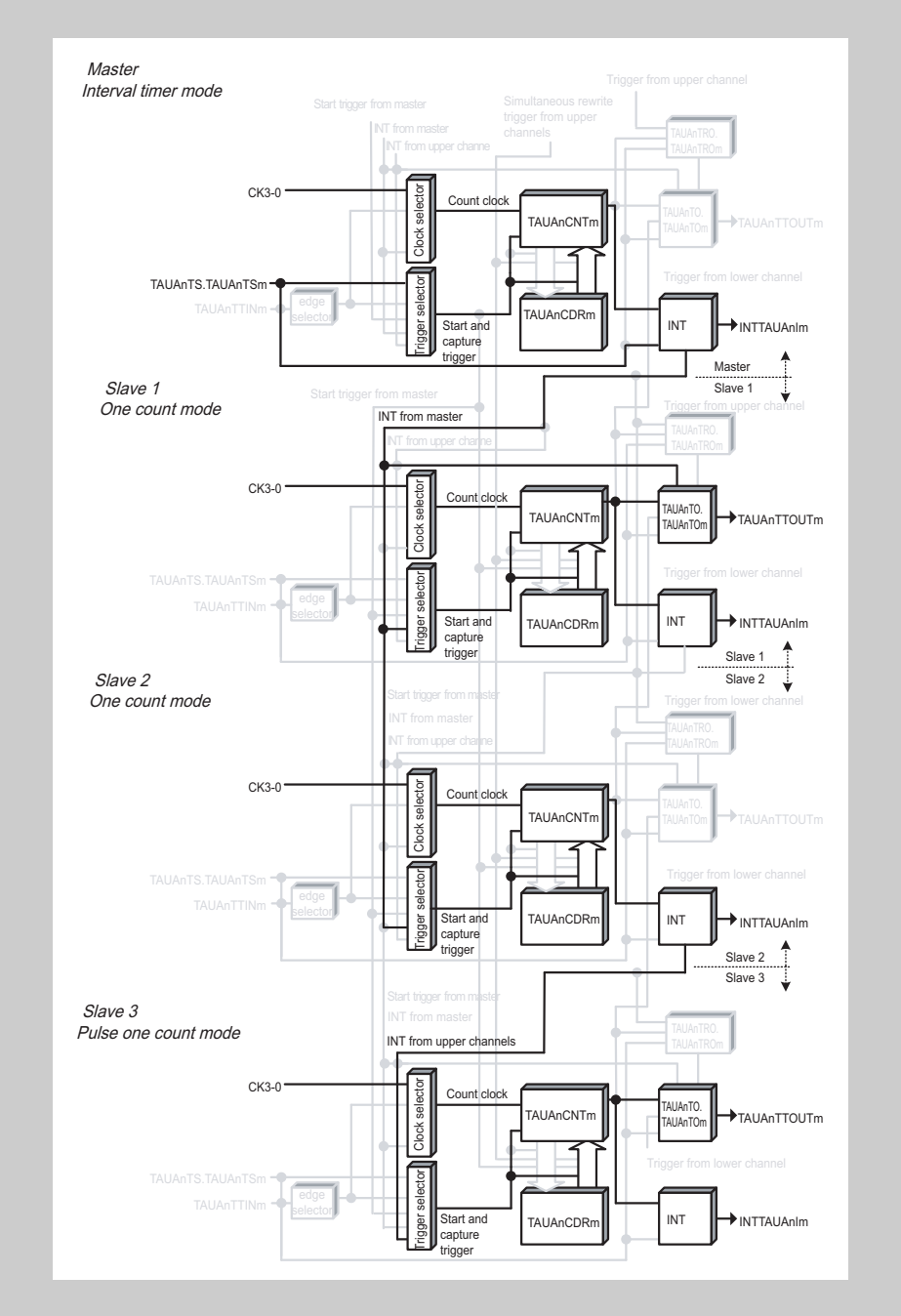


Figure 13-77 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- All channels
  - INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)

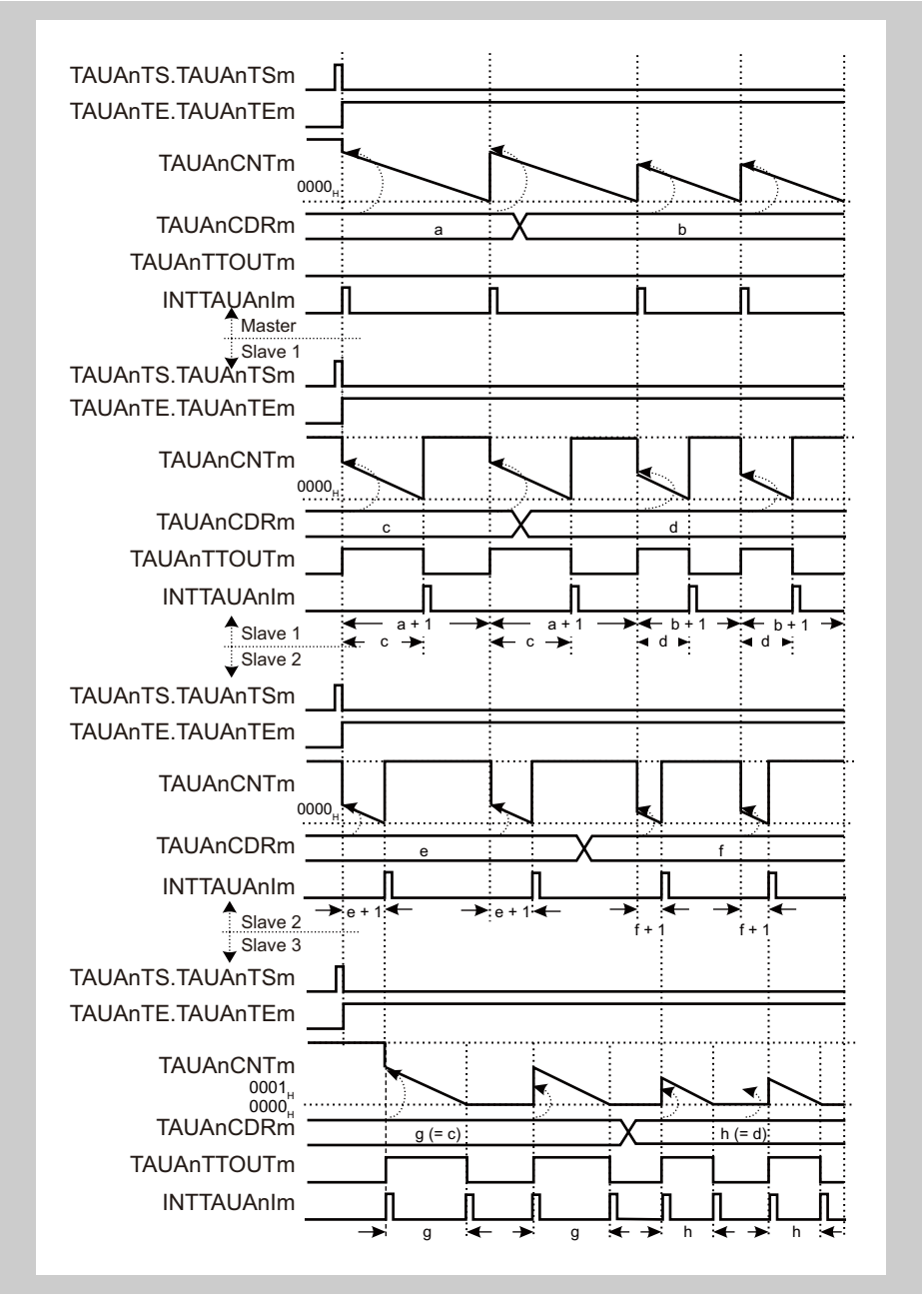


Figure 13-78 General Timing Diagram of Delay Pulse Output Function

(3) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-96 TAUAnCMORm Settings for Master Channels of the Delay Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-97 TAUAnCMURm Settings for Master Channels of Delay Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

TAUAnTOE.TAUAnTOEm is set to 0 because channel output mode is not used for master channels with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-98 Simultaneous Rewrite Settings for Master Channels of Delay Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: INTTAUAnIm signal used to trigger a simultaneous rewrite is not monitored on the channel. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(4) Register Settings for Slave Channel 1

(a) TAUAnCMORm for Slave Channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-99 TAUAnCMORm Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Valid start trigger during operation

(b) TAUAnCMURm for Slave Channel 1

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-100 TAUAnCMURm Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channel 1

Table 13-101 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	0: Synchronous operating mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-102 Simultaneous Rewrite Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: INTTAUAnIm signal used to trigger a simultaneous rewrite is not monitored on the channel. When TAUAnRDS.TAUAnRDSm = 0, the simultaneous rewrite trigger is monitored on the master channel regardless of the value of this bit.

(5) Register Settings for Slave Channel 2

(a) TAUAnCMORm for Slave Channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-103 TAUAnCMORm Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Valid start trigger during operation

(b) TAUAnCMURm for Slave Channel 2

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-104 TAUAnCMURm Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channel 2

TAUAnTOE.TAUAnTOEm is set to 0 because channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Slave Channel 2

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-105 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: INTTAUAnIm signal used to trigger a simultaneous rewrite is not monitored on the channel. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Register Settings for Slave Channel 3

(a) TAUAnCMORm for Slave Channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-106 TAUAnCMORm Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	101: INTTAUAnIm of upper channel (m - 1) is a start trigger regardless of master setting.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1010: Pulse one-count mode
TAUAnMD0	1: Valid start trigger during operation

(b) TAUAnCMURm for Slave Channel 3

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-107 TAUAnCMURm Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channel 3

Table 13-108 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channel 3

The simultaneous rewrite settings of master and slave channels should be identical.

Table 13-109 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: INTTAUAnIm signal used to trigger a simultaneous rewrite is not monitored on the channel. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(7) Operating Procedure for Delay Pulse Output Function

Table 13-110 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in (3) Register Settings for Master Channels. Slave channel 1: Set TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in (4) Register Settings for Slave Channel 1. Slave channel 2: Set TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in (5) Register Settings for Slave Channel 2. Slave channel 3: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in (6) Register Settings for Slave Channel 3. Set TAUAnCDRm register value of every channel.	Channel operation is stopped.

Table 13-110 Operating Procedure for Delay Pulse Output Function (2/2)

	Operation	TAUAn Status
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (slave channel 1) is set.
During Operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.RSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of master channel and slave channels 1 and 2 load TAUAnCDRm value and count down. When the counter of master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCDRm value is reloaded into TAUAnCNTm (master) to continue count operation. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave 1/2) to start countdown. • TAUAnTTOUTm (slave 1) is set. When TAUAnCNTm (slave 1) reaches 0000_H: • INTTAUAnIm (slave 1) is generated. • TAUAnTTOUTm (slave 1) is reset. When TAUAnCNTm (slave 2) reaches 0000_H: • INTTAUAnIm (slave 2) is generated. • TAUAnTTOUTm (slave 3) is set. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave 3) to start a countdown operation. When TAUAnCNTm (slave 3) reaches 0000_H: • INTTAUAnIm (slave 3) is generated. • TAUAnTTOUTm (slave 3) is reset.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

13.14.3 AD Conversion Trigger Output Function Type 1

(1) Overview

Summary This function is identical to section 13.14.1, PWM Output Function, except that TAUAnTTOUTm is not output.

This is achieved by setting the channel output mode for the slave to independent channel output mode controlled by software.

(2) Block Diagram and General Timing Diagram

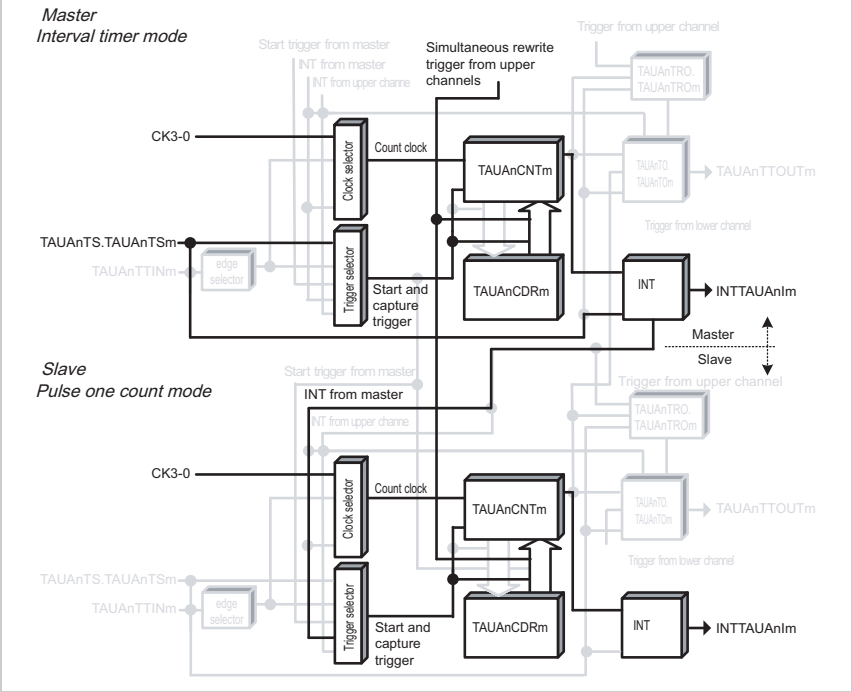


Figure 13-79 Block Diagram of AD Conversion Trigger Output Function Type 1

(3) General Timing Diagram

The following settings apply to the general timing diagram.

- Slave channels: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

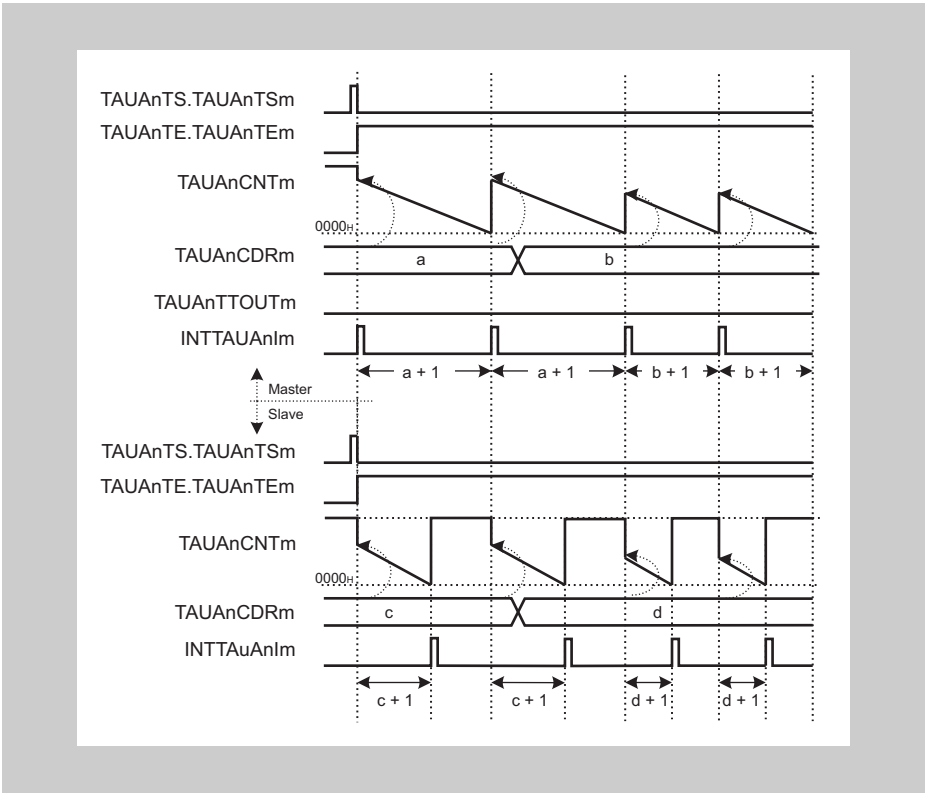


Figure 13-80 General Timing Diagram of AD Conversion Trigger Output Function Type 1

13.14.4 One-Shot Pulse Output Function

(1) Overview

Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See table 13-111, TAUAnCMORm Settings for Master Channels of One-Shot Pulse Output Function.)
- The operating mode for slave channels should be set to pulse one-count mode. (See table 13-114, TAUAnCMORm Settings for Slave Channels of One-Shot Pulse Output Function.)
- TAUAnTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channel should be set to independent channel output mode 2. (See section 13.8, Channel Output Modes.)
- TAUAnTTINm (master) has to be detected while TAUAnCNTm (master) and TAUAnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUAnTTINm (slave).

Functional description

The counters are enabled by setting the channel trigger bits (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm, enabling count operation.

- Master channel:

  When the next valid TAUAnTTINm input edge is detected, the current value of TAUAnCDRm is loaded into TAUAnCNTm. The counter starts to count down from this value. If TAUAnCMORm.MD0 = 0, a trigger (TAUAnTTINm) which is detected within the delay time is ignored.

  When the counter of master channel reaches 0000_H, INTTAUAnIm is generated. The counter is reset to $FFFF_H$ and waits for the next valid TAUAnTTINm input edge.

- Slave channels:

  INTTAUAnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUAnCDRm (slave) is loaded into TAUAnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUAnTTOUTm signal is set.

  When the counter reaches 0001_H, INTTAUAnIm is generated and TAUAnTTOUTm signal is reset. The counter stops at 0000_H and waits for the next INTTAUAnIm of master channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Setting TAUAnTS.TAUAnTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

Conditions

- If TAUAnCMORn.TAUAnMD0 of master channel is set to 0, TAUAnTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See section 13.7, Simultaneous Rewrite.

Equations

Delay to input pulse = (TAUAnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUAnCDRm (slave)) × count clock cycle

(2) Block Diagram and General Timing Diagram

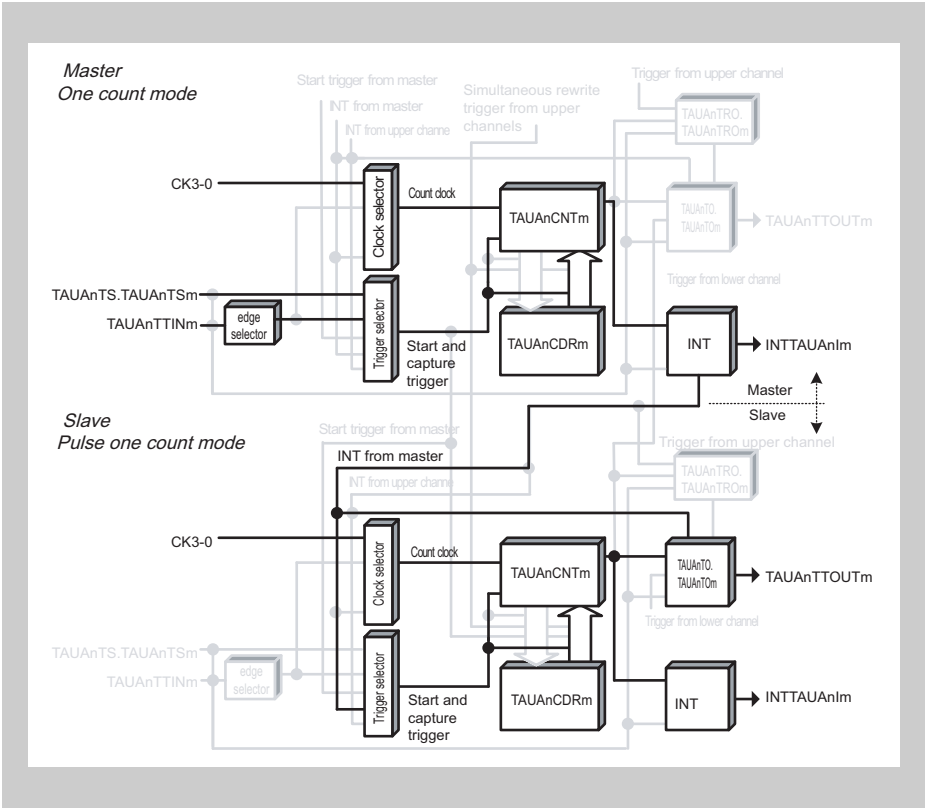


Figure 13-81 Block Diagram of One-Shot Pulse Output Function

The following settings apply to the general timing diagram.

- Start trigger detection is disabled during counting (TAUAnCMORm.TAUAnMD0 = 0)
- Detection of falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

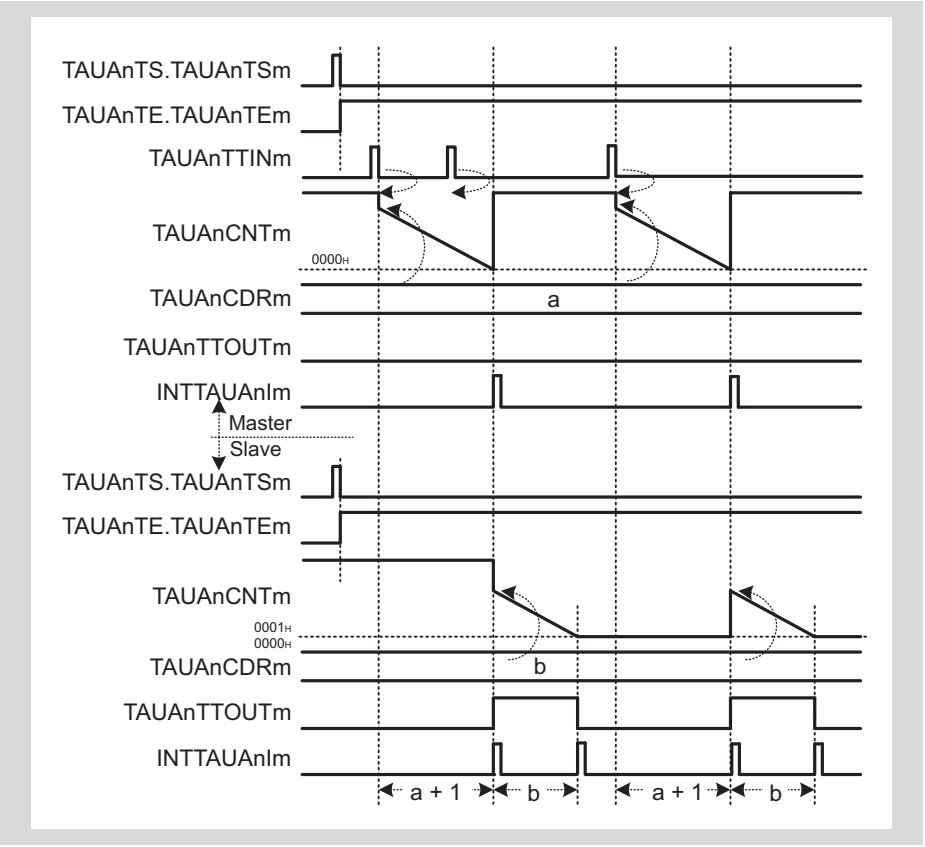


Figure 13-82 General Timing Diagram of One-Shot Pulse Output Function

(3) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-111 TAUAnCMORm Settings for Master Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. MD0 bit of master and slave channels should have the same value.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-112 TAUAnCMURm Settings for Master Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges 11: Setting prohibited.

(c) Channel Output Mode for Master Channels

TAUAnTOE.TAUAnTOEm is set to 0 because channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

The simultaneous rewrite settings for master and slave channel should be identical.

Table 13-113 Simultaneous Rewrite Settings for Master Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: INTTAUAnIm signal used to trigger a simultaneous rewrite is not monitored on the channel. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(4) Register Settings for Slave Channels

(a) TAUAnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-114 TAUAnCMORm Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1010: Pulse one-count mode
TAUAnMD0	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. MD0 bit of master and slave channels should have the same value.

(b) TAUAnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-115 TAUAnCMURm Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Output Mode for Slave Channels

Table 13-116 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set this bit to 0.
TAUAnTDL.TAUAnTDLm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set this bit to 0.
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set this bit to 0.
TAUAnTRC.TAUAnTRCm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set this bit to 0.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-117 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUAnRDM.TAUAnRDMm	0: Generates a simultaneous rewrite trigger signal when master channel starts to count.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used to trigger a simultaneous rewrite. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(5) Operating Procedure for One-Shot Pulse Output Function

Table 13-118 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (3) Register Settings for Master Channels. Slave channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (4) Register Settings for Slave Channels. Set the values of TAUAnCDRm register of all channels.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUAnTTINm input.
During Operation	TAUAnCDRm is changeable at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm are readable at any time. TAUAnRDT.TAUAnRDTm is changeable during operation.	When valid TAUAnTTINm input edge is detected, TAUAnCDRm value of master channel is loaded into TAUAnCNTm to start countdown. When the counter reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) returns to $FFFF_H$ again and waits for the next valid TAUAnTTINm input edge. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave) to start countdown. • INTTAUAnlm (slave) is generated. • TAUAnTTOUTm (slave) is set. When TAUAnCNTm (slave) reaches 0001_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is reset. If TAUAnTTINm input is detected on the master channel during count operation and TAUAnCMORm.TAUAnMD0 = 0, the input is ignored.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop, and retain their current values.

Restart

13.14.5 Offset Trigger Output Function

(1) Overview

Summary

This function generates a PWM output using a master channel and a slave channel, enabling the pulse width (duration) of the TAUAnTTOUTm to be set. The pulse cycle is set by detecting a valid input edge of master channel. The pulse width is specified on the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See table 13-119, TAUAnCMORm Settings for Master Channels of Offset Trigger Output Function.*)*
- The operating mode for slave channels should be set to one-count mode. (See table 13-122, TAUAnCMORm Settings for Slave Channels of Offset Trigger Output Function.*)*
- The output mode for slave channels should be set to independent channel output mode 2. (See section 13.8, Channel Output Modes.*)*
- TAUAnTTOUTm is not used with the master channel of this function.

Functional description

The counter can be started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This makes TAUAnTE.TAUAnTEm = 1, enabling the counter to count up. The master channel counter (TAUAnCNTm) starts to count up from 0000_H.

- Master channels:

  When a valid TAUAnTTINm input edge is detected, the current value of the counter (TAUAnCNTm) is loaded into the data register of master channel (TAUAnCDRm). INTTAUAnIm is generated and the counter restarts to count up from 0000_H.

- Slave channels:

  The INTTAUAnIm of master channel sets the TAUAnTTOUTm (slave) signal and triggers the counter of the slave channel. The current value of TAUAnCDRm (slave) is loaded into TAUAnCNTm (slave) and the counter starts to count down from this value.

  When the counter reaches 0000_H (duty time has elapsed), INTTAUAnIm is generated and TAUAnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of master channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for master and slave channels. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

(2) Equations

Pulse width = (TAUAnCDRm (slave) + 1) × count clock cycle

Duty cycle [%] = [TAUAnCDRm (slave)/TAUAnTTINm cycle + 1)] × 100

– Duty cycle = 0%

  TAUAnCDRm (slave) = 0000_H

– Duty cycle = 100%

  TAUAnCDRm (slave) ≥ TAUAnTTINm cycle + 1

(3) Block Diagram and General Timing Diagram

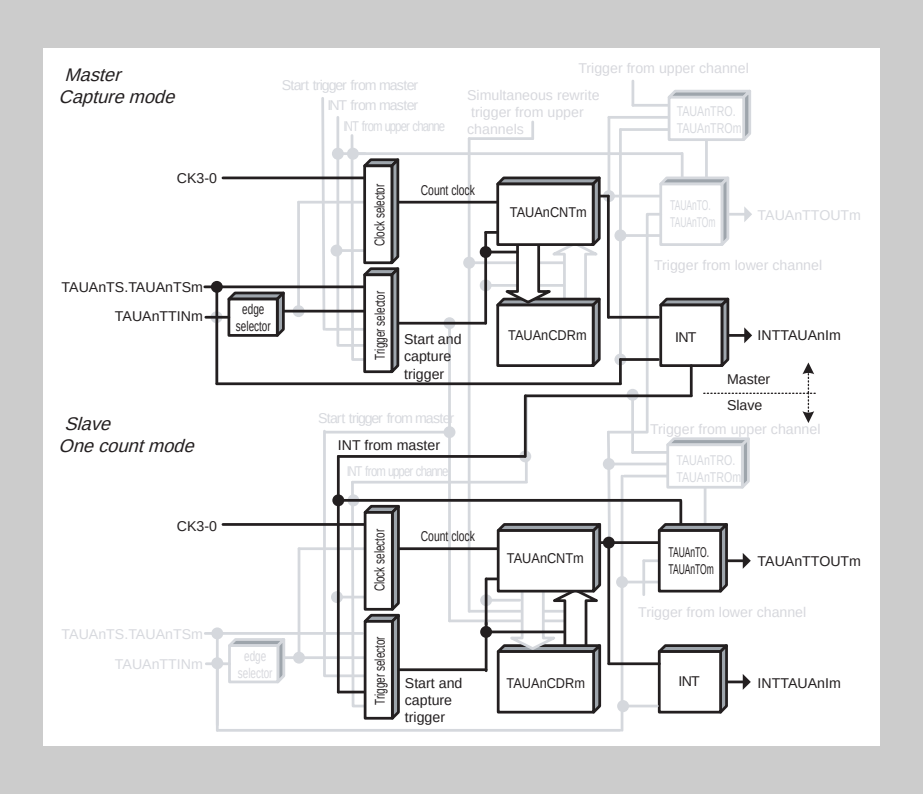


Figure 13-83 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

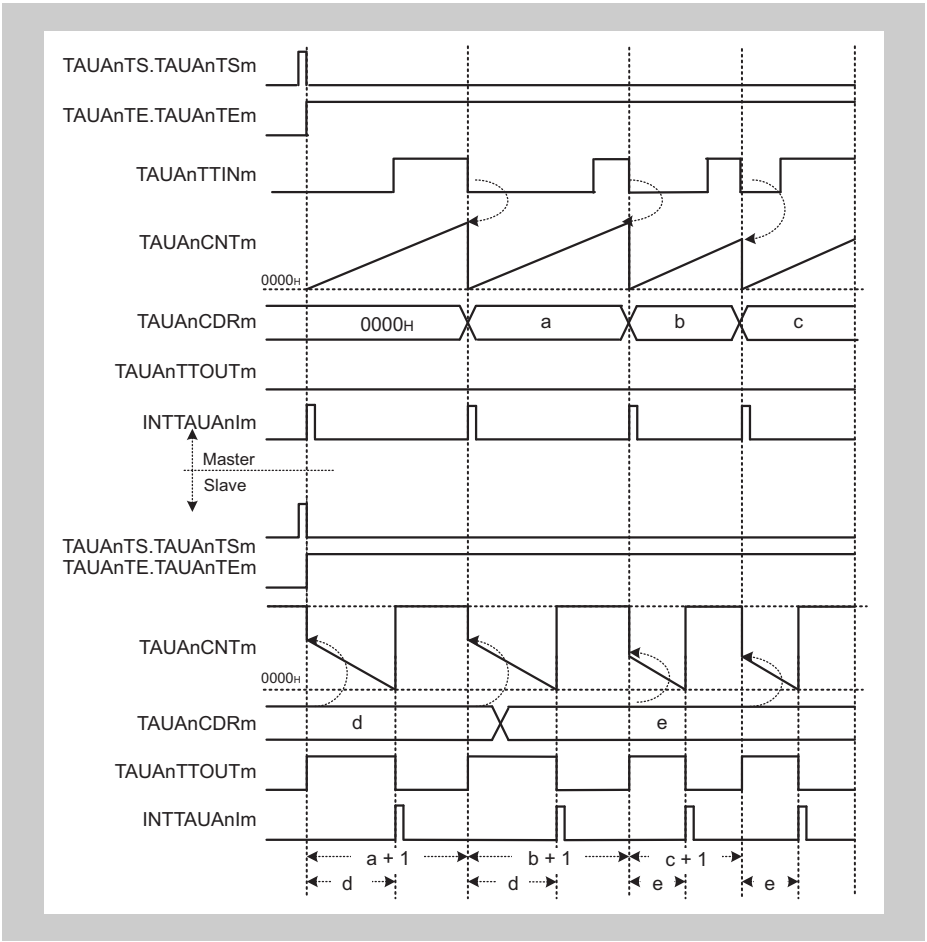


Figure 13-84 General Timing Diagram of Offset Trigger Output Function

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-119 TAUAnCMORm Settings for Master Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as a start trigger.
TAUAnCOS[1:0]	11: Capture register is updated upon detection of a valid TAUAnTTINm input edge or when a counter overflow occurs: • Detection of valid TAUAnTTINm input edge: The counter value is written into TAUAnCDRm. • Occurrence of overflow: $FFFF_H$ is written into TAUAnCDRm. A valid TAUAnTTINm input edge to be detected next is ignored. TAUAnCSRm.TAUAnOVF is set when a counter overflow occurs, and cleared by setting TAUAnCSCm.TAUAnCLOV = 1.
TAUAnMD[4:1]	0010: Capture mode
TAUAnMD0	1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-120 TAUAnCMURm Settings for Master Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges 11: Disables setting.

(c) Channel Output Mode for Master Channels

TAUAnTOE.TAUAnTOEm is set to 0 because channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-121 Simultaneous Rewrite Settings for Master Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Register Settings for Slave Channels

(a) TAUAnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-122 TAUAnCMORm Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Enables start trigger detection during counting.

(b) TAUAnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-123 TAUAnCMURm Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channels

Table 13-124 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channels

Simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 13-125 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0.
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(6) Operating Procedure for Offset Trigger Output Function

Table 13-126 Operating Procedure for Offset Trigger Output Function

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (4) Register Settings for Master Channels. Slave channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (5) Register Settings for Slave Channels. Set the value of TAUAnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start: • TAUAnCNTm (master) counts up. • TAUAnCDRm value is loaded into TAUAnCNTm (slave) to perform count down. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (slave) is set.
During Operation	TAUAnCDRm (slave) can be changed immediately after INTTAUAnIm (master) is generated. TAUAnCSCm.TAUAnCLOV can be set to 1.	When TAUAnCNTm of slave channel reaches 0000_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is reset. When TAUAnTTINm input edge is detected on the master channel: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) is reset to 0000_H and then continues count operation subsequently. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave) to perform count down. • TAUAnTTOUTm (slave) is set.
Stop Operation	Set TAUAnTT.TAUAnTTm for master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

13.14.6 Triangle PWM Output Function

(1) Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUAnTTOUTm to be set using the master and slave channels respectively.
The master channel generates a carrier cycle. The first cycle of master channel controls the down status and the second cycle controls the up status of the slave counter.

Prerequisites

- Two channels
- The operating mode for master channels should be set to interval timer mode. (See table 13-127, TAUAnCMORm Settings for Master Channels of Triangle PWM Output Function*.)*
- The operating mode for slave channels should be set up/down count mode. (See table 13-131, TAUAnCMORm Settings for Slave Channels of Triangle PWM Output Function*.)*
- The channel output mode for master channels should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See section 13.8, Channel Output Modes.)
- The following settings allows TAUAnTTOUTm to be at high level during the down status of a carrier cycle.
  - If TAUAnCMORm.TAUAnMD0 (master) bit is set to 0, TAUAnTO.TAUAnTOm should be set to 1 while TAUAnTOE.TAUAnTOEm is set to 0 (recommended setting).
  - If TAUAnCMORm.TAUAnMD0 (master) bit is set to 1, TAUAnTO.TAUAnTOm should be set to 0 while TAUAnTOE.TAUAnTOEm is set to 0 (recommended setting).

Functional description

The counters are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1 for every channel. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation.
The current values of TAUAnCDRm (master and slave) are loaded into TAUAnCNTm (master and slave) and the counters start counting down from these values. When the TAUAnCMORm.TAUAnMD0 bit of master channel is set to 1, an interrupt is generated and TAUAnTTOUTm signal of master toggles.

- Master channels:

  When the counter of master channel reaches 0000_H (pulse cycle time has elapsed), INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. TAUAnCNTm then reloads the TAUAnCDRm value and counts down.

- Slave channels:

  The INTTAUAnIm of master channel triggers the counter of the slave channel:

  – If the slave counter is counting down, the count direction changes.

  – If the slave counter is counting up, TAUAnCDRm value is reloaded and the counter starts to count down.

  When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUAnIm is generated and the TAUAnTTOUTm (slave) signal is set/reset.

  The counter continues count-up/-down and waits for the next INTTAUAnlm of master channel.

  Setting TAUAnTOL.TAUAnTOLm allows TAUAnTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values.

Conditions This function enables simultaneous rewrite. See section 13.7, Simultaneous Rewrite.

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUAnCDRm (master) < $FFFF_H$

Carrier cycle (down/up) = (TAUAnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle [%] =
[(TAUAnCDRm (master) + 1 - TAUAnCDRm (slave))/(TAUAnCDRm (master) + 1)] × 100

– Duty cycle = 100 %

  TAUAnCDRm (slave) = 0000_H

– Duty cycle = 0 %

  TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

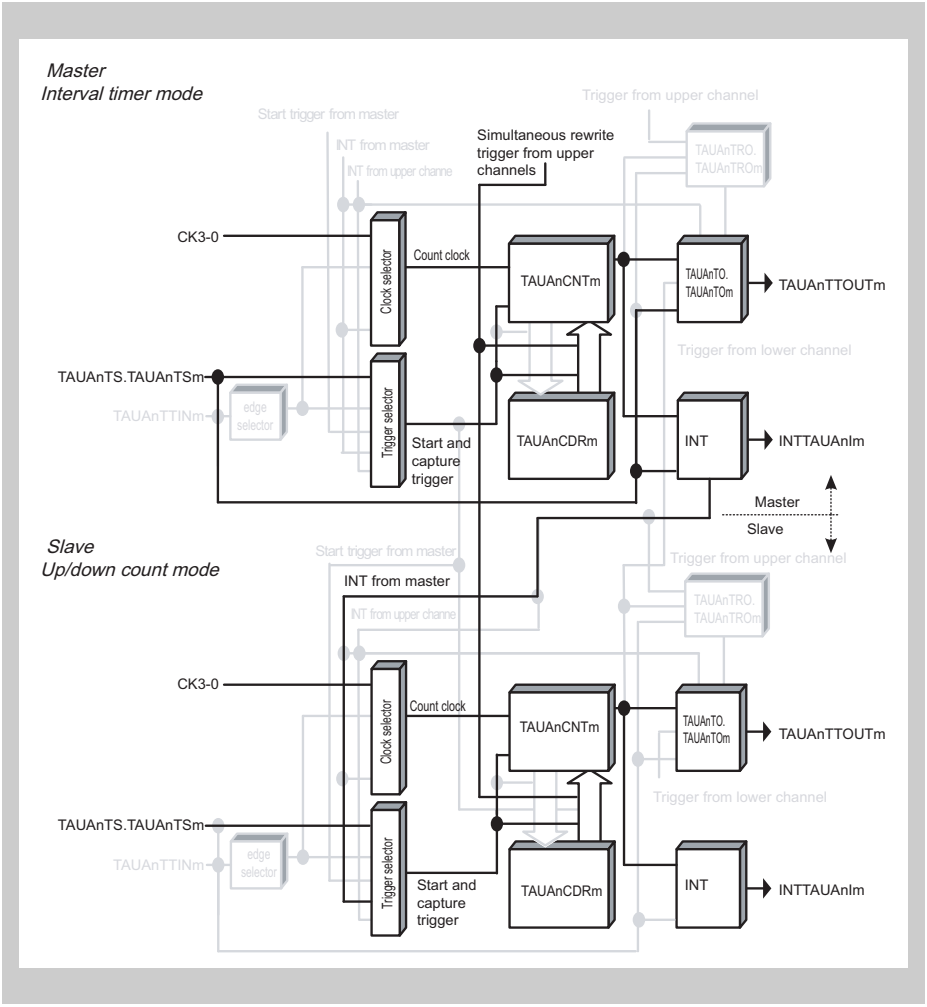


Figure 13-85 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)

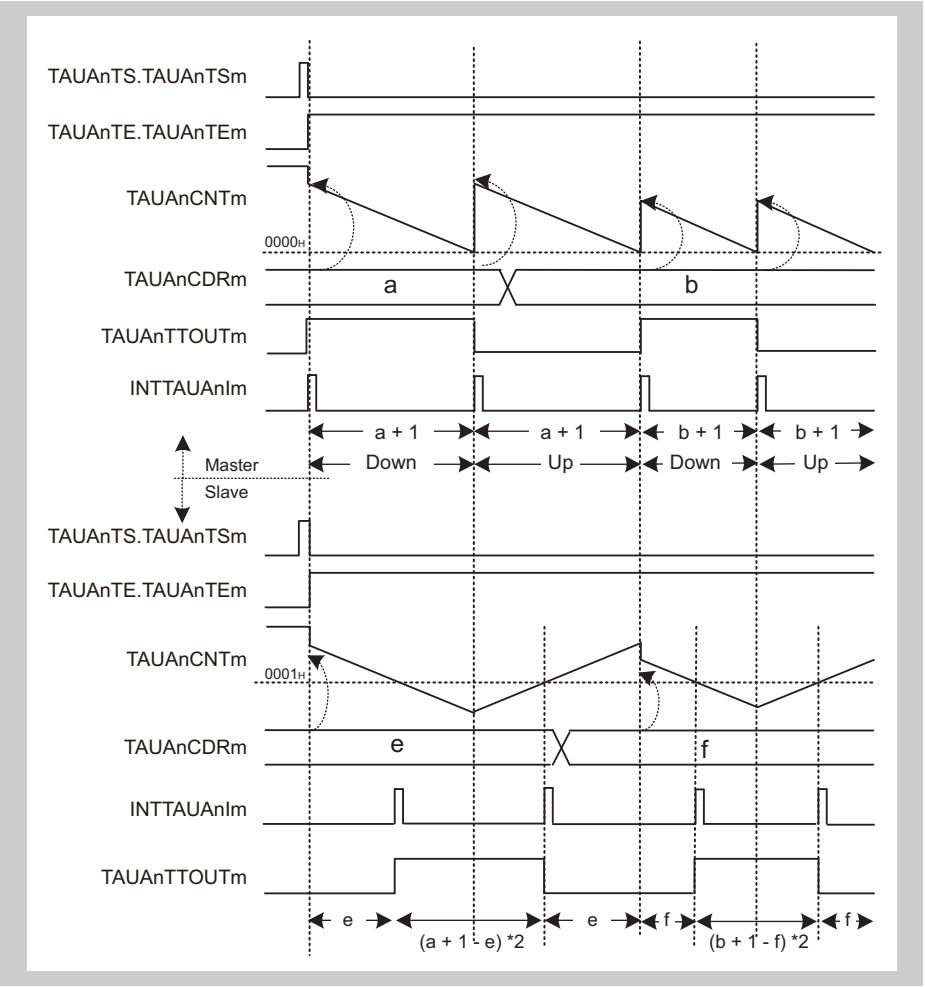


Figure 13-86 General Timing Diagram of Triangle PWM Output Function

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS[1:0]		TAUAnCCS[1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS[1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-127 TAUAnCMORm Settings for Master Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated and TAUAnTTOUTm is not toggled at the beginning of operation. 1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-128 TAUAnCMURm Settings for Master Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

Table 13-129 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (Toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Master Channels

The simultaneous rewrite settings of master and slave channels should be identical.

Table 13-130 Simultaneous Rewrite Settings for Master Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note If TAUAnRDS.TAUAnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

(5) Register Settings for Slave Channels

(a) TAUAnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-131 TAUAnCMORm Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	111: Up/down output trigger signal of master channel
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1001: Up/down count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation.

(b) TAUAnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-132 TAUAnCMURm Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channels

Table 13-133 Control Bit Settings in Synchronous Channel Output Mode 2

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Slave Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-134 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel ismonitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Operating Procedure for Triangle PWM Output Function

Table 13-135 Operating Procedure for Triangle PWM Output Function

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (4) Register Settings for Master Channels. Slave channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channels Set the value of TAUAnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counter of master/ slave channel starts. INTTAUAnIm (master) is generated on the master channel if TAUAnCMORm.TAUAnMD0 is set to 1.
During Operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCDRm value of master and slave channels is loaded into TAUAnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnTTOUTm (master) is toggled. • TAUAnCDRm value is reloaded into TAUAnCNTm (master) to continue count operation. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave) or counting is started in opposite direction. When TAUAnCNTm of slave channel reaches 0001_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is set in the count-down status or reset in count-up status.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific Timing Diagrams

(a) Duty Cycle = 0%

The following settings apply to the general timing diagram.

- Master channels:
  - INTTAUAnIm is generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 1).
  - TAUAnCDRm = a = 5_H
- Slave channels:
  - TAUAnCDRm = 6_H

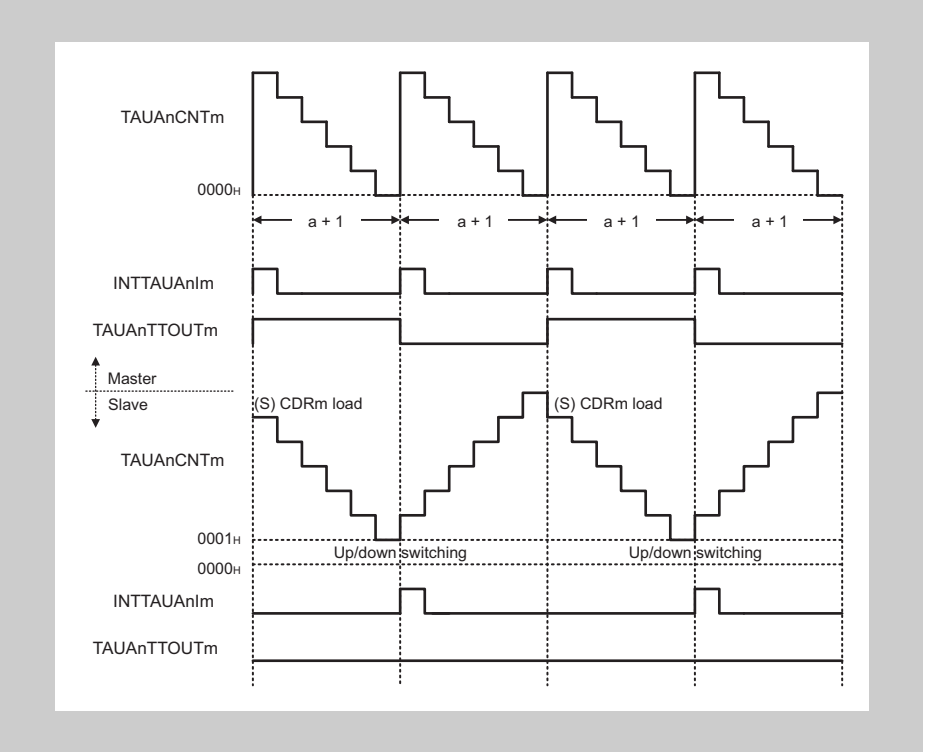


Figure 13-87 TAUAnCDRm (Slave) ≥ TAUAnCDRm (Master) + 1

- If TAUAnCDRm (slave) value is greater than TAUAnCDRm (master) value, INTTAUAnIm is not generated while the counter of slave channel is counting down. TAUAnTTOUTm remains at the inactive level because there is no set signal of TAUAnTTOUTm to be detected.

(b) Duty Cycle = 100%

The following settings apply to the general timing diagram.

- Master channels:
  - INTTAUAnIm is generated at the beginning of operation (TAUAnCMORm.TAUAnMD0 = 1)
  - TAUAnCDRm = a = 5_H
- Slave channels:
  - TAUAnCDRm = 0_H

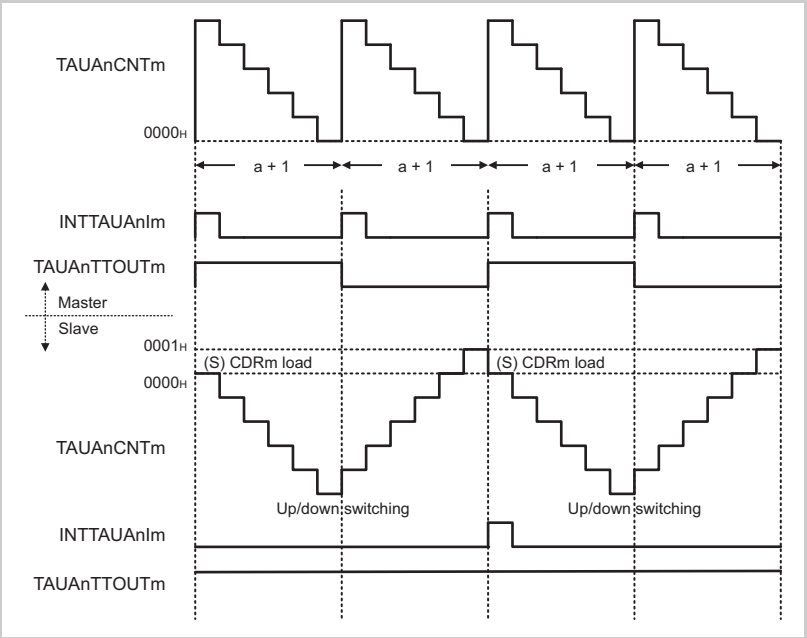


Figure 13-88 TAUAnCDRm (Slave) = 0000_H

- If TAUAnCDRm (slave) = 0000_H, INTTAUAnIm is not generated while the counter of slave channel is counting up. TAUAnTTOUTm remains at the active level because there is no reset signal of TAUAnTTOUTm to be detected.

13.14.7 Triangle PWM Output Function with Dead Time

(1) Overview

Summary This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals are output via TAUAnTTOUTm of the slave channel 3, enabling the pulse cycle (frequency) and the duty cycle of TAUAnTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on master channel. The first pulse controls the down status of slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUAnTTOUTm of slave channels to be set/reset. Depending on the settings of TAUAnTDL.TAUAnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUAnTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select even channel CH (a) and odd channel CH (a + 1).
- The operating mode for master channels should be set to interval timer mode. (See table 13-137, TAUAnCMORm Settings for Master Channels of Triangle PWM Output Function with Dead Time*.)*
- The operating mode for slave channel 2 should be set to up/down count mode. (See table 13-141, TAUAnCMORm Settings for Slave Channel 2 of Triangle PWM Output Function with Dead Time*.)* Slave channel 2 should be an even channel.
- The operating mode for slave channel 3 should be set to one-count mode. (See table 13-145, TAUAnCMORm Settings for Slave Channel 3 of Triangle PWM Output Function with Dead Time*.)* Slave channel 3 should be an odd channel.
- The channel output mode for master channels should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes*.)*
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See section 13.8, Channel Output Modes*.)*
- The following settings make a TAUAnTTOUTm signal at high level during the down status of the carrier cycle:
  - If TAUAnCMORm.TAUAnMD0 (master) bit is set to 0, TAUAnTO.TAUAnTOm should be set to 1 while TAUAnTOE.TAUAnTOEm is set to 0. (recommended setting)
  - If TAUAnCMORm.TAUAnMD0 (master) bit is set to 1, TAUAnTO.TAUAnTOm should be set to 0 while TAUAnTOE.TAUAnTOEm is set to 0.

Note The triangle PWM output function with dead time does not use slave channel 1.

Functional description

The counter starts by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This makes TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded into TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value. If TAUAnCMORm.TAUAnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUAnTTOUTm signal is toggled.

- Master channels:

  When the counter of master channel reaches 0000_H, an INTTAUAnIm is generated and the TAUAnTTOUTm signal is toggled. The TAUAnCDRm value is reloaded to continue countdown.

- Slave channel 2:

  If INTTAUAnIm is generated on the master channel, the counter of slave channel 2 is triggered.

  – If the slave counter is counting down, the counting direction changes.

  – If the slave counter is counting up, the TAUAnCDRm value is reloaded and the counter starts to count down.

  The counter continues to count down/up and waits for the next INTTAUAnlm of master channel.

- Slave channel 3:

  If INTTAUAnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUAnCDRm (slave 3) is loaded into TAUAnCNTm (slave 3) and the counter starts to count down from the TAUAnCDRm value.

  When the counter reaches 0000_H, INTTAUAnIm occurs. The counter returns to $FFFF_H$ and waits for the next INTTAUAnlm of slave channel 2.

As described in table 13-136, Operation of TAUAnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2, the set/reset timing (right after occurrence of an interrupt or after dead time has elapsed) depends on the TAUAnTDL.TAUAnTDLm setting of the corresponding channel.

The setting of TAUAnTOL.TAUAnTOLm also determines whether a high level signal (TAUAnTOL.TAUAnTOLm = 0) or a low level signal (TAUAnTOL.TAUAnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values.

TAUAnTTOUTm can be 100 % output by setting the TAUAnCDRm value of slave channel 2 to 0000_H.

Conditions

This function enables simultaneous rewrite. See section 13.7, Simultaneous Rewrite.

TAUAnTOL.TAUAnTOLm and TAUAnTDL.TAUAnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUAnTOL.TAUAnTOLm or TAUAnTDL.TAUAnTDLm.

Table 13-136 Operation of TAUAnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

TAUAnTDL. TAUAnTDLm	Count Direction of Slave Channel 2 upon Occurrence of Interrupt	TAUAnTTOUTm Set/Reset Timing
0	Down	Set after elapse of dead time
	Up	Reset right after interrupt occurs
1	Down	Set right after interrupt occurs
	Up	Reset after elapse of dead time

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUAnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUAnCDRm (master) +1) × 2 × count clock cycle

PWM signal width (normal phase) = [(TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2)) × 2 - (TAUAnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase) = [(TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2)) × 2 + (TAUAnCDRm (slave 3) + 1)] × count clock cycle

(3) Block Diagram and General Timing Diagram

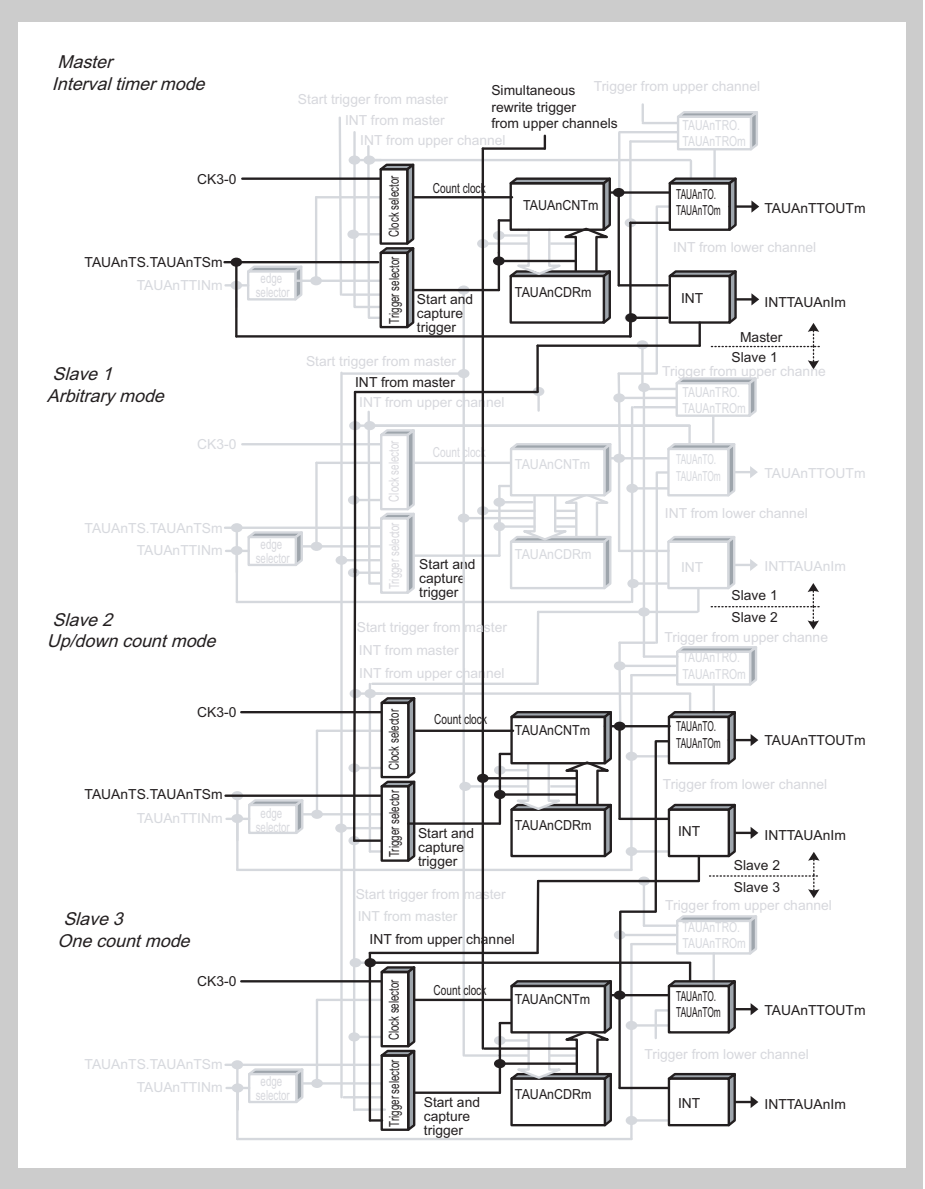


Figure 13-89 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channels:
  - INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)
- Slave channel 2:
  - INTTAUAnIm is not generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 0)
  - TAUAnTDL.TAUAnTDLm = 0
  - Positive logic (TAUAnTOL.TAUAnTOLm = 0)
- Slave channel 3:
  - INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)
  - TAUAnTDL.TAUAnTDLm = 1
  - Negative logic (TAUAnTOL.TAUAnTOLm = 1)

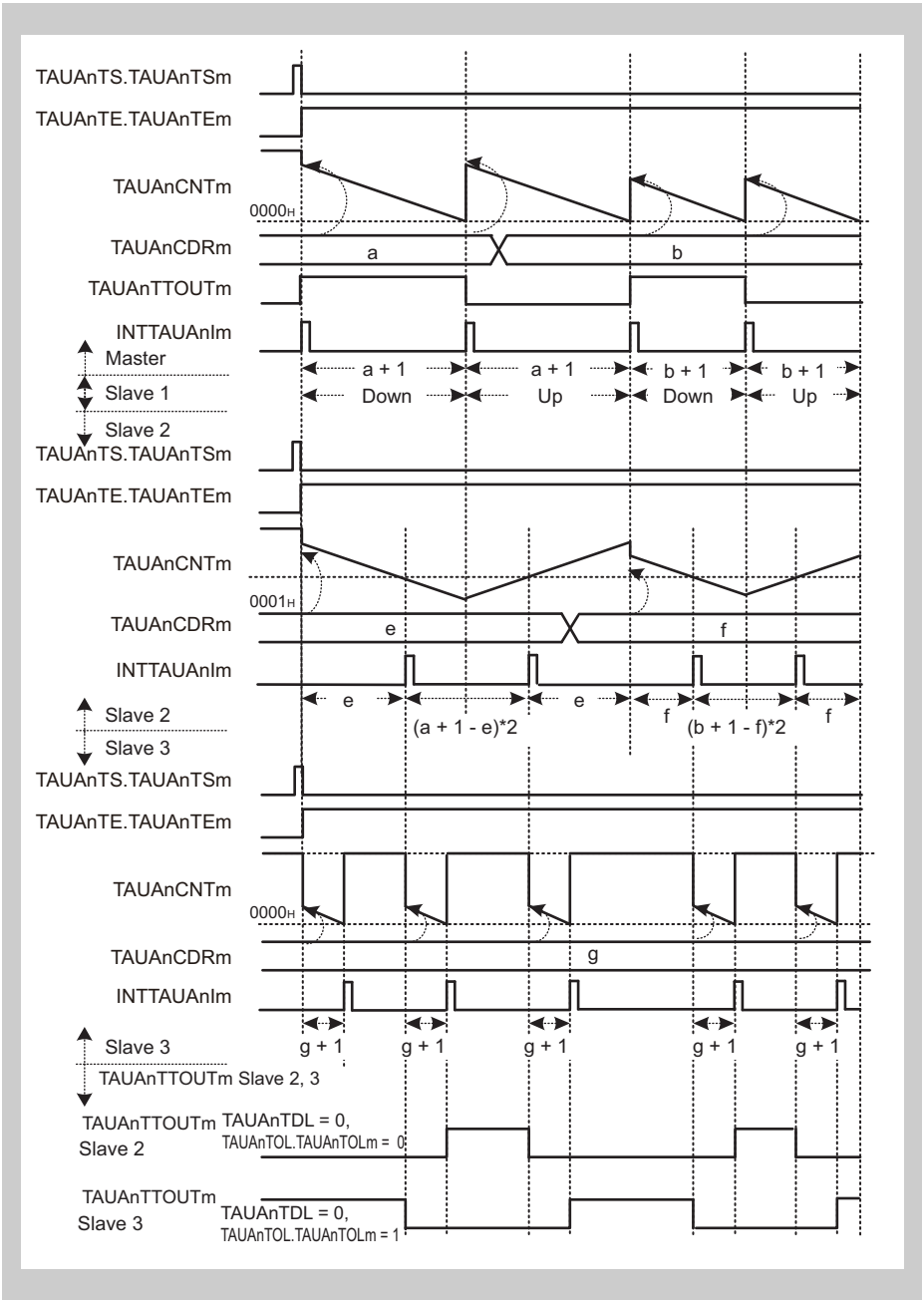


Figure 13-90 General Timing Diagram of Triangle PWM Output Function with Dead Time

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-137 TAUAnCMORm Settings for Master Channels of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated and TAUAnTTOUTm is not toggled at the beginning of operation. 1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-138 TAUAnCMURm Settings for Master Channels of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

Table 13-139 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-140 Simultaneous Rewrite for Master Channels of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note If TAUAnRDS.TAUAnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

(5) Register Settings for Slave Channel 2

(a) TAUAnCMORm for Slave Channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-141 TAUAnCMORm Settings for Slave Channel 2 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	111: Up/down output trigger signal of master channel
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1001: Up/down count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation.

(b) TAUAnCMURm for Slave Channel 2

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-142 TAUAnCMURm Settings for Slave Channel 2 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channel 2

Table 13-143 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation.
TAUAnTDM.TAUAnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUAnTDL.TAUAnTDLm are satisfied.
TAUAnTDL.TAUAnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Caution Set TAUAnTDLm exclusively from odd channels.

(d) Simultaneous Rewrite for Slave Channel 2

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-144 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Register Settings for Slave Channel 3

(a) TAUAnCMORm for Slave Channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-145 TAUAnCMORm Settings for Slave Channel 3 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	110: Dead time trigger
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Enables start trigger detection during counting.

(b) TAUAnCMURm for Slave Channel 3

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-146 TAUAnCMURm Settings for Slave Channel 3 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channel 3

Table 13-147 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation.
TAUAnTDM.TAUAnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUAnTDL.TAUAnTDLm are satisfied.
TAUAnTDL.TAUAnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

Caution Set TAUAnTDL.TAUAnTDLm exclusively from even channels.

(d) Simultaneous Rewrite for Slave Channel 3

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-148 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(7) Operating Procedure for Triangle PWM Output Function with Dead Time

Table 13-149 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (4) Register Settings for Master Channels. Slave channel 2: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channel 2. Slave channel 3: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (6) Register Settings for Slave Channel 3. Set the value of TAUAnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counter of master/slave channel starts. INTTAUAnIm (master) is generated on the master channel if TAUAnCMORm.TAUAnMD0 is set to 1.
During Operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCDRm value of master channel and slave channel 2 is loaded into TAUAnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCDRm value is reloaded into TAUAnCNTm (master) to continue count operation. • TAUAnCDRm value is reloaded into TAUAnCNTm (slave 2) or counting is started in opposite direction. When TAUAnCNTm of slave channel 2 reaches 0001_H: • INTTAUAnIm (slave 2) is generated. • TAUAnCDRm value of slave channel 3 is loaded into TAUAnCNTm perform counting down. When TAUAnCNTm of slave channel 3 reaches 0000_H: • INTTAUAnIm is generated.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(8) Specific Timing Diagrams

(a) Duty Cycle = 0%

The following settings apply to the general timing diagram.

- Slave channel 2:
  - Positive logic (TAUAnTDL.TAUAnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUAnTDL.TAUAnTDLm = 1)

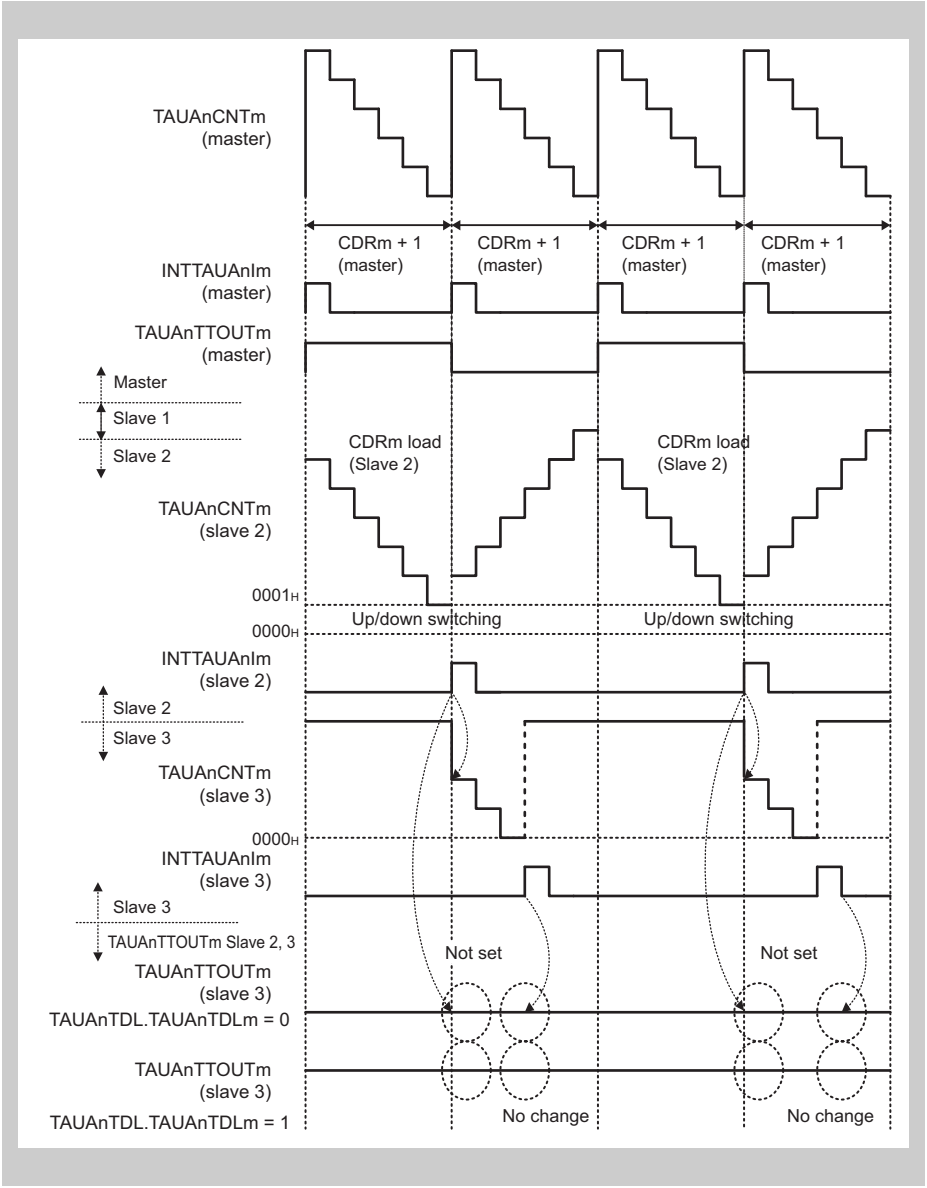


Figure 13-91 TAUAnCDRm (Slave 2) ≥ TAUAnCDRm (Master) + 1

- If TAUAnCDRm (slave 2) is greater than TAUAnCDRm (master), INTTAUAnIm is not generated while the counter of slave channels 2 or 3 is counting down. TAUAnTTOUTm remains at the inactive level because there is no set signal of TAUAnTTOUTm to be detected.

(b) Duty Cycle = 100%

The following settings apply to the general timing diagram.

- Slave channel 2:
  - Positive logic (TAUAnTDL.TAUAnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUAnTDL.TAUAnTDLm = 1)

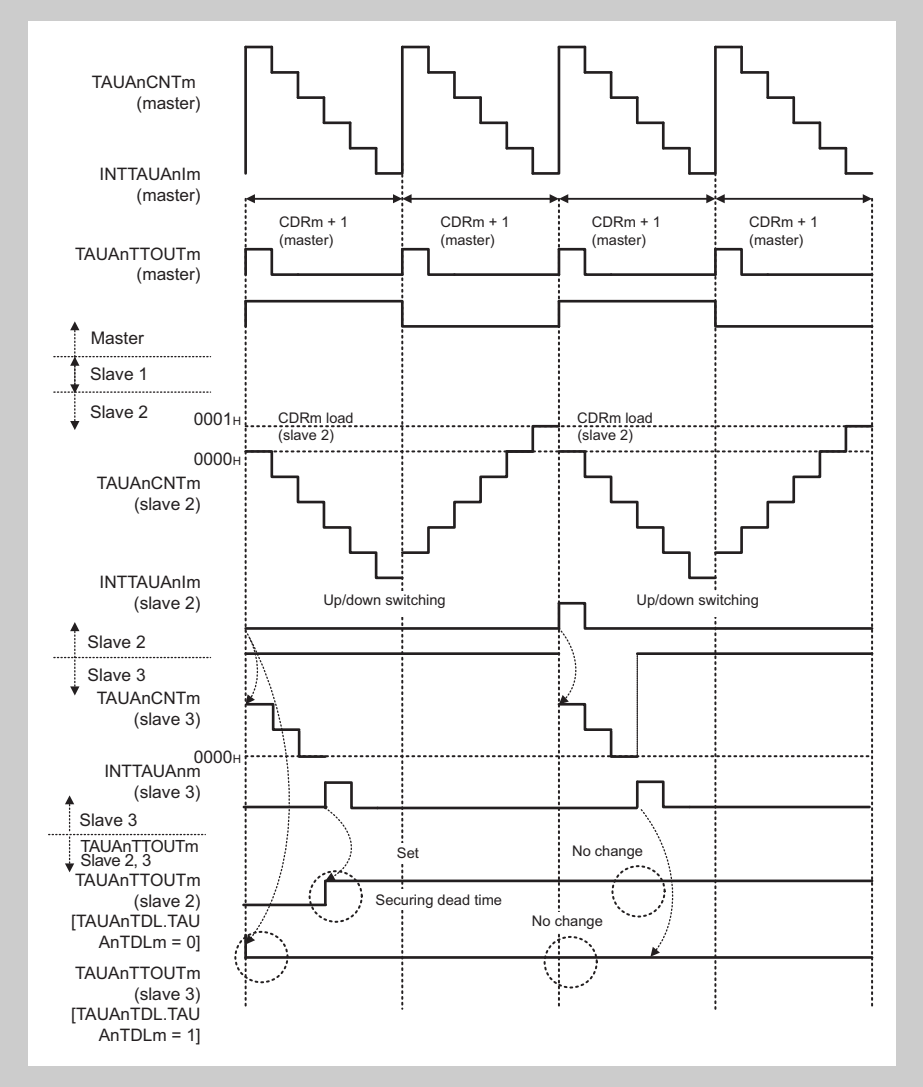


Figure 13-92 TAUAnCDRm (Slave 2) = 0000_H

- If TAUAnCDRm (slave 2) = 0000_H, INTTAUAnIm is not generated while the counter of slave channel is counting up. TAUAnTTOUTm remains at the active level because there is no reset signal of TAUAnTTOUTm to be detected. After setting TAUAnTS.TAUAnTSm = 1, inverse phase output of TAUAnTTOUTm is forcibly set and remains at the active level thereafter; positive phase output is set after securing dead time and remains at the active level thereafter.

13.14.8 AD Conversion Trigger Output Function Type 2

(1) Overview

Summary This function is identical to section 13.14.6, Triangle PWM Output Function, except that TAUAnTTOUTm is not output.

This function is enabled by setting channel output mode for the slave to independent channel output mode controlled by software.

(2) Block Diagram and General Timing Diagram

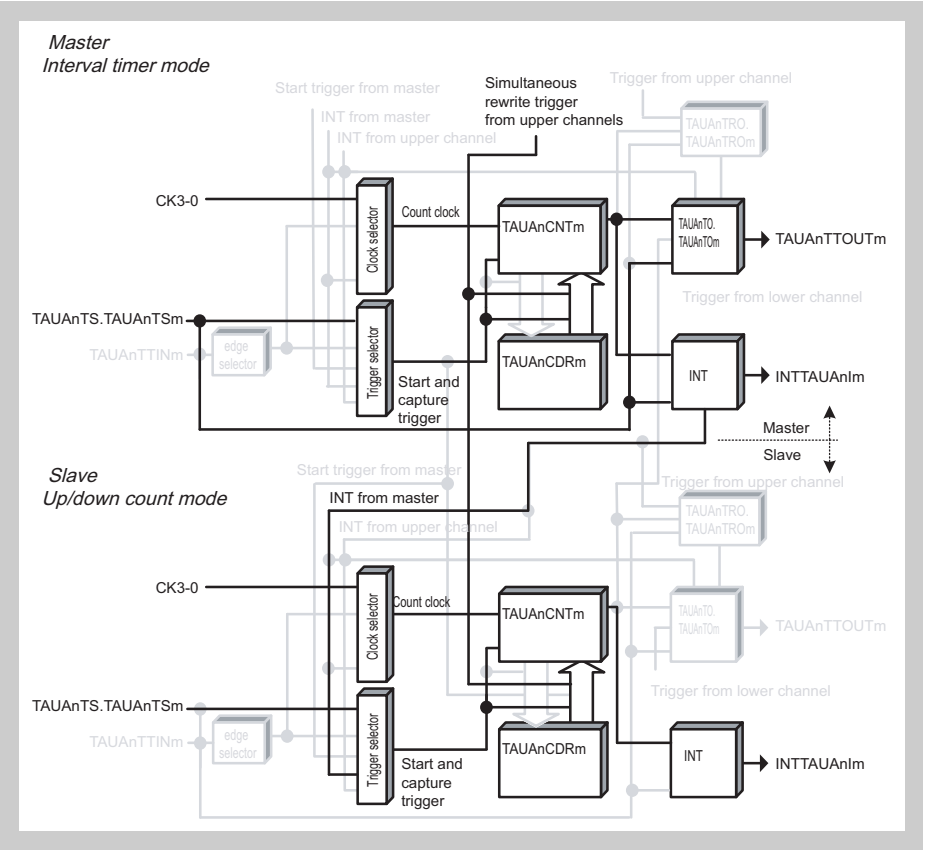


Figure 13-93 Block Diagram of AD Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 0)

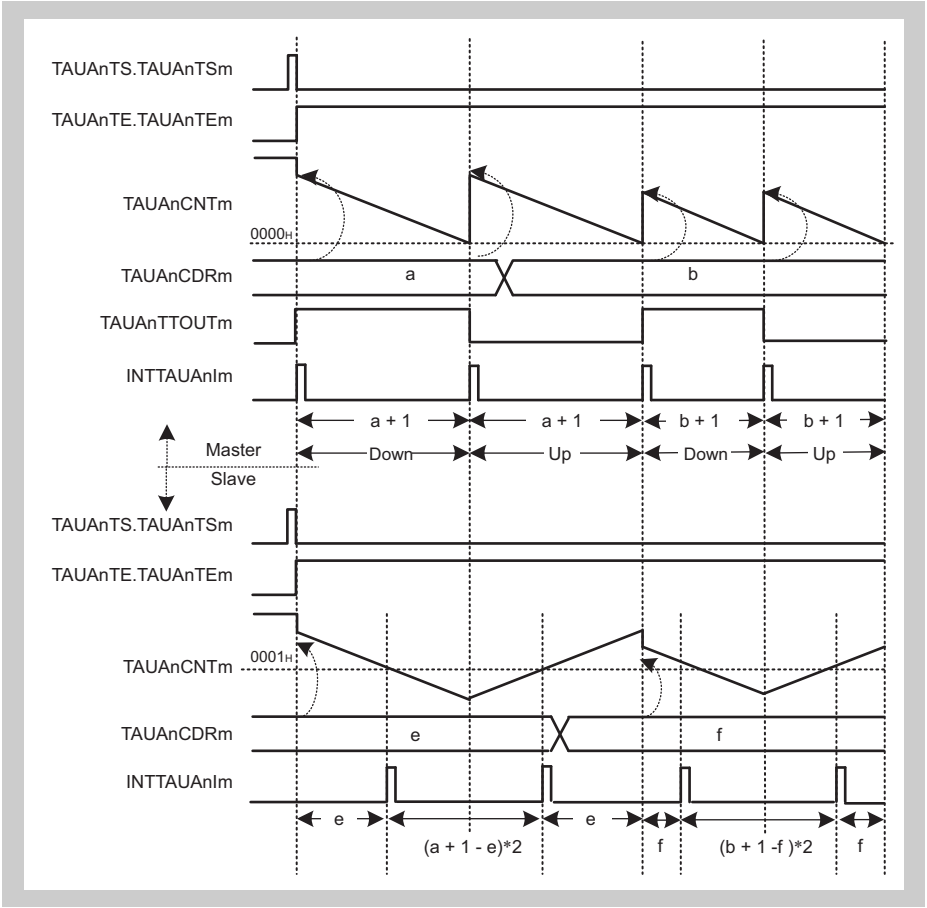


Figure 13-94 General Timing Diagram of AD Conversion Trigger Output Function Type 2

13.14.9 Non-Complementary Modulation Output Function Type 1

(1) Overview

Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUAnTTOUTm depending on the values of the real-time output bits (TAUAnTRO.TAUAnTROm) and the modulation output enable bits (TAUAnTME.TAUAnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operating mode for master channel should be set to interval timer mode. (See table 13-151, TAUAnCMORm Settings for Master Channels of Non-Complementary Modulation Output Function Type 1.)
- The operating mode for slave channels 1 to 7 should be set to one-count mode. (See table 13-154, TAUAnCMORm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1.)
- TAUAnTTOUTm is not used with the master channel of this function.
- TAUAnTTOUTm of slave channel 1 is not used with this function, but TAUAnTRC.TRCm should be set to 1. (See section 13.8, Channel Output Modes.)
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output. (See section 13.8, Channel Output Modes.)
- TAUAnCDRm of slave channel 1 should be set to 0000_H.

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The value of data register (TAUAnCDRm) is loaded into the counter (TAUAnCNTm) and the counter starts to count down. When the counter reaches 0000_H, INTTAUAnlm occurs.

- Slave channel 1:

  Slave channel 1 is set as a channel that triggers real-time output (TAUAnTRC.TAUAnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUAnCDRm is fixed to 0000_H), the value of real-time output bit (TAUAnTRO.TAUAnTROm) of the channel that has detected the interrupt on slave channel 1 changes. After that, the counter returns to $FFFF_H$ and waits for the next interrupt of master channel.

- Slave channel 2:

  Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in table 13-150, TAUAnTTOUTm Output from One Pair of Slave Channels of Non-Complementary Modulation Output Function Type 1, a signal output from TAUAnTTOUTm depends on the value of the real-time output bit (TAUAnTRO.TAUAnTROm) and modulation output bit (TAUAnTME.TAUAnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and

slave channels stop but retain their current values. The counter can be restarted by setting TAUAnTS.TAUAnTSm.

Conditions

- If TAUAnTME.TMEm = 0 on slave channels 2 to 7:
  - If the channel's TAUAnTRO.TAUAnTROm is set to 1, TAUAnTTOUTm outputs a high-level signal.
  - If the channel's TAUAnTRO.TAUAnTROm is set to 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTME.TMEm = 1 on slave channels 2 to 7:
  - If the channel's TAUAnTRO.TAUAnTROm is set to 1, TAUAnTTOUTm outputs PWM corresponding to the channel.
  - If the channel's TAUAnTRO.TAUAnTROm is set to 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTOL.TAUAnTOLm is set to 1, high-level and low-level signals output from TAUAnTTOUTm are inverted. There is no impact on PWM signal. Only the initial setting of TAUAnTOL.TAUAnTOLm is permitted (cannot be changed during operation).

Table 13-150 TAUAnTTOUTm Output from One Pair of Slave Channels of Non-Complementary Modulation Output Function Type 1

TAUAnTME. TAUAnTMEm	TAUAnTRO. TAUAnTROm	TAUAnTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWMm

- This function enables simultaneous rewrite. See section 13.7, Simultaneous Rewrite.
- TAUAnCDRm value of slave channel 1 should be set to 0000_H so that a real-time output is triggered at the same time with PWM generation on slave channels 2 to 7.
- If TAUAnTOL.TAUAnTOLm is set to 0 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set low before TAUAnTE.TAUAnTEm is set to 0.
- If TAUAnTOL.TAUAnTOLm is set to 1 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set high before TAUAnTE.TAUAnTEm is set to 0.

(2) Equations

Slave channels 2 to 7:

PWM output cycle = [TAUAnCDRm (master) + 1] × count clock

PWM output duty time = [TAUAnCDRm (slave)] × count clock

(3) Block Diagram and General Timing Diagram

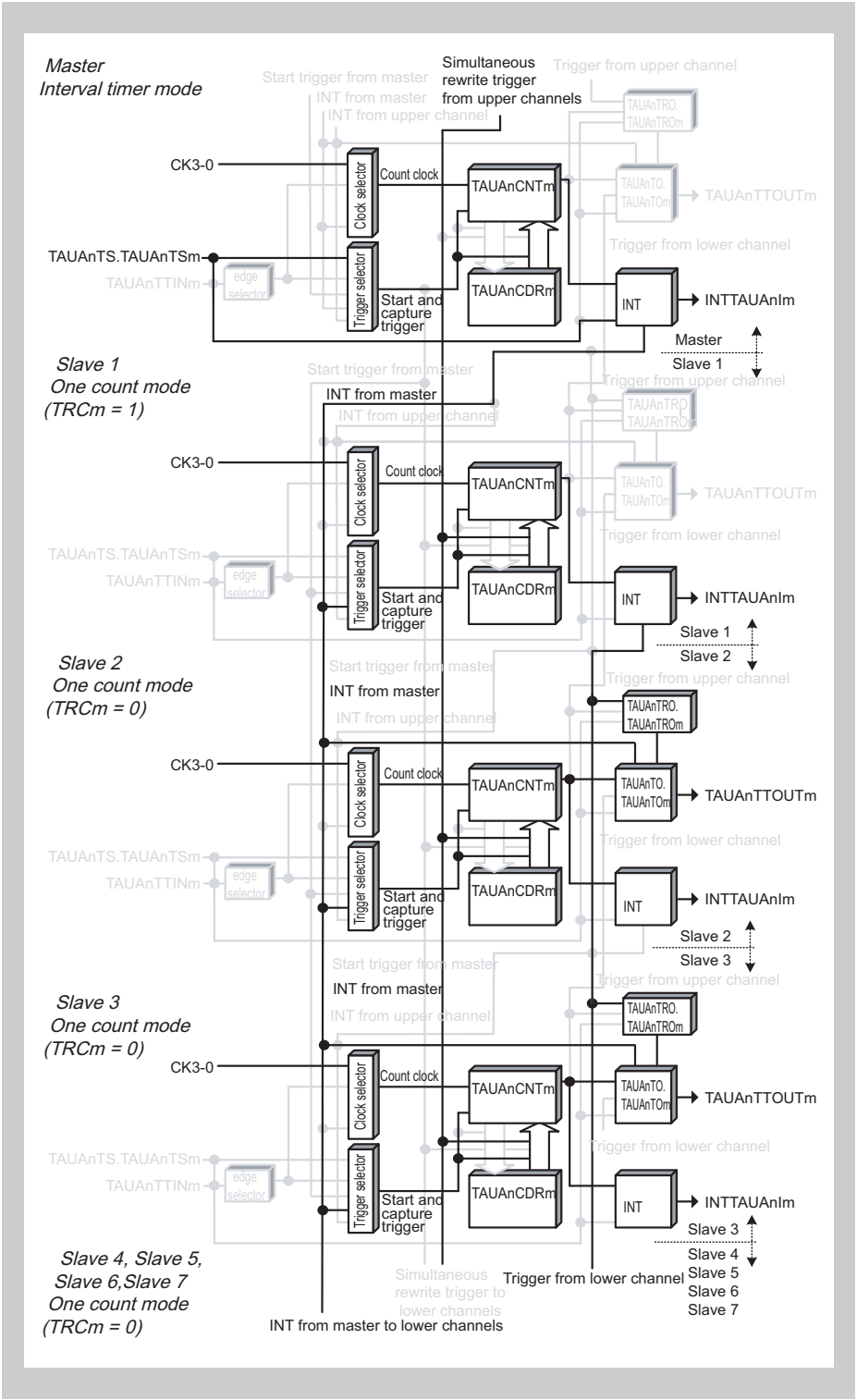


Figure 13-95 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

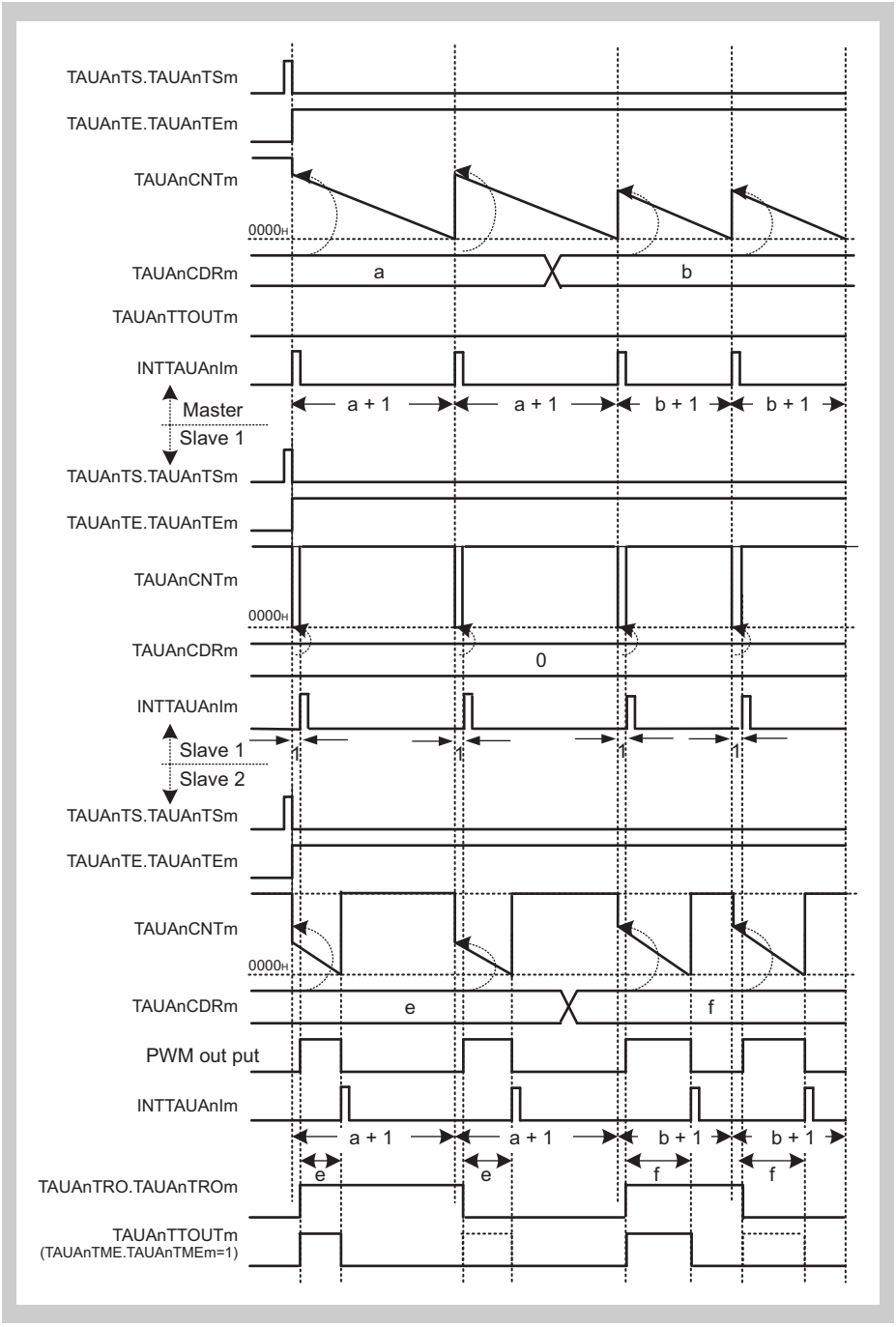


Figure 13-96 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-151 TAUAnCMORm Settings for Master Channels of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	1: INTTAUAnlm is generated at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-152 TAUAnCMURm Settings for Master Channels of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used with other functions or in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-153 Simultaneous Rewrite Settings for Master Channels of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: Simultaneous rewrite trigger signal is generated when master channel counter is started.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note If TAUAnRDS.TAUAnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

(5) Register Settings for Slave Channel 1

(a) TAUAnCMORm for Slave Channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-154 TAUAnCMORm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Slave Channel 1

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-155 TAUAnCMURm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

Caution TAUAnTRC.TAUAnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous Rewrite for Slave Channel 1

The simultaneous rewrite settings of the master and slave channels should be identical.

Table 13-156 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: Simultaneous rewrite trigger signal is generated when master channel counter is started.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Register Settings for Slave Channels 2 to 7

(a) TAUAnCMORm for Slave Channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-157 TAUAnCMORm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of master channel is a start trigger.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Slave Channels 2 to 7

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-158 TAUAnCMURm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Slave Channels 2 to 7

Table 13-159 Control Bit Settings for Slave Channels 2 to 7 in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation. 1: Enables modulation.

(d) Simultaneous Rewrite of Slave Channels 2 to 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-160 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: Simultaneous rewrite trigger signal is generated when master channel counter is started.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(7) Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 13-161 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (4) Register Settings for Master Channels. Slave channel 1: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channel 1. Slave channels 2 to 7: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (6) Register Settings for Slave Channels 2 to 7. Set the value of TAUAnCDRm register of every channel. Set a pulse cycle with TAUAnCDRm of master channel, 0000_H in TAUAnCDRm of slave channel 1, and duty width with TAUAnCDRm of slave channels 2 to 7. Set TAUAnTRC.TAUAnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 13-161 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (2/2)

	Operation	TAUAn Status
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. (Set only slave channels 2 to 7 at a channel restart time.) TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, and TAUAnTMEm.TAUAnTMEm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUAnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCDRm value is reloaded into TAUAnCNTm to continue counting down. • PWM output signals of slave channels 2 to 7 are set/reset. • TAUAnCDRm value of slave channel 1 is reloaded into TAUAnCNTm to perform counting down. • TAUAnCDRm value of slave channels 2 to 7 is reloaded into TAUAnCNTm to perform counting down. • When the counter of slave channel 1 or slave channels 2 to 7 reaches 0000_H: - INTTAUAnIm is generated. • When the counter of slave channels 2 to 7 reaches 0000_H: - PWM output signals of slave channels 2 to 7 are set/reset. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUAnTRO.TAUAnTROm) and modulation output bit (TAUAnTME.TAUAnTMEm) of a pair of slave channels.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(8) Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

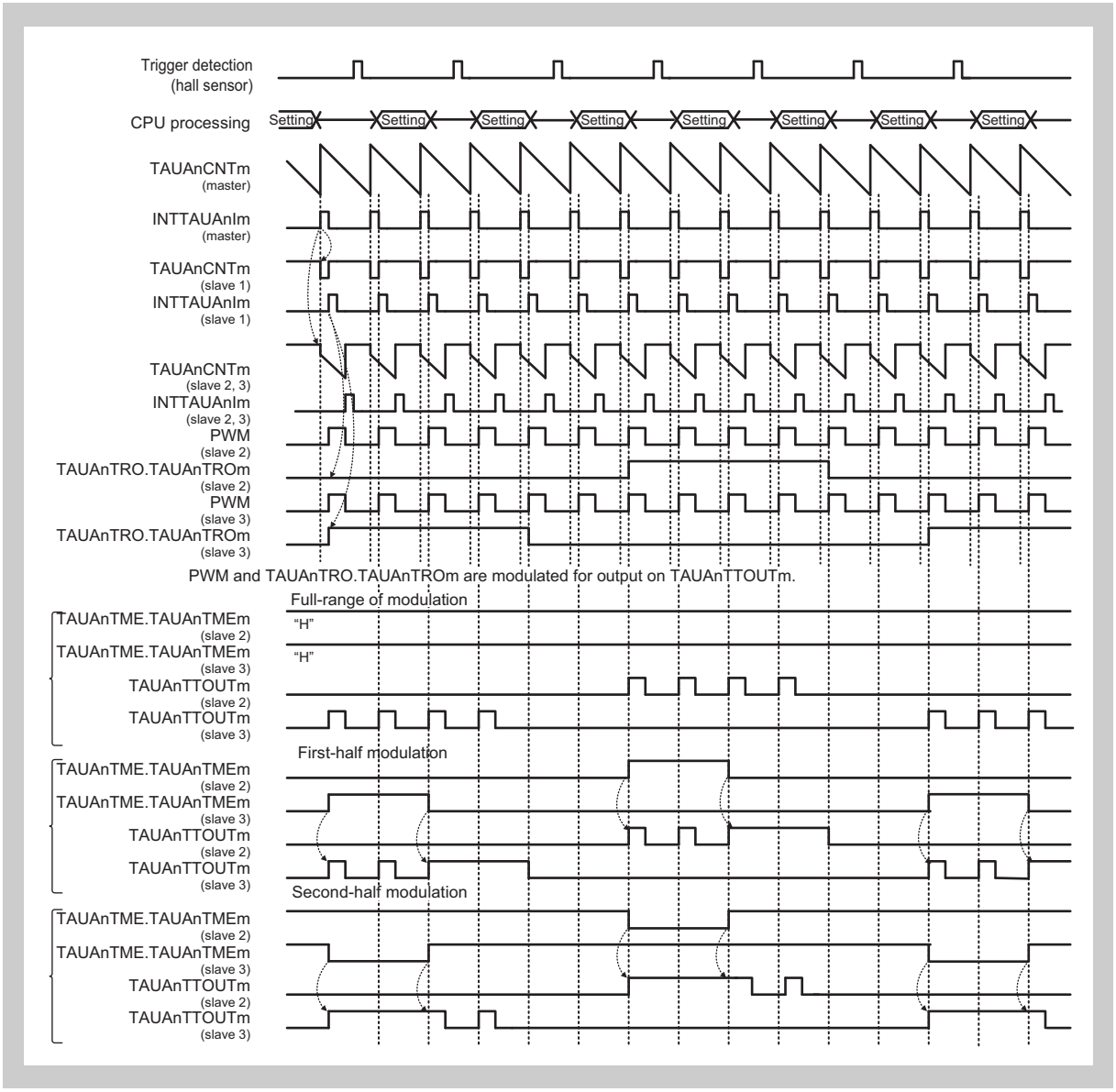


Figure 13-97 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, fist-half modulation, and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm, and TAUAnTRO.TAUAnTROm can be changed.

TAUAnTME.TAUAnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUAnTTOUTm.

TAUAnTRO.TAUAnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

13.14.10 Non-Complementary Modulation Output Function Type 2

(1) Overview

Summary

This function outputs a PWM signal, a high-level signal, or low-level signal fromTAUAnTTOUTm depending on the real-time output bit value (TAUAnTRO.TAUAnTROm) and the modulation output enable bit value (TAUAnTME.TAUAnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operating mode for the master channel should be set to interval timer mode. (See table 13-163, TAUAnCMORm Settings for Master Channels of Non-Complementary Modulation Output Function Type 2.)
- The operating mode for slave channel 1 should be set to event count mode. (See table 13-167, TAUAnCMORm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2.)
- The operating mode for slave channels 2 to 7 should be set to up/down count mode. (See table 13-170, TAUAnCMORm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2.)
- This function does not use TAUAnTTOUTm on the master channel.
- This function does not use TAUAnTTOUTm of slave channel 1 but TAUAnTRC.TAUAnTRCm should be set to 1. (See section 13.8, Channel Output Modes.*)*
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output. (See section 13.8, Channel Output Modes.*)*

Functional description

The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The value of data register (TAUAnCDRm) is loaded into the counter (TAUAnCNTm).

- Master channels:

  The counter of master channel starts to count down. When the counter reaches 0000_H, INTTAUAnIm occurs.

- Slave channel 1:

  When slave channel 1 detects an interrupt from the master channel, the count decreases by 1. When the counter reaches to 0000_H, slave channel 1 waits from the next interrupt from the master channel. And the TAUAnCDRm value is reloaded into TAUAnCNTm (slave 1) and INTTAUAnIm is generated.

  Since slave channel 1 is set as a real-time output trigger channel (TAUAnTRC.TAUAnTRCm = 1), the real-time output bit (TAUAnTRO.TAUAnTROm) of the channel which detects an interrupt on the corresponding channel changes when an interrupt occurs on slave 1 channel.

- Slave channel 2:

  Once detecting an interrupt from the master channel, TAUAnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUAnCDRm value is reloaded and then the counter starts to count down.

If TAUAnCNTm = 0001_H, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operates like slave channel 2.

A signal that is output from TAUAnTTOUTm depends on a real-time output bit value (TAUAnTRO.TAUAnTROm) and a modulation output bit value (TAUAnTME.TAUAnTMEm) of the slave channel, as described in table 13-162, TAUAnTTOUTm Output of a Pair of Slave Channels in Non-Complementary Modulation Output Function Type 2.

This function cannot make a forced restart. The counter operation can be stopped by setting the TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. Both TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their own values. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions

- If TAUAnTME.TAUAnTMEm of slave channel is set to 1:
  - If TAUAnTRO.TAUAnTROm of the channel is 1, TAUAnTTOUTm outputs the corresponding PWM of the channel.
  - If TAUAnTRO.TAUAnTROm of the channel is 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTME.TAUAnTMEm of the slave channel is set to 0:
  - If TAUAnTRO.TAUAnTROm of the channel is set to 1, TAUAnTTOUTm outputs a high-level signal.
  - If TAUAnTRO.TAUAnTROm of the channel is set to 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTOL.TAUAnTOLm is set to 1, high-level and low-level signals output from TAUAnTTOUTm are inverted, having no effect on PWM signals. TAUAnTOL.TAUAnTOLm can only be initialized (unchangeable during operation).

Table 13-162 TAUAnTTOUTm Output of a Pair of Slave Channels in Non-Complementary Modulation Output Function Type 2

TAUAnTME. TAUAnTMEm	TAUAnTRO. TAUAnTROm	TAUAnTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWMm

- Simultaneous rewrite can be used with this function. See section 13.7, Simultaneous Rewrite.
- If TAUAnTOL.TAUAnTOLm of slave channels 2 to 7 is set to 0, TAUAnTO.TAUAnTOm is set low before setting TAUAnTE.TAUAnTEm = 0.
- If TAUAnTOL.TAUAnTOLm of slave channels 2 to 7 is set to 1, TAUAnTO.TAUAnTOm is set high before setting TAUAnTE.TAUAnTEm = 0.

(2) Equations

Slave channels 2 to 7:

PWM output cycle = [TAUAnCDRm (master) + 1] × count clock cycle

PWM output duty time
= [TAUAnCDRm (master) + 1 - TAUAnCDRm (slave)] × 2 × count clock

(3) Block diagram and general timing diagram

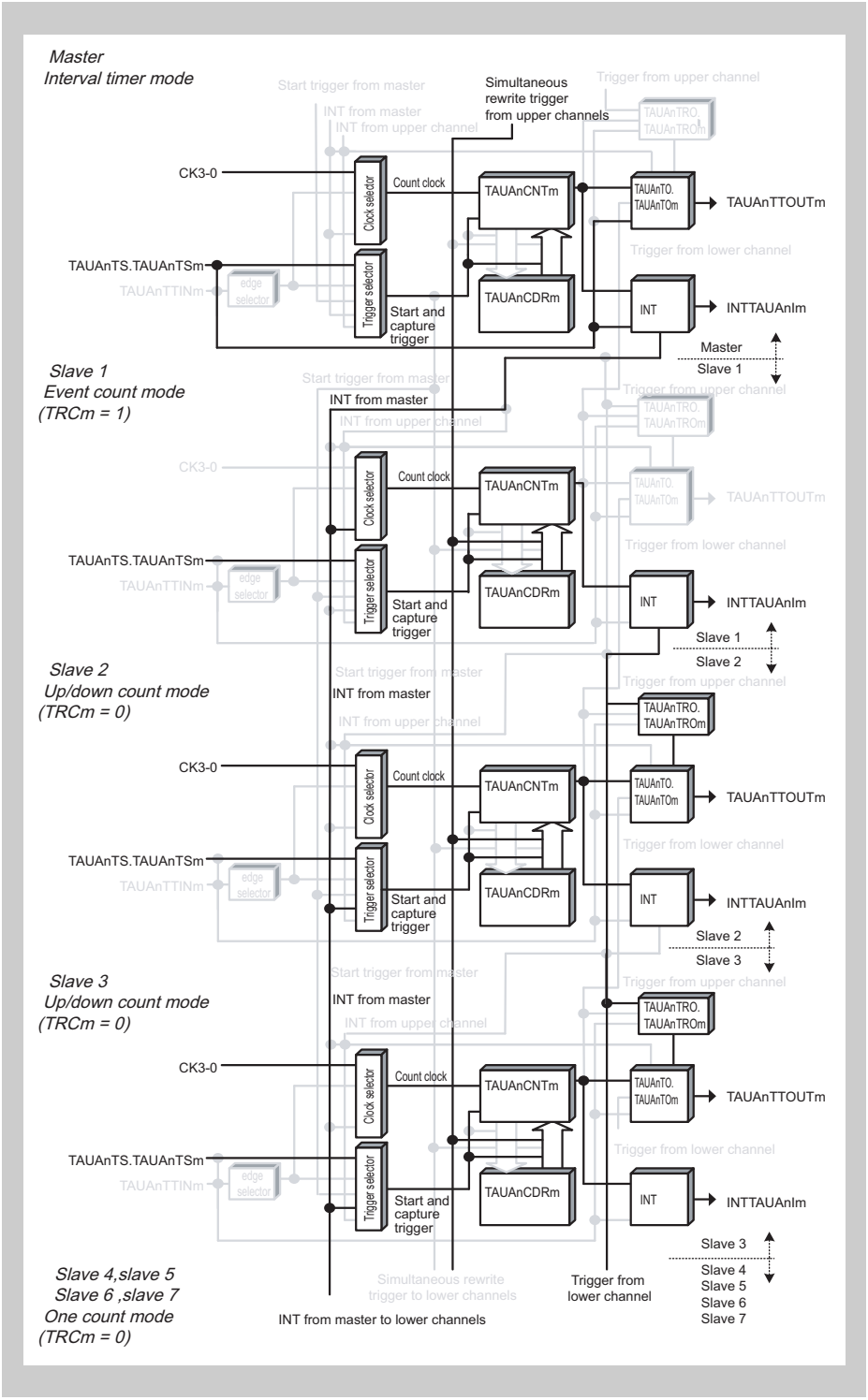


Figure 13-98 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channels: INTTAUAnIm is not generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

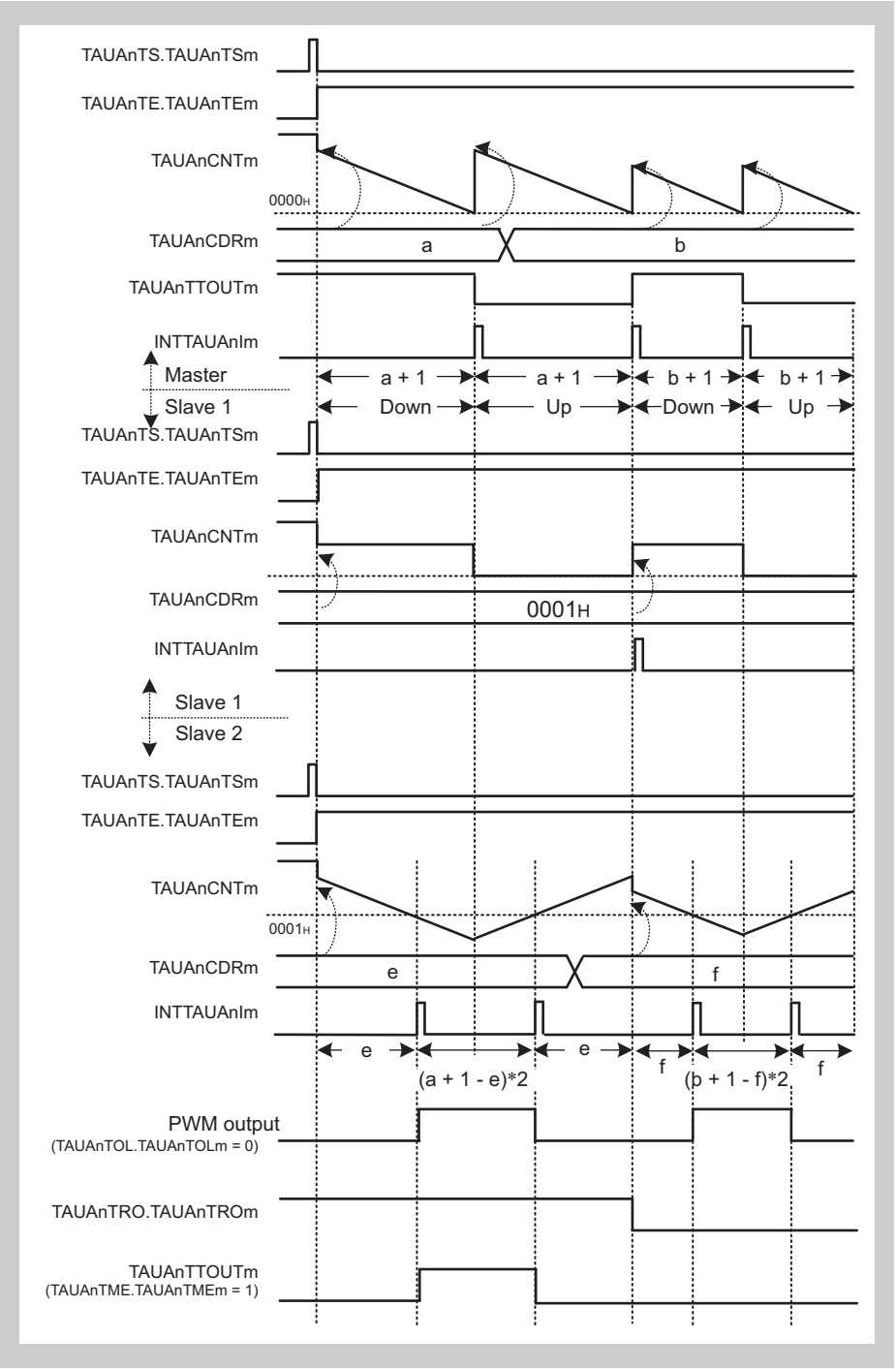


Figure 13-99 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-163 TAUAnCMORm Settings for Master Channels of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUAnIm is generated at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-164 TAUAnCMURm Settings for Master Channels of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

Table 13-165 Control Bit Settings for Master Channels in Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (toggle mode with TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low.
TAUAnTRC.TAUAnTRCm	0: The next upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-166 Simultaneous Rewrite Settings for Master Channels of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channels for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: When disabling simultaneous rewrite (TAUAnRDE.TAUAnRDEm = 0), set this bit to 0. When TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note If TAUAnRDS.TAUAnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

(5) Register Settings for Slave Channel 1

(a) TAUAnCMORm for Slave Channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-167 TAUAnCMORm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	11: INTTAUAnIm of master channel is used as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	000: Triggers the counter by software. 011: Triggers simultaneous rewrite.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0011: Event count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation or a restart time.

(b) TAUAnCMURm for Slave Channel 1

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-168 TAUAnCMURm Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

Caution TAUAnTRC.TAUAnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous Rewrite for Slave Channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-169 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Register Settings for Slave Channels 2 to 7

(a) TAUAnCMORm for Slave Channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-170 TAUAnCMORm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	111: Up/down output trigger signal of master channel
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1001: Up/down count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Slave Channels 2 to 7

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-171 TAUAnCMURm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Output mode for slave channels 2 to 7

Table 13-172 Control Bit Settings for Slave Channels 2 to 7 in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disablings dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: The next upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation. 1: Enables modulation.

(d) Simultaneous rewrite for slave channels 2 to 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-173 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(7) Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 13-174 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (1/2)

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (4) Register Settings for Master Channels. Slave channel 1: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channel 1. Slave channels 2 to 7: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (6) Register Settings for Slave Channels 2 to 7. Set the value of TAUAnCDRm register of every channel. Set pulse cycle in TAUAnCDRm of master channel, and in TAUAnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates an input signal. Set duty width in TAUAnCDRm of slave channels 2 to 7. Set TAUAnTRC.TAUAnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 13-174 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (2/2)

	Operation	TAUAn Status
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. (Set only slave channels 2 to 7 at a channel restart time.) TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm of master and slave channels is set to 1 and the counter starts to count down.
During Operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, and TAUAnTME.TAUAnTMEm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm are readable at any time. TAUAnRDT.TAUAnRDTm is changeable during operation.	TAUAnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUAnCNTm to perform count down. The TAUAnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCDRm value is reloaded into TAUAnCNTm to continue counting down. • The TAUAnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. • TAUAnCNTm of slave channels 2 to 7 performs counting in opposite direction. • When the counter of slave channel 1 reaches 0000_H, it waits for a next interrupt from the master channel. When an interrupt is detected: - INTTAUAnIm is generated. • When the counter of slave channels 2 to 7 reaches 0001_H: - INTTAUAnIm is generated. - PWM output signals of slave channels 2 to 7 are set/reset. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUAnTRO.TAUAnTROm) and modulation output bit (TAUAnTME.TAUAnTMEm) of a pair of slave channels.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(8) Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

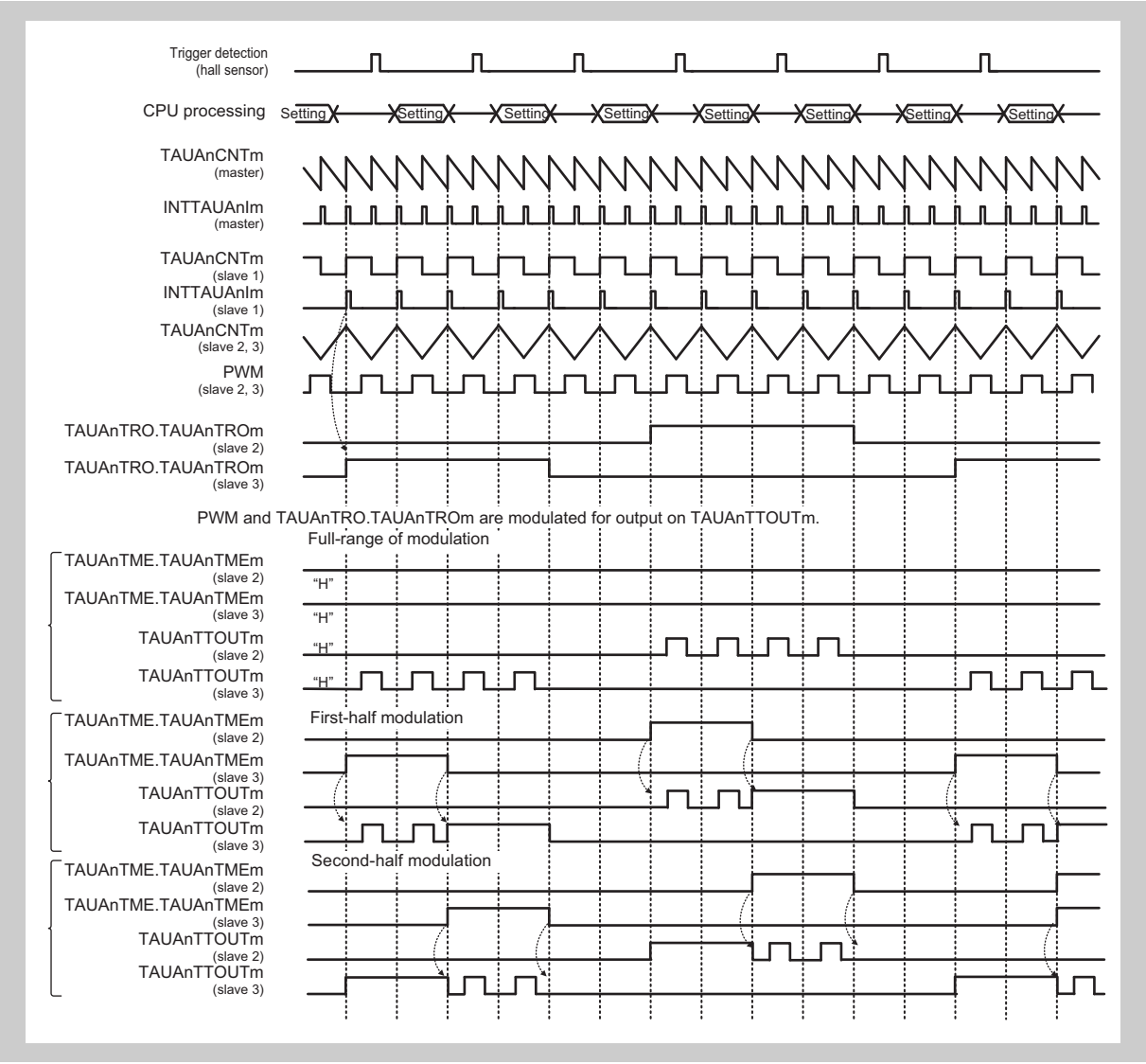


Figure 13-100 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, fist-half modulation, and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm, and TAUAnTRO.TAUAnTROm can be changed.

TAUAnTME.TAUAnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUAnTRO.TAUAnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

13.14.11 Complementary Modulation Output Function

(1) Overview

Summary

This function outputs a PWM signal, a high-level signal, or low-level signal from TAUAnTTOUTm depending on the real-time output bit value (TAUAnTRO.TAUAnTROm) and the modulation output enable bit value (TAUAnTME.TAUAnTMEm) of a pair of slave channels. Three pairs o channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operating mode for the master channel should be set to interval timer mode. (See table 13-177, TAUAnCMORm Settings for Master Channels of Complementary Modulation Output Function.*)*
- The operating mode for slave channel 1 should be set to event count mode. (See table 13-181, TAUAnCMORm Settings for Slave Channel 1 of Complementary Modulation Output Function.)
- The operating mode for slave channels 2, 4 and 6 should be set to up/down count mode. (See table 13-184, TAUAnCMORm Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function.*)*
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode. (See table 13-188, TAUAnCMORm Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function.*)*
- The output mode for master channels should be set to independent channel output mode 1. (See section 13.8, Channel Output Modes.*)*
- This function does not use TAUAnTTOUTm of slave channel 1 but TAUAnTRC.TAUAnTRCm should be set to 1. (See section 13.8, Channel Output Modes.*)*
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output. (See section 13.8, Channel Output Modes.*)*

Functional description

- Master channels:

  The counter of master channel is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The value of data register (TAUAnCDRm) of master channel is loaded into the counter (TAUAnCNTm) and the counter starts to count down from this value.

  When the counter of master channel reaches 0000_H, INTTAUAnIm occurs.

  This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.

- Slave channel 1:

  When the counter reaches 0000_H, slave channel 1 waits for the next interrupt from the master channel. And the TAUAnCDRm value is reloaded into TAUAnCNTm (slave 1) and INTTAUAnIm is generated.

  Slave channel 1 is set as a real-time output trigger channel (TAUAnTRC.TAUAnTRCm = 1). The value of real-time output bit (TAUAnTRO.TAUAnTROm) of each channel is applied to the channel that detects the occurrence of an interrupt on slave channel 1. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.

- Slave channel 2:

  When the slave channel 2 counter reaches 0001_H, the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000_H, an interrupt occurs.

- Slave channels 2 and 3:

  The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.

- Slave channels 4 to 7:

  Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

A signal that is output from TAUAnTTOUTm depends on a real-time output bit value (TAUAnTRO.TROm), a modulation output bit value (TAUAnTME.TAUAnTMEm) an output level bit value (TAUAnTDL.TAUAnTDLm) of the slave channel, as described in table 13-150, TAUAnTTOUTm Output from One Pair of Slave Channels of Non-Complementary Modulation Output Function Type 1.

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit). It is necessary to postpone a transition to a high level by appending a delay to the channel to be switched to a high level in order to prevent both channels from setting high at the moment one channel is switched to a low level and the other to a high level. To achieve this, dead time set by a channel dead time output level register (TAUAnTDL.TAUAnTDLm) value is appended to a normal or reverse phase PWM signal.

The counter can be stopped by setting the TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. Setting TAUAnTS.TAUAnTSm to 1 restarts counting.

Conditions

- If TAUAnTME.TAUAnTMEm of a pair of channels is set to 1:
  - If TAUAnTRO.TAUAnTROm of one channel is set to 1, TAUAnTTOUTm outputs the corresponding PWM of the channel.
  - If TAUAnTRO.TAUAnTROm of one channel is set to 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTME.TAUAnTMEm of a pair of channels is set to 0:
  - If TAUAnTRO.TAUAnTROm of one channel is set to 1, TAUAnTTOUTm outputs a high-level signal.
  - If TAUAnTRO.TAUAnTROm of one channel is set to 0, TAUAnTTOUTm outputs a low-level signal.
- If TAUAnTOL.TAUAnTOLm is set to 1, high-level and low-level signals output from TAUAnTTOUTm are inverted. No PWM signals are changed depending on the setting of TAUAnTOL.TAUAnTOLm.

Table 13-175 TAUAnTTOUTm Output (TAUAnTOL.TAUAnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

TAUAnTME.TAUAnTME2	TAUAnTME.TAUAnTME3	TAUAnTRO.TAUAnTRO2	TAUAnTRO.TAUAnTRO3	TAUAnTTOUT2 Output	TAUAnTTOUT3 Output
0	0	0	0	Low level	Low level
		0	1	Low level	High level
		1	0	High level	Low level
		1	1	Setting prohibited	Setting prohibited
1	1	0	0	Low level	Low level
		0	1	~PWMm	PWMm
		1	0	PWMm	~PWMm
		1	1	Setting prohibited	Setting prohibited

Note 1. ~PWM indicates an inverted PWM signal. PWM and ~PWM are set by TAUAnTDL.TAUAnTDLm.

Note 2. Any settings not listed above are prohibited.

- If TAUAnTME.TAUAnTMEm is continuously set to 1 while TAUAnTRO.TAUAnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUAnTME.TAUAnTMEm is set to 1 at the first half of the period while TAUAnTRO.TAUAnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUAnTME.TAUAnTMEm is set to 1 at the second half of the period while TAUAnTRO.TAUAnTROm of one of paired channels is set to 1, second-half modulation is applied.
- Whether dead time is added to a normal or reverse phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUAnTDL.TAUAnTDLm bit value.
  - If TAUAnTDL.TAUAnTDLm = 0, dead time is added to a normal phase PWM signal.
  - If TAUAnTDL.TAUAnTDLm = 1, dead time is added to a reverse phase PWM signal.
  - The operation defined by a TAUAnTDL.TAUAnTDLm bit value should be conducted by application software during operation. To modify TAUAnTDL.TAUAnTDLm, rewrite it during the period when TAUAnTRO.TAUAnTROm is 00_B.

Table 13-176 TAUAnTDL.TAUAnTDLm Settings (TAUAnTOL.TAUAnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

TAUAnTME. TAUAnTME2	TAUAnTME. TAUAnTME3	TAUAnTRO. TAUAnTRO2	TAUAnTRO. TAUAnTRO3	TAUAnTDL. TAUAnTDL2	TAUAnTDL. TAUAnTDL3
0	0	0	0	1	1
		0	1	1	0
		1	0	0	1
1	1	0	0	1	1
		0	1	1	0
		1	0	0	1

- The value of TAUAnCDRm of slave channel 1 should be set to 1 so that INTTAUAnIm is generated on slave channel 1 at the peak of a carrier cycle.
- Set TAUAnCMORm.TAUAnMD0 of master channel to 0.
- This function enables simultaneous rewrite. See section 13.7, Simultaneous Rewrite.

(2) Equations

For slave channels 2, 4, and 6:

In case of TAUAnTOL.TAUAnTOLm = 0 and TAUAnCDR (slave 1):

PWM output cycle = 2 × [TAUAnCDRm (master) + 1] × count clock

PWM output duty time = { [TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2)] × 2 - [TAUAnCDRm (slave 3) + 1] } × count clock

(3) Block Diagram and General Timing Diagram

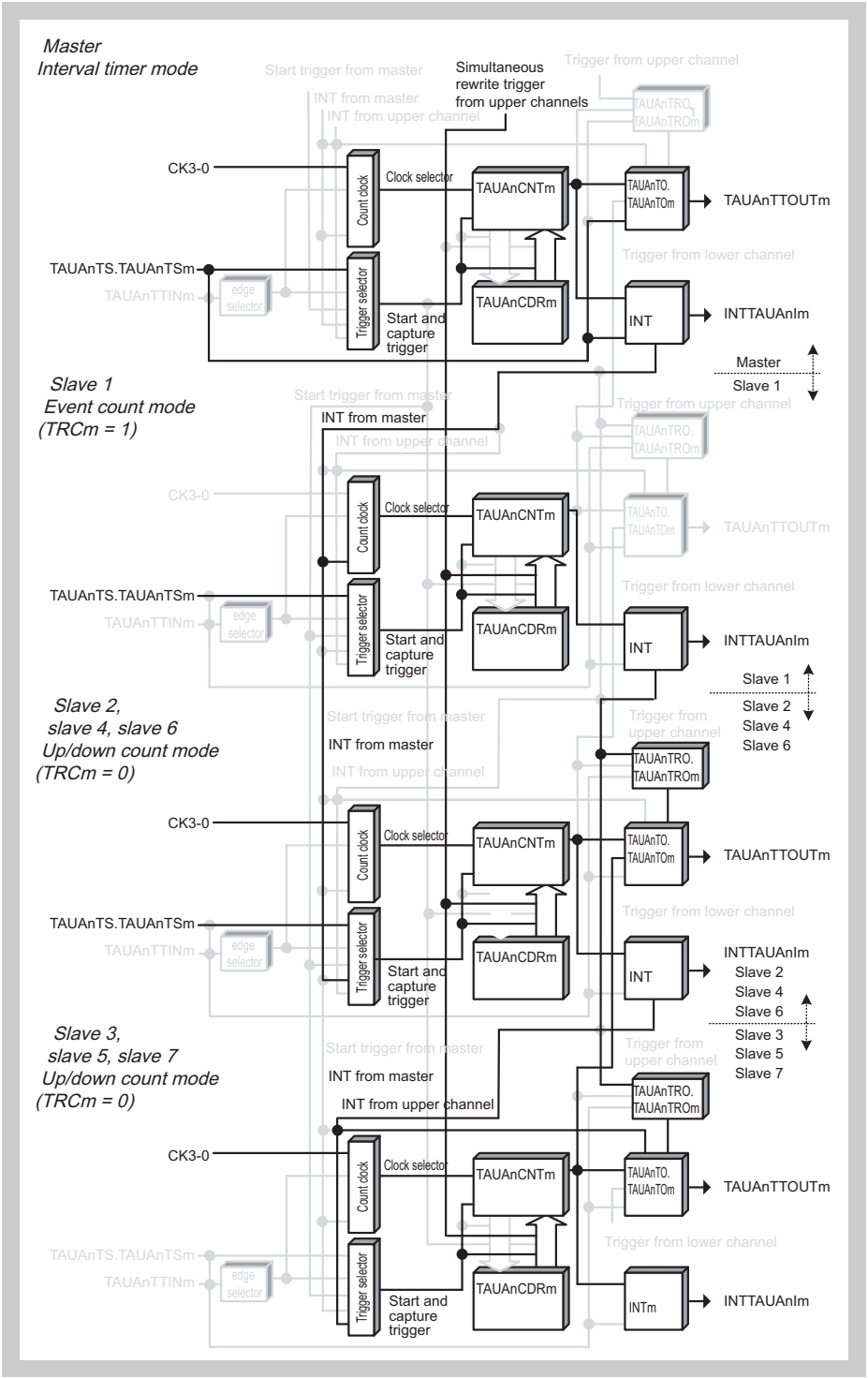


Figure 13-101 Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channels: INTTAUAnIm is not generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

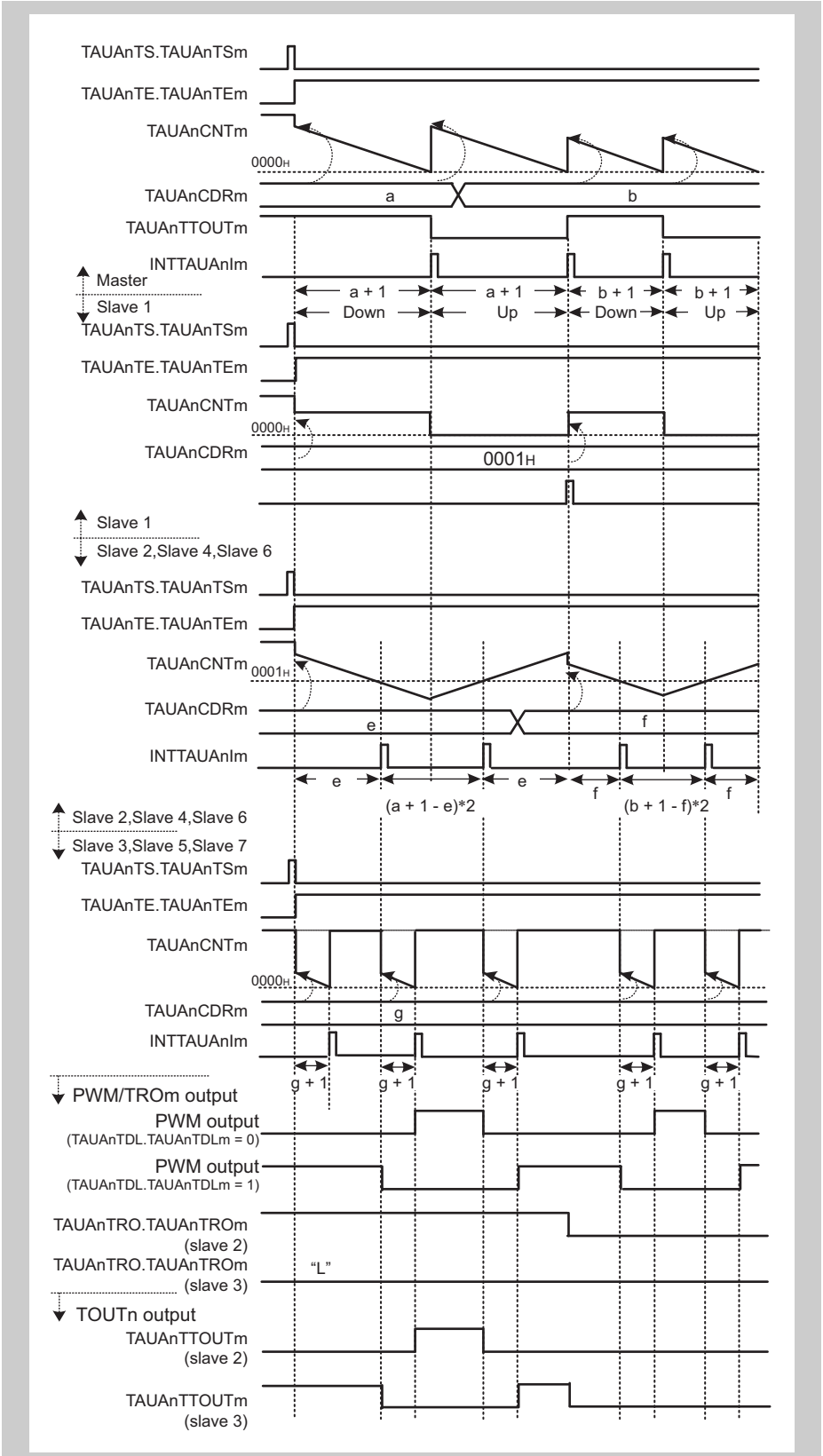


Figure 13-102 General Timing Diagram of Complementary Modulation Output Function

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-177 TAUAnCMORm Settings for Master Channels of Complementary Modulation Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated and TAUAnTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUAnIm is generated and TAUAnTTOUTm is toggled at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-178 TAUAnCMURm Settings for Master Channels of Complementary Modulation Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode

Table 13-179 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operating mode 1 (toggle mode when TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When disabling dead time operation (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: When disabling real-time output (TAUAnTRE.TAUAnTREm = 0), set these bits to 0.
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-180 Simultaneous Rewrite Settings for Master Channels of Complementary Modulation Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

Note If TAUAnRDS.TAUAnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

(5) Register Settings for Slave Channel 1

(a) TAUAnCMORm for Slave Channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-181 TAUAnCMORm Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	11: INTTAUAnIm of master channel is used as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	000: Triggers the counter by software. 011: Triggers simultaneous rewrite.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0011: Event count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Slave Channel 1

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-182 TAUAnCMURm Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode

TAUAnTOE.TAUAnTOEm is set to 0 on slave channel 1 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

Caution TAUAnTRC.TAUAnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(d) Simultaneous Rewrite for Slave Channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-183 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Register Settings for Slave Channels 2, 4, and 6

(a) TAUAnCMORm for Slave Channels 2, 4, and 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-184 TAUAnCMORm Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	111: Up/down output trigger signal of master channel
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	1001: Up/down count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation or at a restart time.

(b) TAUAnCMURm for Slave Channels 2, 4, and 6

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-185 TAUAnCMURm Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Output Mode for Slave Channels 2, 4, and 6

Table 13-186 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables a dead time operation.
TAUAnTDM.TAUAnTDMm	0: Adds dead time upon detection of a duty cycle on an even upper channel.
TAUAnTDL.TAUAnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation. 1: Enables modulation.

Caution Set TAUAnTDL.TAUAnTDLm exclusively from odd channels.

(d) Simultaneous Rewrite for Slave Channels 2, 4, and 6

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-187 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(7) Register Settings for Slave Channels 3, 5, and 7

(a) TAUAnCMORm for Slave Channels 3, 5, and 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-188 TAUAnCMORm Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	110: Dead time trigger
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0100: One-count mode
TAUAnMD0	1: Enables start trigger detection while counting.

(b) TAUAnCMURm for Slave Channels 3, 5, and 7

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-189 TAUAnCMURm Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Output Mode for Slave Channels 3, 5, and 7

Table 13-190 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables indepenent channel output mode.
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operating mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	0: Adds dead time upon detection of a duty cycle on an even upper channel.
TAUAnTDL.TAUAnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUAnTRE.TAUAnTREm	1: Enables real-time output.
TAUAnTRO.TAUAnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUAnTRC.TAUAnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation. 1: Enables modulation.

Caution Set TAUAnTDL.TAUAnTDLm exclusively from even channels.

(d) Simultaneous Rewrite for Slave Channels 3, 5, and 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-191 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channels other than the channel group for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(8) Operating Procedure for Complementary Modulation Output Function

Table 13-192 Operating Procedure for Complementary Modulation Output Function (1/2)

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (4) Register Settings for Master Channels. Slave channel 1: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channel 1. Slave channels 2, 4, and 6: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (6) Register Settings for Slave Channels 2, 4, and 6. Slave channels 3, 5, and 7: Set TAUAnCMORm/ TAUAnCMURm register and channel output mode as described in (7) Register Settings for Slave Channels 3, 5, and 7. Set the value of TAUAnCDRm register of every channel. Set a pulse cycle using TAUAnCDRm of master channel, and an interrupt count ignored using TAUAnCDRm of slave channel 1. Also set a duty width in TAUAnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7. Set TAUAnTRC.TAUAnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 13-192 Operating Procedure for Complementary Modulation Output Function (2/2)

	Operation	TAUAn Status
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. (Set only slave channels 2 to 7 at a channel restart time.) TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm of master and slave channels is set to 1 and the counter starts to count down.
During Operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUAnCNTm perform count down. TAUAnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCDRm value is reloaded into TAUAnCNTm to continue counting down. • TAUAnCNTm value of slave channel 1 decreases by 1 and the counter waits for the next master channel interrupt. • TAUAnCNTm of slave channels 2, 4, and 6 performs counting in the reverse direction. • The counter of slave channel 1 waits for the next interrupt from the master channel when reaching 0000_H. When the interrupt is detected: - TAUAnCDRm value is reloaded into TAUAnCNTm and the counter waits for the next master channel interrupt. - INTTAUAnIm is generated. - TAUAnTRO.TAUAnTROm is changeable. • When the counter of slave channels 2, 4, and 6 reaches 0001_H: - INTTAUAnIm is generated. - PWM output of slave channel m is reset. - TAUAnCDRm value of slave channels 3, 5, and 7 is loaded into TAUAnCNTm to perform count down. • When the counter of slave channels 3, 5, and 7 reaches 0001_H: - INTTAUAnIm is generated. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bit (TAUAnTRO.TAUAnTROm), modulation output bit (TAUAnTME.TAUAnTMEm), and output level bit (TAUAnTDL.TAUAnTDLm) of a pair of slave channels.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(9) Specific Timing Diagrams

The following settings apply to the timing diagram.

- Slave channels 2 to 7: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

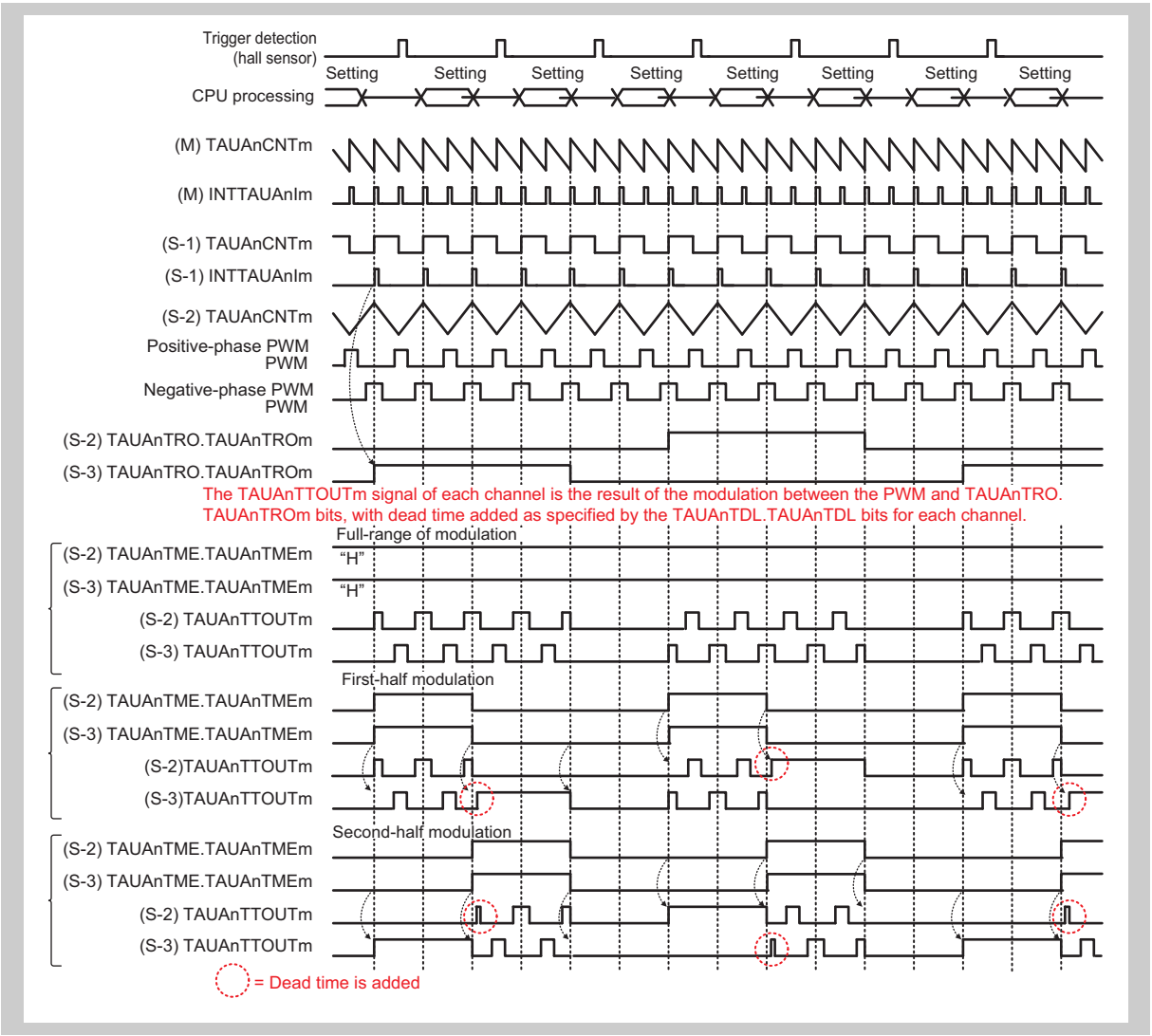


Figure 13-103 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, fist-half modulation and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUAnTRO.TAUAnTROm bit value are output from slave channels 2 and 3. Therefore, a type (normal/reverse phase) of PWM signal to be output from each channel depends on this bit value.

TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm bit values are specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

Note Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

The "Setting"symbol indicates a time period when the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm, TAUAnTRO.TAUAnTROm, and TAUAnTDL.TAUAnTDLm can be changed.

13.14.12 Interrupt Sampling Function

(1) Overview

Summary

This function divides the number of interrupts of the master channel by a value specified using a slave channel.

The interrupt sampling function is a sub function of the following functions:

- PWM output function

  (Section 13.14.1, PWM Output Function*)*

- Triangle PWM output function

  (Section 13.14.6, Triangle PWM Output Function*)*

- Triangle PWM output function with dead time

  (Section 13.14.7, Triangle PWM Output Function with Dead Time*)*

Prerequisites

- Two channels
- The operating mode for master channel should be set to interval timer mode. (See table 13-193, TAUAnCMORm Settings for Master Channels of Interrupt Sampling Function*.)*
- The operating mode for slave channels should be set to event count mode. (See table 13-196, TAUAnCMORm Settings for Slave Channels of Interrupt Sampling Function*.)*
- TAUAnTTOUTm is not used for slave channels of this function.

Functional description

The counter (both master and slave channels) is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This sets TAUAnTE.TAUAnTEm, enabling count operation. The current value of data register (TAUAnCDRm) of master and slave counters is loaded into the counter (TAUAnCNTm).

- Master channels:

  When the counter of master channel reaches 0000_H, INTTAUAnIm is generated and TAUAnCDRm value is loaded in TAUAnCNTm.

- Slave channels:

  Every time the master channel generates an INTTAUAnIm, the counter of the slave channel decrements by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUAnCNTm (slave) to load the value of TAUAnCDRm, and an INTTAUAnIm is generated.

This function cannot make a forced restart. The counter can be stopped by setting TAUAnTT.TAUAnTTm of master and slave channels to 1. This sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values.

Conditions

This function can conduct simultaneous rewrite. See table 13.7, Simultaneous Rewrite*.*

(2) Equations

Interrupt division operator = TAUAnCDRm (slave channel)

– One INTTAUAnIm is generated for the INTTAUAnIm count of the master channel defined by TAUAnCDRm (slave channel) + 1.

(3) Block Diagram and General Timing Diagram

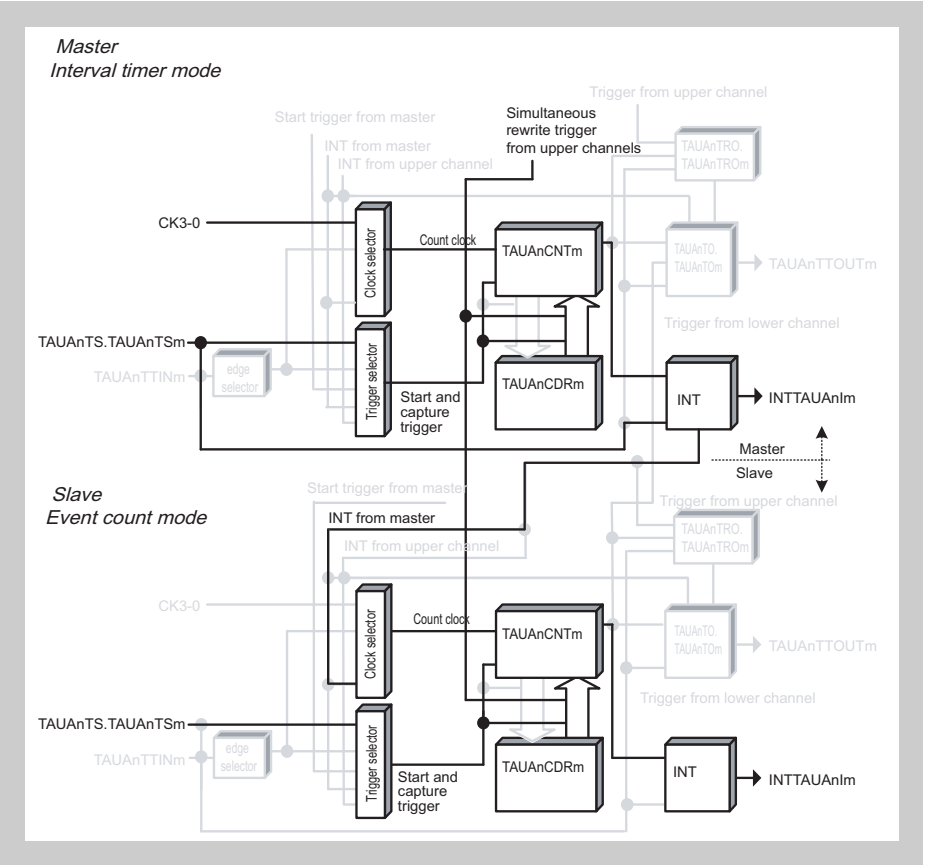


Figure 13-104 Block Diagram of Interrupt Sampling Function

The following settings apply to the general timing diagram.

Master channels:

- INTTAUAnIm is generated at the beginning of operation. (TAUAnCMORm.TAUAnMD0 = 1)

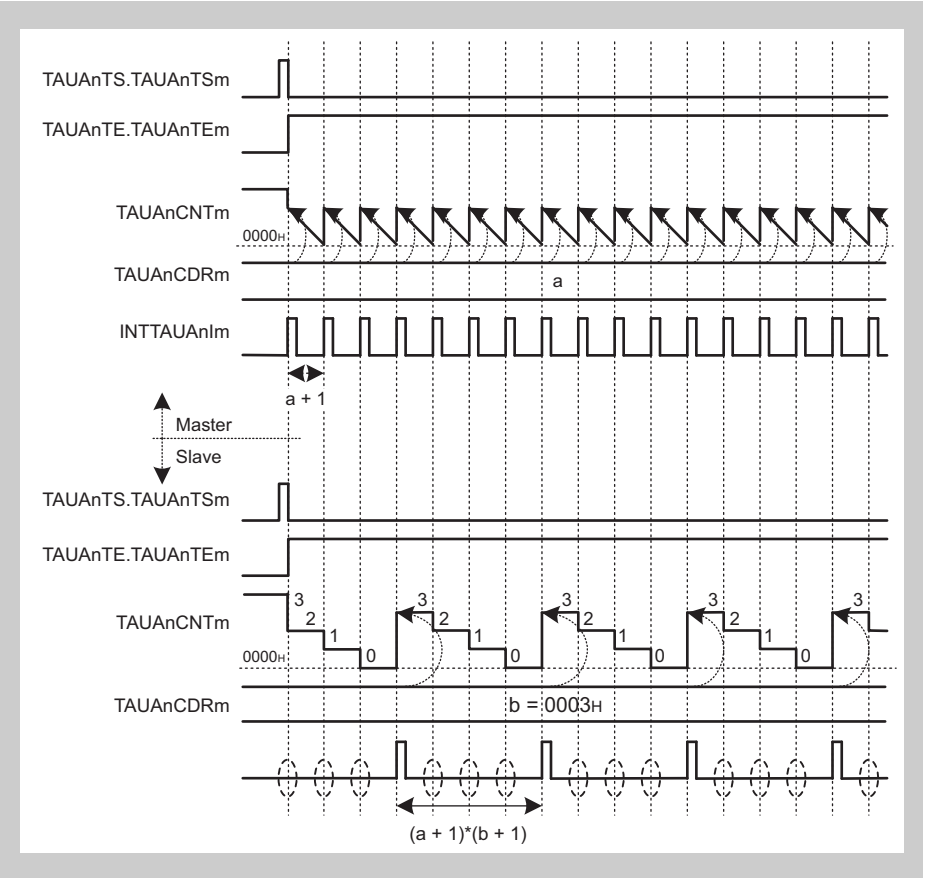


Figure 13-105 General Timing Diagram of Interrupt Sampling Function

A simultaneous rewrite trigger signal is generated when the master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.

(4) Register Settings for Master Channels

(a) TAUAnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-193 TAUAnCMORm Settings for Master Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	00: Uses a sampling clock as a count clock.
TAUAnMAS	1: Master channel
TAUAnSTS[2:0]	000: Triggers the counter by software.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation. 1: INTTAUAnIm is generated at the beginning of operation.

(b) TAUAnCMURm for Master Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-194 TAUAnCMURm Settings for Master Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Channel Output Mode for Master Channels

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Master Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-195 Simultaneous Rewrite Settings for Master Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: A simultaneous rewrite trigger signal is generated when the master channel counter starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(5) Register Settings for Slave Channels

(a) TAUAnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:1]				TAUAnMD0

Table 13-196 TAUAnCMORm Settings for Slave Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUAnCKS[1:0] bits of master and slave channels should have the same value.
TAUAnCCS[1:0]	11: Uses INTTAUAnIm of master channel as a count clock.
TAUAnMAS	0: Slave channel
TAUAnSTS[2:0]	000: Triggers the counter by software. 011: Triggers simultaneous rewrite.
TAUAnCOS[1:0]	00: Unused. Set to 00.
TAUAnMD[4:1]	0011: Event count mode
TAUAnMD0	0: INTTAUAnIm is not generated at the beginning of operation.

(b) TAUAnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUAnTIS[1:0]	

Table 13-197 TAUAnCMURm Settings for Slave Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnTIS[1:0]	00: Unused. Set to 00.

(c) Output Mode for Slave Channels

TAUAnTOE.TAUAnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite for Slave Channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 13-198 Simultaneous Rewrite Settings for Slave Channels of Interrupt Sampling Function

Bit Name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite.
TAUAnRDS.TAUAnRDSm	0: Monitors master channel for simultaneous rewrite triggers.
TAUAnRDM.TAUAnRDMm	0: A simultaneous rewrite trigger signal is generated when the master channel counter starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for INTTAUAnIm signals used as simultaneous rewrite triggers. If TAUAnRDS.TAUAnRDSm = 0, master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit.

(6) Operating Procedure for Interrupt Sampling Function

Table 13-199 Operating Procedure for Interrupt Sampling Function

	Operation	TAUAn Status
Initial Channel Setting	Master channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (4) Register Settings for Master Channels. Slave channels: Set TAUAnCMORm/ TAUAnCMURm register and the channel output mode as described in (5) Register Settings for Slave Channels. Set the value of TAUAnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUAnTS.TAUAnTSm of master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm of master and slave channels is set to 1 and the counters of master and slave channels start counting down. INTTAUAnIm is generated on the master channel.
During Operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of master channel loads TAUAnCDRm value and the counter starts to count down. When the counter of master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) loads TAUAnCDRm value and continues count operation. • TAUAnCNTm of slave channels counts down each time INTTAUAnIm of master channel is detected. When TAUAnCNTm of slave channels reaches 0000_H: • INTTAUAnIm (slave) is generated.
Stop Operation	Set TAUAnTT.TAUAnTTm of master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, which is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific Timing Diagrams

(a) Interrupt Count (Master) = Interrupt Count (Slave)

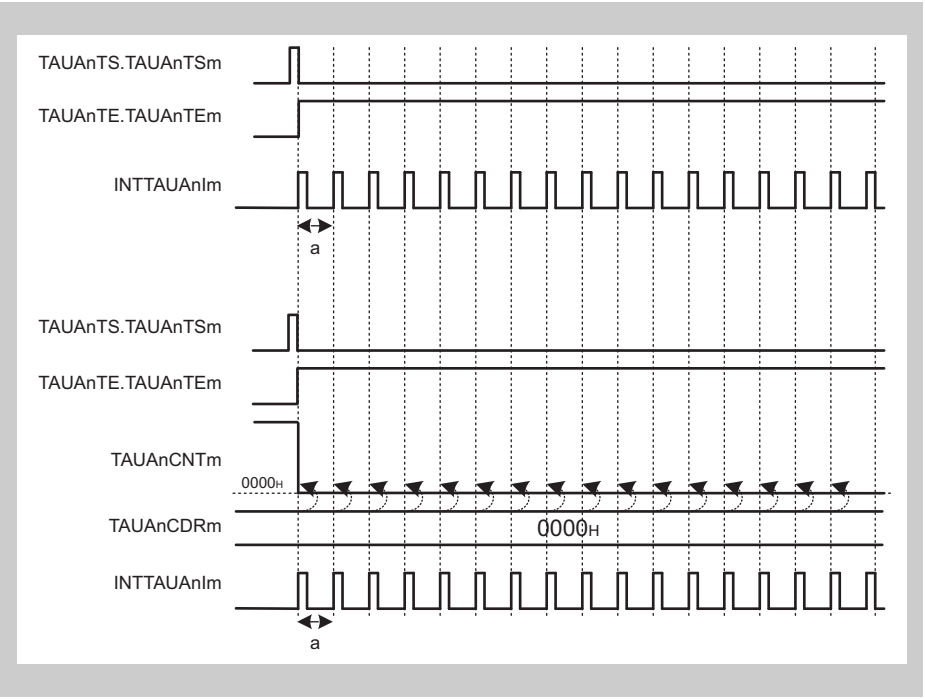


Figure 13-106 TAUAnCDRm (Slave) = 0000_H

- If TAUAnCDRm = 0000_H, TAUAnCDRm value of slave channels is loaded into TAUAnCNTm each time INTTAUAnIm of master channel is detected. In other words, TAUAnCNTm is always 0000_H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

13.15 Registers

This section contains a description of all the registers of the 16-bit timer array unit A.

13.15.1 Overview of TAUAn Registers

TAUAn is controlled and operated by the registers listed below.

Table 13-200 Overview of TAUAn Registers (1/2)

Register Name	Abbreviation	Address
TAUAn prescaler registers		
TAUAn prescaler clock select register	TAUAnTPS	<TAUAn_base0> + 240_H
TAUAn prescaler baud rate setting register	TAUAnBRS	<TAUAn_base0> + 244_H
TAUAn control registers		
TAUAn channel data register m	TAUAnCDRm	<TAUAn_base1> + m × 4_H
TAUAn channel counter register m	TAUAnCNTm	<TAUAn_base1> + 80_H + m × 4_H
TAUAn channel mode OS register m	TAUAnCMORm	<TAUAn_base0> + 200_H + m × 4_H
TAUAn channel mode user register m	TAUAnCMURm	<TAUAn_base1> + $C0_H$ + m × 4_H
TAUAn channel status register m	TAUAnCSRm	<TAUAn_base1> + 140_H + m × 4_H
TAUAn channel status clear trigger register m	TAUAnCSCm	<TAUAn_base1> + 180_H + m × 4_H
TAUAn channel start trigger register	TAUAnTS	<TAUAn_base1> + $1C4_H$
TAUAn channel enable status register	TAUAnTE	<TAUAn_base1> + $1C0_H$
TAUAn channel stop trigger register	TAUAnTT	<TAUAn_base1> + $1C8_H$
TAUAn output registers		
TAUAn channel output enable register	TAUAnTOE	<TAUAn_base1> + $5C_H$
TAUAn channel output register	TAUAnTO	<TAUAn_base1> + 58_H
TAUAn channel output mode register	TAUAnTOM	<TAUAn_base0> + 248_H
TAUAn channel output configuration register	TAUAnTOC	<TAUAn_base0> + $24C_H$
TAUAn channel output active level register	TAUAnTOL	<TAUAn_base1> + 40_H
TAUAn channel dead time output enable register	TAUAnTDE	<TAUAn_base0> + 250_H
TAUAn channel dead time output mode register	TAUAnTDM	<TAUAn_base0> + 254_H
TAUAn channel dead time output level register	TAUAnTDL	<TAUAn_base1> + 54_H
TAUAn channel real-time output register	TAUAnTRO	<TAUAn_base1> + $4C_H$
TAUAn channel real-time output enable register	TAUAnTRE	<TAUAn_base0> + 258_H
TAUAn channel real-time output control register	TAUAnTRC	<TAUAn_base0> + $25C_H$
TAUAn channel modulation output enable register	TAUAnTME	<TAUAn_base1> + 50_H
TAUAn simultaneous rewrite register		
TAUAn channel reload data enable register	TAUAnRDE	<TAUAn_base0> + 260_H
TAUAn channel reload data mode register	TAUAnRDM	<TAUAn_base0> + 264_H
TAUAn channel reload data control channel select register	TAUAnRDS	<TAUAn_base0> + 268_H
TAUAn channel reload data control register	TAUAnRDC	<TAUAn_base0> + $26C_H$
TAUAn channel reload data trigger register	TAUAnRDT	<TAUAn_base1> + 44_H

Table 13-200 Overview of TAUAn Registers (2/2)

Register Name	Abbreviation	Address
TAUAn channel reload status register	TAUAnRSF	<TAUAn_base1> + 48_H
TAUAn DMA window registers		
TAUAn DMA window address setting register 0	TAUAnDAS0	<TAUAn_base0> + 270_H
TAUAn DMA window address setting register 1	TAUAnDAS1	<TAUAn_base0> + 274_H
TAUAn DMA window address setting register 2	TAUAnDAS2	<TAUAn_base0> + 278_H
TAUAn DMA window address setting register 3	TAUAnDAS3	<TAUAn_base0> + $27C_H$
TAUAn DMA window address setting register 4	TAUAnDAS4	<TAUAn_base0> + 280_H
TAUAn DMA window address setting register 5	TAUAnDAS5	<TAUAn_base0> + 284_H
TAUAn DMA window address setting register 6	TAUAnDAS6	<TAUAn_base0> + 288_H
TAUAn DMA window address setting register 7	TAUAnDAS7	<TAUAn_base0> + $28C_H$
TAUAn DMA window register 0	TAUAnDWR0	<TAUAn_base1> + 100_H
TAUAn DMA window register 1	TAUAnDWR1	<TAUAn_base1> + 104_H
TAUAn DMA window register 2	TAUAnDWR2	<TAUAn_base1> + 108_H
TAUAn DMA window register 3	TAUAnDWR3	<TAUAn_base1> + $10C_H$
TAUAn DMA window register 4	TAUAnDWR4	<TAUAn_base1> + 110_H
TAUAn DMA window register 5	TAUAnDWR5	<TAUAn_base1> + 114_H
TAUAn DMA window register 6	TAUAnDWR6	<TAUAn_base1> + 118_H
TAUAn DMA window register 7	TAUAnDWR7	<TAUAn_base1> + $11C_H$
TAUAn DMA window register 8	TAUAnDWR8	<TAUAn_base1> + 120_H
TAUAn DMA window register 9	TAUAnDWR9	<TAUAn_base1> + 124_H
TAUAn DMA window register 10	TAUAnDWR10	<TAUAn_base1> + 128_H
TAUAn DMA window register 11	TAUAnDWR11	<TAUAn_base1> + $12C_H$
TAUAn DMA window register 12	TAUAnDWR12	<TAUAn_base1> + 130_H
TAUAn DMA window register 13	TAUAnDWR13	<TAUAn_base1> + 134_H
TAUAn DMA window register 14	TAUAnDWR14	<TAUAn_base1> + 138_H
TAUAn DMA window register 15	TAUAnDWR15	<TAUAn_base1> + $13C_H$

Note TAUAn base addresses <TAUAn_base0> and <TAUAn_base1> are defined according to table 13-2, Register Base Addresses <TAUAn_basen>.

Caution If a specified value is described for each register bit, any other value than the specified one should not be written into the corresponding bit.

13.15.2 Details of TAUAn Prescaler Registers

(1) TAUAnTPS - TAUAn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3_PRE by the factor specified in TAUAnBRS.

Access Readable/writable in 16-bit units.

Address <TAUAn_base0> + 240_H

Initial value $FFFF_H$

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnPRS3[3:0]				TAUAnPRS2[3:0]				TAUAnPRS1[3:0]				TAUAnPRS0[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-201 Description of TAUAnTPS Register (1/4)

Bit Position	Bit Name	Function
15 to 12	TAUAnPRS3 [3:0]	Specifies CK3_PRE clock. CK3_PRE clock is an input clock to BRG unit which supplies prescaler output CK3 to all channels. **TAUAnPRS3[3:0] / CK3_PRE Clock** 0000_B / PCLK/2^0 0001_B / PCLK/2^1 0010_B / PCLK/2^2 0011_B / PCLK/2^3 0100_B / PCLK/2^4 0101_B / PCLK/2^5 0110_B / PCLK/2^6 0111_B / PCLK/2^7 1000_B / PCLK/2^8 1001_B / PCLK/2^9 1010_B / PCLK/2^{10} 1011_B / PCLK/2^{11} 1100_B / PCLK/2^{12} 1101_B / PCLK/2^{13} 1110_B / PCLK/2^{14} 1111_B / PCLK/2^{15} The above bits are rewritable only when all the counters using CK3 are stopped (TAUAnTE.TAUAnTEm = 0).

Table 13-201 Description of TAUAnTPS Register (2/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUAnPRS2 [3:0]</td><td>Specifies prescaler output CK2 clock.
<table>
<tr><th>TAUAnPRS2[3:0]</th><th>Prescaler Output CK2 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

Table 13-201 Description of TAUAnTPS Register (3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>TAUAnPRS1 [3:0]</td><td>Specifies prescaler output CK1 clock.
<table>
<tr><th>TAUAnPRS1[3:0]</th><th>Prescaler Output CK1 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

Table 13-201 Description of TAUAnTPS Register (4/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUAnPRS0 [3:0]</td><td>Specifies prescaler output CK0 clock.
<table>
<tr><th>TAUAnPRS0[3:0]</th><th>Prescaler Output CK0 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK0 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

(2) TAUAnBRS - TAUAn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in

TAUAnTPS.TAUAnPRS3[3:0].

Access Readable/writable in 8-bit units.

Address <TAUAn_base0> + 244_H

Initial value 00_H

This register is initialized by any reset source.

7	6	5	4	3	2	1	0
TAUAnBRS[07:00]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-202 Description of TAUAnBRS Register

Bit Position	Bit Name	Function
7 to 0	TAUAnBRS [07:00]	Specifies a CK3_PRE clock division coefficient for generating prescaler output CK3.

TAUAnBRS[07:00]	Prescaler Output CK3 Clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

13.15.3 Details of TAUAn Control Registers

(1) TAUAnCDRm - TAUAn Channel Data Register m

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUAnCMORm.TAUAnMD[4:1].

Access Readable/writable in 16-bit units.

- Readable in capture mode. Any write operation is ignored.
- Readable/writable in compare mode.

Address <TAUAn_base1> + m × 4_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCDR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-203 Description of TAUAnCDRm Register

Bit Position	Bit Name	Function
15 to 0	TAUAnCDR [15:0]	Data register for capture/compare values

(2) TAUAnCNTm - TAUAn Channel Counter Register m

This is a channel m counter register.

Access Readable in 16-bit units.

Address <TAUAn_base1> + 80_H + m × 4_H

Initial value 0000_H or $FFFF_H$

An initial value depends on an operating mode. See table 13-205, TAUAnCNTm Read Values after Re-Enabling Counter. This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13-204 Description of TAUAnCNTm Register

Bit Position	Bit Name	Function
15 to 0	TAUAnCNT [15:0]	16-bit counter value

A read value depends on a counter value, a changed operating mode, TAUAnTS.TAUAnTSm or TAUAnTT.TAUAnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUAnTT.TAUAnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUAnTE.TAUAnTEm = 0) and re-enabled (TAUAnTS.TAUAnTSm = 1).
The table also contains the counter read value one count after the counter is enabled (TAUAnTS.TAUAnTSm = 1) with the counter waiting for a start trigger.

Table 13-205 TAUAnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUAnCNTm Value		
		After Reset	After Stop Trigger	After One Count
Interval timer mode	Count down	$FFFF_H$	Stop value	-
Judge mode	Count down	$FFFF_H$	Stop value	-
Capture mode	Count up	0000_H	Stop value	-
Event count mode	Count down	$FFFF_H$	Stop value	-
One-count mode	Count down	$FFFF_H$	Stop value	$FFFF_H$
Capture and one-count mode	Count up	0000_H	Stop value	Capture value + 1 (TAUAnCDRm)
Judge and one-count mode	Count down	$FFFF_H$	Stop value	TAUAnCNTm value - 1
Up/down count mode	Count down/up	$FFFF_H$	Stop value	-
Pulse one count mode	Count down	$FFFF_H$	Stop value	0000_H
Count capture mode	Count up	0000_H	Stop value	-
Gate count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000_H	Stop value	Stop value

Note If the operating mode is changed while the counter is stopped, the initial counter value after a counter restart becomes undefined. The operating mode is changed by the TAUAnCMORm.TAUAnMD[4:1] register.

(3) TAUAnCMORm - TAUAn Channel Mode OS Register m

This register controls channel m operation.

Access Readable/writable in 16-bit units. Writable when the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 200_H + m × 4_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCKS [1:0]		TAUAnCCS [1:0]		TAUAnMAS	TAUAnSTS[2:0]			TAUAnCOS [1:0]		-	TAUAnMD[4:0]				
R/W		R/W		R/W	R/W			R/W		R	R/W				

Table 13-206 Description of TAUAnCMORm Register (1/3)

Bit Position	Bit Name	Function
15, 14	TAUAnCKS [1:0]	Selects a sampling clock. A sampling clock is used for the TAUAnTTINm input edge detection circuit. TAUAnCMORm.TAUAnCCS[1:0] bit setting enables use as a counter clock.
13, 12	TAUAnCCS [1:0]	Selects a count clock for TAUAnCNTm counter.
11	TAUAnMAS	Specifies whether the channel is a master channel or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm_odd) are fixed to 0.

TAUAn CKS1	TAUAn CKS0	Selection of Sampling Clock
0	0	Prescaler output CK0
0	1	Prescaler output CK1
1	0	Prescaler output CK2
1	1	Prescaler output CK3

TAUAn CCS1	TAUAn CCS0	Selection of Count Clock
0	0	Sampling clock specified by TAUAnCMORm.TAUAnCKS[1:0]
0	1	Valid edge of TAUAnTTINm input signal
1	0	Setting prohibited
1	1	INTTAUAnIm signal of master channel

Table 13-206 Description of TAUAnCMORm Register (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUAnSTS [2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUAn STS2</th><th>TAUAn STS1</th><th>TAUAn STS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUAnTTINm input signal, which is specified by TAUAnCMURm.TAUAnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUAnTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Simultaneous rewrite trigger</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INT of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INT of upper channel (m - 1) regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead time output signal of TAUAnTTOUTm generating unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of master channel</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUAnCOS [1:0]</td><td>Specifies the timing for updating capture register TAUAnCDRm and overflow flag TAUAnCSRm.TAUAnOVF of channel m.
These bits are valid only when channel m is in capture mode.
<table>
<tr><th>TAUAn COS1</th><th>TAUAn COS0</th><th>TAUAnCDRm</th><th>TAUAnCSRm.TAUAnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUAnTTINm input.</td><td>Updated (cleared or set) by detecting valid edge of TAUAnTTINm input:
• If a counter overflow has occurred since the last detection of valid edge, set TAUAnCSRm. TAUAnOVF.
• If no counter overflow has occurred since the last detection of valid edge, clear TAUAnCSRm.TAUAnOVF.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUAnCSCm.TAUAnCLOV is set to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUAnTTINm input and at the occurrence of counter overflow:
• Detection of valid edge of TAUAnTTINm input: Counter value is written into TAUAnCDRm.
• Occurrence of overflow: $FFFF_H$ is loaded into TAUAnCDRm. The next detection of valid of TAUAnTTINm input is ignored.</td><td>Not set</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUAnCSCm.TAUAnCLOV is set to 1.</td></tr>
</table></td></tr>
</table>

Table 13-206 Description of TAUAnCMORm Register (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUAnMD [4:0]</td><td>Specifies an operating mode.
<table>
<tr><th>TAUAn MD4</th><th>TAUAn MD3</th><th>TAUAn MD2</th><th>TAUAn MD1</th><th>TAUAn MD0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Judge mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Event count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Judge and one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Up/down count mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Pulse one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr>
</table>
</td></tr>
</table>

Mode	Role of TAUAnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUAnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUAnIm is not generated. 1: INTTAUAnIm is generated.
Event count mode Up/down count mode	This bit should be set to 0.
One-count mode Gate count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.
Capture and one-count mode Capture and gate count mode	This bit should be set to 0.
Judge mode Judge and one-count mode	Specifies INTTAUAnIm output timing. 0: When TAUAnCNTm ≤ TAUAnCDRm 1: When TAUAnCNTm > TAUAnCDRm

(4) TAUAnCMURm - TAUAn Channel Mode User Register m

This register specifies a type of valid edge detection used for TAUAnTTINm input.

Access Readable/writable in 8-bit units.

Address <TAUAn_base1> + $C0_H$ + m × 4_H

Initial value 00_H

This register is initialized by any reset source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUAnTIS[1:0]	
R	R	R	R	R	R	R/W	R/W

Table 13-207 Description of TAUAnCMURm Register

Bit Position	Bit Name	Function
1, 0	TAUAnTIS [1:0]	Specifies a valid edge of TAUAnTTINm input signal. • Edge detection of TAUAnTTINm input signal is based on the sampling clock selected by TAUAnCMORm.TAUAnCKS[1:0].

TAUAn TIS1	TAUAn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detects both of falling and rising edges (Selects low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detects both of falling and rising edges (Selects high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

(5) TAUAnCSRm - TAUAn Channel Status Register m

This register indicates the count direction and overflow status of channel m counter.

Access Readable in 8-bit units.

Address <TAUAn_base1> + 140_H + m × 4_H

Initial value 00_H

This register is initialized by any reset source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUAnCSF	TAUAnOVF
R	R	R	R	R	R	R	R

Table 13-208 Description of TAUAnCSRm Register

Bit Position	Bit Mame	Function
1	TAUAnCSF	Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in the following mode: • Up/down count
0	TAUAnOVF	Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the following modes: • Capture mode • Capture and one-count mode • Count capture mode • Capture and gate count mode The function of this bit depends on the setting of control bit TAUAnCMORm.TAUAnCOS[1:0].

(6) TAUAnCSCm - TAUAn Channel Status Clear Trigger Register m

This is a trigger register for clearing the overflow flag TAUAnCSRm.TAUAnOVF of channel m.

Access Writable in 8-bit units. This value is always read as 00_H.

Address <TAUAn_base1> + 180_H + m × 4_H

Initial value 00_H

This register is initialized by any reset source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	0	TAUAnCLOV
R	R	R	R	R	R	R	W

Table 13-209 Description of TAUAnCSCm Register

Bit Position	Bit Name	Function
0	TAUAnCLOV	Controls whether to clear the overflow flag (TAUAnCSRm.TAUAnOVF bit) of channel m. 0: Invalid (Setting this bit to 0 does not affect the overflow flag TAUAnCSRm.TAUAnOVF bit.) 1: Clears overflow flag TAUAnCSRm.TAUAnOVF.

(7) TAUAnTS - TAUAn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access Writable in 16-bit units. This value is always read as 0000_H.

Address <TAUAn_base1> + $1C4_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTS15	TAUAnTS14	TAUAnTS13	TAUAnTS12	TAUAnTS11	TAUAnTS10	TAUAnTS09	TAUAnTS08	TAUAnTS07	TAUAnTS06	TAUAnTS05	TAUAnTS04	TAUAnTS03	TAUAnTS02	TAUAnTS01	TAUAnTS00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 13-210 Description of TAUAnTS Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTSm	Enables the counter operation of channel m. 0: Invalid (Setting these bits to 0 does not start the counter operation of channel m.) 1: Enables the counter operation and sets TAUAnTE.TAUAnTEm to 1. The counter operation is only enabled when TAUAnTE.TAUAnTEm is set to 1. Whether counting is started or not depends on a selected operating mode.

(8) TAUAnTE - TAUAn Channel Enable Status Register

This register enables/disables a counter operation.

Access Readable in 16-bit units.

Address <TAUAn_base1> + $1C0_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTE15	TAUAnTE14	TAUAnTE13	TAUAnTE12	TAUAnTE11	TAUAnTE10	TAUAnTE09	TAUAnTE08	TAUAnTE07	TAUAnTE06	TAUAnTE05	TAUAnTE04	TAUAnTE03	TAUAnTE02	TAUAnTE01	TAUAnTE00
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13-211 Description of TAUAnTE Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTEm	Enables/disables the counter operation of channel m. 0: Disables counter operation. 1: Enables counter operation. This bit is set to 1 when trigger input of TAUAnTSSTm (synchronous channel start trigger signal) is detected or when TAUAnTS.TAUAnTSm is set to 1. This bit is set to 0 when TAUAnTT.TAUAnTTm is set to 1.

Note For details on how to make a simultaneous start between units, see section 26.4.1, Simultaneous Start Trigger Function.

(9) TAUAnTT - TAUAn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access Writable in 16-bit units. This value is always read as 0000_H.

Address <TAUAn_base1> + $1C8_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTT15	TAUAnTT14	TAUAnTT13	TAUAnTT12	TAUAnTT11	TAUAnTT10	TAUAnTT09	TAUAnTT08	TAUAnTT07	TAUAnTT06	TAUAnTT05	TAUAnTT04	TAUAnTT03	TAUAnTT02	TAUAnTT01	TAUAnTT00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 13-212 Description of TAUAnTT Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTTm	Stops the counter operation of channel m. 0: Invalid (Setting these bits to 0 does not stop the counter operation of channel m.) 1: Stops the counter operation and resets TAUAnTE.TAUAnTEm. TAUAnCNTm, TAUAnTO.TAUAnTOm, and TAUAnTTOUTm retain the values provided before the counter is stopped.

13.15.4 Details of TAUAn Output Registers

(1) TAUAnTOE - TAUAn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm bit = 0).

Address <TAUAn_base1> + $5C_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTOE15	TAUAnTOE14	TAUAnTOE13	TAUAnTOE12	TAUAnTOE11	TAUAnTOE10	TAUAnTOE09	TAUAnTOE08	TAUAnTOE07	TAUAnTOE06	TAUAnTOE05	TAUAnTOE04	TAUAnTOE03	TAUAnTOE02	TAUAnTOE01	TAUAnTOE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-213 Description of TAUAnTOE Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTOEm	Enables/disables the independent timer output function. 0: Disables the independent timer output function (software control). 1: Enables the independent timer output function.

Note The output from TAUAnTTOUTm pin is controlled by TAUAnTO.TAUAnTOm (software control).

(2) TAUAnTOM - TAUAn Channel Output Mode Register

This register specifies the output mode of each channel.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 248_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTOM15	TAUAnTOM14	TAUAnTOM13	TAUAnTOM12	TAUAnTOM11	TAUAnTOM10	TAUAnTOM09	TAUAnTOM08	TAUAnTOM07	TAUAnTOM06	TAUAnTOM05	TAUAnTOM04	TAUAnTOM03	TAUAnTOM02	TAUAnTOM01	TAUAnTOM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-214 Description of TAUAnTOM Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation As described in table 13-9, Channel Output Modes, the output mode depends on the setting of each channel output control bit.

(3) TAUAnTOC - TAUAn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUAnTOMm.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + $24C_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTOC15	TAUAnTOC14	TAUAnTOC13	TAUAnTOC12	TAUAnTOC11	TAUAnTOC10	TAUAnTOC09	TAUAnTOC08	TAUAnTOC07	TAUAnTOC06	TAUAnTOC05	TAUAnTOC04	TAUAnTOC03	TAUAnTOC02	TAUAnTOC01	TAUAnTOC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-215 Description of TAUAnTOC Register

Bit Position	Bit Name	Function
15 to 0	TAUAn TOCm	Specifies an output mode. 0: Operating mode 1 1: Operating mode 2 As listed below, the output mode depends on the setting of TAUAnTOM.TAUAnTOMm.

TOMm	TOCm	Functional Description
0	0	Toggle mode: Toggle operation is conducted when INTTAUAnIm occurs.
	1	Set/reset mode: Set when INTTAUAnIm occurs at the beginning of count operation, and reset when INTTAUAnIm is caused by detection of a match between TAUAnCNTm and TAUAnCDRm.
1	0	Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels.
	1	Synchronous channel operating mode 2: Set when INTTAUAnIm occurs in count-down status, and reset when INTTAUAnIm occurs in count-up status.

(4) TAUAnTDE - TAUAn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 250_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTDE15	TAUAnTDE14	TAUAnTDE13	TAUAnTDE12	TAUAnTDE11	TAUAnTDE10	TAUAnTDE09	TAUAnTDE08	TAUAnTDE07	TAUAnTDE06	TAUAnTDE05	TAUAnTDE04	TAUAnTDE03	TAUAnTDE02	TAUAnTDE01	TAUAnTDE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-216 Description of TAUAnTDE Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTDEm	Enables/disables the dead time control operation of channel m. 0: Disables dead time operation. 1: Enables dead time operation. The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, TAUAnTOC.TAUAnTOCm = 1

(5) TAUAnTDM - TAUAn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 254_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTDM15	TAUAnTDM14	TAUAnTDM13	TAUAnTDM12	TAUAnTDM11	TAUAnTDM10	TAUAnTDM09	TAUAnTDM08	TAUAnTDM07	TAUAnTDM06	TAUAnTDM05	TAUAnTDM04	TAUAnTDM03	TAUAnTDM02	TAUAnTDM01	TAUAnTDM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-217 Description of TAUAnTDM Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTDMm	Specifies the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even channel (duty dead time output). 1: When detecting the TIN input edge of a lower odd channel (one-phase dead time output). The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, TAUAnTOC.TAUAnTOCm, TAUAnTDE.TAUAnTDEm = 1

(6) TAUAnTDL - TAUAn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm bit = 0) (except the channel output mode for modulation output).

Address <TAUAn_base1> + 54_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTDL15	TAUAnTDL14	TAUAnTDL13	TAUAnTDL12	TAUAnTDL11	TAUAnTDL10	TAUAnTDL09	TAUAnTDL08	TAUAnTDL07	TAUAnTDL06	TAUAnTDL05	TAUAnTDL04	TAUAnTDL03	TAUAnTDL02	TAUAnTDL01	TAUAnTDL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-218 Description of TAUAnTDL Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTDLm	Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, TAUAnTOC.TAUAnTOCm, TAUAnTDE.TAUAnTDEm = 1

(7) TAUAnTRE - TAUAn Channel Real-Time Output Enable Register

This register enables/disables real-time output.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 258_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTRE15	TAUAnTRE14	TAUAnTRE13	TAUAnTRE12	TAUAnTRE11	TAUAnTRE10	TAUAnTRE09	TAUAnTRE08	TAUAnTRE07	TAUAnTRE06	TAUAnTRE05	TAUAnTRE04	TAUAnTRE03	TAUAnTRE02	TAUAnTRE01	TAUAnTRE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-219 Description of TAUAnTRE Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTREm	Enables or disables real-time output of channel m. 0: Disables real-time output. 1: Enables real-time output. These bit settings are applied only when TAUTnTOE.TAUAnTOEm = 1. When TAUAnTRE.TREm = 0, TAUAnTTOUTm is not affected by real-time output. When TAUAnTRE.TREm = 1, TAUAnTTOUTm outputs the value of real-time output bit TAUAnTRO.TROm in response to a timer operation.

(8) TAUAnTRC - TAUAn Channel Real-Time Output Control Register

This register controls the real-time output trigger of each channel.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + $25C_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTRC15	TAUAnTRC14	TAUAnTRC13	TAUAnTRC12	TAUAnTRC11	TAUAnTRC10	TAUAnTRC09	TAUAnTRC08	TAUAnTRC07	TAUAnTRC06	TAUAnTRC05	TAUAnTRC04	TAUAnTRC03	TAUAnTRC02	TAUAnTRC01	TAUAnTRC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-220 Description of TAUAnTRC Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTRCm	Specifies a channel on which the real-time output trigger for channel m is generated. 0: Next upper channel with this bit set to 1 1: Channel m These bit settings are applied only when TAUTnTRE.TAUAnTREm = 1.

(9) TAUAnTRO - TAUAn Channel Real-Time Output Register

This register sets a value which is output to TAUAnTTOUTm.

Access Readable/writable in 16-bit units.

Address <TAUAn_base1> + $4C_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTRO15	TAUAnTRO14	TAUAnTRO13	TAUAnTRO12	TAUAnTRO11	TAUAnTRO10	TAUAnTRO09	TAUAnTRO08	TAUAnTRO07	TAUAnTRO06	TAUAnTRO05	TAUAnTRO04	TAUAnTRO03	TAUAnTRO02	TAUAnTRO01	TAUAnTRO00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-221 Description of TAUAnTRO Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTROm	Sets a value which is output to TAUAnTTOUTm. 0: Low 1: High TAUAnTROm value is not output to TAUAnTTOUTm when TAUAnTRE.TAUAnTRE = 0, even if a real-time trigger occurs.

(10) TAUAnTME - TAUAn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm bit = 0) (except the channel output mode for modulation output).

Address <TAUAn_base1> + 50_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTME15	TAUAnTME14	TAUAnTME13	TAUAnTME12	TAUAnTME11	TAUAnTME10	TAUAnTME09	TAUAnTME08	TAUAnTME07	TAUAnTME06	TAUAnTME05	TAUAnTME04	TAUAnTME03	TAUAnTME02	TAUAnTME01	TAUAnTME00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-222 Description of TAUAnTME Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTMEm	Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation. 1: Enables modulation. These bit settings are applied only when TAUAnTOE.TAUAnTOEm and TAUAnTRE.TAUAnTREm = 1.

13.15.5 Details of TAUAn Channel Output Level Registers

(1) TAUAnTO - TAUAn Channel Output Register

This register specifies and reads a TAUAnTTOUTm level.

Access Readable/writable in 16-bit units.

Address <TAUAn_base1> + 58_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTO15	TAUAnTO14	TAUAnTO13	TAUAnTO12	TAUAnTO11	TAUAnTO10	TAUAnTO09	TAUAnTO08	TAUAnTO07	TAUAnTO06	TAUAnTO05	TAUAnTO04	TAUAnTO03	TAUAnTO02	TAUAnTO01	TAUAnTO00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-223 Description of TAUAnTO Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTOm	Specifies and reads a TAUAnTTOUTm level. 0: Low level 1: High level TAUAnTOm bit is writable when TAUAnTOE.TAUAnTOEm = 0.

(2) TAUAnTOL - TAUAn Channel Output Active Level Register

This register specifies the output logic of channel output bit (TAUAnTO.TAUAnTOm).

Access Readable/writable in 16-bit units.

Address <TAUAn_base1> + 40_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnTOL15	TAUAnTOL14	TAUAnTOL13	TAUAnTOL12	TAUAnTOL11	TAUAnTOL10	TAUAnTOL09	TAUAnTOL08	TAUAnTOL07	TAUAnTOL06	TAUAnTOL05	TAUAnTOL04	TAUAnTOL03	TAUAnTOL02	TAUAnTOL01	TAUAnTOL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-224 Description of TAUAnTOL Register

Bit Position	Bit Name	Function
15 to 0	TAUAnTOLm	Specifies the output logic of channel m output bit (TAUAnTO.TAUAnTOm). 0: Positive logic (active high) 1: Inverted logic (active low)

13.15.6 Details of TAUAn Simultaneous Rewrite Registers

(1) TAUAnRDE - TAUAn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUAnCDRm/ TAUAnTOLm data register.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 260_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRDE15	TAUAnRDE14	TAUAnRDE13	TAUAnRDE12	TAUAnRDE11	TAUAnRDE10	TAUAnRDE09	TAUAnRDE08	TAUAnRDE07	TAUAnRDE06	TAUAnRDE05	TAUAnRDE04	TAUAnRDE03	TAUAnRDE02	TAUAnRDE01	TAUAnRDE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-225 Description of TAUAnRDE Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite. 1: Enables simultaneous rewrite.

(2) TAUAnRDS - TAUAn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 268_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRDS15	TAUAnRDS14	TAUAnRDS13	TAUAnRDS12	TAUAnRDS11	TAUAnRDS10	TAUAnRDS09	TAUAnRDS08	TAUAnRDS07	TAUAnRDS06	TAUAnRDS05	TAUAnRDS04	TAUAnRDS03	TAUAnRDS02	TAUAnRDS01	TAUAnRDS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-226 Description of TAUAnRDS Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRDSm	Selects a channel that monitors a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

(3) TAUAnRDM - TAUAn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + 264_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRDM15	TAUAnRDM14	TAUAnRDM13	TAUAnRDM12	TAUAnRDM11	TAUAnRDM10	TAUAnRDM09	TAUAnRDM08	TAUAnRDM07	TAUAnRDM06	TAUAnRDM05	TAUAnRDM04	TAUAnRDM03	TAUAnRDM02	TAUAnRDM01	TAUAnRDM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-227 Description of TAUAnRDM Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied only when TAUAnRDE.TAUAnRDEm = 1 and TAUAnRDS.TAUAnRDSm = 0.

(4) TAUAnRDC - TAUAn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUAnlm signal to trigger simultaneous rewrite.

Access Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base0> + $26C_H$

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRDC15	TAUAnRDC14	TAUAnRDC13	TAUAnRDC12	TAUAnRDC11	TAUAnRDC10	TAUAnRDC09	TAUAnRDC08	TAUAnRDC07	TAUAnRDC06	TAUAnRDC05	TAUAnRDC04	TAUAnRDC03	TAUAnRDC02	TAUAnRDC01	TAUAnRDC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-228 Description of TAUAnRDC Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied only when TAUAnRDS.TAUAnRDSm = 1.

(5) TAUAnRDT - TAUAn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite pending state.

Access Writable in 16-bit units. This value is always read as 0000_H.

Address <TAUAn_base1> + 44_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRDT15	TAUAnRDT14	TAUAnRDT13	TAUAnRDT12	TAUAnRDT11	TAUAnRDT10	TAUAnRDT09	TAUAnRDT08	TAUAnRDT07	TAUAnRDT06	TAUAnRDT05	TAUAnRDT04	TAUAnRDT03	TAUAnRDT02	TAUAnRDT01	TAUAnRDT00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 13-229 Description of TAUAnRDT Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRDTm	Triggers a simultaneous rewrite pending state. 0: Invalid (Setting these bits to 0 does not trigger a simultaneous rewrite pending state.) 1: Triggers a simultaneous rewrite pending state. The simultaneous rewrite pending flag (TAUAnRSF.TAUAnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger.

(6) TAUAnRSF - TAUAn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access Readable in 16-bit units.

Address <TAUAn_base1> + 48_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnRSF15	TAUAnRSF14	TAUAnRSF13	TAUAnRSF12	TAUAnRSF11	TAUAnRSF10	TAUAnRSF09	TAUAnRSF08	TAUAnRSF07	TAUAnRSF06	TAUAnRSF05	TAUAnRSF04	TAUAnRSF03	TAUAnRSF02	TAUAnRSF01	TAUAnRSF00
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13-230 Description of TAUAnRSF Register

Bit Position	Bit Name	Function
15 to 0	TAUAnRSFm	Indicates simultaneous rewrite status. 0: Indicates that a simultaneous rewrite trigger has started simultaneous rewrite. 1: Indicates that simultaneous rewrite is in the pending state (TAUAnRDT.TAUAnRDTm = 1).

13.15.7 TAUAn DMA Window Registers

(1) TAUAnDASi - TAUAn DMA Window Address Setting Register i (i = 0 to 7)

This register specifies a window register address for DMA. Eight TAUAnDASi registers control 16 TAUAnDWRm registers. In other words, each TAUAnDASi register controls two TAUAnDWRm register.

Access Readable/writable in 16-bit units.

Address <TAUAn_base0> + 270_H + m × 4_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnDAS *odd7*	TAUAnDAS *odd6*	TAUAnDAS *odd5*	TAUAnDAS *odd4*	TAUAnDAS *odd3*	TAUAnDAS *odd2*	TAUAnDAS *odd1*	TAUAnDAS *odd0*	TAUAnDAS *even7*	TAUAnDAS *even6*	TAUAnDAS *even5*	TAUAnDAS *even4*	TAUAnDAS *even3*	TAUAnDAS *even2*	TAUAnDAS *even1*	TAUAnDAS *even0*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

odd = 01, 03, 05, 07, 09, 11, 13, 15

even = 00, 02, 04, 06, 08, 10, 12, 14

The following table shows the relationship between the TAUAnDASi control registers and the TAUAnDWRm registers.

Table 13-231 Relationship between TAUAnDASi and TAUAnDWRm Registers

Control Register	Control Bit	Control Target Register TAUAnDWRm
TAUAnDAS0	Bits 7 to 0	TAUAnDWR0
TAUAnDAS0	Bits 15 to 8	TAUAnDWR1
TAUAnDAS1	Bits 7 to 0	TAUAnDWR2
TAUAnDAS1	Bits 15 to 8	TAUAnDWR3
TAUAnDAS2	Bits 7 to 0	TAUAnDWR4
TAUAnDAS2	Bits 15 to 8	TAUAnDWR5
TAUAnDAS3	Bits 7 to 0	TAUAnDWR6
TAUAnDAS3	Bits 15 to 8	TAUAnDWR7
TAUAnDAS4	Bits 7 to 0	TAUAnDWR8
TAUAnDAS4	Bits 15 to 8	TAUAnDWR9
TAUAnDAS5	Bits 7 to 0	TAUAnDWR10
TAUAnDAS5	Bits 15 to 8	TAUAnDWR11
TAUAnDAS6	Bits 7 to 0	TAUAnDWR12
TAUAnDAS6	Bits 15 to 8	TAUAnDWR13
TAUAnDAS7	Bits 7 to 0	TAUAnDWR14
TAUAnDAS7	Bits 15 to 8	TAUAnDWR15

Table 13-232 Description of TAUAnDASm Register

Bit Position	Bit Name	Function
15 to 8	TAUAn DAS*odd*[15:8]	00_H to $3C_H$: Specifies CDR0 to CDR15 registers. 40_H: Specifies TAUAnTOL register. 44_H: Specifies TAUAnRDT register. 48_H: Specifies TAUAnRSF register. $4C_H$: Specifies TAUAnTRO register. 50_H: Specifies TAUAnTME register. 54_H: Specifies TAUAnTDL register. 58_H: Specifies TAUAnTO register. $5C_H$: Specifies TAUAnTOE register. 60_H to $7C_H$: Setting prohibited 80_H to BC_H: Specifies CNT0 to CNT15 registers. $C0_H$ to FC_H: Setting prohibited
7 to 0	TAUAn DAS*even*[7:0]	00_H to $3C_H$: Specifies CDR0 to CDR15 registers. 40_H: Specifies TAUAnTOL register. 44_H: Specifies TAUAnRDT register. 48_H: Specifies TAUAnRSF register. $4C_H$: Specifies TAUAnTRO register. 50_H: Specifies TAUAnTME register. 54_H: Specifies TAUAnTDL register. 58_H: Specifies TAUAnTO register. $5C_H$: Specifies TAUAnTOE register. 60_H to $7C_H$: Setting prohibited 80_H to BC_H: Specifies CNT0 to CNT15 registers. $C0_H$ to FC_H: Setting prohibited

Set TAUAnDASm [9:8] and [1:0] to 0.

(2) TAUAnDWRm - TAUAn DMA Window Register m

This is a register for DMA (m = 0 to 15). TAUAnDWRm mirrors an address specified by the corresponding TAUAnDASi (i = 0 to 7) register. (See the description on (1) TAUAnDASi - TAUAn DMA Window Address Setting Register i (i = 0 to 7)).

Access Readable/writable in 16-bit units.

Address <TAUAn_base1> + 100_H + m × 4_H

Initial value 0000_H

This register is initialized by any reset source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA window address specified by TAUAnDASi (i = 0 to 7)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Section 14 Timer Array Unit J (TAUJ)

This section provides general description of the timer array unit J (TAUJ).

The first part of this section describes all V850E2/Px4 specific properties, such as instances, register base addresses, input/output signal names, etc. The subsequent parts describe the features that apply to all TAUJ implementations.

Caution Some channels of the timer array unit J may not be usable for a combination of channels or input/output selection for ports.

14.1 Features of TAUJ

Instances This product has following number of instances of the timer array unit J.

Table 14-1 Instances of TAUJ

	TAUJ
No. of instances	2
Name	TAUJ0, TAUJ1

Instances index n Throughout this section, the individual instances of a timer array unit J is identified by the index "n" (n = 0 or 1), such as the TAUJn channel output mode register (TAUJnTOM).

Channel index m The timer array unit J has 4 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 3), thus a certain channel is denoted as CHm.
The even numbered channels (m = 0, 2) are denoted as CHm_even.
The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

Register addresses All TAUJn register addresses are given as address offsets to the individual base addresses <TAUJn_base0> and <TAUJn_base1>.
The register base addresses <TAUJn_base0> and <TAUJn_base1> of each TAUJn are listed in the following table.

Table 14-2 Register Base Addresses <TAUJn_base0> and <TAUJn_base1>

TAUJn	<TAUJn_base0> Address	<TAUJn_base1> Address
TAUJ0	FF81 1000_H	FFFF $C200_H$
TAUJ1	FF81 2000_H	FFFF $C300_H$

Clock supply The timer array unit J provides one clock input.

Table 14-3 TAUJn Clock Supply

TAUJn	Clock	Connected to
TAUJ0	PCLK	Clock controller
TAUJ1	PCLK	Clock controller

Interrupts and DMA/DTS The time array unit J can generate the following interrupt requests and DMA/DTS requests.

Table 14-4 TAUJn Interrupt and DMA/DTS Requests

TAUJn Signals	Function	Connected to
TAUJ0:		
INTTAUJ0I0-INTTAUJ0I3	Channels 0 to 3 interrupt	Interrupt Controllers INTTAUJ0I0 to INTTAUJ0I3 DMA Controller triggers 19 to 22 DTS Controller triggers 19 to 22
TAUJ1:		
INTTAUJ1I0-INTTAUJ1I3	Channels 0 to 3 interrupt	Interrupt Controllers INTTAUJ1I0 to INTTAUJ1I3 DMA Controller triggers 23 to 26 DTS Controller triggers 23 to 26

I/O signals The I/O signals of the timer array unit J are listed in the following table.

Table 14-5 TAUJn I/O Signals

TAUJ Signal	Function	Connected to
TAUJnTTINm	Channel m input	Port TAUJnlm
TAUJnTTOUTm	Channel m output	Port TAUJnOm
TAUJ0TTIN1	Channel 1 input	PIC CAN0 (time stamp function)
TAUJ0TTIN3	Channel 3 input	PIC CAN1 (time stamp function)
TAUJnTSSTm	Simultaneous start trigger input	PIC Simultaneous timer start trigger function

TAUJn interrupt and I/O signals are shown below.

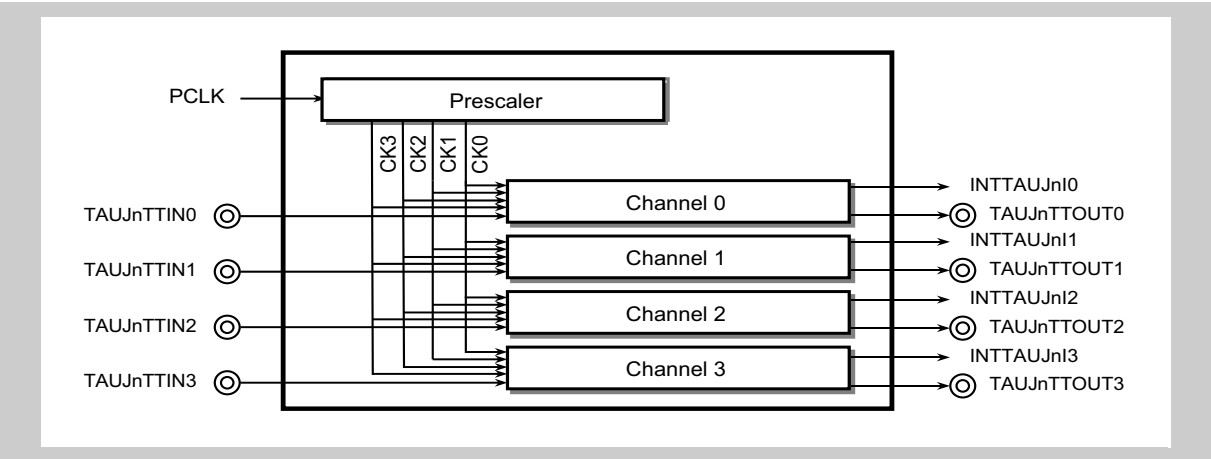


Figure 14-1 TAUJ I/O and Interrupt Signals

14.2 Functional Overview

Features summary

The TAUJ has the following functions.

- 4 channels
- 32-bit counter and 32-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operations)
- Generation of different types of output signals
- Counter can be triggered by external signals
- Interrupt generation

Caution The timing diagrams shown in this section are operation timing images. To the timer input, delay time is added. For details, see Section 14.11, TAUJnTTINm Edge Detection.

The following figure shows the main components of the TAUJ.

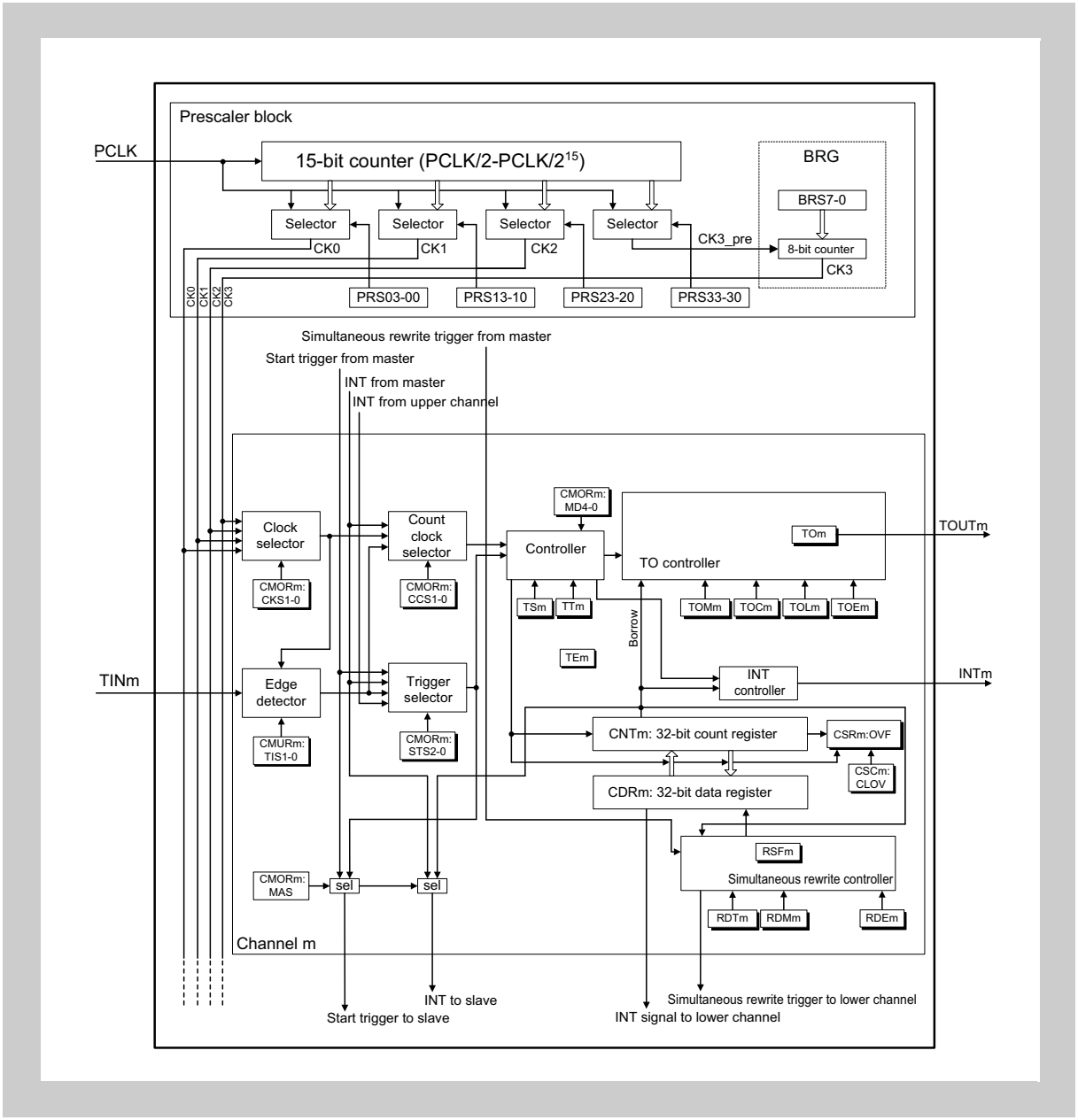


Figure 14-2 Block Diagram of TAUJ

"TAUJn" has been omitted from the register names for the sake of clarity in the above block diagram.

14.2.1 Terms

In this section, the following terms are used.

- **Independent/synchronous channel operation**

  Independent or synchronous channel operation describes the dependency of channels on each other.

  – If a channel operates independently of all other channels, this is called independent channel operation.

  – If a channel operates depending on other channels, this is called synchronous channel operation.

- **Channel group**

  In synchronous channel operation, all channels that depend on each other are referred to as a "channel group".

  A channel group has one master channel and one or more slave channels.

- **Operating mode**

  An operating mode can be selected for every channel m. The operating mode defines the basic operation and features of a channel.

  In synchronous channel operation, every channel in the channel group can operate in a different operating mode.

  Examples are "capture mode" and "interval timer mode".

- **Channel output mode**

  The channel output mode defines the operation of TAUJnTTOUTm of:

  – a single channel (independent output operation) or

  – all channels in a channel group (synchronous output operation).

  Independent channel output mode 1 is a channel output mode.

- **Channel operation function**

  The channel operation function defines the complete function and all features of:

  – a single channel (independent channel operation) or

  – all channels in a channel group (synchronous channel operation).

- **Upper/lower channel**

  A channel can be referred to as an "upper" or "lower" channel as defined below.

  – Upper channel: Channel with a smaller channel number

  – Lower channel: Channel with a larger channel number

  For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel.

14.3 Functional Description

The timer array unit J is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously), i.e. multiple channels depend on each other with one master and one or more slave channels.

When a channel is operated independently, its operating mode and functions are not affected by those of other channels. When a channel is operated synchronously, it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, one channel can control count start timing or reset timing of another channel.

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Prescaler outputs CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth prescaler output CK3 can be adjusted more precisely by an additional division factor that is not a power of 2.

Count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of the prescaler outputs CK0 to CK3 (selected by the clock selector)
- INTTAUJnIm from master channel
- Valid edge of the TAUJnTTINm input signal

Controller

The controller controls the main operations of the counter.

- Operating mode (selected by TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

  When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

Depending on the selected operating mode, the counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
  For details about simultaneous start between TAUJ units, see Section 26.4.1, Simultaneous Start Trigger Function.
- Valid edge of TAUJnTTINm input
- INTTAUJnIm from the master channel

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operating modes. The data registers of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller The output control of every channel enables the generation of various output signals such as PWM signals.

14.3.1 Timer Operation Functions

This timer provides the following functions by operating individual channels independently or a combination of channels.

Table 14-6 TAUJ Operation Functions

Independent Operation Function	Synchronous Operation Function
Independent channel operation function	Synchronous channel operation function
Interval timer function	PWM output function
TAUJnTTINm input interval timer function	
Independent channel signal measurement function	
TAUJnTTINm input pulse interval measurement function	
TAUJnTTINm input signal width measurement function	
TAUJnTTINm input period count detection function	
Other independent channel functions	
TAUJnTTINm input position detection function	

14.4 General Operating Procedure

The following lists the general operation procedure for TAUJn.

After a reset release, the operation of each channel is stopped. When the clock supply is started, writing to each register is enabled. The control register of TAUJnTTOUTm is also initialized to output a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   – Set the operating mode
   – Set the channel output mode
   – Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.

   The counter starts counting immediately or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

Note For details about necessary control bits and the operation of each function, see Section 14.12, Independent Channel Operation Functions, and Section 14.13, Synchronous Channel Functions.

14.5 Operating Modes

The TAUJ contains seven operating modes.

One operating mode can be set for each channel. It is specified using the TAUJnCMORm.TAUJnMD[4:0] bits.

Note Depending on the operation functions, some of the registers and bits have fixed values and the other can be set by the user.

For the values of registers and bits, see the sections of the corresponding operation functions.

14.6 Concepts of Synchronous Channel Operation

In synchronous channel operation, multiple channels depend on each other, or are affected by changes in other channels. Therefore, several rules apply to the use of synchronous channel functions. These rules are detailed in Section 14.6.1, Rules.

Two special features for synchronous channel operation are detailed in the following parts.

- Section 14.6.2, Simultaneous Start and Stop of Synchronous Channel Counters
- Section 14.7, Simultaneous Rewrite

14.6.1 Rules

Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
  Example: If CH0 and CH2 are master channels, CH1 can be set as slave channels for CH0, but CH3 cannot.

Count clock

- The same count clock should be set for the master channel and the slave channels synchronizing to the master channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and count clocks are illustrated in the following figure.

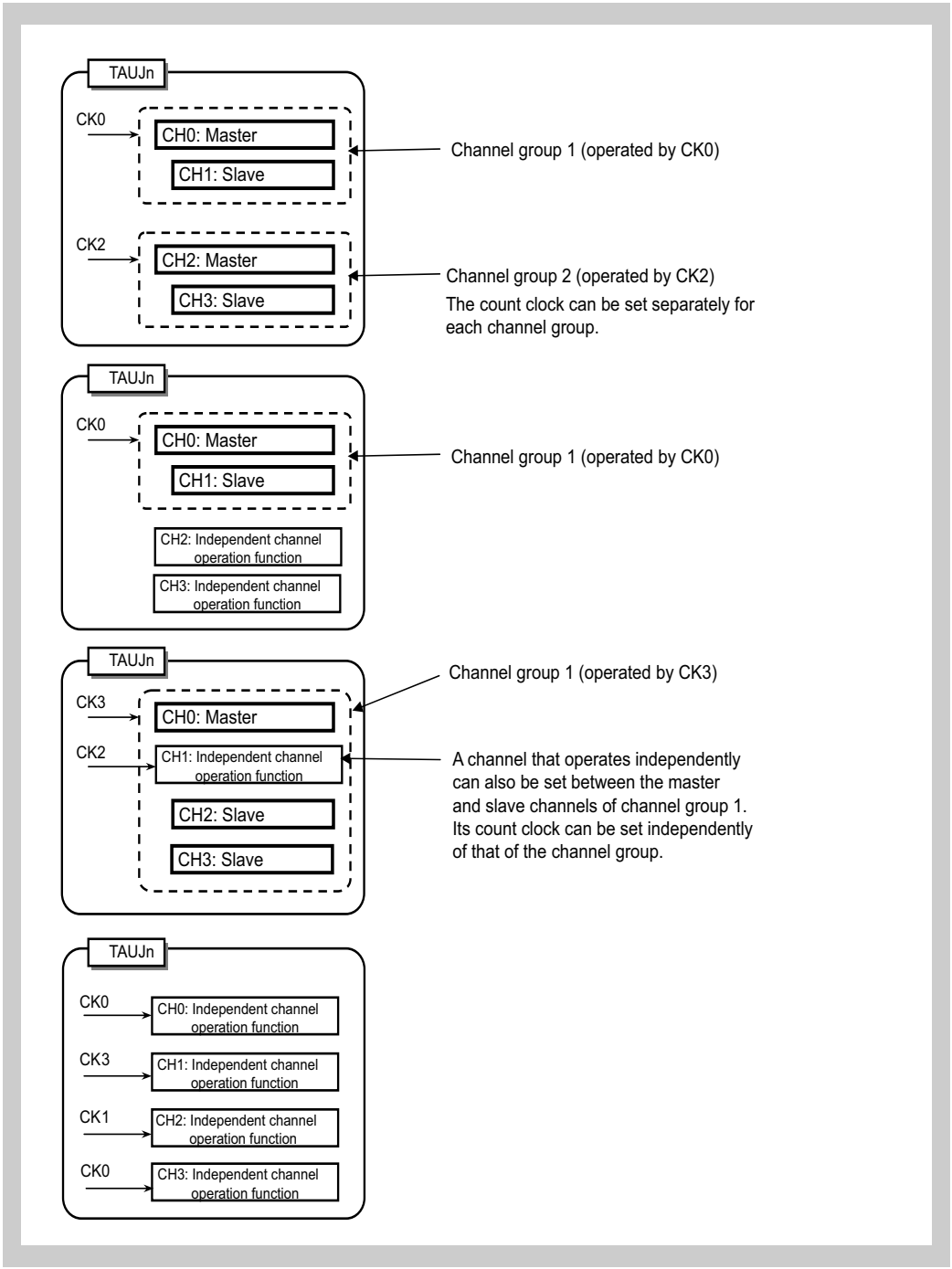


Figure 14-3 Grouping of Channels and Assignment of Count Clocks

Control trigger signals of master and slave channels

- Master channels can send a control signal to slave channels.
- Slave channels can use the control trigger signal sent from the master channels but cannot send their own control trigger signals to the lower channels.
- A master channel cannot use the control trigger signal sent from an upper master channel.

14.6.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

(1) Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEm bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

(2) Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to make a simultaneous start between TAUJ units, see Section 26.4.1, Simultaneous Start Trigger Function.

14.7 Simultaneous Rewrite

14.7.1 Overview

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUJnCDRm and TAUJnTOLm) can be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by the counter on the master channel reaching a certain value.

The following table shows the settings for simultaneous rewrite (TAUJnRDM.TAUJnRDMm = 0).

Table 14-7 Simultaneous Rewrite Settings

Method	Simultaneous Rewrite Trigger Timing	TAUJnRDE.TAUJnRDEm
-	No simultaneous rewrite	0
A	The master channel restarts/starts counting	1

14.7.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

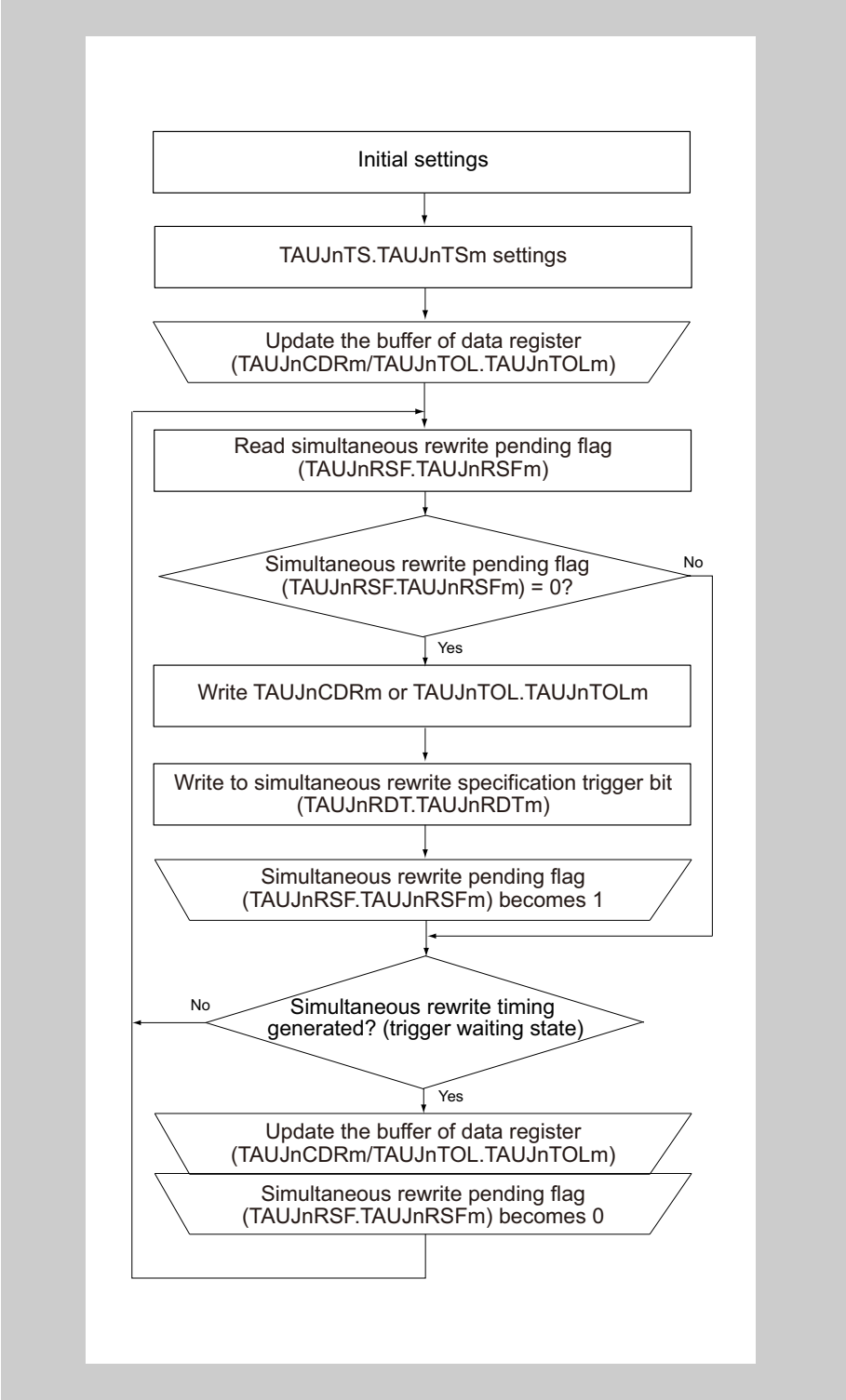


Figure 14-4 General Procedure for Simultaneous Rewrite

(1) Initial Settings

- Set TAUJnRDE.TAUJnRDEm = 1 to enable simultaneous rewriting on channel m.

(2) Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. TAUJnTOL.TAUJnTOLm and the values in the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

(3) Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is completed, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

14.7.3 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJnTTOUTm outputs an invalid wave.

14.7.4 Simultaneous Rewrite Procedure

Simultaneous rewrite is executed when the master channel starts or restarts counting.

The simultaneous rewrite procedure is described in the following figure.

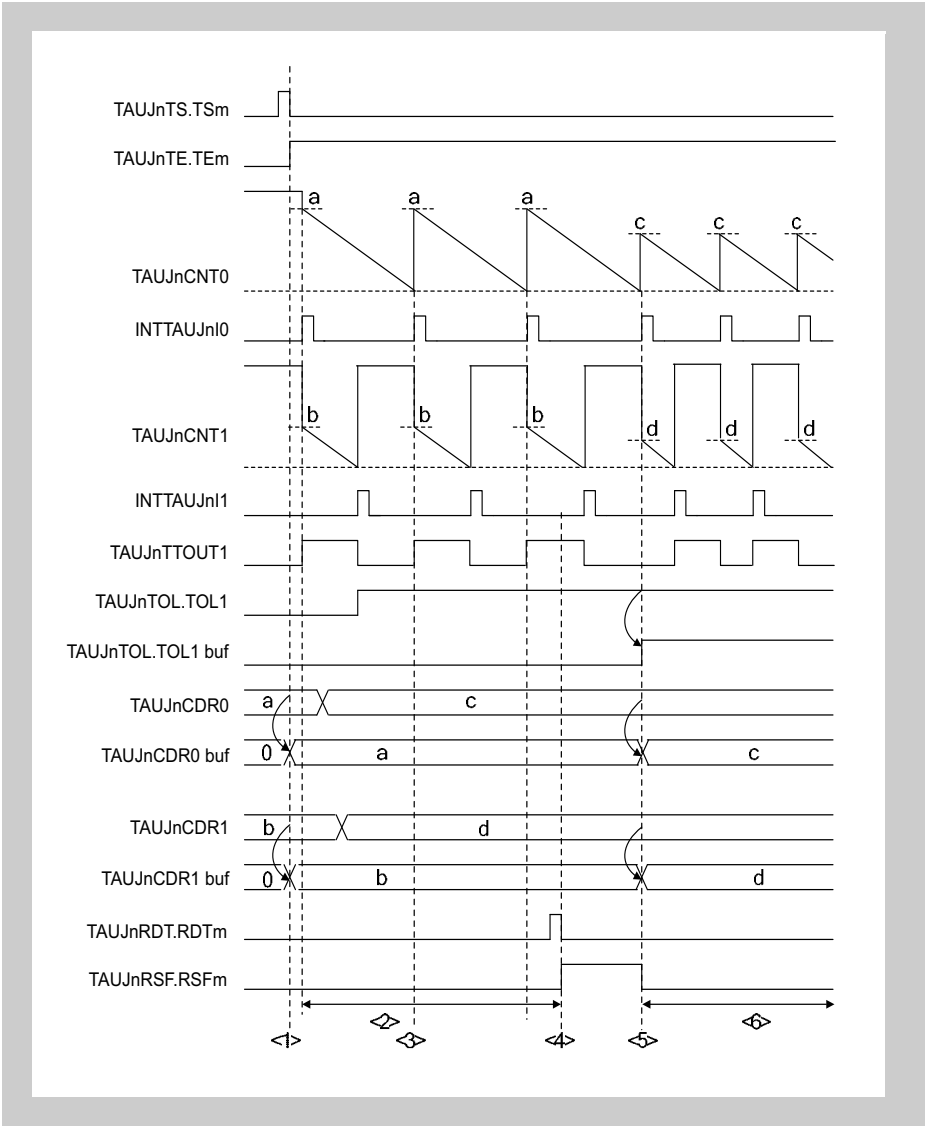


Figure 14-5 Simultaneous Rewrite When the Master Channel Starts/Restarts to Count

Setting

- CH0 is a master channel used to count down, and CH1 represents an arbitrary slave channel. Simultaneous rewrite is applied when the master channel starts counting.

Description:

1. When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.
2. The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.
3. CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).
4. The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.
5. Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.
6. The counter counts down and awaits the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

14.8 Channel Output Modes

The output of the TAUJnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- Control by software (TAUJnTOE.TAUJnTOEm = 0)

  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is output through the output pin (TAUJnTTOUTm).

- Control by TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)

  When controlled by TAUJ signals, the output level of TAUJnTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJnTTOUTm.

  – Independent control (TAUJnTOM.TAUJnTOMm = 0)

    In independent operation, the output of the TAUJnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).

  – Synchronous control (TAUJnTOM.TAUJnTOMm = 1)

    In synchronous operation, the output of the TAUJnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronizing channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJnTTOUTm, regardless of whether the pin is controlled by software or operated independently or synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in table 14-8, Channel Output Modes.

The channel output modes are described in detail in:

- Section 14.8.2, Channel Output Modes Controlled Independently by TAUJn Signals
- Section 14.8.3, Channel Output Modes Controlled Synchronously by TAUJn Signals

Batch operation of TAUJnTOm bit

The TAUJnTOE.TAUJnTOEm bit is used to control whether settings are to be reflected to the TAUJnTOm bit or not.

During the write operation to the TAUJnTO register, the setting of TAUJnTOm is written only to the bit (channel) for which TAUJnTOE.TAUJnTOEm = 0 is set. To the bit (channel) for which TAUJnTOE.TAUJnTOEm = 1 is set, the setting of TAUJnTOm is not reflected.

Note The TAUJnTO.TAUJnTOm bit is assigned so that the bit number corresponds to the channel number.

Output logic Positive logic or inverted logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of the TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can be changed during operation only with PWM output function. If TAUJnTOL.TAUJnTOLm is changed after the start of counter operation, an invalid TAUJnTTOUTm signal is output.

See Section 14.7, Simultaneous Rewrite.

The channel output modes and the channel output control bits are listed in the following table (TAUJnTOC.TAUJnTOCm = 0).

Table 14-8 Channel Output Modes

Channel Output Mode	TAUJnTOE. TAUJnTOEm	TAUJnTOM. TAUJnTOMm
By software		
Independent channel output mode controlled by software	0	X
By TAUJ signals, independently		
Independent channel output mode 1	1	0
By TAUJ signals, synchronously		
Synchronous channel output mode 1	1	1

Caution 1. The combinations not listed in this table are forbidden.
Caution 2. The bit marked with an "x" can be set to any value.
Caution 3. The following bits should not be changed during count operation (TAUJnTE.TAUJnTEm = 1).

- TAUJnTOE.TAUJnTOEm
- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

14.8.1 General Procedure for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJnTTOUTm channel output mode. The prerequisite is that a timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

1. Set TAUJnTO.TAUJnTOm to specify an initial level of the TAUJnTTOUTm output.
2. Set channel output mode according to table 14-8, Channel Output Modes, and set the output logic using the TAUJnTOL.TAUJnTOLm bit.
3. Start the counter (TAUJnTS.TAUJnTSm = 1).

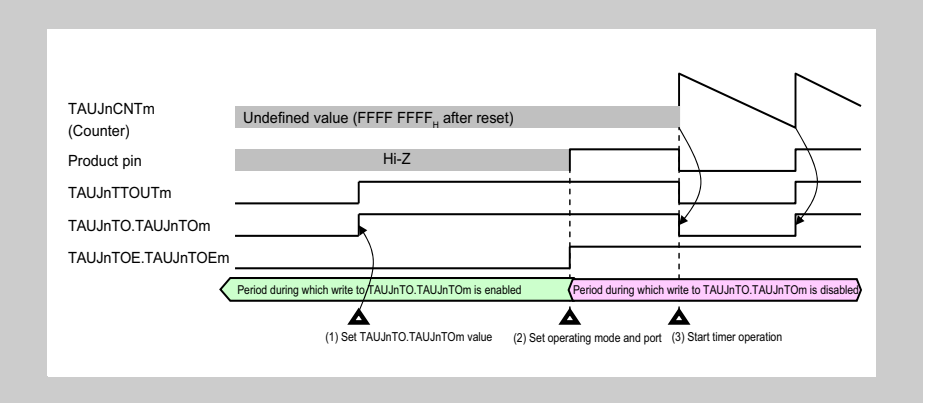


Figure 14-6 General Procedure for Specifying a TAUJnTTOUTm Channel Output Mode

14.8.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in table 14-8, Channel Output Modes.

(1) Independent Channel Output Mode 1

Set/reset conditions In this output mode, TAUJnTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites There are no prerequisites other than those shown in table 14-8, Channel Output Modes.

14.8.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in table 14-8, Channel Output Modes.

(1) Synchronous Channel Output Mode 1

Set/reset conditions In this output mode, INTTAUJnIm of the master channel serves as the set signal and INTTAUJnIm of the slave channel as the reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel (the master channel is ignored).

Prerequisites There are no prerequisites other than those shown in table 14-8, Channel Output Modes.

14.9 Start Timing of Operating Modes

This section describes the counter operation start timing after TAUJnTS.TAUJnTSm is set to 1.

The value of the data register (TAUJnCDRm register) and whether or not an interrupt occurs depend on mode and register settings.

Caution The count start timing described in this section is for your reference. Actually, the count start timing depends on count clock timing.

14.9.1 Interval Timer Mode and Capture Mode

The counter starts operating with the next count clock after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.

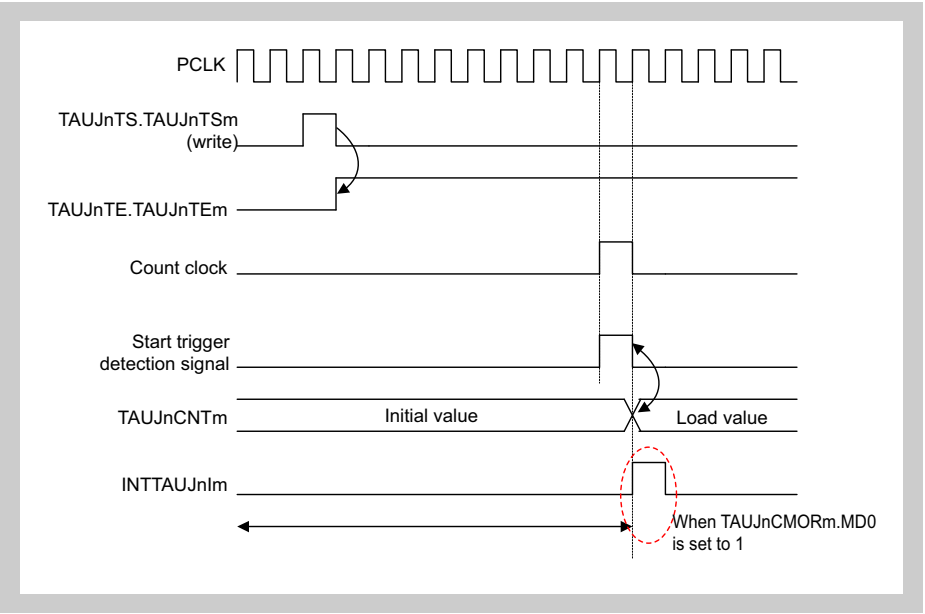


Figure 14-7 Start Timing in Interval Timer Mode and Capture Mode

14.9.2 Other Operating Modes

In other operating modes, the counter operation is triggered only by detection of a valid TAUJnTTINm edge. The value of data register is also loaded when the counter starts. The count clock cycles, which are unrelated to counter operation start, determine the frequency with which all operations take place.

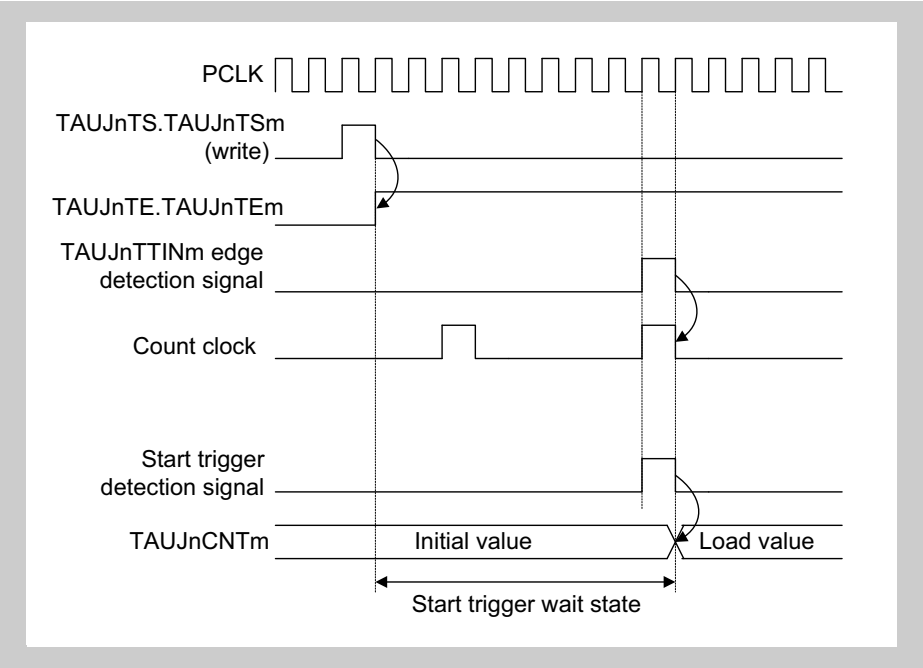


Figure 14-8 Count Start Timing in Other Operating Modes

14.10 TAUJnTTOUTm Output and INTTAUJnIm Generation When Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The effect of the bit depends on the selected mode, as shown in the following table. The effects of INTTAUJnIm on TAUJnTTOUTm depend on the selected channel operation function.

Table 14-9 Effect of TAUJnCMORm.TAUJnMD0 Bit on Generation of INTTAUJnIm when Counter is Triggered

Mode	TAUJnCMORm.TAUJnMD0 Bit	INTTAUJnIm Generated when Counter Starts/Restarts or TAUJnTTINm Input Signal Trigger is Detected
Interval timer mode Capture mode Count capture mode	0	Not generated
	1	Generated
Capture & one-count mode Capture & gate count mode	0	Not generated
One-count mode Gate count mode	0/1	Not generated, regardless of setting of TAUJnCMORm.TAUJnMD0 bit

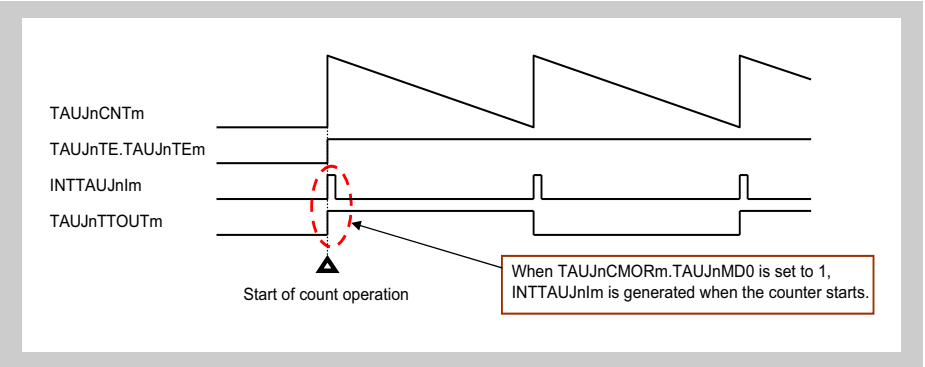


Figure 14-9 INTTAUJnIm Generated when Counter Starts

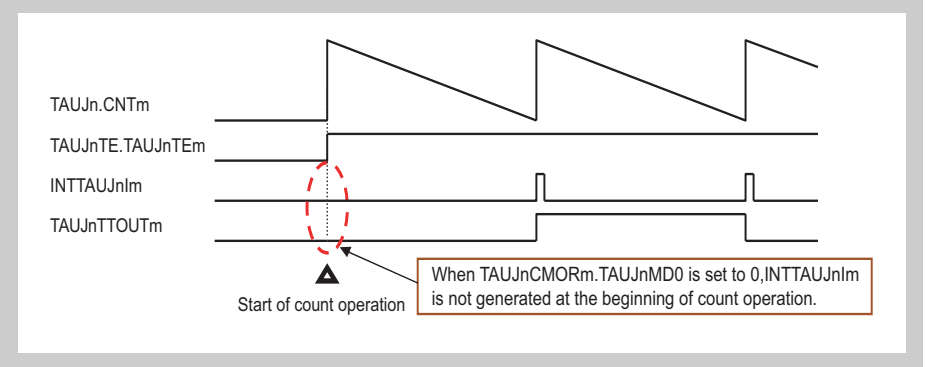


Figure 14-10 INTTAUJnIm Not Generated when Counter Starts

14.11 TAUJnTTINm Edge Detection

The TAUJnTTINm edge is detected at the rising edge of a sampling clock. A delay of one sampling clock cycle or less may occur.

The following figure shows when edge detection takes place.

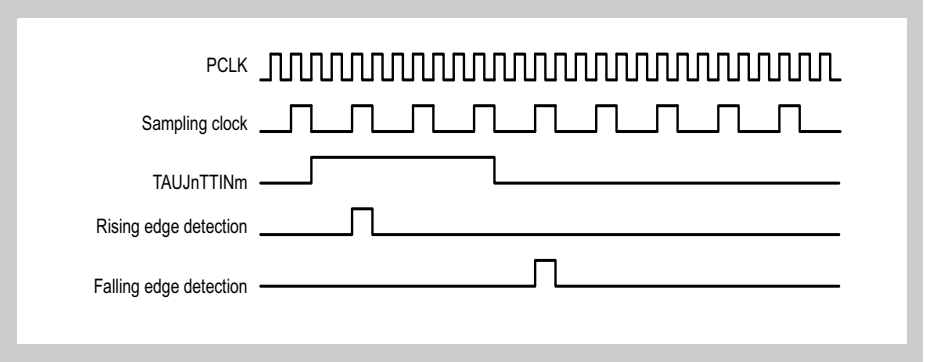


Figure 14-11 Basic Edge Detection Timing

Caution *figure14-11 Basic Edge Detection Timing*, shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

- When the noise filter is used

  Delay time at the noise filter + delay time in edge detection (maximum 1 sampling clock)
- When the noise filter is not used

  Delay time at the synchronization circuit (maximum 2 PLCK) + delay time in edge detection (maximum 1 sampling clock)

14.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the timer array unit J. For a general overview of independent channel operation, see Section 14.3, Functional Description.

14.12.1 Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a rectangular wave.

Prerequisites
- The operating mode should be set to interval timer mode. (See table 14-10, TAUJnCMORm Settings for Interval Timer Function.)
- The channel output mode should be set to independent channel output mode 1. (See Section 14.8, Channel Output Modes.)

Functional description The counter is started by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded into TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJnTTOUTm signal toggles. Then the TAUJnCDRm value is loaded in TAUJnCNTm and the counter subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted without making a stop once by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJnTTOUTm does not toggle. This results in an inverted TAUJnTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1. For details, see Section 14.10, TAUJnTTOUTm Output and INTTAUJnIm Generation When Counter Starts or Restarts.

(2) Equations

INTTAUJnIm cycle = Count clock cycle x (TAUJnCDRm + 1)

TAUJnTTOUTm rectangular wave cycle = Count clock cycle x (TAUJnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

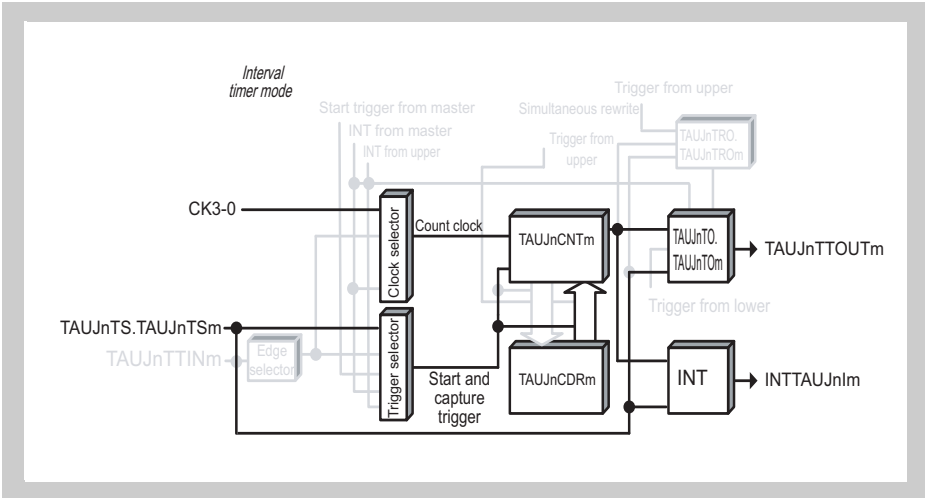


Figure 14-12 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at the beginning of operation. (TAUJnCMORm.TAUJnMD0 = 1)

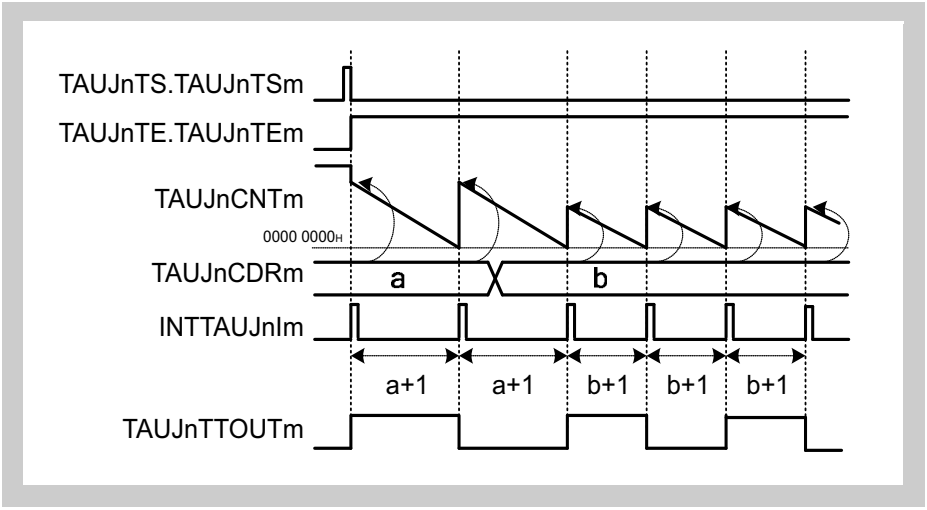


Figure 14-13 General Timing Diagram of Interval Timer Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-10 TAUJnCMORm Settings for Interval Timer Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects the sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	0: Unused. Set to 0
TAUJnSTS[2:0]	000: Counter triggered by software
TAUJnCOS[1:0]	00: Unused. Set to 00
TAUJnMD[4:1]	0000: Interval Timer Mode
TAUJnMD0	0: INTTAUJnIm is not generated and TAUJnTTOUTm does not toggle at the beginning of operation 1: INTTAUJnIm is generated and TAUJnTTOUTm toggles at beginning or restart time of operation

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-11 TAUJnCMURm Settings for Interval Timer Function

Bit Name	Setting
TAUJnTIS[1:0]	00: Unused. Set to 00

(c) Channel Output Mode

Table 14-12 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	1: Enables independent channel output mode
TAUJnTO.TAUJnTOm	0: Low level 1: High level
TAUJnTOM.TAUJnTOMm	0: Independent channel output
TAUJnTOC.TAUJnTOCm	0: Operating mode 1 (Toggle mode if TAUJnTOM.TAUJnTOMm = 0)
TAUJnTOL.TAUJnTOLm	0: Positive logic

Note The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details, see Section 14.8, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Interval Timer Function. Therefore, these registers should be set to 0.

Table 14-13 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for Interval Timer Function

Table 14-14 Operating Procedure for Interval Timer Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in 14-10 TAUJnCMORm Settings for Interval Timer Function, and table 14-11, TAUJnCMURm Settings for Interval Timer Function. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in table 14-12, Control Bit Settings for Independent Channel Output Mode 1.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCDRm value is loaded in TAUJnCNTm. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCDRm value is loaded again in TAUJnCNTm and count operation continues. • INTTAUJnIm is generated and TAUJnTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart

(6) Specific Timing Diagrams

(a) TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK/2

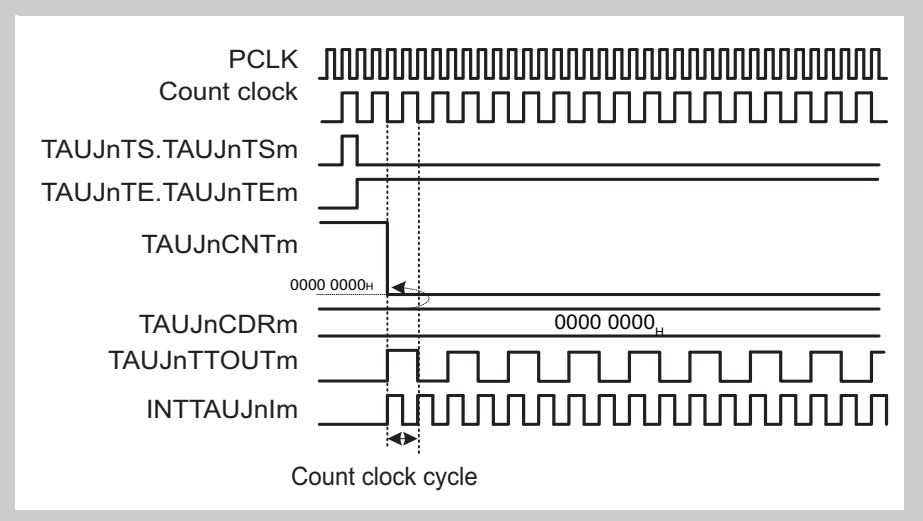


Figure 14-14 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK/2^1, the TAUJnCDRm value is loaded into TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is generated every count clock, resulting in TAUJnTTOUTm toggling every count clock.

(b) TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

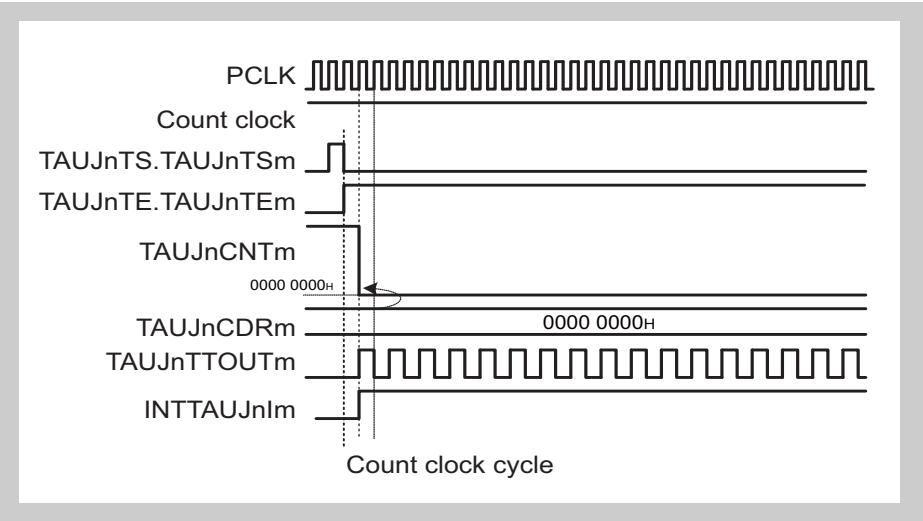


Figure 14-15 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK, the TAUJnCDRm value is loaded into TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is generated continuously, resulting in TAUJnTTOUTm toggling every PCLK clock.

(c) Operation stop and restart

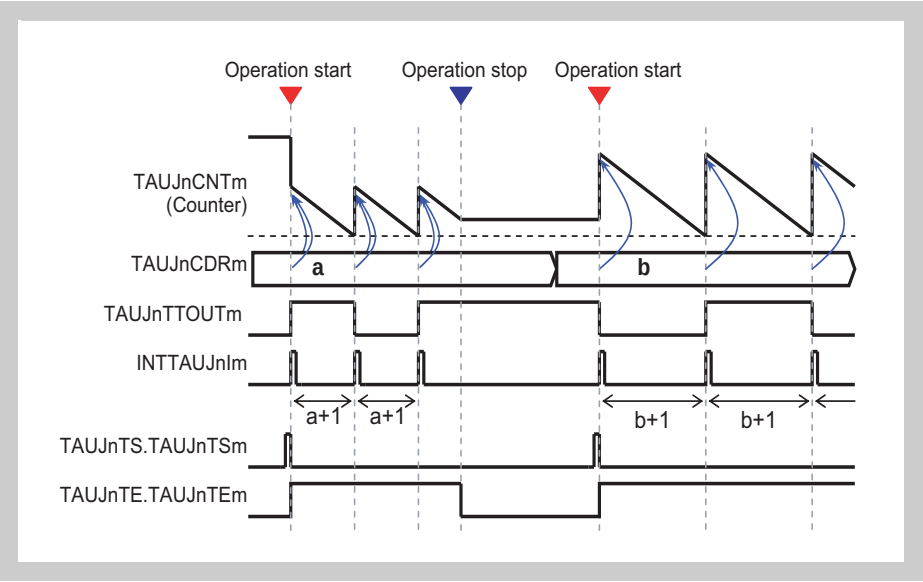


Figure 14-16 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(d) Forced restart

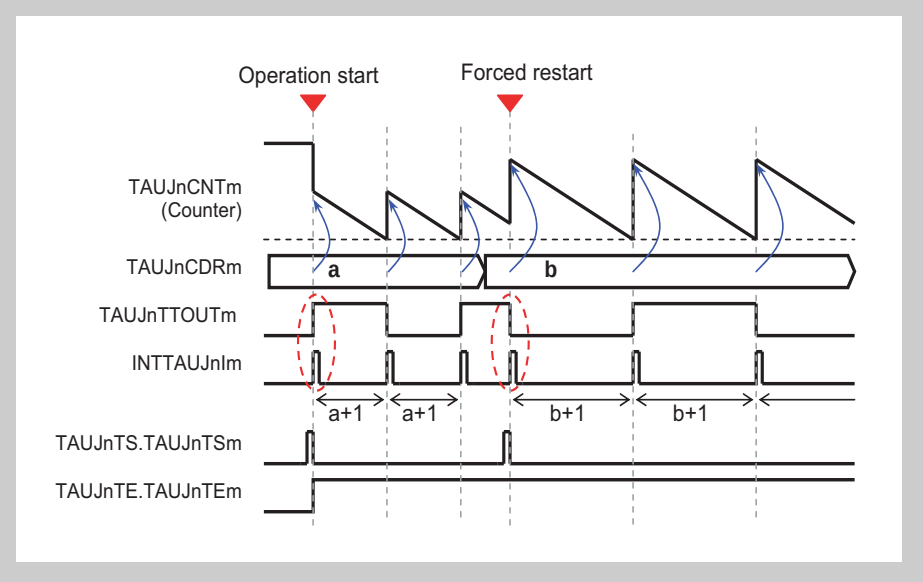


Figure 14-17 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.

14.12.2 TAUJnTTINm Input Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJnTTINm input edge is detected. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a rectangular wave.

Prerequisites

- The operating mode should be set to interval timer mode. (See 14-15 TAUJnCMORm Settings for TAUJnTTINm Input Interval Timer Function.)
- The channel output mode should be set to independent channel output mode 1. (See Section 14.8, Channel Output Modes.)

Functional description This function operates in the same manner as the interval timer function (see Section 14.12.1, Interval Timer Function), except that this function is restarted by a valid TAUJnTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm. TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

(2) Equations

INTTAUJnIm cycle = Count clock cycle x (TAUJnCDRm + 1)

TAUJnTTOUTm rectangular wave cycle = Count clock cycle × (TAUJnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

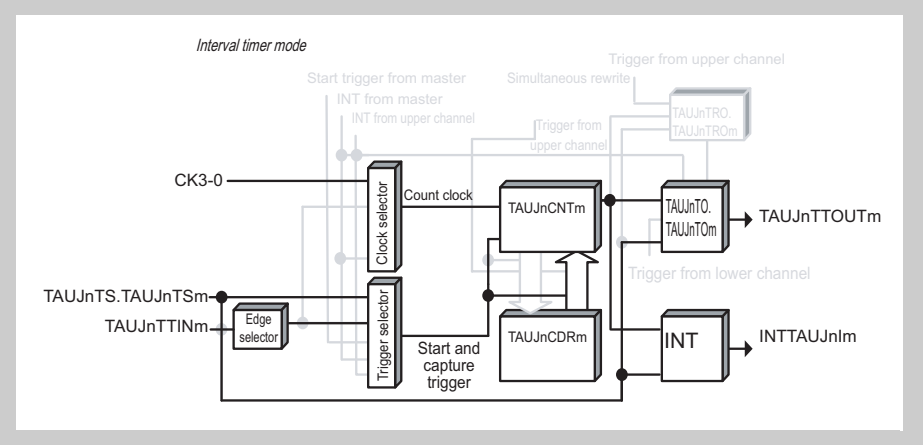


Figure 14-18 Block Diagram of TAUJnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at the beginning of operation. (TAUJnCMORm.TAUJnMD0 = 1)
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

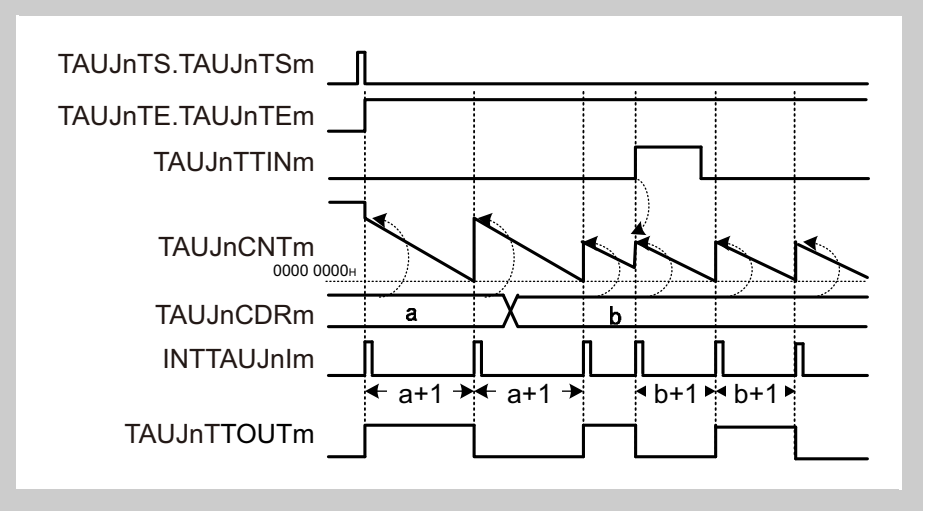


Figure 14-19 General Timing Diagram of TAUJnTTINm Input Interval Timer Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-15 TAUJnCMORm Settings for TAUJnTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects the sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as the count clock
TAUJnMAS	0: Unused. Set to 0
TAUJnSTS[2:0]	001: Valid TAUJnTTINm input edge signal is used as the external start trigger
TAUJnCOS[1:0]	00: Unused. Set to 00
TAUJnMD[4:1]	0000: Interval Timer Mode
TAUJnMD0	0: INTTAUJnIm is not generated and TAUJnTTOUTm does not toggle at the beginning of operation 1: INTTAUJnIm is generated and TAUJnTTOUTm toggles at the beginning of operation

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-16 TAUJnCMURm Settings for TAUJnTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel Output Mode

Table 14-17 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	1: Enables Independent Channel Output Mode
TAUJnTO.TAUJnTOm	0: Low level 1: High level
TAUJnTOM.TAUJnTOMm	0: Independent channel output
TAUJnTOC.TAUJnTOCm	0: Operating mode 1 (Toggle mode if TAUJnTOM.TAUJnTOMm = 0)
TAUJnTOL.TAUJnTOLm	0: Positive logic

Note The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details, see Section 14.8, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 14-18 Simultaneous Rewrite Settings for TAUJnTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Interval Timer Function

Table 14-19 Operating Procedure for TAUJnTTINm Input Interval Timer Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in table 14-15, TAUJnCMORm Settings for TAUJnTTINm Input Interval Timer Function, and table 14-16, TAUJnCMURm Settings for TAUJnTTINm Input Interval Timer Function. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in table 14-17, Control Bit Settings for Independent Channel Output Mode 1.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCDRm value is loaded in TAUJnCNTm. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and the TAUJnCDRm register can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJnTTINm edge.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation • INTTAUJnIm is generated and TAUJnTTOUTm toggles When a valid TAUJnTTINm input edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart

(6) Specific Timing Diagrams

Although the timing diagrams in Section 14.12.1, Interval Timer Function, also apply, but it is possible to use valid TAUJnTTINm input edge, by excluding this function, to restart the counter.

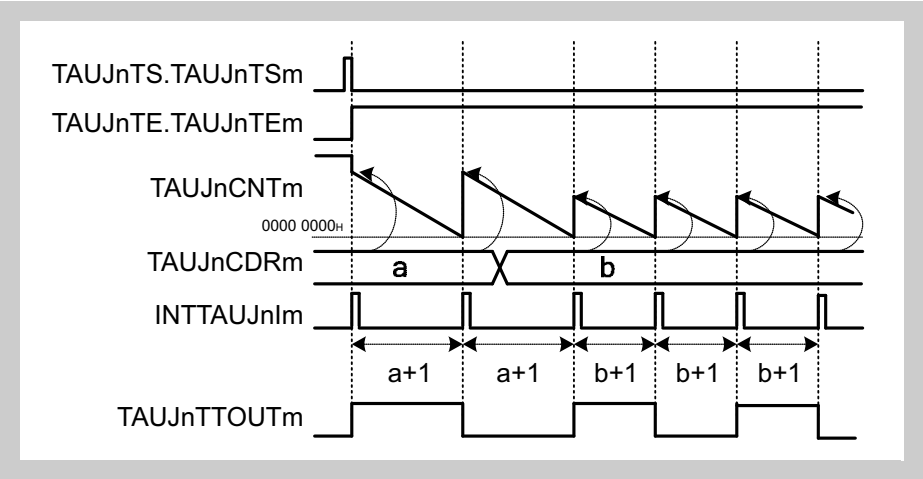


Figure 14-20 Counter Triggered by Rising TAUJnTTINm Input Edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

- If a valid TAUJnTTINm input edge is detected, an interrupt is generated which causes TAUJnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

14.12.3 TAUJnTTINm Input Pulse Interval Measurement Function

(1) Overview

Summary This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJnTTINm input signals.

Prerequisites
- The operating mode should be set to capture mode. (See table 14-21, TAUJnCMORm Settings for TAUJnTTINm Input Pulse Interval Measurement Function.)
- TAUJnTTOUTm is not used with this function.

Functional description The counter is started by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm edge is detected, the value of TAUJnCNTm is captured and transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counters reset to 0000 0000_H and subsequently continue operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm edge is detected, a counter overflow occurs. The counter is reset to 0000 0000_H and continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of TAUJnCMORm.TAUJnCOS[1:0] bits.

Table 14-20 Effects of an Overflow

TAUJnCMORm. TAUJnCOS[1:0]	When Overflow Occurs		When Valid TAUJnTTINm Input Is Detected after Overflow	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm, TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded into TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm is set to 0 and TAUJnCDRm remains unchanged	0
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can be cleared only by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJnTTINm signal. However, if multiple overflows occur before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate their occurrences.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, valid TAUJnTTINm input edge detection and TAUJnCNTm capture are not performed.

The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during counting.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see section 14.10, TAUJnTTOUTm Output and INTTAUJnIm Generation When Counter Starts or Restarts.

Note When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded into TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input pulse interval = Count clock cycle x [(TAUJnCSRm.TAUJnOVF x (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

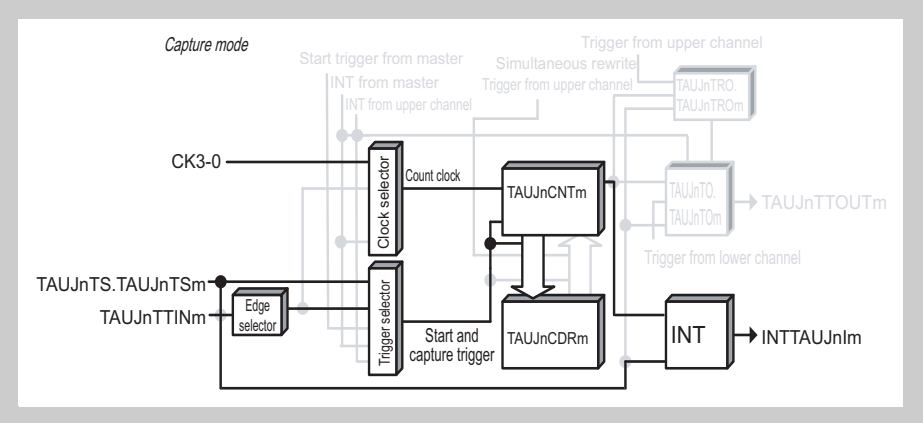


Figure 14-21 Block Diagram of TAUJnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is not generated at the beginning of operation. (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

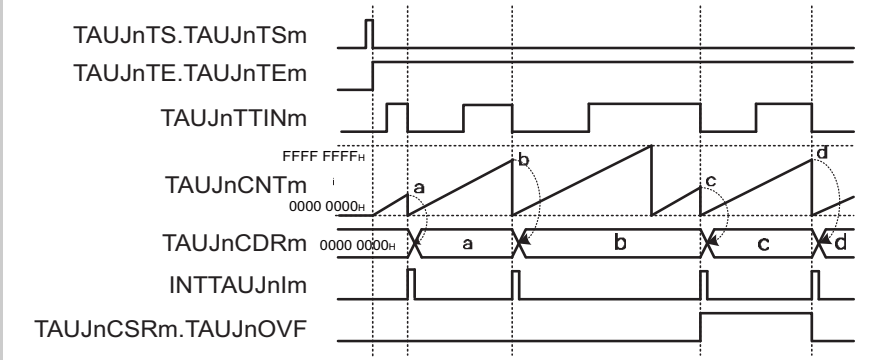


Figure 14-22 General Timing Diagram of TAUJnTTINm Input Pulse Interval Measurement Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJ nMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 14-21 TAUJnCMORm Settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects the sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	0: Unused. Set to 0
TAUJnSTS[2:0]	001: Valid edge of TAUJnTTINm input signal is used as an external capture trigger
TAUJnCOS[1:0]	See table 14-20, Effects of an Overflow
TAUJnMD[4:1]	0010: Capture mode
TAUJnMD0	0: INTTAUJnIm is not generated at the beginning of operation 1: INTTAUJnIm is generated at the beginning of operation

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-22 TAUJnCMURm Settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Pulse Interval Measurement Function. Therefore, these registers should be set to 0.

Table 14-23 Simultaneous Rewrite Settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

(5) Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

Table 14-24 Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in table 14-21, TAUJnCMORm Settings for TAUJnTTINm Input Pulse Interval Measurement Function, and table 14-22, TAUJnCMURm Settings for TAUJnTTINm Input Pulse Interval Measurement Function. TAUJnCDRm register operates as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated.
During operation	Detection of TAUJnTTINm edge. The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. TAUJnCDRm and TAUJnCSRm registers are readable at any time. TAUJnCSCm.TAUJnCLOV can be set to 1 (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000_H. When a valid edge of TAUJnTTINm is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm and returns to 0000 0000_H. • Then, INTTAUJnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops. TAUJnCNTm and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific Timing Diagrams: Overflow Operation

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

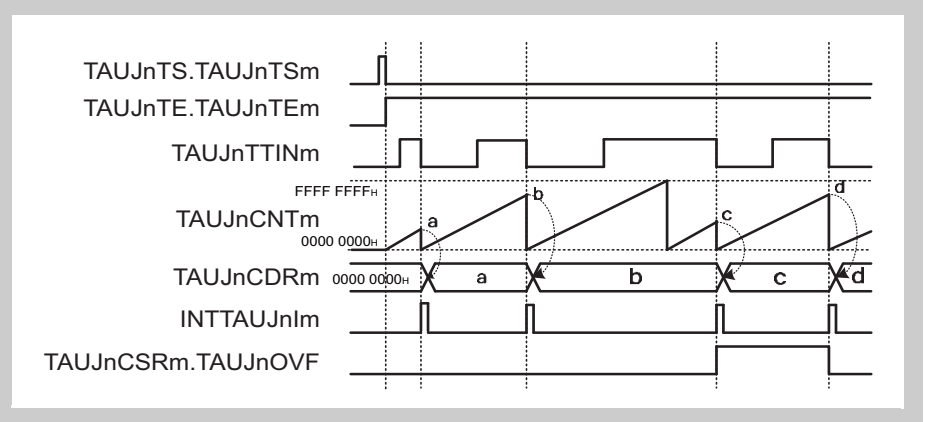


Figure 14-23 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded into TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

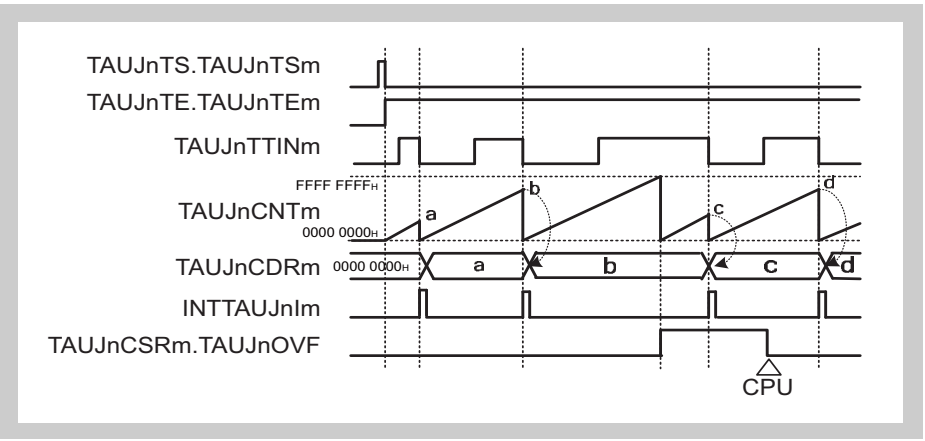


Figure 14-24 TAUJnCMORm.TAUJnCOS[1:0] = 01_B,TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded into TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

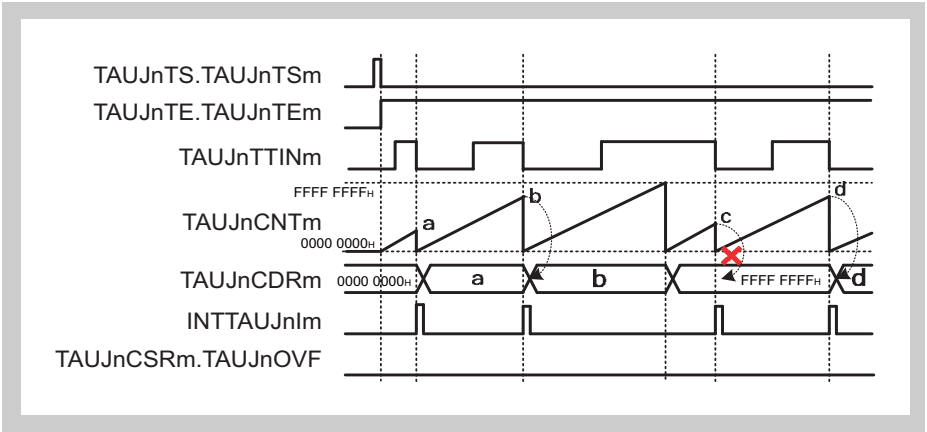


Figure 14-25 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$ and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next valid TAUJnTTINm input edge after an overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

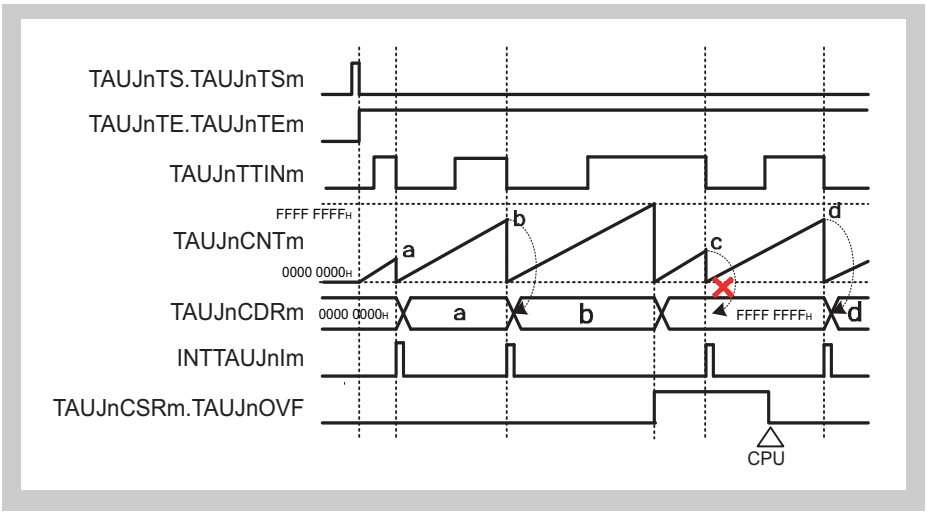


Figure 14-26 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next valid TAUJnTTINm input edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

14.12.4 TAUJnTTINm Input Signal Width Measurement Function

(1) Overview

Summary This function measures the width of a TAUJnTTINm input signal.

Prerequisites
- The operating mode should be set to Capture & One-Count Mode. (See table 14-26, TAUJnCMORm Settings for TAUJnTTINm Input Signal Width Measurement Function.)
- TAUJnTTOUTm is not used with this function.
- TAUJnCMORm.TAUJnMD0 should be set to 0.

Functional description The counter is started by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJnTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm stop edge is detected, the value of TAUJnCNTm is captured and transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJnTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of TAUJnCMORm.TAUJnCOS[1:0] bits.

Table 14-25 Effects of Overflow

TAUJnCMORm. TAUJnCOS[1:0]	When Overflow Occurs		When Valid TAUJnTTINm Input Stop Edge Is Detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm, TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded into TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting. TAUJnCDRm remains unchanged	0
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, overflow bit TAUJnCSRm.TAUJnOVF can be cleared only by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJnTTINm signal. However, if multiple overflows occur before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate their occurrences.

Caution The TAUJnTTINm input signal width measurement function cannot make a forced restart.

Note If TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded into TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input signal width = Count clock cycle × [(TAUJnCSRm.TAUJnOVF × (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

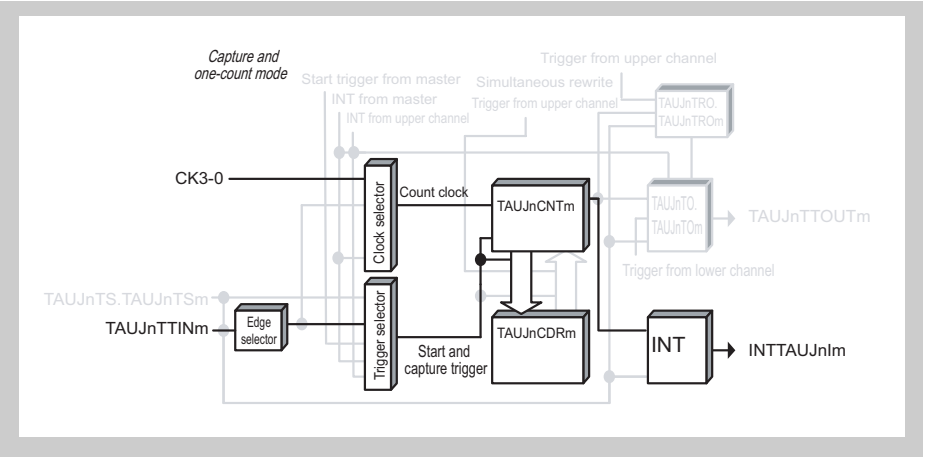


Figure 14-27 Block Diagram of TAUJnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

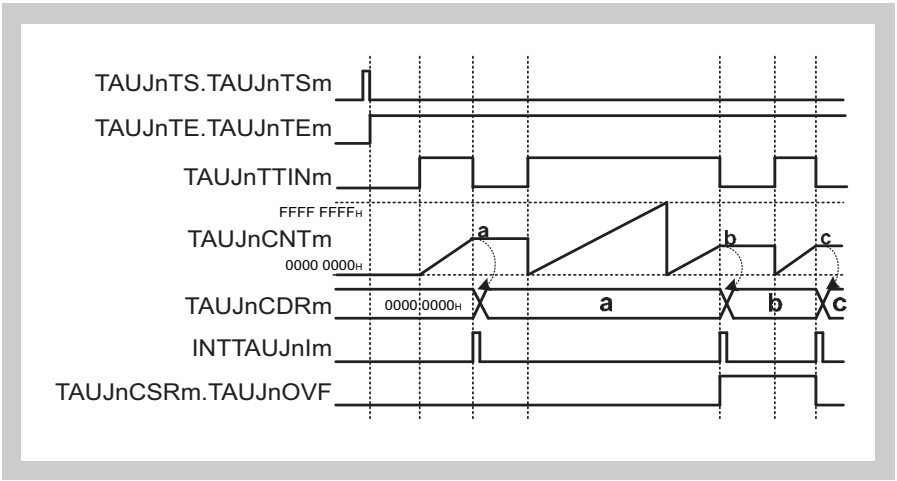


Figure 14-28 General Timing Diagram of TAUJnTTINm Input Signal Width Measurement Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-26 TAUJnCMORm Settings for TAUJnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects the sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock.
TAUJnMAS	0: Unused. Set to 0.
TAUJnSTS[2:0]	010: Valid edge of TAUJnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger
TAUJnCOS[1:0]	See table 14-25, Effects of Overflow
TAUJnMD[4:1]	0110: Capture & one-count mode
TAUJnMD0	0: Disables the start trigger during operation.

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-27 TAUJnCMURm Settings for TAUJnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Signal Width Measurement Function. Therefore, these registers should be set to 0.

Table 14-28 Simultaneous Rewrite Settings for TAUJnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When disabling simultaneous rewrite (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

Table 14-29 Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in table 14-26, TAUJnCMORm Settings for TAUJnTTINm Input Signal Width Measurement Function, and table 14-27, TAUJnCMURm Settings for TAUJnTTINm Input Signal Width Measurement Function. TAUJnCDRm register operates as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm awaits TAUJnTTINm start edge detection. When TAUJnTTINm start edge is detected, TAUJnCNTm starts to count up.
During operation	Detection of TAUJnTTINm edge. TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers are readable at any time. TAUJnCSC.CLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a valid TAUJnTTINm edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm and retains its value. • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops. TAUJnCNTm and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific Timing Diagrams: Overflow Operation

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

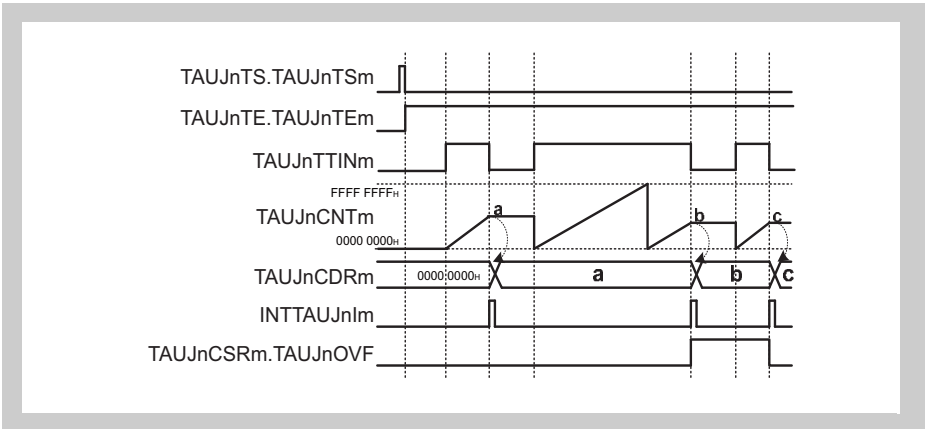


Figure 14-29 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded into TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

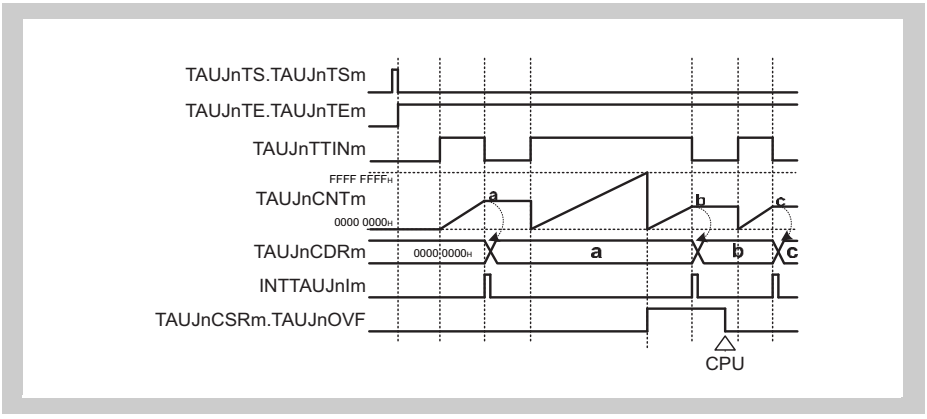


Figure 14-30 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded into TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

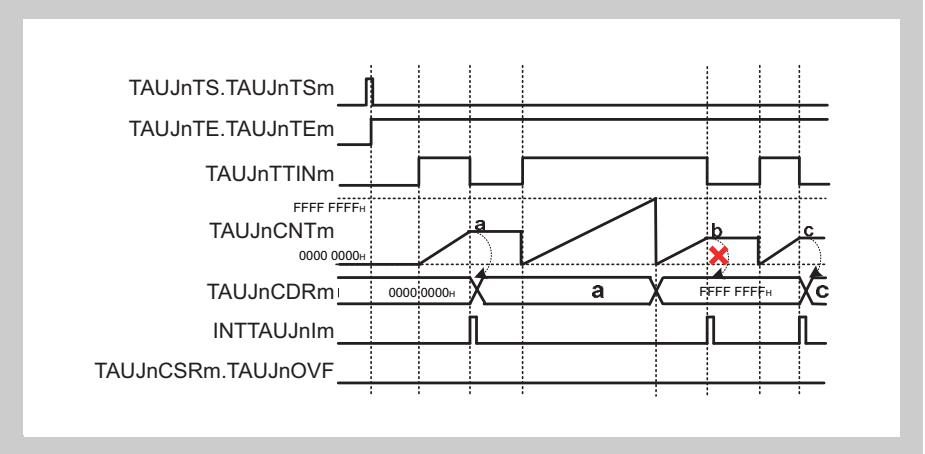


Figure 14-31 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$ and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next valid TAUJnTTINm input edge after the overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

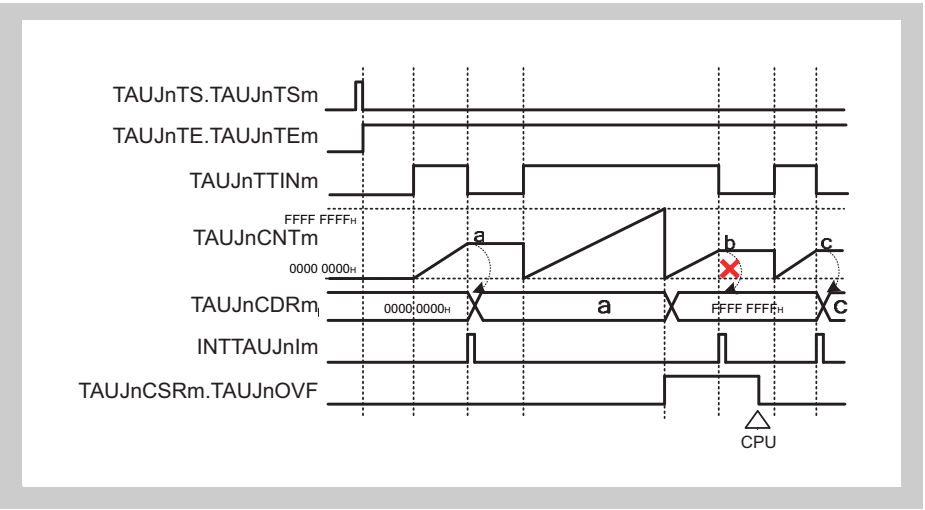


Figure 14-32 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next valid TAUJnTTINm input edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

14.12.5 TAUJnTTINm Input Period Count Detection Function

(1) Overview

Summary This function measures the cumulative width of a TAUJnTTINm input signal.

Prerequisites
- The operating mode should be set to capture & gate count mode. (See table 14-30, TAUJnCMORm Settings for TAUJnTTINm Input Period Count Detection Function.)
- TAUJnTTOUTm is not used with this function.

Functional description The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJnTTINm input edge.

When a valid TAUJnTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJnTTINm input stop edge is detected, the current TAUJnCNTm value is loaded into TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJnTTINm input start edge is detected.

When the next TAUJnTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF $FFFF_H$, the TAUJnCSRm.TAUJnOVF bit is set to 1 and the counter restarts counting from 0000 0000_H. The value of TAUJnCSRm.TAUJnOVF is cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

Note An input signal TAUJnTTINm is sampled at the frequency of the prescaler output set by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions Valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.
- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJnTTINm input high period is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

(2) Equations

Cumulative TAUJnTTINm input width = Count clock cycle × [(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + (TAUJnCDRm capture value + 1)]

(3) Block Diagram and General Timing Diagram

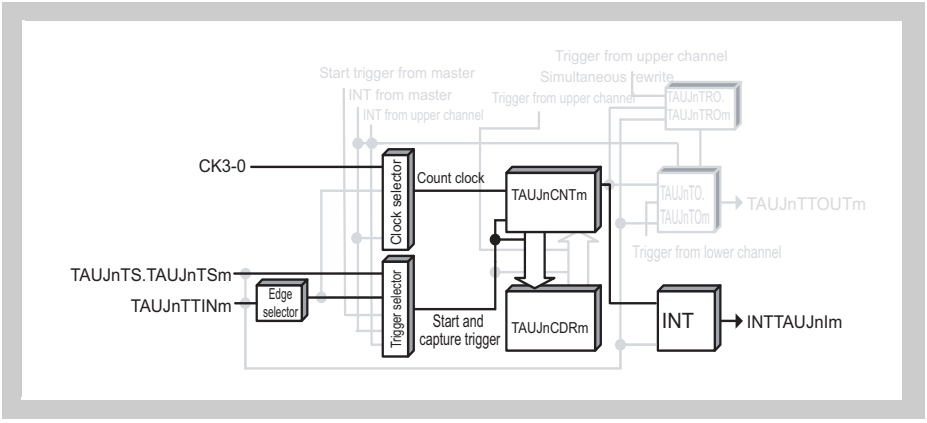


Figure 14-33 Block Diagram of TAUJnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

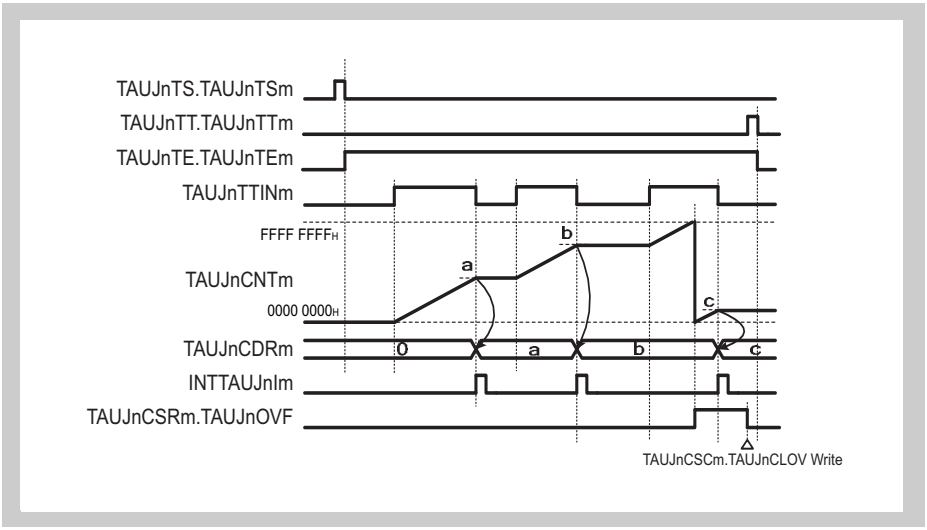


Figure 14-34 General Timing Diagram of TAUJnTTINm Input Period Count Detection Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-30 TAUJnCMORm Settings for TAUJnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects a sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	0: Unused. Set to 0
TAUJnSTS[2:0]	010: Valid edge of the TAUJnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger
TAUJnCOS[1:0]	01: Overflow (TAUJnCSRm.TAUJnOVF) is set upon occurrence of a counter overflow and cleared by setting TAUJnCSCm.TAUJnCLOV to 1
TAUJnMD[4:1]	1101: Capture & gate count mode
TAUJnMD0	0: Disables the start trigger during operation

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-31 TAUJnCMURm Settings for TAUJnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Period Count Detection Function. Therefore, these registers should be set to 0.

Table 14-32 Simultaneous Rewrite Settings for TAUJnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When disabling simultaneous rewrite (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Period Count Detection Function

Table 14-33 Operating Procedure for TAUJnTTINm Input Period Count Detection Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in table 14-30, TAUJnCMORm Settings for TAUJnTTINm Input Period Count Detection Function, and table 14-31, TAUJnCMURm Settings for TAUJnTTINm Input Period Count Detection Function. TAUJnCDRm register operates as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUJnTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm awaits TAUJnTTINm start edge detection. When TAUJnTTINm start edge is detected, TAUJnCNTm is cleared to 0000 0000_H and starts to count up.
During operation	Detection of TAUJnTTINm edge. TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers are readable at any time. TAUJnCSCm.TAUJnCLOV bit can be set to 1.	When a TAUJnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts counting up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers its value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the "value transferred to TAUJnCDRm + 1" value and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. When TAUJnCNTm reaches FFFF $FFFF_H$, the counter overflows and TAUJnCSRm.TAUJnOVF is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops. TAUJnCNTm and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific Timing Diagrams

(a) Operation Stop and Restart

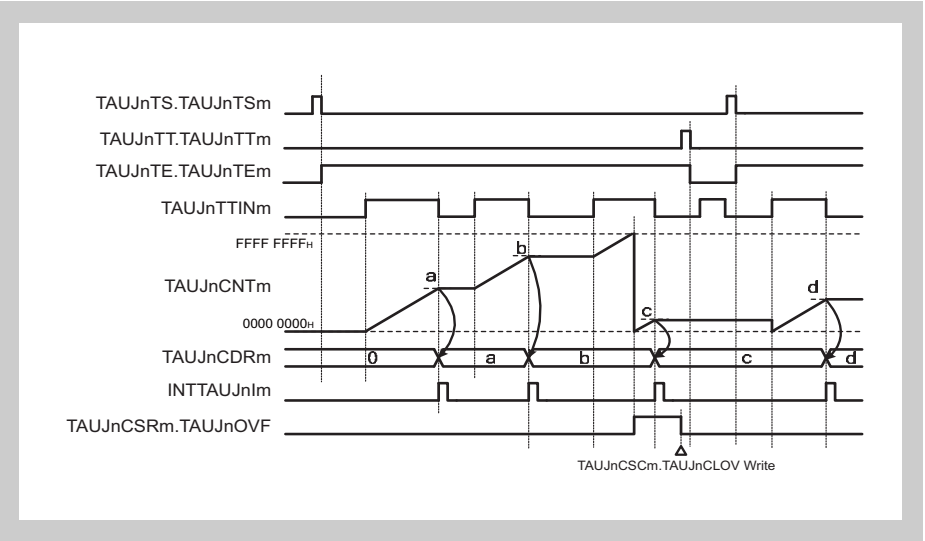


Figure 14-35 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and retains the current value.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts counting from 0000 0000_H.

14.12.6 TAUJnTTINm Input Position Detection Function

(1) Overview

Summary This function measures a time period from count start timing until a TAUJnTTINm input signal.

Prerequisites
- The operating mode should be set to count capture mode. (See table 14-34, TAUJnCMORm Settings for TAUJnTTINm Input Position Detection Function.)
- TAUJnTTOUTm is not used with this function.

Functional description The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 $0000_{H.}$ When a valid TAUJnTTINm input edge is detected, the current TAUJnCNTm value is loaded into TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The count operation continues.

When the counter reaches FFFF $FFFF_H$, the TAUJnCSRm.TAUJnOVF bit is set to 1 and the counter restarts from 0000 0000_H. The value of TAUJnCSRm.TAUJnOVF is cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see Section 14.10, TAUJnTTOUTm Output and INTTAUJnIm Generation When Counter Starts or Restarts.

(2) Equations

Functional duration at a TAUJnTTINm input pulse =

Count clock cycle × [(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + (TAUJnCDRm capture value + 1)]

(3) Block Diagram and General Timing Diagram

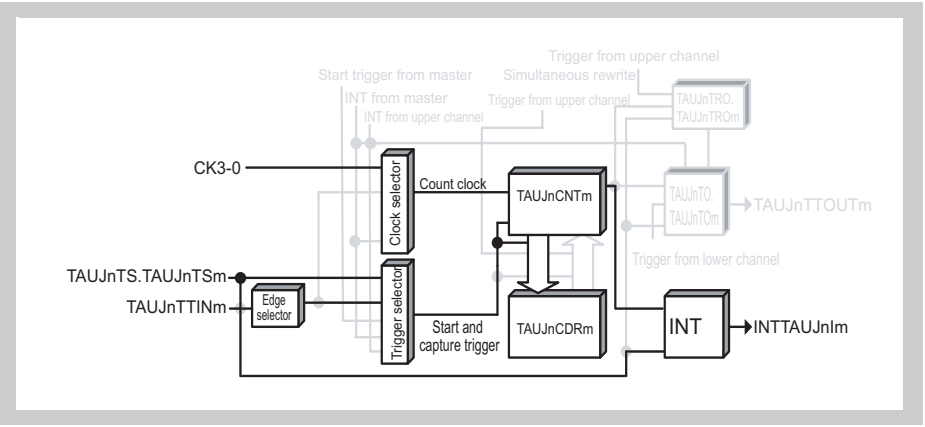


Figure 14-36 Block Diagram of TAUJnTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated at the beginning of operation. (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

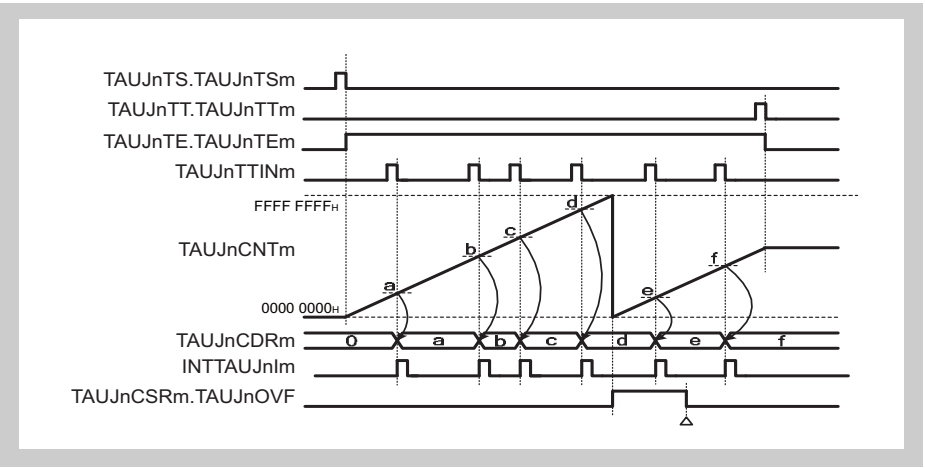


Figure 14-37 General Timing Diagram of TAUJnTTINm Input Position Detection Function

(4) Register Settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-34 TAUJnCMORm Settings for TAUJnTTINm Input Position Detection Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects a sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	0: Unused. Set to 0
TAUJnSTS[2:0]	001: Valid edge of the TAUJnTTINm input signal is used as an external capture trigger
TAUJnCOS[1:0]	01:Overflow (TAUJnCSRm.TAUJnOVF) is set upon occurrence of a counter overflow and cleared by setting TAUJnCSCm.TAUJnCLOV to 1
TAUJnMD[4:1]	1011: Count capture mode
TAUJnMD0	0: INTTAUJnIm is not generated at the beginning of operation 1: INTTAUJnIm is generated at the beginning of operation

(b) TAUJnCMURm

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-35 TAUJnCMURm Settings for TAUJnTTINm Input Position Detection Function

Bit Name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel Output Mode

This function does not use any channel output mode. However, channel output mode is available in independent channel output mode controlled by software. For details, see Section 14.8, Channel Output Modes.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Position Detection Function. Therefore, these registers should be set to 0.

Table 14-36 Simultaneous Rewrite Settings for TAUJnTTINm Input Position Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0:Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When disabling simultaneous rewrite (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Position Detection Function

Table 14-37 Operating Procedure for TAUJnTTINm Input Position Detection Function

	Operation	TAUJn Status
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in table 14-34, TAUJnCMORm Settings for TAUJnTTINm Input Position Detection Function, and table 14-35, TAUJnCMURm Settings for TAUJnTTINm Input Position Detection Function. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter is started. When TAUJnCMORm.TAUJnMD0 is set to 1, INTTAUJnlm is generated.
During operation	The TAUJnCMURm.TAUJnTIS[1:0] bits are changeable at any time. The TAUJnCDRm and TAUJnCSRm registers are readable at any time. TAUJnCSC.CLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a valid TAUJnTTINm edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm. • INTTAUJnlm is generated. • The counter value is not cleared to 0000 0000_H and TAUJnCNTm continues to count. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops. TAUJnCNTm and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific Timing Diagrams

(a) Operation Stop and Restart

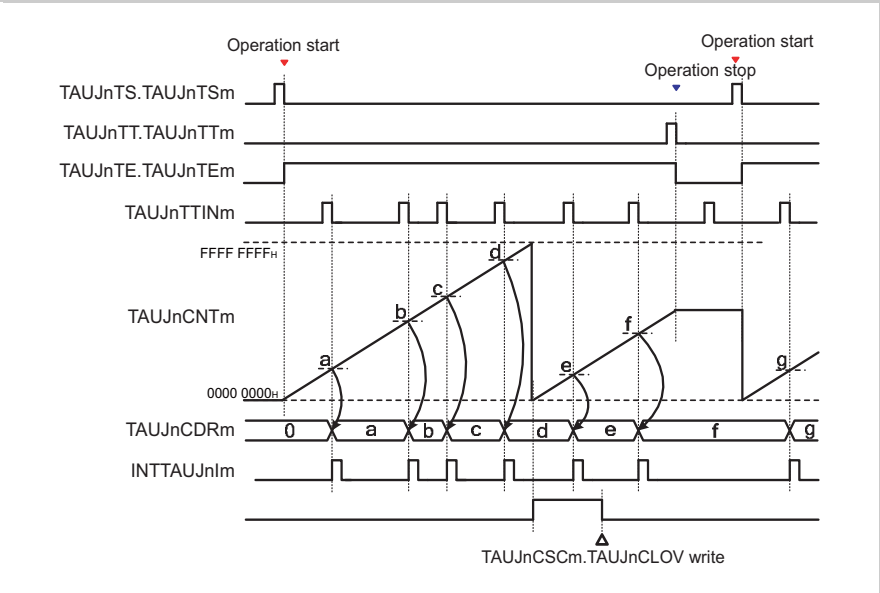


Figure 14-38 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and retains the current value.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts counting from 0000 0000_H.

14.13 Synchronous Channel Functions

This section describes about the PWM output function which generates PWM signals at a regular interval. For the overview of the synchronous channel operation, see Section 14.3, Functional Description.

14.13.1 PWM Output Functions

(1) Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJnTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for master channels should be set to interval timer mode. (See table 14-38, TAUJnCMORm Settings for Master Channels of PWM Output Function.)
- The operating mode for slave channels should be set to one-count mode. (See table 14-41, TAUJnCMORm Settings for Slave Channels of PWM Output Function.)
- TAUJnTTOUTm is not used with the master channels of this function.
- The channel output mode for slave channels should be set to synchronous channel output mode 1 (See table 14-8, Channel Output Modes).

Functional description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded into TAUJnCNTm and the counters start to count down from these values. PWM output is implemented by generating INTTAUJnIm on the master channel and setting/resetting TAUJnTTOUTm (slave).

- Master channel

  When the counter of the master channel reaches 0000 0000_H and pulse cycle time has elapsed, INTTAUJnIm is generated. The counter reloads the TAUJnCDRm value and counts down.

- Slave channel

  The INTTAUJnIm on master channel triggers the counter operation on slave channels. The current value of TAUJnCDRm (slave) is loaded into TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJnTTOUTm signal is set to the active level.

  When the counter reaches 0000 0000_H, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJnTTOUTm signal is set to the inactive level. The counter returns to FFFF $FFFF_H$ and awaits the next INTTAUJnIm (the start of the next pulse cycle) of the master channel.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channels, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See Section 14.7, Simultaneous Rewrite.

(2) Equation

Pulse cycle = (TAUJnCDRm (master) + 1) x count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave) / (TAUJnCDRm (master) + 1) x 100

– Duty cycle = 0 %

TAUJnCDRm (slave) = 0000 0000_H

– Duty cycle = 100 %

TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

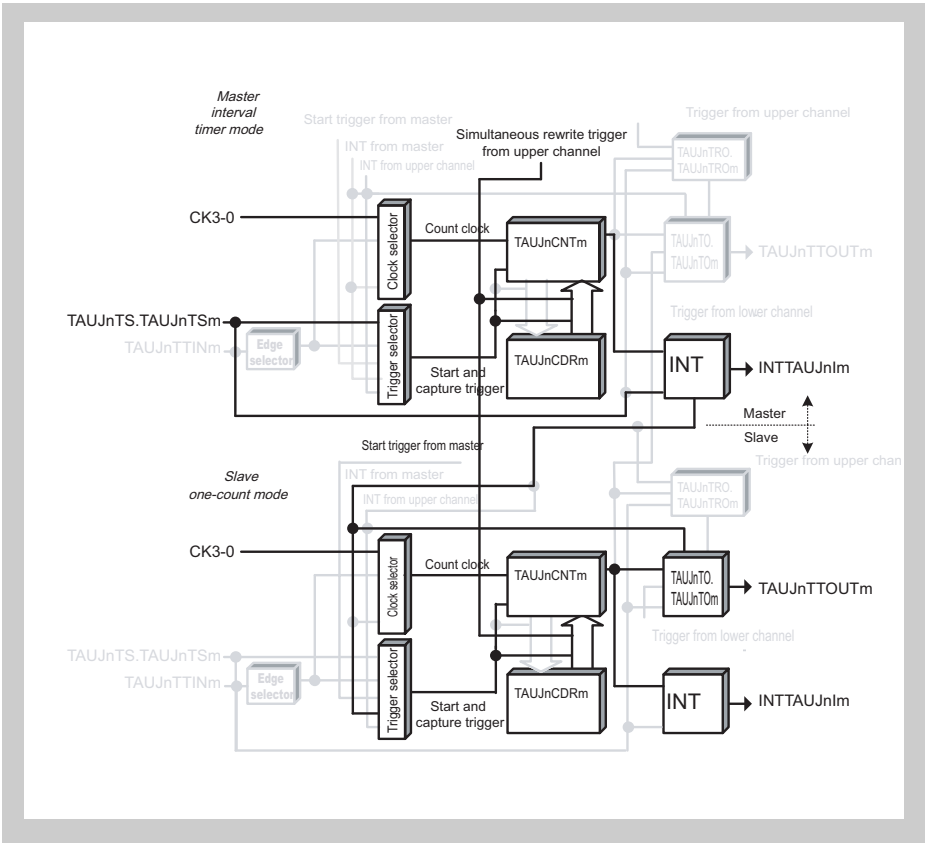


Figure 14-39 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

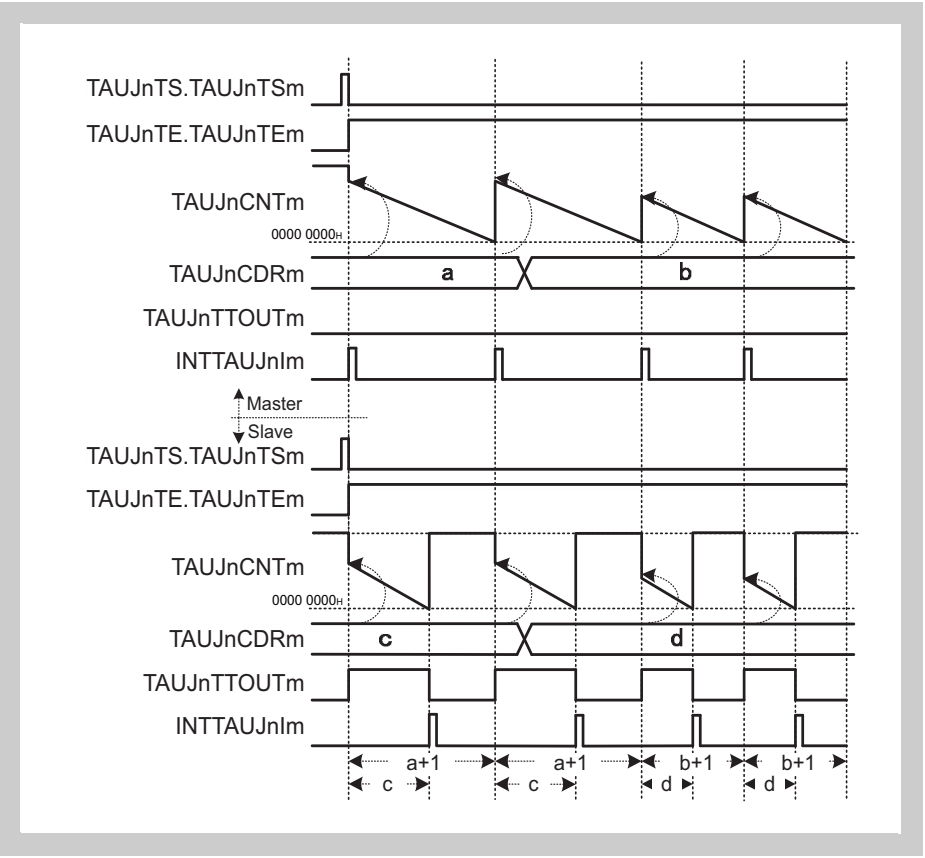


Figure 14-40 General Timing Diagram of PWM Output Function

Note The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm, whereas for the master channel the interval is the corresponding TAUJnCDRm + 1.

(4) Register Settings for Master Channels

(a) TAUJnCMORm for Master Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-38 TAUJnCMORm Settings for Master Channels of PWM Output Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects a sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUJnCKS[1:0] bits of master and slave channels should have the same value
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	1: Channel is a master channel.
TAUJnSTS[2:0]	000: Triggers the counter by software
TAUJnCOS[1:0]	00: Unused. Set to 00
TAUJnMD[4:1]	0000: Interval timer mode
TAUJnMD0	1: INTTAUJnIm is generated at the beginning of operation

(b) TAUJnCMURm for Master Channels of PWM Output

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-39 TAUJnCMURm Settings for Master Channels of PWM Output Function

Bit Name	Setting
TAUJnTIS[1:0]	00: Unused. Set to 00

(c) Channel output mode for master channels

The channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(d) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 14-40 Simultaneous Rewrite Settings for Master Channels of PWM Output Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts counting

(5) Register Settings for Slave Channels

(a) TAUJnCMORm for Slave Channels

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		-	TAUJnMD[4:1]				TAUJnMD0

Table 14-41 TAUJnCMORm Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUJnCKS[1:0]	Selects a sampling clock 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 TAUJnCKS[1:0] bits of master and slave channels should have the same value
TAUJnCCS[1:0]	00: Sampling clock is used as a count clock
TAUJnMAS	0: Channel is a slave channel
TAUJnSTS[2:0]	100: INTTAUJnIm of master channel is a start trigger
TAUJnCOS[1:0]	00: Unused. Set to 00
TAUJnMD[4:1]	0100: One-Count Mode
TAUJnMD0	1: Start trigger is valid during operation

(b) TAUJnCMURm for Slave Channels

7	6	5	4	3	2	1	0
-						TAUJnTIS[1:0]	

Table 14-42 TAUJnCMURm for Slave Channels

Bit Name	Setting
TAUJnTIS[1:0]	00: Unused. Set to 00

(c) Channel output mode for slave channels

Table 14-43 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	1: Enables independent channel output mode
TAUJnTO.TAUJnTOm	0: Low level 1: High level
TAUJnTOM.TAUJnTOMm	1: Synchronous channel operation
TAUJnTOC.TAUJnTOCm	0: Operating mode 1
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Inverted logic

(d) Simultaneous rewrite for slave channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 14-44 Simultaneous Rewrite Settings for Slave Channels of the PWM Output Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts counting

(6) Operating Procedure for PWM Output Function

Table 14-45 Operating Procedure for PWM Output Function

	Operation	TAUJn Status
Initial channel setting	Master channels: Set the TAUJnCMORm/ TAUJnCMURm register and channel output mode as described in section 14.13.1 (4) Register Settings for Master Channels. Slave channels: Set TAUJnCMORm/ TAUJnCMURm registers and channel output mode as described in section 14.13.1 (5) Register Settings for Slave Channels. Set the value of TAUJnCDRm register of every channel.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of both master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of master and slave channels are started. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set.
During operation	TAUJnCDRm is changeable at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm is readable at any time. TAUJnRDT.TAUJnRDTm is changeable during operation.	TAUJnCNTm of master channel loads TAUJnCDRm value and counts down. When the counter reaches 0000 0000_H, • INTTAUJnIm (master) occurs. • TAUJnCDRm value is loaded into TAUJnCNTm (master) to continue count operation. • TAUJnCDRm value is loaded into TAUJnCNTm (slave) to start counting down. • TAUJnTTOUTm (slave) is set to the active level. If TAUJnCNTm (slave) reaches 0000 0000_H, • INTTAUJnIm (slave) occurs. • TAUJnTTOUTm (slave) is set to the inactive level.
Stop operation	Set TAUJnTT.TAUJnTTm of master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, which is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counters stop. TAUJnCNTm and TAUJnTTOUTm stop, and retain their current values.

Restart

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

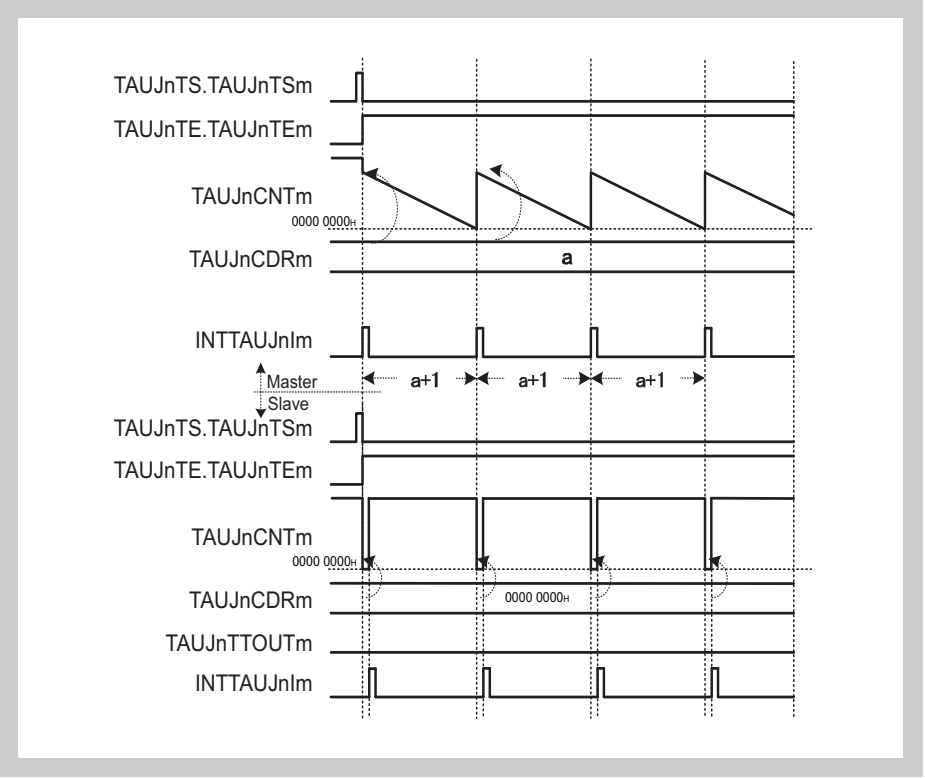


Figure 14-41 TAUJnCDRm (Slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded into TAUJnCNTm (slave). Therefore, TAUJnCNTm (slave) cannot start to count and TAUJnTTOUTm remains inactive.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(b) Duty cycle = 100%

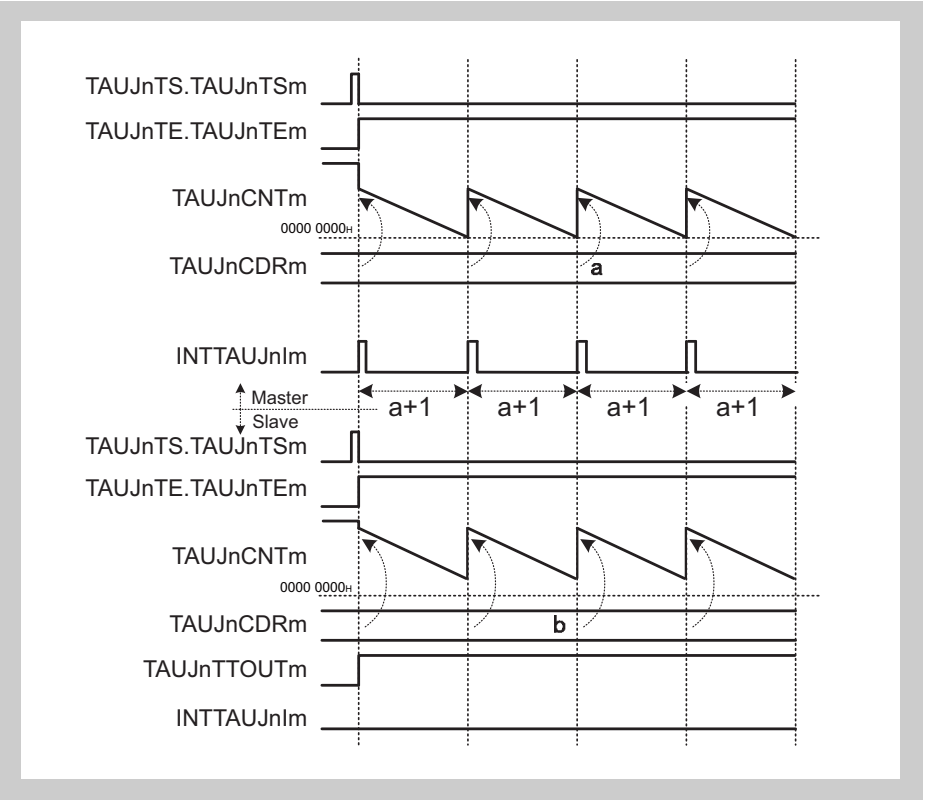


Figure 14-42 TAUJnCDRm (Slave) ≥ TAUJnCDRm (Master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000_H. TAUJnTTOUTm remains active.

(c) Operation stop and restart

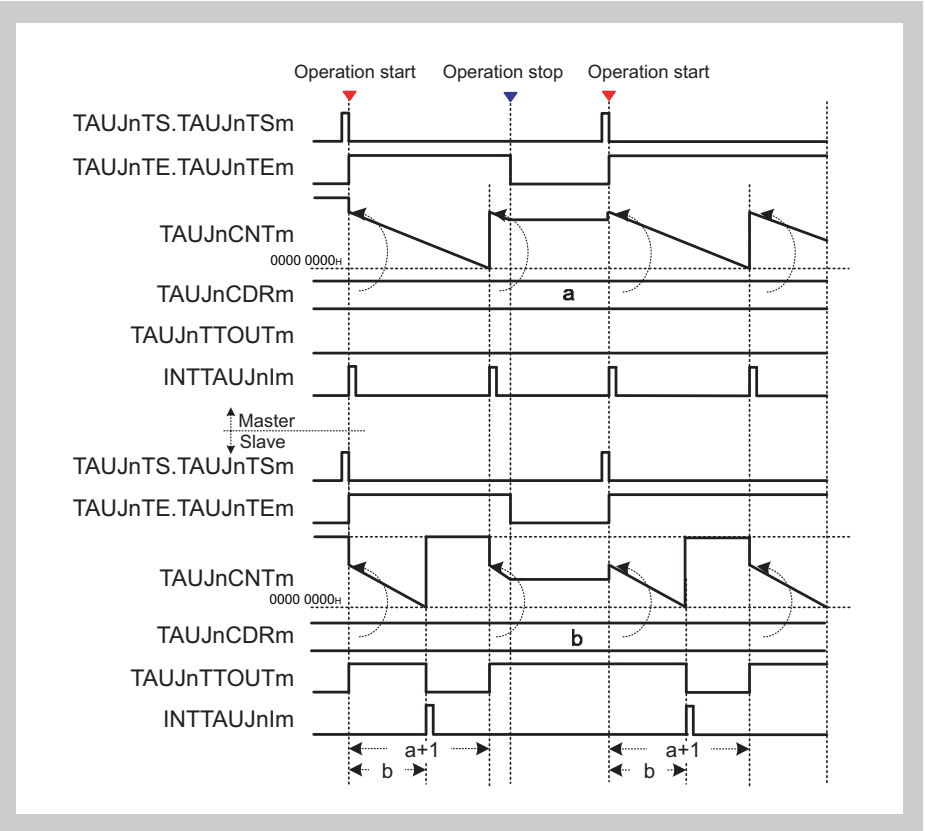


Figure 14-43 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

14.14 Registers

This section describes all the registers of the 32-bit TAUJ.

14.14.1 Overview of TAUJn Registers

The TAUJn is controlled and operated by the registers in the following table.

Table 14-46 Overview of TAUJn Registers

Register Name	Abbreviation	Address
TAUJn prescaler registers		
TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base0> + 90_H
TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base0> + 94_H
TAUJn control registers		
TAUJn channel data register m	TAUJnCDRm	<TAUJn_base1> + m × 4_H
TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base1> + 10_H + m × 4_H
TAUJn channel mode OS register m	TAUJnCMORm	<TAUJn_base0> + 80_H + m × 4_H
TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base1> + 20_H + m × 4_H
TAUJn channel status register m	TAUJnCSRm	<TAUJn_base1> + 30_H + m × 4_H
TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base1> + 40_H + m × 4_H
TAUJn channel start trigger register	TAUJnTS	<TAUJn_base1> + 54_H
TAUJn channel enable status register	TAUJnTE	<TAUJn_base1> + 50_H
TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base1> + 58_H
TAUJn output registers		
TAUJn channel output enable register	TAUJnTOE	<TAUJn_base1> + 60_H
TAUJn channel output register	TAUJnTO	<TAUJn_base1> + $5C_H$
TAUJn channel output mode register	TAUJnTOM	<TAUJn_base0> + 98_H
TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base0> + $9C_H$
TAUJn channel output active level register	TAUJnTOL	<TAUJn_base1> + 64_H
TAUJn reload data registers		
TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base0> + $A0_H$
TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base0> + $A4_H$
TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base1> + 68_H
TAUJn channel reload status register	TAUJnRSF	<TAUJn_base1> + $6C_H$

Note TAUJn base addresses <TAUJn_base0> and <TAUJn_base1> are defined in the first part of this section "Register Addresses".

Caution If a specified value is described for each register bit, any other value than the specified one should not be written into the corresponding bit.

14.14.2 Details of TAUJn Prescaler Registers

(1) TAUJnTPS - TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access Readable/writable in 16-bit units.

Address <TAUJn_base0> + 90_H

Initial value $FFFF_H$

Any reset source triggers initialization.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-47 Description of TAUJnTPS Register (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUJnPRS3 [3:0]	Specifies a CK3_PRE clock. The CK3_PRE clock is an input clock of the BRG unit which supplies prescaler output CK3 to all channels.

TAUJnPRS3[3:0]	CK3_PRE clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm= 0).

Table 14-47 Description of TAUJnTPS Register (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUJnPRS2 [3:0]</td><td>Specifies prescaler output CK2.
<table>
<tr><th>TAUJnPRS2[3:0]</th><th>Prescaler Output CK2</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUJnPRS1 [3:0]</td><td>Specifies prescaler output CK1.
<table>
<tr><th>TAUJnPRS1[3:0]</th><th>Prescaler Output CK1</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

Table 14-47 Description of TAUJnTPS Register (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUJnPRS0 [3:0]</td><td>Specifies prescaler output CK0.
<table>
<tr><th>TAUJnPRS0[3:0]</th><th>Prescaler Output CK0</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

(2) TAUJnBRS - TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler output CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS.PRS3[3:0].

Access Readable/writable in 8-bit units.

Address <TAUJn_base0> + 94_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
TAUJnBRS[07:00]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-48 Description of TAUJnBRS Register

Bit Position	Bit Name	Function
7 to 0	TAUJnBRS [07:00]	Specifies a CK3_PRE clock division factor for generating prescaler output CK3.

TAUJnBRS[07:00]	Prescaler Output CK3
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

14.14.3 Details of TAUJn Control Registers

(1) TAUJnCDRm - TAUJn Channel Data Register m

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

Access Readable/writable in 32-bit units.

- Readable in capture mode. Any write operation is ignored.
- Readable/writable in compare mode.

Address <TAUJn_base1> + m × 4_H

Initial value 0000 0000_H

Any reset source triggers initialization.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCDR[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCDR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-49 Description of TAUJnCDRm Register

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

(2) TAUJnCNTm - TAUJn Channel Counter Register m

This is a channel m counter register.

Access Readable in 32-bit units.

Address <TAUJn_base1> + 10_H + m × 4_H

Initial value 0000 0000 $_H$ or FFFF $FFFF_H$

The initial value depends on the operating mode. See table 14-51, TAUJnCNTm Read Values after Counter is Re-enabled.
Any reset source triggers initialization.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCNT[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14-50 Description of TAUJnCNTm Register

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT [31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 14-51 TAUJnCNTm Read Values after Counter is Re-enabled

Mode Name	Count Method (Up/Down)	TAUJnCNTm Value		
		After Changing Operating Mode after Reset	After Stop Trigger	After One Count
Interval timer mode	Count-down	FFFF FFFF$_H$	Stop value	-
Capture mode	Count-up	0000 0000$_H$	Stop value	-
One-count mode	Count-down	FFFF FFFF$_H$	Stop value	FFFF FFFF$_H$
Capture and one-count mode	Count-up	0000 0000$_H$	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count-up	0000 0000$_H$	Stop value	-
Gate count mode	Count-down	FFFF FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count-up	0000 0000$_H$	Stop value	Stop value

Note If operating mode is changed while the counter is stopped, the initial counter value after a counter restart becomes undefined. Operating mode can be changed by the TAUJnCMORm.TAUJnMD[4:0] register.

(3) TAUJnCMORm - TAUJn Channel Mode OS Register m

This register controls channel m operation.

Access Readable/writable in 16-bit units. Writable only when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base0> + 80_H + m × 4_H

Initial value 0000_H

Any reset source triggers initialization.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCKS [1:0]		TAUJnCCS [1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS [1:0]		-	TAUJnMD[4:0]				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 14-52 Description of TAUJnCMORm Register (1/4)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS [1:0]	Selects a sampling clock, which is used with the TAUJnTTINm input edge detection circuit. Setting of TAUJnCMORm.TAUJnCCS[1:0] bit also allows the sampling clock to serve as a count clock.

TAUJn CKS1	TAUJn CKS0	Selection of Sampling Clock
0	0	Prescaler output CK0
0	1	Prescaler output CK1
1	0	Prescaler output CK2
1	1	Prescaler output CK3

Bit Position	Bit Name	Function
13, 12	TAUJnCCS [1:0]	Selects a count clock for TAUJnCNTm counter.

TAUJn CCS1	TAUJn CCS0	Selection of Count Clock
0	0	Sampling clock specified by TAUJnCMORm.TAUJnCKS[1:0].
0	1	Valid edge of TAUJnTTINm input signal
1	0	Setting prohibited
1	1	

Bit Position	Bit Name	Function
11	TAUJnMAS	Specifies whether the channel is a master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels are fixed to 0.

Table 14-52 Description of TAUJnCMORm Register (2/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUJnSTS [2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUJn STS2</th><th>TAUJn STS1</th><th>TAUJn STS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJnTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJnTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUJnlm of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
</table>

Table 14-52 Description of TAUJnCMORm Register (3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TAUJnCOS [1:0]</td><td>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m.
These bits are valid only when channel m is in capture mode.
<table>
<tr><th>TAUJn COS1</th><th>TAUJn COS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJnTTINm input is detected.</td><td>Updated (cleared or set) when valid edge of TAUJnTTINm input is detected:
• Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.
• Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJnTTINm input is detected and when a counter overflow occurs.
• Detection of valid edge of TAUJnTTINm input: The counter value is written into TAUJnCDRm.
• Occurrence of overflow: FFFF FFFF$_H$ is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJnTTINm is ignored.</td><td>No setting</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
</table></td></tr>
</table>

Table 14-52 Description of TAUJnCMORm Register (4/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUJnMD [4:0]</td><td>Specifies operating mode.
<table>
<tr><th>TAUJn MD4</th><th>TAUJn MD3</th><th>TAUJn MD2</th><th>TAUJn MD1</th><th>TAUJn MD0</th><th>Functional description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td rowspan="4">1/0</td><td rowspan="4">Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr>
</table>

<table>
<tr><th>Mode</th><th>Role of TAUJnMD0 Bit</th></tr>
<tr><td>Interval timer mode
Capture mode
Count capture mode</td><td>Specifies whether an INTTAUJnlm signal is generated or not at the beginning of count operation (at the input of start trigger).
0: INTTAUJnlm is not generated.
1: INTTAUJnlm is generated.</td></tr>
<tr><td>One-count mode
Gate count mode</td><td>Enables/disables detection of a start trigger during counting.
0: Disable
1: Enable</td></tr>
<tr><td>Capture and one-count mode
Capture and gate count mode</td><td>This bit should be set to 0.</td></tr>
</table>
</td></tr>
</table>

(4) TAUJnCMURm - TAUJn Channel Mode User Register m

This register specifies a type of valid edge detection used for TAUJnTTINm input.

Access Readable/writable in 8-bit units.

Address <TAUJn_base1> + 20_H + m × 4_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUJnTIS[1:0]	
R	R	R	R	R	R	R/W	R/W

Table 14-53 Description of TAUJnCMURm Register

Bit Position	Bit Name	Function
1, 0	TAUJnTIS [1:0]	Specifies a valid edge of TAUJnTTINm input signal. • Edge detection of TAUJnTTINm input signal is based on the sampling clock selected by TAUJnCMORm.TAUJnCKS[1:0].

TAUJn TIS1	TAUJn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

(5) TAUJnCSRm - TAUJn Channel Status Register m

This register indicates the overflow status of channel m.

Access Readable in 8-bit units.

Address <TAUJn_base1> + 30_H + m × 4_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TAUJnOVF
R	R	R	R	R	R	R	R

Table 14-54 Description of TAUJnCSRm Register

Bit Position	Bit Name	Function
0	TAUJn OVF	Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: • Capture mode • Capture and one-count mode • Count capture mode • Capture and gate count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

(6) TAUJnCSCm - TAUJn Channel Status Clear Trigger Register m

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access Writable in 8-bit units. The read value is always 00_H.

Address <TAUJn_base1> + 40_H + m × 4_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	0	TAUJnCLOV
R	R	R	R	R	R	R	W

Table 14-55 Description of TAUJnCSCm Register

Bit Position	Bit Name	Function
0	TAUJnCLOV	Controls a clear operation of the overflow flag (TAUJnCSRm.TAUJnOVF) of channel m. 0: Invalid (Setting this bit to 0 does not affect the overflow flag TAUJnCSRm.TAUJnOVF bit.) 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

(7) TAUJnTS - TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access Writable in 8-bit units. The read value is always 00_H.

Address <TAUJn_base1> + 54_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTS 03	TAUJnTS 02	TAUJnTS 01	TAUJnTS 00
W	W	W	W	W	W	W	W

Table 14-56 Description of TAUJnTS Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: Invalid (Setting these bits to 0 does not enable the counter operation of channel m.) 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1 Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1 Whether the counter is started or not depends on the selected of operating mode.

(8) TAUJnTE - TAUJn Channel Enable Status Register

This register enables/disables a counter operation.

Access Readable in 8-bit units.

Address <TAUJn_base1> + 50_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTE 03	TAUJnTE 02	TAUJnTE 01	TAUJnTE 00
R	R	R	R	R	R	R	R

Table 14-57 Description of TAUJnTE Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTEm	Enables/disables channel m's counter operation. 0: Disables the counter operation 1: Enables the counter operation This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is set to 0 when TAUJnTT.TAUJnTTm is set to 1.

(9) TAUJnTT - TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access Writable in 8-bit units. The read value is always 00_H.

Address <TAUJn_base1> + 58_H

Initial value 00_H

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTT 03	TAUJnTT 02	TAUJnTT 01	TAUJnTT 00
W	W	W	W	W	W	W	W

Table 14-58 Description of TAUJnTT Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTTm	Stops channel m's counter operation. 0: Invalid (Setting these bits to 0 does not stop the counter operation of channel m.) 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJnTTOUTm retain the values provided before the counter is stopped.

14.14.4 Details of TAUJn Output Registers

(1) TAUJnTOE - TAUJn Channel Output Enable Register

This register enables/disables independent channel output mode controlled by software.

Access Readable/writable in 8-bit units. Writable only while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base1> + 60_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOE 03	TAUJnTOE 02	TAUJnTOE 01	TAUJnTOE 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-59 Description of TAUJnTOE Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTOEm	Enables/disables the independent timer output function: 0: Disables the independent timer output function (controlled by software) 1: Enables the independent timer output function

Note The output from TAUJnTTOUTm pin is controlled by TAUJnTO.TAUJnTOm (controlled by software).

(2) TAUJnTOM - TAUJn Channel Output Mode Register

This register specifies output mode of each channel.

Access Readable/writable in 8-bit units. Writable only while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base0> + 98_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOM 03	TAUJnTOM 02	TAUJnTOM 01	TAUJnTOM 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-60 Description of TAUJnTOM Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTOMm	Specifies output mode. 0: Independent channel operation 1: Synchronous channel operation As described in table 14-8, Channel Output Modes, output mode specification depends on the setting of output control bit of each channel.

(3) TAUJnTOC - TAUJn Channel Output Configuration Register

This register specifies output mode of each channel in combination with TAUJnTOMm.

Access Readable/writable in 8-bit units. Writable only while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base0> + $9C_H$

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-61 Description of TAUJnTOC Register

Bit Position	Bit Name	Function
3-0	TAUJn TOCm	Specifies output mode: 0: Operating mode 1 (Toggle mode) 1: Setting prohibited

14.14.5 Details of TAUJn Channel Output Level Registers

(1) TAUJnTO - TAUJn Channel Output Register

This register specifies and reads a TAUJnTTOUTm level.

Access Readable/writable in 8-bit units.

Address <TAUJn_base1> + $5C_H$

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTO 03	TAUJnTO 02	TAUJnTO 01	TAUJnTO 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-62 Description of TAUJnTO Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTOm	Specifies and reads a TAUJnTTOUTm level. 0: Low level 1: High level The TAUJnTOm bit is writable when TAUJnTOE.TAUJnTOEm = 0.

(2) TAUJnTOL - TAUJn Channel Output Active Level Register

This register specifies an output logic for the channel output bit (TAUJnTO.TAUJnTOm).

Access Readable/writable in 8-bit units.

Address <TAUJn_base1> + 64_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOL 03	TAUJnTOL 02	TAUJnTOL 01	TAUJnTOL 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-63 Description of TAUJnTOL Register

Bit Position	Bit Name	Function
3 to 0	TAUJnTOLm	Specifies the output logic of channel m output bit (TAUJnTO.TAUJnTOm). 0: Positive logic (active high) 1: Inverted logic (active low)

14.14.6 Details of TAUJn Reload Data Registers

(1) TAUJnRDE - TAUJn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of the data register TAUJnCDRm/TAUJnTOLm.

Access Readable/writable in 8-bit units. Writable only while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base0> + $A0_H$

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDE 03	TAUJnRDE 02	TAUJnRDE 01	TAUJnRDE 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-64 Description of TAUJnRDE Register

Bit Position	Bit Name	Function
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enables simultaneous rewrite

(2) TAUJnRDM - TAUJn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access Readable/writable in 8-bit units. Writable only while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base0> + $A4_H$

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDM 03	TAUJnRDM 02	TAUJnRDM 01	TAUJnRDM 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-65 Description of TAUJnRDM Register

Bit Position	Bit Name	Function
3 to 0	TAUJnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts counting 1: Setting prohibited These bit settings are applied only when TAUJnRDE.TAUJnRDEm = 1.

(3) TAUJnRDT - TAUJn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite pending state.

Access Writable in 8-bit units. The read value is always 00_H.

Address <TAUJn_base1> + 68_H

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDT 03	TAUJnRDT 02	TAUJnRDT 01	TAUJnRDT 00
W	W	W	W	W	W	W	W

Table 14-66 Description of TAUJnRDT Register

Bit Position	Bit Name	Function
3 to 0	TAUJnRDTm	Triggers a simultaneous rewrite pending state: 0: Invalid (Setting these bits to 0 does not trigger a simultaneous rewrite pending state.) 1: Triggers a simultaneous rewrite pending state. The simultaneous rewrite pending flag (TAUJnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger.

(4) TAUJnRSF - TAUJn Channel Reload Status Register

This register indicates the simultaneous rewrite status.

Access Readable in 8-bit units.

Address <TAUJn_base1> + $6C_H$

Initial value 00_H

Any reset source triggers initialization.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRSF 03	TAUJnRSF 02	TAUJnRSF 01	TAUJnRSF 00
R	R	R	R	R	R	R	R

Table 14-67 Description of TAUJnRSF Register

Bit Position	Bit Name	Function
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status: 0: Indicates that simultaneous rewrite has been made due to occurrence of a simultaneous rewrite trigger 1: Indicates that TAUJ is in the simultaneous rewrite pending status (TAUJnRDTm = 1)

Section 15 TSG2 (TSG20, TSG21)

15.1 Functions of TSG2n

Channels This product provides two channels of TSG2n.

Table 15-1 TSG2 Channels

TSG2	
Number of Channels	2
Name	TSG20, TSG21

Meaning of n Throughout this section, the TSG2 channels are identified by suffix n (n = 0 or 1). For example, n is used as in TSG2n control register (TSnCTL0).

Register addresses TSG2n register addresses are given as address offsets from the individual base address <TSG2n_base0> or <TSG2n_base1>. Table 15-2 shows the base addresses of TSG2n.

Table 15-2 TSG2 Register Base Addresses

TSG2n	<TSG2n_base0> Address	<TSG2n_base1> Address
TSG20	FF82 E000$_H$	FFFF CC00$_H$
TSG21	FF82 F000$_H$	FFFF D000$_H$

Clock supply TSG2n is connected to PCLK and is supplied with the PCLK clock signal input.

Table 15-3 TSG2 Clock Supply

TSG2n	Supplied Clock
TSG20, TSG21	PCLK

I/O signals The I/O signals of the TSG2 are listed in Table 15-4.

Table 15-4 List of TSG2 I/O Signals

TSG2n Signal	Function	Connected to
TSG2nCLKI*	Clock enable input	Port
TSG2nPTSI0 to TSG2nPTSI2	External pattern input	Port
TSG2nO0 to TSG2nO7	Timer output	Port
TSnOPCI0, TSnOPCI1	Trigger input signal	Internal signal (input)
TSnADTRG0, TSnADTRG1	A/D trigger signal	Internal signal (output)
TSnPTE	Pattern input change detection signal	Internal signal (output)
TSnPEC	Two phase encoder count signal	Internal signal (output)

Note * Not provided in the µPD70F3505A.

Interrupt requests The TSG2 interrupt requests are listed in Table 15-5.

Table 15-5 List of TSG2 Interrupt Requests

TSG2n Interrupt Request	Function	Connected to
INTTSG2nI00	TSnCMP0 compare match interrupt	Interrupt controller
INTTSG2nI01	TSnCMP1 compare match interrupt	Interrupt controller
INTTSG2nI02	TSnCMP2 compare match interrupt	Interrupt controller
INTTSG2nI03	TSnCMP3 compare match interrupt	Interrupt controller
INTTSG2nI04	TSnCMP4 compare match interrupt	Interrupt controller
INTTSG2nI05	TSnCMP5 compare match interrupt	Interrupt controller
INTTSG2nI06	TSnCMP6 compare match interrupt	Interrupt controller
INTTSG2nI07	TSnCMP7 compare match interrupt	Interrupt controller
INTTSG2nI08	TSnCMP8 compare match interrupt	Interrupt controller
INTTSG2nI09	TSnCMP9 compare match interrupt	Interrupt controller
INTTSG2nI10	TSnCMP10 compare match interrupt	Interrupt controller
INTTSG2nI11	TSnCMP11 compare match interrupt	Interrupt controller DMA, DTS
INTTSG2nI12	TSnCMP12 compare match intererupt	Interrupt controller DMA, DTS
INTTSG2nIPEK	Peak interrupt	Interrupt controller DMA, DTS
INTTSG2nIVLY	Valley interrupt	Interrupt controller DMA, DTS
INTTSG2nIER	Error interrupt	Interrupt controller
INTTSG2nIWN	Warning interrupt	Interrupt controller

15.2 Functional Overview

The TSG2n is a 16-bit timer counter with various motor control functions.

- Count clock resolution: Minimum 12.5 ns (count clock = 80 MHz)
- Operating mode corresponding to various motor control methods
- Compare registers with reload buffer
- 10-bit dead time counter
  - Dead time counter with reload buffer
  - Independent dead time can be set for positive to inverse phase change and inverse to positive phase change.
- A/D conversion trigger signal generation
  - Three A/D conversion trigger signals can be generated by the compare registers TSnDCMP0W and TSnDCMP2.
  - Skipping function of A/D conversion trigger signals TSnADTRG0 and TSnADTRG1 can be set independently. The skipping ratio can be selected among 1/1, 1/2, 1/4, and 1/8.
  - The dedicated pin (TSG2nO7) can be used to output the toggle or diagnostic signal set by the TSnADTRG0 signal and reset by the TSnADTRG1 signal.
- Interrupt skipping
  - Skipping rate: 1/1 to 1/32
- Forced output stop function
  - Using the TOP function allows the high impedance control of the TSG2nO1 to TSG2nO6 pin output.
- Compare value setting
  - Reload (simultaneous rewrite) or anytime rewrite can be selected.
- Reload mode
  - Writing to TSnCMP1 (setting the reload request flag (TSnRSF)) enables reload, and allows simultaneous transfer of the values of multiple registers.
  - Data can be transferred at peak/valley/peak or valley reload timing*
  - Reload request flag (TSnRSF)
  - Register address assignment allowing DMA transfer
  - Reload skipping
- HT-PWM mode
  - 0 to 100% PWM duty cycle output is possible (with possible dead time reduction).
  - The LSB in the compare register can be used to append an additional pulse to the PWM output during count up, thus improving the output resolution without extra load on software.
- 120-DC control
  - Semi-automatic cruise function (trigger signal can be generated by an offset in conjunction with two-phase encoder, three-phase encoder, or ENCA).
- Three-phase encoder function (hall sensor signals can be input).
- Active level of the output pins (TSG2nO1 to TSG2nO6) can be set individually.
- Fail-safe function (warning interrupt or error interrupt can be generated)
  - Simultaneous active output detection function for positive and inverse phases.
  - Abnormal input detection function of the three-phase encoder

15.3 Configuration

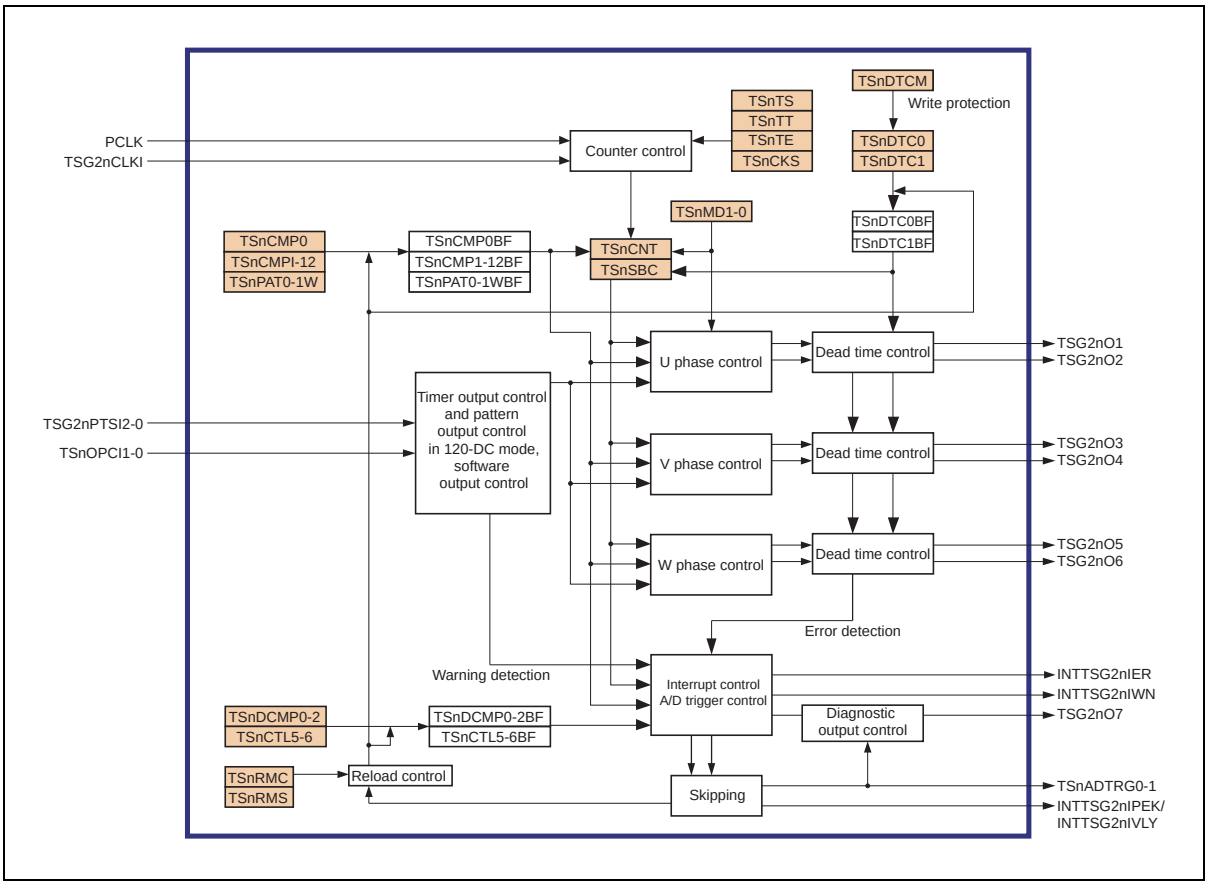


Figure 15-1 Block Diagram of TSG2n

15.4 Registers

This section contains a description of all registers of the TSG2n.

15.4.1 TSG2n Registers Overview

The TSG2n is controlled and operated based on the settings of the following registers listed in Table 15-6.

Table 15-6 TSG2n Registers (1/2)

Register Name	Symbol	Address
TSG2n control register 0	TSnCTL0	<TSG2n_base0> + 208_H
TSG2n control register 1	TSnCTL1	<TSG2n_base0> + $20C_H$
TSG2n control register 2	TSnCTL2*	<TSG2n_base1> + 078_H
TSG2n control register 3	TSnCTL3	<TSG2n_base1> + 004_H
TSG2n control register 4	TSnCTL4	<TSG2n_base1> + $07C_H$
TSG2n control register 5	TSnCTL5	<TSG2n_base1> + 008_H
TSG2n control register 6	TSnCTL6	<TSG2n_base1> + $00C_H$
TSG2n I/O control register 0	TSnIOC0	<TSG2n_base0> + 200_H
TSG2n I/O control register 1	TSnIOC1	<TSG2n_base0> + 204_H
TSG2n I/O control register 2	TSnIOC2	<TSG2n_base1> + 000_H
TSG2n I/O control register 3	TSnIOC3	<TSG2n_base1> + 074_H
TSG2n status register 0	TSnSTR0	<TSG2n_base1> + 010_H
TSG2n status register 1	TSnSTR1	<TSG2n_base1> + 014_H
TSG2n status register 2	TSnSTR2	<TSG2n_base1> + 018_H
TSG2n status clear trigger register	TSnSTC	<TSG2n_base1> + $01C_H$
TSG2n option register 0	TSnOPT0	<TSG2n_base1> + 020_H
TSG2n option register 1	TSnOPT1	<TSG2n_base1> + 024_H
TSG2n trigger register 0	TSnTRG0	<TSG2n_base1> + 030_H
TSG2n trigger register 1	TSnTRG1	<TSG2n_base1> + 034_H
TSG2n counter read buffer register	TSnCNT	<TSG2n_base1> + 028_H
TSG2n sub-counter read buffer register	TSnSBC	<TSG2n_base1> + $02C_H$
TSG2n compare register 0	TSnCMP0	<TSG2n_base1> + 058_H
TSG2n compare register 1W	TSnCMP1W	<TSG2n_base1> + 040_H
TSG2n compare register 3W	TSnCMP3W	<TSG2n_base1> + $04C_H$
TSG2n compare register 5W	TSnCMP5W	<TSG2n_base1> + 044_H
TSG2n compare register 7W	TSnCMP7W	<TSG2n_base1> + 050_H
TSG2n compare register 9W	TSnCMP9W	<TSG2n_base1> + 048_H
TSG2n compare register 11W	TSnCMP11W	<TSG2n_base1> + 054_H
TSG2n compare registers 1 to 12	TSnCMP1-12	<TSG2n_base1> + 080_H - $0AC_H$
TSG2n diagnostic output control register 0 and 1	TSnDCMP0W	<TSG2n_base1> + $05C_H$
TSG2n diagnostic output control register 2	TSnDCMP2	<TSG2n_base1> + 060_H

Note * Not provided in the μPD70F3505A.

Table 15-6 TSG2n Registers (2/2)

Register Name	Symbol	Address
TSG2n pattern register 0W	TSnPAT0W	<TSG2n_base1> + 064_H
TSG2n pattern register 1W	TSnPAT1W	<TSG2n_base1> + 068_H
TSG2n dead time setting register 0W	TSnDTC0W	<TSG2n_base1> + $06C_H$
TSG2n dead time setting register 1W	TSnDTC1W	<TSG2n_base1> + 070_H
TSG2n HT-PWM U phase compare register	TSnCMPU	<TSG2n_base1> + $0B0_H$
TSG2n HT-PWM V phase compare register	TSnCMPV	<TSG2n_base1> + $0B4_H$
TSG2n HT-PWM W phase compare register	TSnCMPW	<TSG2n_base1> + $0B8_H$
TSG2n SP-PWM U phase active width setting register	TSnUPW	<TSG2n_base1> + $0BC_H$
TSG2n SP-PWM V phase active width setting register	TSnVPW	<TSG2n_base1> + $0C0_H$
TSG2n SP-PWM W phase active width setting register	TSnWPW	<TSG2n_base1> + $0C4_H$
TSG2n dead time protection register	TSnDTPR	<TSG2n_base0> + 210_H

15.4.2 TSG2n Register Details

(1) TSG2n Control Register 0 (TSnCTL0)

This register specifies the pulse width for the diagnostic output and operating mode of the TSG2n.

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

Address <TSG2n_base0> + 208_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	TSn DWD	-	-	TSnMD [1:0]	
R	R	R	R/W	R	R	R/W	R/W

Table 15-7 TSnCTL0 Register Contents

Bit Position	Bit Name	Function
4	TSnDWD	Selects the pulse width for the diagnostic output. 0: The output pulse width is set to 8 clocks. 1: The output pulse width is set to 16 clocks. • The setting of this bit is valid when diagnostic output is enabled (TSnIOC1.TSnTGS = 1).
1, 0	TSnMD[1:0]	Selects timer mode.

TSnMD1	TSnMD0	Timer Mode
0	0	PWM mode
0	1	HT-PWM mode (HT-PWM)
1	0	Shift pulse PWM mode (SP-PWM)
1	1	120-DC mode

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

(2) TSG2n Control Register 1 (TSnCTL1)

This register controls the flags of TSG2n.

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

Address <TSG2n_base0> + $20C_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSn TBA2	TSn TBA1	TSn TBA0	TSn PPC	TSn PEC	TSn TDC	TSn NDC	TSn PRC	TSnPTC [1:0]	
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-8 TSnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
9	TSnTBA2	Enables or disables detection of the simultaneous active states of the TSG2nO5 and TSG2nO6 pins. 0: Disables detection of simultaneous active states of the TSG2nO5 and TSG2nO6 pins. 1: Enables detection of simultaneous active states of the TSG2nO5 and TSG2nO6 pins. • If the simultaneous active state is detected when the TSnIOC1.TSnEOC and TSnTBA2 bits in TSnIOC1 are 1, the positive phase and inverse phase simultaneous active detection flag 2 (TSnTBF2) is set to 1, and an error interrupt (INTTSG2nIER) is generated.
8	TSnTBA1	Enables or disables detection of the simultaneous active states of the TSG2nO3 and TSG2nO4 pins. 0: Disables detection of simultaneous active states of the TSG2nO3 and TSG2nO4 pins. 1: Enables detection of simultaneous active states of the TSG2nO3 and TSG2nO4 pins. • If the simultaneous active state is detected when the TSnIOC1.TSnEOC and TSnTBA1 bits in TSnIOC1 are 1, the positive phase and inverse phase simultaneous active detection flag 1 (TSnTBF1) is set to 1, and an error interrupt (INTTSG2nIER) is generated.
7	TSnTBA0	Enables or disables detection of the simultaneous active states of the TSG2nO1 and TSG2nO2 pins. 0: Disables detection of simultaneous active states of the TSG2nO1 and TSG2nO2 pins. 1: Enables detection of simultaneous active states of the TSG2nO1 and TSG2nO2 pins. • If the simultaneous active state is detected when the TSnIOC1.TSnEOC and TSnTBA0 bits in TSnIOC1 are 1, the positive phase and inverse phase simultaneous active detection flag 0 (TSnTBF0) is set to 1, and an error interrupt (INTTSG2nIER) is generated.
6	TSnPPC	Enables or disables detection of the pattern phase difference (TSnSTR2.TSnPPF) between the TSG2nPTSI2 to TSG2nPTSI0 and TSnOPF2 to TSnOPF0. 0: Disables detection of I/O pattern difference. 1: Enables detection of I/O pattern difference.

Caution If TSnDTC0 or TSnDTC1 is set to 0000_H (without dead time), the TSnTBA2 to TSnTBA0 bits should be set to 0
This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

Table 15-8 TSnCTL1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>5</td><td>TSnPEC</td><td>Enables or disables detection of the pattern error (TSnSTR2.TSnPEF) of the TSG2nPTSI2 to TSG2nPTSI0 pins.
0: Disables detection of the pattern error of the TSG2nPTSI2 to TSG2nPTSI0 pins.
1: Enables detection of the pattern error of the TSG2nPTSI2 to TSG2nPTSI0 pins.</td></tr>
<tr><td>4</td><td>TSnTDC</td><td>Enables or disables detection of the simultaneous trigger (TSnSTR2.TSnTDF) of the TSnOPCI0 and TSnOPCI1 signals.
0: Disables detection of the simultaneous trigger of the TSnOPCI0 and TSnOPCI1 signals.
1: Enables detection of the simultaneous trigger of the TSnOPCI0 and TSnOPCI1 signals.</td></tr>
<tr><td>3</td><td>TSnNDC</td><td>Enables or disables detection of the noise generation (two or more pins change simultaneously) (TSnSTR2.TSnNDF) on the TSG2nPTSI2 to TSG2nPTSI0 pins.
0: Disables detection of the noise generation on the TSG2nPTSI2 to TSG2nPTSI0 pins.
1: Enables detection of the noise generation on the TSG2nPTSI2 to TSG2nPTSI0 pins.</td></tr>
<tr><td>2</td><td>TSnPRC</td><td>Enables or disables detection of the reversal of the pattern (TSnSTR2.TSnPRF) of the TSG2nPTSI2 to TSG2nPTSI0 pins.
0: Disables detection of the reversal of the pattern of the TSG2nPTSI2 to TSG2nPTSI0 pins.
1: Enables detection of the reversal of the pattern of the TSG2nPTSI2 to TSG2nPTSI0 pins.</td></tr>
<tr><td>1, 0</td><td>TSnPTC[1:0]</td><td>Enables or disables detection of an abnormal toggle (TSnSTR2.TSnPTF) of the TSG2nPTSI2 to TSG2nPTSI0 pins between TSnOPCI1 and TSnOPCI0 triggers.
<table>
<tr><th>TSnPTC1</th><th>TSnPTC0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Disables detection of an abnormal toggle of the TSG2nPTSI2 to TSG2nPTSI0 pins.</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>Enables detection of an abnormal toggle of the TSG2nPTSI2 to TSG2nPTSI0 pins.</td></tr>
<tr><td>1</td><td>1</td><td>Enables detection of an abnormal toggle of the TSG2nPTSI2 to TSG2nPTSI0 pins.
When an abnormal toggle is detected, the pattern output trigger is automatically switched from pattern switch to trigger switch (TSnPOT is switched from 1 to 0).</td></tr>
</table>
</td></tr>
</table>

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

(3) TSG2n Control Register 2 (TSnCTL2)

This register specifies the count clock.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 078_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TSnCKS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15-9 TSnCTL2 Register Contents

Bit Position	Bit Name	Function
0	TSnCKS	Selects the count clock. 0: Selects the PCLK as the count clock. 1: Counts based on the PCLK when the clock enable input (TSG2nCLKI) is high.

Caution The TSnCKS bit should be set to 0 in HT-PWM mode.

(4) TSG2n Control Register 3 (TSnCTL3)

This register selects the rewrite method of the compare registers.

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

Address <TSG2n_base1> + 004_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSn RIA	TSn RMC
R	R	R	R	R	R	R/W	R/W

Table 15-10 TSnCTL3 Register Contents

Bit Position	Bit Name	Function
1	TSnRIA	Selects the reload timing of the compare register values. 0: The reload timing is set to peak reload timing (set by TSnCTL4.TSnPRE) and valley reload timing (set by TSnCTL4.TSnVRE). 1: The reload timing is set to peak interrupt timing and valley interrupt timing. • The setting of this bit is valid in reload mode (TSnRMC = 0).
0	TSnRMC	Selects the transfer timing of the compare register values. 0: Reload mode (simultaneous rewrite) Writing to TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW) enables reloading and the register values are rewritten simultaneously at the next reload timing. Writing to any register other than TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW) does not enable reloading. 1: Anytime rewrite mode The compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected immediately. TSnRSF is cleared.

(5) TSG2n Control Register 4 (TSnCTL4)

This register enables or disables generation of a peak interrupt and a valley interrupt, and the reload timing.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + $07C_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TSn PRE	TSn VRE	TSn PIE	TSn VIE	TSnRCC[04:00]				
R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-11 TSnCTL4 Register Contents (1/2)

Bit Position	Bit Name	Function
8	TSnPRE	Enables or disables the peak reload timing. 0: Disables reload operation at the peak timing of the 16-bit counter. 1: Enables reload operation at the peak timing of the 16-bit counter. • The peak reload timing means the peak timing of the 16-bit counter in HT-PWM mode and the clear timing of the 16-bit counter by compare match in any mode other than HT-PWM mode. • When the reload operation at the peak timing of the 16-bit counter is disabled (TSnPRE = 0), reload is not executed in any mode other than HT-PWM mode.
7	TSnVRE	Enables or disables the valley reload timing. 0: Disables reload operation at the valley timing of the 16-bit counter. 1: Enables reload operation at the valley timing of the 16-bit counter. • The setting of this bit is valid only in HT-PWM mode.
6	TSnPIE	Enables or disables generation of a peak interrupt (INTTSG2nIPEK). 0: Disables generation of a peak interrupt (INTTSG2nIPEK) at the peak timing of the 16-bit counter. Interrupts are not skipped. 1: Enables generation of a peak interrupt (INTTSG2nIPEK) at the peak timing of the 16-bit counter. Interrupts are skipped.
5	TSnVIE	Enables or disables generation of a valley interrupt (INTTSG2nIVLY). 0: Disables generation of a valley interrupt (INTTSG2nIVLY) at the valley timing of the 16-bit counter. Interrupts are not skipped. 1: Enables generation of a valley interrupt (INTTSG2nIVLY) at the valley timing of the 16-bit counter. Interrupts are skipped. • The setting of this bit is valid only in HT-PWM mode.

Table 15-11 TSnCTL4 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TSnRCC
[04:00]</td><td>Specifies the skipping rate of the interrupts (INTTSG2nIPEK and INTTSG2nIVLY) and reload.
<table>
<tr><th>TSnRCC04</th><th>TSnRCC03</th><th>TSnRCC02</th><th>TSnRCC01</th><th>TSnRCC00</th><th>Skipping Rate</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>None</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1/4</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1/30</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1/31</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1/32</td></tr>
</table>
• When a write access is made (including a write of the same value to TSnRCC04 to TSnRCC00) to TSnCTL4 during timer operation (TSnSTR0.TSnTE = 1), the interrupt skipping counter is cleared.</td></tr>
</table>

(6) TSG2n Control Register 5 (TSnCTL5)

This register controls A/D conversion trigger output (TSnADTRG0).

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

Address <TSG2n_base1> + 008_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	TSnACC [01:00]		TSn AT09	TSn AT08	TSn AT07	TSn AT06	TSn AT05	TSn AT04	TSn AT03	TSn AT02	TSn AT01	TSn AT00
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-12 TSnCTL5 Register Contents (1/3)

Bit Position	Bit Name	Function
11, 10	TSnACC [01:00]	Specifies the skipping rate of the A/D conversion trigger (TSnADTRG0). **TSnACC01 / TSnACC00 / Skipping Rate** 0 / 0 / None 0 / 1 / 1/2 1 / 0 / 1/4 1 / 1 / 1/8 • When a write access is made (including a write of the same value to TSnACC01 and TSnACC00) to TSnCTL5 during timer operation (TSnSTR0.TSnTE = 1), the interrupt skipping counter is cleared.
9	TSnAT09	Specifies generation of A/D conversion trigger (TSnADTRG0) at the (peak) timing when the 16-bit sub-counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the peak timing of the 16-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 16-bit sub-counter. • The TSnAT09 bit can be set to 1 only in HT-PWM mode. In other modes, the TSnAT09 bit should be set to 0. • Do not set the TSnAT09 bit to 1 when TSnDTC0W is not 0000_H and TSnDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter even if set so.
8	TSnAT08	Specifies generation of A/D conversion trigger (TSnADTRG0) at the (valley) timing when the 16-bit sub-counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the valley timing of the 16-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 16-bit sub-counter. • The TSnAT08 bit can be set to 1 only in HT-PWM mode. In other modes, the TSnAT08 bit should be set to 0. • Do not set the TSnAT08 bit to 1 when TSnDTC0W is 0000_H and TSnDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 16-bit sub-counter even if set so.

Table 15-12 TSnCTL5 Register Contents (2/3)

Bit Position	Bit Name	Function
7	TSnAT07	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP2 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP2 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP2 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
6	TSnAT06	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value.
5	TSnAT05	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
4	TSnAT04	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1 value.
3	TSnAT03	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
2	TSnAT02	Specifies generation of A/D conversion trigger (TSnADTRG0) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP0 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP0 value. 1: Enables generation of the A/D conversion trigger at the timing of the 16-bit counter value during incrementation with the TSnDCMP0.
1	TSnAT01	Specifies generation of A/D conversion trigger (TSnADTRG0) at the timing (peak interrupt) when the 16-bit counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG2nIPEK) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG2nIPEK) after being skipped.

Table 15-12 TSnCTL5 Register Contents (3/3)

Bit Position	Bit Name	Function
0	TSnAT00	Specifies generation of A/D conversion trigger (TSnADTRG0) at the timing (valley interrupt) when the 16-bit counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG2nIVLY) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG2nIVLY) after being skipped. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.

(7) TSG2n Control Register 6 (TSnCTL6)

This register controls the A/D conversion trigger output (TSnADTRG1).

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

Address <TSG2n_base1> + $00C_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	TSnACC [11:10]		TSn AT19	TSn AT18	TSn AT17	TSn AT16	TSn AT15	TSn AT14	TSn AT13	TSn AT12	TSn AT11	TSn AT10
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-13 TSnCTL6 Register Contents (1/3)

Bit Position	Bit Name	Function
11, 10	TSnACC [11:10]	Specifies the skipping rate of the A/D conversion trigger (TSnADTRG1). **TSnACC11 / TSnACC10 / Skipping Rate** 0 / 0 / None 0 / 1 / 1/2 1 / 0 / 1/4 1 / 1 / 1/8 • When a write access is made (including a write of the same value to TSnACC11 and TSnACC10) to TSnCTL5 during timer operation (TSnSTR0.TSnTE = 1), the skipping counter is cleared.
9	TSnAT19	Specifies generation of A/D conversion trigger (TSnADTRG1) at the (peak) timing when the 16-bit sub-counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the peak timing of the 16-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 16-bit sub-counter. • The TSnAT19 bit can be set to 1 only in HT-PWM mode. In other modes, the TSnAT19 bit should be set to 0. • Do not set the TSnAT19 bit to 1 when TSnDTC0W is not 0000_H and TSnDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter even if set so.
8	TSnAT18	Specifies generation of A/D conversion trigger (TSnADTRG1) at the (valley) timing when the 16-bit sub-counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the valley timing of the 16-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 16-bit sub-counter. • The TSnAT18 bit can be set to 1 only in HT-PWM mode. In other modes, the TSnAT18 bit should be set to 0. • Do not set the TSnAT18 bit to 1 when TSnDTC0W is 0000_H and TSnDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 16-bit sub-counter even if set so.

Table 15-13 TSnCTL6 Register Contents (2/3)

Bit Position	Bit Name	Function
7	TSnAT17	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during defragmentation and the TSnDCMP2 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP2 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP2 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
6	TSnAT16	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP2 value.
5	TSnAT15	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP1 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
4	TSnAT14	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP1 value.
3	TSnAT13	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during defragmentation with the TSnDCMP0 value. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
2	TSnAT12	Specifies generation of A/D conversion trigger (TSnADTRG1) at the match timing of the 16-bit counter value during incrementation with the TSnDCMP0. 0: Disables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP0 value. 1: Enables generation of the A/D conversion trigger at the match timing of the 16-bit counter value during incrementation with the TSnDCMP0 value.
1	TSnAT11	Specifies generation of A/D conversion trigger (TSnADTRG1) at the timing (peak interrupt) when the 16-bit counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG2nIPEK) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG2nIPEK) after being skipped.

Table 15-13 TSnCTL6 Register Contents (3/3)

Bit Position	Bit Name	Function
0	TSnAT10	Specifies generation of A/D conversion trigger (TSnADTRG1) at the timing (valley interrupt) when the 16-bit counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG2nIVLY) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG2nIVLY) after being skipped. • This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.

(8) TSG2n I/O Control Register 0 (TSnIOC0)

This register controls the timer output pins (TSG2nO1 to TSG2nO6).

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

Address <TSG2n_base0> + 200_H

Initial value $7E_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	TSn TOE6	TSn TOE5	TSn TOE4	TSn TOE3	TSn TOE2	TSn TOE1	-
R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 15-14 TSnIOC0 Register Contents

Bit Position	Bit Name	Function
6 to 1	TSnTOE6 to TSnTOE1	Enables or disables writing to TSnIOC2. 0: Enables control of TSG2nO6 to TSG2nO1 by rewriting TSnIOC2. 1: Disables control of TSG2nO6 to TSG2nO1 by rewriting TSnIOC2 (rewriting TSnIOC2 is ignored).

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

(9) TSG2n I/O Control Register 1 (TSnIOC1)

This register controls the timer output pins (TSG2nO1 to TSG2nO6).

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

Address <TSG2n_base0> + 204_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	TSn PTS	TSn EOC	TSn WOC	TSn TGS	TSn TOS
R	R	R	R/W	R/W	R/W	R/W	R/W

Table 15-15 TSnIOC1 Register Contents (1/2)

Bit Position	Bit Name	Function
4	TSnPTS	Enables or disables output of the edge detection signal (TSnPTE) of TSG2nPTSI0 to TSG2nPTSI2 and two-phase encoder count signal (TSnPEC). 0: Disables output of the toggle signal by edge detection of TSG2nPTSI0 to TSG2nPTSI2. 1: Enables output of the toggle signal by edge detection of TSG2nPTSI0 to TSG2nPTSI2. • When TSnPTS is changed from 1 to 0, the levels of the TSnPTE and TSnPEC remain unchanged.
3	TSnEOC	Enables or disables detection of the error condition at the motor control. 0: Disables generation of an error interrupt (INTTSG2nIER). 1: Enables generation of an error interrupt (INTTSG2nIER). • For details on controlling the error interrupt, see section 15.10.1, Error Interrupt Function.
2	TSnWOC	Enables or disables detection of the warning condition at the motor control. 0: Disables generation of a warning interrupt (INTTSG2nIWN). 1: Enables generation of a warning interrupt (INTTSG2nIWN). • For details on the controlling generation of warning interrupt, see section 15.10.2, Warning Interrupt Function.

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

Table 15-15 TSnIOC1 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TSnTGS	Selects the A/D conversion trigger diagnostic output (TSG2nO7) signal. 0: Selects A/D conversion trigger output. 1: Selects diagnostic output.
0	TSnTOS	Selects the timer counter increment/decrement status output (TSG2nO0) signal. 0: Outputs the up/down count flag of the 16-bit counter. 1: Outputs the up/down count flag of the 16-bit sub-counter. • When TSnTOS is 0, the status of TSnSTR0.TSnCUF is output to TSG2nO0. When TSnTOS is 1, the status of TSnSTR0.TSnSUF is output to TSG2nO0. • The setting of this bit is valid only in HT-PWM mode.

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

(10) TSG2n I/O Control Register 2 (TSnIOC2)

This register controls the timer output pins (TSG2nO1 to TSG2nO6).

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

Address <TSG2n_base1> + 000_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	TSnOL6	TSnOL5	TSnOL4	TSnOL3	TSnOL2	TSnOL1	-	-	TSnTO6	TSnTO5	TSnTO4	TSnTO3	TSnTO2	TSnTO1	-
R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 15-16 TSnIOC2 Register Contents

Bit Position	Bit Name	Function
14 to 9	TSnOL6 to TSnOL1	Specifies the active level of TSG2nO6 to TSG2nO1 outputs. 0: High 1: Low
6 to 1	TSnTO6 to TSnTO1	Specifies the latch level of the output buffer of the TSG2nO6 to TSG2nO1. 0: Low 1: High

Caution When the counting is stopped (TSnTE = 0), the TSG2nO1 to TSG2nO6 pins maintain their previous output states. The output level should be changed by setting the TSnIOC0.TSnTOEm bit to 0, using the TSnTOm bit. This register can be rewritten when TSnIOC0.TSnTOEm = 0 (m = 1 to 6).

Note While the timer is stopped (TSnSTR0.TSnTE = 0) and control of TSG2nOm is enabled (TSnIOC0.TSnTOEm = 0), the TSnOLm and TSnTOm of TSnIOC2 select the output level for the corresponding TSG2nOm output as listed in the table below.

TSnOLm	TSnTOm	Output level from TSG2nOm
0	0	Low
0	1	High
1	0	High
1	1	Low

(11) TSG2n I/O Control Register 3 (TSnIOC3)

This register controls timer output pins (TSG2nO1 to TSG2nO6).

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 074_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	TSn TOL6	TSn TOL5	TSn TOL4	TSn TOL3	TSn TOL2	TSn TOL1	-
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 15-17 TSnIOC3 Register Contents

Bit Position	Bit Name	Function
6 to 1	TSnTOL6 to TSnTOL1	Controls the set/clear level of output. 0: Outputs the normal level. 1: Outputs the reversed level. • Setting of this bit is reflected at the start of output. The change of the output level is reflected at the next compare match timing after the change.

Caution TSnTOL6 to TSnTOL1 should be set to 0 in HT-PWM mode.

(12) TSG2n Status Register 0 (TSnSTR0)

This register controls the flags.

Access This register can only be read in 8-bit units.

Address <TSG2n_base1> + 010_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	TSnCUF	TSnSUF	TSnRSF	TSnTE
R	R	R	R	R	R	R	R

Table 15-18 TSnSTR0 Register Contents

Bit Position	Bit Name	Function
3	TSnCUF	Indicates the count direction of the 16-bit counter. 0: The 16-bit counter is incremented. 1: The 16-bit counter is decremented. • TSnCUF is valid only in HT-PWM mode. In other modes, it is invalid (TSnCUF = 0).
2	TSnSUF	Indicates the count direction of the 16-bit sub-counter. 0: The 16-bit sub-counter is incremented. 1: The 16-bit sub-counter is decremented. • TSnSUF detects counting of the 16-bit sub-counter from 0000_H to (TSnCMP0 value - 0002_H) as up-counting, and counting from the TSnCMP0 value to 0002_H as down-counting. • This bit is valid only in HT-PWM mode.
1	TSnRSF	Indicates whether there is a reload request. 0: No reload request or reload has completed. 1: There is a reload request. • This bit is valid only in TSnRMC = 0. • This bit indicates that the data to be transferred next is held. • This bit is set to 1 by writing to TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW), and cleared to 0 when reload has completed. • When TSnRMC is changed from 0 to 1 in HT-PWM mode, TSnRSF is cleared to 0.
0	TSnTE	Indicates the TSG2n operation status. 0: TSG2n is stopped. 1: TSG2n is operating. • This bit is set when TSnTRG0.TSnTS = 1, and cleared when TSnTRG1.TSnTT = 1.

(13) TSG2n Status Register 1 (TSnSTR1)

This register controls the flags.

Access This register can only be read in 8-bit units.

Address <TSG2n_base1> + 014_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	TSn TSF	TSnOPF[2:0]		
R	R	R	R	R	R	R	R

Table 15-19 TSnSTR1 Register Contents

Bit Position	Bit Name	Function
3	TSnTSF	Indicates the pattern change order of TSG2nPTSI0 to TSG2nPTSI2. 0: Indicates that patterns are input to TSG2nPTSI0 to TSG2nPTSI2 in the normal rotation pattern order 1: Indicates that patterns are input to TSG2nPTSI0 to TSG2nPTSI2 in the reverse rotation pattern order. Normal Rotation → Reverse Rotation ← TSG2nPTSI2 to TSG2nPTSI0: [1, 0, 1] [1, 0, 0] [1, 1, 0] [0, 1, 0] [0, 1, 1] [0, 0, 1] • Normal or reverse rotation can be detected from the first change of TSG2nPTSI0 to TSG2nPTSI2 after TSnTRG0.TSnTS has been set to 1. For details, see section 15.7.5 b Detection of Input Pattern Order.
2 to 0	TSnOPF [2:0]	Indicates the output pattern of the timer output pins (TSG2nO1 to TSG2nO6).

(14) TSG2n Status Register 2 (TSnSTR2)

This register controls the flags.

Access This register can only be read in 16-bit units.

Address <TSG2n_base1> + 018_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSn TBF2	TSn TBF1	TSn TBF0	TSn PPF	TSn PEF	TSn TDF	TSn NDF	TSn PRF	TSn PTF	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15-20 TSnSTR2 Register Contents (1/2)

Bit Position	Bit Name	Function
9	TSnTBF2	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSnCTL1.TSnTBA2 is 1. 0: Positive phase (TSG2nO5) and inverse phase (TSG2nO6) are not active simultaneously. 1: Positive phase (TSG2nO5) and inverse phase (TSG2nO6) are active simultaneously. • TSnTBF2 is set to 1 when the simultaneous active state of the positive phase (TSG2nO5) and inverse phase (TSG2nO6) is detected, and an error interrupt (INTTSG2nIER) is generated. TSnTBF2 can be cleared by writing 1 to TSnSTC.TSnTBR2. • The simultaneous active state is not detected when TSnTBA2 = 0.
8	TSnTBF1	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSnCTL1.TSnTBA1 is 1. 0: Positive phase (TSG2nO3) and inverse phase (TSG2nO4) are not active simultaneously. 1: Positive phase (TSG2nO3) and inverse phase (TSG2nO4) are active simultaneously. • TSnTBF1 is set to 1 when the simultaneous active state of the positive phase (TSG2nO3) and inverse phase (TSG2nO4) is detected, and an error interrupt (INTTSG2nIER) is generated. TSnTBF1 can be cleared by writing 1 to TSnSTC.TSnTBR1. • The simultaneous active state is not detected when TSnTBA1 = 0.
7	TSnTBF0	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSnCTL1.TSnTBA0 is 1. 0: Positive phase (TSG2nO1) and inverse phase (TSG2nO2) are not active simultaneously. 1: Positive phase (TSG2nO1) and inverse phase (TSG2nO2) are active simultaneously. • TSnTBF0 is set to 1 when the simultaneous active state of the positive phase (TSG2nO1) and inverse phase (TSG2nO2) is detected, and an error interrupt (INTTSG2nIER) is generated. TSnTBF0 can be cleared by writing 1 to TSnSTC.TSnTBR0. • The simultaneous active state is not detected when TSnTBA0 = 0.

Table 15-20 TSnSTR2 Register Contents (2/2)

Bit Position	Bit Name	Function
6	TSnPPF	Indicates detection of the difference between the input patterns (TSG2nPTSI0 to TSG2nPTSI2) and the output patterns (TSG2nO1 to TSG2nO6) after they are compared. 0: No phase difference detected between the input patterns (TSG2nPTSI0 to TSG2nPTSI2) and the output patterns (TSG2nO1 to TSG2nO6). 1: A phase difference detected between the input patterns (TSG2nPTSI0 to TSG2nPTSI2) and the output patterns (TSG2nO1 to TSG2nO6). • TSnPPF is set to 1 when a difference between input and output patterns is detected, and a warning interrupt (INTTSG2nIWN) is generated. TSnPPF can be cleared by writing 1 to TSnSTC.TSnPPR. • TSnPPF is valid when TSnOPT0.TSnSOC = 0, TSnPOT = 1, and TSnCTL1.TSnPPC = 1.
5	TSnPEF	Indicates whether an abnormal input (000_B or 111_B is input to TSG2nPTSI0 to TSG2nPTSI2) is detected. 0: No abnormal input (000_B or 111_B) to TSG2nPTSI0 to TSG2nPTSI2 detected. 1: Abnormal input (000_B or 111_B) to TSG2nPTSI0 to TSG2nPTSI2 detected. • TSnPEF is set to 1 when an input of 000_B or 111_B to TSG2nPTSI0 to TSG2nPTSI2 is detected, and a warning interrupt (INTTSG2nIWN) is generated. TSnPEF can be cleared by writing 1 to TSnSTC.TSnPER.
4	TSnTDF	Indicates whether simultaneous generation of the TSnOPCI0 and TSnOPCI1 triggers is detected. 0: Simultaneous generation of the TSnOPCI0 and TSnOPCI1 triggers not detected. 1: Simultaneous generation of the TSnOPCI0 and TSnOPCI1 triggers detected. • TSnTDF is set to 1 when simultaneous generation of the TSnOPCI0 and TSnOPCI1 triggers is detected, and a warning interrupt (INTTSG2nIWN) is generated. TSnTDF can be cleared by writing 1 to TSnSTC.TSnTDR.
3	TSnNDF	Indicates whether noise on TSG2nPTSI0 to TSG2nPTSI2 is detected. 0: Noise on TSG2nPTSI0 to TSG2nPTSI2 due to simultaneous change of two or more pins not detected. 1: Noise on TSG2nPTSI0 to TSG2nPTSI2 due to simultaneous change of two or more pins detected. • TSnNDF is set to 1 when simultaneous change of two or more pins in TSG2nPTSI0 to TSG2nPTSI2 is detected, and a warning interrupt (INTTSG2nIWN) is generated. TSnNDF can be cleared by writing 1 to TSnSTC.TSnNDR.
2	TSnPRF	Indicates whether reversal of the TSG2nPTSI0 to TSG2nPTSI2 input order is detected. 0: The reversal of the TSG2nPTSI0 TSG2nPTSI2 input order not detected. 1: The reversal of the TSG2nPTSI0 to TSG2nPTSI2 input order detected. • TSnPRF is set to 1 when TSnSTR1.TSnTSF changes, and a warning interrupt (INTTSG2nIWN) is generated. TSnPRF can be cleared by writing 1 to TSnSTC.TSnPRR. Detection is possible from the second TSG2nPTSI0 to TSG2nPTSI2 change timing after setting TSnTRG0.TSnTS = 1.
1	TSnPTF	Indicates whether an abnormal toggle of TSG2nPTSI0 to TSG2nPTSI2 is detected. 0: No abnormal toggle of TSG2nPTSI0 to TSG2nPTSI2 detected. 1: An abnormal toggle of TSG2nPTSI0 to TSG2nPTSI2 detected. • TSnPTF is set to 1 when TSG2nPTSI0 to TSG2nPTSI2 (TSnPTE signal toggle) are changed three times or more during TSnOPCI0 trigger or TSG2nPTSI0 to TSG2nPTSI2 (TSnPTE signal toggle) are changed three times or more during TSnOPCI1 trigger, and a warning interrupt (INTTSG2nIWN) is generated. TSnPTF can be cleared by writing 1 to TSnSTC.TSnPTR.

(15) TSG2n Status Clear Trigger Register (TSnSTC)

This register controls the flags.

Access This register can only be written in 16-bit units.

Address <TSG2n_base1> + $01C_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSn TBR2	TSn TBR1	TSn TBR0	TSn PPR	TSn PER	TSn TDR	TSn NDR	TSn PRR	TSn PTR	-
R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	R

Table 15-21 TSnSTC Register Contents (1/2)

Bit Position	Bit Name	Function
9	TSnTBR2	This is a trigger bit that clears TSnSTR2.TSnTBF2. 0: Does not clear TSnTBF2. 1: Clears TSnTBF2. • When TSnTBR2 writing and TSnSTR2.TSnTBF2 setting occur simultaneously, TSnSTR2.TSnTBF2 setting has a priority, and the flag is not cleared.
8	TSnTBR1	This is a trigger bit that clears TSnSTR2.TSnTBF1. 0: Does not clear TSnTBF1. 1: Clears TSnTBF1. • When TSnTBR1 writing and TSnSTR2.TSnTBF1 setting occur simultaneously, TSnSTR2.TSnTBF1 setting has a priority, and the flag is not cleared.
7	TSnTBR0	This is a trigger bit that clears TSnSTR2.TSnTBF0. 0: Does not clear TSnTBF0. 1: Clears TSnTBF0. • When TSnTBR0 writing and TSnSTR2.TSnTBF0 setting occur simultaneously, TSnSTR2.TSnTBF0 setting has a priority, and the flag is not cleared.
6	TSnPPR	This is a trigger bit that clears TSnSTR2.TSnPPF. 0: Does not clear TSnPPF. 1: Clears TSnPPF. • When TSnPPR writing and TSnSTR2.TSnPPF setting occur simultaneously, TSnSTR2.TSnPPF setting has a priority, and the flag is not cleared.
5	TSnPER	This is a trigger bit that clears TSnSTR2.TSnPEF. 0: Does not clear TSnPEF. 1: Clears TSnPEF. • When TSnPER writing and TSnSTR2.TSnPEF setting occur simultaneously, TSnSTR2.TSnPEF setting has a priority, and the flag is not cleared.
4	TSnTDR	This is a trigger bit that clears TSnSTR2.TSnTDF. 0: Does not clear TSnTDF. 1: Clears TSnTDF. • When TSnTDR writing and TSnSTR2.TSnTDF setting occur simultaneously, TSnSTR2.TSnTDF setting has a priority, and the flag is not cleared.

Table 15-21 TSnSTC Register Contents (2/2)

Bit Position	Bit Name	Function
3	TSnNDR	This is a trigger bit that clears TSnSTR2.TSnNDF. 0: Does not clear TSnNDF. 1: Clears TSnNDF. • When TSnNDR writing and TSnSTR2.TSnNDF setting occur simultaneously, TSnSTR2.TSnNDF setting has a priority, and the flag is not cleared.
2	TSnPRR	This is a trigger bit that clears TSnSTR2.TSnPRF. 0: Does not clear TSnPRF. 1: Clears TSnPRF. • When TSnPRR writing and TSnSTR2.TSnPRF setting occur simultaneously, TSnSTR2.TSnPRF setting has a priority, and the flag is not cleared.
1	TSnPTR	This is a trigger bit that clears TSnSTR2.TSnPTF. 0: Does not clear TSnPTF. 1: Clears TSnPTF. • When TSnPTR writing and TSnSTR2.TSnPTF setting occur simultaneously, TSnSTR2.TSnPTF setting has a priority, and the flag is not cleared.

(16) TSG2n Option Register 0 (TSnOPT0)

This register sets the optional functions.

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

Address <TSG2n_base1> + 020_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	TSn SOC	TSn STE	TSn POT	TSn PSS	TSn IDC	TSn PSC	-
R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 15-22 TSnOPT0 Register Contents (1/2)

Bit Position	Bit Name	Function
6	TSnSOC	Enables or disables control of the timer output (TSG2nO1 to TSG2nO6 pins) by software. 0: Disables control by software. 1: Enables control by software. • When TSnSOC is set to 1, timer output is switched to the software control/ trigger control output pattern specified by TSnSPC2 to TSnSPC0. The dead time is secured by the dead time counter.
5	TSnSTE	Enables or disables control by the pattern output trigger. 0: Disables the TSG2nPTSI0 to TSG2nPTSI2 and TSnOPCI0, and TSnOPCI1 inputs. 1: Enables TSG2nPTSI0 to TSG2nPTSI2 and TSnOPCI0, and TSnOPCI1 inputs. • The pattern output trigger is selected by TSnPOT. • TSnSTE is valid in 120-DC mode and when software output control function is enabled.
4	TSnPOT	Selects the pattern output trigger. 0: Switches the output pattern by the external pattern input pins (TSG2nPTSI0 to TSG2nPTSI2) (pattern switch method). 1: Switches the output pattern by the rising edge of the TSnOPCI0 and TSnOPCI1 (trigger switch method).
3	TSnPSS	Selects the pattern output order switch factor. 0: The pattern output order is not switched by TSnPSC. 1: The pattern output order is switched by TSnPSC. • TSnPSS is valid when TSnSTE = 1 and TSnPOT = 1 (trigger switch method). At this time, TSnPSS should be set to 1.
2	TSnIDC	Determines the output pattern from the TSG2nO1 to TSG2nO6 pins in combination with the TSnIDC and TSnSTR1.TSnTSF and TSnPSC signals. For the timer output order and patterns to be output, see Figure 15-67 to Figure 15-70, Example of Operation in 120-DC Mode, in section 15.11.4 5 Operation in 120-DC Mode.

Table 15-22 TSnOPT0 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TSnPSC	Selects the pattern output order when the semi-automatic cruise function is enabled. 0: Switches the timer output (TSG2nO1 to TSG2nO6) in the normal rotation. 1: Switches the timer output (TSG2nO1 to TSG2nO6) in the reverse rotation. • TSnPSC specifies the timer output pattern order assuming the output pattern specified by TSnSPC2 to TSnSPC0 as the initial pattern. TSnPSC is valid when TSnPOT = 1 and TSnPSS = 1. • It is recommended to rewrite TSnPSC when TSnSTR0.TSnTE = 0 or TSnPOT = 0. If TSnPSC is rewritten when TSnPOT = 1, the unexpected timer output pattern might be caused. • If the signal input to TSG2nPTSI0 to TSG2nPTSI2 changes with TSG2n operation being stopped (TSnSTR0.TSnTE = 0), the TSnTRG0.TSnTS bit should be set to 1 after matching the input signal change logic with the TSnPSC order. • For output order in normal or reverse rotation, see section 15.11.4, 120-DC Mode. Here, normal rotation and reverse rotation refer to the change of output, and they are different from normal rotation and reverse rotation of a motor.

(17) TSG2n Option Register 1 (TSnOPT1)

This register sets the optional functions.

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

Address <TSG2n_base1> + 024_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	TSnSPC[2:0]		
R	R	R	R	R	R/W	R/W	R/W

Table 15-23 TSnOPT1 Register Contents

Bit Position	Bit Name	Function
2 to 0	TSnSPC [2:0]	Specifies the timer output pattern when software output function is enabled and in 120-DC mode. For the output pattern, see section 15.11.5, Software Output Control Function, and section 15.11.4, 120-DC Mode.

(18) TSG2n Trigger Register 0 (TSnTRG0)

This register controls the start of the timer.

Access This register can only be written in 8-bit units.

Address <TSG2n_base1> + 030_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TSn TS
R	R	R	R	R	R	R	W

Table 15-24 TSnTRG0 Register Contents

Bit Position	Bit Name	Function
0	TSnTS	This bit is a trigger bit that controls the start of the timer. 0: The timer is not started. 1: The timer is started (restarted) (TSnSTR0.TSnTE = 1). • This bit is always read as 0.

(19) TSG2n Trigger Register 1 (TSnTRG1)

This register controls the stop of the timer.

Access This register can only be written in 8-bit units.

Address <TSG2n_base1> + 034_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TSnTT
R	R	R	R	R	R	R	W

Table 15-25 TSnTRG1 Register Contents

Bit Position	Bit Name	Function
0	TSnTT	This is a trigger bit that controls the stop of the timer. 0: The timer is not stopped. 1: The timer is stopped (TSnSTR0.TSnTE = 0). • This bit is always read as 0.

(20) TSG2n Counter Read Buffer Register (TSnCNT)

From this register the counter value can be read.

Access This register can only be read in 16-bit units.

Address <TSG2n_base1> + 028_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit counter															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16-bit counter This register is a timer read buffer register from which the 16-bit counter value can be read. In HT-PWM mode, the 16-bit counter provides the triangle wave control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0.

In other modes, the 16-bit counter provides the sawtooth wave control in which the counter value is incremented by 1.

Table 15-26 TSnCNT Register Count Value

Operating Mode	At the Beginning	Minimum Value	Maximum Value
HT-PWM mode	TSnDTC0	TSnDTC0	TSnDTC0 + TSnCMP0*
Other modes	0000_H	0000_H	TSnCMP0

Note * Use with the set value of TSnDTC0+TSnCMP0 < $FFFF_H$.

(21) TSG2n Sub-Counter Read Buffer Register (TSnSBC)

From this register the sub-counter value can be read.

Access This register can only be read in 16-bit units.

Address <TSG2n_base1> + $02C_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit sub-counter															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16-bit sub-counter This register is a timer read buffer register from which the 16-bit sub-counter value can be read. In HT-PWM mode, the 16-bit sub-counter provides the triangle wave control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0. (This register can be used only in HT-PWM mode.)

Table 15-27 TSnSBC Register Count Value

Operating Mode	At the Beginning	Minimum Value	Maximum Value
HT-PWM mode	TSnDTC0	0000_H	TSnDTC0 + TSnDTC1 + TSnCMP0*
Other modes	0000_H	0000_H	0000_H

Note * Use with the set value of TSnDTC0 + TSnDTC1 + TSnCMP0 < $FFFF_H$.

(22) TSG2n Compare Register 0 (TSnCMP0)

This register specifies the PWM period.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 058_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit compare register															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-28 TSnCMP0 Register Setting

Operating Mode	PWM Period	Minimum Value (Period)	Maximum Value (Period)
HT-PWM mode	TSnCMP0*	0002_H	$FFFE_H$
Other modes	TSnCMP0 + 1	1 (TSnCMP0 = 0000_H)	10000_H (TSnCMP0 = $FFFF_H$)

Note * In HT-PWM mode, the LSB is ignored.

(23) TSG2n Compare Register 1, 2 (TSnCMP1W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 040_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP2 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP1 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-29 TSnCMP1W Register Setting

Operating Mode	TSnCMP1	TSnCMP2	Minimum Value	Maximum Value
PWM mode	TSG2nO1 clear timing	TSG2nO1 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	TSG2nO1 clear timing/TSG2nO2 set timing	TSG2nO1 set timing/TSG2nO2 clear timing	0000_H	TSnCMP0 + TSnDTC0 + TSnDTC1
SP-PWM mode	TSG2nO1 clear timing/TSG2nO2 set timing	TSG2nO1 set timing/TSG2nO2 clear timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
120-DC mode	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is enabled in all operating modes.
In HT-PWM mode, the match timing between this register and TSnSBC is used.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(24) TSG2n Compare Register 3, 4 (TSnCMP3W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + $04C_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP4 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP3 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-30 TSnCMP3W Register Setting

Operating Mode	TSnCMP3	TSnCMP4	Minimum Value	Maximum Value
PWM mode	TSG2nO2 clear timing	TSG2nO2 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	-	-	-	-
SP-PWM mode	-	-	-	-
120-DC mode	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is enabled in all operating modes.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(25) TSG2n Compare Register 5, 6 (TSnCMP5W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 044_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP6 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP5 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-31 TSnCMP5W Register Setting

Operating Mode	TSnCMP5	TSnCMP6	Minimum Value	Maximum Value
PWM mode	TSG2nO3 clear timing	TSG2nO3 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	TSG2nO3 clear timing/TSG2nO4 set timing	TSG2nO3 set timing/TSG2nO4 clear timing	0000_H	TSnCMP0 + TSnDTC0 + TSnDTC1
SP-PWM mode	TSG2nO3 clear timing/TSG2nO4 set timing	TSG2nO3 set timing/TSG2nO4 clear timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
120-DC mode	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is enabled in all operating modes.
In HT-PWM mode, match timing between this register and TSnSBC is used.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(26) TSG2n Compare Register 7, 8 (TSnCMP7W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 050_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP8 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP7 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-32 TSnCMP7W Register Setting

Operating Mode	TSnCMP7	TSnCMP8	Minimum Value	Maximum Value
PWM mode	TSG2nO4 clear timing	TSG2nO4 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	-	-	-	-
SP-PWM mode	-	-	-	-
120-DC mode	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is enabled in all operating modes.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(27) TSG2n Compare Register 9, 10 (TSnCMP9W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 048_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP10 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP9 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-33 TSnCMP9W Register Setting

Operating Mode	TSnCMP9	TSnCMP10	Minimum Value	Maximum Value
PWM mode	TSG2nO5 clear timing	TSG2nO5 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	TSG2nO5 clear timing/TSG2nO6 set timing	TSG2nO5 set timing/TSG2nO6 clear timing	0000_H	TSnCMP0 + TSnDTC0 + TSnDTC1
SP-PWM mode	TSG2nO5 clear timing/TSG2nO6 set timing	TSG2nO5 set timing/TSG2nO6 clear timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
120-DC mode	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	Duty when TSG2nO1, TSG2nO3, or TSG2nO5 output pattern is selected by TSnPAT0	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is enabled in all operating modes.
In HT-PWM mode, the match timing between this register and TSnSBC is used.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(28) TSG2n Compare Register 11, 12 (TSnCMP11W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 054_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnCMP12 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMP11 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-34 TSnCMP11W Register Setting

Operating Mode	TSnCMP11	TSnCMP12	Minimum Value	Maximum Value
PWM mode	TSG2nO6 clear timing	TSG2nO6 set timing	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$
HT-PWM mode	-	-	-	-
SP-PWM mode	-	-	-	-
120-DC mode	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	Duty when TSG2nO2, TSG2nO4, or TSG2nO6 output pattern is selected by TSnPAT1	0000_H	TSnCMP0 + 1 (TSnCMP0 < $FFFF_H$) or $FFFF_H$

Note The dead time function is used in all operating modes.
In 120-DC mode, the output from TSG2nO1 to TSG2nO6 is controlled by TSnCMPm, TSnPAT0, and TSnPAT1.

(29) TSG2n Compare Register 1 to 12 (TSnCMP1 to TSnCMP12)

This register specifies the compare value.

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

Address TSnCMP1 <TSG2n_base1> + 080_H

TSnCMP2 <TSG2n_base1> + 084_H

TSnCMP3 <TSG2n_base1> + 098_H

TSnCMP4 <TSG2n_base1> + $09C_H$

TSnCMP5 <TSG2n_base1> + 088_H

TSnCMP6 <TSG2n_base1> + $08C_H$

TSnCMP7 <TSG2n_base1> + $0A0_H$

TSnCMP8 <TSG2n_base1> + $0A4_H$

TSnCMP9 <TSG2n_base1> + 090_H

TSnCMP10 <TSG2n_base1> + 094_H

TSnCMP11 <TSG2n_base1> + $0A8_H$

TSnCMP12 <TSG2n_base1> + $0AC_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit compare register															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The function of this register is the same as the function of the corresponding registers with the same name in TSnCMP1W, TSnCMP3W, TSnCMP5W, TSnCMP7W, TSnCMP9W, and TSnCMP11W. (When this register is accessed, it is reflected to the register with the same name.)

(30) TSG2n Diagnostic Output Compare Register 0, 1 (TSnDCMP0W)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + $05C_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSnDCMP1 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnDCMP0 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Setting of this register is used to control the diagnostic output or A/D conversion trigger timing in all the modes. A pulse is generated by a match of this register value with the 16-bit counter value.

(31) TSG2n Diagnostic Output Compare Register 2 (TSnDCMP2)

This register specifies the compare value.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 060_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnDCMP2 (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Setting of this register is used to control the diagnostic output or A/D conversion trigger timing in all the modes. A pulse is generated by a match of this register value with the 16-bit counter value.

(32) TSG2n Pattern Register 0 (TSnPAT0W)

This register specifies the output pattern.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 064_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	PAT5T	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAT5T	PAT4T			PAT3T			PAT2T			PAT1T			PAT0T		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output pattern This register controls UT/VT/WT output in 120-DC mode.

Table 15-35 TSnPAT0W Set Value and Output Control

PATmT value	Output Control
000	Fixed to low.
001	PWM output set by TSnCMP1
010	PWM output set by TSnCMP2
011	PWM output set by TSnCMP5
100	PWM output set by TSnCMP6
101	PWM output set by TSnCMP9
110	PWM output set by TSnCMP10
111	Fixed to high.

(m = 0, 1, 2, 3, 4, 5)

(33) TSG2n Pattern Register 1 (TSnPAT1W)

This register specifies the output pattern.

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 068_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	PAT5B	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAT5B	PAT4B			PAT3B			PAT2B			PAT1B			PAT0B		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output pattern This register controls UB/VB/WB output in 120-DC mode.

Table 15-36 TSnPAT1W Set Value and Output Control

PATmB value	Output Control
000	Fixed to low.
001	PWM output set by TSnCMP3
010	PWM output set by TSnCMP4
011	PWM output set by TSnCMP7
100	PWM output set by TSnCMP8
101	PWM output set by TSnCMP11
110	PWM output set by TSnCMP12
111	Fixed to high.

(m = 0, 1, 2, 3, 4, 5)

(34) TSG2n Dead Time Setting Register 0 (TSnDTC0W)

This register sets the dead time value (the period from inverse phase inactivation to positive phase activation).

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + $06C_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	Write protection code check														
R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSnDTC0 (10-bit dead time compare)									
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To rewrite TSnDTC0W[0:9], set bit 14 to bit 0 and TSnDTCM to 0 in TSnDTPR, and rewrite the TSnDTC0W. At this time, when the rewritten value of TSnDTC0W[30:16] and the TSnDTPR value match, TSnDTC0W is rewritten.

During timer operation (TSnSTR0.TSnTE = 1), rewriting should be performed in reload mode (TSnCTL3.TSnRMC = 0).

(35) TSG2n Dead Time Setting Register 1 (TSnDTC1W)

This register sets the dead time (the period from positive phase inactivation to inverse phase activation).

Access This register can be read/written in 32-bit units.

Address <TSG2n_base1> + 070_H

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	Write protection code check														
R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	TSnDTC1 (10-bit dead time compare)									
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To rewrite TSnDTC1W[0:9], set bit 30 to bit 16 and TSnDTCM to 0 in TSnDTPR, and rewrite the TSnDTC1W. At this time, when the rewritten value of TSnDTC1W[30:16] and the TSnDTPR value match, TSnDTC1W is rewritten.

During timer operation (TSnSTR0.TSnTE = 1), rewriting should be performed in reload mode (TSnCTL3.TSnRMC = 0).

(36) TSG2n HT-PWM U Phase Compare Register (TSnCMPU)

This register sets the compare value for U phase in HT-PWM mode.

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

Address <TSG2n_base1> + $0B0_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMPU (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in HT-PWM mode.
Note 2. The TSnCMPU set value is indicated on TSnCMP1W (TSnCMP1, TSnCMP2).
Note 3. When the TSnCMPU is read, the TSnCMP1 value is actually read.

(37) TSG2n HT-PWM V Phase Compare Register (TSnCMPV)

This register sets the compare value for V phase in HT-PWM mode.

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

Address <TSG2n_base1> + $0B4_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMPV (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in HT-PWM mode.
Note 2. The TSnCMPV set value is indicated on TSnCMP5W (TSnCMP5, TSnCMP6).
Note 3. When the TSnCMPV is read, the TSnCMP5 value is actually read.

(38) TSG2n HT-PWM W Phase Compare Register (TSnCMPW)

This register sets the compare value for W phase in HT-PWM mode.

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

Address <TSG2n_base1> + $0B8_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnCMPW (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in HT-PWM mode.
Note 2. The TSnCMPW set value is indicated on TSnCMP9W (TSnCMP9, TSnCMP10).
Note 3. When the TSnCMPW is read, the TSnCMP9 value is actually read.

(39) TSG2n SP-PWM U Phase Active Width Setting Register (TSnUPW)

This register sets the active width for U phase in SP-PWM mode.

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

Address <TSG2n_base1> + $0BC_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnUPW (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in SP-PWM mode.
Note 2. The sum of the TSnUPW set value and the TSnCMP2 set value is indicated on TSnCMP1.
Note 3. When the TSnUPW is read, the TSnCMP1 value is actually read.

(40) TSG2n SP-PWM V Phase Active Width Setting Register (TSnVPW)

This register sets the active width for V phase in SP-PWM mode.

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

Address <TSG2n_base1> + $0C0_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnVPW (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in SP-PWM mode.
Note 2. The sum of the TSnVPW set value and the TSnCMP6 set value is indicated on TSnCMP5.
Note 3. When the TSnVPW is read, the TSnCMP5 value is actually read.

(41) TSG2n SP-PWM W Phase Active Width Setting Register (TSnWPW)

This register sets the active width for W phase in SP-PWM mode.

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

Address <TSG2n_base1> + $0C4_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnWPW (16-bit compare register)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This register can be used only in SP-PWM mode.
Note 2. The sum of the TSnWPW set value and the TSnCMP10 set value is indicated on TSnCMP9.
Note 3. When the TSnWPW is read, the TSnCMP9 value is actually read.

(42) TSG2n Dead Time Protection Register (TSnDTPR)

This register controls protection of the write access to the dead time register.

Access This bit can be read/written in 16-bit units.

Address <TSG2n_base0> + 210_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSn DTCM	Write protection code check														
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-37 TSnDTPR Register Contents

Bit Position	Bit Name	Function
15	TSnDTCM	Enables or disables rewriting of TSnDTC0 and TSnDTC1. 0: Enables rewriting of TSnDTC0 and TSnDTC1. 1: Disables rewriting of TSnDTC0 and TSnDTC1.
14 to 0	TSnDTPR [14:0]	Sets the write protection code (any value from 0000 to 7FFF).

This register protects TSnDTC0 and TSnDTC1 from illegal rewriting.

Functions are described below.

- TSnDTCM enables or disables rewriting of TSnDTC0 and TSnDTC1.
- Rewriting of TSnDTC0 and TSnDTC1 is enabled or disabled by checking a match of the write protection code (bits 30 to 16) of TSnDTC0 with TSnDTC1 and the write protection code of TSnDTPR, and the TSnDTCM setting.

Caution This register should be set when the timer is stopped (TSnSTR0.TSnTE = 0). Only the same value can be written during timer operation (TSnSTR0.TSnTE = 1). If the different value is written to this register when TSnSTR0.TSnTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

15.5 Basic Operation

15.5.1 Basic Operation of 16-Bit Counter

The basic operation of the 16-bit counter is described. For details, see section 15.11, Operating Modes.

Counting start

The 16-bit counter of TSG2n starts counting from the initial value 0000_H in all modes except for HT-PWM mode.

The counter increments from 0000_H, 0001_H, 0002_H, 0003_H, … in all modes except for HT-PWM mode.

Counter clear

The 16-bit counter is cleared by the match of the counter value and the compare register set value.

Counter read during counting

In the TSG2n, the 16-bit counter value during counting can be read through TSnCNT.

Interrupt operation

In the TSG2n, the following interrupts are generated.

- INTTSG2nI00: A period interrupt by a match of the 16-bit counter value with the TSnDTC0 value in HT-PWM mode. A compare match interrupt of the 16-bit counter value with the TSnCMP0 buffer register in any mode other than HT-PWM mode.
- INTTSG2nI01: A compare match interrupt of the 16-bit counter value with the TSnCMP1 buffer register.
- INTTSG2nI02: A compare match interrupt of the 16-bit counter value with the TSnCMP2 buffer register.
- INTTSG2nI03: A compare match interrupt of the 16-bit counter value with the TSnCMP3 buffer register.
- INTTSG2nI04: A compare match interrupt of the 16-bit counter value with the TSnCMP4 buffer register.
- INTTSG2nI05: A compare match interrupt of the 16-bit counter value with the TSnCMP5 buffer register.
- INTTSG2nI06: A compare match interrupt of the 16-bit counter value with the TSnCMP6 buffer register.
- INTTSG2nI07: A compare match interrupt of the 16-bit counter value with the TSnCMP7 buffer register.
- INTTSG2nI08: A compare match interrupt of the 16-bit counter value with the TSnCMP8 buffer register.
- INTTSG2nI09: A compare match interrupt of the 16-bit counter value with the TSnCMP9 buffer register.
- INTTSG2nI10: A compare match interrupt of the 16-bit counter value with the TSnCMP10 buffer register.
- INTTSG2nI11: A compare match interrupt of the 16-bit counter value with the TSnCMP11 buffer register.
- INTTSG2nI12: A compare match interrupt of the 16-bit counter value with the TSnCMP12 buffer register.
- INTTSG2nIPEK: A peak interrupt when the 16-bit counter switches from incrementing to decrementing.
- INTTSG2nIVLY: A valley interrupt when the 16-bit counter switches from decrementing to incrementing.
- INTTSG2nIER: A simultaneous active state detection interrupt of the positive phase and inverse phase
- INTTSG2nIWN: A warning detection interrupt

15.5.2 Functions of Compare Registers

The functions of the compare registers in each operating mode are shown in Table 15-38.

Table 15-38 Compare Register Functions in Each Mode (1/4)

Operating Mode	TSnCMP0	TSnCMP1W	TSnCMP3W	TSnCMP5W
PWM mode	Setting PWM period	TSnCMP1: Setting TSG2nO1 clear timing TSnCMP2: Setting TSG2nO1 set timing	TSnCMP3: Setting TSG2nO2 clear timing TSnCMP4: Setting TSG2nO2 set timing	TSnCMP5: Setting TSG2nO3 clear timing TSnCMP6: Setting TSG2nO3 set timing
HT-PWM mode	Setting PWM period	TSnCMP1: Setting TSG2nO1 clear timing and TSG2nO2 set timing TSnCMP2: Setting TSG2nO1 set timing and TSG2nO2 clear timing	-	TSnCMP5: Setting TSG2nO3 clear timing and TSG2nO4 set timing TSnCMP6: Setting TSG2nO3 set timing and TSG2nO4 clear timing
SP-PWM mode	Setting PWM period	TSnCMP1: Setting TSG2nO1 clear timing and TSG2nO2 set timing TSnCMP2: Setting TSG2nO1 set timing and TSG2nO2 clear timing	-	TSnCMP5: Setting TSG2nO3 clear timing and TSG2nO4 set timing TSnCMP6: Setting TSG2nO3 set timing and TSG2nO4 clear timing
120-DC mode	Setting PWM period	TSnCMP1, TSnCMP2: Selecting TSG2nO1, TSG2nO3, and TSG2nO5 outputs by TSnPAT0.	TSnCMP3, TSnCMP4: Selecting TSG2nO2, TSG2nO4, and TSG2nO6 outputs by TSnPAT1.	TSnCMP5, TSnCMP6: Selecting TSG2nO1, TSG2nO3, and TSG2nO5 outputs by TSnPAT0.

Table 15-38 Compare Register Functions in Each Mode (2/4)

Operating Mode	TSnCMP7W	TSnCMP9W	TSnCMP11W	TSnCMP1 to TSnCMP12
PWM mode	TSnCMP7: Setting TSG2nO4 clear timing TSnCMP8: Setting TSG2nO4 set timing	TSnCMP9: Setting TSG2nO5 clear timing TSnCMP10: Setting TSG2nO5 set timing	TSnCMP11: Setting TSG2nO6 clear timing TSnCMP12: Setting TSG2nO6 set timing	See TSNCMP1W to TSnCMP11W.
HT-PWM mode	-	TSnCMP9: Setting TSG2nO5 clear timing and TSG2nO6 set timing TSnCMP10: Setting TSG2nO5 set timing and TSG2nO6 clear timing	-	See TSnCMP1W, TSnCMP5W, and TSnCMP9W.
SP-PWM mode	-	TSnCMP9: Setting TSG2nO5 clear timing and TSG2nO6 set timing TSnCMP10: Setting TSG2nO5 set timing and TSG2nO6 clear timing	-	See TSnCMP1W, TSnCMP5W, and TSnCMP9W.
120-DC mode	TSnCMP7, TSnCMP8: Selecting TSG2nO2, TSG2nO4, and TSG2nO6 outputs by TSnPAT1.	TSnCMP9, TSnCMP10: Selecting TSG2nO1, TSG2nO3, and TSG2nO5 outputs by TSnPAT0.	TSnCMP11, TSnCMP12: Selecting TSG2nO2, TSG2nO4, and TSG2nO6 outputs by TSnPAT1.	See TSnCMP1W to TSnCMP11W.

Table 15-38 Compare Register Functions in Each Mode (3/4)

Operating Mode	TSnDCMP0W	TSnDCMP2W	TSnCMPU	TSnCMPV
PWM mode	Setting diagnostic output or A/D conversion trigger timing	Setting diagnostic output or A/D conversion trigger timing	-	-
HT-PWM mode	Setting diagnostic output or A/D conversion trigger timing	Setting diagnostic output or A/D conversion trigger timing	The TSnCMPU set value is used as the TSnCMP1W (TSnCMP1, TSnCMP2) set value.	The TSnCMPV set value is used as the TSnCMP5W (TSnCMP5, TSnCMP6) set value.
SP-PWM mode	Setting diagnostic output or A/D conversion trigger timing	Setting diagnostic output or A/D conversion trigger timing	-	-
120-DC mode	Setting diagnostic output or A/D conversion trigger timing	Setting diagnostic output or A/D conversion trigger timing	-	-

Table 15-38 Compare Register Functions in Each Mode (4/4)

Operating Mode	TSnCMPW	TSnUPW	TSnVPW	TSnWPW
PWM mode	-	-	-	-
HT-PWM mode	The TSnCMPW set value is used as the TSnCMP9W (TSnCMP9, TSnCMP10) set value.	-	-	-
SP-PWM mode	-	The sum of the TSnUPW set value and the TSnCMP2 set value is used as the TSnCMP1 set value.	The sum of the TSnVPW set value and the TSnCMP6 set value is used as the TSnCMP5 set value.	The sum of the TSnWPW set value and the TSnCMP10 set value is used as the TSnCMP9 set value.
120-DC mode	-	-	-	-

15.5.3 Compare Register Rewrite Operation

The following compare registers are rewritten by reload (TSnCTL3.TSnRMC = 0) or anytime rewrite (TSnCTL3.TSnRMC = 1).

- TSnCMP0
- TSnCMP1 to TSnCMP12 (TSnCMP1W, TSnCMP3W, TSnCMP5W, TSnCMP7W, TSnCMP9W, TSnCMP11W)
- TSnPAT0W, TSnPAT1W
- TSnDTC0W, TSnDTC1W
- TSnDCMP0W, TSnDCMP2
- TSnCTL2, TSnCTL4
- TSnIOC3

Anytime rewrite mode

In this mode, the compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected immediately.

Caution

In HT-PWM mode and in anytime rewrite mode, if rewrite is performed again before a transfer to the buffer register is completed, the written value is not reflected immediately.
If the rewrite is performed while the 16-bit counter is counting up, the value is reflected at the next peak timing of the 16-bit sub-counter. If the rewrite is performed while the 16-bit counter is counting down, the value is reflected at the next valley timing of the 16-bit sub-counter.

Reload mode (simultaneous rewrite function)

Writing to TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW) enables reload (sets the reload request flag (TSnSTR0.TSnRSF)), and the values of all the pertinent registers are updated simultaneously at the next reload timing (reload).

The reload timing is the peak or valley timing of the 16-bit counter when the TSnTRG0.TSnTS bit is changed from 0 to 1. Reloading is controlled by TSnCTL4.TSnPRE and TSnVRE.

Writing to any register other than TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW) does not enable reloading.

Do not write to the registers to be reloaded until the next reload timing after reloading is enabled by writing to TSnCMP1 (TSnCMP1W, TSnCMPU, TSnUPW). The pertinent registers should be rewritten when the reload request flag (TSnSTR0.TSnRSF) is 0.

Rewriting registers to be reloaded by DMA transfer

Some of the registers to be reloaded can be rewritten by DMA transfer. DMA transfer is performed as follows.

Table 15-39 Example of DMA Transfer Order of Registers to be Reloaded

Address	Register Name	DMA Transfer Order (Example)
<TSG2n_base1> + 040_H	TSnCMP1W	
<TSG2n_base1> + 044_H	TSnCMP5W	
<TSG2n_base1> + 048_H	TSnCMP9W	
<TSG2n_base1> + $04C_H$	TSnCMP3W	
<TSG2n_base1> + 050_H	TSnCMP7W	
<TSG2n_base1> + 054_H	TSnCMP11W	
<TSG2n_base1> + 058_H	TSnCMP0	
<TSG2n_base1> + $05C_H$	TSnDCMP0W	
<TSG2n_base1> + 060_H	TSnDCMP2	
<TSG2n_base1> + 064_H	TSnPAT0W	
<TSG2n_base1> + 068_H	TSnPAT1W	
<TSG2n_base1> + $06C_H$	TSnDTC0W	
<TSG2n_base1> + 070_H	TSnDTC1W	

Table 15-40 Duty Setting in HT-PWM Mode

Address	Register Name	DMA Transfer Order (Example)
<TSG2n_base1> + $0B0_H$	TSnCMPU	
<TSG2n_base1> + $0B4_H$	TSnCMPV	
<TSG2n_base1> + $0B8_H$	TSnCMPW	

Table 15-41 Active Width Setting in SP-PWM Mode

Address	Register Name	DMA Transfer Order (Example)
<TSG2n_base1> + $0BC_H$	TSnCMP1W	
<TSG2n_base1> + $0C0_H$	TSnCMP5W	
<TSG2n_base1> + $0C4_H$	TSnCMP9W	

Note 1. TSnCTL2, TSnCTL4, and TSnIOC3 should be rewritten individually.
Note 2. Since writing to TSnCMP1W (including TSnCMP1, TSnCMPU, and TSnUPW) enables reloading, it should be rewritten after all the other registers to be reloaded have been rewritten (ready to be reloaded).

(1) Example of Operation in Anytime Rewrite Mode

In this mode, the values written to the compare registers (TSnCMP1 to TSnCMP12) are transferred to the internal buffer registers immediately, and are compared with the counter value.

The values are transferred to the internal compare buffer registers one clock cycle (PCLK) after being written to the compare registers (TSnCMP1 to TSnCMP12).

The transfer timing of the TSnCMP0 is the peak or valley timing (only in HT-PWM mode) of the 16-bit counter after being written to the compare registers, or at the match timing of the TSnCMP0 value with the 16-bit counter value (in any mode other than HT-PWM mode).

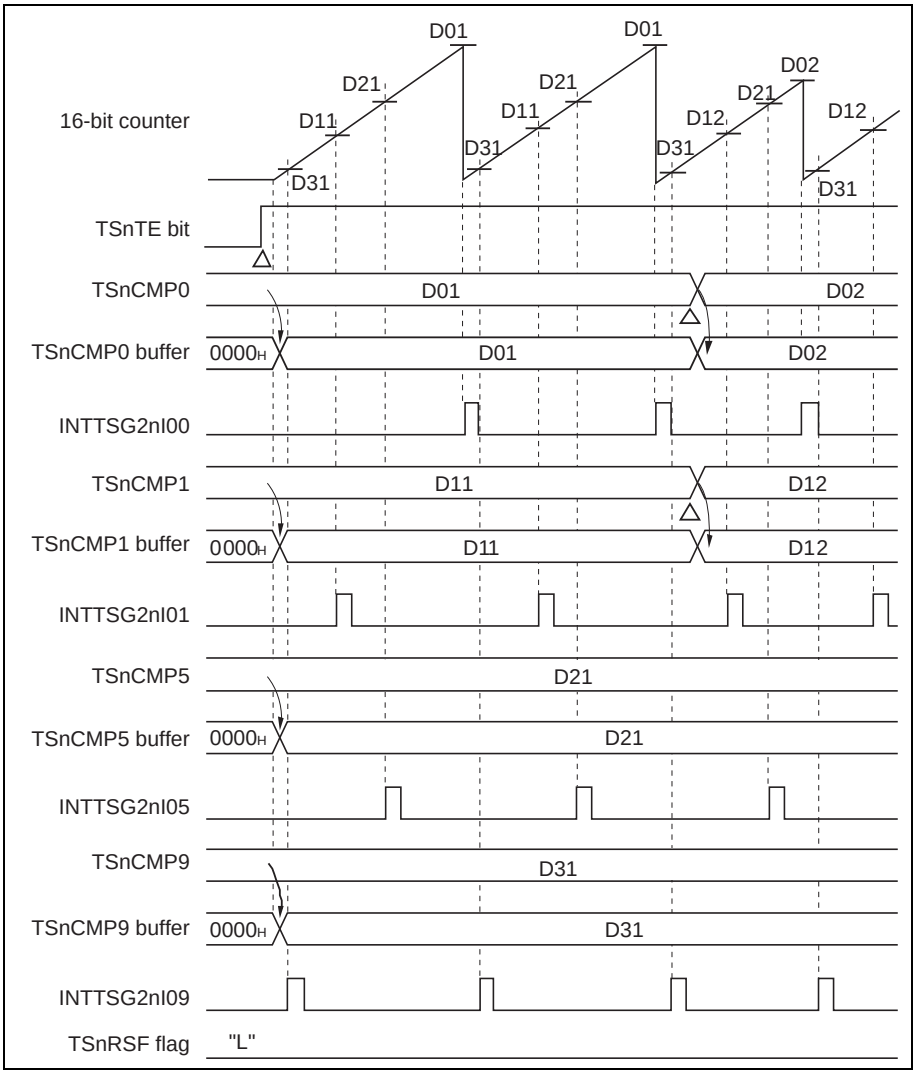


Figure 15-2 Anytime Rewrite Timing (Example in PWM Mode)

Note 1. D01, D02: TSnCMP0 setting value (0000_H to $FFFF_H$)
D11, D12: TSnCMP1 setting value (0000_H to $FFFF_H$)
D21: TSnCMP5 setting value (0000_H to $FFFF_H$)
D31: TSnCMP9 setting value (0000_H to $FFFF_H$

Note 2. Δ: Write access

(a) Data Reflection on PWM at Anytime Rewrite in HT-PWM Mode

In HT-PWM mode, the following output control is performed depending on the anytime rewrite timing of the compare registers.

- If anytime rewrite is performed after the inverse phase is cleared while the 16-bit counter is counting up (the 16-bit sub-counter is counting up), the positive phase is set if the rewritten value is TSnCMPm < TSnSBC. The inverse phase is set if the rewritten value is TSnCMPm > TSnCNT. (The dead time is inserted after clearance of the positive phase and inverse phase.)
- If anytime rewrite is performed after the positive phase is set while the 16-bit counter is counting up (the 16-bit sub-counter is counting up), the positive phase is cleared if the rewritten value is TSnCMPm > TSnSBC. The positive phase is cleared and the inverse phase is set if the rewritten value is TSnCMPm > TSnCNT. (The dead time is inserted after clearance of the positive phase and inverse phase.)
- If anytime rewrite is performed before the inverse phase is cleared while the 16-bit counter is counting up (the 16-bit sub-counter is counting up), the inverse phase is cleared if the rewritten value is TSnCMPm < TSnCNT. The inverse phase is cleared and the positive phase is set if the rewritten value is TSnCMPm < TSnSBC. (The dead time is inserted after clearance of the positive phase and inverse phase.)
- If anytime rewrite is performed after the positive phase is cleared while the 16-bit counter is counting up (the 16-bit sub-counter is counting down), the inverse phase is set if the rewritten value is TSnCMPm > TSnCNT. (The dead time is inserted after clearance of the positive phase.)
- If anytime rewrite is performed before the positive phase is cleared while the 16-bit counter is counting up (the 16-bit sub-counter is counting down), the positive phase is cleared if the rewritten value is TSnCMPm > TSnSBC. The positive phase is cleared and the inverse phase is set if the rewritten value is TSnCMPm > TSnCNT. (The dead time is inserted after clearance of the positive phase and inverse phase.)
- If anytime rewrite is performed after the positive phase is set while the 16-bit counter is counting up (the 16-bit sub-counter is counting up), the positive phase is cleared if the rewritten value is TSnCMPm > TSnSBC. The positive phase is cleared and the inverse phase is set if the rewritten value is TSnCMPm > TSnCNT. (The dead time is inserted after clearance of the positive phase and inverse phase.)
- If anytime rewrite is performed after the positive phase is set while the 16-bit counter is counting down (the 16-bit sub-counter is counting up), the positive phase is cleared if the rewritten value is TSnCMPm > TSnSBC. (The dead time is inserted after clearance of the positive phase.)
- If anytime rewrite is performed after the positive phase is set while the 16-bit counter is counting down (the 16-bit sub-counter is counting up), the positive phase is cleared if the rewritten value is TSnCMPm > TSnCNT. The positive phase is cleared and the inverse phase is set if the rewritten value is TSnCMPm > TSnSBC. (The dead time is inserted after clearance of the positive phase.)

(2) Example of Operation in Reload Mode (Simultaneous Rewrite Function)

The rewritten values of the registers to be reloaded (TSnCMP0 to TSnCMP12, TSnCTL2, TSnCTL3, TSnIOC3, TSnPAT0W, TSnPAT1W, TSnDTC0W, TSnDTC1W, TSnDCMP0W, and TSnDCMP2) can be transferred to the corresponding buffer registers simultaneously at the reload timing.

The registers should be rewritten when the pertinent reload request flag (TSnSTR0.TSnRSF) is 0.

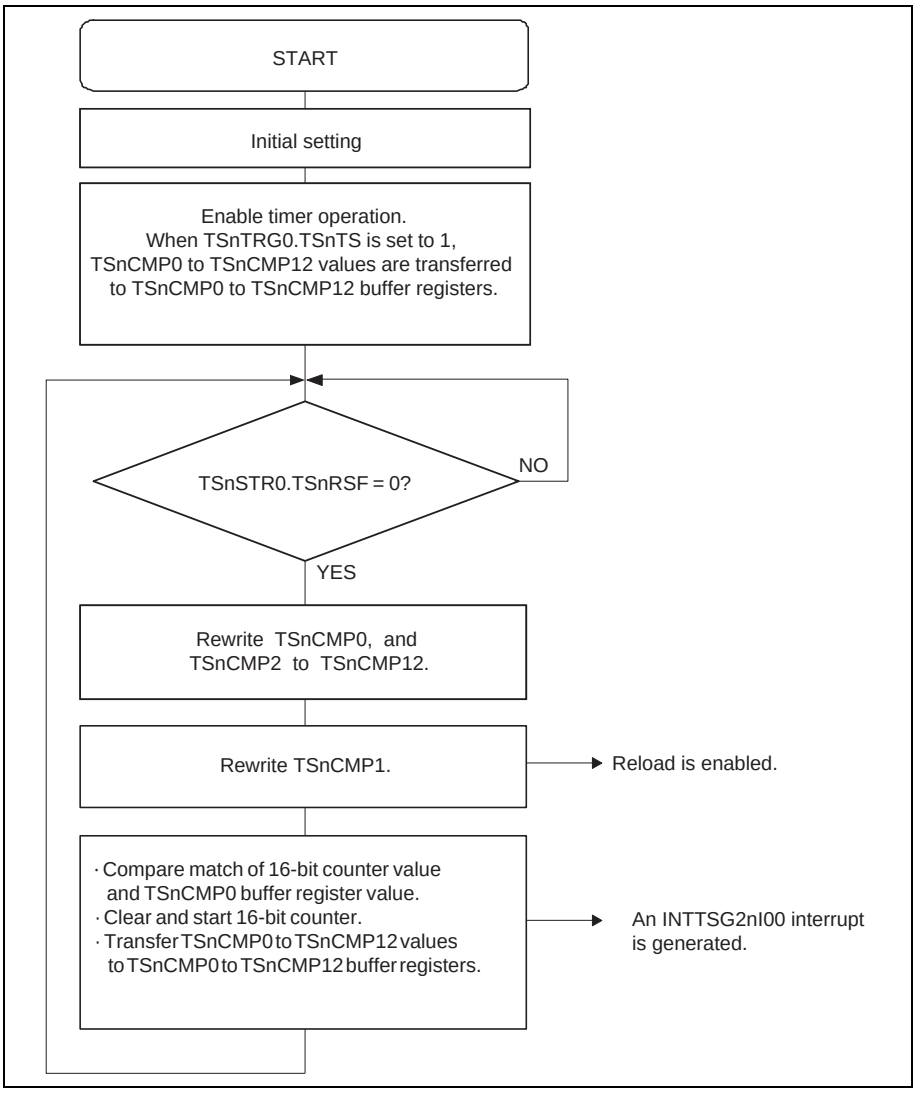


Figure 15-3 Basic Operation Flow in Reload Mode (Simultaneous Rewrite Function) (Example of PWM Mode)

Caution Writing to TSnCMP1 also enables reloading. Therefore, TSnCMP1 should be rewritten after TSnCMP0 and TSnCMP2 to TSnCMP12 registers have been rewritten.

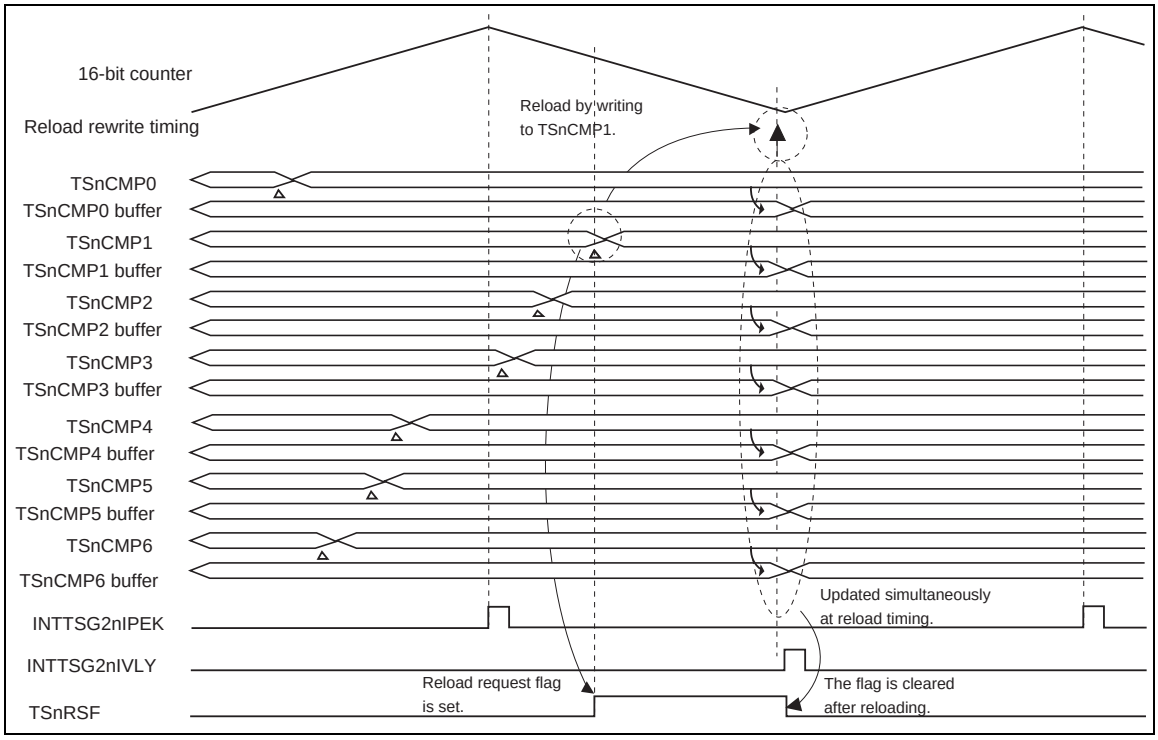


Figure 15-4 Simultaneous Rewrite Timing (Example of HT-PWM Mode)

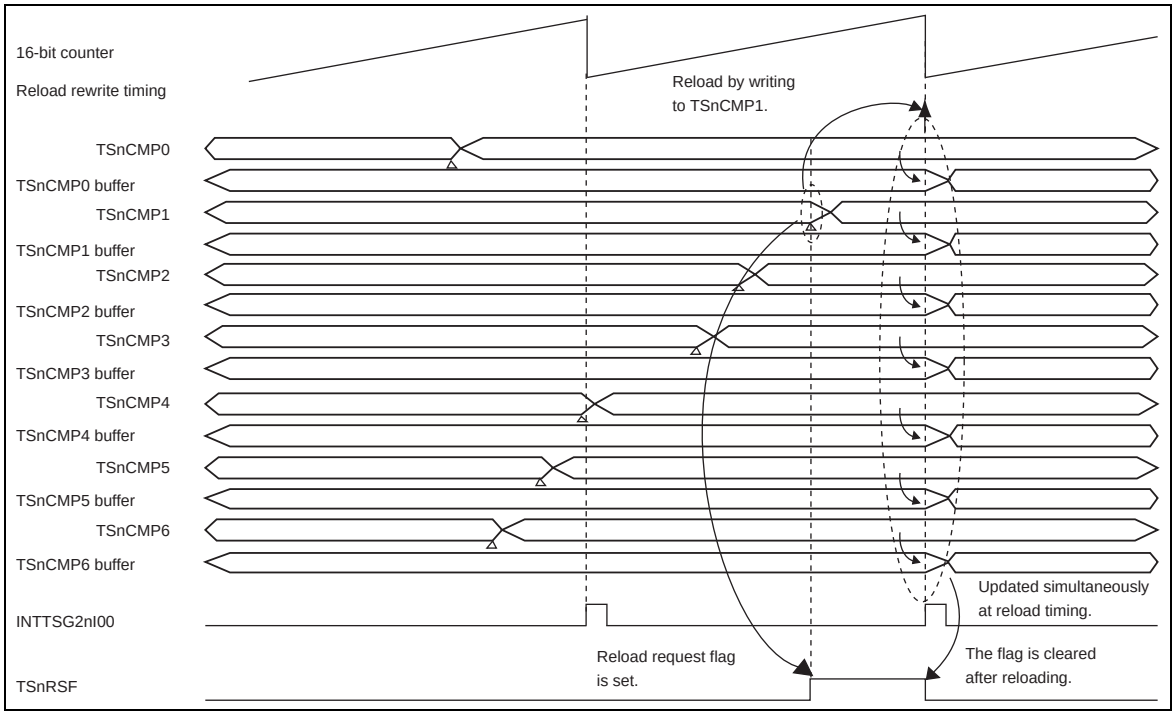


Figure 15-5 Simultaneous Rewrite Timing (Example of PWM Mode)

(a) Reload Rewrite Setting Example in Each Mode

Reloading conditions and setting examples are shown in Table 15-42 and Table 15-43.

Table 15-42 List of Reload Settings (when TSnCTL3.TSnRIA = 0)

Mode	TSnCTL4. TSnPRE	TSnCTL4. TSnVRE	TSnCTL4. TSnPIE	TSnCTL4. TSnVIE	TSnCTL4. TSnRCC4 to TSnRCC0	Reload
PWM mode	0	0/1	0/1	0/1	Any value	Setting prohibited
SP-PWM mode	1	0	0/1	0/1	Any value	When INTTSG2nI00 is generated.
120-DC mode	1	1	0/1	0/1	Any value	When INTTSG2nI00 is generated.
HT-PWM mode	0	0	0/1	0/1	Any value	Setting prohibited
	0	1	0/1	0/1	Any value	When INTTSG2nIVLY is generated.
	1	0	0/1	0/1	Any value	When INTTSG2nIPEK is generated.
	1	1	0/1	0/1	Any value	When INTTSG2nIPEK or INTTSG2nIVLY is generated.

Table 15-43 List of Reload Settings (when TSnCTL3.TSnRIA = 1)

Mode	TSnCTL4. TSnPRE	TSnCTL4. TSnVRE	TSnCTL4. TSnPIE	TSnCTL4. TSnVIE	TSnCTL4. TSnRCC4- TSnRCC0	Reload
PWM mode	0	0/1	0/1	0/1	Any value	Setting prohibited
SP-PWM mode	1	0	0	0/1	Any value	Setting prohibited
120-DC mode	1	0	1	0/1	Any value	When INTTSG2nI00 is generated.
	1	1	0	0/1	Any value	Setting prohibited
	1	1	1	0/1	Any value	When INTTSG2nI00 is generated.
HT-PWM mode	0	0	0/1	0/1	Any value	Setting prohibited
	0	1	0	0	Any value	Setting prohibited
	0	1	0	1	Any value	When INTTSG2nIVLY is generated.
	0	1	1	0	Any value	Setting prohibited
	0	1	1	1	Any value	When INTTSG2nIVLY is generated.
	1	0	0	0/1	Any value	Setting prohibited
	1	0	1	0/1	Any value	When INTTSG2nIPEK is generated.
	1	1	0	0	Any value	Setting prohibited
	1	1	0	1	Any value	When INTTSG2nIVLY is generated.
	1	1	1	0	Any value	When INTTSG2nIPEK is generated.
	1	1	1	1	Any value	When INTTSG2nIPEK or INTTSG2nIVLY is generated.

15.5.4 List of Outputs in Each Mode

(1) Timer Output in Each Mode

The list of timer outputs (TSG2nO0 to TSG2nO7 pins) in each mode is shown in Table 15-44.

Table 15-44 List of Timer Outputs in Each Mode (1/3)

Operating Mode	TSG2nO0 Pin	TSG2nO1 Pin	TSG2nO2 Pin
PWM mode	- (Fixed to low.)	Outputs a PWM signal by compare match of TSnCMP1W (TSnCMP1 and TSnCMP2).	Outputs a PWM signal by compare match of TSnCMP3W (TSnCMP3 and TSnCMP4).
HT-PWM mode	Outputs the status indicating whether the 16-bit counter or 16-bit sub-counter is incremented or decremented.	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP1W (TSnCMP1 and TSnCMP2).	Outputs an inverse phase PWM signal (with dead time) to TSG2nO1 pin.
SP-PWM mode	- (Fixed to low.)	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP1W (TSnCMP1 and TSnCMP2).	Outputs an inverse phase PWM signal (with dead time) to TSG2nO1 pin.
120-DC mode	- (Fixed to low.)	Outputs a PWM signal by TSnCMP1W (TSnCMP1 and TSnCMP2), TSnCMP5W (TSnCMP5 and TSnCMP6), and TSnCMP9W (TSnCMP9 and TSnCMP10).	Outputs a PWM signal by TSnCMP3W (TSnCMP3 and TSnCMP4), TSnCMP7W (TSnCMP7 and TSnCMP8), and TSnCMP11W (TSnCMP11 and TSnCMP12).

Table 15-44 List of Timer Outputs in Each Mode (2/3)

Operating Mode	TSG2nO3 Pin	TSG2nO4 Pin	TSG2nO5 Pin
PWM mode	Outputs a PWM signal by compare match of TSnCMP5W (TSnCMP5 and TSnCMP6).	Outputs a PWM signal by compare match of TSnCMP7W (TSnCMP7 and TSnCMP8).	Outputs a PWM signal by compare match of TSnCMP9W (TSnCMP9 and TSnCMP10).
HT-PWM mode	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP5W (TSnCMP5 and TSnCMP6).	Outputs an inverse phase PWM signal (with dead time) to TSG2nO3 pin.	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP9W (TSnCMP9 and TSnCMP10).
SP-PWM mode	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP5W (TSnCMP5, TSnCMP6)	Outputs an inverse phase PWM signal (with dead time) to TSG2nO3 pin.	Outputs a positive phase PWM signal (with dead time) by compare match of TSnCMP9W (TSnCMP9 and TSnCMP10).
120-DC mode	Outputs a PWM signal by TSnCMP1W (TSnCMP1 and TSnCMP2), TSnCMP5W (TSnCMP5 and TSnCMP6), and TSnCMP9W (TSnCMP9 and TSnCMP10).	Outputs a PWM signal by TSnCMP3W (TSnCMP3 and TSnCMP4), TSnCMP7W (TSnCMP7 and TSnCMP8), and TSnCMP11W (TSnCMP11 and TSnCMP12).	Outputs a PWM signal by TSnCMP1W (TSnCMP1 and TSnCMP2), TSnCMP5W (TSnCMP5 and TSnCMP6), and TSnCMP9W (TSnCMP9 and TSnCMP10).

Table 15-44 List of Timer Outputs in Each Mode (3/3)

Operating Mode	TSG2nO6 Pin	TSG2nO7 Pin
PWM mode	Outputs a PWM signal by compare match of TSnCMP11W (TSnCMP11 and TSnCMP12)	Outputs a diagnostic signal or A/D conversion trigger.
HT-PWM mode	Outputs an inverse phase PWM signal (with dead time) to TSG2nO5 pin.	Outputs a diagnostic signal or A/D conversion trigger.
SP-PWM mode	Outputs an inverse phase PWM signal (with dead time) to TSG2nO5 pin.	Outputs a diagnostic signal or A/D conversion trigger*.
120-DC mode	Outputs a PWM signal by TSnCMP3W (TSnCMP3 and TSnCMP4), TSnCMP7W (TSnCMP7 and TSnCMP8), and TSnCMP11W (TSnCMP11 and TSnCMP12).	Outputs a diagnostic signal or A/D conversion trigger*.

Note * For TSG2nO7, see section 15.5.4 (1) a TSG2nO7 Pin Output Control.

(a) TSG2nO7 Pin Output Control

The TSG2nO7 pin can output a pulse of A/D conversion trigger (TSnIOC1.TSnTGS = 0) or diagnostic output (TSnIOC1.TSnTGS = 1). When outputting a pulse of A/D conversion trigger, the TSG2nO7 pin is activated at the rising edge of the TSnADTRG0 signal, and inactivated at the rising edge of the TSnADTRG1 signal. When the TSnADTRG0 signal is detected while the TSG2nO7 pin is active, the TSG2nO7 pin remains active. When the TSnADTRG1 signal is detected while the TSG2nO7 pin is inactive, the TSG2nO7 pin remains inactive. If TSnADTRG0 and TSnADTRG1 signal triggers occur simultaneously, the TSG2nO7 pin is inactivated.

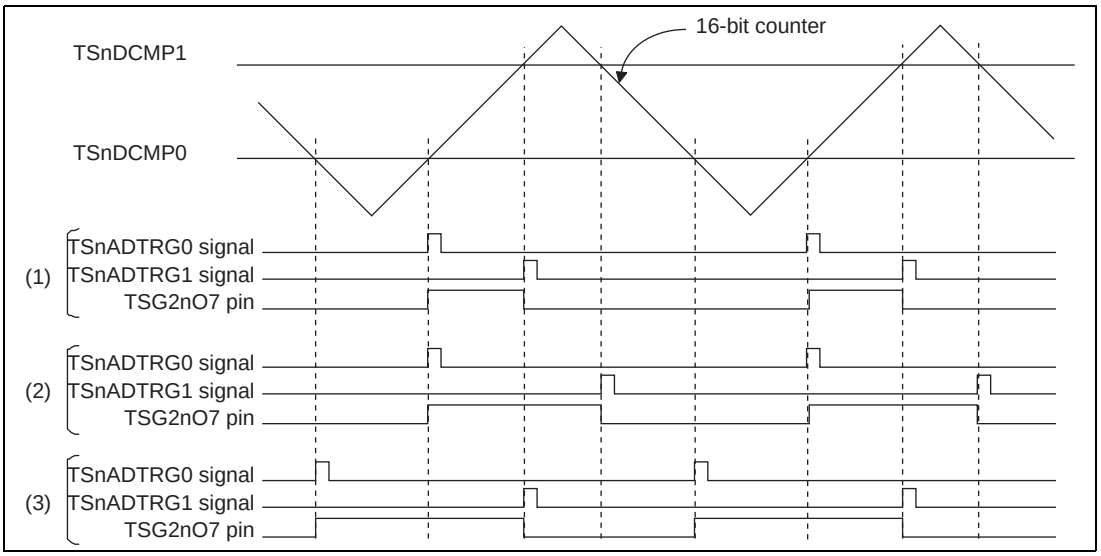


Figure 15-6 Example of A/D Trigger Output Timing of TSG2nO7 Pin (TSnIOC1.TSnTGS = 0)

Note (1) When TSnDCMP0 < TSnDCMP1, TSnCTL5 = 0004_H, and TSnCTL6 = 0010_H
(2) When TSnDCMP0 < TSnDCMP1, TSnCTL5 = 0004_H, and TSnCTL6 = 0020_H
(3) When TSnDCMP0 < TSnDCMP1, TSnCTL5 = 0008_H, and TSnCTL6 = 0010_H
For TSG2nO7, see section 15.5.4 (1) a TSG2nO7 Pin Output Control.

The TSG2nO7 pin during diagnostic output outputs the active level with the output width specified by TSnCTL0.TSnDWD at the match timing of the TSnDCMP0 to TSnDCMP2 values with the 16-bit counter value. If another match of the TSnDCMP0 to TSnDCMP2 values with the 16-bit counter value occurs consecutively within the output width specified by TSnDWD causing their active level widths to be overlapped, the TSG2nO7 pin outputs a pulse within 16 clock cycles (PCLK).

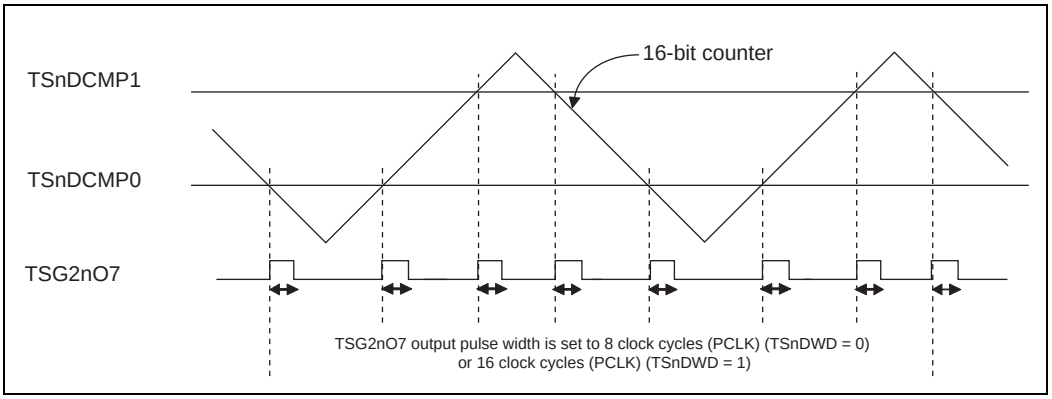


Figure 15-7 Example of TSG2nO7 Pin Diagnostic Pulse Output Timing (1) (TSnIOC1.TSnTGS = 1)

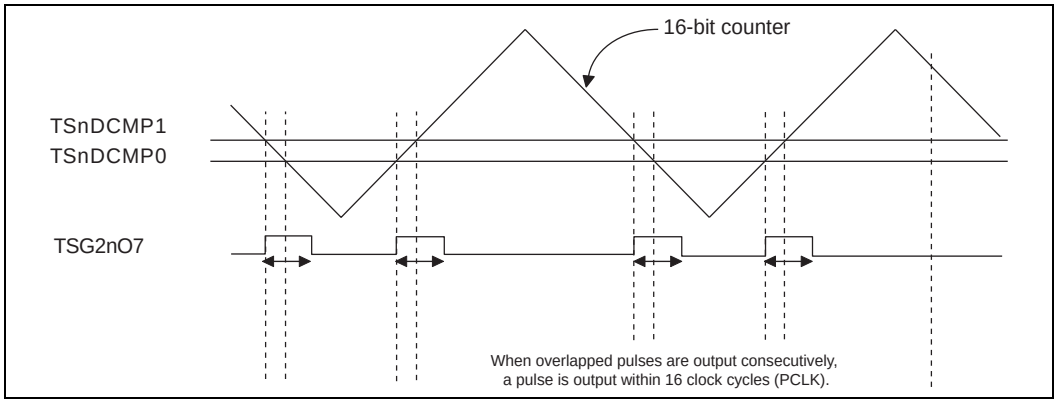


Figure 15-8 Example of TSG2nO7 Pin Diagnostic Pulse Output Timing (2) (with Pulse Output Width Overlapped)

(2) Interrupts in Each Mode

A list of interrupts (INTTSG2nI00 to INTTSG2nI12, INTTSG2nIPEK, INTTSG2nIVLY, INTTSG2nIER, and INTTSG2nIWN) in each mode is shown in Table 15-45.

Table 15-45 List of Interrupts in Each Mode (1/5)

Operating Mode	INTTSG2nI00	INTTSG2nI01	INTTSG2nI02	INTTSG2nI03
PWM mode	TSnCMP0 compare match interrupt	TSnCMP1 compare match interrupt*1	TSnCMP2 compare match interrupt*1	TSnCMP3 compare match interrupt*1
HT-PWM mode	Period interrupt	TSnCMP1 compare match interrupt*2	TSnCMP2 compare match interrupt*2	-
SP-PWM mode	TSnCMP0 compare match interrupt	TSnCMP1 compare match interrupt*1	TSnCMP2 compare match interrupt*1	-
120-DC mode	TSnCMP0 compare match interrupt	TSnCMP1 compare match interrupt*1	TSnCMP2 compare match interrupt*1	TSnCMP3 compare match interrupt*1

Note 1. When TSnCMP0 < TSnCMPm, a compare match interrupt is not generated (m = 1 to 12).

Note 2. When $0000_H \leq$ TSnCMPm < TSnDTC0, and (TSnCMP0 + TSnDTC0) < TSnCMPm, a compare match interrupt is not generated.

Table 15-45 List of Interrupts in Each Mode (2/5)

Operating Mode	INTTSG2nI04	INTTSG2nI05	INTTSG2nI06	INTTSG2nI07
PWM mode	TSnCMP4 compare match interrupt*1	TSnCMP5 compare match interrupt*1	TSnCMP6 compare match interrupt*1	TSnCMP7 compare match interrupt*1
HT-PWM mode	-	TSnCMP5 compare match interrupt*2	TSnCMP6 compare match interrupt*2	-
SP-PWM mode	-	TSnCMP5 compare match interrupt*1	TSnCMP6 compare match interrupt*1	-
120-DC mode	TSnCMP4 compare match interrupt*1	TSnCMP5 compare match interrupt*1	TSnCMP6 compare match interrupt*1	TSnCMP7 compare match interrupt*1

Note 1. When TSnCMP0 < TSnCMPm, a compare match interrupt is not generated (m = 1 to 12).

Note 2. When $0000_H \leq$ TSnCMPm < TSnDTC0, and (TSnCMP0 + TSnDTC0) < TSnCMPm, a compare match interrupt is not generated.

Table 15-45 List of Interrupts in Each Mode (3/5)

Operating Mode	INTTSG2nI08	INTTSG2nI09	INTTSG2nI10	INTTSG2nI11
PWM mode	TSnCMP8 compare match interrupt*1	TSnCMP9 compare match interrupt*1	TSnCMP10 compare match interrupt*1	TSnCMP7 compare match interrupt*1
HT-PWM mode	-	TSnCMP9 compare match interrupt*2	TSnCMP10 compare match interrupt*2	-
SP-PWM mode	-	TSnCMP9 compare match interrupt*1	TSnCMP10 compare match interrupt*1	-
120-DC mode	TSnCMP8 compare match interrupt*1	TSnCMP9 compare match interrupt*1	TSnCMP10 compare match interrupt*1	TSnCMP7 compare match interrupt*1

Note 1. When TSnCMP0 < TSnCMPm, a compare match interrupt is not generated (m = 1 to 12).

Note 2. When 0000_H ≤ TSnCMPm < TSnDTC0, and (TSnCMP0 + TSnDTC0) < TSnCMPm, a compare match interrupt is not generated.

Table 15-45 List of Interrupts in Each Mode (4/5)

Operating Mode	INTTSG2nI12	INTTSG2nIPEK	INTTSG2nIVLY
PWM mode	TSnCMP12 compare match interrupt*	Peak interrupt at the same timing as INTTSG2nI00	-
HT-PWM mode	-	Peak interrupt	Valley interrupt
SP-PWM mode	-	Valley interrupt at the same timing as INTTSG2nI00	-
120-DC mode	TSnCMP12 compare match interrupt*	Peak interrupt at the same timing as INTTSG2nI00	-

Note * When TSnCMP0 < TSnCMPm, a compare match interrupt is not generated (m = 1 to 12).

Table 15-45 List of Interrupts in Each Mode (5/5)

Operating Mode	INTTSG2nIER	INTTSG2nIWN
PWM mode	Error interrupt	Warning interrupt
HT-PWM mode	Error interrupt	Warning interrupt
SP-PWM mode	Error interrupt	Warning interrupt
120-DC mode	Error interrupt	Warning interrupt

15.6 Match Interrupt

The TSG2n can generate interrupts such as a compare match interrupt (INTTSG2nIm), a peak interrupt (INTTSG2nIPEK), and a valley interrupt (INTTSG2nIVLY). For an error interrupt and warning interrupt (INTTSG2nIER and INTTSG2nIWN), see section 15.10, Error/Warning Interrupt.

A period interrupt (INTTSG2nI00) is generated for each timer period. In HT-PWM mode, it is generated when the TSnDTC0 buffer register value matches with the 16-bit counter value. When the 16-bit counter performs sawtooth wave operation (PWM mode, SP-PWM mode, and 120-DC mode), it is generated after the 16-bit counter value has matched with the TSnCMP0 buffer register value.

A compare-match interrupt (INTTSG2nIm) is generated by a match of the TSnCMPm buffer register value with the 16-bit counter value depending on the compare register to be used in each operating mode (m = 01 to 12).

A peak interrupt (INTTSG2nIPEK) is generated in all the modes. In HT-PWM mode, it is generated when the 16-bit counter switches from incrementing to decrementing. When the 16-bit counter performs sawtooth wave operation (PWM mode, SP-PWM mode, and 120-DC mode), it is generated after the 16-bit counter value has matched with the TSnCMP0 buffer register value (the same timing as an INTTSG2nI00 interrupt).

A valley interrupt (INTTSG2nIVLY) is generated when the 16-bit counter switches from decrementing to incrementing in HT-PWM mode.

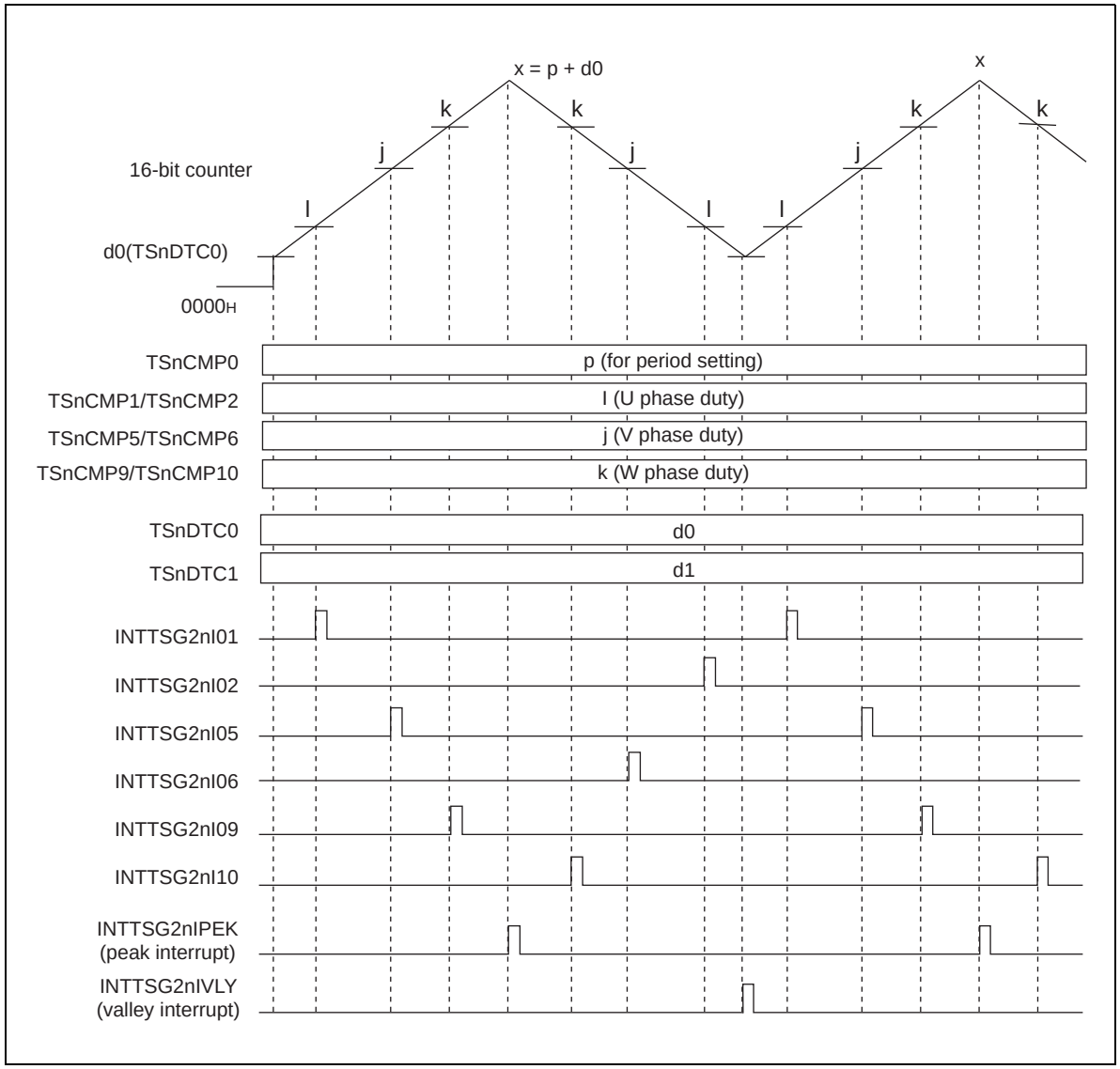


Figure 15-9 Interrupt Generation Example (Example of HT-PWM Mode)

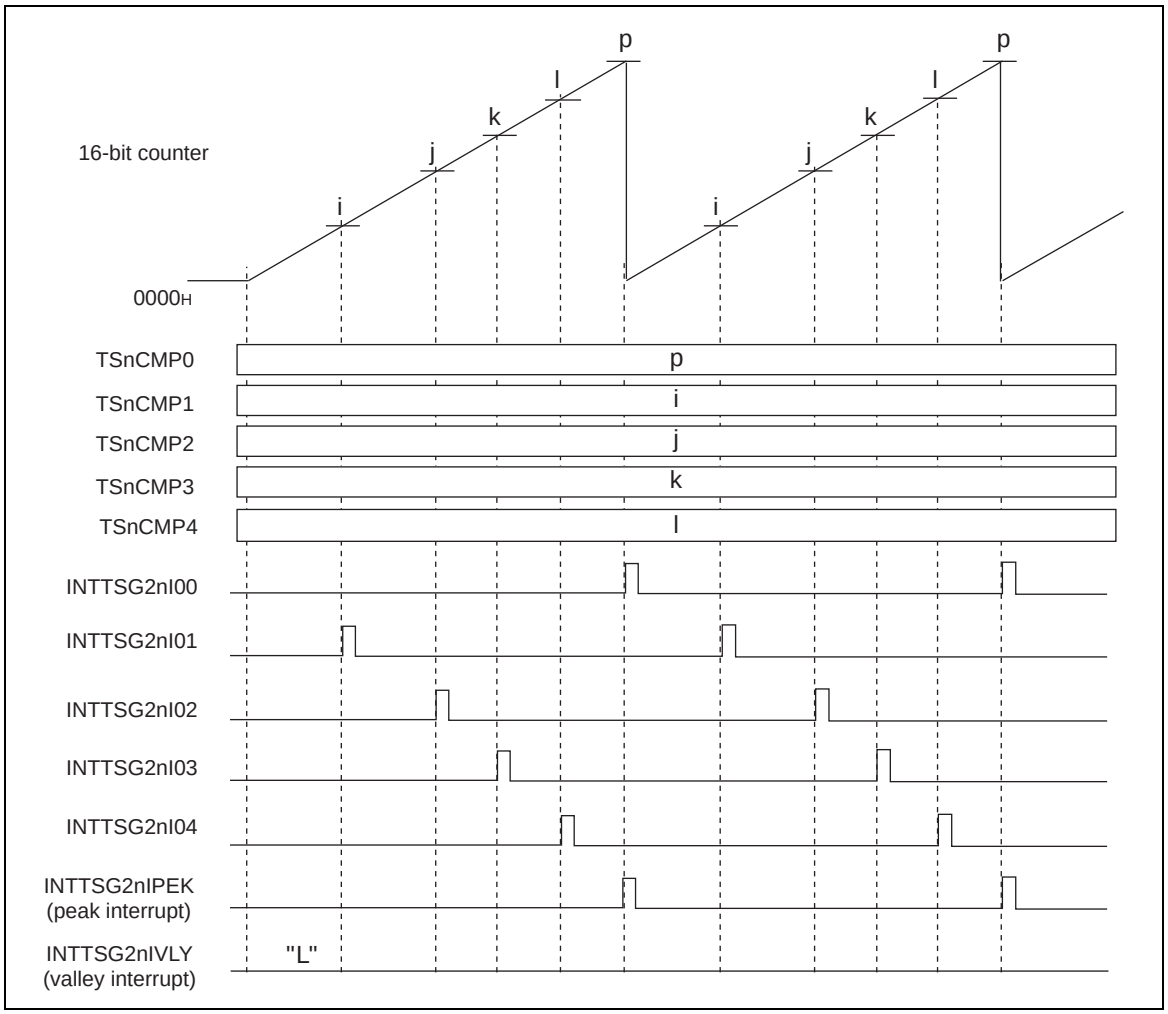


Figure 15-10 Interrupt Generation Example (Example of PWM Mode)

15.7 Flags

Table 15-46 List of Flags

No.	Flag Name	Symbol	Register	Operating Mode
(1)	Up count flag	TSnCUF	TSnSTR0	HT-PWM mode
		TSnSUF	TSnSTR0	
(2)	Positive phase and inverse phase simultaneous active state detection flag	TSnTBF0 to TSnTBF2	TSnSTR2	All operating modes
(3)	Reload request flag	TSnRSF	TSnSTR0	All operating modes
(4)	Noise detection flag	TSnNDF	TSnSTR2	All operating modes
(5)	Pattern order detection flag	TSnTSF	TSnSTR1	All operating modes
(6)	Pattern error detection flag	TSnPEF	TSnSTR2	All operating modes
(7)	Pattern reversal detection flag	TSnPRF	TSnSTR2	All operating modes
(8)	TSG2nPTSI2 to TSG2nPTSI0 pin abnormal toggle detection flag	TSnPTF	TSnSTR2	All operating modes
(9)	TSnSTCI0 and TSnSTCI1 signal simultaneous trigger detection flag	TSnTDF	TSnSTR2	All operating modes
(10)	Pattern phase difference detection flag	TSnPPF	TSnSTR2	All operating modes
(11)	Timer output pattern flag	TSnOPF0 to TSnOPF2	TSnSTR1	All operating modes
(12)	Pattern switch detection signal (internal signal)	TSnPTE	-	All operating modes

15.7.1 Up Count Flag (TSnCUF and TSnSUF)

Name Up count flag (TSnSTR0.TSnCUF and TSnSUF)

Description There are following two up count flags.

TSnCUF is an up/down count flag of the 16-bit counter.

TSnSUF is an up/down count flag of the 16-bit sub-counter.

For both TSnCUF and TSnSUF, 0 means increment, and 1 means decrement.

TSnCUF and TSnSUF can be used only in HT-PWM mode.

Example of operation

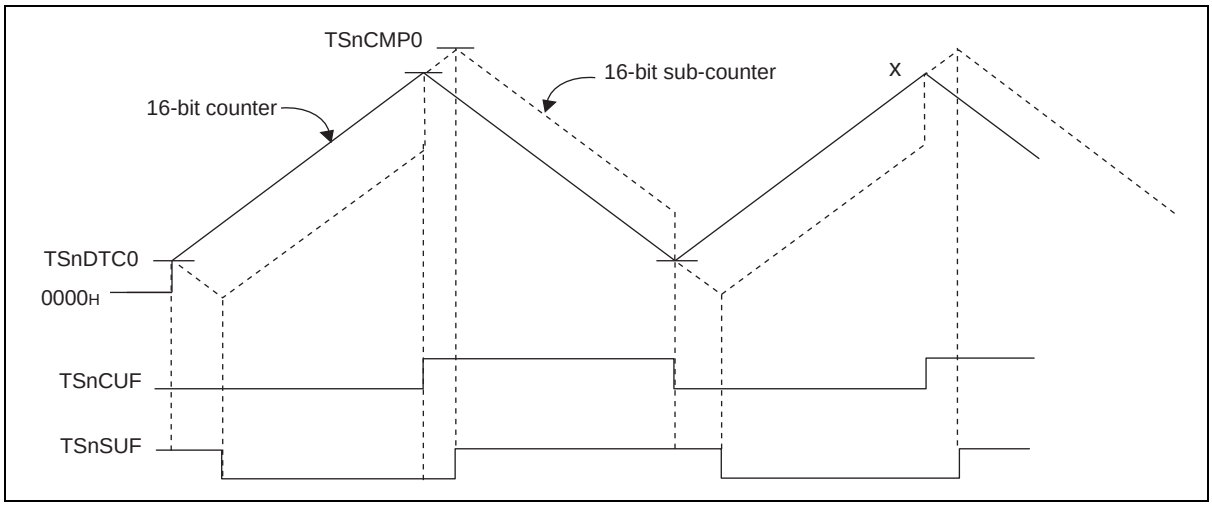


Figure 15-11 Example of Up Count Flag Operation

Note 1. TSnCUF value is:
0 (up count) when TSnDTC0 ≤ 16-bit counter ≤ (TSnCMP0 + TSnDTC0-2)
1 (down count) when (TSnCMP0 + TSnDTC0-2) ≥ 16-bit counter ≥ TSnDTC0 + 2

Note 2. TSnSUF value is:
0 (up count) when 0 ≤ 16-bit sub-counter ≤ (TSnCMP0 + TSnDTC0 + TSnDTC1-2)
1 (down count) when (TSnCMP0 + TSnDTC0 + TSnDTC1) ≥ 16-bit sub-counter ≥ 2

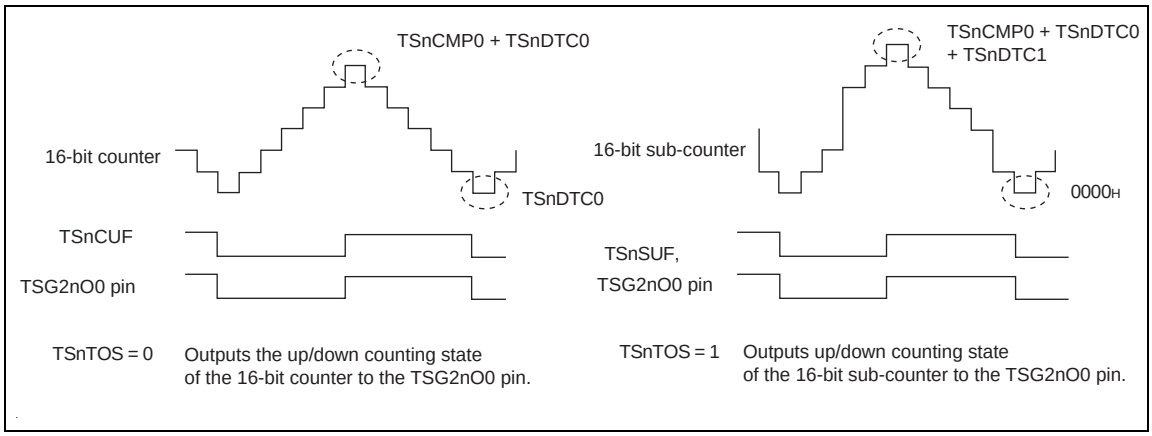


Figure 15-12 TSG2nO0 Pin Output depending onTSnIOC1.TSnTOS Setting

Operating mode TSnCUF and TSnSUF can be used only in HT-PWM mode.

15.7.2 Positive Phase and Inverse Phase Simultaneous Active State Detection Flag (TSnTBF0 to TSnTBF2)

Name Positive phase and inverse phase simultaneous active state detection flag (TSnSTR2.TSnTBF0 to TSnTBF2 flags)

Description When any of TSnCTL1.TSnTBA2 to TSnTBA0 is 1, TSnTBF0 to TSnTBF2 can detect the simultaneous active state of the positive phase and inverse phase of TSG2n.

When the simultaneous active state of the positive phase and inverse phase of the TSG2n is detected, the corresponding TSnTBF0 to TSnTBF2 flags are set to 1, and an error interrupt (INTTSG2nIER) is generated. The flags are cleared when 1 is written to TSnSTC.TSnTBR0 to TSnTBR2, respectively.

Example of operation

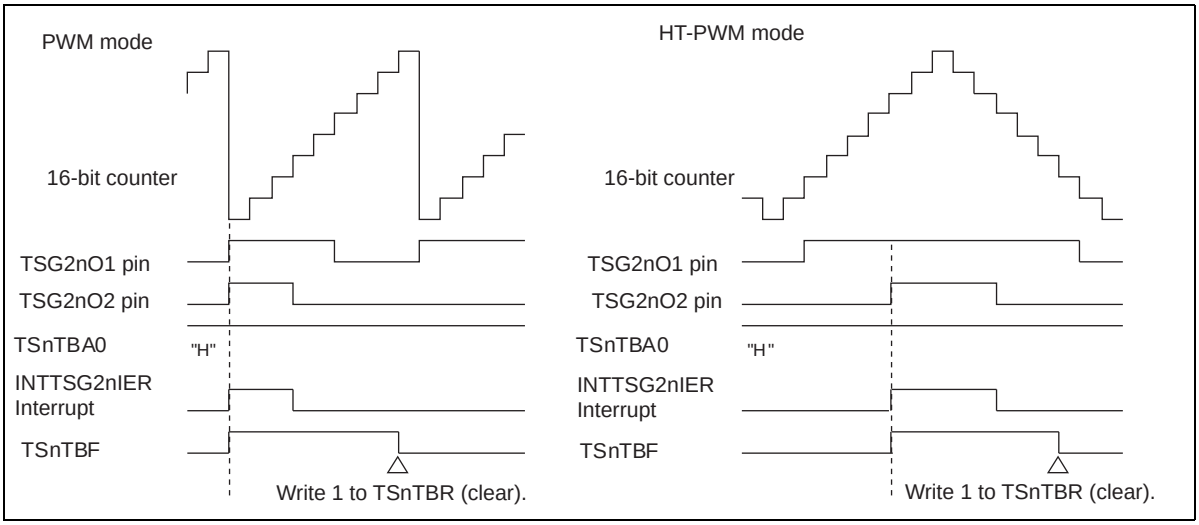


Figure 15-13 Example of Positive Phase and Inverse Phase Simultaneous Active State Detection Flag Operation

Operating mode TSnTBF0 to TSnTBF2 can be used in all operating modes.

Caution TSnTBF0 to TSnTBF2 are valid only when TSnCTL1.TSnTBA0 to TSnTBA2 = 1 and TSnSTR0.TSnTE = 1.

15.7.3 Reload Request Flag (TSnRSF)

Name Reload request flag (TSnSTR0.TSnRSF)

Description TSnRSF is set to 1 when a reload request is generated (when a value is written to TSnCMP1 (TSnCMP1W, TSnCMPU, TSnCPW)), and cleared to 0 when the value is transferred to all the buffer registers.

Example of operation

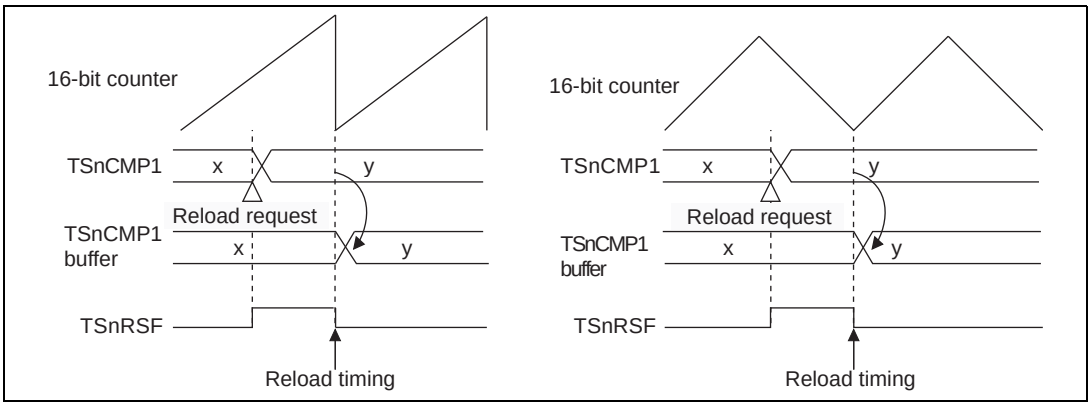


Figure 15-14 Example of Reload Request Flag Operation

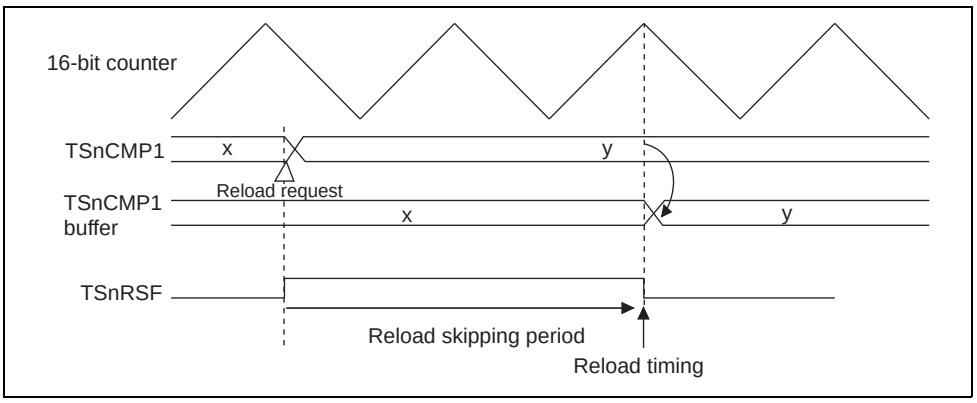


Figure 15-15 Reload Request Flag and Reload Skipping Period

Operating Mode TSnRSF can be used in all operating modes.

15.7.4 Noise Detection Flag (TSnNDF)

Name Noise detection flag (TSnSTR2.TSnNDF)

Description TSnNDF can detect that two or more pins of TSG2nPTSI2 to TSG2nPTSI0 have changed simultaneously (a noise is generated).

TSnNDF is set to 1 when two or more pins of TSG2nPTSI2 to TSG2nPTSI0 have changed simultaneously (a noise is generated), and a warning interrupt (INTTSG2nIWN) is generated. The TSnNDF flag is cleared to 0 when 1 is written to the TSnSTC.TSnNDR bit.

Example of operation

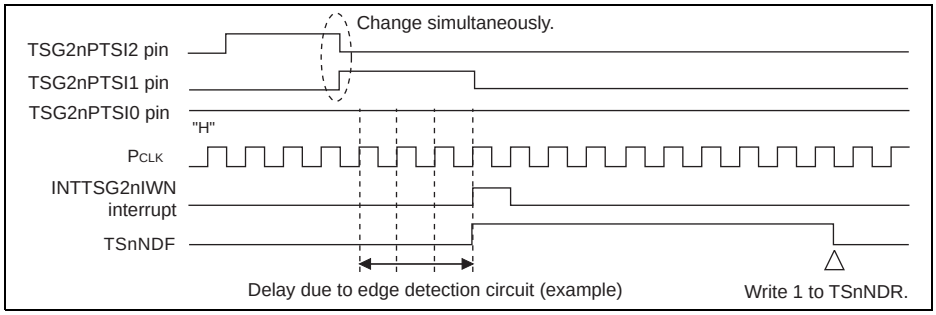


Figure 15-16 Example of Noise Detection Flag Operation

Operating mode TSnNDF can be used in all operating modes

Caution TSnNDF is valid only when TSnCTL1.TSnNDC = 1 and TSnSTR0.TSnTE = 1.

15.7.5 Pattern Order Detection Flag (TSnTSF)

Name Pattern order detection flag (TSnSTR1.TSnTSF)

Description TSnTSF can detect the order of patterns input to the TSG2nPTSI2 to TSG2nPTSI0 pins.

TSnTSF is set depending on the values input to the TSG2nPTSI2 to TSG2nPTSI0 pins as shown in the table below.

TSnTSF	Values input to TSG2nPTSI2 to TSG2nPTSI0 Pins
0	[1, 0, 1] → [1, 0, 0] → [1, 1, 0] → [0, 1, 0] → [0, 1, 1] → [0, 0, 1]
1	[1, 0, 1] ← [1, 0, 0] ← [1, 1, 0] ← [0, 1, 0] ← [0, 1, 1] ← [0, 0, 1]

Example of operation

(a) When Normal Input to TSG2nPTSI2 to TSG2nPTSI0 Pins is Detected

As shown in Figure 15-17, if the TSG2nPTSI2 to TSG2nPTSI0 pins change in the normal order, 0 or 1 is set according to the change order at the change timing.

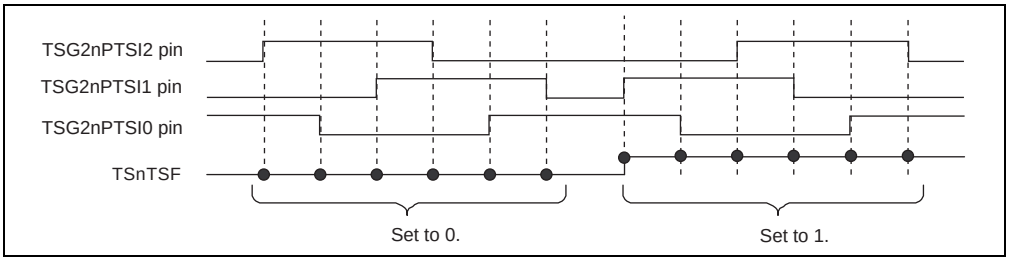


Figure 15-17 Example of Pattern Order Detection Flag Operation (Normal Operation)

(b) Detection of Input Pattern Order

Immediately after TSG2n starts operation, the rotation direction cannot be determined. Therefore, TSnTSF cannot detect the change (normal or reverse rotation) in the patterns input to the TSG2nPTSI2 to TSG2nPTSI0 pins. To enable detection of change immediately after the beginning of operation, TSnPSC should be set before operation starts (when TSnTE = 0, the TSnPSC value is reflected).

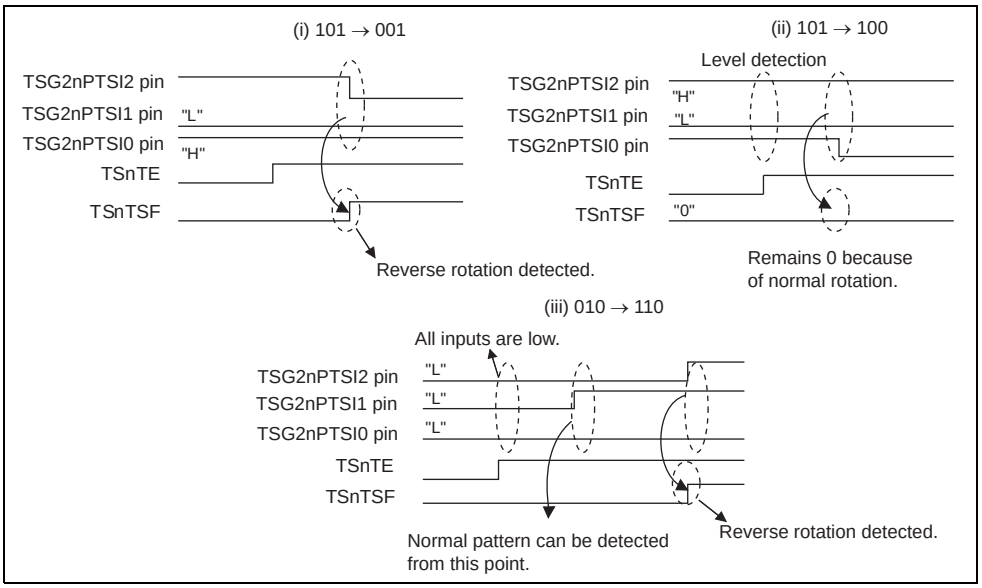


Figure 15-18 Example of Detecting Change (Normal/Reverse Rotation) in Pattern Input to TSG2nPTSI2 to TSG2nPTSI0 Pins

(c) When Abnormal Input to TSG2nPTSI2 to TSG2nPTSI0 Pins is Detected

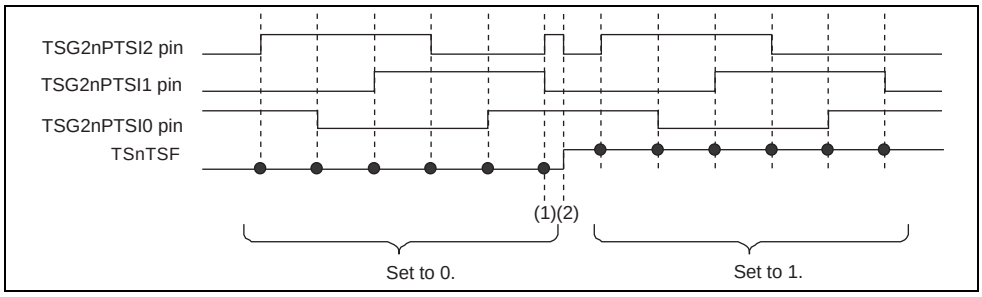


Figure 15-19 Example of Operation when Values Input to Two Pins of TSG2nPTSI2 to TSG2nPTSI0 Change (Abnormal Operation)

(1) TSnTSF does not change at this point because it expects the input pattern change to {0, 1, 0} or {0, 0, 1} (if values of two pins change, TSnTSF does not change).

(2) TSG2nPTSI2 to TSG2nPTSI0 pins are determined to have been changed from {1, 0, 1} to {0, 0 ,1}, and TSnTSF is set to 1.

Operating Mode TSnTSF can be used in all operating modes.

15.7.6 Pattern Error Detection Flag (TSnPEF)

Name Pattern error detection flag (TSnSTR2.TSnPEF)

Description TSnPEF can detect that 000 or 111 is input to the TSG2nPTSI2 to TSG2nPTSI0 pins.

TSnPEF is set to 1 when the levels of the TSG2nPTSI2 to TSG2nPTSI0 pins are 111 or 000, and a warning interrupt (INTTSG2nIWN) is generated. TSnPEF is cleared to 0 when 1 is written to TSnSTC.TSnPER.

Example of operation

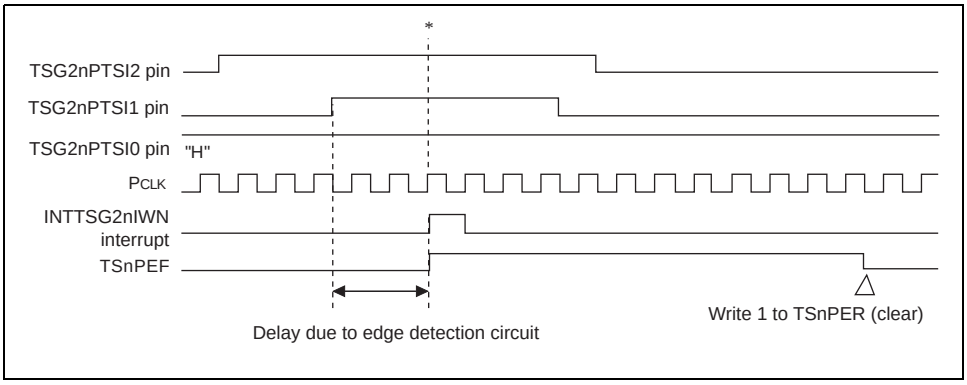


Figure 15-20 Example of Pattern Error Detection Flag Operation (TSG2nPTSI2 to TSG2nPTSI0 Pins = 111)

Note * 111 is detected.

Operating mode TSnPEF can be used in all operating modes.

Caution TSnPEF is valid only when TSnCTL1.TSnPEC = 1 and TSnSTR0.TSnTE = 1.

15.7.7 Pattern Reversal Detection Flag (TSnPRF)

Name Pattern reversal detection flag (TSnSTR2.TSnPRF)

Description TSnPRF can detect that the pattern change order of the TSG2nPTSI2 to TSG2nPTSI0 pins have been reversed.

TSnPRF is set to 1 when the pattern order detection flag (TSnTSF) changes, and a warning interrupt (INTTSG2nIWN) is generated. However, immediately after TSnSTR0.TSnTE is set to 1, TSnPRF is valid at the timing of the second and subsequent change in TSG2nPTSI2 to TSG2nPTSI0 pins.

TSnPRF is cleared to 0 when 1 is written to the TSnSTC.TSnPRR bit.

Example of operation

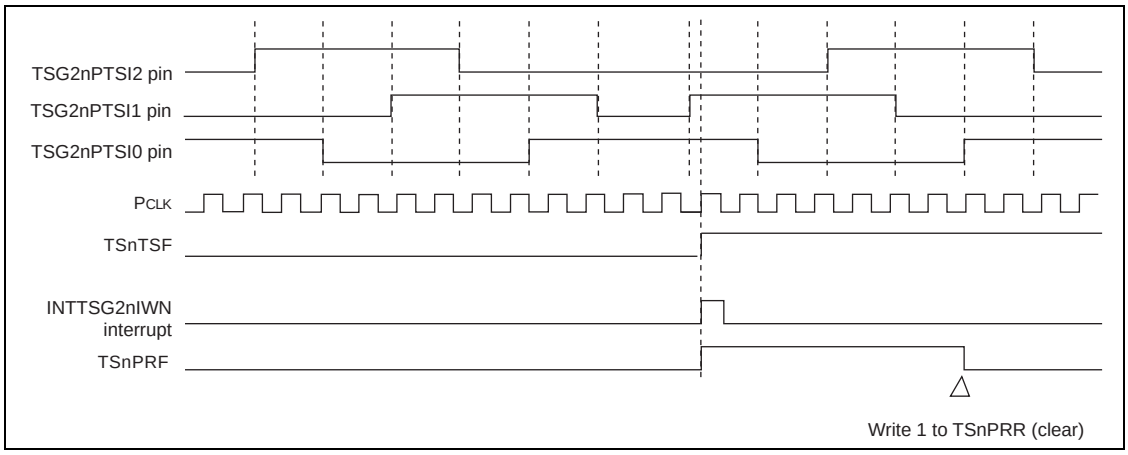


Figure 15-21 Example of Pattern Reversal Detection Flag Operation

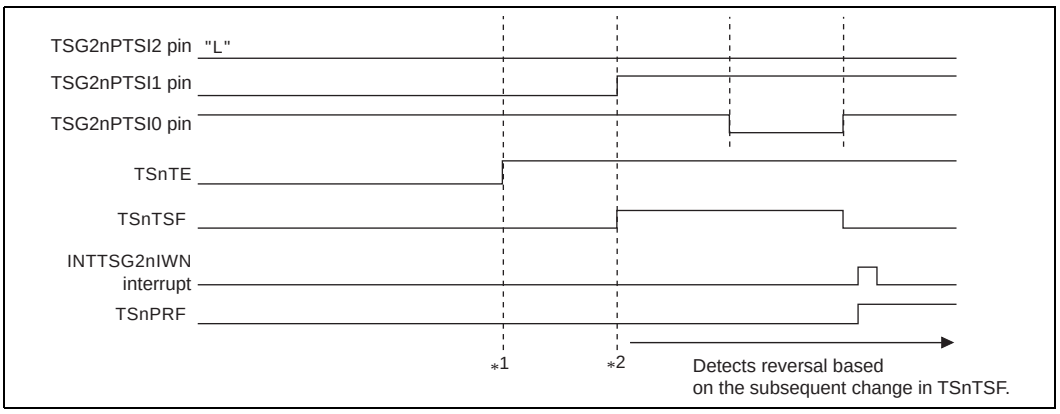


Figure 15-22 Example of Operation immediately after TSnTE Flag in TSnSTR0 is Set to 1

Note 1. Operation starts.
Note 2. If TSnTSF is set to 1 by the first change in the TSG2nPTSI2 to TSG2nPTSI0 pins immediately after TSnTE is set to 1, reversal is not detected.

Operating mode TSnPRF can be used in all operating modes.

Caution TSnPRF is valid only when TSnCTL1.TSnPRC = 1 and TSnSTR0.TSnTE = 1.

15.7.8 TSG2nPTSI2 to TSG2nPTSI0 Pin Abnormal Toggle Detection Flag (TSnPTF)

Name TSG2nPTSI2 to TSG2nPTSI0 pin abnormal toggle detection flag (TSnSTR2.TSnPTF)

Description TSnPTF can detect that the values of the TSG2nPTSI2 to TSG2nPTSI0 pins change three or more times during the TSnOPCI0 or TSnOPCI1 signal trigger.

TSnPTF is set to 1 when the third trigger of TSnOPCI0 or TSnOPCI1 signal occurs simultaneously with the change in TSG2nPTSI2 to TSG2nPTSI0, and a warning interrupt (INTTSG2nIWN) is generated.

TSnPTF is cleared to 0 when 1 is written to TSnSTC.TSnPTR.

Example of operation

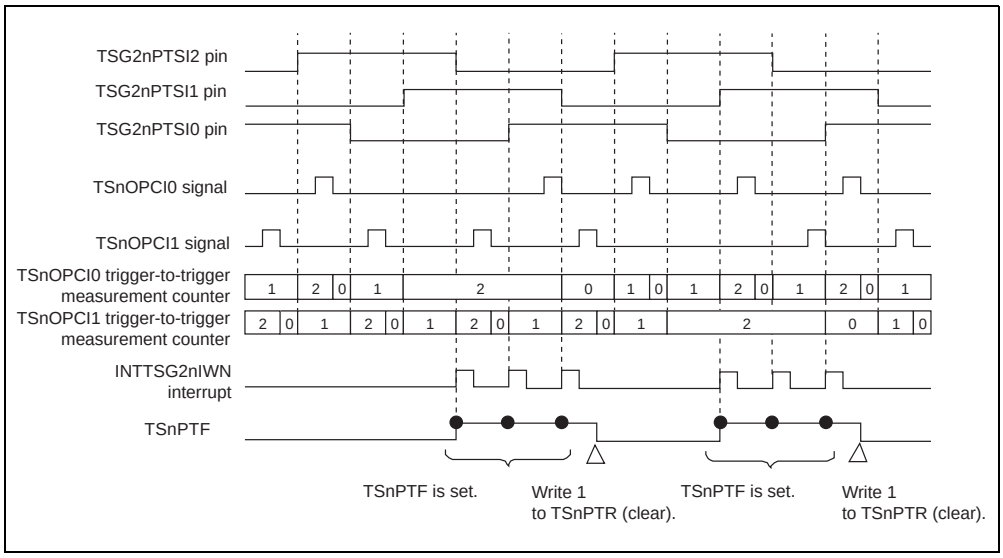


Figure 15-23 Example of TSG2nPTSI2 to TSG2nPTSI0 Pin Abnormal Toggle Detection Flag Operation

Operating mode TSnPTF can be used in all operating modes.

Caution 1. TSnPTF is valid only when TSnCTL1.TSnPTC1 bit = 1 and TSnSTR0.TSnTE = 1.

Caution 2. When TSnPTC0 bit = 1 and TSnPTC1 bit = 1, TSG2nO1 to TSG2nO6 pin output switch control is automatically switched to pattern switch method (TSnOPT0.TSnPOT bit = 0) if an abnormal toggle is detected.

15.7.9 TSnSTCI0 and TSnSTCI1 Signal Simultaneous Trigger Detection Flag (TSnTDF)

Name TSnOPCI0 and TSnOPCI1 signal simultaneous trigger detection flag (TSnSTR2.TSnTDF)

Description TSnTDF can detect that TSnOPCI0 and TSnOPCI1 signals are generated simultaneously.

TSnTDF is set to 1 when the TSnOPCI0 and TSnOPCI1 signals are generated simultaneously, and a warning interrupt (INTTSG2nIWN) is generated. TSnTDF is cleared to 0 when 1 is written to TSnSTC.TSnTDR.

Example of operation

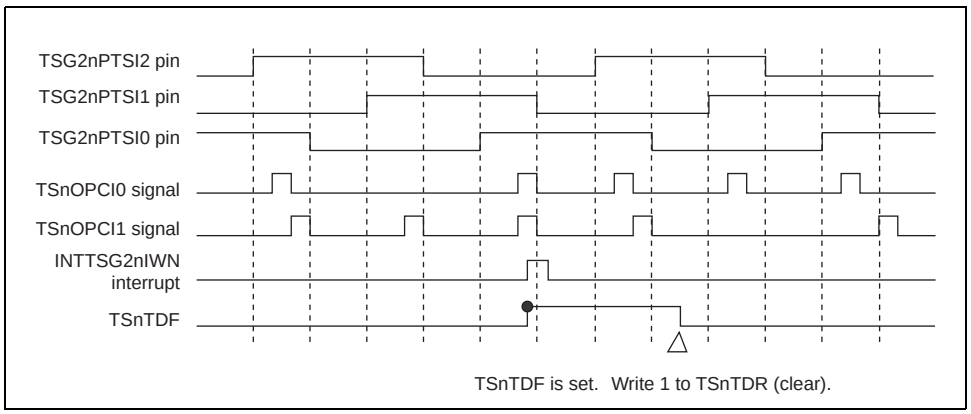


Figure 15-24 Operation of TSG2nPTSI2 to TSG2nPTSI0 Pin Abnormal Toggle Detection Flag Operation

Operating mode TSnTDF can be used in all operating modes.

Caution TSnTDF is valid only when TSnCTL1.TSnTDC = 1 and TSnSTR0.TSnTE = 1.

15.7.10 Pattern Phase Difference Detection Flag (TSnPPF)

Name Pattern phase difference detection flag (TSnSTR2.TSnPPF)

Description TSnPPF can detect the phase difference between the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) and the output pattern (TSnSTR1.TSnOPF2 to TSnOPF0 flags).

TSnPPF is set to 1 when the pattern phase difference is detected when the TSnOPCI0 and TSnOPCI1 signal triggers are input, and a warning interrupt (INTTSG2nIWN) is generated. TSnPPF remains 1 until it is cleared to 0 when 1 is written to TSnSTC.TSnPPR by software. When the phase difference is detected, TSnPPF is set at each operation clock cycle (PCLK). TSnPPF should be cleared to 0 when no phase difference occurs.

Table 15-47 Correspondence between Normal Input Patterns and Output Patterns

TSG2nPTSI2 to TSG2nPTSI0 pins (input)	"1, 0, 1"	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"	"0, 0, 1"
TSnOPF2 to TSnOPF0 flags (output)	"0, 0, 1" "1, 0, 1" "1, 0, 0"	"1, 0, 1" "1, 0, 0" "1, 1, 0"	"1, 0, 0" "1, 1, 0" "0, 1, 0"	"1, 1, 0" "0, 1, 0" "0, 1, 1"	"0, 1, 0" "0, 1, 1" "0, 0, 1"	"0, 1, 1" "0, 0, 1" "1, 0, 1"

Example of operation

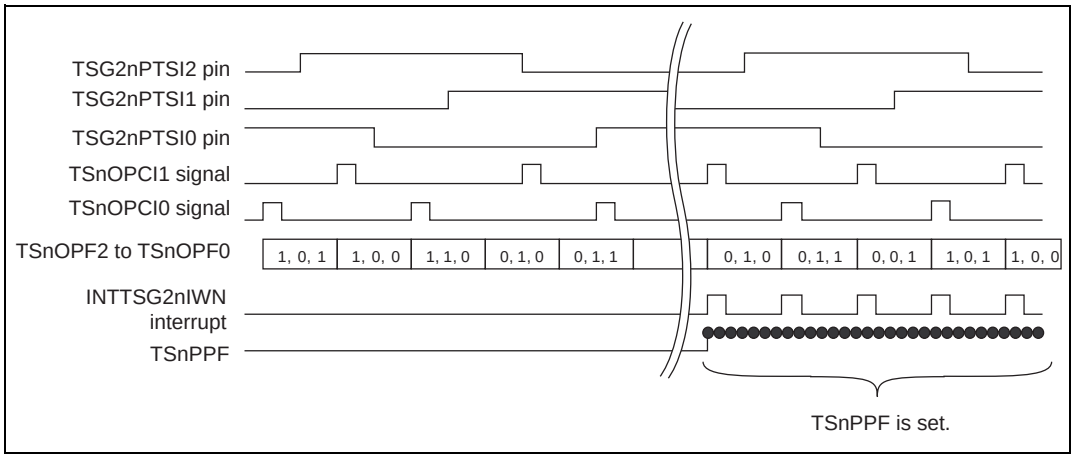


Figure 15-25 Example of Pattern Difference Detection Flag Operation

Operating mode TSnPPF can be used in all operating modes.

Caution 1. TSnPPF is valid only when TSnCTL1.TSnPPC = 1 and TSnSTR0.TSnTE = 1.
Caution 2. When 000 or 111 is input to the TSG2nPTSI2 to TSG2nPTSI0 pins, or when TSnOPF2 to TSnOPF0 are set to 000 or 111, TSnPPF is not set.

15.7.11 Timer Output Pattern Flag (TSnOPF2 to TSnOPF0)

Name Timer output pattern flag (TSnSTR1.TSnOPF2 to TSnOPF0)

Description TSnOPF2 to TSnOPF0 flags indicate the timer output patterns.

For details, see section 15.11.4, 120-DC Mode, and section 15.11.5, Software Output Control Function.

Operating mode TSnOPF2 to TSnOPF0 can be used in all operating modes.

15.7.12 Pattern Switch Detection Signal (TSnPTE)

Name Pattern switch detection signal (TSnPTE signal)

Description The TSnPTE signal toggles when the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) changes.

The toggle pattern is determined by the TSnPSC bit (TSnOPT0.TSnPSS = 1).

TSnPSC = 0

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	Toggle	-	-	-	-
	100	-	-	-	-	Toggle	-	-	-
	110	-	-	-	-	-	Toggle	-	-
	010	-	-	-	-	-	-	Toggle	-
	011	-	-	-	-	-	-	-	Toggle
	001	-	-	Toggle	-	-	-	-	-

TSnPSC = 1

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	-	-	-	-	Toggle
	100	-	-	Toggle	-	-	-	-	-
	110	-	-	-	Toggle	-	-	-	-
	010	-	-	-	-	Toggle	-	-	-
	011	-	-	-	-	-	Toggle	-	-
	001	-	-	-	-	-	-	Toggle	-

Example of operation

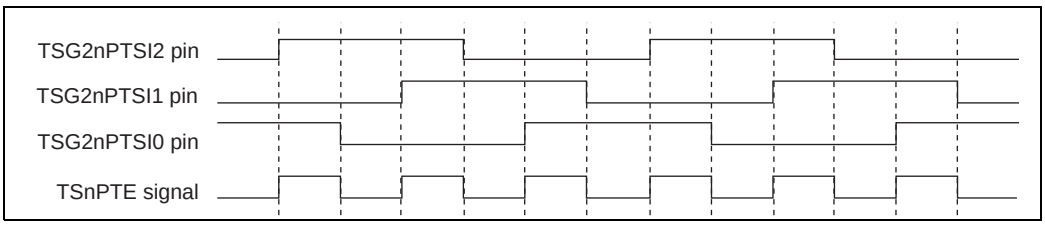


Figure 15-26 Example of Pattern Switch Detection Signal Operation

Operating mode The TSnPTE signal can be used in all operating modes.

Caution The TSnPTE signal is valid only when TSnIOC1.TSnPTS = 1 and TSnSTR0.TSnTE = 1.

15.8 Interrupt Skipping Function

Operation related to the interrupt skipping function is described below.

- Peak interrupts (INTTSG2nIPEK) and valley interrupts (INTTSG2nIVLY) can be skipped.
- TSnCTL4.TSnPIE enables outputting of the INTTSG2nIPEK interrupt and specifies whether to skip the interrupts.
- TSnCTL4.TSnVIE enables outputting of the INTTSG2nIVLY interrupt and specifies whether to skip the interrupts.

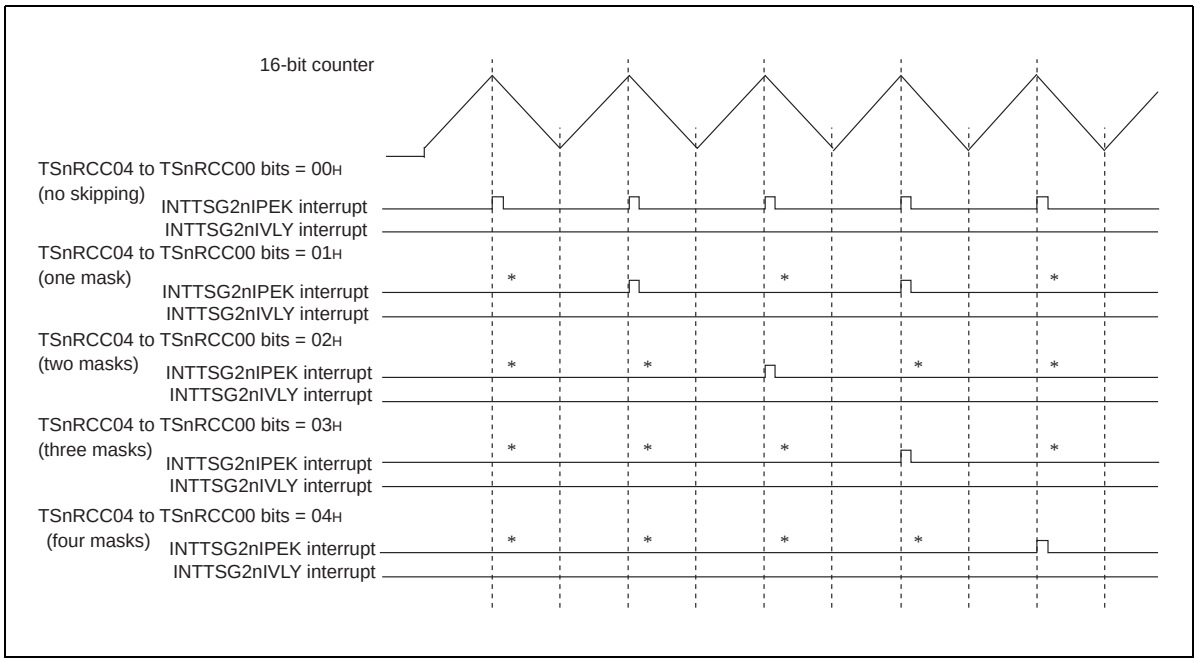
When TSnCTL3.TSnRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

When TSnCTL3.TSnRIA is set to 0 (without reload skipping), reload is executed at the specified reload timing regardless of interrupt skipping.

Caution When a value is written to TSnCTL4, and TSnRCC04 to TSnRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may be long temporarily. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSnCTL3.TSnRIA = 1).

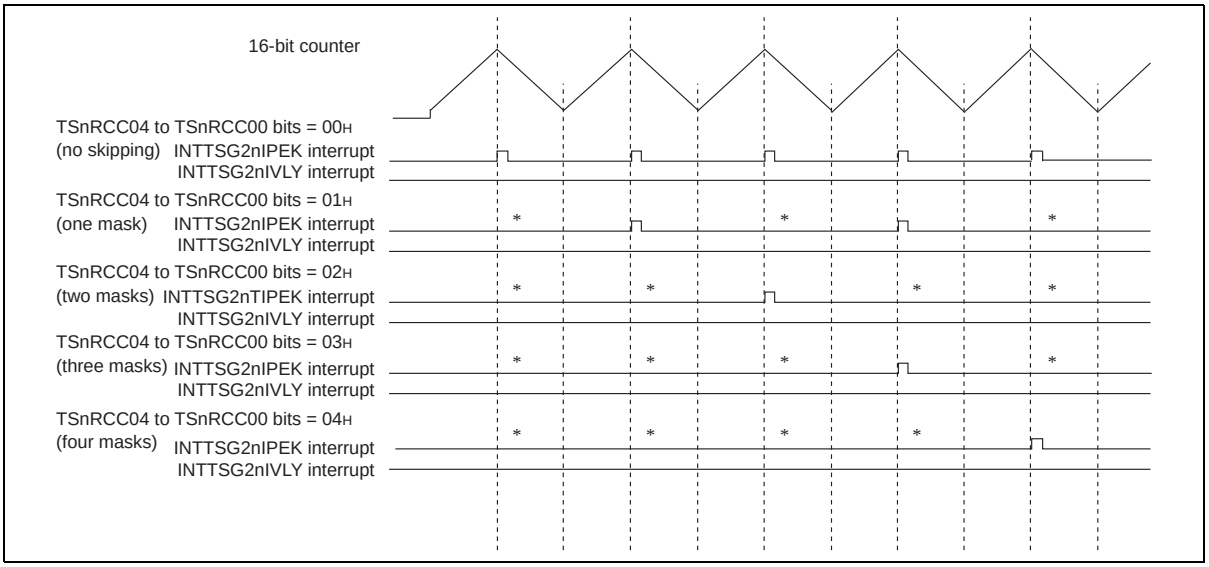
15.8.1 Operation of Interrupt Skipping Function

(1) Interrupt Skipping Operation when TSnPIE = 1 and TSnVIE = 1 in TSnCTL4 (Peak and Valley Interrupt Generation in HT-PWM Mode)



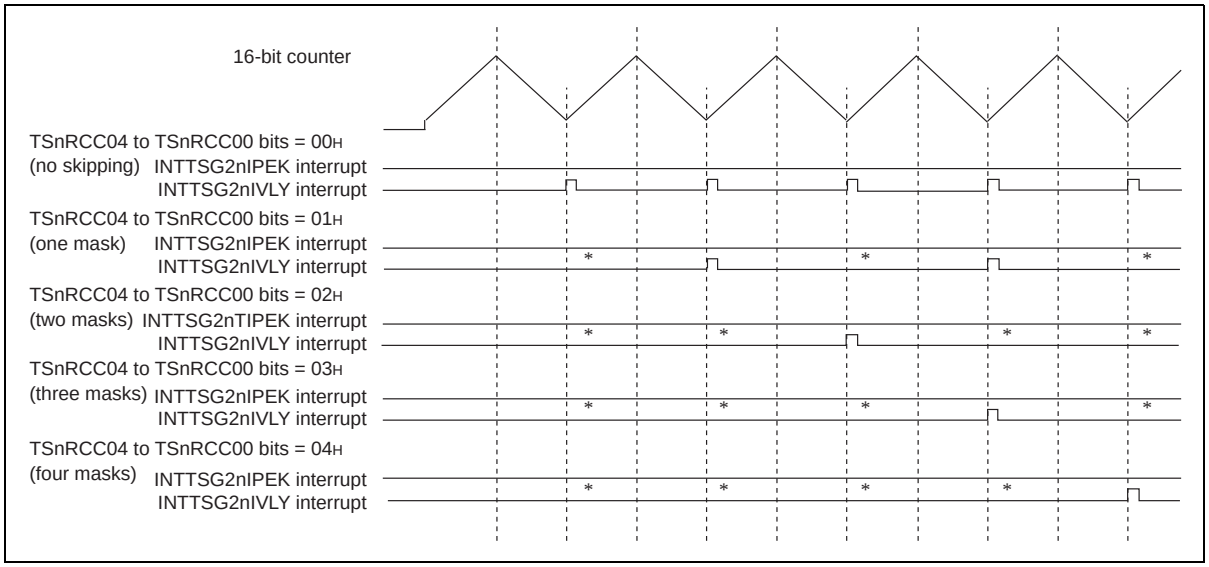
Note * Skipped interrupt request

(2) Interrupt Skipping Operation when TSnPIE = 1 and TSnVIE = 0 in TSnCTL4 Register (only Peak Interrupt Generation in HT-PWM Mode)



Note * Skipped interrupt request

(3) Interrupt Skipping Operation when TSnPIE = 0 and TSnVIE = 1 in TSnCTL4 Register (only Valley Interrupt Generation in HT-PWM Mode)



Note * Skipped interrupt request

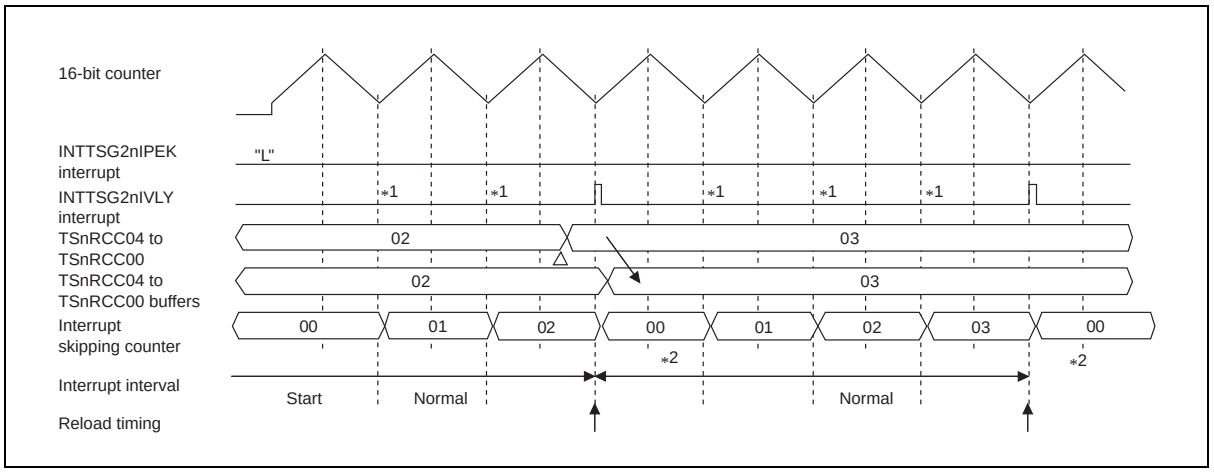


Figure 15-27 When TSnRMC = 0, TSnRIA = 1 in TSnCTL3 (with Reload Skipping)

Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared.

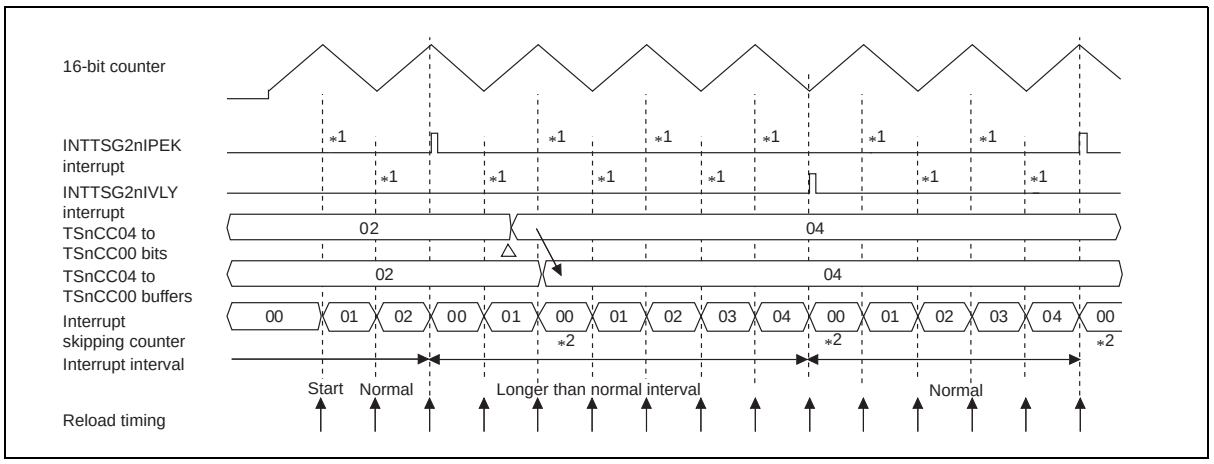


Figure 15-28 When TSnRMC = 0, TSnRIA = 0 in TSnCTL3 (without Reload Skipping)

Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared.

Caution Interrupt interval might be longer.

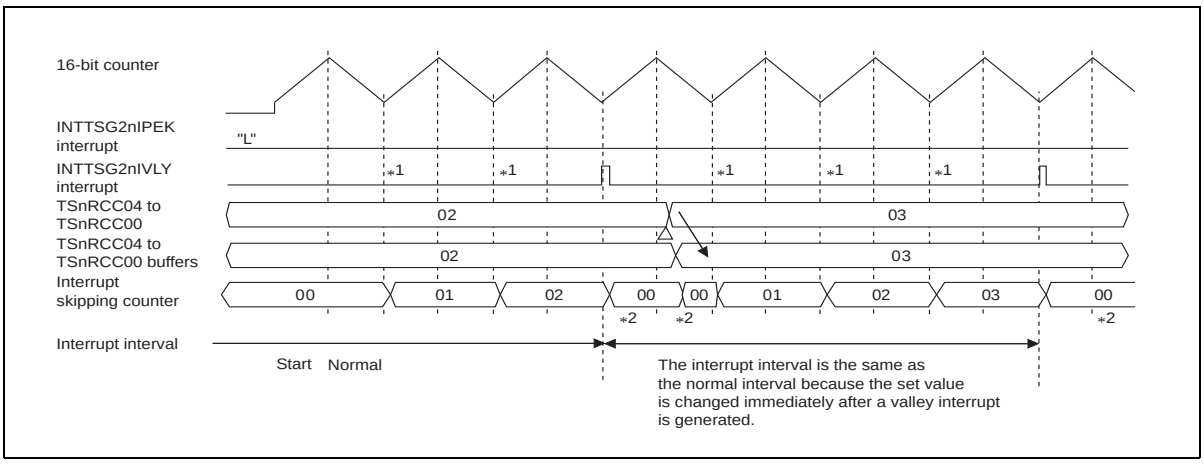


Figure 15-29 When TSnRMC = 1 in TSnCTL3 (Anytime Rewrite Mode)

Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared

Caution Interrupt interval might be longer.

Note After rewriting, the value is reflected immediately regardless of the reload timing. The interrupt skipping counter is cleared when the value is transferred to the TSnRCC04 to TSnRCC00 buffers, not when the pertinent register is rewritten.

15.8.2 Example of Operation when Peak Interrupt is Generated (in PWM Mode)

Operation related to the interrupt skipping function in PWM mode is described below.

- Valley interrupts (INTTSG2nIPEK) can be skipped. In PWM mode, it is generated by compare match of TSnCMP0 buffer register and 16-bit counter.
- TSnCTL4.TSnPIE enables outputting of the INTTSG2nIPEK interrupt and specifies whether to skip the interrupts.
- The setting of TSnCTL4.TSnVIE is disabled. At this time, the INTTSG2nIVLY interrupt is not generated.
  When TSnCTL3.TSnRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

Caution When a value is written to TSnCTL4, and TSnRCC04 to TSnRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may be long temporarily. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSnCTL3.TSnRIA = 1).

(1) Example of Operation

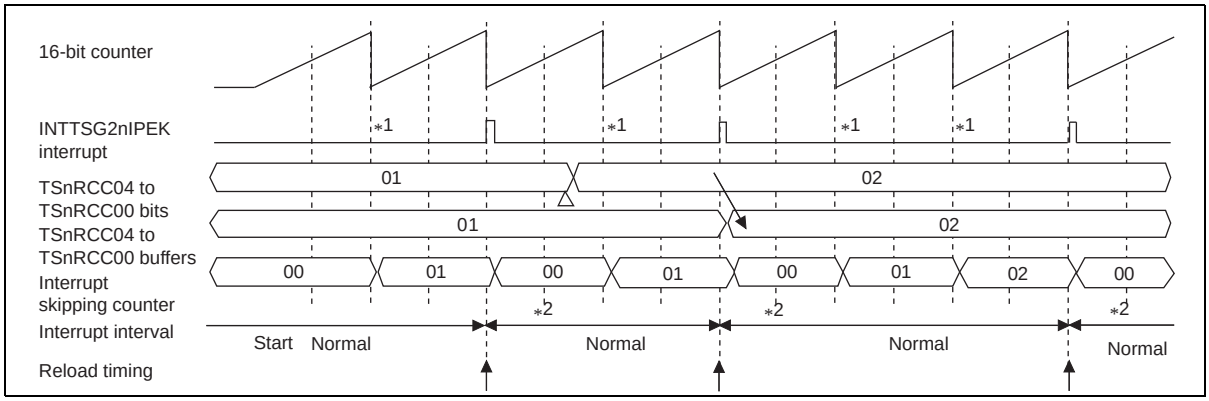


Figure 15-30 When TSnCTL3.TSnRMC = 0, TSnRIA = 1, TSnCTL4.TSnPRE = 1 (Recommended Setting)

Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared.

Note When TSnCTL3.TSnRIA = 1, reload is executed at the same timing as the interrupt after being skipped.

15.9 A/D Conversion Trigger Function

A/D conversion trigger operation is described below.

TSnDCMP0W and TSnDCMP2 are used as compare registers of the A/D conversion trigger function.

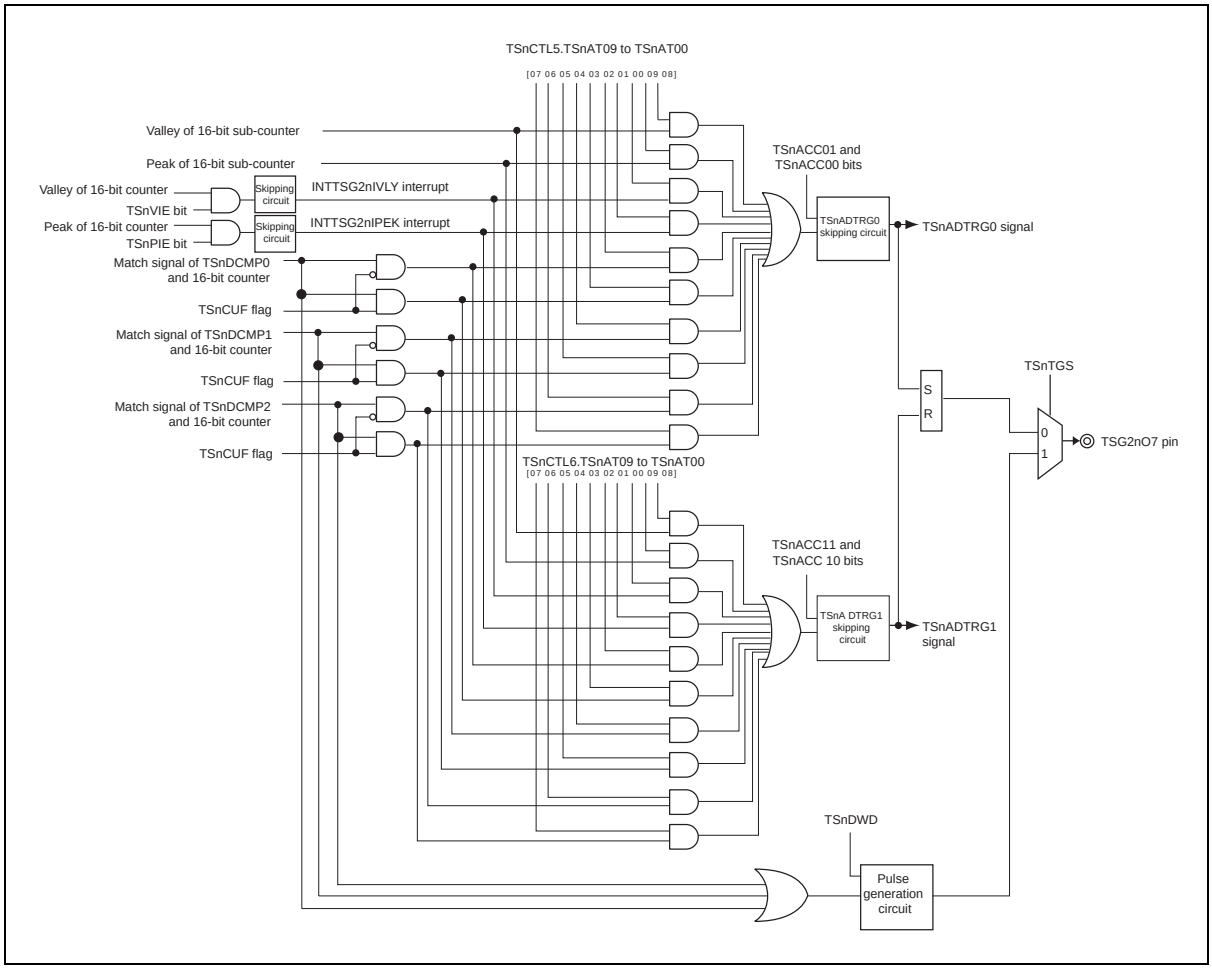


Figure 15-31 A/D Conversion Trigger and Diagnostic Output Control Circuit

As shown in Figure 15-31, a logical ORed signal can be generated by selecting the compare match of TSnDCMP0 to TSnDCMP2 with the 16-bit counter, a peak interrupt (INTTSG2nIPEK) and a valley interrupt (INTTSG2nIVLY) of the 16-bit counter, the peak timing of 16-bit sub-counter, and the valley timing of 16-bit sub-counter.

TSG2n has two channels of the identical A/D conversion trigger control circuits, which can be controlled independently. TSG2n also provides the A/D conversion trigger skipping function with the skipping rate of 1/1, 1/2, 1/4, or 1/8.

15.9.1 Operation of A/D Conversion Trigger

TSG2n has a function to generate A/D conversion start triggers (TSnADTRG0 and TSnADTRG1 signals) by selecting any of ten trigger sources as required. The trigger sources are selected by TSnAT09 to TSnAT00 in TSnCTL5 and TSnAT19 to TSnAT10 in TSnCTL6.

(1) TSnADTRG0/TSnADTRG1 Signal Output Control (TSnCTL5 and TSnCTL6)

[Trigger source]

- TSnAT00/TSnAT10 = 1 : A valley interrupt (INTTSG2nIVLY) causes an A/D conversion trigger pulse to be generated.
- TSnAT01/TSnAT11 = 1 : A peak interrupt (INTTSG2nIPEK) causes an A/D conversion trigger pulse to be generated.
- TSnAT02/TSnAT12 = 1 : While the 16-bit counter is counting up, a TSnDCMP0 compare match enables A/D conversion trigger to be generated.
- TSnAT03/TSnAT13 = 1 : While the 16-bit counter is counting down, a TSnDCMP0 compare match enables A/D conversion trigger to be generated.
- TSnAT04/TSnAT14 = 1 : While the 16-bit counter is counting up, a TSnDCMP1 compare match enables A/D conversion trigger to be generated.
- TSnAT05/TSnAT15 = 1 : While the 16-bit counter is counting down, a TSnDCMP1 compare match enables A/D conversion trigger to be generated.
- TSnAT06/TSnAT16 = 1 : While the 16-bit counter is counting up, a TSnDCMP2 compare match enables A/D conversion trigger to be generated.
- TSnAT07/TSnAT17 = 1 : While the 16-bit counter is counting down, a TSnDCMP2 compare match enables A/D conversion trigger to be generated.
- TSnAT08/TSnAT18 = 1 : A valley timing of the 16-bit sub-counter (at a switch from decrementing to incrementing) enables A/D conversion trigger to be generated.
- TSnAT09/TSnAT19 = 1 : A peak timing of 16-bit sub-counter (at a switch from incrementing to decrementing) enables A/D conversion trigger to be generated.

[Skipping setting]

- TSnACC01 and TSnACC00, and TSnACC11 and TSnACC10
  : Set the skipping rate of the TSnADTRG0 and TSnADTRG1 signals, respectively.

All A/D conversion triggers selected by TSnAT09 to TSnAT00 and TSnAT19 to TSnAT10 are logically ORed, and the resultant signals are subjected to skipping control specified by TSnACC01 and TSnACC00, and TSnACC11 and TSnACC10, and then the TSnADTRG0 and TSnADTRG1 signals are generated.

A peak interrupt (INTTSG2nIPEK) and a valley interrupt (INTTSG2nIVLY) selected by TSnAT00 and TSnAT01, and TSnAT10 and TSnAT11 are interrupt signals obtained after skipped. Therefore, they are output at the timing according to interrupt skipping control. If the interrupt output is not enabled by TSnCTL4.TSnPIE and TSnVIE, A/D conversion trigger is not output.

TSnACC01, TSnACC00, and TSnAT09 to TSnAT00, and TSnACC11, TSnACC10, and TSnAT19 to TSnAT10 can be rewritten during timer operation.

If A/D conversion trigger setting bits are rewritten during operation, the rewritten values are reflected on the A/D conversion trigger output status immediately. Such control bits are rewritten at anytime regardless of operating modes. If a write access is made to TSnCTL5 and TSnCTL6 (including a rewrite of the same value), the A/D conversion trigger skipping counter is cleared and starts counting from zero.

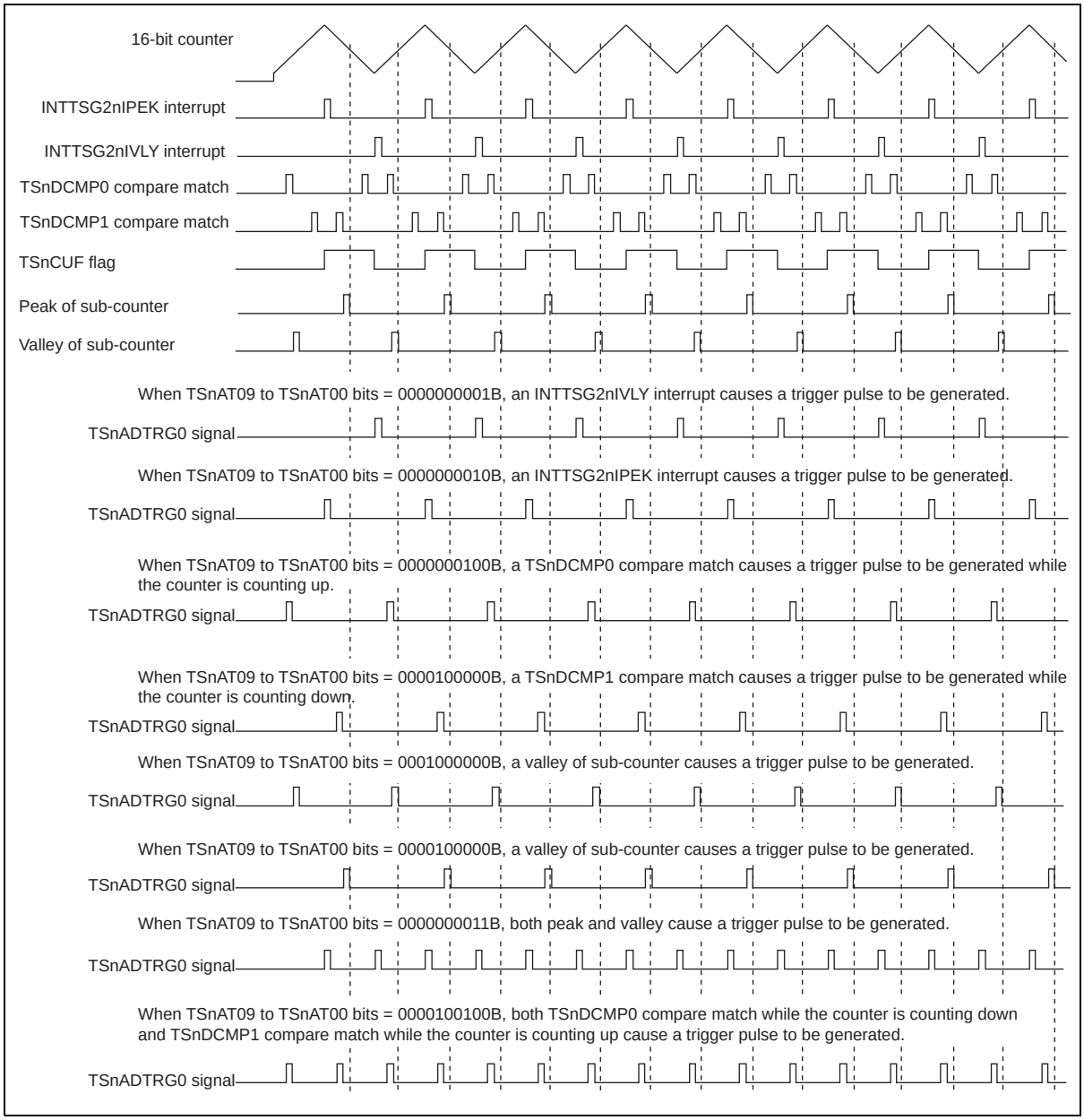


Figure 15-32 When TSnPIE = 1, TSnVIE = 1, and TSnRCC04 to TSnRCC00 = 00B in TSnCTL4, and TSnACC01 and TSnACC00 = 00B in TSnCTL5 (HT-PWM Mode)

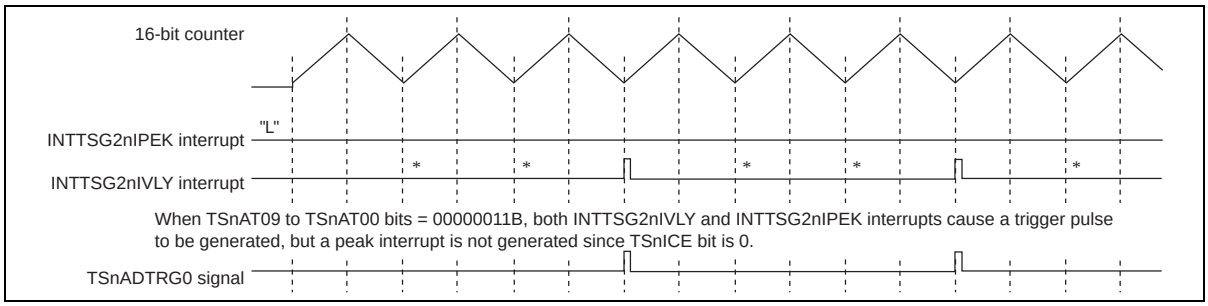


Figure 15-33 When TSnPIE = 0, TSnVIE = 1, and TSnRCC04 to TSnRCC00 = 02B in TSnCTL4 and TSnACC01 and TSnACC00 = 00B in TSnCTL5 (HT-PWM Mode)

Note * Skipped interrupt request

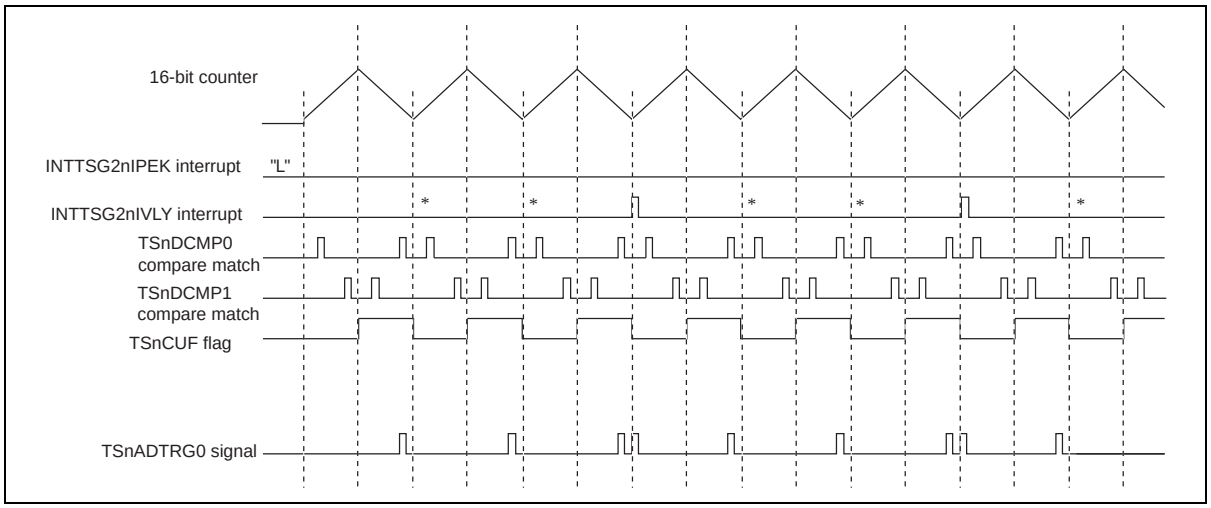


Figure 15-34 When TSnPIE = 0, TSnVIE = 1, and TSnRCC04 to TSnRCC00 = 02B in TSnCTL4 and TSnACC01 and TSnACC00 = 00B, and TSnAT09 to TSnAT00 = 00001001B in TSnCTL5 (HT-PWM Mode)

Note * Skipped interrupt request

(2) A/D Conversion Trigger Skipping Function

Example of operation of the A/D conversion trigger skipping function is shown in Figure 15-35.

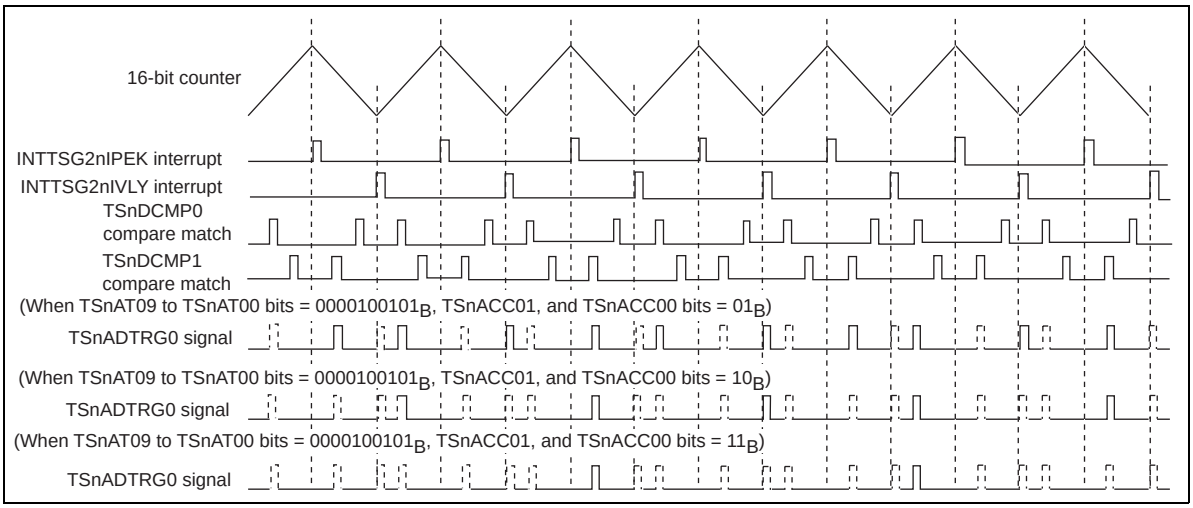


Figure 15-35 Example of Operation of A/D Conversion Trigger Skipping Function

Note Broken-lined pulses indicate A/D conversion trigger pulses skipped by the A/D conversion trigger skipping function.

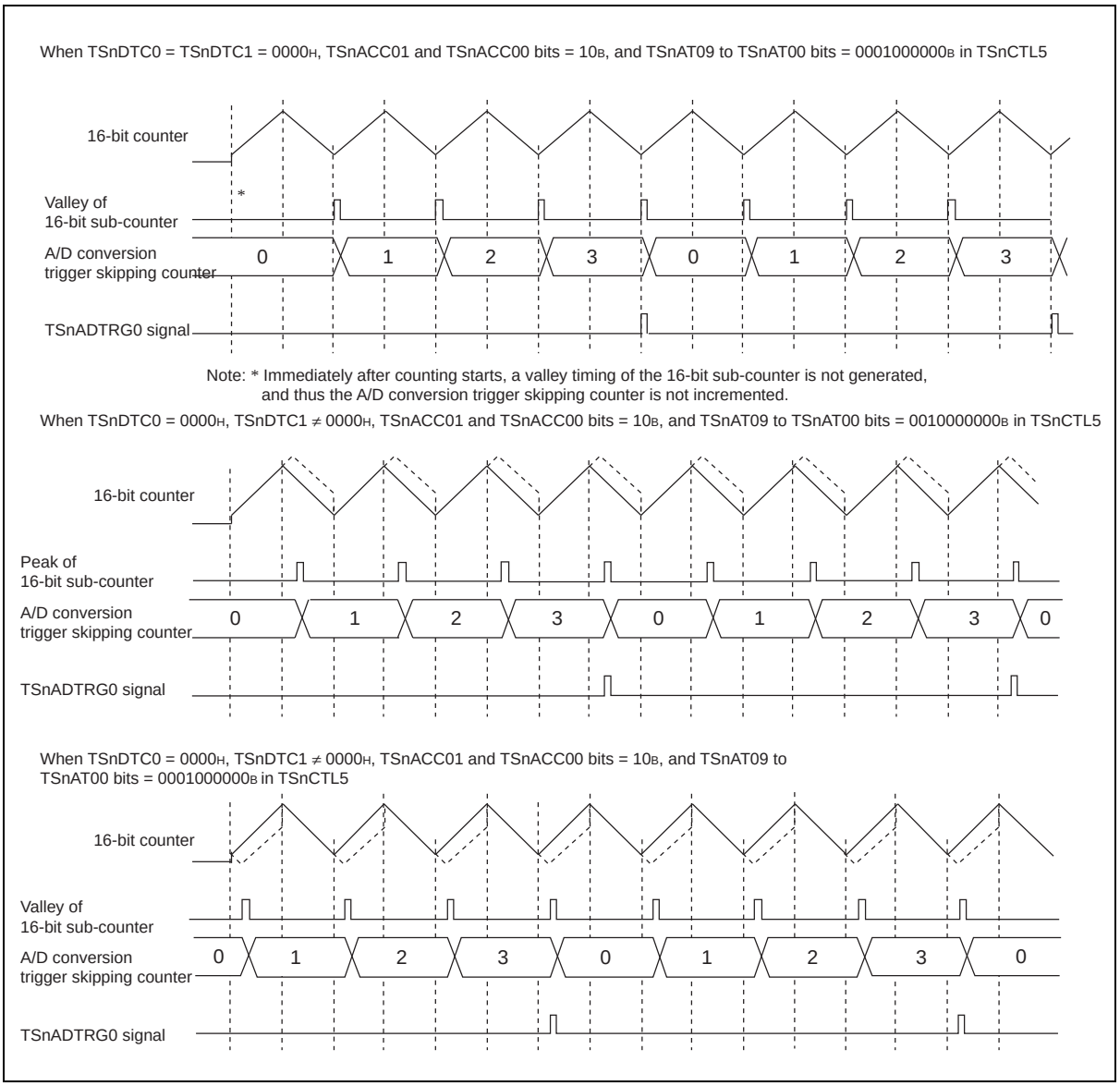


Figure 15-36 Example of Operation of A/D Conversion Trigger Skipping Function

(3) Notes on A/D Conversion Trigger

- If the same value is written to TSnDCMP0 and TSnDCMP1 or TSnDCMP2, and the same condition (when the 16-bit counter increments or decrements) is set as the valid A/D conversion trigger, A/D conversion trigger skipping counter is incremented by one and one trigger pulse is output upon a match of the 16-bit counter with these registers.
- In PWM mode, SP-PWM mode, and 120-DC mode, a valley interrupt (INTTSG2nIVLY) is not generated. Only a peak interrupt (INTTSG2nIPEK) is valid.
- In 120-DC mode, when the 16-bit counter is cleared during the carrier period due to switch of the output pattern, the A/D conversion trigger is not generated if TSnDCMP2 to TSnDCMP0 values do not match with the 16-bit counter value and a peak interrupt (INTTSG2nIPEK) is not generated.

15.10 Error/Warning Interrupt

15.10.1 Error Interrupt Function

If the simultaneous active state of the positive phase and inverse phase is detected after the error interrupt function is enabled (TSnIOC1.TSnEOC = 1), TSnSTR2.TSnTBF is set, and an error interrupt (INTTSG2nIER) of TSG2n is generated. Whether or not to detect an error of each phase (TSG2nO1 and TSG2nO2, TSG2nO3 and TSG2nO4, and TSG2nO5 and TSG2nO6 pins) can be selected by TSnCTL1.TSnTBA2 to TSnTBA0, respectively.

When an error occurs, outputs of the TSG2nO1 to TSG2nO6 pins can be set to high-impedance. For details, see section 19.4.1, Hi-Z Control Functions.

The following table shows whether or not the simultaneous active state of the positive phase and inverse phase can be detected in each mode.

Mode	Positive Phase and Inverse Phase Simultaneous Active State Detection
PWM mode	Possible
HT-PWM mode	Possible
SP-PWM mode	Possible
120-DC mode	Possible

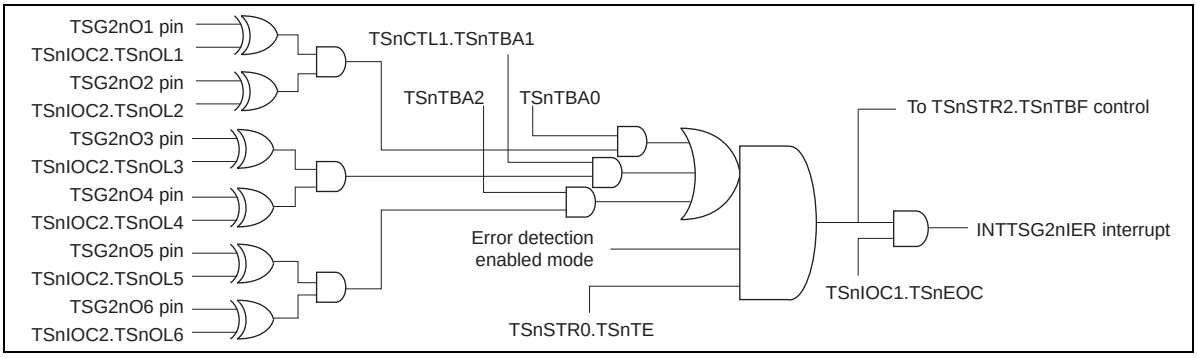


Figure 15-37 Error Interrupt (INTTSG2nIER) Generation Control Circuit

Caution When an error interrupt is generated, the error status should be canceled within an error interrupt handling. Otherwise, subsequent error interrupts are not generated.

(1) PWM Mode and 120-DC Mode

In PWM mode, if TSnCMP1 and TSnCMP2, and TSnCMP3 and TSnCMP4 are set so that the TSG2nO1 and TSG2nO2 pins output the active level simultaneously, an error interrupt (INTTSG2nIER) is generated. With the same setting, if TSnCMP5, TSnCMP6, TSnCMP7, TSnCMP8, TSnCMP9, TSnCMP10, TSnCMP11, and TSnCMP12 are set so that the TSG2nO3 and TSG2nO4, and TSG2nO5 and TSG2nO6 pins output the active level simultaneously, an error interrupt (INTTSG2nIER) is generated.

In 120-DC mode, if TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and TSnCMP10, TSnCMP3, TSnCMP4, TSnCMP7, TSnCMP8, TSnCMP11, and TSnCMP12, and TSnPAT0W and TSnPAT1W are set so that the TSG2nO1 and TSG2nO2 pins output the active level simultaneously, an error interrupt (INTTSG2nIER) is generated. With the same setting, the TSG2nO3 and TSG2nO4, and TSG2nO5 and TSG2nO6 pins also output the active level simultaneously, an error interrupt (INTTSG2nIER) is generated.

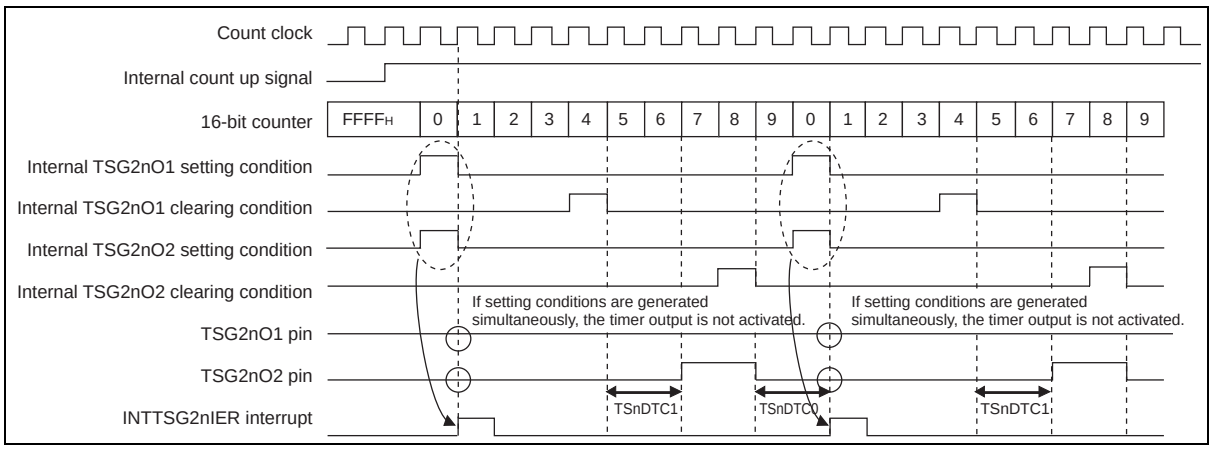


Figure 15-38 Example of Error Interrupt (INTTSG2nIER) Generation (PWM Mode)

Note TSG2nO3 and TSG2nO4, and TSG2nO5 and TSG2nO6 behave the same.

When the active level of output is switched by operating TSnIOC2.TSnOL1 and TSnOL2, an error interrupt is generated as shown in Figure 15-39.

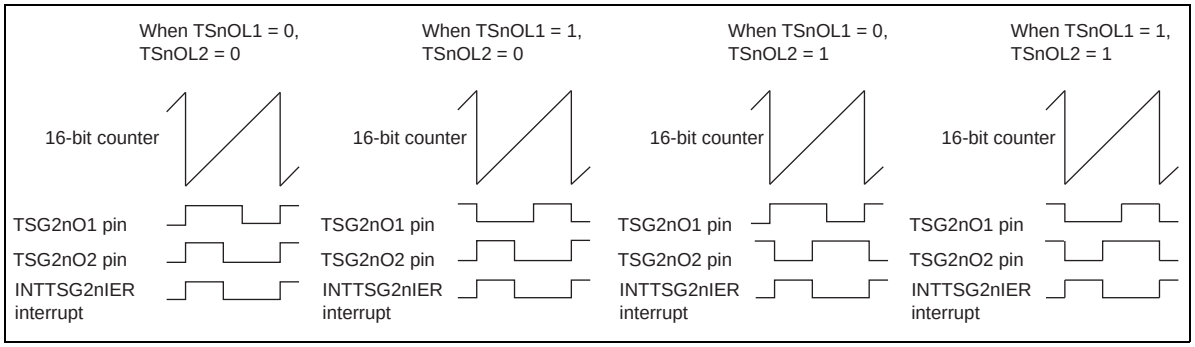


Figure 15-39 Example of Error Interrupt (INTTSG2nIER) Generation for each Active Level

(2) HT-PWM Mode and SP-PWM Mode

Either TSG2n dead time setting register 0 or 1 (TSnDTC0 or TSnDTC1) is 0000_H, an error may occur.

Note If an error occurs when the dead time control function is used (both TSnDTC0 and TSnDTC1 are not 0000_H), internal circuit failure may occur.

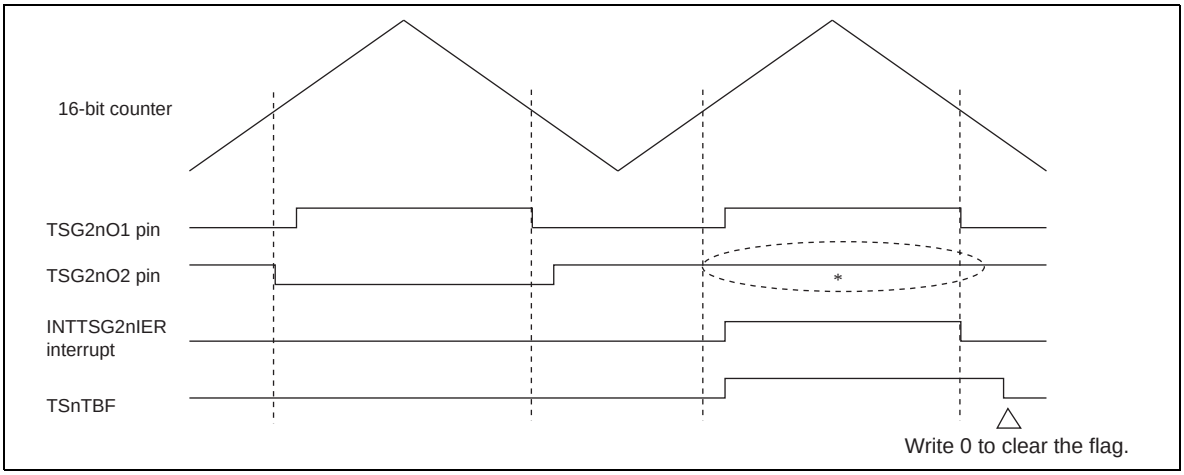


Figure 15-40 Example of Error Interrupt Operation

Note * TSG2nO2 pin control circuit failure occurs.

15.10.2 Warning Interrupt Function

TSG2n has a warning interrupt (INTTSG2nIWN).

Warning interrupt (INTTSG2nIWN) is generated when any of the following conditions is detected.

For details, see section 15.7, Flags.

- When simultaneous change in two or more pins of TSG2nPTSI2 to TSG2nPTSI0 is detected
  See section 15.7.4, Noise Detection Flag (TSnNDF).
- When reversal is detected of the TSG2nPTSI2 to TSG2nPTSI0 pin input pattern
  See section 15.7.7, Pattern Reversal Detection Flag (TSnPRF).
- When 000 or 111 is detected from the TSG2nPTSI2 to TSG2nPTSI0 pins
  See section 15.7.6, Pattern Error Detection Flag (TSnPEF).
- When a toggle of the TSnPTSI2 to TSG2nPTSI0 pins is generated three times or more between TSnSTCI0 and TSnSTCI1 signal triggers.
  See section 15.7.8, TSG2nPTSI2 to TSG2nPTSI0 Pin Abnormal Toggle Detection Flag (TSnPTF).
- When the TSnSTCI0 and TSnSTCI1 signal triggers are detected simultaneously
  See section 15.7.9, TSnSTCI0 and TSnSTCI1 Signal Simultaneous Trigger Detection Flag (TSnTDF).
- When the phase difference between the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) and output patterns (TSnOPF2 to TSnOPF0) is detected
  See section 15.7.10, Pattern Phase Difference Detection Flag (TSnPPF).

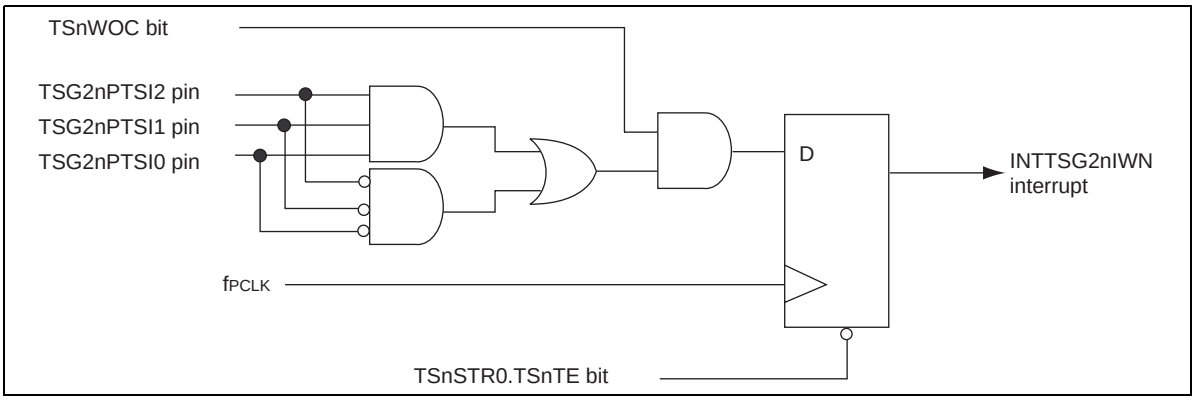


Figure 15-41 Detection of Abnormality of TSG2nPTSI2 to TSG2nPTSI0 Pins

15.11 Operating Modes

Table 15-48 List of Modes

TSnCTL0 Register		Timer Modes
TSnMD1	**TSnMD0**	
0	0	PWM mode
0	1	HT-PWM mode (HT-PWM)
1	0	Shifted-pulse - pulse width modulation mode (SP-PWM)
1	1	120-DC mode

15.11.1 PWM Mode

Overview

A PWM signal is output at the TSG2nO1 to TSG2nO6 pins according to set timing/clear timing of TSnCMP1 to TSnCMP12 registers with the PWM period set in the TSnCMP0 register.

Prerequisites

- Set the set timing to the compare register with an even number: TSnCMP2 (set timing of the TSG2nO1 output), TSnCMP4 (set timing of the TSG2nO2 output), TSnCMP6 (set timing of the TSG2nO3 output), TSnCMP8 (set timing of the TSG2nO4 output), TSnCMP10 (set timing of the TSG2nO5 output) and TSnCMP12 (set timing of the TSG2nO6 output)
- Set the clear timing to the compare register with an odd number: TSnCMP1 (clear timing of the TSG2nO1 output), TSnCMP3 (clear timing of the TSG2nO2 output), TSnCMP5 (clear timing of the TSG2nO3 output), TSnCMP7 (clear timing of the TSG2nO4 output), TSnCMP9 (clear timing of the TSG2nO5 output) and TSnCMP11 (clear timing of the TSG2nO6 output)

Functional description

Set the PWM period and set/clear timing of the TSG2nO1 to TSG2nO6 outputs. Set TSnTRG0.TSnTS = 1 to start the timer counter.

The TSG2nO1 to TSG2nO6 outputs are set to the inactive state at the same time the counting begins. The outputs are set to the active state by the match of the buffer registers TSnCMP2, TSnCMP4, TSnCMP6, TSnCMP8, TSnCMP10, and TSnCMP12 with the 16-bit counter.

Next, the TSG2nO1 to TSG2nO6 outputs are set to the inactive state by the match of the buffer registers TSnCMP1, TSnCMP3, TSnCMP5, TSnCMP7, TSnCMP9, and TSnCMP11 with the 16-bit counter.

During counting, a compare match interrupt (INTTSG2nI00 to INTTSG2nI12) is generated by the match of the buffer register TSnCMP0 to TSnCMP12 with the 16-bit counter.

Caution Reload is executed when writing to the TSnCMP1 register at TSnCTL3.TSnRMC = 0. Therefore, even when it is needed to rewrite only the value of the TSnCMP0 register, a write operation to the TSnCMP1 register is necessary. When only the TSnCMP0 register is rewritten, reload is not done.

Note The PWM mode is set when TSnCTL0.TSnMD1 and TSnMD0 = 00_B.

(a) When TSnCMP0 and TSnCMP1 to TSnCMP12 are Not Rewritten during Timer Operation

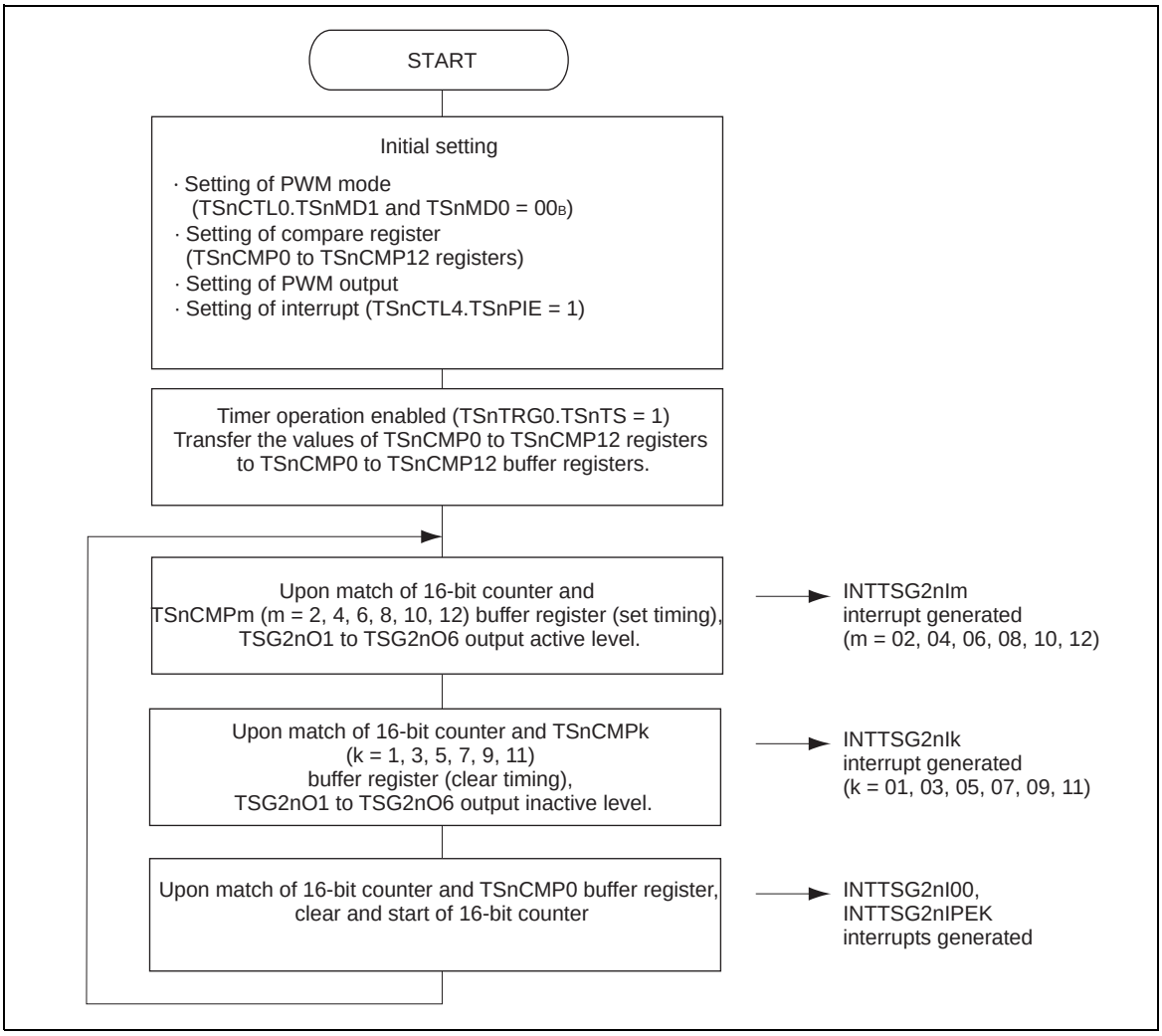


Figure 15-42 Basic Operation Flow of PWM Mode (1/2)

(b) When TSnCMP0 and TSnCMP1 to TSnCMP12 are Rewritten during Timer Operation

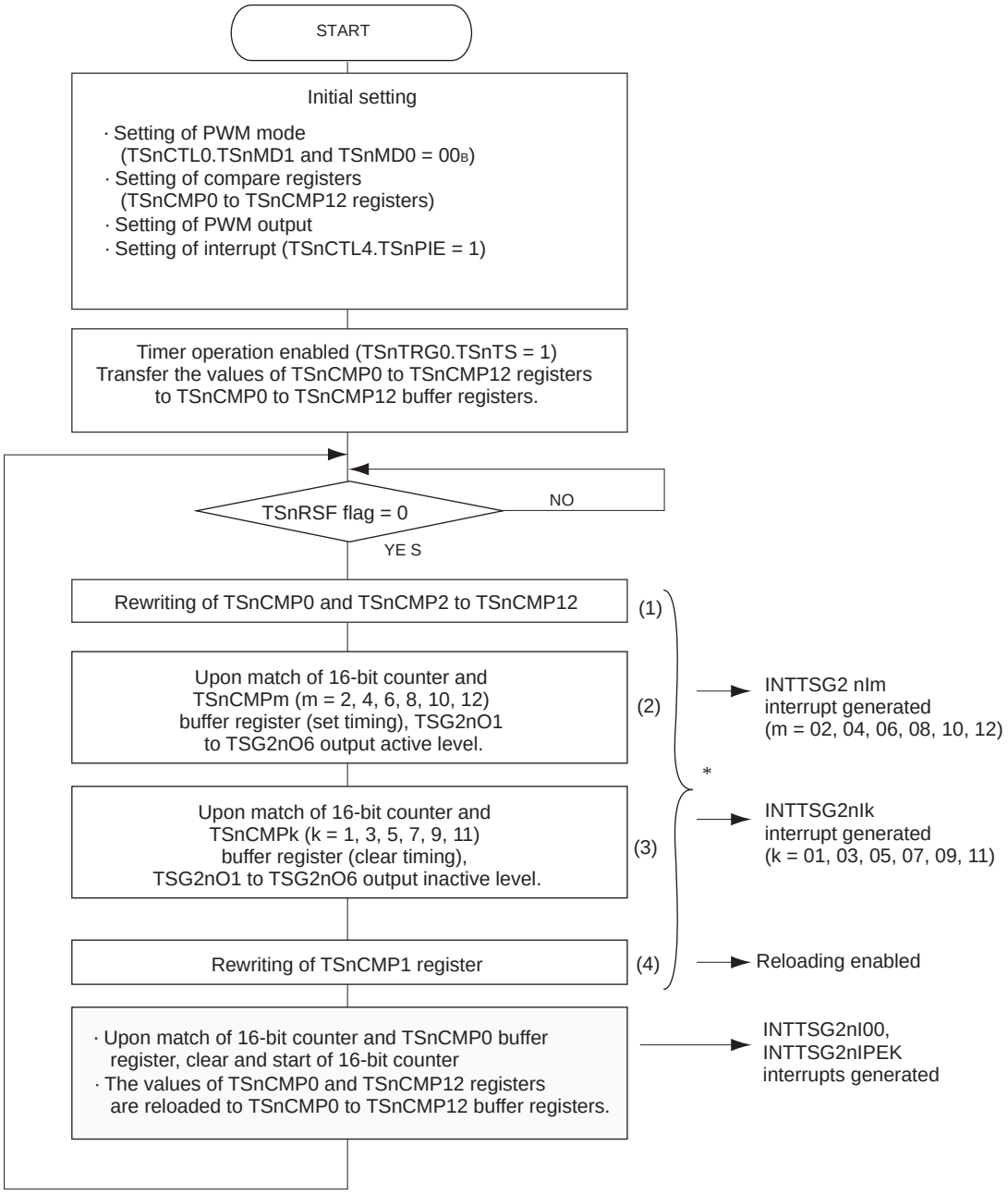


Figure 15-42 Basic Operation Flow of PWM Mode (2/2)

Note The timing of (2) and (3) might be different depending on the rewriting timing of (1) and (4) and the TSnCMP1 value. Please make sure that (1) is always done before (4).

Caution Please rewrite compare registers after confirming that the reload request flag TSnRSF is 0.

(1) List of Operations in PWM Mode

Table 15-49 Counter Functions in PWM Mode

Operation		Setting Conditions
16-bit counter	Start	TSnTRG0.TSnTS = 0 → 1, or a simultaneous start trigger
	Clear	Compare match of TSnCMP0 buffer register and 16-bit counter
	Stop	TSnTRG1.TSnTT = 0 → 1

Table 15-50 Functions of Compare Registers and Dead Time Setting Register in PWM Mode

Register	Rewriting Method	Rewrite during Operation	Function
TSnCMP0	Reload/Anytime rewrite	Possible	Setting period
TSnCMPm (m = 1 to 12)	Reload/Anytime rewrite	Possible	Setting set/clear timing
TSnDCMP0W, TSnDCMP2	Reload/Anytime rewrite	Possible	Diagnostic output or A/D conversion trigger
TSnDTC0, TSnDTC1	Reload	Possible*	Setting dead time

Note * Please refer to 3 Controlling Dead Time in PWM Mode.

Table 15-51 Timer Input in PWM Mode

Pin	Function
TSG2nCLKI	Clock enable input

Table 15-52 Timer Output in PWM Mode

Pin	Function
TSG2nOm (m = 1 to 6)	PWM output by compare match of TSnCMPk buffer register and 16-bit counter (k = 1 to 12)
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Table 15-53 Interrupt Requests in PWM Mode

Interrupt	Function
INTTSG2nIm (m = 00 to 12)	Compare match of TSnCMPm buffer register and 16-bit counter (m = 0 to 12)
INTTSG2nIER	Error (simultaneous active state of TSG2nO1 and TSG2nO2, or TSG2nO3 and TSG2nO4, or TSG2nO5 and TSG2nO6)
INTTSG2nIVLY	-
INTTSG2nIPEK	Peak interrupt (generated at the same timing as INTTSG2nI00)
INTTSG2nIWN	Warning interrupt

Note "-": Not available in PWM mode

Table 15-54 Compare Match Timing in PWM Mode

Compare Match	Timing
TSnCMP0	When 16-bit counter changes from TSnCMP0 to 0000_H
TSnCMPm (m = 1 to 12)	After detecting the match of 16-bit counter and TSnCMPm (m = 1 to 12)

Table 15-55 Example of Setting each Timer Output Condition in PWM Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG2nOm (m = 1 ot 6)	PWM output	(TSnCMP0 + 1) × count clock	Output an inactive level throughout one period (duty cycle 0%)	TSnCMPm = TSnCMP (m + 1) or TSnCMP (m + 1) > TSnCMP0 (m = 1, 3, 5, 7, 9, 11)
			Output an active level of one count clock in one period	TSnCMPm = TSnCMP (m + 1) + 1 TSnCMP (m + 1) = TSnCMPm - 1 (m = 1, 3, 5, 7, 9, 11)
			Output an inactive level of one count clock in one period	TSnCMPm = TSnCMP (m + 1) - 1 TSnCMP (m + 1) = TSnCMPm + 1 (m = 1, 3, 5, 7, 9, 11)
			Output an active level throughout one period (duty cycle 100%)	TSnCMPm > TSnCMP0 TSnCMP (m + 1) ≤ TSnCMP0 (m = 1, 3, 5, 7, 9, 11)

Note -: Not available in PWM mode

When only TSnCMP2 is rewritten and TSG2nO1 is output

(TSnIOC0.TSnTOE1 = 1, TSnIOC2.TSnOL1 = 0)

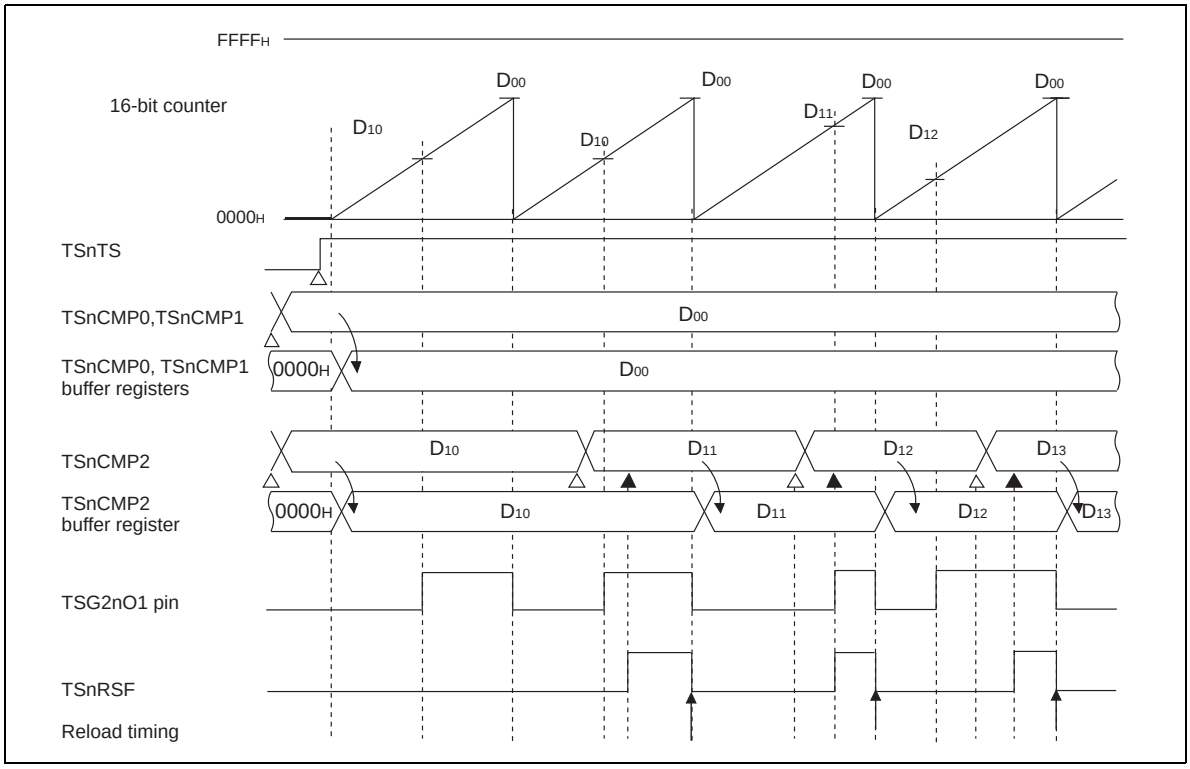


Figure 15-43 Example of Basic Operation Timing of PWM Mode (1/2)

Note 1. D00: Set values of TSnCMP0 and TSnCMP1 (0000_H - $FFFF_H$)
D10, D11, D12 and D13: Set values of TSnCMP2 (0000_H - $FFFF_H$)
TSG2nO1 (PWM duty cycle) = (TSnCMP1 - TSnCMP2) × count clock
TSG2nO1 (PWM period) = (TSnCMP0 + 1) × count clock

Note 2. TSG2nO2 to TSG2nO6 pins behave similarly to the TSG2nO1 pin.

Note 3. Δ: Write access

Note 4. ▲: TSnCMP1 write access (equivalent).

When TSnCMP0-TSnCMP2 are rewritten, and TSG2nO1 is output
(TSnIOC0.TSnTOE1 = 1, TSnIOC2.TSnOL1 = 0)

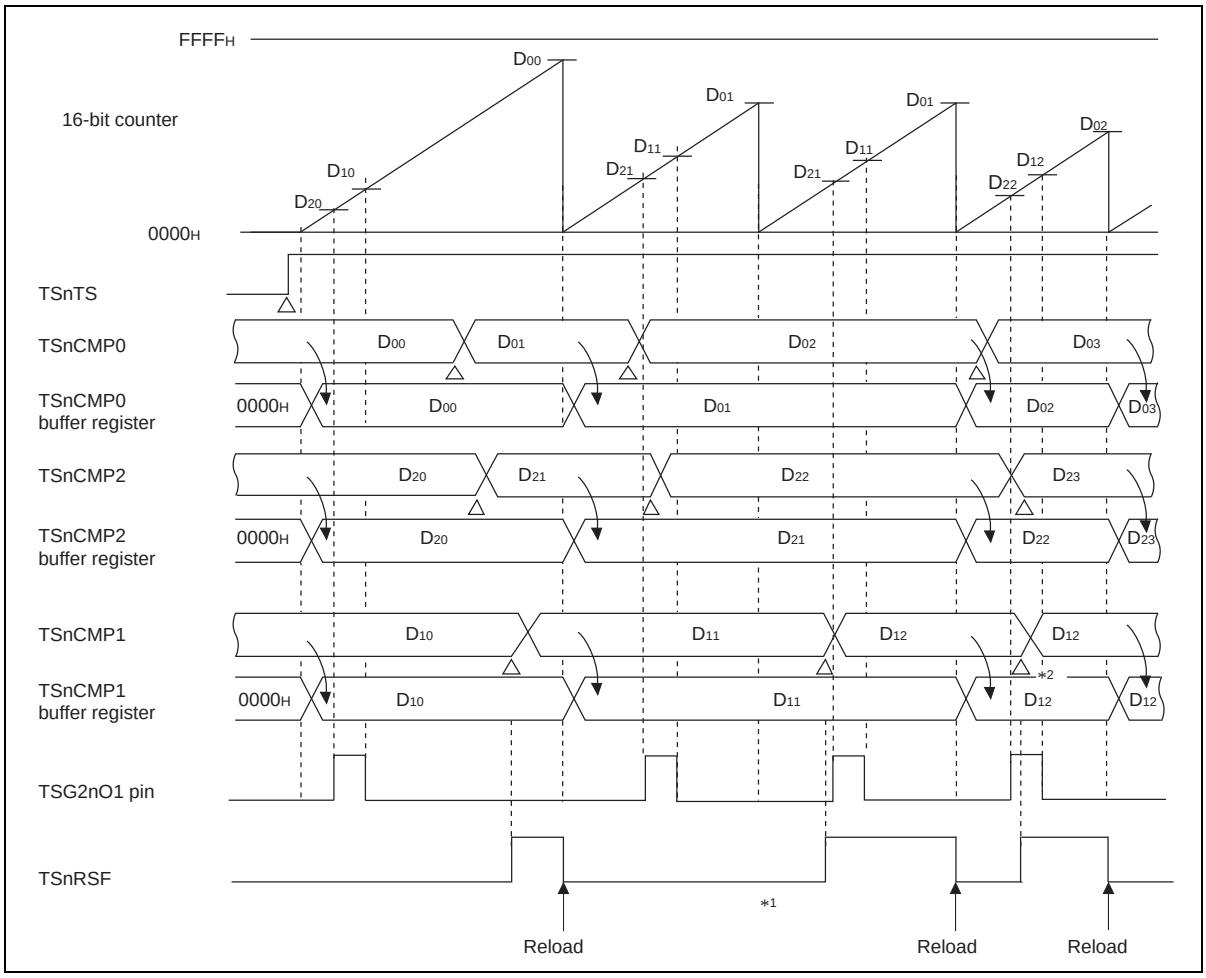


Figure 15-43 Example of Basic Operation Timing of PWM Mode (2/2)

Note 1. The TSnCMP0 buffer register is not reloaded, because TSnCMP1 register was not rewritten.
Note 2. Equivalent writing

Note 1. D00, D01, D02, and D03: Set point of TSnCMP0 (0000_H - $FFFF_H$)
D10, D11, D12, and D13: Set point of TSnCMP1 (0000_H - $FFFF_H$)
D20, D21, D22, and D23: Set point of TSnCMP2 (0000_H - $FFFF_H$)
Note 2. Outputs from TSG2nO2 to TSG2nO6 behave similarly to the TSG2nO1 pin.
Note 3. Δ: Write access

(2) Interrupt/Reload Skipping Function in PWM Mode

By setting TSnCTL4.TSnPRE and TSnPIE to 1 and setting the TSnRCC04 to TSnRCC00 and TSnCTL3.TSnRIA, the reload and interrupt skipping function can be used.

By setting TSnPRE to 1 and setting the TSnRCC04 to TSnRCC00, the interrupt skipping function can be used.

(3) Controlling Dead Time in PWM Mode

By setting the TSnDTC0 and TSnDTC1 registers in PWM mode, it is possible to control the dead time. The dead time is controlled according to the switch timing of the TSG2nO1 output, the TSG2nO2 output, the TSG2nO3 output, the TSG2nO4 output, the TSG2nO5 output, and the TSG2nO6 output.

Table 15-56 Dead Time in PWM Mode

Switch Timing	Dead Time
TSG2nO1: High level to low level TSG2nO2: Low level to high level	Value of TSnDTC1 register
TSG2nO2: High level to low level TSG2nO1: Low level to high level	Value of TSnDTC0 register
TSG2nO3: High level to low level TSG2nO4: Low level to high level	Value of TSnDTC1 register
TSG2nO4: High level to low level TSG2nO3: Low level to high level	Value of TSnDTC0 register
TSG2nO5: High level to low level TSG2nO6: Low level to high level	Value of TSnDTC1 register
TSG2nO6: High level to low level TSG2nO5: Low level to high level	Value of TSnDTC0 register

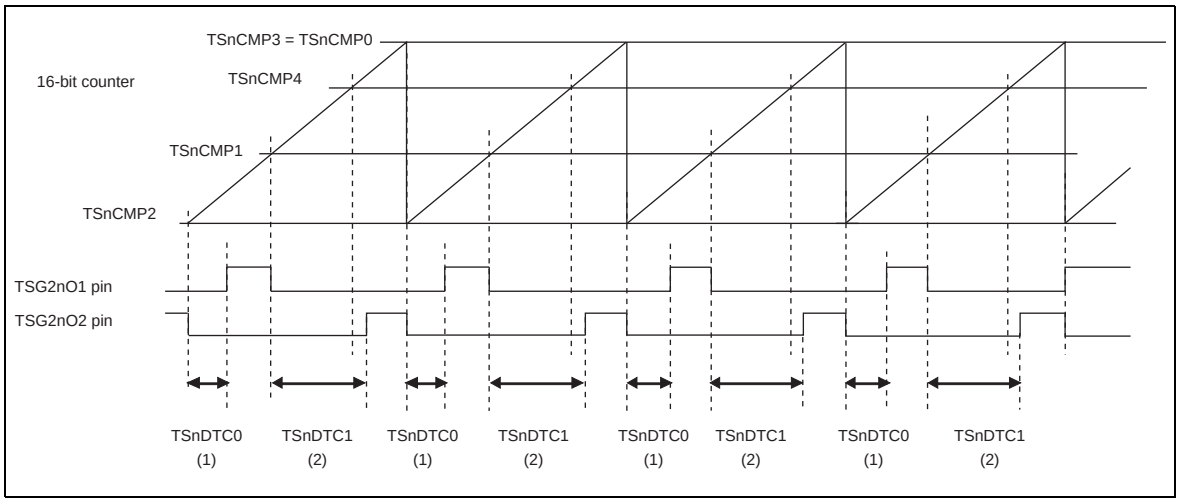


Figure 15-44 Example of Dead Time Control between TSG2nO1 and TSG2nO2 Outputs (1/2)

At (1), the dead time counter starts counting at the falling edge of the TSG2nO2 output. Here, although the 16-bit counter is 0000_H, the TSG2nO1 output stays inactive because the dead time counter is still operating. The TSG2nO1 output becomes active at the timing when the dead time count operation ends.

At (2), the dead time counter starts counting at the falling edge of the TSG2nO1 output. Even after the match of the 16-bit counter and TSnCMP4, the TSG2nO2 output stays inactive because the dead time counter is still operating. The TSG2nO2 output becomes active at the timing when the dead time count operation ends.

Note 1. The TSG2nO1 and TSG2nO2 pin outputs are active high.
Note 2. The TSG2nO3 to TSG2nO6 pin outputs behave similarly.

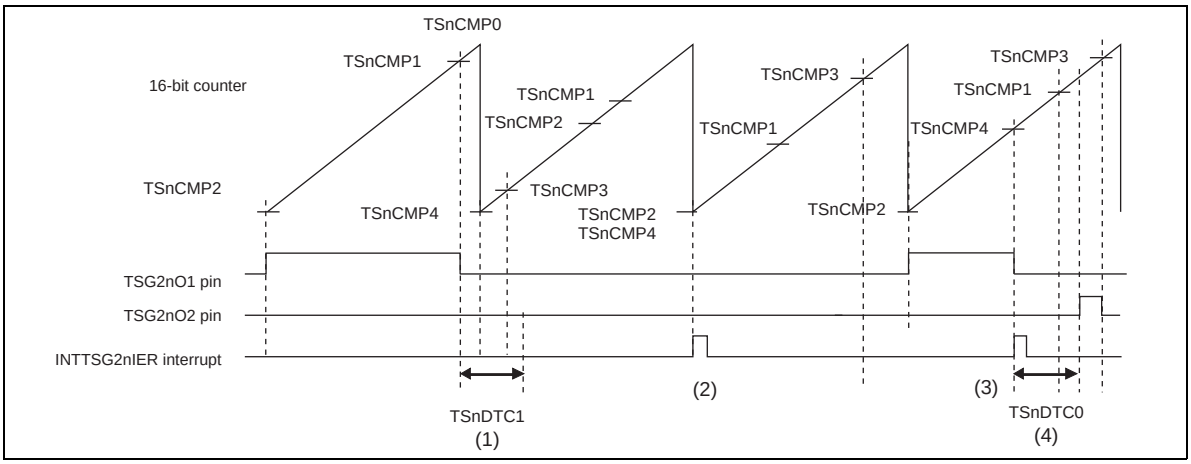


Figure 15-44 Example of Dead Time Control between TSG2nO1 and TSG2nO2 Outputs (2/2)

During (1), the dead time counter starts counting at the falling edge of the TSG2nO1 output. Even after the 16-bit counter reaches 0000_H and the match occurs between the 16-bit counter and TSnCMP4, the TSG2nO2 output stays inactive because the dead time counter is still operating. Moreover, since the TSnCMP3 register compare match occurs before the operation of the dead time counter ends, the TSG2nO2 output stays inactive.

TSnCMP1 + TSnDTC1 ≥ TSnCMP0 + TSnCMP2 (TSG2nO2 stays inactive)

TSnCMP2 + TSnDTC0 ≥ TSnCMP0 + TSnCMP1 (TSG2nO1 stays inactive)

At (2), the INTTSG2nIER interrupt occurs because the TSnCMP2 register and the TSnCMP4 register are set so that the TSG2nO1 and TSG2nO2 outputs rise simultaneously. Here, the TSG2nO1 output and the TSG2nO2 output are inactive.

At (3), compare match with the TSnCMP4 register generates an INTTSG2nIER interrupt and both TSG2nO1 and TSG2nO2 outputs become inactive.

At (4), the falling edge (inactive) of the TSG2nO1 output is caused by the simultaneous active state detected and the dead time counter starts counting. After the end of the dead time counter operation, the TSG2nO2 output becomes active.

Note 1. TSG2nO1 and TSG2nO2 are set to active high.
Note 2. The TSG2nO3 to TSG2nO6 pin outputs behave similarly.

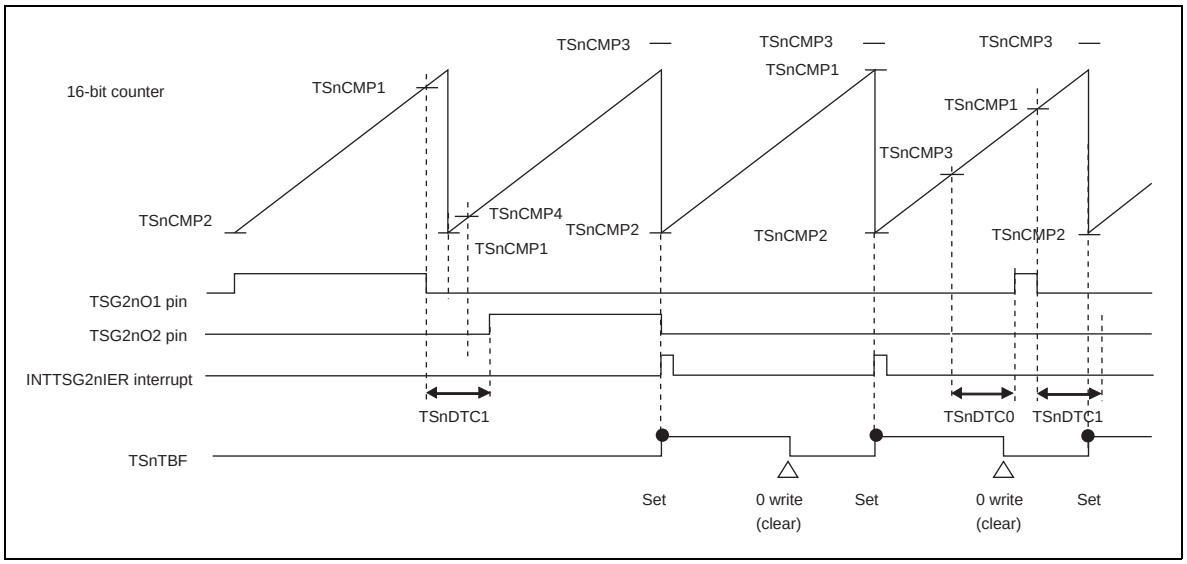


Figure 15-45 Example of 100% Duty Output at Dead Time Control

When the TSG2nO2 pin is set to duty cycle of 100% (TSnCMP3 ≥TSnCMP0 + 1), the output of the TSG2nO1 pin is fixed to a low level. This control is intended to mask the active condition of TSG2nO1 output since the TSG2nO2 output is active before the TSG2nO1 output becomes active. In this case, the INTTSG2nIER interrupt is also generated because TSG2nO1 and TSG2nO2 outputs become high simultaneously.

Note 1. TSG2nO1 and TSG2nO2 are set to active high.
Note 2. The TSG2nO3 to TSG2nO6 pin outputs behave similarly.

(4) Dead Time Rewriting during Timer Operation in PWM Mode

In PWM mode, it is possible to rewrite TSG2n dead time setting registers TSnDTC0 and TSnDTC1 while counting. The new settings are active at reload timing. It is not possible to change the dead time setting by rewriting at anytime.

Please enable reloading by writing to the TSnCMP1 register.

15.11.2 HT-PWM Mode (High accuracy Triangular - Pulse Width Modulation Mode)

Overview

In this mode, the 16-bit counter (up/down count by ±2 bits, practically 15 bits) and the 16-bit compare registers (LSB is used to control additional pulse) are used to generate a 6-phase PWM signal.

Prerequisites

- Set the carrier wave period to TSnCMP0.
- Set the duty cycle of the voltage data signals of the U phase, V phase, and W phase with TSnCMPU, TSnCMPV, and TSnCMPW (The values set to TSnCMPU, TSnCMPV, and TSnCMPW are reflected immediately to the corresponding TSnCMPm (m = 1, 2, 5, 6, 9, 10)).
- Symmetric triangular wave control is described in this section. Please refer to 15.11.2 (10) Asymmetric Triangular Wave Control in HT-PWM Mode, for asymmetric triangular wave control.

Functional description

Set the period of carrier wave and the duty cycle of the U phase, the V phase, and the W phase. Counting up begins when TSnTRG0.TSnTS is set to 1.

The 16-bit counter counts up from TSnDTC0 as the minimum value, and counts down upon the match of the maximum value of TSnCMP0 + TSnDTC0.

The dead time is set with TSnDTC0 and TSnDTC1. TSnDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch and TSnDTC1 sets the dead time of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSnDTT1 to TSnDTT3) for dead time generation load the set values of TSnDTC0 and TSnDTC1 by the compare match of the 16-bit counter and the TSnCMPm buffer register (m = 1, 2, 5, 6, 9, 10), and start down-counting.

The INTTSG2nIm interrupts (m = 01, 02, 05, 06, 09, 10) are generated by the compare match of the 16-bit counter with TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and the TSnCMP10 buffer registers.

Note The HT-PWM mode is enabled when TSnTRG0.TSnMD1 and TSnMD0 = 01_B.

(1) Block Diagram and Basic Timing Chart

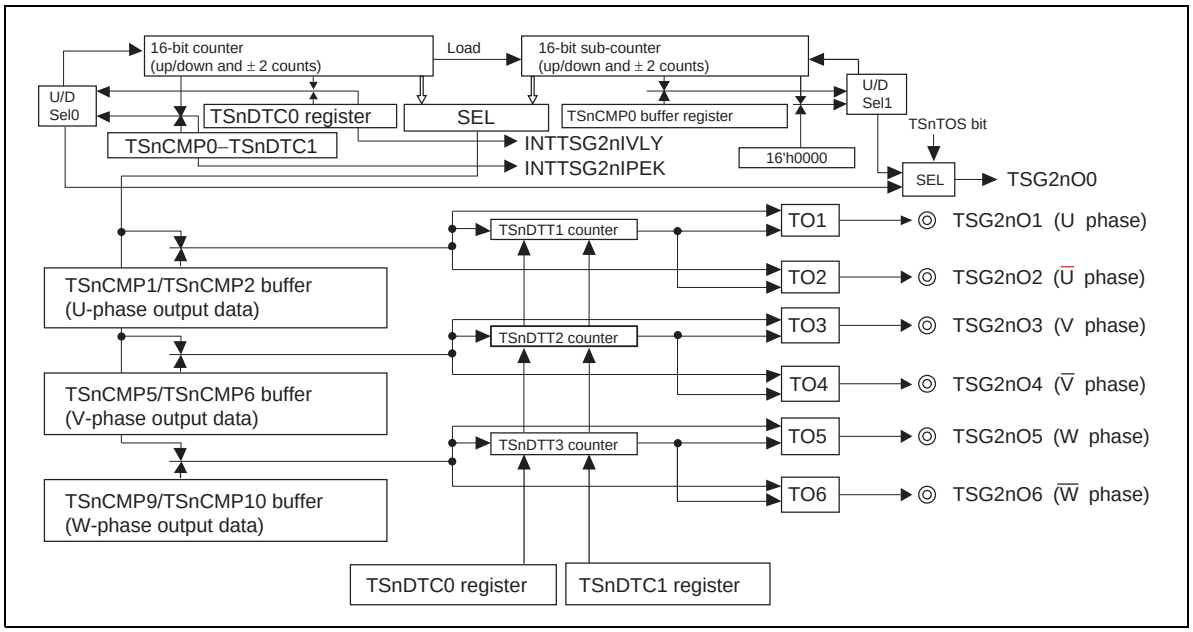


Figure 15-46 Block Diagram in HT-PWM Mode

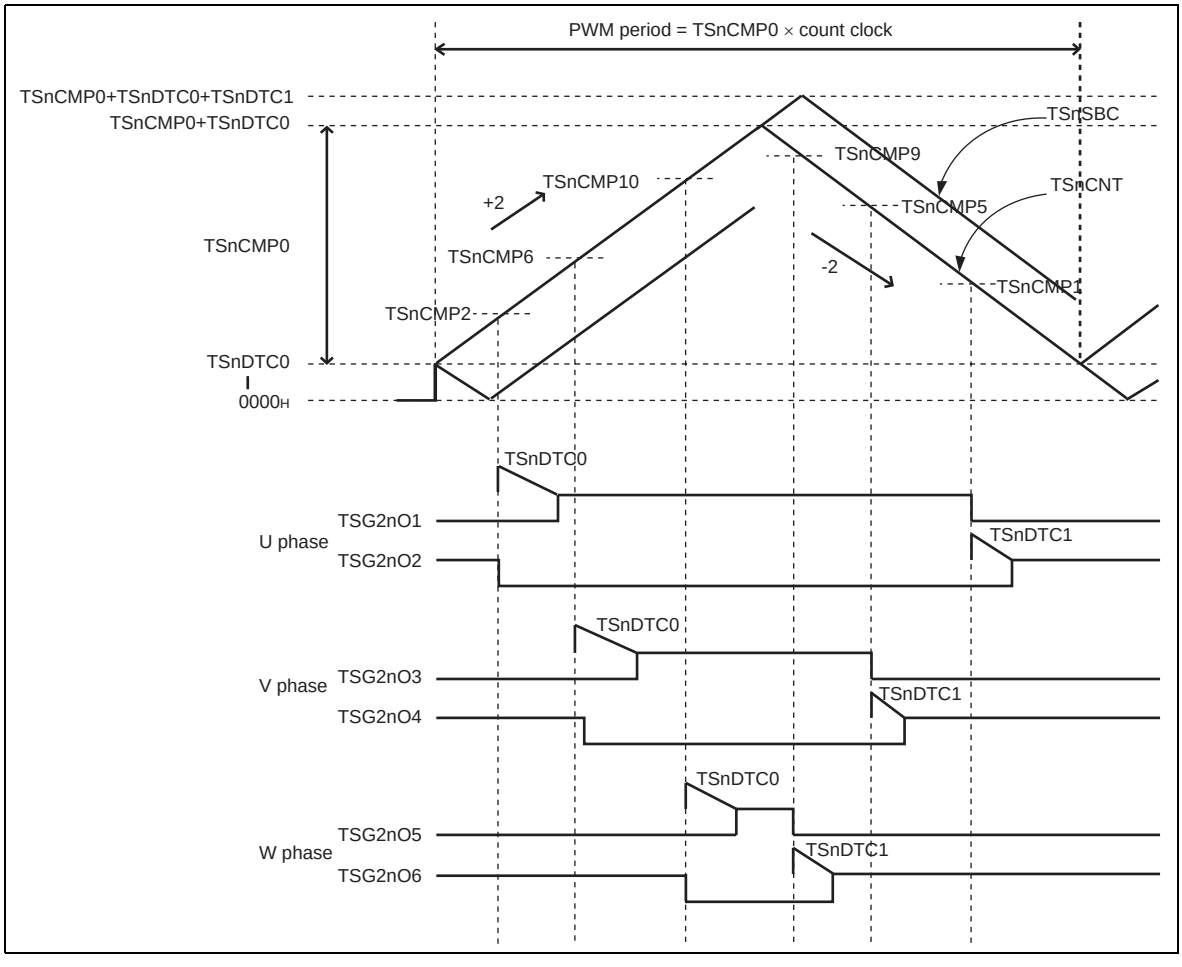


Figure 15-47 Basic Timing in HT-PWM Mode

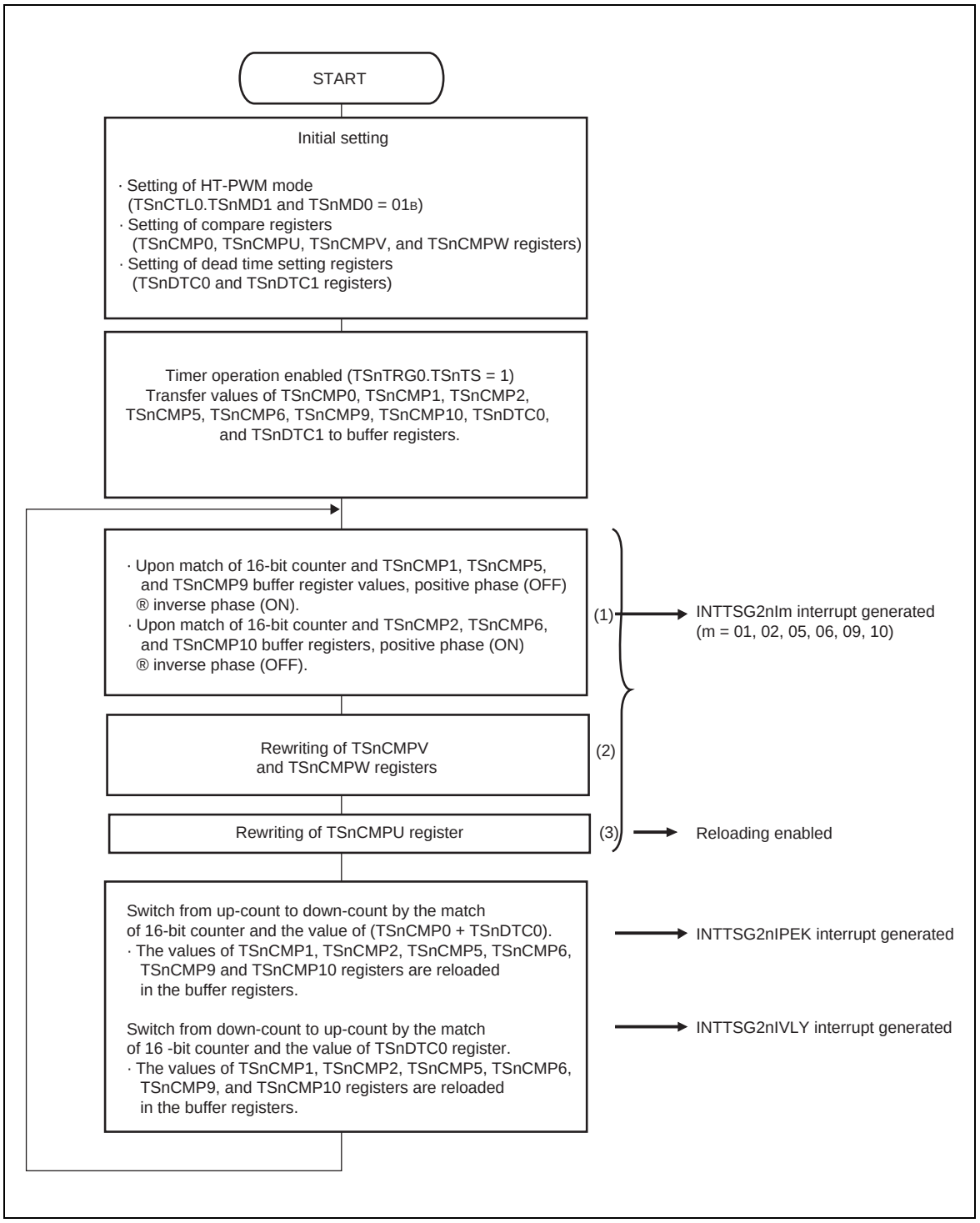


Figure 15-48 Basic Operation Flow in HT-PWM Mode

Note
- Write access to TSnCMPU (TSnCMP1) includes reloading enabling operation. Therefore, (3) must be done after (2).
- The INTTSG2nIPEK interrupt is generated only when TSnCTL4.TSnPRE = 1.
- The INTTSG2nIVLY interrupt is generated only when TSnCTL4.TSnVRE = 1.

(2) List of HT-PWM Mode Operations

Table 15-57 Counter Function in HT-PWM Mode

Operation		Setting Condition
16-bit counter	Start	TSnTRG0.TSnTS = 0 → 1 (up count from TSnDTC0)
	Up count	Compare match of TSnDTC0 buffer register and 16-bit counter
	Down count	Compare match of TSnCMP0 + TSnDTC0 and 16-bit counter
	Clear	-
	Stop	TSnTRG1.TSnTT = 0 → 1
16-bit sub-counter	Start	TSnTRG0.TSnTS = 0 → 1 (down count from TSnDTC0)
	Up count	Underflow
	Down count	Compare match of TSnCMP0 + TSnDTC0 + TSnDTC1 buffer register and 16-bit sub-counter
	Load	• TSnCMP0 + TSnDTC0: When value of 16-bit counter matches the value of buffer register TSnCMP0 + TSnDTC0 • TSnDTC0: When value of 16-bit counter matches the value of the buffer register TSnDTC0
	Clear	-
	Stop	TSnTRG1.TSnTT = 0 → 1

Table 15-58 Compare Register and Dead Time Setting Register Functions in HT-PWM Mode

Register	Rewrite Method	Rewrite during Operation	Function
TSnCMP0	Reload/Anytime rewrite	Possible	Setting period
TSnCMPU	-	Possible	PWM control for U phase
TSnCMP1W (TSnCMP1, TSnCMP2)	Reload/Anytime rewrite		
TSnCMPV	-	Possible	PWM control for V phase
TSnCMP5W (TSnCMP5, TSnCMP6)	Reload/Anytime rewrite		
TSnCMPW	-	Possible	PWM control for W phase
TSnCMP9W (TSnCMP9, TSnCMP10)	Reload/Anytime rewrite		
TSnDCMP0W, TSnDCMP2	Reload/Anytime rewrite	Possible	Diagnostic signal output or A/D conversion trigger
TSnDTC0, TSnDTC1	Reload	Possible conditionally	Period and dead time setting

Note
- The values written to TSnCMPU, TSnCMPV, and TSnCMPW registers are set to both the lower and upper 16-bits of the respective TSnCMP1W (TSnCMP1, TSnCMP2), TSnCMP5W (TSnCMP5, TSnCMP6), and TSnCMP9W (TSnCMP9, TSnCMP10) registers.
- Please refer to (8) (a), TSnDTC0 and TSnDTC1 Rewriting, on how to rewrite the TSnDTC0 and TSnDTC1.

Table 15-59 Timer Input Function in HT-PWM Mode

Pin	Function
TSG2nCLKI	- (cannot be used in HT-PWM mode)

Table 15-60 Timer Output Function in HT-PWM Mode

Pin	Function
TSG2nO0	Active level output during up count, inactive level output at down count of the 16-bit counter/sub-counter
TSG2nO1	PWM output with dead time by compare match of TSnCMP1 buffer register and 16-bit counter (down count) and compare match of TSnCMP2 buffer register and 16-bit counter (up count)
TSG2nO2	Inverse phase output to TSG2nO1
TSG2nO3	PWM output with dead time by compare match of TSnCMP5 buffer register and 16-bit counter (down count) and compare match of TSnCMP6 buffer register and 16-bit counter (up count).
TSG2nO4	Inverse phase output to TSG2nO3
TSG2nO5	PWM output with dead time by compare match of TSnCMP9 buffer register and 16-bit counter (down count) and TSnCMP10 buffer register and 16-bit counter (up count).
TSG2nO6	Inverse phase output to TSG2nO5
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Note State of TSG2nO0 output can be switched with TSnIOC1.TSnTOS.

Table 15-61 Interrupt Request in HT-PWM Mode

Interrupt	Function
INTTSG2nI00	Compare match of TSnDTC0 buffer register and 16-bit counter (periodic interrupt)
INTTSG2nIm (m = 01, 02, 05, 06, 09, 10)	Compare match of TSnCMPm buffer register and 16-bit counter
INTTSG2nIER	Error interrupt
INTTSG2nIVLY	Valley interrupt
INTTSG2nIPEK	Peak interrupt
INTTSG2nIWN	Warning interrupt

Table 15-62 Compare Match Timing in HT-PWM Mode

Compare Match	Timing
TSnCMP0	When 16-bit counter changes from TSnCMP0 + TSnDTC0 to TSnCMP0 + TSnDTC0 - 2
TSnCMPm (m = 1, 2, 5, 6, 9, 10)	When 16-bit counter changes from TSnCMPm to TSnCMPm ± 2 (m = 1, 2, 5, 6, 9, 10)

Table 15-63 Example of Setting Each Timer Output Condition in HT-PWM Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG2nO0	Toggle output	TSnCMP0 × count clock	Output an active level during up count, and an inactive level during down count	-
TSG2nO1, TSG2nO3, TSG2nO5	PWM output	TSnCMP0 × count clock	Output an inactive level throughout one period (0% duty)	TSnCMPm = TSnCMP0 + TSnDTC0 + TSnDTC1 (m = U, V, W)
			Output an active level of one count clock in one period	TSnCMPm = TSnCMP0 - 1 (m = U, V, W)
			Output an inactive level of one count clock in one period	TSnCMPm = 0001_H (m = U, V, W)
			Active level during all period (100% duty)	TSnCMPm = 0000_H (m = U, V, W)
TSG2nO2, TSG2nO4, TSG2nO6	PWM output		Inactive level during all period (0% duty)	TSnCMPm = 0000_H (m = U, V, W)
			Output an active level of one count clock in one period	TSnCMPm = TSnDTC0 + TSnDTC1 + 1 (m = U, V, W)
			Output an inactive level of one count clock in one period	TSnCMPm = TSnCMP0 + TSnDTC0 + TSnDTC1 - 1 (m = U, V, W)
			Active level during all period (100% duty)	TSnCMPm = TSnCMP0 + TSnDTC0 + TSnDTC1 (m = U, V, W)
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger	TSnCMP0 × count clock	Please refer to section 15.9, A/D Conversion Trigger Function.	

(3) Various Settings of HT-PWM Mode

Mode Setting

HT-PWM can be used when TSnCTL0.TSnMD1 and TSnMD0 is set to 01_B.

Setting timer output

The output pins TSG2nO1 to TSG2nO6 are controlled by setting TSnIOC0, TSnIOC2, and TSnIOC3.

The output pin TSG2nO0 indicates the up/down count status of the 16-bit counter or the 16-bit sub-counter. Switch between the 16-bit sub-counter and the 16-bit counter is done with the TSnIOC1.TSnTOS bit.

The TSG2nO7 pin outputs pulses of the diagnostic output or the A/D conversion trigger. Please set it as required.

Enabling error interrupt generation

Error interrupt (INTTSG2nIER) due to the detection of the simultaneous active state of the positive phase and inverse phase is enabled by setting TSnIOC1.TSnEOC to 1. In HT-PWM mode, with any value set in the compare register, the simultaneous active state of the positive phase and inverse phase is not possible. Please refer for to section 15.10, Error/Warning Interrupt, for details.

Setting register rewriting timing with reload function

With TSnCTL3.TSnRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSnCTL4.TSnPRE or TSnVRE to 1.

The reload timing is not generated if both the TSnPRE bits and TSnVRE bits are set to 0.

When "anytime rewrite" is specified, the unintended output may be generated depending on the rewrite timing.

Setting interrupts and skipping function

Interrupts and the skipping function are set with TSnCTL4. TSnPIE should be set to 1 when peak interrupt (INTTSG2nIPEK) is necessary and TSnVIE to 1 when valley interrupt (INTTSG2nIVLY) is necessary. To use the skipping function for peak/valley interrupts, the TSnRCC4 to TSnRCC0 must be set.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSnADTRG0 signal), use TSnCTL5.TSnAT09 to TSnAT00.

With TSnAT09 to TSnAT00, A/D conversion trigger output is enabled or disabled at the match of 16-bit counter (during up count) with TSnDCMP2 to TSnDCMP0, the match of the 16-bit counter (during down count) with TSnDCMP2 to TSnDCMP0, the 16-bit counter peak interrupt (INTTSG2nIPEK), the 16-bit counter valley interrupt (INTTSG2nIVLY), the 16-bit sub-counter peak timing, and 16-bit sub-counter valley timing.

To set A/D conversion trigger 1 (TSnADTRG1 signal), use TSnCTL6.TSnAT19 to TSnAT10.

To set the match timing of 16-bit counter and TSnDCMP2 to TSnDCMP0, set the compare value to the pertinent register.

The skipping function can be used for TSnADTRG0 and the TSnADTRG1 signals. Use TSnACC00, TSnACC01 of TSnCTL5, TSnACC10, and TSnACC11 of TSnCTL6 to select the skipping rate among 1/1, 1/2, 1/4, and 1/8.

Caution Set TSnCTL5, TSnCTL6, and TSnDCMP2 to TSnDCMP0 correctly when using the TSG2nO7 output for the A/D conversion trigger timing pulse.

Setting dead time

The dead time can be set with TSnDTC0 and TSnDTC1.

The dead time is calculated by the following expressions:

PCLK × TSnDTC0

PCLK × TSnDTC1

TSnDTC0 can set the time between a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the inactive state and a change of TSG2nO1, TSG2nO3, and TSG2nO5 to the active state, respectively.

TSnDTC1 can set the time between a change of TSG2nO1, TSG2nO3, TSG2nO5 to the inactive state and a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the active state, respectively.

TSnDTC0 and TSnDTC1 can only be set to an even value.

Carrier period

Set the carrier period with TSnCMP0 according to the following expression:

TSnCMP0 = Carrier period/count clock period (PCLK)

Satisfy the following requirements when setting the TSnCMP0 register regarding the dead time:

- TSnCMP0 + TSnDTC0 + TSnDTC1 ≤ $FFFE_H$
- TSnCMP0 > TSnDTC0
- TSnCMP0 > TSnDTC1
- TSnCMP0 > 3 × MAX (TSnDTC0, TSnDTC1)
- TSnCMP0: Even number

Note MAX (A, B) indicates the larger value of A and B.

Duty (PWM width) setting

The duty of the U phase, the V phase, and the W phase is set with TSnCMPm (m = U, V, W, or 1, 2, 5, 6, 9, and 10), respectively. The setting range of the compare registers is as follows:

0000_H ≤ TSnCMPm ≤ TSnCMP0 + TSnDTC0 + TSnDTC1

LSB (least significant bit) of TSnCMPU, TSNCMPV, and TSnCMPW indicates the setting of an additional pulse. When TSnCMPU = 0003_H, the change in the inverse phase (TSG2nO2 output) is done one count clock later compared to the TSnCMPU = 0002_H setting (when the 16-bit counter is up-counting). The additional pulse cannot be set to TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, or TSnCMP10 (only even numbers can be set to these registers).

(4) 16-Bit Counter Operation in HT-PWM Mode

The 16-bit counter is initialized to 0000_H and the value of TSnDTC0 is loaded immediately after starting the TSG2n timer operation (TSnTRG0.TSnTS = 1). Afterwards, counting is done by +2. After 16-bit counter reaches the value of TSnCMP0 + TSnDTC0, counting is done by -2.

Figure 15-49 shows 16-bit counter operation.

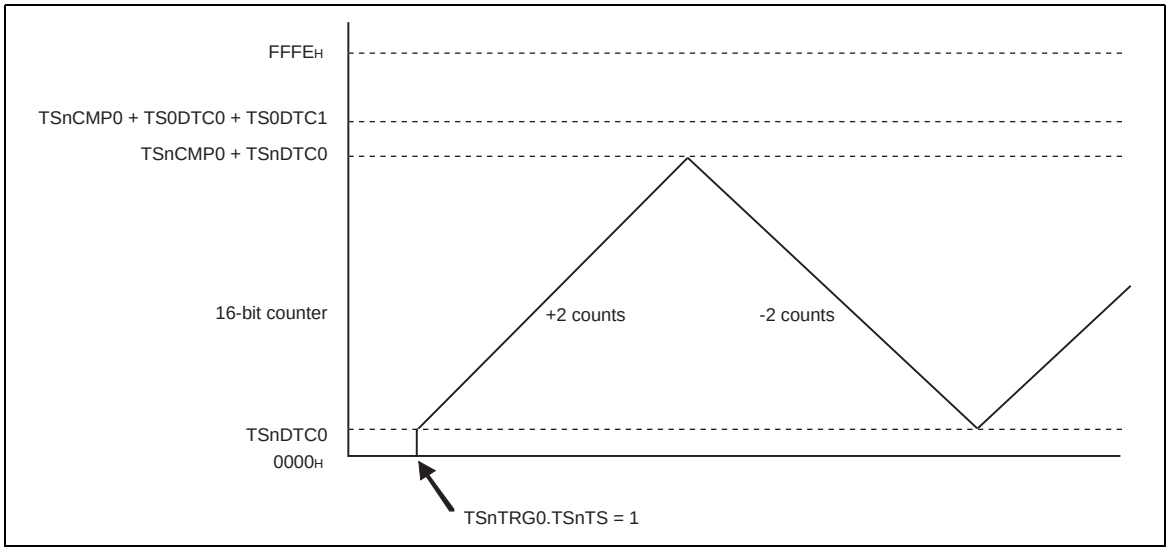


Figure 15-49 Example of 16-Bit Counter Operation in HT-PWM Mode

Note
Minimum 16-bit counter value: TSnDTC0
Maximum 16-bit counter value: TSnCMP0 + TSnDTC0
Carrier period: TSnCMP0 x count clock period (PCLK)

The 16-bit sub-counter is initialized to 0000_H and the value of TSnDTC0 is loaded immediately after starting the TSG2n timer operation (TSnTRG0.TSnTS = 1). Afterwards, counting by -2 is done until 0000_H is reached and counting by +2 begins. Next, the value of the 16-bit counter is loaded into the 16-bit sub-counter at a change timing of the 16-bit counter into the down count. Counting up by the 16-bit sub-counter continues until the value reaches the value of TSnCMP0 + TSnDTC0 + TSnDTC1, and then counting by -2 begins. Similarly, when the 16-bit counter value matches the TSnDTC0 value, the 16-bit counter value is loaded to the 16-bit sub-counter and the down count is continued.

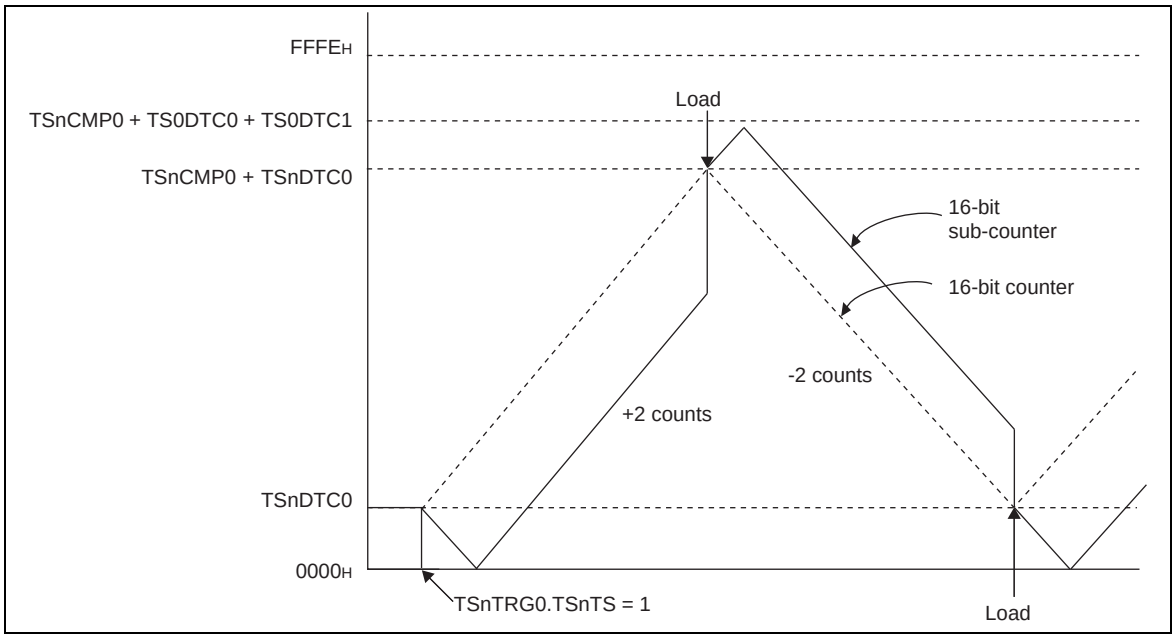


Figure 15-50 Example of 16-bit Sub-Counter Operation in HT-PWM Mode

Note Minimum 16-bit sub-counter value: 0000_H
Maximum 16-bit sub-counter value: TSnCMP0 + TSnDTC0 + TSnDTC1

(5) Basic Operation of HT-PWM Mode

(a) Example of Timer Output Immediately after the Start of the TSG2n Timer Operation

Figure 15-51 shows the timing chart when TSnCMP0 = $000E_H$, TSnDTC0 = 0002_H, TSnDTC1 = 0004_H and TSnCMPU = 0000_H-0014_H (excerpt). In this example, TSnIOC2.TSnOL1 - TSnOL6 = 000000_B.

In the case of TSnCMPU = TSnDTC0, when the initial value of the counter is loaded after TSG2n begins operation (TSnTRG0.TSnTS = 1), the TSG2nO2 output is changed to the active level. When TSnCMPU = TSnDTC0-0001_H, an additional pulse is generated, the TSG2nO2 output is changed one count clock cycle later compared to the case of TSnCMPU = TSnDTC0-0002_H.

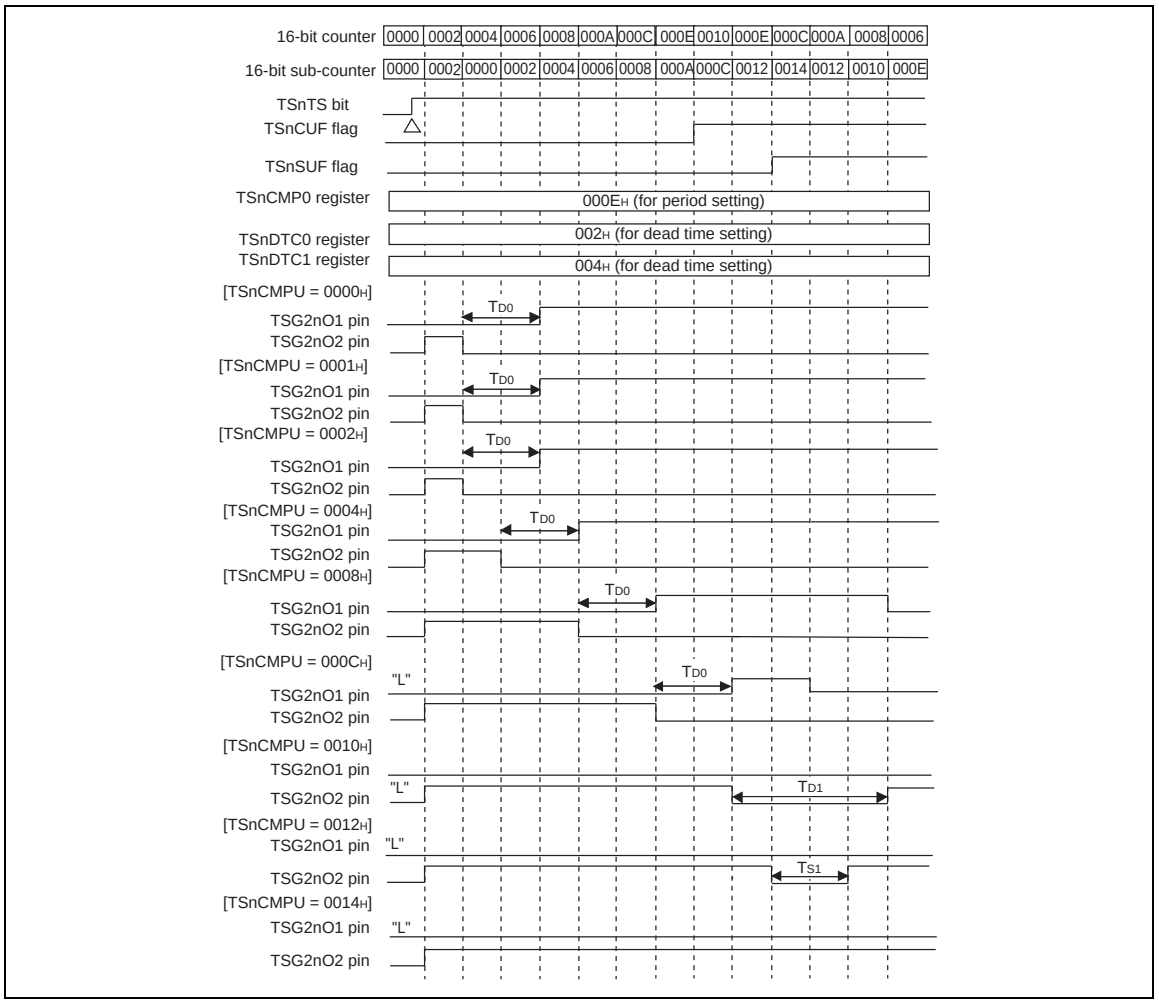


Figure 15-51 Example of Timer Output when TSnTS is Set to 1 (Initial) in HT-PWM Mode

Note 1. TSnCMP0 = $000E_H$, TSnDTC0 = 0002_H, TSnDTC1 = 0004_H

Note 2. T_{D0}: Time depending on setting of the dead time in the TSnDTC0 register
T_{D1}: Time depending on setting of the dead time in the TSnDTC1 register
T_{S1}: Time decided by compare match of the 16-bit sub-counter and TSnCMPU register, when TSnCMPU >16-bit counter maximum value

Note 3. Δ: Write access

(b) Example of Timer Output during TSG2n Timer Operation

The figure below shows the timing chart when TSnCMP0 = $000E_H$, TSnDTC0 = 0002_H, TSnDTC1 = 0004_H, and TSnCMPU is set to 0000_H-0014_H (excerpt). In this example, TSnIOC2.TSnOL1-TSnOL6 = 000000_B.

The range of the active (high level) width of a positive phase (TSG2nO1) output is $0000_H \leq$ TSnCMPU $\leq$ TSnCMP0 (for the additional pulse). The range of the active (high level) width of an inverse phase (TSG2nO2) output is TSnDTC0 + TSnDTC1 $\leq$ TSnCMPU $\leq$ TSnCMP0 + TSnDTC0 + TSnDTC1.

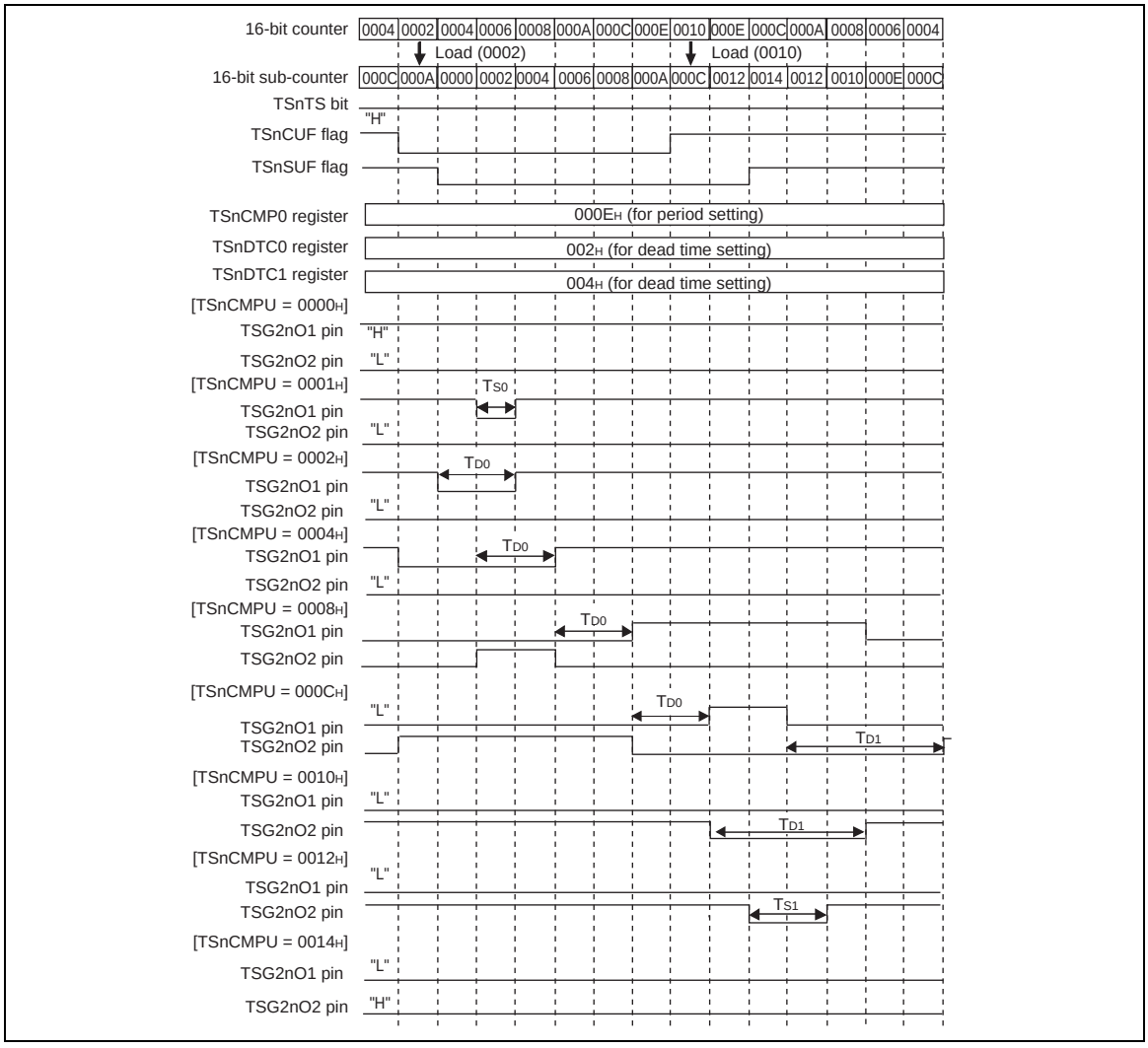


Figure 15-52 Example of Timer Output during TSG2n Operation in HT-PWM Mode

Note 1. TSnCMP0 = $000E_H$, TSnDTC0 = 0002_H, TSnDTC1 = 0004_H

Note 2. T_{D0}: Time depending on setting of the dead time in the TSnDTC0 register
T_{D1}: Time depending on setting of the dead time in the TSnDTC1 register
T_{S0}: Time decided by compare match of 16-bit sub-counter and the TSnCMPU register, when TSnCMPU < 16-bit counter minimum value
T_{S1}: Time decided by compare match of the 16-bit sub-counter and the TSnCMPU register, when TSnCMPU > 16-bit counter maximum value

(6) Additional Pulse Control in HT-PWM Mode

The HT-PWM mode can generate an additional pulse by setting 1 to the LSB of the duty setting registers (TSnCMPU, TSnCMPV, and TSnCMPW). This allows more precise control of the pulse duty than standard pulse control.

The following sections describe two examples of pulse output of TSG2nO1: additional pulse control is used in one example and additional pulse control is not used in another.

(a) Example of Pulse Output when Additional Pulse Control Is Used

Figure 15-53 shows the additional pulse control when an odd value is set to TSnCMPU.

The arrows and numerical values show the width of the duty cycle of the TSG2nO1 output in one period.

When the additional pulse control is used as shown in Figure 15-53, the width of the output of the TSG2nO1 (duty cycle) can be set within a range from the width of 12 clock cycles to the width of 0 clock cycles in one-clock-cycle step.

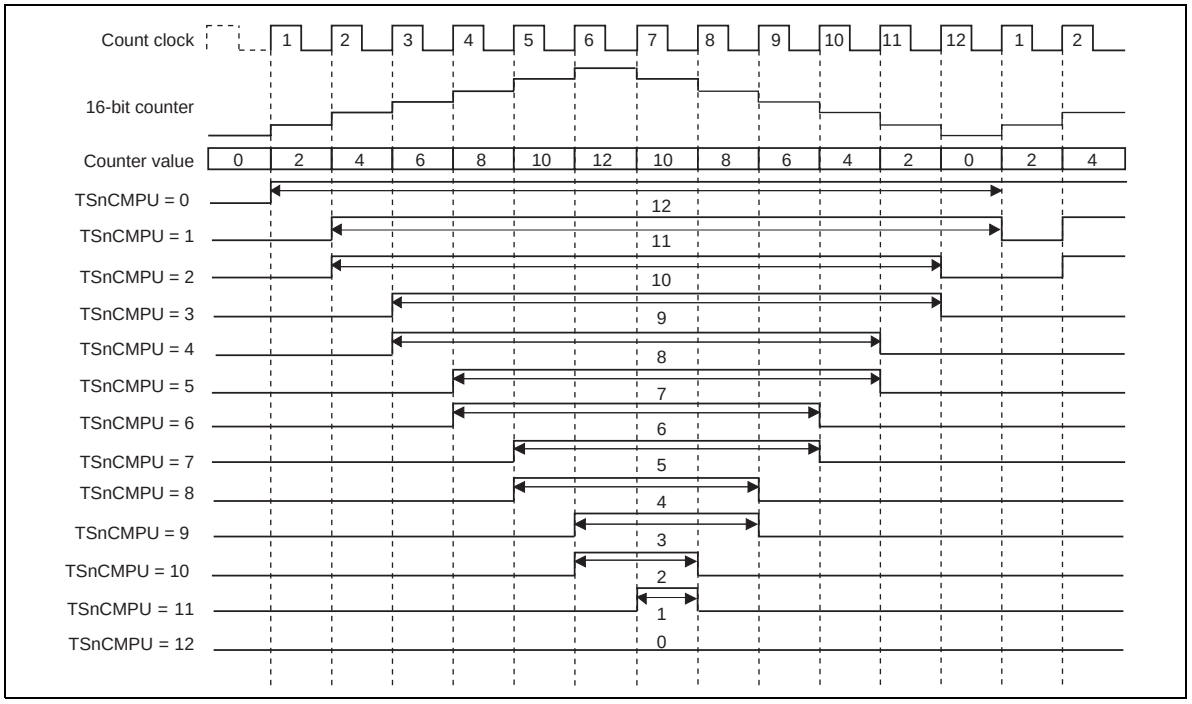


Figure 15-53 Example of TSG2nO1 Output when Additional Pulse Control Is Used in HT-PWM Mode

Note TSnCMP0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

(b) Example of Pulse Output when Additional Pulse Control Is Not Used

The arrows and numerical values in Figure 15-54 show the width of the duty cycle of the TSG2nO1 output in one period.

When the additional pulse control is not used, the width of the TSG2nO1 output can be set within a range from the width of 12 clock cycles to the width of 0 clock cycles in two-clock-cycle step. In this case, the change in duty cycle is larger than that in the case when the additional pulse control is used.

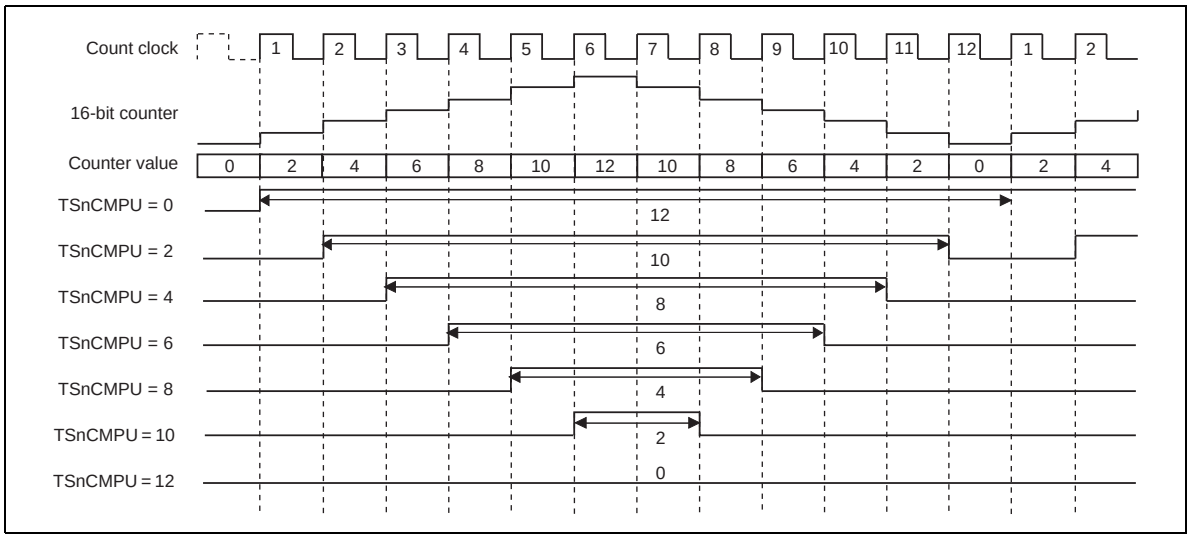


Figure 15-54 Example of Output when Additional Pulse Control Is Not Used in HT-PWM Mode

Note TSnCMP0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

(7) Dead Time Control in HT-PWM Mode

Duty setting registers are TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and TSnCMP10 and registers for setting the period are TSnCMP0, TSnDTC0, and TSnDTC1 in HT-PWM mode. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are two dead time setting registers (TSnDTC0 and TSnDTC1) and six 10-bit down counters that operate synchronously with the count clock of the 16-bit counter. TSnDTC0 is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSnDTC1 is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state.

Figure 15-55 shows the output waveform when TSnDTC0 = x and TSnDTC1 = y.

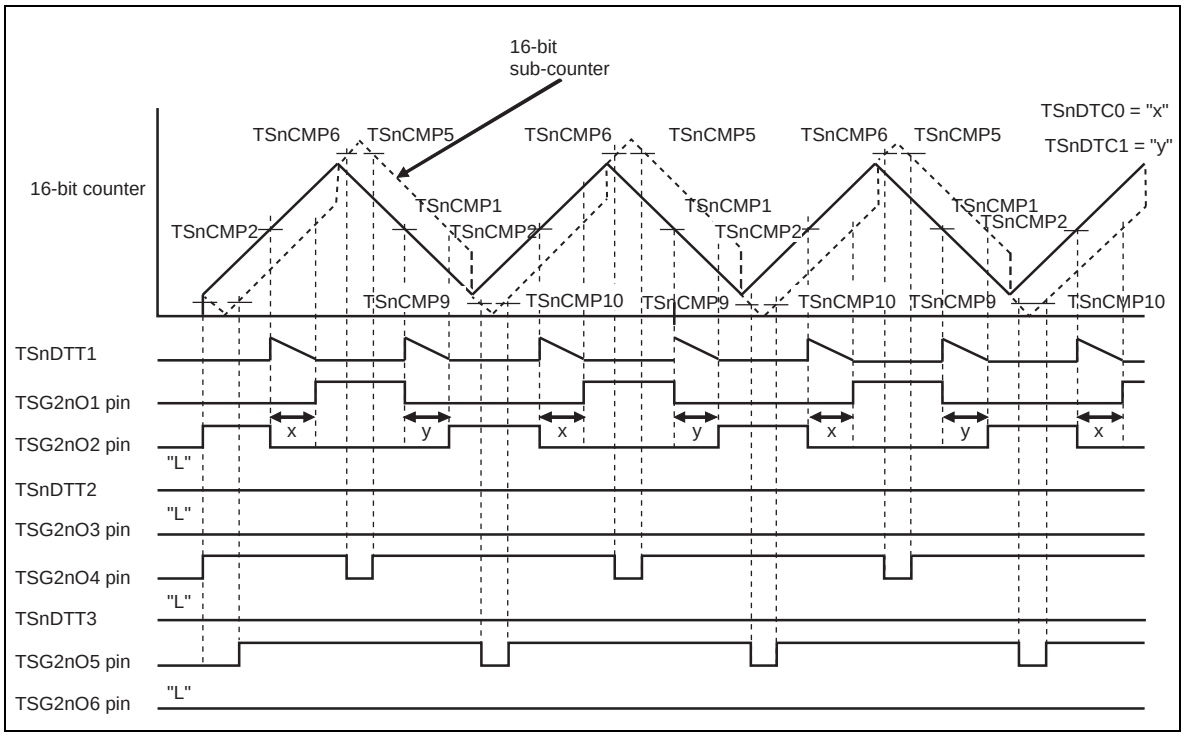


Figure 15-55 Example of Output Waveform with Dead Time in HT-PWM Mode

(8) Notes Concerning Dead Time Control in HT-PWM Mode

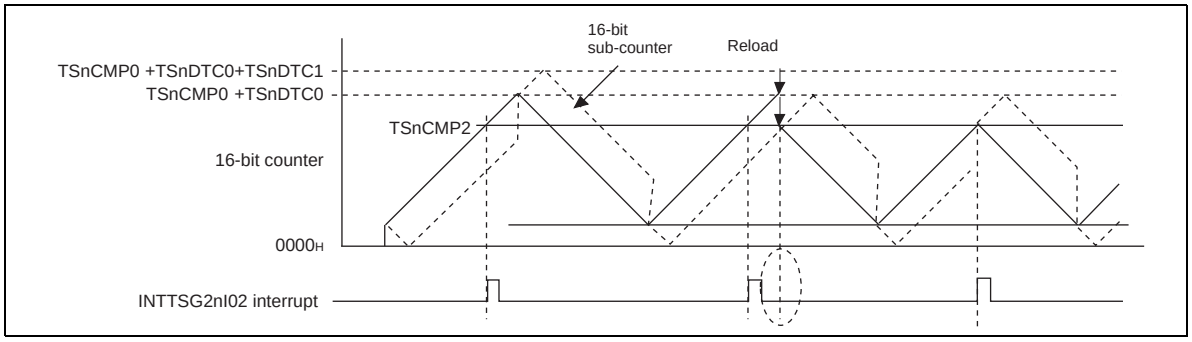
(a) TSnDTC0 and TSnDTC1 Rewriting

It is possible to rewrite the dead time setting in TSnDTC0 and TSnDTC1 registers during timer operation.

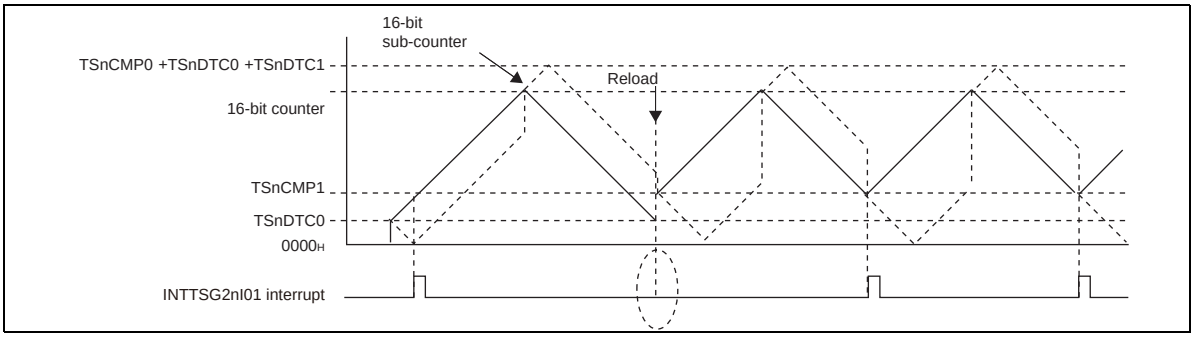
Caution 1. Rewrite TSnDTC0 and TSnDTC1 when the reload function is used (TSnRMC = 0).

Caution 2. The write protection code check function is applied when TSnDTC0 and TSnDTC1 are rewritten. Refer to the pertinent register description for details.

Caution 3. When the TSnCMP0 and TSnDTC1 are updated at the peak of the 16-bit counter:
When a match occurs between the TSnCMPm set value and the updated TSnCMP0-TSnDTC1 (new maximum value with the main counter), the match interrupt (INTTSG2nIm) will not be generated immediately after reload (m = 02, 06, 10).



Caution 4. When the TSnDTC0 is updated at the valley of the 16-bit counter:
When a match occurs between the TSnCMPm set value and the updated TSnDTC0 (new minimum value with the main counter), the match interrupt (INTTSG2nIm) is not generated immediately after reload (m = 01, 05, 09).



(9) Software Output Control Function in HT-PWM Mode

TSnOPT0.TSnSOC, TSnIDC, and TSnOPT1.TSnSPC2-TSnSPC0 are used in HT-PWM mode for software control of timer output control.

As shown in Figure 15-56, with TSnSTE = 0, the output control is switched immediately when TSnSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSnSOC is set to 0, output control is retained. When the reload timing is generated, output control is switched to HT-PWM mode output control.

For details, refer to section 15.11.5, Software Output Control Function.

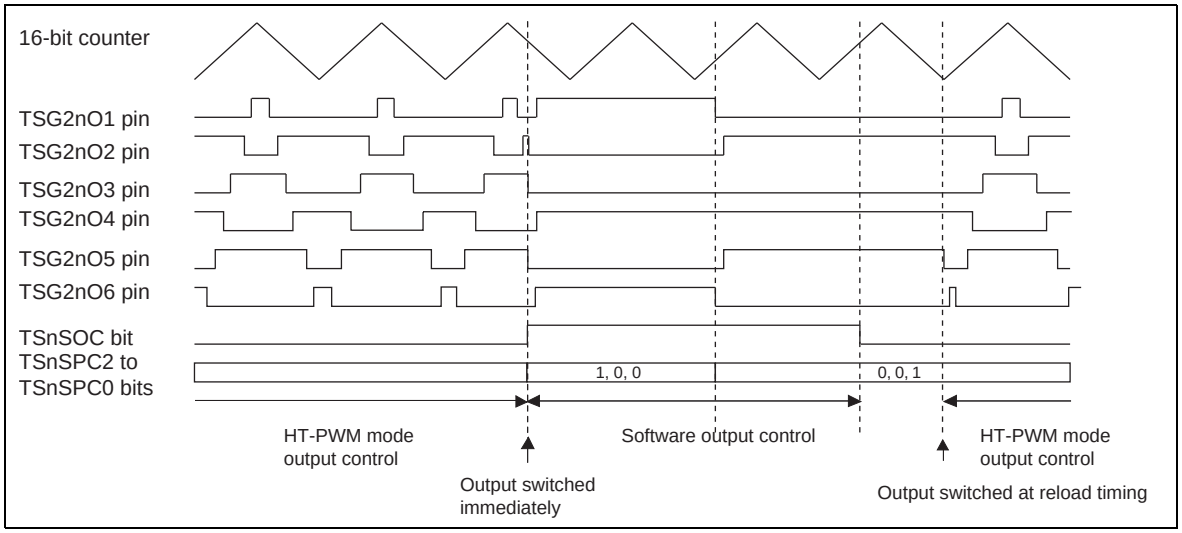


Figure 15-56 Example of Software Output Control Switching in HT-PWM Mode

Caution Use reload (simultaneous rewrite) mode (TSnCTL3.TSnRMC = 0) when software output control function is used.

(a) Procedure for Software Output Control

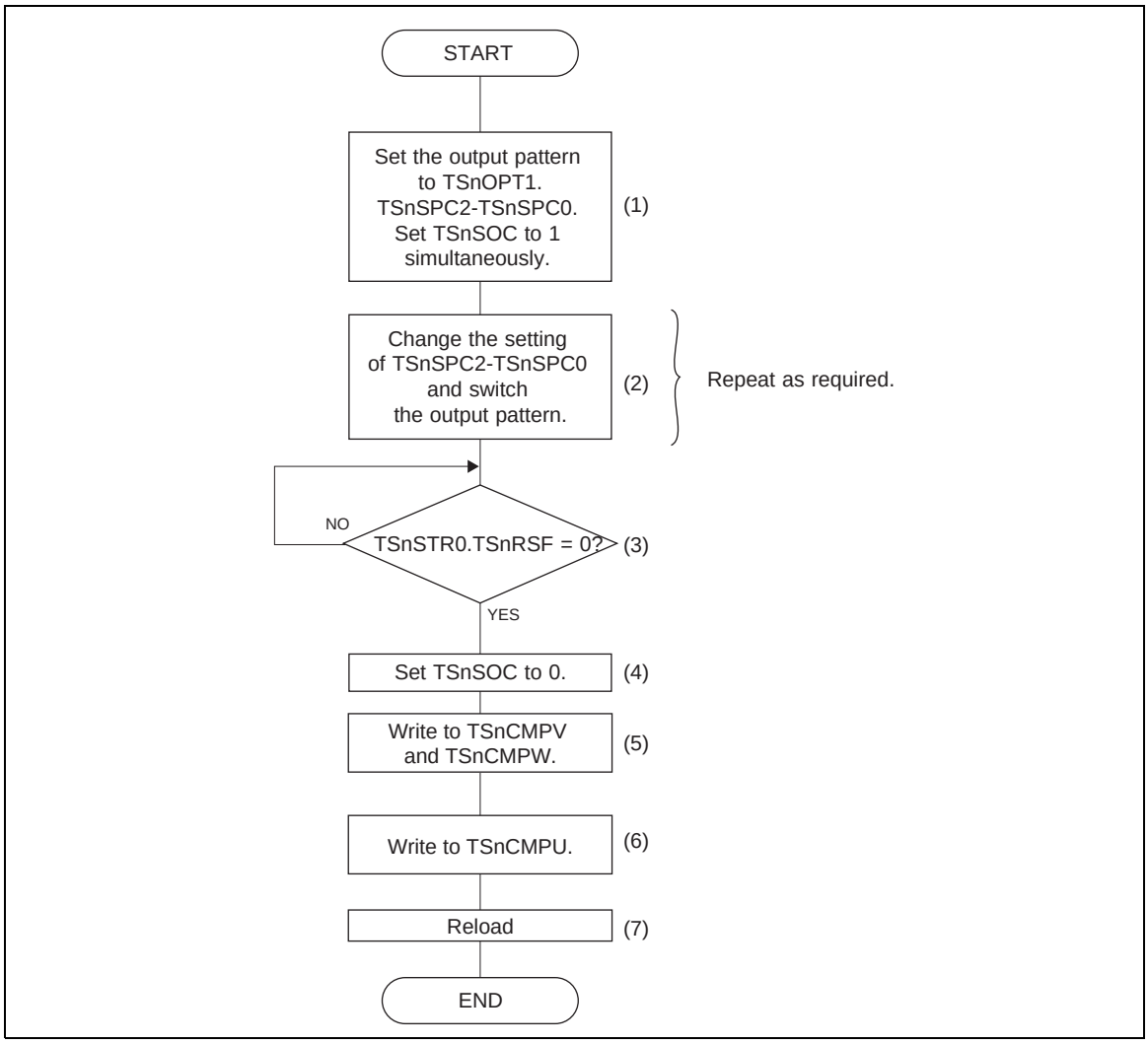


Figure 15-57 Flow of Software Output Control in HT-PWM Mode

The procedure for software output control is described below.

(1) Set the output pattern to the TSnOPT1.TSnSPC2-TSnSPC0. To enable software output control, set TSnOPT0.TSnSOC to 1 simultaneously.

(2) Change the output pattern setting of the TSnSPC2-TSnSPC0 to change the timer output. The registers that can be changed during software control are:
TSnTRG1.TSnTT, TSnCTL3-TSnCTL6, TSnOPT0, TSnOPT1, TSnIOC1.TSnTOS, TSnCMPm (m = 0, 1, 2, 5, 6, 9, 10), TSnDTC0, and TSnDTC1

(3) Confirm that reload request flag TSnSTR0.TSnRSF = 0. In case TSnRSF = 1, do not proceed to the following step until TSnRSF = 0.

(4) By setting TSnSOC = 0, the software control starts to be released (it is not released here yet).

(5) After releasing the software output control, set the necessary compare registers. Proceed to the following step when the register setting is not required. Here, change the registers with the reload function.

(6) Write to TSnCMPU (TSnCMP1) to start reloading.

(7) Reload is executed and software output is released.

Caution Execute reload after executing steps (3), (4), (5), (6), and (7). When reload cannot be executed, the software output cannot be released.

(10) Asymmetric Triangular Wave Control in HT-PWM Mode

In HT-PWM mode, it is possible to control output by an asymmetric triangular waveform by setting the different timings for setting and clearing the U, V, and W phases.

The following describes the differences of the asymmetric triangular wave control from the symmetric triangular wave control.

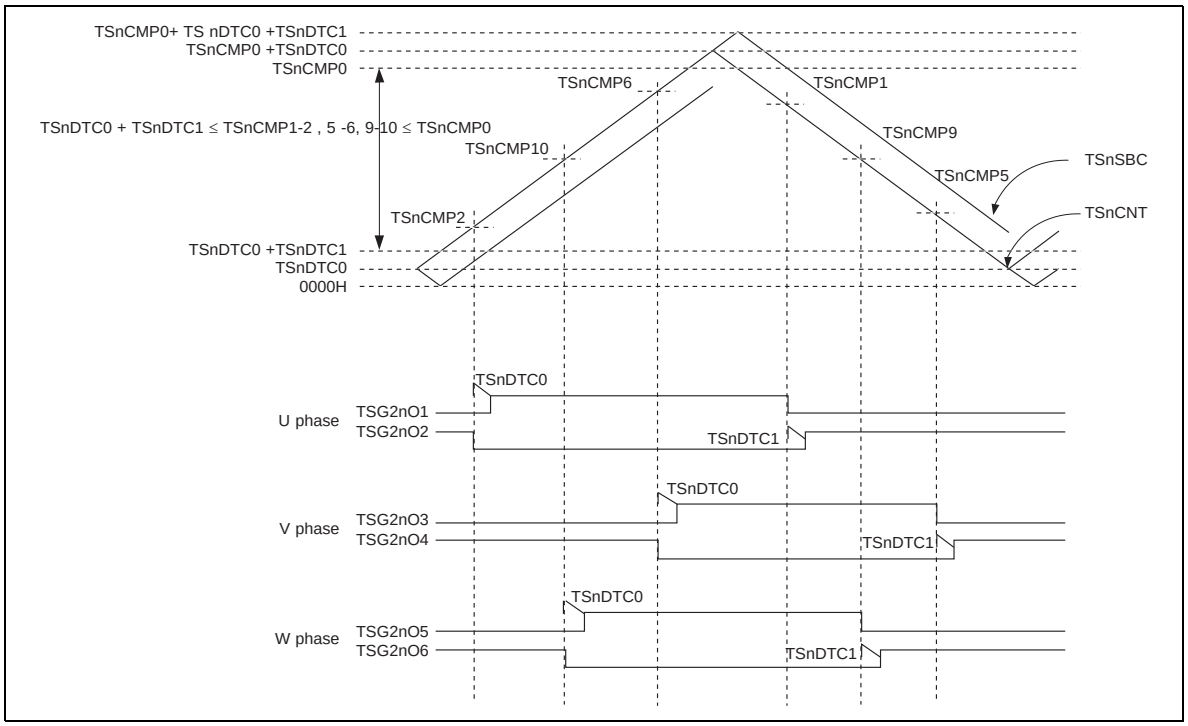
(a) PWM Setting

When a symmetric triangular wave is used, the output control of each of the U phase, V phase, and W phase is done by setting the set timing and the clear timing to the same value to the TSnCMPU, TSnCMPV, and TSnCMPW. When an asymmetric triangular wave is used, the output control of each phase is done by setting TSnCMPm as follows (m = 1, 2, 5, 6, 9, 10):

Prerequisites

- The clear timing of PWM of the voltage data signal of U, V and W phases is set with TSnCMP1, TSnCMP5, and TSnCMP9.
- The set timing of PWM of the voltage data signal of U, V, and W phases is set with TSnCMP2, TSnCMP6, and TSnCMP10.
- The set and the clear timings of each phase can also be set with TSnCMP1W (TSnCMP1 and TSnCMP2), TSnCMP5W (TSnCMP5 and TSnCMP6), and TSnCMP9W (TSnCMP9 and TSnCMP10)
- TSnCMPm can only be set to an even value (m = 1, 2, 5, 6, 9, 10).

(b) Timer Output



Note When output is controlled by the asymmetric triangular waveform, the following conditions apply to setting of TSnCMPm (m = 1, 2, 5, 6, 9, 10):

- TSnDTC0 + TSnDTC1 ≤ TSnCMPm ≤ TSnCMP0
- Only when TSnCMPm = TSnCMP(m+1) or TSnCMPm = TSnCMP(m+1) + 2, it is possible to set TSnCMPm under the condition 0000_H ≤ TSnCMPm ≤ TSnCMP0 + TSnDTC0 + TSnDTC1, which also applies to the case in which the symmetric triangular wave is used.

15.11.3 SP-PWM Mode (Shifted-pulse - Pulse Width Modulation Mode)

Overview

In this mode, a 6-phase PWM can be generated using the 16-bit counter and the 16-bit compare registers.

Prerequisites

- The PWM signal cycle is set in TSnCMP0.
- The set timings of the U phase, V phase, and W phase are set with TSnCMP2, TSnCMP6, and TSnCMP10, while the clear timings of these phases are set with TSnCMP1, TSnCMP5, and TSnCMP9 (when set timing and clear timing are used for control).
- The set timings of the U phase, V phase, and W phase are set with TSnCMP2, TSnCMP6, and TSnCMP10 while the active periods of these phases are set with TSnUPW, TSnVPW, and TSnWPW.
  The sum of the set values of TSnCMP2, TSnCMP6, TSnCMP10, and the set values of TSnUPW, TSnVPW, TSnWPW are set to TSnCMP1, TSnCMP5, and TSnCMP9 respectively (when set timing and active period are used for control).

Functional description

Set the carrier period and the set timings and duty of U phase, V phase, and W phase. The counting up begins when TSnTRG0.TSnTS is set to 1.

The 16-bit counter counts up from 0000_H and is cleared by match with TSnCMP0.

The dead time is set with TSnDTC0 and TSnDTC1. TSnDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch while TSnDTC1 sets that of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSnDTT1 to TSnDTT3) for dead time generation load the set values of TSnDTC0 and TSnDTC1 at compare match of the 16-bit counter with the TSnCMPm buffer register (m = 1, 2, 5, 6, 9, 10) and start down-counting.

INTTSG2nIm interrupts (m = 01, 02, 05, 06, 09, 10) are generated by the compare match of the 16-bit counter with the TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and TSnCMP10 buffer registers.

Note SP-PWM mode is enabled when TSnTRG0.TSnMD1 and TSnMD0 $=10_B$.

(1) Basic Timing Chart

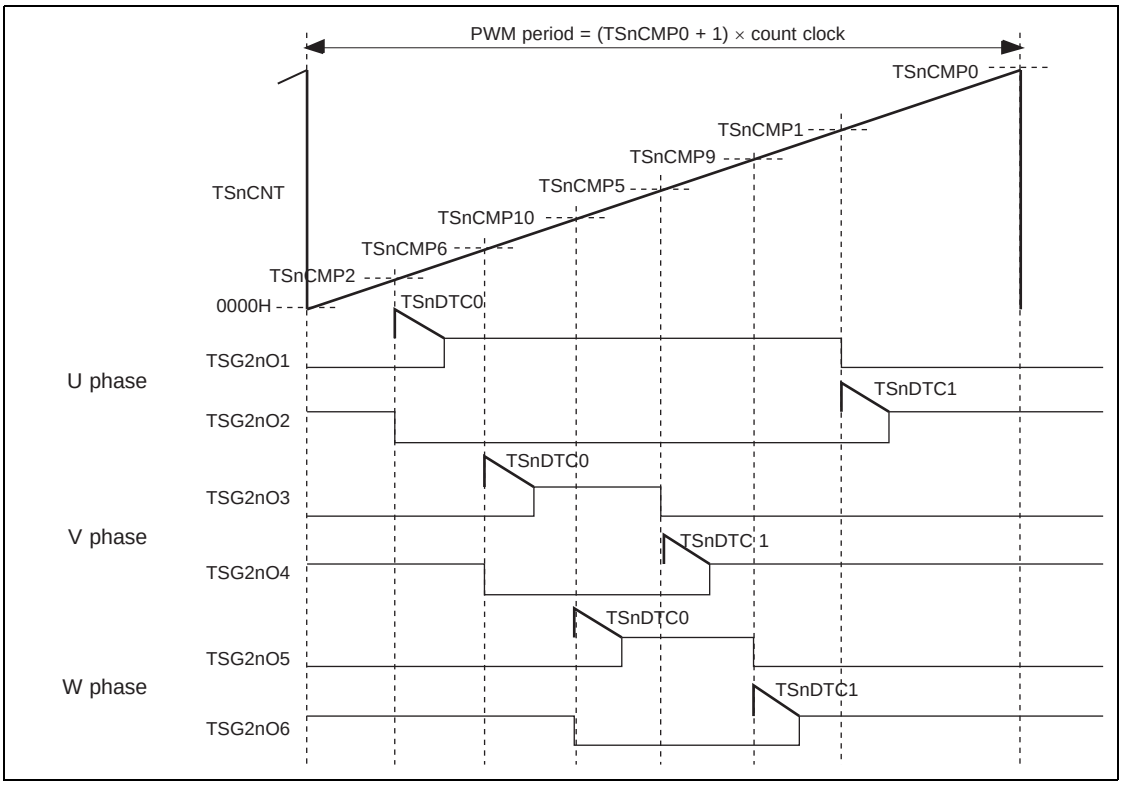


Figure 15-58 Basic Timing in SP-PWM Mode

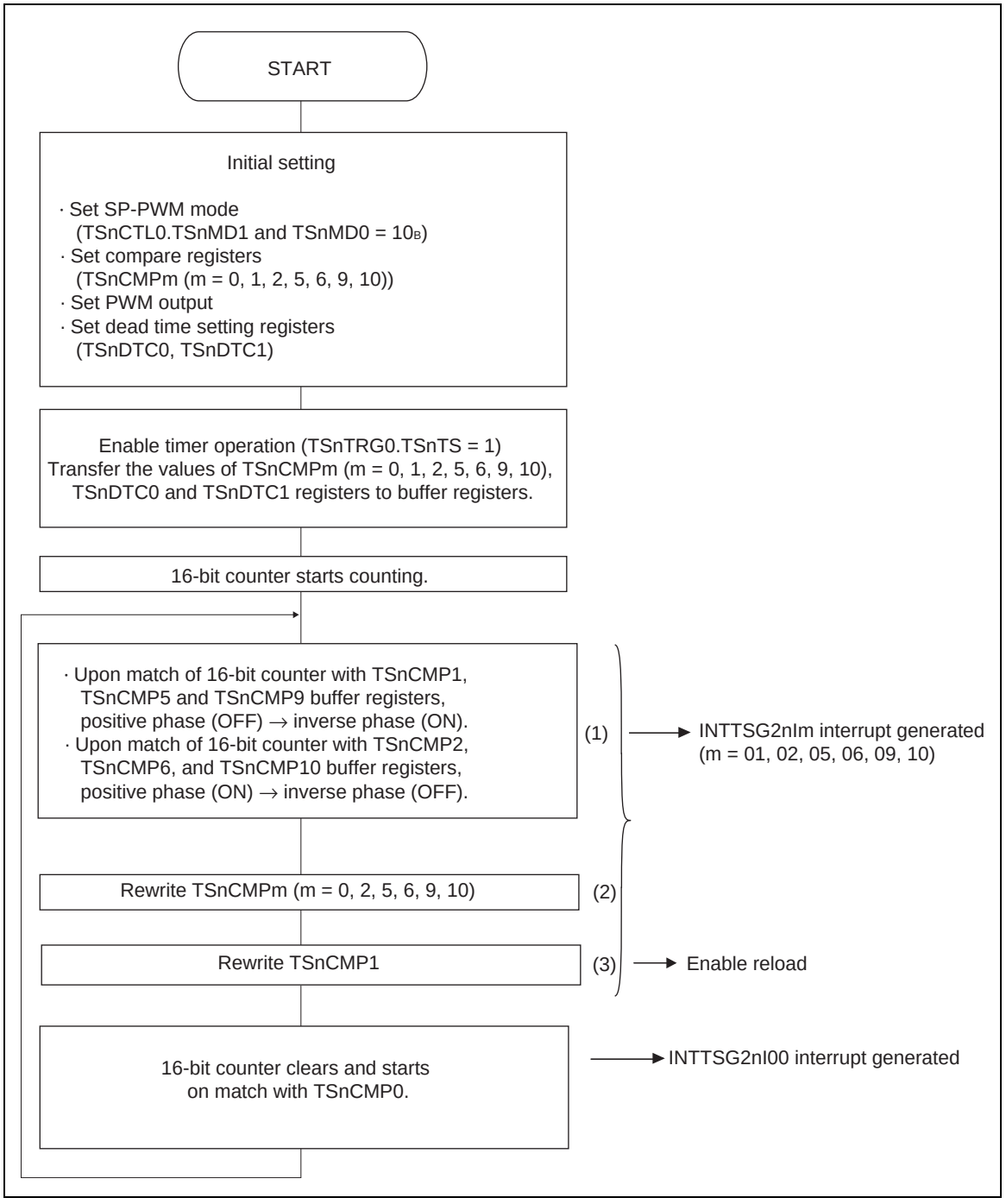


Figure 15-59 Basic Operation Flow in SP-PWM Mode

Note The timing of (1) may be different depending on the rewriting timing of (2) and (3).
Be sure to rewrite (2) followed by rewriting (3).

(2) List of SP-PWM Mode Operations

Table 15-64 Counter Functions in SP-PWM Mode

Operation		Setting Condition
16-bit counter	Start	TSnTRG0.TSnTS = 0 → 1
	Clear	Compare match of TSnCMP0 buffer register with 16-bit counter
	Stop	TSnTRG1.TSnTT = 0 → 1

Table 15-65 Compare Registers and Dead Time Setting Register Functions in SP-PWM Mode

Register	Rewriting Method	Rewrite during Operation	Function
TSnCMP0	Reload/Anytime rewrite	Possible	Setting period
TSnUPW	Reload/Anytime rewrite	Possible	PWM control for U phase
TSnCMP1W (TSnCMP1, TSnCMP2)	Reload/Anytime rewrite		
TSnVPW	Reload/Anytime rewrite	Possible	PWM control for V phase
TSnCMP5W (TSnCMP5, TSnCMP6)	Reload/Anytime rewrite		
TSnWPW,	Reload/Anytime rewrite	Possible	PWM control for W phase
TSnCMP9W (TSnCMP9, TSnCMP10)	Reload/Anytime rewrite		
TSnDTC0, TSnDTC1	Reload	Possible	Period and dead time

Table 15-66 Input Function in SP-PWM Mode

Pin	Function
TSG2nCLKI	Clock enable input

Table 15-67 Output Functions in SP-PWM Mode

Pin	Function
TSG2nO1	PWM output with dead time by compare match of TSnCMP1 buffer register (clear timing) or TSnCMP2 buffer register (set timing) with 16-bit counter
TSG2nO2	Output inverse phase with respect to TSG2nO1 (with dead time)
TSG2nO3	PWM output with dead time by compare match of TSnCMP5 buffer register (clear timing) or TSnCMP6 buffer register (set timing) with 16-bit counter
TSG2nO4	Output inverse phase with respect to TSG2nO3 (with dead time)
TSG2nO5	PWM output with dead time by compare match of the TSnCMP9 buffer register (clear timing) or the TSnCMP10 buffer register (set timing) with the 16-bit counter
TSG2nO6	Output inverse phase with respect to TSG2nO5 (with dead time)
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Table 15-68 Interrupt Requests in SP-PWM Mode

Interrupt	Function
INTTSG2nIm (m = 00, 01, 02, 05, 06, 09, 10)	Compare match of TSnCMPm buffer register with 16-bit counter (m = 0, 1, 2, 5, 6, 9, 10)
INTTSG2nIER	Error
INTTSG2nIVLY	-
INTTSG2nIPEK	Peak interrupt (generated at the same timing as INTTSG2nI00)
INTTSG2nIWN	Warning

Table 15-69 Compare Match Timing in SP-PWM Mode

Compare Match	Timing
TSnCMP0	When 16-bit counter changes from TSnCMP0 to 0000_H
TSnCMPm (m = 1, 2, 5, 6, 9, 10)	After match of 16-bit counter and TSnCMPm is detected (m = 1, 2, 5, 6, 9, 10)

Table 15-70 Example of Setting Each Timer Output Condition in SP-PWM Mode (1/2)

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG2nO1, TSG2nO3, TSG2nO5	PWM output	(TSnCMP0 + 1) × count clock	Output an inactive level throughout one period (duty 0%)	TSnCMPm = TSnCMP(m + 1) or TSnCMP(m + 1) > TSnCMP0 (m = 1, 5, 9)
			Output an active level of one count clock in one period	TSnCMPm = TSnCMP(m + 1) + 1 TSnCMP(m + 1) = TSnCMPm - 1 (m = 1, 5, 9)
			Output an inactive level of one count clock in one period	TSnCMPm = TSnCMP(m + 1) - 1 TSnCMP(m + 1) = TSnCMPm + 1 (m = 1, 5, 9)
			Output an active level throughout one period (duty 100%)	TSnCMPm > TSnCMP0 TSnCMP(m + 1) ≤ TSnCMP0 (m = 1, 5, 9)

Table 15-70 Example of Setting Each Timer Output Condition in SP-PWM Mode (2/2)

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG2nO2, TSG2nO4, TSG2nO6	PWM output	(TSnCMP0 + 1) × count clock	Output an inactive level throughout one period (duty 0%)	TSnCMPm = TSnCMP(m - 1) or TSnCMP(m - 1) > TSnCMP0 (m = 2, 6, 10)
			Output an active level of one count clock in one period	TSnCMPm = TSnCMP(m - 1) - 1 TSnCMP(m - 1) = TSnCMPm + 1 (m = 2, 6, 10)
			Output an inactive level of one count clock in one period	TSnCMPm = TSnCMP(m - 1) + 1 TSnCMP(m - 1) = TSnCMPm - 1 (m = 2, 6, 10)
			Output an active level throughout one period (duty 100%)	TSnCMPm > TSnCMP0 (m = 2, 6, 10)
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger	(TSnCMP0 + 1) × count clock	Please refer to section 15.9, A/D Conversion Trigger Function.	

(3) Various Settings of SP-PWM Mode

Mode Setting

SP-PWM mode is entered by setting TSnCTL0.TSnMD1-TSnMD0 to 10_B.

Setting timer output

The output pins TSG2nO1-TSG2nO6 are controlled by setting TSnIOC0, TSnIOC2, and TSnIOC3.

The TSG2nO7 pin provides output pulse as diagnostic output or analog to digital conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG2nIER) due to the detection of the simultaneous active state of the positive and inverse phases is enabled by setting TSnIOC1.TSnEOC to 1. For details, refer to section 15.10, Error/Warning Interrupt.

Setting rewriting timing of register with reload function

With the TSnCTL3.TSnRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSnCTL4.TSnPRE to 1.

No reload timing is generated when TSnPRE = 0.

When "anytime rewrite" is specified, the unintended output may be generated depending on the rewrite timing.

Setting analog to digital conversion trigger output

The analog to digital conversion trigger 0 (TSnADTRG0 signal) is set with TSnCTL5.TSnAT09 to TSnAT00.

TSnAT09 to TSnAT00 is used to enable or disable the A/D conversion trigger output on timing match of TSnDCMP2 to TSnDCMP0 with the 16-bit counter (up count).

TSnCTL6.TSnAT19 to TSnAT10 is used to set the analog to digital conversion trigger 1 (TSnADTRG1 signal).

To set the match timing of the 16-bit counter and TSnDCMP2 to TSnDCMP0, set the compare value to each register.

The skipping function can be used for TSnADTRG0 and TSnADTRG1 signals. TSnACC00 and TSnACC01 of TSnCTL5, and TSnACC10 and TSnACC11 of TSnCTL6 can be used to select the skipping rate among 1/1, 1/2, 1/4, and 1/8.

Caution

- Be sure to set TSnCTL5, TSnCTL6, and TSnDCMP2 to TSnDCMP0 correctly when the timing pulse of analog to digital conversion trigger is output to TSG2nO7.
- In SP-PWM mode, no valley interrupt (INTTSG2nIVLY) is generated. Therefore, TSnCTL5.TSnAT00 and TSnCTL6.TSnAT10 must be set to 0.
- In SP-PWM mode, the 16-bit sub-counter does not operate. Therefore, TSnAT09, and TSnAT08 in TSnCTL5, and TSnAT19, and TSnAT18 in TSnCTL6 must be set to 0.
- In SP-PWM mode, down counting by the 16-bit counter is not generated. Therefore, TSnAT07, TSnAT05, and TSnAT03 in TSnCTL5 and TSnAT17, TSnAT15, and TSnAT13 in TSnCTL6 should be set to 0.

Setting dead time

The dead time can be set with TSnDTC0 and TSnDTC1.

The dead time is calculated by the following expressions:

PCLK × TSnDTC0

PCLK × TSnDTC1

TSnDTC0 can set the time between a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the inactive state and a change of TSG2nO1, TSG2nO3, and TSG2nO5 to the active state, respectively.

TSnDTC1 can set the time between a change of TSG2nO1, TSG2nO3, and TSG2nO5 to the inactive state, and a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the active state, respectively.

Carrier period

Set the carrier period with TSnCMP0 according to the following expression:

TSnCMP0 = (carrier period/count clock period) - 1

Caution PWM output with 100% duty cannot be produced when TSnCMP0 = $FFFF_H$.

Setting duty (PWM width)

The duty of U phase, V phase, and W phase is set with TSnCMPm, TSnUPW, TSnVPW, and TSnWPW (m = 1, 2, 5, 6, 9, 10), respectively.

- The set timings of the U phase, V phase, and W phase are set with TSnCMP2, TSnCMP6, and TSnCMP10, and the clear timings are set with TSnCMP1, TSnCMP5, and TSnCMP9. (The set and clear timing setting is used for control.)
- The set timings of U phase, V phase, and W phase is set with TSnCMP2, TSnCMP6, and TSnCMP10 while the active periods of these phases are set with TSnUPW, TSnVPW, and TSnWPW.
  The sum of the set values of TSnCMP2, TSnCMP6, and TSnCMP10, and the set values of TSnUPW, TSnVPW, and TSnWPW are set to TSnCMP1, TSnCMP5, and TSnCMP9 respectively (when set timing and active period are used for control).

(4) Dead Time Control in SP-PWM mode

Duty setting registers are TSnCMPm (m = 1, 2, 5, 6, 9, 10), TSnUPW, TSnVPW, and TSnWPW and register for setting the period is TSnCMP0. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are six 10-bit down counters that operate synchronously with the count clock of the 16-bit counter and two dead time setting registers (TSnDTC0 and TSnDTC1). TSnDTC0 is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSnDTC1 is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state. Figure 15-60 shows an example of the output waveform.

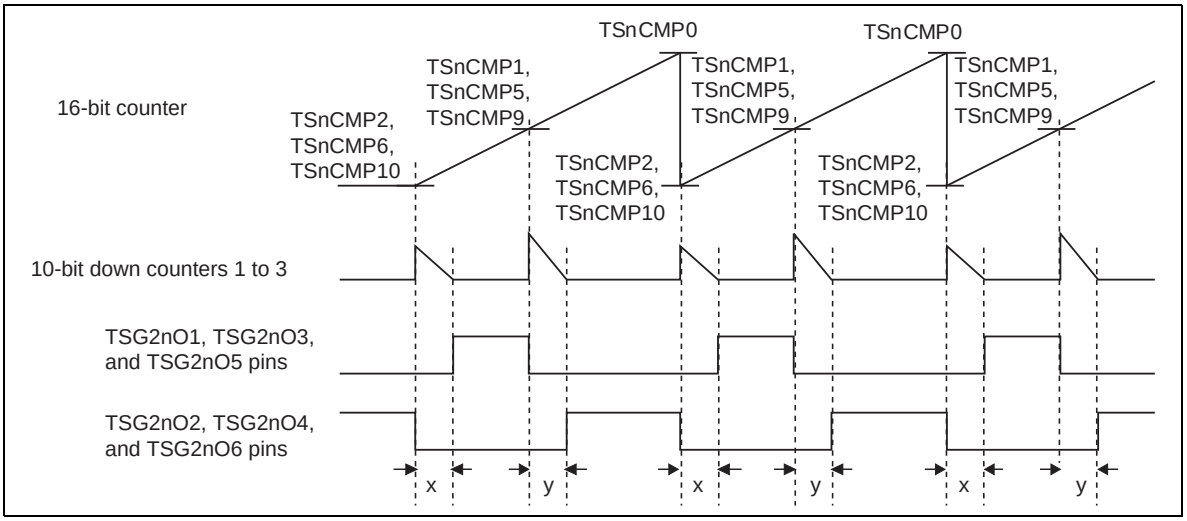


Figure 15-60 Example of Output Waveform in SP-PWM Mode

Note x: TSnDTC0 register values
y: TSnDTC1 register values

(5) Software Output Control Function in SP-PWM Mode

TSnOPT0.TSnSOC, TSnIDC, and TSnOPT1.TSnSPC2 to TSnSPC0 are used to control timer output by software.

As shown in Figure 15-61, the output control is switched immediately when TSnSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSnSOC is set to 0, output control is retained. When the reload timing is generated, output control is switched to SP-PWM mode output control.

For details, refer to section 15.11.5, Software Output Control Function.

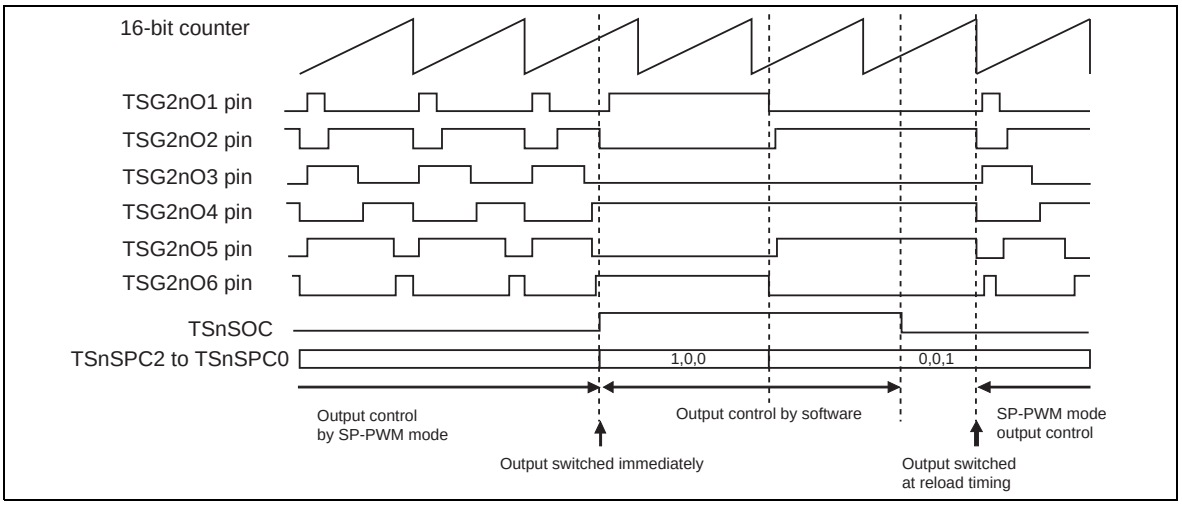


Figure 15-61 Example of Output Control Switching from SP-PWM Mode Control to Software Control

(a) Procedure on Software Output Control Processing

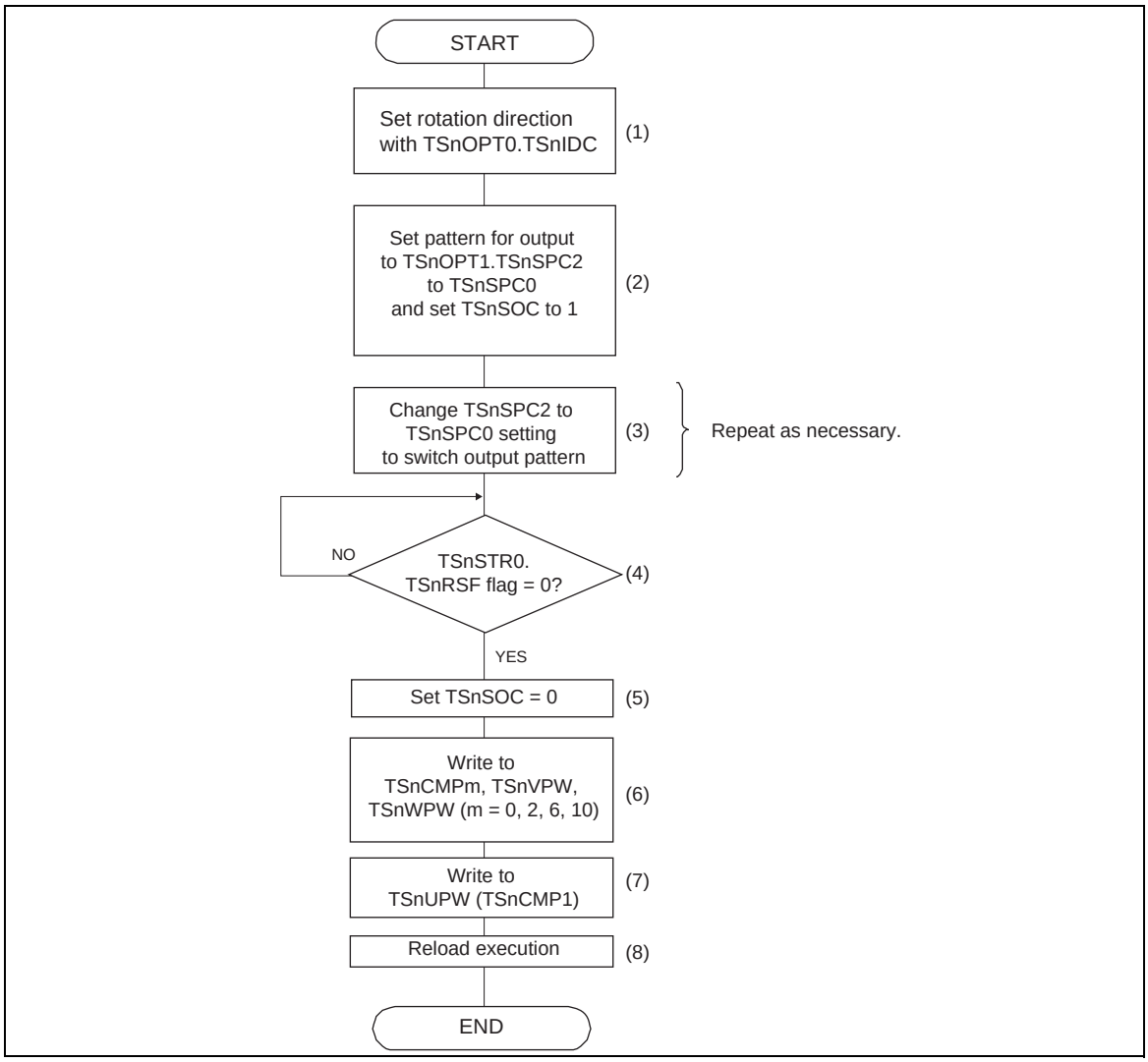


Figure 15-62 Flow of Software Output Control

The procedure for software output control is described below.

(1) Set TSnOPT0 to determine the rotation direction. The timer output has a 180-degree phase shift between when TSnIDC = 0 and when TSnIDC = 1. With the software output control function, the timer output will not change only by rewriting the TSnOPT0 bit.

(2) Set the pattern for output to TSnOPT1. At the same time, set TSnOPT0.TSnSOC to 1 to switch to software output control mode.

(3) Change the output pattern setting for TSnSPC2-TSnSPC0 to change the timer output. During software control mode, the following registers can be modified:
TSnTRG0.TSnTS, TSnCTL3-TSnCTL6, TSnOPT0, TSnOPT1, TSnCMP0-TSnCMP12, TSnDTC0, and TSnDTC1.

(4) Ensure that the reload request flag TSnSTR0.TSnRSF is 0. If TSnRSF is 1, do not shift to the following procedure until it goes 0.

(5) By clearing TSnSOC to 0, software control starts to be released (not yet released here).

(6) After software output control is released, set the compare registers if necessary. Move to the following procedure if no setting is required. In addition, change the registers with the reload function if necessary.

(7) Write TSnUPW (TSnCMP1) to start reloading.

(8) Reload is executed and software output control is released.

Caution Be sure to execute reload after execution of steps (4), (5), (6), and (7). Unless reload can be executed, software output control cannot be released

15.11.4 120-DC Mode

Overview In this mode, PWM output period set to TSnCMP0 and timer output (TSG2nO1 to TSG2nO6) according to the duty cycle set to TSnCMP1 to TSnCMP12 are controlled with three types of pattern inputs (software output control method, pattern switch method, and trigger switch method) to perform 120-DC control.

Prerequisites

- Set the PWM period to TSnCMP0.
- Set the PWM duty to TSnCMP1 to TSnCMP12 and set the output pattern to TSnPAT0W and TSnPAT1W.

Functional description Set the PWM period, set the duty cycle to individual compare register, and set the pattern to be output to the pattern register. Setting TSnTRG0.TSnTS to 1 starts counting.

The 16-bit counter starts counting from 0000_H, and is cleared by the match with TSnCMP0.

INTTSG2nI01 to INTTSG2nI12 interrupts are generated by a compare match of the 16-bit counter and TSnCMP1 to TSnCMP12 buffer registers, respectively.

Note 120-DC mode is valid when TSnCTL0.TSnMD1 to TSnMD0 are set to 11_B.

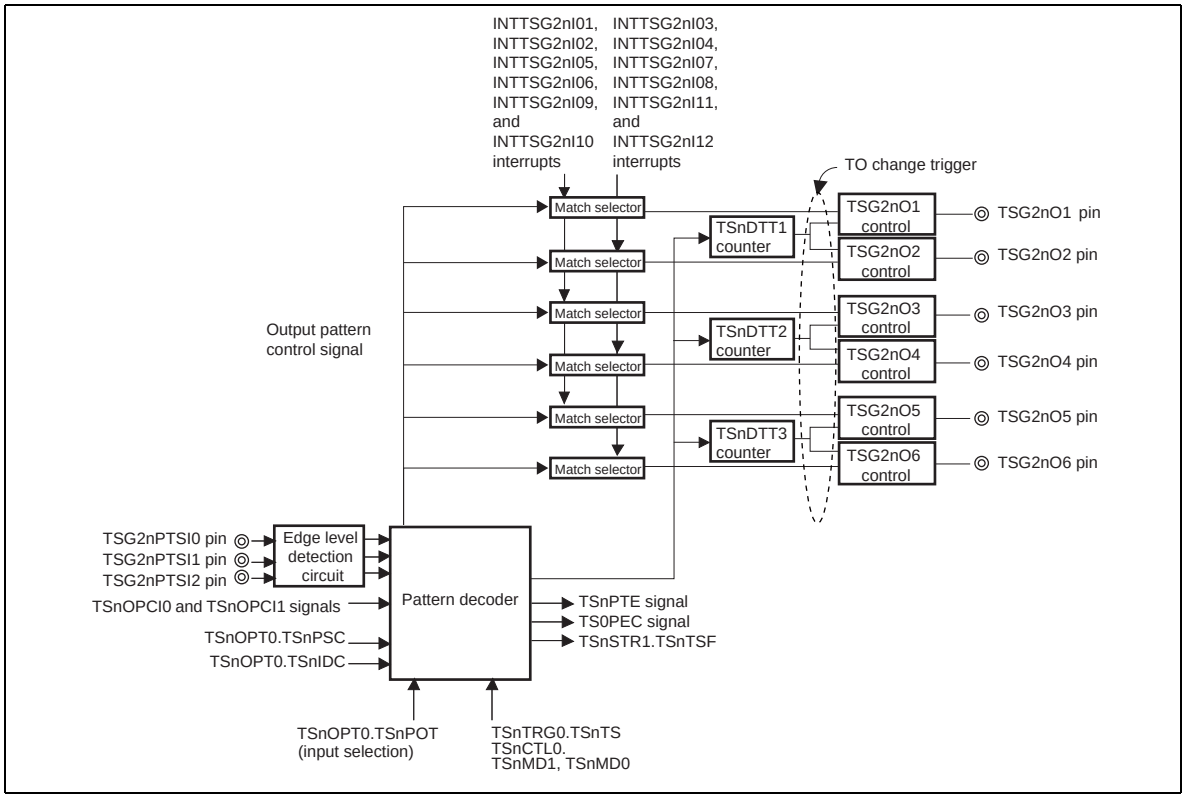


Figure 15-63 Block Diagram in 120-DC Mode

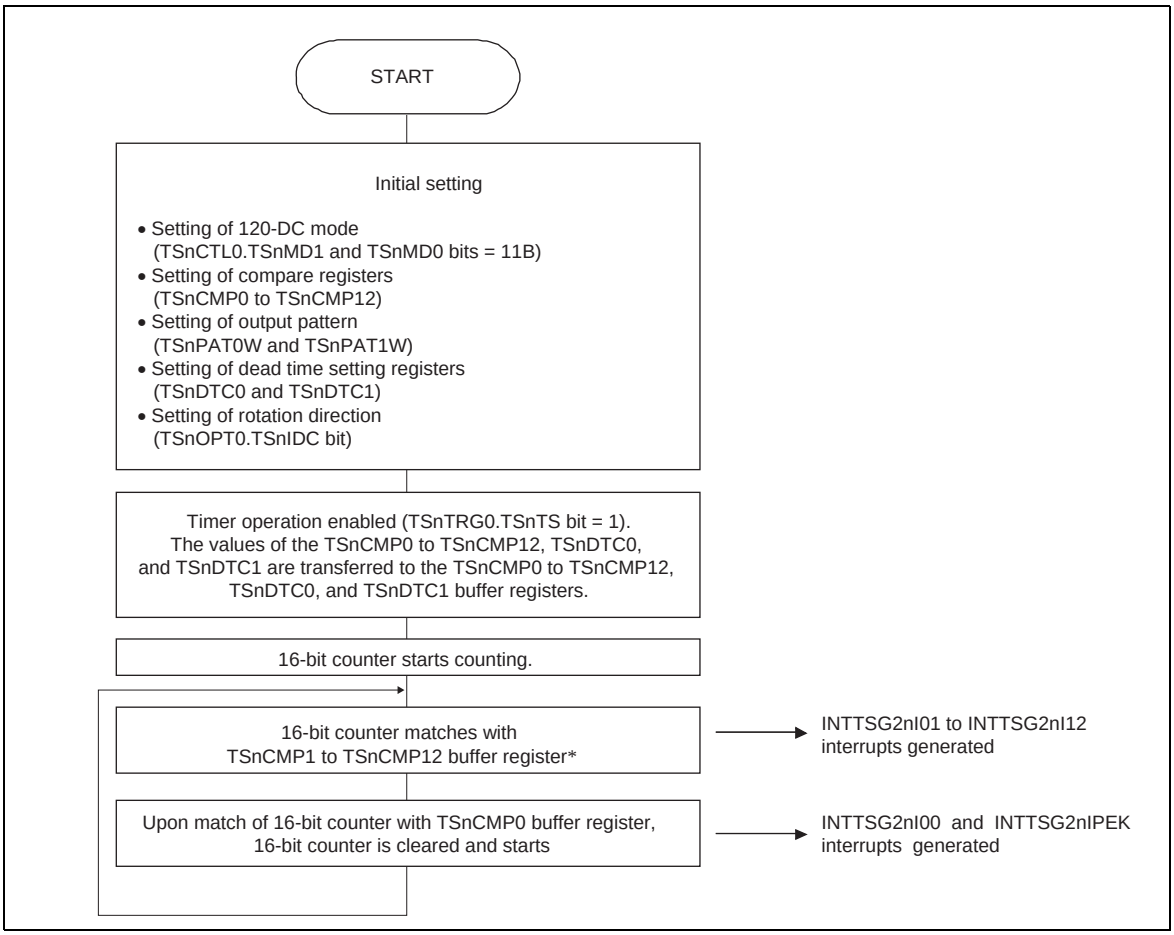


Figure 15-64 Basic Operation Flow in 120-DC Mode

Note * The 16-bit counter is not cleared by match of the 16-bit counter with the TSnCMP1 to TSnCMP12 buffer registers.

(1) List of Operations in 120-DC Mode

Table 15-71 Counter Functions in 120-DC Mode

Operation		Setting Condition
16-bit counter	Start	TSnTRG0.TSnTS = 0 → 1
	Clear	Match of TSnCMP0 value and 16-bit counter value, or output pattern switch timing
	Stop	TSnTRG1.TSnTT = 0 → 1

Table 15-72 Functions of Compare Registers and Dead Time Setting Registers in 120-DC Mode

Register	Rewrite Method	Rewrite during Operation	Function
TSnCMP0	Reload	Possible	Setting period
TSnCMPm (m = 1 to 12)	Reload	Possible	Setting PWM duty
TSnDCMP0W, TSnDCMP2	Reload	Possible	Outputting diagnostic signal or setting A/D conversion trigger
TSnDTC0, TSnDTC1	Reload	Possible	Setting dead time

Table 15-73 Timer Input Function in 120-DC Mode

Pin/Signal	Function
TSG2nCLKI	Clock enable input
TSG2nPTSI2 to TSG2nPTSI0 pins	Pattern input (3 phases)
TSnOPCI0, TSnOPCI1 signals	Trigger input

Table 15-74 Timer Output Function in 120-DC Mode

Pin/Signal	Function
TSG2nO1 pin	PWM output (with dead time) by compare match of the TSnCMPm buffer register (m = 1, 2, 5, 6, 9, 10) with the 16-bit counter and by selecting output pattern through TSnPAT0W setting.
TSG2nO2 pin	PWM output (with dead time) by compare match of the TSnCMPm buffer register (m = 3, 4, 7, 8, 11, 12) and the 16-bit counter and by selecting output pattern through TSnPAT1W setting.
TSG2nO3 pin	PWM output (with dead time) by compare match of the TSnCMPm buffer register (m = 1, 2, 5, 6, 9, 10) and the 16-bit counter and by selecting output pattern through TSnPAT0W setting.
TSG2nO4 pin	PWM output (with dead time) by compare match of the TSnCMPm buffer register (m = 3, 4, 7, 8, 11, 12) and the 16-bit counter and by selecting output pattern through TSnPAT1W setting.
TSG2nO5 pin	PWM output (with dead time) by compare match of the TSnCMPm buffer register (m = 1, 2, 5, 6, 9, 10) and the 16-bit counter and by selecting output pattern through TSnPAT0W setting.
TSG2nO6 pin	PWM output (with dead time) by a compare match of the TSnCMPm buffer register (m = 3, 4, 7, 8, 11, 12) and the 16-bit counter and by selecting output pattern through TSnPAT1W setting.
TSG2nO7 pin	Diagnostic signal output or pulse output by A/D conversion trigger
TSnPTE signal	Toggle signal by change in input pattern

Table 15-75 Interrupt Requests in 120-DC Mode

Interrupt	Function
INTTSG2nIm (m = 00 to 12)	Compare match of TSnCMPm buffer register and 16-bit counter (m = 0 to 12)
INTTSG2nIER	Error
INTTSG2nIVLY	-
INTTSG2nIPEK	Peak interrupt (generated at the same timing as INTTSG2nI00)
INTTSG2nIWN	Warning interrupt

Table 15-76 Compare Match Timing in 120-DC Mode

Compare Match	Timing
TSnCMP0	When 16-bit counter changes from TSnCMP0 to 0000_H
TSnCMPm (m = 1 to 12)	After detecting the match of 16-bit counter and TSnCMPm (m = 1 to 12)

Table 15-77 Example of Setting Each Timer Output Condition in 120-DC Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG2nOm (m = 1 to 6)	PWM output	(TSnCMP0 + 1) × count clock	See section 15.11.4 6 List of Output Patterns in 120-DC Mode.	-
TSG2nO7	Diagnostic signal output or pulse output by A/D conversion trigger	(TSnCMP0 + 1) × count clock	See section 15.9, A/D Conversion Trigger Function.	-

(2) Various Settings of 120-DC Mode

Mode setting

120-DC mode can be used by setting TSnCTL0.TSnMD1 and TSnMD0 are set to 11_B.

Setting timer output

The output pins TSG2nO1 to TSG2nO6 are controlled by setting TSnIOC0, TSnIOC2, and TSnIOC3.

The TSG2nO7 pin outputs pulses of the diagnostic output or the A/D conversion trigger. Please set it as required.

Enabling error interrupt generation

With TSnIOC1.TSnEOC = 1, the error interrupt (INTTSG2nIER) generation is enabled when the simultaneous active state of the positive phase and inverse phase is detected. For details, see section 15.10, Error/Warning Interrupt.

Setting register rewrite timing

Reloading the registers with the reload function is activated with TSnCTL3.TSnRMC (simultaneous rewrite; default setting is 0 = reload). Set TSnCTL4.TSnPRE to 1 when reload is used.

The reload timing is not generated if TSnPRE is 0.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSnADTRG0 signal), use TSnCTL5.TSnAT09 to TSnAT00.

With TSnAT09 to TSnAT00, A/D conversion trigger output is enabled or disabled at the match of the 16-bit counter and TSnDCMP2 to TSnDCMP0 (during up count).

To set A/D conversion trigger 1 (TSnADTRG1 signal), use TSnCTL6.TSnAT19 to TSnAT10.

To set the match timing of the 16-bit counter and TSnDCMP2 to TSnDCMP0, set the compare value to the pertinent register.

The skipping function can be used for TSnADTRG0 and TSnADTRG1 signals. Use TSnACC01, TSnACC00 of TSnCTL5 and TSnACC11, and TSnACC10 in TSnCTL6 to select the skipping rate among 1/1, 1/2, 1/4, and 1/8.

Caution

- Set TSnCTL5, TSnCTL6, and TSnDCMP2 to TSnDCMP0 correctly when using the TSG2nO7 output for the A/D conversion trigger timing pulse.
- In 120-DC mode, a valley interrupt (INTTSG2nIVLY) is not generated. Therefore, set TSnAT00 and TSnAT10 in TSnCTL5 and TSnCTL6 to 0.
- In 120-DC mode, the 16-bit sub-counter does not operate. Therefore, set TSnAT09, TSnAT08, TSnAT19, and TSnAT18 in TSnCTL5 and TSnCTL6 to 0.
- In 120-DC mode, the 16-bit counter does not decrement. Therefore, set TSnAT07, TSnAT05, TSnAT03, TSnAT17, TSnAT15, and TSnAT13 in TSnCTL5 and TSnCTL6 to 0.

Setting dead time

The dead time can be set with TSnDTC0 and TSnDTC1.

The dead time is calculated by the following expressions:

PCLK × TSnDTC0

PCLK × TSnDTC1

TSnDTC0 can set the time between a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the inactive state and a change of TSG2nO1, TSG2nO3, and TSG2nO5 to the active state, respectively.

TSnDTC0 can set the time between a change of TSG2nO1, TSG2nO3, and TSG2nO5 to the inactive state to a change of TSG2nO2, TSG2nO4, and TSG2nO6 to the active state.

Carrier period

Set the carrier period with TSnCMP0 according to the following expression:

TSnCMP0 = (carrier period/count clock period) - 1

Duty (PWM width) setting

The duty of PWM output is set with TSnCMP1 to TSnCMP12. The setting range of the compare registers is as follows:

$0000_H \leq$ TSnCMPm $\leq$ TSnCMP0 + 1

Caution Do not set TSnCMPm to TSnCMP0 + 1 (m = 1 to 12) only when TSnCMP0 + 1 < TSnCMPm and TSnCMP0 = $FFFF_H$.

Output PWM setting

In 120-DC mode, the output pins TSG2nO1, TSG2nO3, and TSG2nO5 are controlled by TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and TSnCMP10, and the output pins TSG2nO2, TSG2nO4, TSG2nO6 are controlled by TSnCMP3, TSnCMP4, TSnCMP7, TSnCMP8, TSnCMP11, and TSnCMP12. The duty cycle of a PWM period (TSnCMP0) can be set with TSnCMP1 to TSnCMP12. Setting TSnCMP1 to TSnCMP12 to 0000_H sets the PWM duty cycle to 0%. Setting TSnCMP1 to TSnCMP12 to TSnCMP0 + 1 value sets the PWM duty cycle to 100%. This allows chopping output control and rectangular wave output control.

(3) Control Methods in 120-DC Mode

Control methods in 120-DC mode are listed below.

Control Method	Function
Software output control method	Switches the output pattern according to the TSnOPT1.TSnSPC2 to TSnSPC0 setting made by software.
Pattern switch method	Directly switches the output pattern by the pattern input signal of TSG2nPTSI0 to TSG2nPTSI2.
Trigger switch method	Switches the output pattern by the trigger switch method using the trigger input signals TSnOPCI0 and TSnOPCI1 or by the pattern input setting of TSnOPT1.TSnSPC2 to TSnSPC0 in the constant order.

Setting software output control method

Setting TSnOPT0.TSnSTE = 0 switches the output pattern by software output control. The TSG2nO1 to TSG2nO6 pin output is switched according to TSnOPT1.TSnSPC2 to TSnSPC0.

The output order at the beginning of operation is set with TSnOPT0.TSnIDC. The output pattern is set with TSnOPT0.TSnPSC.

Operation of software output control method

The PWM output of TSG2nO1 to TSG2nO6 pins (PWM output defined by TSnCMP1 to TSnCMP12) is selected by TSnOPT1.TSnSPC2 to TSnSPC0 by software. To control the dead time, the dead time counter is activated at the falling edge of the signals in each phase and the dead time is inserted.

The 16-bit counter counts based on the carrier period set in TSnCMP0. The 16-bit counter is cleared by match of the 16-bit counter and TSnCMP0 or by a write access to TSnOPT1.TSnSPC2 to TSnSPC0.

In this method, the pattern is output, which is decoded using information on the output pattern (TSnSPC2 to TSnSPC0), the electric current direction control bit (TSnOPT0.TSnIDC), and TSG2nPTSI2 to TSG2nPTSI0 pattern order detection flag (TSnSTR1.TSnTSF). Figure 15-85 shows the timer output when the output pattern is changed by software output control.

Immediately after the operation starts (TSnTRG0.TSnTS = 1), the output pattern of TSnSPC2 to TSnSPC0 and the pattern set with TSnIDC and TSnPSC (TSnOPT0.TSnPSS = 1) are output.

Setting pattern switch method

Setting TSnOPT0.TSnSTE to 1 and TSnPOT to 0 selects the pattern switch method. The TSG2nO1 to TSG2nO6 pin output is changed at the change timing of the TSG2nPTSI2 to TSG2nPTSI0 pins.

The output order at the beginning of operation is set with TSnOPT0.TSnIDC. The initial output pattern is set with TSnOPT0.TSnPSC. However, after determining the rotation direction (after the value is set to TSnSTR1.TSnTSF), the setting of TSnPSC is disabled.

Operation of pattern switch method

After level detection is performed for the pins (three inputs from the hall sensor), the level-detected signals are decoded. From the decoding result, the PWM output of TSG2nO1 to TSG2nO6 pins (PWM output defined by TSnCMP1 to TSnCMP12) is selected. To control the dead time, the dead time counter is activated at the falling timing of signals in each phase and the dead time is inserted.

The 16-bit counter counts based on the carrier period set in TSnCMP0. The 16-bit counter is cleared by match of the 16-bit counter and TSnCMP0 or by a change of the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins).

In this method, the pattern, which is decoded by using information on input pattern (TSG2nPTSI2 to TSG2nPTSI0), the electric current direction control bit (TSnOPT0.TSnIDC), and TSG2nPTSI2 to TSG2nPTSI0 pattern order detection flag (TSnSTR1.TSnTSF), is output. Figure 15-67 to Figure 15-70 show the timer output when TSG2nPTSI2 to TSG2nPTSI0 pin inputs change. If input pattern 1 is switched to input pattern 4 due to an abnormal input pattern, the output pattern is switched to the pattern corresponding to the input pattern.

Immediately after the operation starts (TSnTRG0.TSnTS = 1), the output pattern set with the input level of TSG2nPTSI2 to TSG2nPTSI0 pins, TSnIDC, and TSnPSC (TSnOPT0.TSnPSS = 1) is output. After the TSnTSF value is determined, the output pattern is determined by TSnTSF instead of TSnPSC.

Caution When connecting the three-phase pulse input signal to the TSG2nPTSI2 to TSG2nPTSI0 pins, confirm that the three-phase pulse input value and the patterns output from the TSG2nO1 to TSG2nO6 pins satisfy the expected conditions.
If the expected conditions are not satisfied, change the connection between the three-phase pulse input signal and the TSG2nPTSI2 to TSG2nPTSI0 pins.

Setting trigger switch method

Setting TSnOPT0.TSnSTE and TSnPOT to 1 selects the trigger switch method. The output pins TSG2nO1 to TSG2nO6 are changed at a rising edge of an external input (TSnOPCI1 and TSnOPCI0 signals).

For pattern output order, see section 15.11.4 5 Operation in 120-DC Mode.

The initial output pattern can be controlled with TSnOPT1.TSnSPC2 to TSnSPC0.

When starting the TSG2n operation (TSnTRG0.TSnTS = 1) after setting TSnSPC2 to TSnSPC0, the initial pattern is output. For details, see section 15.11.4 6 List of Output Patterns in 120-DC Mode.

Operation of trigger switch method

With the trigger input switch method, the rising edges of the TSnOPCI0 and TSnOPCI1 signals are detected and the output switch timing is generated. The initial timer output pattern is set with TSnOPT1.TSnSPC2 to TSnSPC0. The subsequent output patterns are determined with TSnSPC2 to TSnSPC0, TSnOPT0.TSnIDC, and TSnPSC.

The 16-bit counter counts based on the carrier period set in TSnCMP0. The 16-bit counter is cleared by match of the 16-bit counter and TSnCMP0, by a write access to TSnOPT1.TSnSPC2 to TSnSPC0, or by detecting a rising of TSnOPCI0 and TSnOPCI1 signals.

For examples of operation in 120-DC mode when trigger input switch method is used, see Figure 15-67 to Figure 15-70.

Caution The initial pattern should be set according to the read input level of the port to which TSG2nPTSI2 to TSG2nPTSI0 pins are connected.

(4) Timer Output in 120-DC Mode

In 120-DC mode, the PWM output is controlled with TSnPAT0W, TSnPAT1W, and TSnCMP1 to TSnCMP12. TSnPAT0W, TSnCMP1, TSnCMP2, TSnCMP5, TSnCMP6, TSnCMP9, and TSnCMP10 are set to control the output of TSG2nO1, TSG2nO3, and TSG2nO5 pins. TSnPAT1W, TSnCMP3, TSnCMP4, TSnCMP7, TSnCMP8, TSnCMP11, and TSnCMP12 are set with the output of SG2nO2, TSG2nO4, and TSG2nO6 pins.

With PWM output control, eight types of output patterns can be selected for each of TSG2nO1, TSG2nO3, and TSG2nO5 pins and TSG2nO2, TSG2nO4, and TSG2nO6 pins.

Table 15-78 TSnPAT0W Set Value and Output Control

PATmT Value	Output Control
000	Fixed to low
001	PWM output set with TSnCMP1
010	PWM output set with TSnCMP2
011	PWM output set with TSnCMP5
100	PWM output set with TSnCMP6
101	PWM output set with TSnCMP9
110	PWM output set with TSnCMP10
111	Fixed to high

(m = 0, 1, 2, 3, 4, 5)

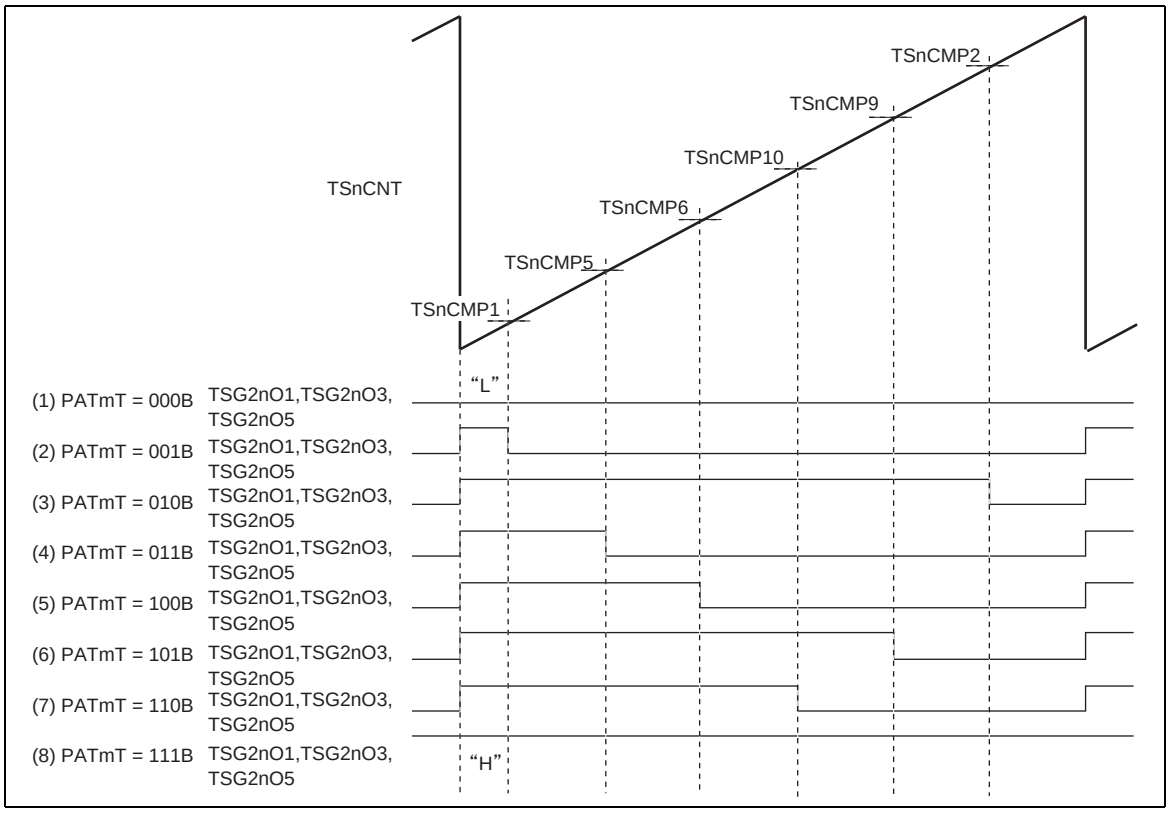


Figure 15-65 TSG2nO1, TSG2nO3, TSG2nO5 Pin Output of Each Output Pattern

Table 15-79 TSnPAT1W Set Value and Output Control

PATmB Value	Output Control
000	Fixed to low
001	PWM output set with TSnCMP3
010	PWM output set with TSnCMP4
011	PWM output set with TSnCMP7
100	PWM output set with TSnCMP8
101	PWM output set with TSnCMP11
110	PWM output set with TSnCMP12
111	Fixed to high

(m = 0, 1, 2, 3, 4, 5)

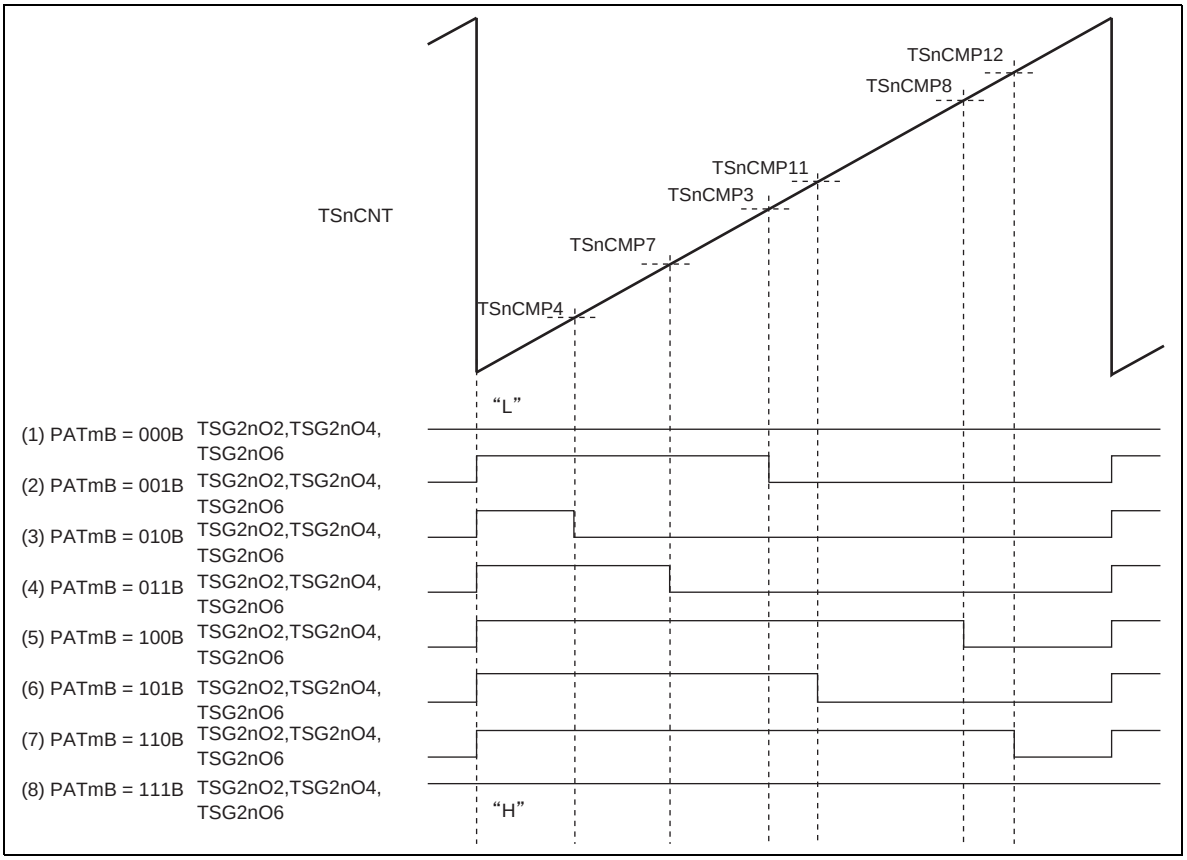


Figure 15-66 TSG2nO2, TSG2nO4, TSG2nO6 Pin Output of Each Output Pattern

(5) Operation in 120-DC Mode

Figure 15-67 to Figure 15-70 show examples of operation in 120-DC mode.

The TSG2nO1 to TSG2nO6 pins detect the input level change of the TSG2nPTSI2 to TSG2nPTSI0 pins, then change the output pattern. The 16-bit counter produces sawtooth waveform, and TSnCMP0 to TSnCMP12 output PWM signal. The 16-bit counter is cleared to 0000_H each time the counter value matches with TSnCMP0 or a change in the TSG2nPTSI2 to TSG2nPTSI0 pins is detected. The timer output pattern is switched each time a change in the TSG2nPTSI2 to TSG2nPTSI0 pins is detected.

Note PAT0T to PAT5T and PAT0B to PAT5B show PWM operation set by TSnCMP1 to TSnCMP12, respectively.

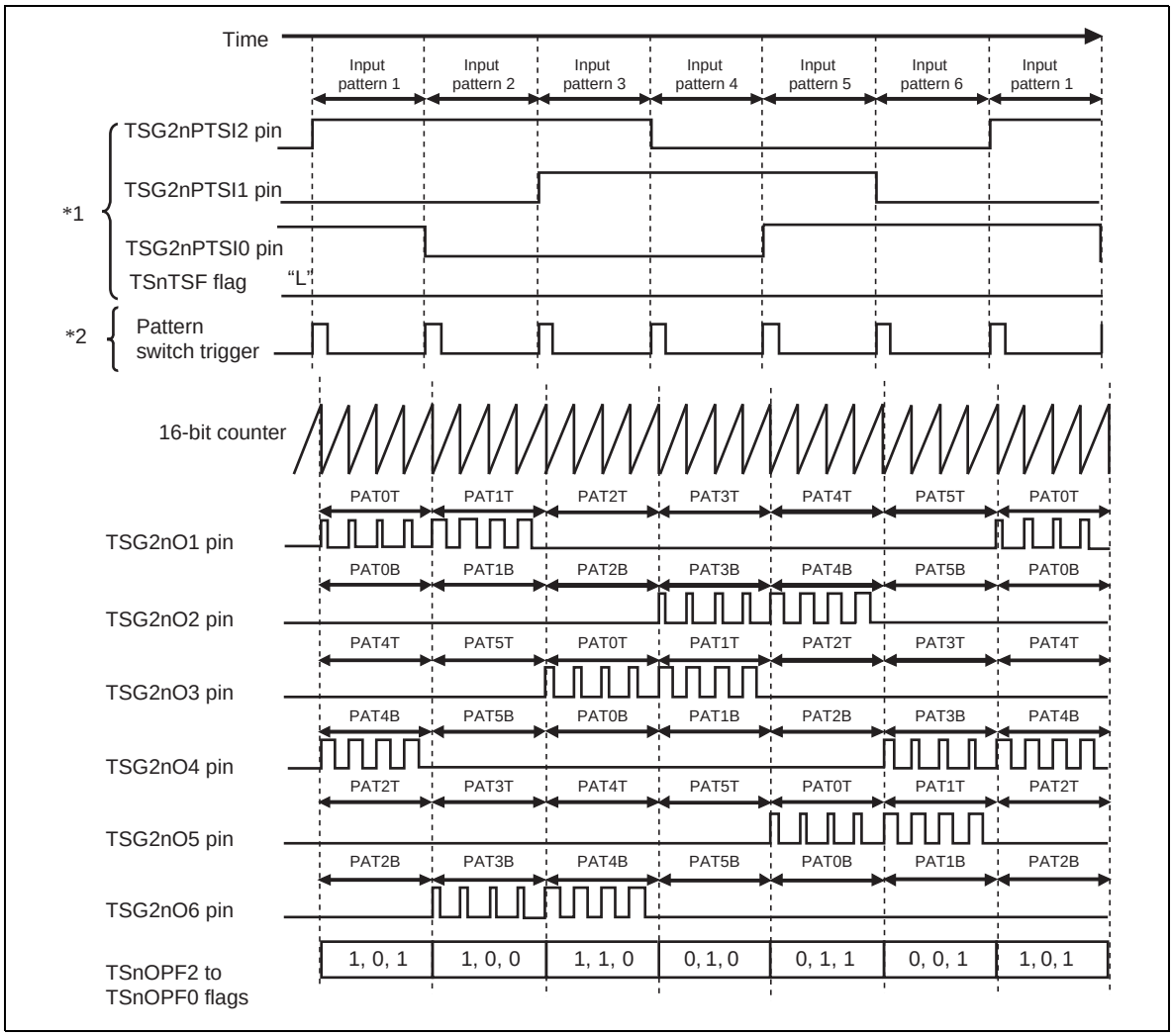


Figure 15-67 Example of Operation in 120-DC Mode (Normal Rotation: TSnSTR1.TSnTSF = 0 and TSnOPT0.TSnIDC = 0)

Note 1. For pattern switch method (TSnOPT0.TSnPOT = 0)
Note 2. For trigger switch method (TSnOPT0.TSnPOT = 1 and TSnPSS = 1)

Note TSnOPT0.TSnSOC = 0

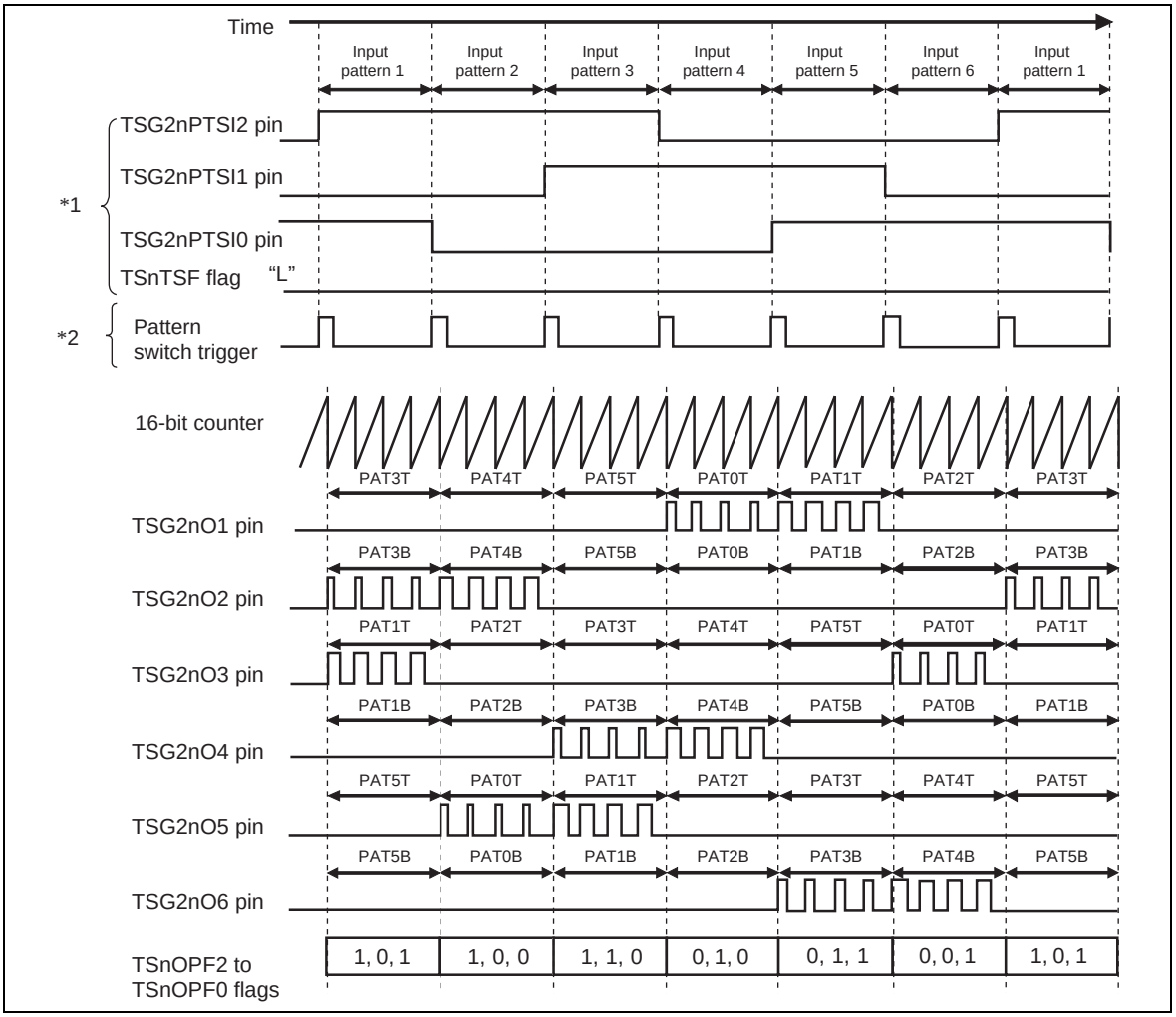


Figure 15-68 Example of Operation in 120-DC Mode (Normal Rotation: TSnSTR1.TSnTSF = 0 and TSnOPT0.TSnIDC = 1)

Note 1. For pattern switch method (TSnOPT0.TSnPOT = 0)
Note 2. For trigger switch method (TSnOPT0.TSnPOT = 1 and TSnPSS = 1)

Note TSnOPT0.TSnSOC = 0

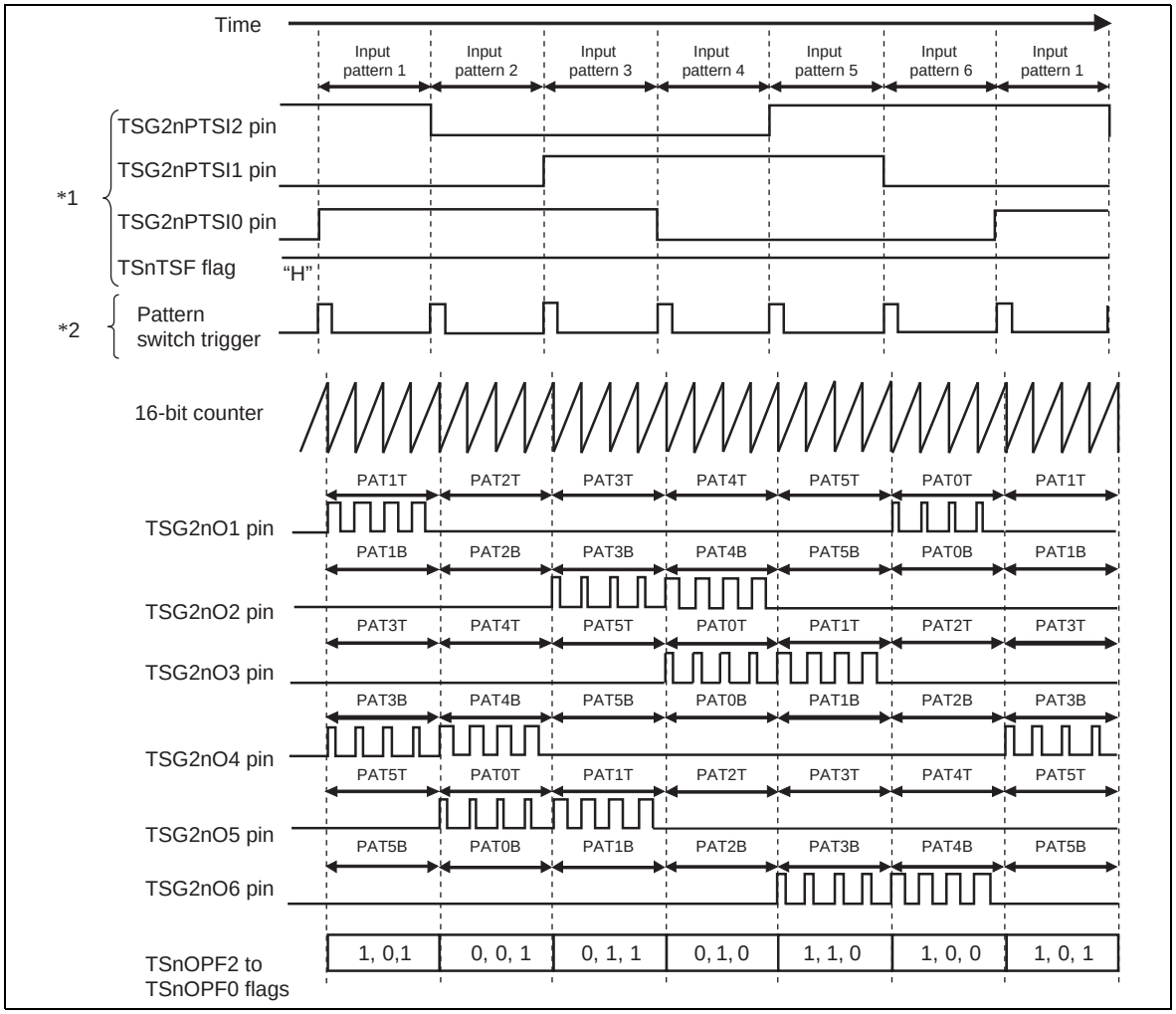


Figure 15-69 Example of Operation in 120-DC Mode (Reverse Rotation: TSnSTR1.TSnTSF = 1 and TSnOPT0.TSnIDC = 0)

Note 1. For pattern switch method (TSnOPT0.TSnPOT = 0)
Note 2. For trigger switch method (TSnOPT0.TSnPOT = 1 and TSnPSS = 1)

Note TSnOPT0.TSnSOC = 0

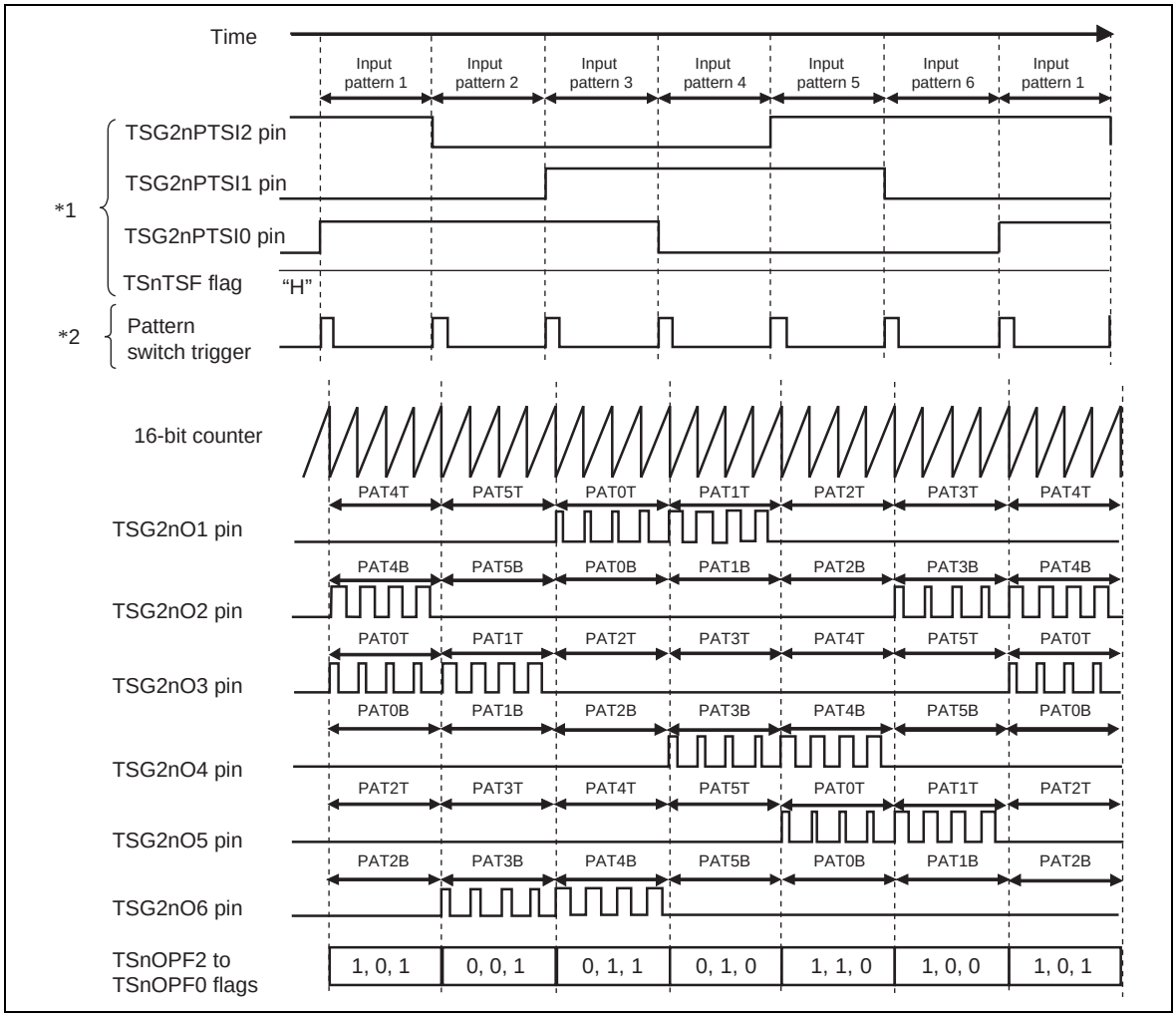


Figure 15-70 Example of Operation in 120-DC Mode (Reverse Rotation: TSnSTR1.TSnTSF = 1 and TSnOPT0.TSnIDC = 1)

Note 1. For pattern switch method (TSnOPT0.TSnPOT = 0)
Note 2. For trigger switch method (TSnOPT0.TSnPOT = 1 and TSnPSS = 1)

Note TSnOPT0.TSnSOC = 0

(6) List of Output Patterns in 120-DC Mode

In 120-DC mode, the output pattern is determined according to the rotation direction and TSnOPT0.TSnIDC.

TSnOPT0		
TSnPOT	**TSnPSS**	**Rotation Direction**
0	-	TSnTSF
1	1	TSnPSC

Normal rotation: TSnOPT0.TSnSOC = 0, TSnPSC = 0,TSnPOT = 1, TSnPSS = 1, TSnIDC = 0

The order of pattern switching

	TSnOPT1.TSnSPC2 to TSnSPC0*/TSnSTR1.TSnOPF2 to TSnOPF0							
Output Pin	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	PAT0T	PAT1T	PAT2T	PAT3T	PAT4T	PAT5T	Low	Low
TSG2nO2	PAT0B	PAT1B	PAT2B	PAT3B	PAT4B	PAT5B	Low	Low
TSG2nO3	PAT4T	PAT5T	PAT0T	PAT1T	PAT2T	PAT3T	Low	Low
TSG2nO4	PAT4B	PAT5B	PAT0B	PAT1B	PAT2B	PAT3B	Low	Low
TSG2nO5	PAT2T	PAT3T	PAT4T	PAT5T	PAT0T	PAT1T	Low	Low
TSG2nO6	PAT2B	PAT3B	PAT4B	PAT5B	PAT0B	PAT1B	Low	Low

Normal rotation: TSnOPT0.TSnSOC = 0, TSnPSC = 0, TSnPOT = 1, TSnPSS = 1, TSnIDC = 1

The order of pattern switching

	TSnOPT1.TSnSPC2 to TSnSPC0*/TSnSTR1.TSnOPF2 to TSnOPF0							
Output Pin	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	PAT3T	PAT4T	PAT5T	PAT0T	PAT1T	PAT2T	Low	Low
TSG2nO2	PAT3B	PAT4B	PAT5B	PAT0B	PAT1B	PAT2B	Low	Low
TSG2nO3	PAT1T	PAT2T	PAT3T	PAT4T	PAT5T	PAT0T	Low	Low
TSG2nO4	PAT1B	PAT2B	PAT3B	PAT4B	PAT5B	PAT0B	Low	Low
TSG2nO5	PAT5T	PAT0T	PAT1T	PAT2T	PAT3T	PAT4T	Low	Low
TSG2nO6	PAT5B	PAT0B	PAT1B	PAT2B	PAT3B	PAT4B	Low	Low

Note * When TSnSPC2 to TSnSPC0 are written to while TSnPOT = 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated on rising of the TSnOPCI0 and TSnOPCI1 signals, the output is switched according to the order of pattern switching. In this case, TSnSPC2 to TSnSPC0 remain unchanged even if the output pattern is switched.

Note 1. PAT0T to PAT5T: PWM output set by TSnCMP1W, TSnCMP5W, and TSnCMP9W

Note 2. PAT0B to PAT5B: PWM output set by TSnCMP3W, TSnCMP7W, and TSnCMP11W

Reverse rotation: TSnOPT0.TSnSOC = 0, TSnPSC = 1, TSnPOT = 1, TSnPSS = 1, TSnIDC = 0

The order of pattern switching

Output Pin	TSnOPT1.TSnSPC2 to TSnSPC0*/TSnSTR1.TSnOPF2 to TSnOPF0							
	101	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	PAT1T	PAT0T	PAT5T	PAT4T	PAT3T	PAT2T	Low	Low
TSG2nO2	PAT1B	PAT0B	PAT5B	PAT4B	PAT3B	PAT2B	Low	Low
TSG2nO3	PAT3T	PAT2T	PAT1T	PAT0T	PAT5T	PAT4T	Low	Low
TSG2nO4	PAT3B	PAT2B	PAT1B	PAT0B	PAT5B	PAT4B	Low	Low
TSG2nO5	PAT5T	PAT4T	PAT3T	PAT2T	PAT1T	PAT0T	Low	Low
TSG2nO6	PAT5B	PAT4B	PAT3B	PAT2B	PAT1B	PAT0B	Low	Low

Reverse rotation: TSnOPT0.TSnSOC = 0, TSnPSC = 1, TSnPOT = 1, TSnPSS = 1, TSnIDC = 1

The order of pattern switching

Output Pin	TSnOPT1.TSnSPC2 to TSnSPC0*/TSnSTR1.TSnOPF2 to TSnOPF0							
	101	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	PAT4T	PAT3T	PAT2T	PAT1T	PAT0T	PAT5T	Low	Low
TSG2nO2	PAT4B	PAT3B	PAT2B	PAT1B	PAT0B	PAT5B	Low	Low
TSG2nO3	PAT0T	PAT5T	PAT4T	PAT3T	PAT2T	PAT1T	Low	Low
TSG2nO4	PAT0B	PAT5B	PAT4B	PAT3B	PAT2B	PAT1B	Low	Low
TSG2nO5	PAT2T	PAT1T	PAT0T	PAT5T	PAT4T	PAT3T	Low	Low
TSG2nO6	PAT2B	PAT1B	PAT0B	PAT5B	PAT4B	PAT3B	Low	Low

Note * When TSnSPC2 to TSnSPC0 are written to while TSnPOT = 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated by rising of TSnOPCI0 and TSnOPCI1 signals, the output is switched according to the pattern switch order. In this case, TSnSPC2 to TSnSPC0 remain unchanged even if the output pattern is switched.

Note 1. PAT0T to PAT5T: PWM output set by TSnCMP1W, TSnCMP5W, and TSnCMP9W

Note 2. PAT0B to PAT5B: PWM output set by TSnCMP3W, TSnCMP7W, and TSnCMP11W

(7) Operation Start Timing in 120-DC Mode

In 120-DC mode, when trigger switch control (TSnOPT0.TSnPOT = 1) is used, pattern set with TSnOPT1.TSnSPC2 to TSnSPC0, rotation direction, and TSnOPT0.TSnIDC can be output. However, when pattern switch control (TSnPOT = 0) is used, the pattern of the TSG2nPTSI2 to TSG2nPTSI0 pins can be detected but rotation direction (TSnSTR1.TSnTSF) cannot be determined. Therefore, set the rotation direction in TSnPSC when TSnTE is 0. The TSnPSC set value is loaded to TSnTSF, and the value can be used for the initial pattern setting.

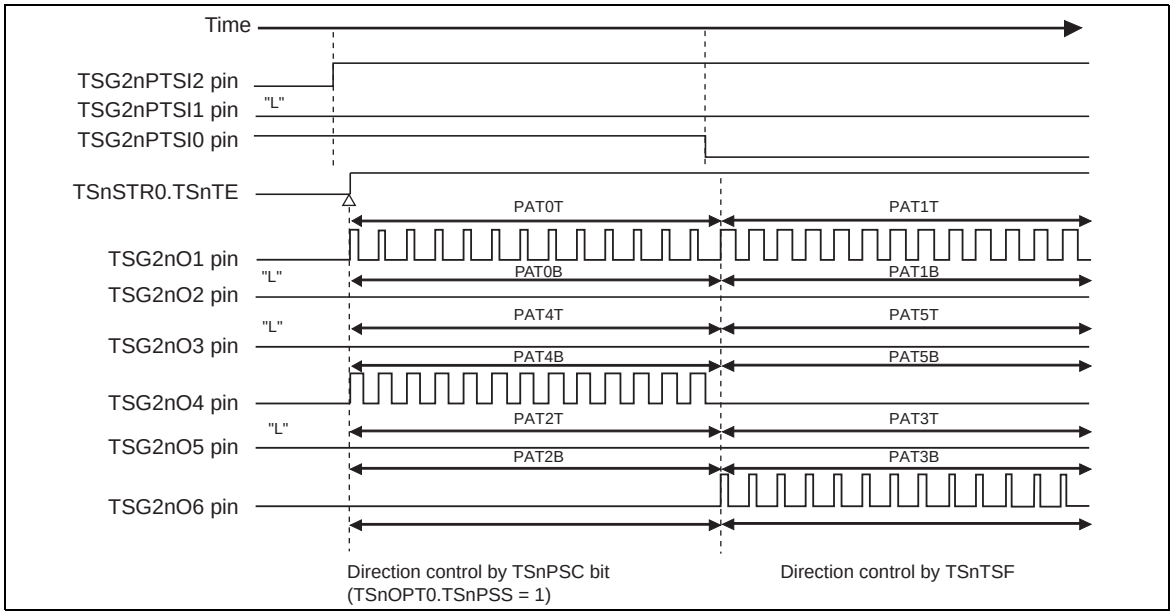


Figure 15-71 Control when Timer Output Starts in Normal Rotation (when Normal Pattern is Input)

TSnOPT0.TSnSOC = 0, TSnPSC = 0, TSnPOT = 0, TSnIDC = 0

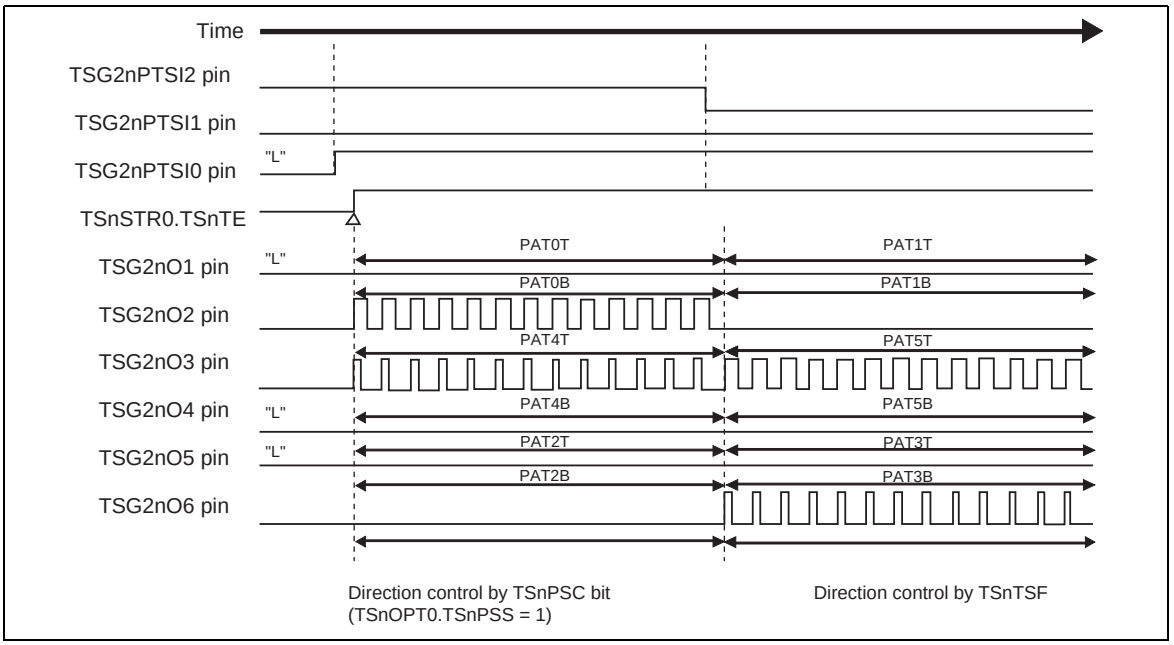


Figure 15-72 Control when Timer Output Starts in Reverse Rotation (when Normal Pattern is Input)

TSnOPT0.TSnSOC = 0, TSnPSC = 1, TSnPOT = 0, TSnIDC = 1

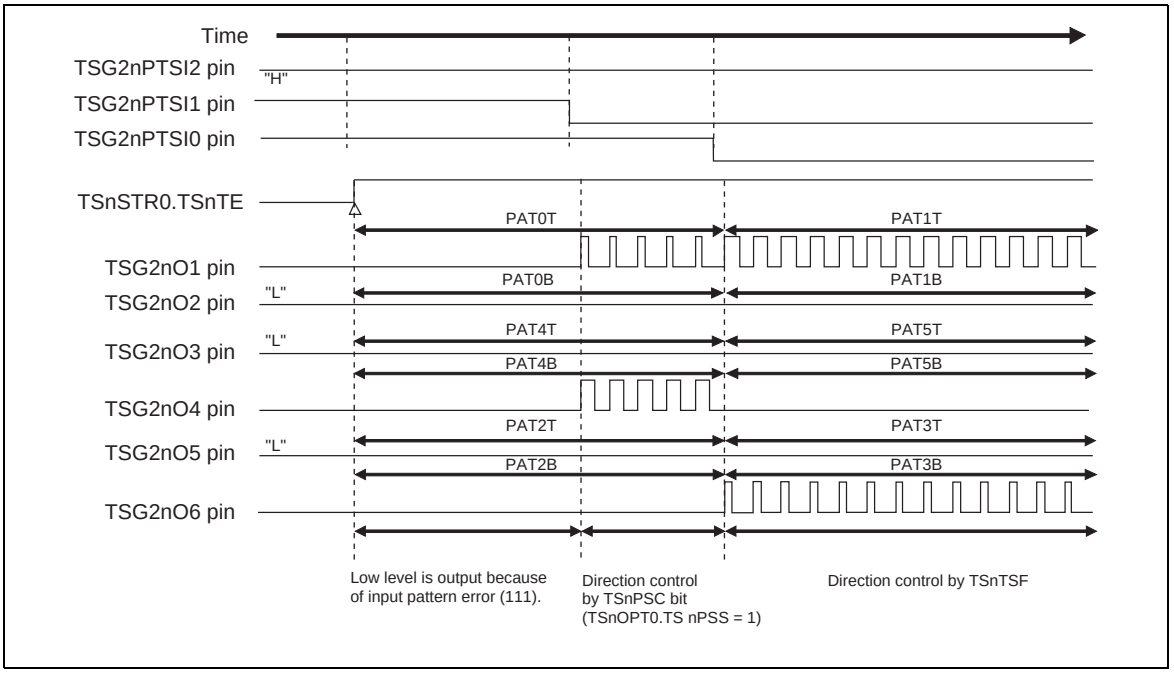


Figure 15-73 Control when Timer Output Starts in Normal Rotation (when Error Pattern is Input)

TSnOPT0.TSnSOC = 0, TSnPSC = 0, TSnPOT = 0, TSnIDC = 0

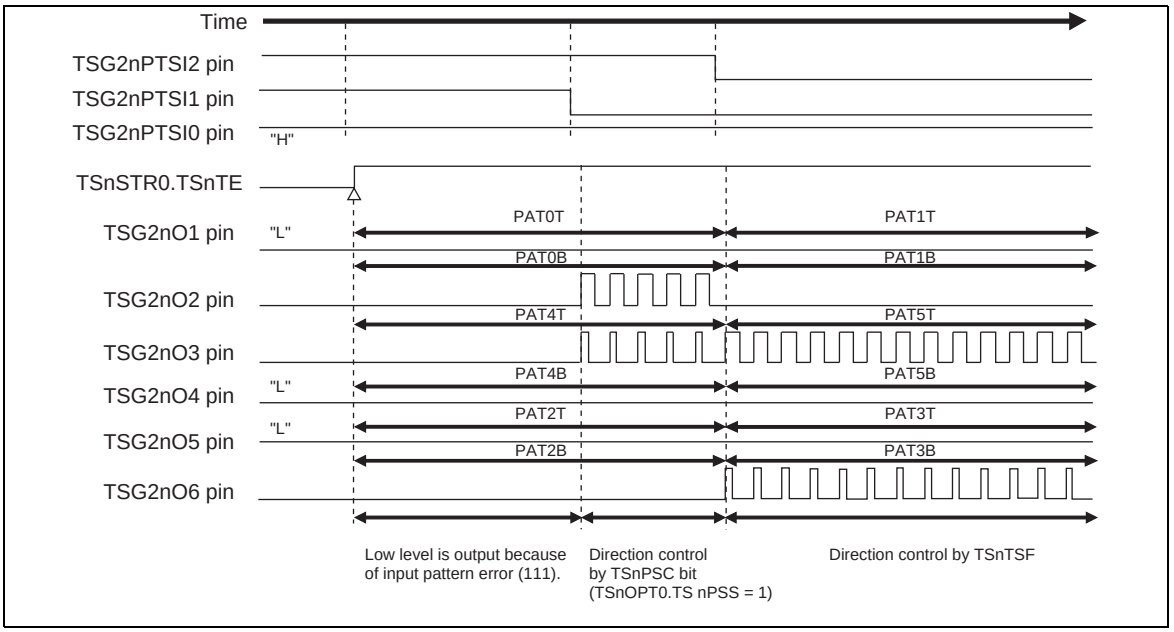


Figure 15-74 Control when Timer Output Starts in Reverse Rotation (when Error Pattern is Input)

TSnOPT0.TSnSOC = 0, TSnPSC = 1, TSnPOT = 0, TSnIDC = 1

(8) Output Switch Timing in 120-DC Mode

As shown in Figure 15-75 to Figure 15-78, in 120-DC mode, the external switch timing for output pattern (TSnOPCI0 and TSnOPCI1 signals, and TSG2nPTSI2 to TSG2nPTSI0 pins) is input irrespective of the 16-bit counter operation. The output is switched to the new pattern by clearing the 16-bit counter using the pattern switch timing signal applied from outside.

In the pattern switch method, if a change in TSG2nPTSI2 to TSG2nPTSI0 pins occurs consecutively within one period, the value edge detected immediately before the period match timing is reflected to PWM output in the following periods. In the trigger switch method, if TSnOPCI0 and TSnOPCI1 signal trigger is input for several times within one period, the output pattern is switched by clearing the 16-bit counter each time the trigger is accepted.

In the trigger switch method, if a rewrite to TSnOPT1.TSnSPC2 to TSnSPC0 and TSnOPCI0 and TSnOPCI1 trigger occur within one period, TSnSPC2 to TSnSPC0 are rewritten. If TSnSPC2 to TSnSPC0 are rewritten more than once within one period, the output pattern is switched by clearing the 16-bit counter each time TSnSPC2 to TSnSPC0 are rewritten.

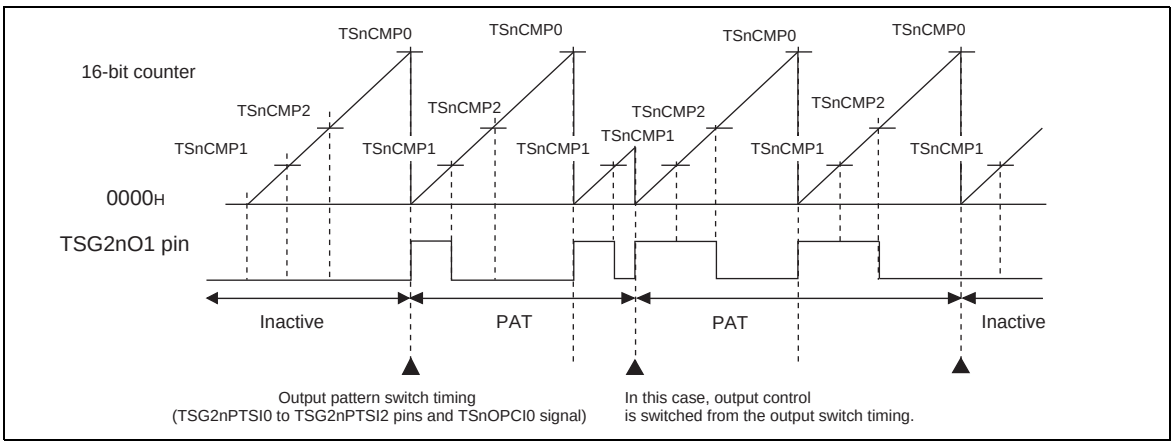


Figure 15-75 Output Switch Example (TSG2nPTSI2 to TSG2nPTSI0 Pins and TSnOPCI0 and TSnOPCI1 Signal Trigger Input)

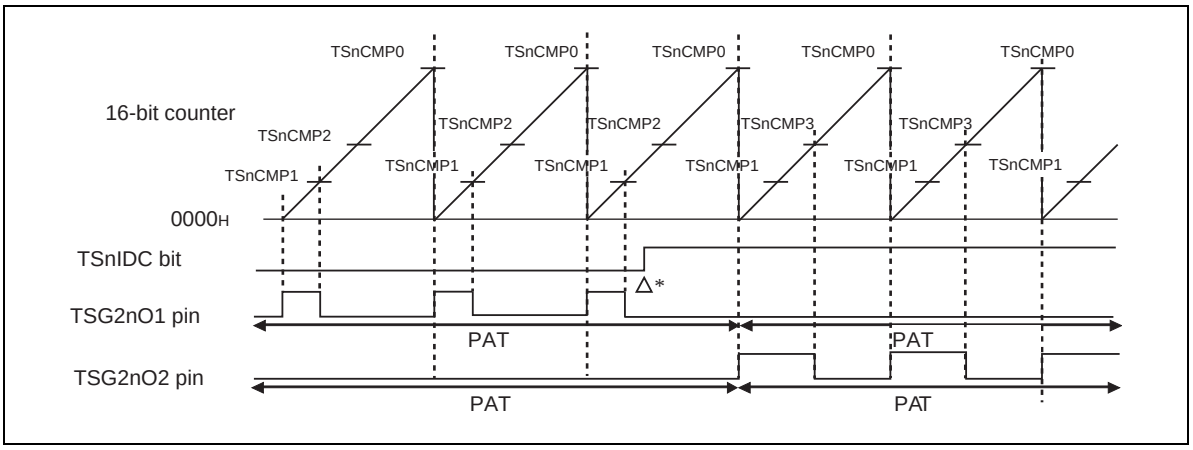


Figure 15-76 Output Switch Example (Switched by TSnOPT0.TSnIDC)

Note * When the pattern is switched here, output control is switched in the next period.

Note If a change in the TSG2nPTSI2 to TSG2nPTSI0 pins occurs by the time the next period when output control is switched by the TSnIDC bit, the 16-bit counter is cleared and output control is switched.

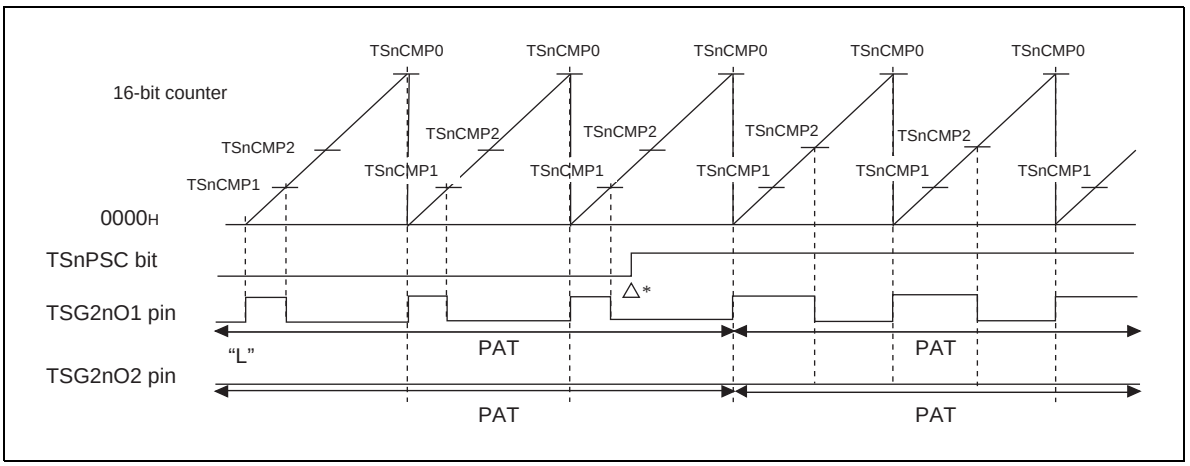


Figure 15-77 Output Switch Example (Switched by TSnOPT0.TSnPSC)

TSnOPT0.TSnPOT = 1

Note * When the pattern is switched here, output control is switched from the next period.

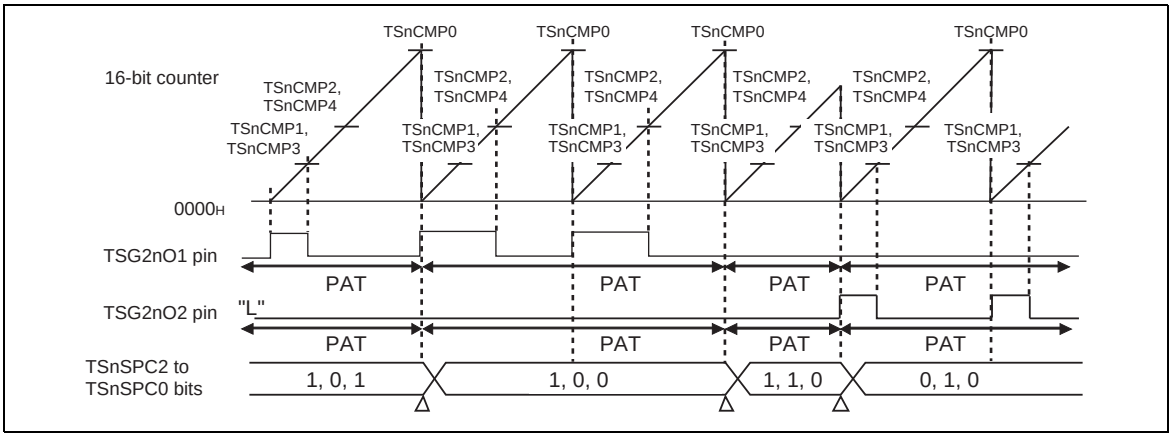


Figure 15-78 Output Switch Example (Switched by TSnOPT1.TSnSPC2 to TSnSPC0)

TSnOPT0.TSnSOC = 0, TSnPOT = 1

Note Δ: Write access

(9) Compare Register Rewrite Timing in 120-DC Mode

Example of operation when TSnCMP1 is reloaded (rewritten simultaneously) is shown below.

Figure 15-79 shows an output example when TSnCMP1 is rewritten. After TSnCMP1 is changed, data is not transferred to the TSnCMP1 buffer register (changed data is not valid) until the next reload timing; therefore, the specified output waveform can be obtained. However, do not write to TSnCMP1 again while the reload is suspended (period from when TSnCMP1 is changed to when simultaneous rewrite is executed). Be sure to read the reload request flag (TSnRSF) to confirm that the flag is 0, and write data to TSnCMP1.

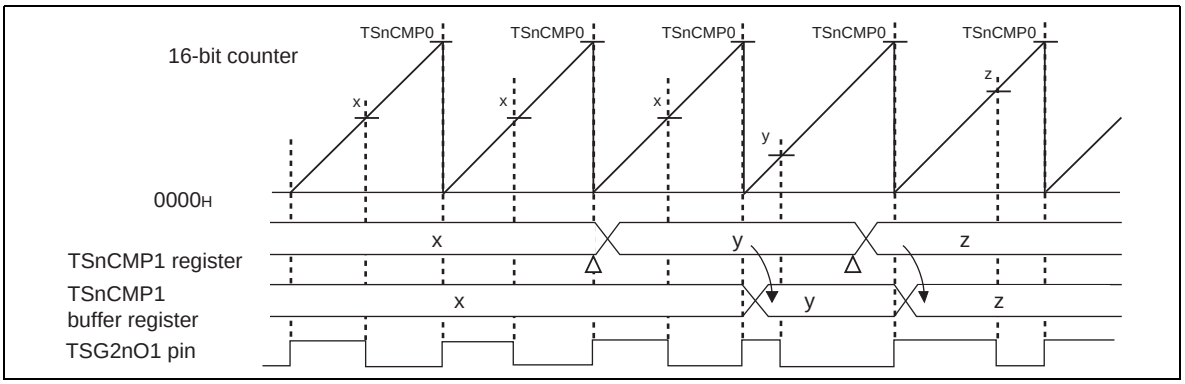


Figure 15-79 Output Example when TSnCMP1 is Rewritten

Note Δ: Write access

(10) Dead Time Control in 120-DC Mode

In 120-DC mode, the dead time is controlled on falling of each phase, and the dead time is added.

The dead time set in TSnDTC1 is inserted on falling of the positive phase, and the dead time set in TSnDTC0 is inserted on falling of the inverse phase.

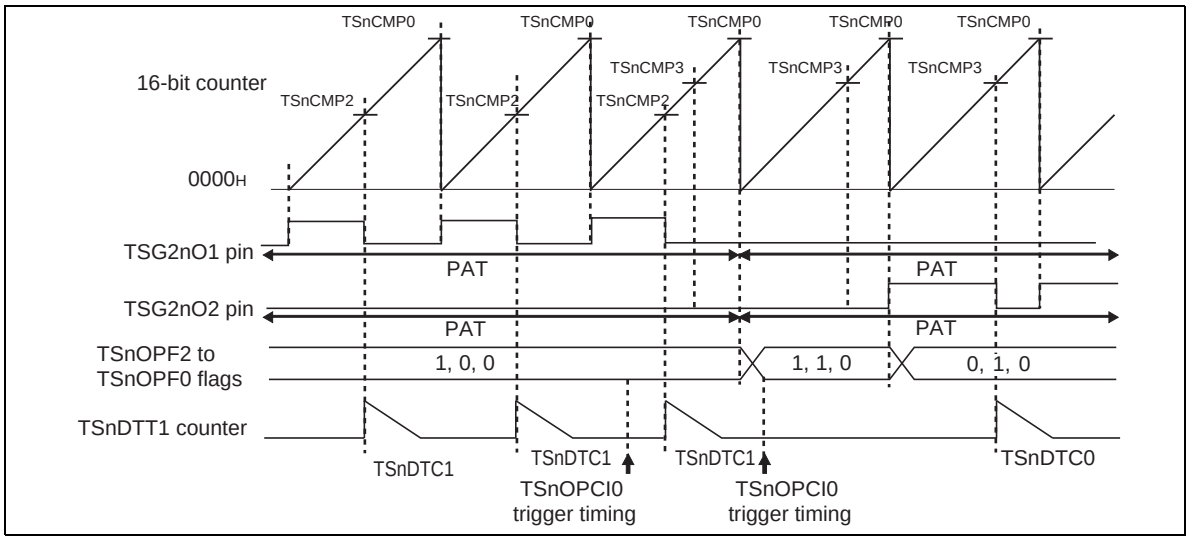


Figure 15-80 Output Switch Example

Caution The dead time control method may affect the timer output. The timer output may not have the specified active level width due to the dead time control under the following conditions:

- When noise is generated on the input pattern in the pattern switch method
- When a change in the input pattern occurs earlier than the PWM period in the pattern switch method
- When TSnOPT1.TSnSPC2 to TSnSPC0 are changed and the output pattern is forcibly changed in the trigger switch method
- When switch method is changed
- When the current direction control bit (TSnOPT0.TSnIDC) is changed
- When the software output control function is used

(11) Output Switch in 120-DC Mode

In 120-DC mode, the output pattern can be controlled by writing values to TSnOPT1.TSnSPC2 to TSnSPC0 when the trigger switch method (TSnOPT0.TSnSTE = 1, and TSnPOT = 1) is used. The dead time is secured by hardware at the switch timing.

Caution When 111_B or 000_B is written to TSnSPC2 to TSnSPC0, the TSG2nO1 to TSG2nO6 pins are driven low.

(12) Operation when Noise is Generated in TSG2nPTSI2 to TSG2nPTSI0 Pins in 120-DC Mode

Input to the TSG2nPTSI2 to TSG2nPTSI0 pins is assumed to be the hall sensor signals of the brushless DC motor. Depending on the system, a noise may be generated on the TSG2nPTSI2 to TSG2nPTSI0 pins. Operation when a noise is generated is described below.

For system product design, be sure to insert a noise filter circuit between the hall sensor and the TSG2nPTSI2 to TSG2nPTSI0 pins.

Figure 15-82 shows a case when a noise is generated on the TSG2nPTSI2 to TSG2nPTSI0 pins during operation with the pattern switch method used.

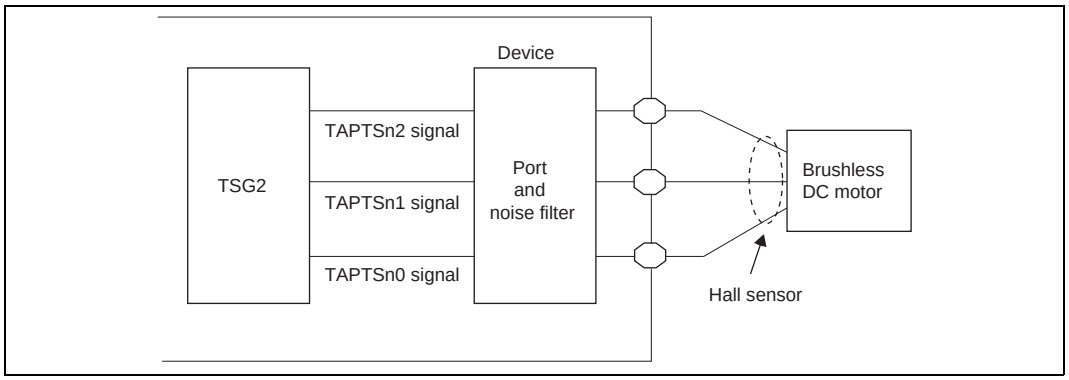


Figure 15-81 Example of Noise Filter Circuit Connection

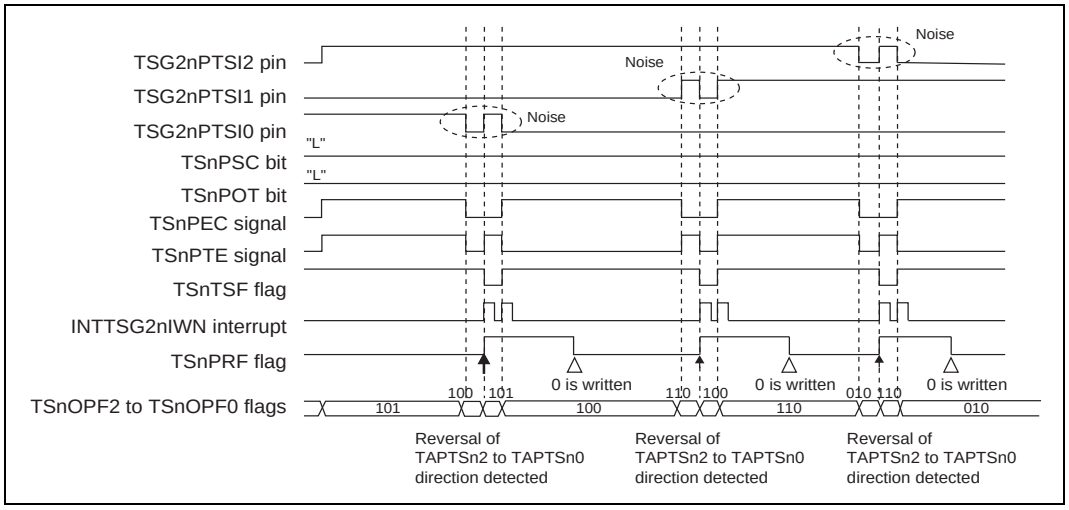


Figure 15-82 Example of Noise Generation at Level Change in TSG2nPTSI2 to TSG2nPTSI0 Pins (Pattern Switch Method)

(a) Change Timing of Input Pattern Change Detection Signal (TSnPTE)

- The TSnPTE signal toggles when the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) changes.

Caution Be sure to specify the rotation direction by TSnPSC (TSnPSS = 1 in TSnOPT0) in TSnOPT0.

(when TSnPSC = 0)

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	Toggle	-	-	-	-
	100	-	-	-	-	Toggle	-	-	-
	110	-	-	-	-	-	Toggle	-	-
	010	-	-	-	-	-	-	Toggle	-
	011	-	-	-	-	-	-	-	Toggle
	001	-	-	Toggle	-	-	-	-	-

(when TSnPSC = 1)

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	-	-	-	-	Toggle
	100	-	-	Toggle	-	-	-	-	-
	110	-	-	-	Toggle	-	-	-	-
	010	-	-	-	-	Toggle	-	-	-
	011	-	-	-	-	-	Toggle	-	-
	001	-	-	-	-	-	-	Toggle	-

(b) Change Timing of Three-Phase Encode Signal (TSnPEC)

- The TSnPEC signal toggles when input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) changes.

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 Pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	Toggle	-	-	-	Toggle
	100	-	-	Toggle	-	Toggle	-	-	-
	110	-	-	-	Toggle	-	Toggle	-	-
	010	-	-	-	-	Toggle	-	Toggle	-
	011	-	-	-	-	-	Toggle	-	Toggle
	001	-	-	Toggle	-	-	-	Toggle	-

(c) Change Timing of TSG2nO1 to TSG2nO6 Pins

- When the pattern switch method is used, the output pattern changes when the input signal of the TSG2nPTSI2 to TSG2nPTSI0 pins* changes. The output is also switched when two or more pins change simultaneously.
- When the trigger switch method is used, the output pattern changes at the rising edge of the TSnSTCI0 and TSnSTCI1 signals. The output also changes when data is written to TSnSPC2 to TSnSPC0* in TSnOPT0.

Note * When the input pattern changes to 000 or 111, the TSG2nO1 to TSG2nO6 pins are driven low.

(d) Change Timing of TSnTSF Flag

- The TSnTSF flag toggles when the input pattern (TSG2nPTSI2 to TSG2nPTSI0 pins) changes.

		TSG2nPTSI2 to TSG2nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG2nPTSI2 to TSG2nPTSI0 Pins	000	-	-	-	-	-	-	-	-
	111	-	-	-	-	-	-	-	-
	101	-	-	-	0	-	-	-	1
	100	-	-	1	-	0	-	-	-
	110	-	-	-	1	-	0	-	-
	010	-	-	-	-	1	-	0	-
	011	-	-	-	-	-	1	-	0
	001	-	-	0	-	-	-	1	-

(e) Set Timing of TSnNDF Flag

- The TSnNTF flag is set when two or more pins of the TSG2nPTSI2 to TSG2nPTSI0 pins change simultaneously, and cleared when 1 is written to the TSnNDR bit. The TSnNDF flag is valid when 1 is set to the TSnNDC bit.

(f) Set Timing of TSnPRF Flag

- The TSnPRF flag is set when the TSnTSF flag changes, and cleared when 1 is written to the TSnPRR bit. The TSnPRF flag is valid when 1 is set to the TSnPRC bit.

(g) Set Timing of TSnPEF Flag

- The TSnPRF flag is set when 000 or 111 is input to the TSG2nPTSI2 to TSG2nPTSI0 pins, and cleared when 1 is written to the TSnPER bit. The TSnPEF flag is valid when the TSnPEC bit is set to 1.

(13) Basic Control Flow in 120-DC Mode

In 120-DC mode, there are eight control states as listed in Table 15-80.

When TSnOPT0.TSnSTE = 1 and TSnPOT = 0, the pattern switch method is used for 120-DC control.

This is defined as fixed phase control. The fixed phase control should be performed considering the factors such as delay from the hall sensor and delay from sensor level detection to timer output. However, acceleration or deceleration can be performed simply by changing the PWM duty.

When TSnOPT0.TSnSTE = 1 and TSnPOT = 1, the trigger switch method is selected for 120-DC control.

This is defined as variable phase control. With the variable phase control, the timer output pattern is set prior to the hall sensor; therefore, acceleration or deceleration control according to the phase difference can be performed. However, control is more complex than the fixed phase control because offset width with respect to the hall sensor and the predicted value with respect to the hall sensor should be considered. For details, see section 26.4.31 Three-Phase Pulse Input Control Function.

When TSnOPT0.TSnPOT = 1 and TSnPSS = 1, the rotation direction of the motor can be set with TSnPSC in TSnOPT0. Set TSnPSC to 0 to set normal rotation, and 1 to set reverse rotation.

TSnIDC in TSnOPT0 sets the direction of control (acceleration or deceleration). If the same value as the rotation direction of the motor (TSnPSC set value) is set, acceleration control is set. If the different value from the rotation direction of the motor is set, deceleration control is set.

Table 15-80 Timer Control Status

Status	TSnPSC in TSnOPT0	TSnTSF in TSnSTR1	TSnIDC in TSnOPT0	TSnPOT in TSnOPT0	Control
A	-	0	0	0	Normal rotation, acceleration, and fixed phase
B	0	-	0	1	Normal rotation, acceleration, and variable phase
C	0	-	1	1	Normal rotation, deceleration, and variable phase
D	-	0	1	0	Normal rotation, deceleration, and fixed phase
E	-	1	1	0	Reverse rotation, acceleration, and fixed phase
F	1	-	1	1	Reverse rotation, acceleration, and variable phase
G	1	-	0	1	Reverse rotation, deceleration, and variable phase
H	-	1	0	0	Reverse rotation, deceleration, and fixed phase

Generally, the state, when the motor rotation stops, is assumed to be a state of the start and the control begins. First the fixed phase control is used to rotate the motor from the stopped state. Afterwards, to accelerate the motor speed to the fast rotation, the variable phase control is switched on. In combination with encoder timer (ENCA) and the variable phase control, the timer output is changed according to timing that is earlier than the change point of the hall sensors (leading).

To decelerate the motor speed from fast rotation, the direction of control is switched to deceleration control by rewriting only TSnIDC in TSnOPT0. When the rotation count can be reduced to low-speed rotation, the rotation can be placed in the stopped state by decreasing the PWM duty.

State transition is shown in Figure 15-83 and Figure 15-84.

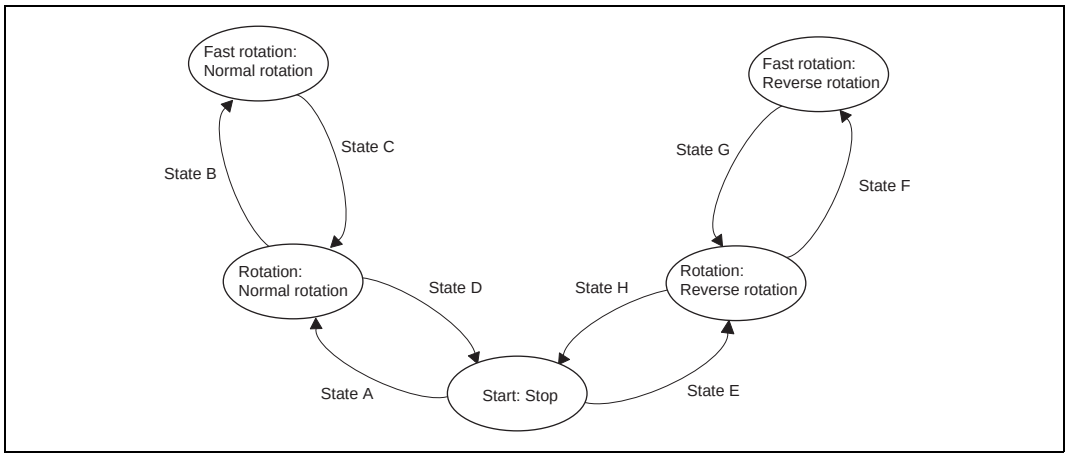


Figure 15-83 State Transition Diagram

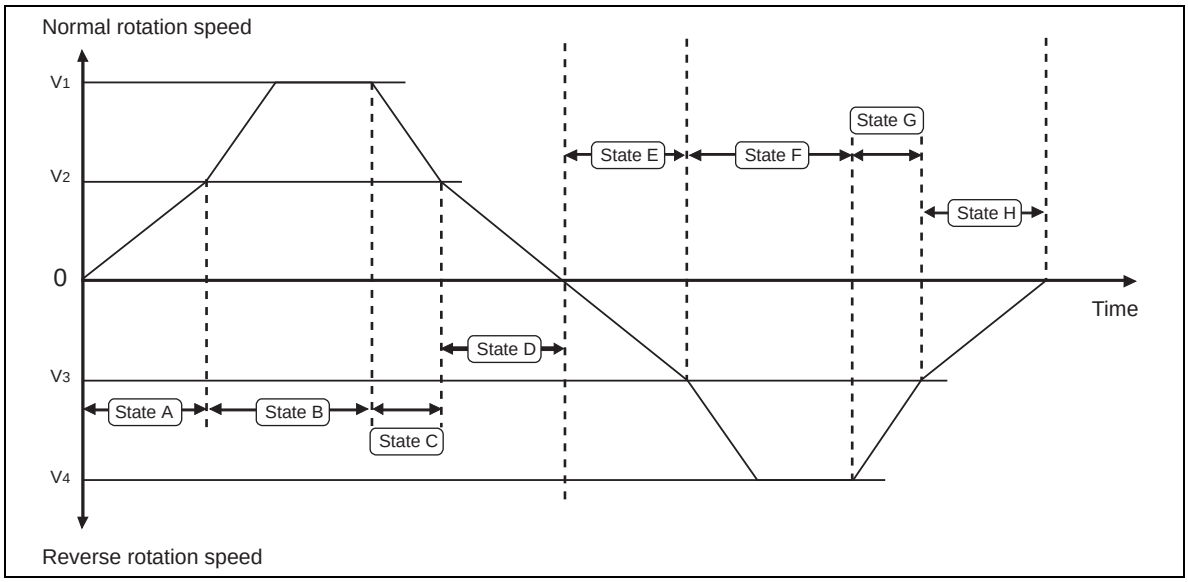


Figure 15-84 Relationship between State Transition and Rotation Speed of Motor

Note V1 and V4: Fast rotation speed of normal rotation and reverse rotation
V2 and V3: Low rotation speed of normal rotation and reverse rotation

(14) Software Output Control Function in 120-DC Mode

TSnOPT0.TSnSOC and TSnIDC, and TSnOPT1.TSnSPC2 to TSnSPC0 are used in 120-DC mode for timer output control by software.

As shown in Figure 15-85, the output control is switched immediately when TSnSOC is set to 1. If the dead time is set, the period of the dead time period is guaranteed. After that, to switch the output control from software output control to 120-DC control, set TSnSOC to 0. At this timing, output control is retained. When the reload timing is generated, output control is switched to 120-DC mode.

For details on software output control function, see section 15.11.5, Software Output Control Function.

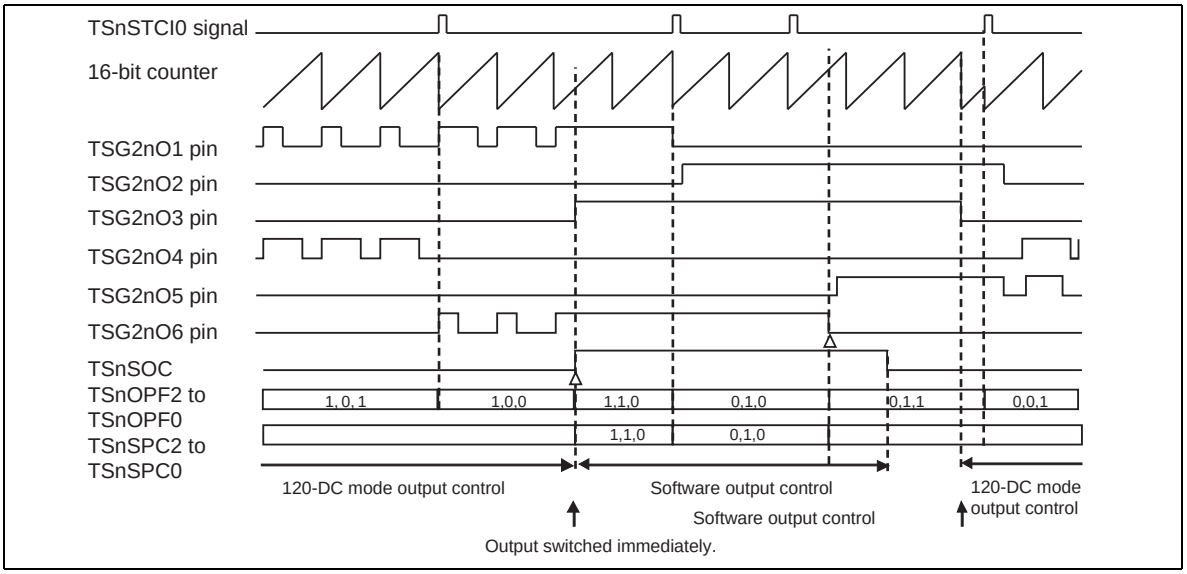


Figure 15-85 Example of Switching from 120-DC Mode to Software Output Control Function

Note Δ: Write access

(a) Procedure for Software Output Control

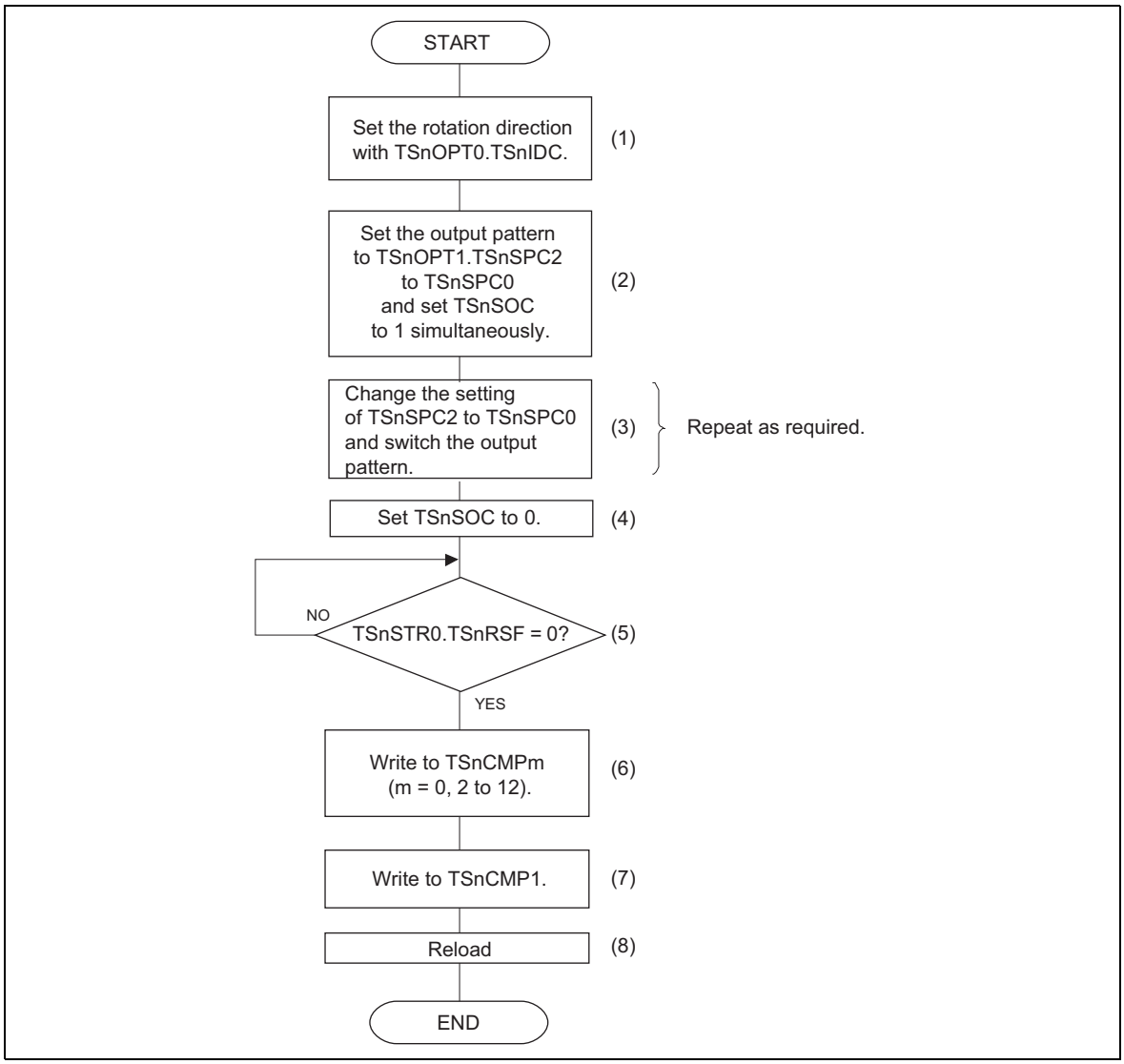


Figure 15-86 Process Flow of Software Output Control

The procedure for software output control is described below.

(1) Set TSnIDC to determine the rotation direction. The phase of the timer output with TSnIDC = 0 is different by 180 degrees from that with TSnIDC = 1. The timer output does not change if only this bit is rewritten using the software output control function. However, if the period match occurs before step (2), the output pattern of 120-DC control changes. So, schedule so as to prevent the period match from occurring before step (2).

(2) Set the output pattern to TSnSPC2 to TSnSPC0. To enable software output control, set TSnSOC to 1 simultaneously.

(3) Change the output pattern setting of TSnSPC2 to TSnSPC0 to change the timer output. The registers that can be changed during software control are: TSnTRG1.TSnTT, TSnCTL4 to TSnCTL6, TSnOPT0, TSnOPT1, TSnCMP0 to TSnCMP12, TSnDTC0, and TSnDTC1.

(4) Confirm that the reload request flag (TSnRSF) = 0. If TSnRFS = 1, do not proceed to the following step until TSnRSF = 0.

(5) By setting TSnSOC = 0 the software control starts to be released (it is not released here yet).

(6) After releasing the software output control, set the necessary compare registers. Proceed to the following step when the register setting is not required. Here, change the registers with the reload function.

(7) Write to TSnCMP1 to start reloading.

(8) Reload is executed and software output is released.

Caution Execute reload after executing steps (4) to (7). When reload cannot be executed, the software output cannot be released.

15.11.5 Software Output Control Function

Software output control function can be used in HT-PWM mode, SP-PWM mode, and 120-DC mode. This function can switch six output patterns for the TSG2nO1 to TSG2nO6 pins using TSnOPT0.TSnSOC, TSnIDC, and TSnOPT1.TSnSPC2 to TSnSPC0.

When TSnSOC is switched from 0 to 1, the output control method of the TSG2nO1 to TSG2nO6 pins is switched to the software output control immediately. On the contrast, when TSnSOC is switched from 1 to 0, the software output control is released at the reload timing.

Table 15-81 Register Description on Software Output Control Function

Register	Setting
TSnOPT0.TSnSOC	TSnSOC = 1
TSnOPT0.TSnSTE	TSnSTE = 0
TSnOPT1.TSnSPC2 to TSnSPC0	Set output patterns listed in Table 15-82 and Table 15-83.
TSnOPT0.TSnIDC	Set output pattern (rotation direction).

Table 15-82 Output Pattern of Software Output Control (TSnOPT0.TSnIDC = 0)

TSnOPT0.TSnSOC = 1, TSnSTE = 0, TSnIDC = 0

	TSnSTR1.TSnOPF2 to TSnOPF0							
Output Pin	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	ACT	ACT	ACT	INACT	INACT	INACT	INACT	ACT
TSG2nO2	INACT	INACT	INACT	ACT	ACT	ACT	ACT	INACT
TSG2nO3	INACT	INACT	ACT	ACT	ACT	INACT	INACT	ACT
TSG2nO4	ACT	ACT	INACT	INACT	INACT	ACT	ACT	INACT
TSG2nO5	ACT	INACT	INACT	INACT	ACT	ACT	INACT	ACT
TSG2nO6	INACT	ACT	ACT	ACT	INACT	INACT	ACT	INACT

Note ACT: Active level is output.
INACT: Inactive level is output.

Table 15-83 Output Pattern of Software Output Control (TSnOPT0.TSnIDC = 1)

TSnOPT0.TSnSOC = 1, TSnSTE = 0, TSnIDC = 1

	TSnSTR1.TSnOPF2-TSnOPF0							
Output Pin	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG2nO1	INACT	INACT	INACT	ACT	ACT	ACT	ACT	INACT
TSG2nO2	ACT	ACT	ACT	INACT	INACT	INACT	INACT	ACT
TSG2nO3	ACT	ACT	INACT	INACT	INACT	ACT	ACT	INACT
TSG2nO4	INACT	INACT	ACT	ACT	ACT	INACT	INACT	ACT
TSG2nO5	INACT	ACT	ACT	ACT	INACT	INACT	ACT	INACT
TSG2nO6	ACT	INACT	INACT	INACT	ACT	ACT	INACT	ACT

Note ACT: Active level is output.
INACT: Inactive level is output.

Section 16 TPBA (TPBA0 and TPBA1)

16.1 Functions of TPBAn

Instances This product provides 2 instances of TPBAn.

Table 16-1 Instances of TPBA

Timer Option	
Instances	2
Name	TPBA0, TPBA1

Instances index n Throughout this section, the instances of the TPBA are identified by the index "n" (n = 0, 1), for example TPBAnCTL for TPBAn control register.

Buffers index m Buffers are identified by index m (m = 00 to 63).

Register addresses TPBAn register addresses are given as address offsets from the individual base addresses <TPBAn_base0> or <TPBAn_base1>. Table 16-2 shows the base address of TPBAn.

Table 16-2 TPBAn Register Base Addresses

TPBAn	<TPBAn_base0> Address	<TPBAn_base1> Address
TPBA0	FF83 2000_H	FFFF $F000_H$
TPBA1	FF83 3000_H	FFFF $F400_H$

Clock supply TPBAn is connected to PCLK and is supplied with the PCLK clock signal input as listed below.

Table 16-3 TPBAn Clock Supply

TPBAn	Supplied Clock
TPBA0, TPBA1	PCLK

I/O signals The I/O signals of the TPBA are listed in table 16-4.

Table 16-4 List of TPBA I/O signals

TPBAn Signal	Function	Connected to
TPBnO	Timer output	Port
TPBAnSST	Synchronous start trigger input	PIC (input)

Interrupts The TPBA interrupt requests are listed in table 16-5.

Table 16-5 List of TPBA Interrupt Requests

TPBAn Interrupt Request	Function	Connected to
INTTPBAnIPRD	Period-matched detection interrupt	Interrupt controller DMA, DTS
INTTPBAnIDTY	Duty-cycle-matched detection interrupt	Interrupt controller DMA, DTS
INTTPBAnIPAT	Number-of-patterns matched detection interrupt	Interrupt controller DMA, DTS

16.2 Functional Overview

TPBAn is a 16-bit PWM timer with the duty setting buffer.

- Count clock resolution: Min. 12.5 ns (count clock: 80 MHz)
- 16-bit counter
- 16-bit duty register
- 16-bit period setting register
- 7-bit address counter register
- 7-bit pattern number setting register
- Interrupt request signals
  - Period-matched detection interrupt
  - Duty-cycle-matched detection interrupt
  - Number-of-patterns matched detection interrupt
- Number of duty patterns
  - 64 patterns (16 bits) or 128 patterns (8 bits)
- Automatic duty generation according to the number of patterns
- Output control by software
- The count clock can be selected from PCLK, PCLK/2, PCLK/4, and PCLK/8 according to the prescaler set value.
- Synchronous start with another timer

16.3 Configuration

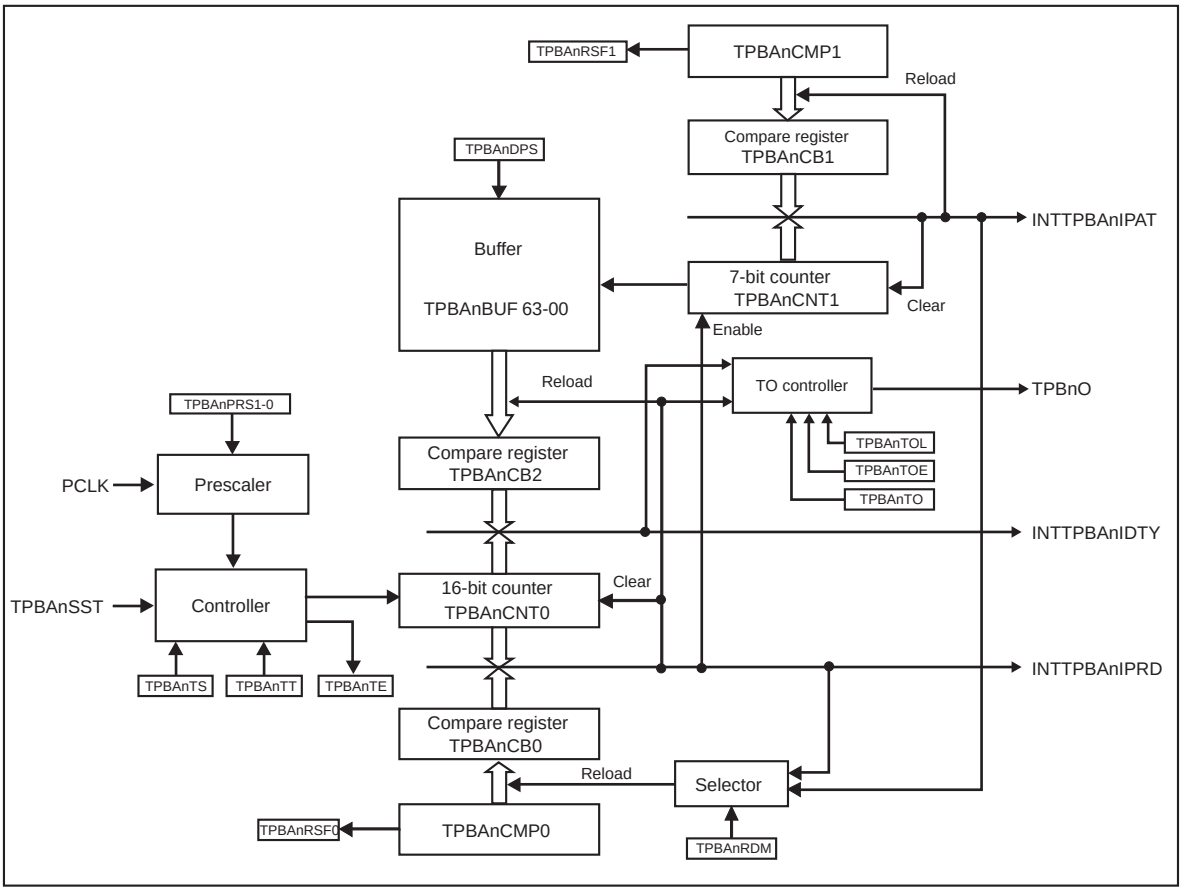


Figure 16-1 Block Diagram of TPBAn

16.4 Registers

This section contains a description of all registers of the TPBAn.

16.4.1 TPBAn Registers Overview

The TPBAn is controlled and operated based on the settings of the following registers listed in table 16-6.

Table 16-6 TPBAn Registers

Register Name	Symbol	Address
TPBAn control register	TPBAnCTL	<TPBAn_base0> + 200_H
TPBAn reload data mode register	TPBAnRDM	<TPBAn_base1> + 118_H
TPBAn reload status register	TPBAnRSF	<TPBAn_base1> + 110_H
TPBAn reload data trigger register	TPBAnRDT	<TPBAn_base1> + 114_H
TPBAn timer output enable register	TPBAnTOE	<TPBAn_base1> + 120_H
TPBAn timer output register	TPBAnTO	<TPBAn_base1> + $11C_H$
TPBAn timer output level register	TPBAnTOL	<TPBAn_base1> + 124_H
TPBAn period setting register	TPBAnCMP0	<TPBAn_base1> + 100_H
TPBAn duty setting register	TPBAnBUFm	<TPBAn_base1> + 000_H to $0FC_H$
TPBAn pattern number setting register	TPBAnCMP1	<TPBAn_base1> + 104_H
TPBAn timer counter register	TPBAnCNT0	<TPBAn_base1> + 108_H
TPBAn address counter register	TPBAnCNT1	<TPBAn_base1> + $10C_H$
TPBAn enable status register	TPBAnTE	<TPBAn_base1> + 128_H
TPBAn start trigger register	TPBAnTS	<TPBAn_base1> + $12C_H$
TPBAn stop trigger register	TPBAnTT	<TPBAn_base1> + 130_H

16.4.2 TPBAn Registers Details

(1) TPBAn Control Register (TPBAnCTL)

This register specifies the operation of the TPBAn.

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

Address <TPBAn_base0> + 200_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	TPBAn PRS1	TPBAn PRS0	-	-	-	TPBAn DPS
R	R	R/W	R/W	R	R	R	R/W

Table 16-7 TPBAnCTL0 Register Contents

Bit Position	Bit Name	Function
4, 5	TPBAnPRS [1:0]	Selects the count clock. **TPBAnPRS1 / TPBAnPRS0 / Description** 0 / 0 / PCLK is selected. 0 / 1 / PCLK/2 is selected. 1 / 0 / PCLK/4 is selected. 1 / 1 / PCLK/8 is selected.
0	TPBAnDPS	Selects the duty setting pattern mode. 0: 16 bits × 64 patterns mode 1: 8 bits × 128 patterns mode

Caution This register should be set when the timer is stopped (TPBAnTE = 0). If this register is erroneously rewritten, set the register again after stopping the timer.

(2) TPBAn Reload Data Mode Register (TPBAnRDM)

This register controls the reload timing of the TPBAn period setting register and TPBAn timer output level register values.

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

Address <TPBAn_base1> + 118_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAn RDM0
R	R	R	R	R	R	R	R/W

Table 16-8 TPBAnRDM Register Contents

Bit Position	Bit Name	Function
0	TPBAnRDM0	Controls the reload timing of the TPBAn period setting register (TPBAnCMP0) and TPBAn timer output level register (TPBAnTOL) values. 0: Reloads the values synchronously with a number-of-patterns matched detection interrupt (INTTPBAnIPAT). 1: Reloads the values synchronously with a period-matched detection interrupt (INTTPBAnIPRD).

Caution Although this register can be rewritten during operation, the rewritten value is reflected at any time. Accordingly, during operation, this register should be rewritten when the reload request flag (TPBAnRSF) is 0.

(3) TPBAn Reload Status Register (TPBAnRSF)

This register indicates whether or not reload requests from the corresponding registers have been generated.

Access This register can be read in 8-bit units.

Address <TPBAn_base1> + 110_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TPBAn RSF1	TPBAn RSF0
R	R	R	R	R	R	R	R

Table 16-9 TPBAnRFS Register Contents

Bit Position	Bit Name	Function
1	TPBAnRSF1	Indicates whether or not a reload request from TPBAnCMP1 has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT1 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed.
0	TPBAnRSF0	Indicates whether or not a reload request from TPBAnCMP0 and TPBAnTOL has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT0 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed.

(4) TPBAn Reload Data Trigger Register (TPBAnRDT)

This register enables reload of the register values.

Access This register can be written in 8-bit units. It is always read as 0.

Address <TPBAn_base1> + 114_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TPBAn RDT1	TPBAn RDT0
R	R	R	R	R	R	W	W

Table 16-10 TPBAnRDT Register Contents

Bit Position	Bit Name	Function
1	TPBAnRDT1	Enables reload of the TPBAnCMP1 values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF1 is set to 1). The values are updated simultaneously at the next reload timing (reload).
0	TPBAnRDT0	Enables reload of the TPBAnCMP0 and TPBAnTOL values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF0 is set to 1). The values are updated simultaneously at the next reload timing (reload).

(5) TPBAn Timer Output Enable Register (TPBAnTOE)

This register enables or disables the timer output.

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

Address <TPBAn_base1> + 120_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAn TOE0
R	R	R	R	R	R	R	R/W

Table 16-11 TPBAnTOE Register Contents

Bit Position	Bit Name	Function
0	TPBAnTOE0	Enables or disables the timer output (TPBnO). 0: Disables the timer output based on counter operation. 1: Enables the timer output based on counter operation. • When the timer output is disabled, the level specified in TPBAnTO is output from the TPBnO pin, and can be controlled by software. • When the timer output is enabled, TPBAnTO is set or cleared by the timer operation, and a PWM signal is output. Write access is prohibited (ignored).

(6) TPBAn Timer Output Register (TPBAnTO)

This register reads the output settings and the output level.

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

Address <TPBAn_base1> + $11C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAn TO0
R	R	R	R	R	R	R	R/W

Table 16-12 TPBAnTO Register Contents

Bit Position	Bit Name	Function
0	TPBAnTO0	Reads the setting of the TPBnO pin output and the output level • When the timer output is disabled (TPBAnTOE.TPBAnTOE0 = 0) 0: Outputs low level. 1: Outputs high level. • When the timer output is enabled (TPBAnTOE.TPBAnTOE0 = 1) 0: Low level is being output by the timer output. 1: High level is being output by the timer output. When the timer output is enabled, rewrite to this register is ignored.

(7) TPBAn Timer Output Level Register (TPBAnTOL)

This register controls the timer output level.

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

Address <TPBAn_base1> + 124_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAn TOL0
R	R	R	R	R	R	R	R/W

Table 16-13 TPBAnTOL Register Contents

Bit Position	Bit Name	Function
0	TPBAnTOL0	Specifies the active level of the timer output. 0: High 1: Low. • Setting of this bit is enabled when the timer output is enabled (TPBAnTOE.TPBAnTOE0 = 1). • Setting of this bit is reflected when the timer output is started, and change of the output level is reflected at the next reload timing.

Caution This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see section 16.5.2, Compare Register Rewrite Operation.

(8) TPBAn Period Setting Register (TPBAnCMP0)

This is a 16-bit compare register for setting the PWM period.

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

Address <TPBAn_base1> + 100_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPBAnCMP0															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-14 TPBAnCMP0 Register Setting

Operating Mode	PWM Period	Minimum Value (Period)	Maximum Value (Period)
8 bits	TPBAnCMP0 + 1	1	100_H
16 bits	TPBAnCMP0 + 1	1	10000_H

Caution The PWM period is (TPBAnCMP0 + 1) count clock periods. Accordingly, for PWM output with 100% duty cycle, the maximum settable value is $FFFE_H$ (FE_H).

Caution This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see section 16.5.2, Compare Register Rewrite Operation.

(9) TPBAn Duty Setting Register (TPBAnBUFm)

This register is a 16 × 64 buffer register for duty setting.

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

Address <TPBAn_base1> + 000_H to $0FC_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPBAnBUFm															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-15 TPBAnBUFm Register Contents

Bit Position	Bit Name	Function
15 to 0	TPBAnBUFm15 to TPBAnBUFm0	Sets the duty value. This register can set the duty value either in 16 bits × 64 patterns mode (TPBAnDPS = 0) or 8 bits × 128 patterns mode (TPBAnDPS = 1) by setting the TPBAnDPS bit. In either mode, this register is accessed in 16-bit units by the CPU. For details, see section 16.5.3, Duty Rewrite Operation.

Caution The value set to this register is transferred to the duty setting buffer register (TPBAnCB2) synchronously with a period-matched detection interrupt (INTTPBAnIPRD). Rewrite during timer operation is reflected at any time. For details, see section 16.5.3, Duty Rewrite Operation.

(10) TPBAn Pattern Number Setting Register (TPBAnCMP1)

This register sets the number of PWM output patterns.

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

Address <TPBAn_base1> + 104_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	TPBAnCMP1						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-16 TPBAnCMP1 Register Contents

Bit Position	Bit Name	Function
6 to 0	TPBAnCMP1 [6:0]	Sets the number of patterns within the following range. TPBAnDPS = 0: 0 to 63 TPBAnDPS = 1: 0 to 127

Caution This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see section 16.5.2, Compare Register Rewrite Operation.

Caution If 64 or a greater number is set as the number of patterns when the duty setting pattern is in 16 bits × 64 patterns mode (TPBAnDPS = 0), the address pointer changes from 63 to 00, and the duty value is transferred from 00 again. A number-of-patterns matched detection interrupt signal (INTTPBAnIPAT) is output by the match of the specified number of patterns and the lower 7-bit values of TPBAnCNT1.

(11) TPBAn Timer Counter Register (TPBAnCNT0)

This register is a 16-bit counter that generates PWM output.

Access This register can only be read in 16-bit units.

Address <TPBAn_base1> + 108_H

Initial value $FFFF_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPBAnCNT0															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16-bit counter This register is a counter register through which the 16-bit counter value can be read.

(12) TPBAn Address Counter Register (TPBAnCNT1)

This register is a counter register that indicates the address pointer to the duty setting register.

Access This register can only be read in 8-bit units.

Address <TPBAn_base1> + $10C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	TPBAnCNT1						
R	R	R	R	R	R	R	R

7-bit counter This register indicates TPBAnBUFm that holds the duty value to be transferred next.

(13) TPBAn Enable Status Register (TPBAnTE)

This register indicates whether the timer counter is operating or stopped.

Access This register can only be read in 8-bit units.

Address <TPBAn_base1> + 128_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAnTE0
R	R	R	R	R	R	R	R

Table 16-17 TPBAnTE Register Contents

Bit Position	Bit Name	Function
0	TPBAnTE0	Indicates whether the timer counter is operating or stopped. 0: The timer counter is stopped. 1: The timer counter is operating. • The TPBAnTE0 bit is set to 1 when 1 is written to the TPBAnTS bit or when a synchronous start trigger is input. • The TPBAnTE0 bit is cleared to 0 when 1 is written to the TPBAnTT bit.

(14) TPBAn Start Trigger Register (TPBAnTS)

This register controls the timer counter start trigger.

Access This register can only be written in 8-bit units. It is always read as 0.

Address <TPBAn_base1> + $12C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAnTS0
R	R	R	R	R	R	R	W

Table 16-18 TPBAnTS Register Contents

Bit Position	Bit Name	Function
0	TPBAnTS0	This bit is a trigger bit that enables the timer counter. 0: Write access is ignored. 1: Starts counting (TPBAnTE = 1).

Caution Write access to this register during counting (TPBAnTE = 1) is ignored.

(15) TPBAn Stop Trigger Register (TPBAnTT)

This register controls the timer counter stop trigger.

Access This register can only be written in 8-bit units. It is always read as 0.

Address <TPBAn_base1> + 130_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TPBAnTT0
R	R	R	R	R	R	R	W

Table 16-19 TPBAnTT Register Contents

Bit Position	Bit Name	Function
0	TPBAnTT0	This bit is a trigger bit that disables the timer counter. 0: Write access is ignored. 1: Disables counting (TPBAnTE = 0).

16.5 Basic Operation

16.5.1 Basic Operation of Counter

(1) Basic Operation of 16-Bit Counter (TPBAnCNT0)

Counting start The 16-bit counter (TPBAnCNT0) starts counting from the initial value $FFFF_H$.

Counter clear The 16-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB0) set value of TPBAnCMP0.

Counter read during counting The 16-bit counter value during counting can be read through TPBAnCNT0.

(2) Basic Operation of 7-Bit Counter (TPBAnCNT1)

Counting start The 7-bit counter (TPBAnCNT1) is initialized to 00_H and starts counting. Subsequently, the counter value is incremented synchronously with a period-matched detection interrupt (INTTPBAnIPRD).

Counter clear The 7-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB1) set value of TPBAnCMP1.

Counter read during counting The 7-bit counter value during counting can be read through TPBAnCNT1. The read value indicates TPBAnBUFm in which the duty value to be transferred next is stored.

16.5.2 Compare Register Rewrite Operation

The following registers are rewritten by reload.

- TPBAnCMP0
- TPBAnCMP1
- TPBAnTOL

Reload mode (simultaneous rewrite function)

Writing to TPBAnRDT enables reload of the registers corresponding to the set bits (sets the reload request flag (TPBAnRSF.TPBAnRSFk)), and the values of all the pertinent registers are updated simultaneously at the next reload timing (reload).

The reload timing of TPBAnCMP0 and TPBAnTOL is set by TPBAnRDM.

The reload timing of TPBAnCMP1 is the match timing (INTTPBAnIPAT) of the 7-bit counter (TPBAnCNT1) and the buffer register (TPBAnCB1) of TPBAnCMP1.

The registers to be reloaded should be rewritten when the reload request flag (TPBAnRFS.TPBAnRSFk) is 0.

Note: k = 0, 1

Setting Flow for Registers to Be Reloaded

The rewritten values of the registers to be reloaded (TPBAnCMP0, TPBAnCMP1, and TPBAnTOL) can be transferred to the respective buffer registers simultaneously at the reload timing.

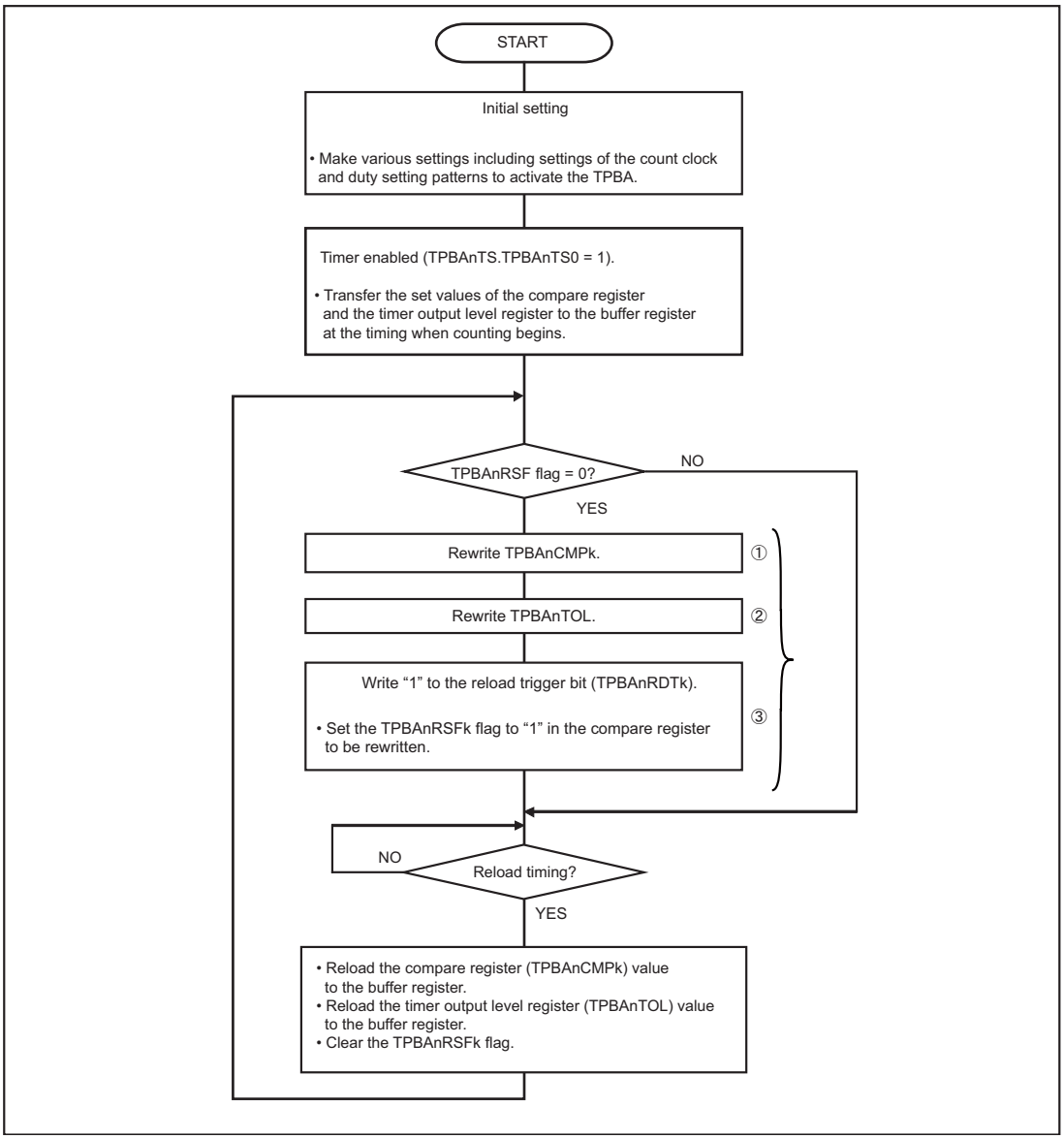


Figure 16-2 Basic Operation Flow of Reload (Simultaneous Rewrite Function)

Note: k = 0, 1

Caution Setting the TPBAnRDTk bit to 1 enables reload. Accordingly, the TPBAnRDTk bit should be rewritten after the registers to be reloaded have been rewritten.

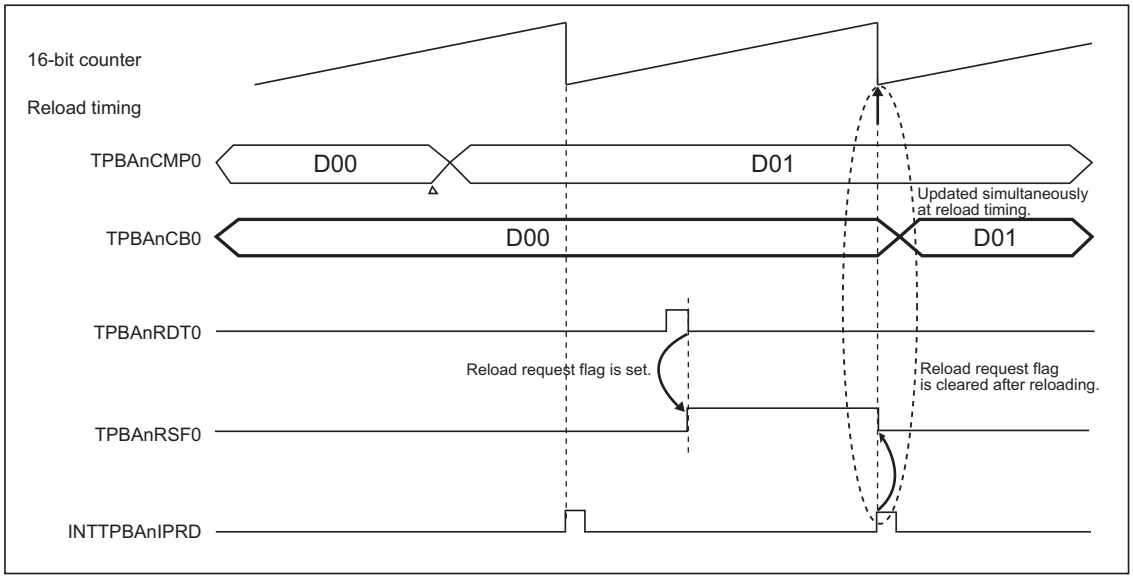


Figure 16-3 Simultaneous Rewrite Timing (TPBAnDPS = 0, TPBAnRDM = 0, and TPBAnTOL = 0)

16.5.3 Duty Rewrite Operation

TPBAnBUFm can be rewritten during operation.

The rewritten setting is reflected immediately.

(1) TPBAnBUFm Setting Flow

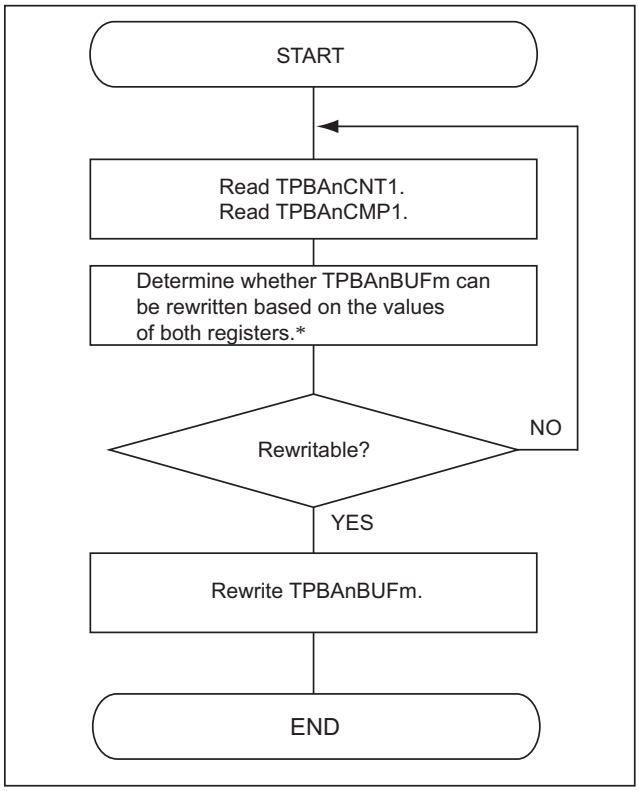


Figure 16-4 Basic Rewrite Flow of TPBAnBUFm

Note * Since TPBAnBUFm can be written at any time, it is recommended that this register is rewritten during last duty cycle output (TPBAnCNT1 = 00_H) of the pattern period (the number of patterns specified by TPBAnCMP1).

(2) Access to TPBAnBUFm

TPBAnBUFm is accessed in 16 bit units. The following shows the access in 16 bits × 64 patterns mode and the access in 8 bits × 128 patterns mode.

- In 16 bits × 64 patterns mode (TPBAnDPS = 0)

  This register is accessed by the CPU in units of one 16-bit pattern.

15	0
Pattern 64	$00FC_H$
Pattern 63	$00F8_H$
:	:
Pattern 3	0008_H
Pattern 2	0004_H
Pattern 1	0000_H

- In 8 bits × 128 patterns mode (TPBAnDPS = 1)

  This register is accessed by the CPU in units of two 8-bit patterns.

15	0	
Pattern 128	Pattern 127	$00FC_H$
Pattern 126	Pattern 125	$00F8_H$
:	:	:
Pattern 6	Pattern 5	0008_H
Pattern 4	Pattern 3	0004_H
Pattern 2	Pattern 1	0000_H

(3) Relationship between TPBAnCNT1 Read Value and TPBAnBUFm

The duty value of the currently output PWM waveform can be obtained by reading the TPBAnCNT1 count value during operation. TPBAnBUFm in which the currently output duty value is stored can be found by one of the following formulas.

TPBAnDPS0 Bit	**Formula**		
	TPBAnCNT1 ≠ 00_H		TPBAnCNT1 = 00_H
0: 16 bits × 64 patterns mode	(1) TPBAnCNT1 - 01_H		(2) TPBAnCMP1
1: 8 bits × 128 patterns mode	TPBAnCNT1 value is an odd number	(3) TPBAnCNT1/2	(5) TPBAnCMP1/2
	TPBAnCNT1 value is an even number	(4) (TPBAnCNT1 / 2) - 01_H	

(1) When TPBAnDPS = 0 and the TPBAnCNT1 ≠ 00_H

The applicable register is found by the formula TPBAnCNT1 - 01_H.

(Example) When TPBAnCNT1 = 08_H: 08_H - 01_H = 07_H -> TPBA0BUF07

(2) When TPBAnDPS = 0 and the TPBAnCNT1 = 00_H

The applicable register is found by the TPBAnCMP1 value.

(Example) When TPBAnCMP1 = 08_H: TPBAnBUF08

(3) When TPBAnDPS = 1 and the TPBAnCNT1 = an odd number

The applicable register is found by the formula TPBAnCNT1 / 2

(Example) When TPBAnCNT1 = 07_H: 07_H / 02_H = 03_H -> TPBAnBUF03 (lower 8 bits)

(4) When TPBAnDPS = 1 and the TPBAnCNT1 = an even number

The applicable register is found by the formula (TPBAnCNT1 / 2) - 01_H.

(Example) When TPBAnCNT1 = 08_H: (08_H / 02_H) - 01_H = 03_H -> TPBAnBUF03 (upper 8 bits)

(5) When TPBAnDPS = 1 and the TPBAnCNT1 = 00_H

The applicable register is found by the formula TPBAnCMP1 / 2.

(Example) When TPBAnCMP1 = 08_H: 08_H / 2 = 04_H -> TPBAnBUF04 (lower 8 bits)

16.5.4 Basic Operation Example

Overview

A PWM signal is output from the TPBnO pin according to the PWM period set in the TPBAnCMP0 register and duty cycle set in the TPBAnBUF00 to TPBAnBUF63 registers.

Prerequisites

- Select 16 bits × 64 patterns mode or 8 bits × 128 patterns mode by setting TPBAnDPS.
- Set the duty cycle to TPBAnBUF00 to TPBAnBUF63.
- Set the number of patterns to TPBAnCMP1.

Functional description

Set the PWM period, the number of patterns, duty cycle, and level to be output. Set TPBAnTS.TPBSnTS0 = 1 (or input a synchronous start trigger) to start incrementing the timer counter value.

The TPBnO output is set to the active level at the same time the counting begins. TPBAnCNT1 is incremented, and points to the address of the buffer in which the subsequent duty value is stored.

The output is set to the inactive level by the match of the 16-bit counter and the TPBAnBUFm buffer register (TPBAnCB2).

The duty value is then transferred from TPBAnBUFm to the buffer register (TPBAnCB2) by the match of the 16-bit counter and the TPBAnCMP0 buffer register (TPBAnCB0). Then, TPBAnCNT1 is incremented, and a period-matched detection interrupt (INTTPBAnIPRD) is generated. The TPBnO output is set to the active level after one count clock.

During counting, a duty-cycle-matched detection interrupt (INTTPBAnIDTY) is generated by the match of the 16-bit counter and the buffer register (TPBAnCB2) of TPBAnBUFm.

A number-of-patterns matched detection interrupt (INTTPBAnIPAT) is generated by the match of the 7-bit counter and the TPBAnCMP1 buffer register (TPBAnCB1).

(1) List of Operations

Table 16-20 16-Bit Counter Function

Operation		Setting Condition
16-bit counter	Start	Writing 1 to TPBAnTS or synchronous start trigger
	Clear	Compare match of TPBAnCMP0 buffer register and 16-bit counter
	Stop	Writing 1 to TPBAnTT

Table 16-21 7-Bit Counter Function

Operation		Setting Condition
7-bit counter	Start	Writing 1 to TPBAnTS or synchronous start trigger
	Clear	Compare match of TPBAnCMP1 buffer register and 7-bit counter
	Stop	Writing 1 to TPBAnTT

Table 16-22 Functions of Compare Registers

Register	Rewrite Method	Rewrite during Operation	Function
TPBAnCMP0	Reload	Possible	Setting period
TPBAnCMP1	Reload	Possible	Setting number of patterns
TPBAnBUFm	Rewrite at any time	Possible	Setting duty
TPBAnTOL	Reload	Possible	Setting output level

Table 16-23 Timer Output Function

Pin	Function
TPBnO	• When output is enabled (TPBAnTOE = 01_H) PWM output by compare match of the TPBAnBUFm buffer register (TPBAnCB2) and the 16-bit counter • When output is disabled (TPBAnTOE = 00_H) TPBAnTO set value

Table 16-24 Interrupt Requests

Interrupt	Function
INTTPBAnIPRD	Period-matched detection interrupt
INTTPBAnIDTY	Duty-cycle-matched detection interrupt
INTTPBAnIPAT	Number-of-patterns matched detection interrupt

Table 16-25 Compare Match Timing

Compare Match	Timing
TPBAnCMP0	When the 16-bit counter changes from TPBAnCMP0 to 0000_H.
TPBAnCMP1	When the 7-bit counter changes from TPBAnCMP1 to 00_H.
TPBAnBUFm	When the 16-bit counter matches with the buffer register (TPBAnCB2).

Table 16-26 Example of Setting Each Timer Output Condition

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TPBnO	PWM output	(TPBAnCMP0 + 1) × count clock	Outputs an inactive level throughout one period (duty cycle 0%).	TPBAnBUFm = 0000_H
			Outputs an active level of one count clock in one period.	TPBAnBUFm = 0001_H
			Outputs an inactive level of one count clock in one period	TPBAnBUFm = TPBAnCMP0
			Outputs an active level throughout one period (duty cycle 100%).	TPBAnBUFm ≥ TPBAnCMP0 + 1

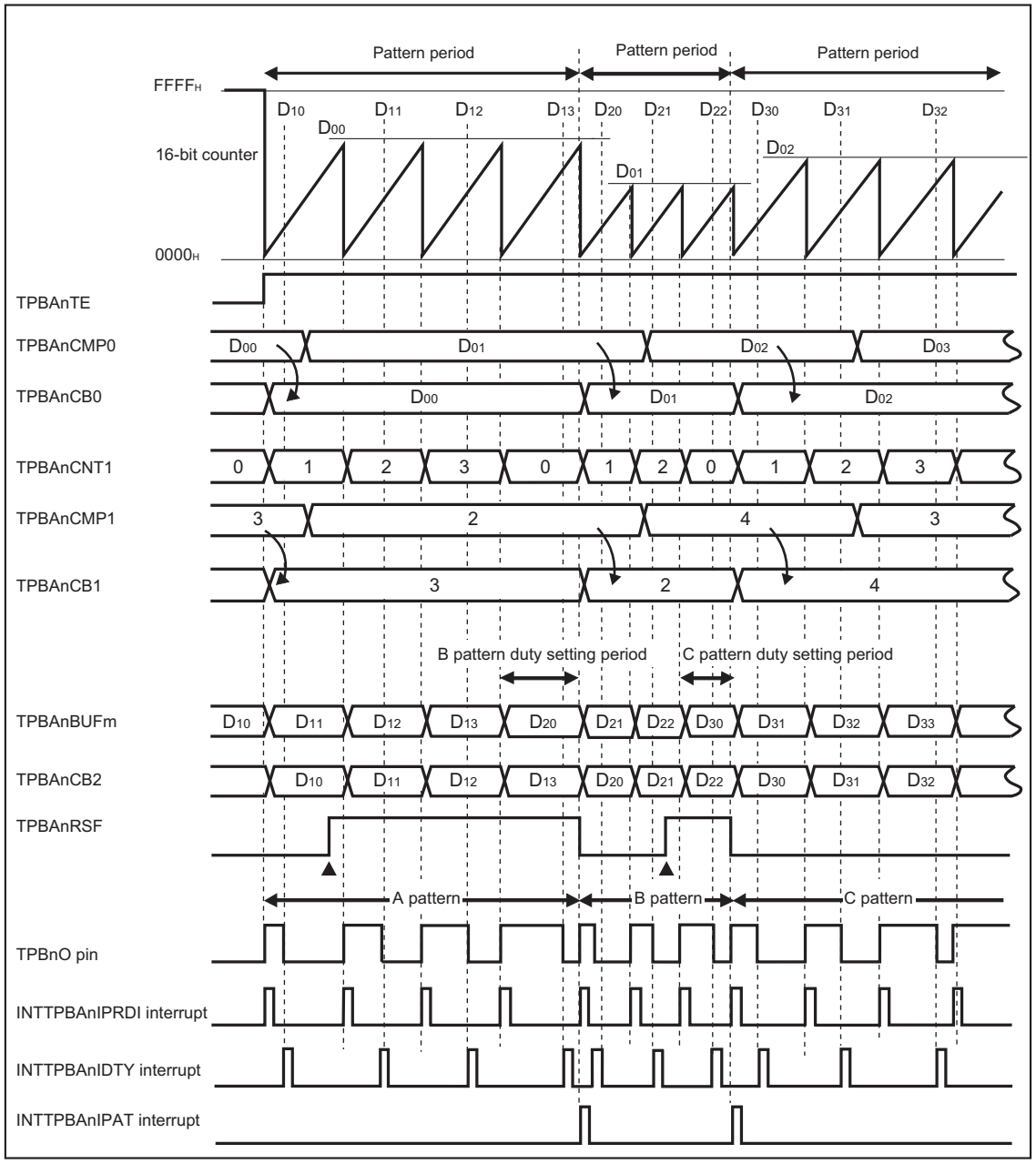


Figure 16-5 Example of Basic Timing (1/2)

When a number-of-patterns matched detection interrupt is used as a trigger of the TPBAnCMP0 and TPBAnTOL reload timing (TPBAnIRDM.TPBAnRDM0 = 0 and TPBAnTOL = 0)

Note 1. PWM duty cycle = (TPBAnBUFm) / (TPBAnCMP0 set value + 1)
PWM period = (TPBAnCMP0 set value + 1) × (count clock period)

Note 2. ▲: Writing 1 to TPBAnRDT

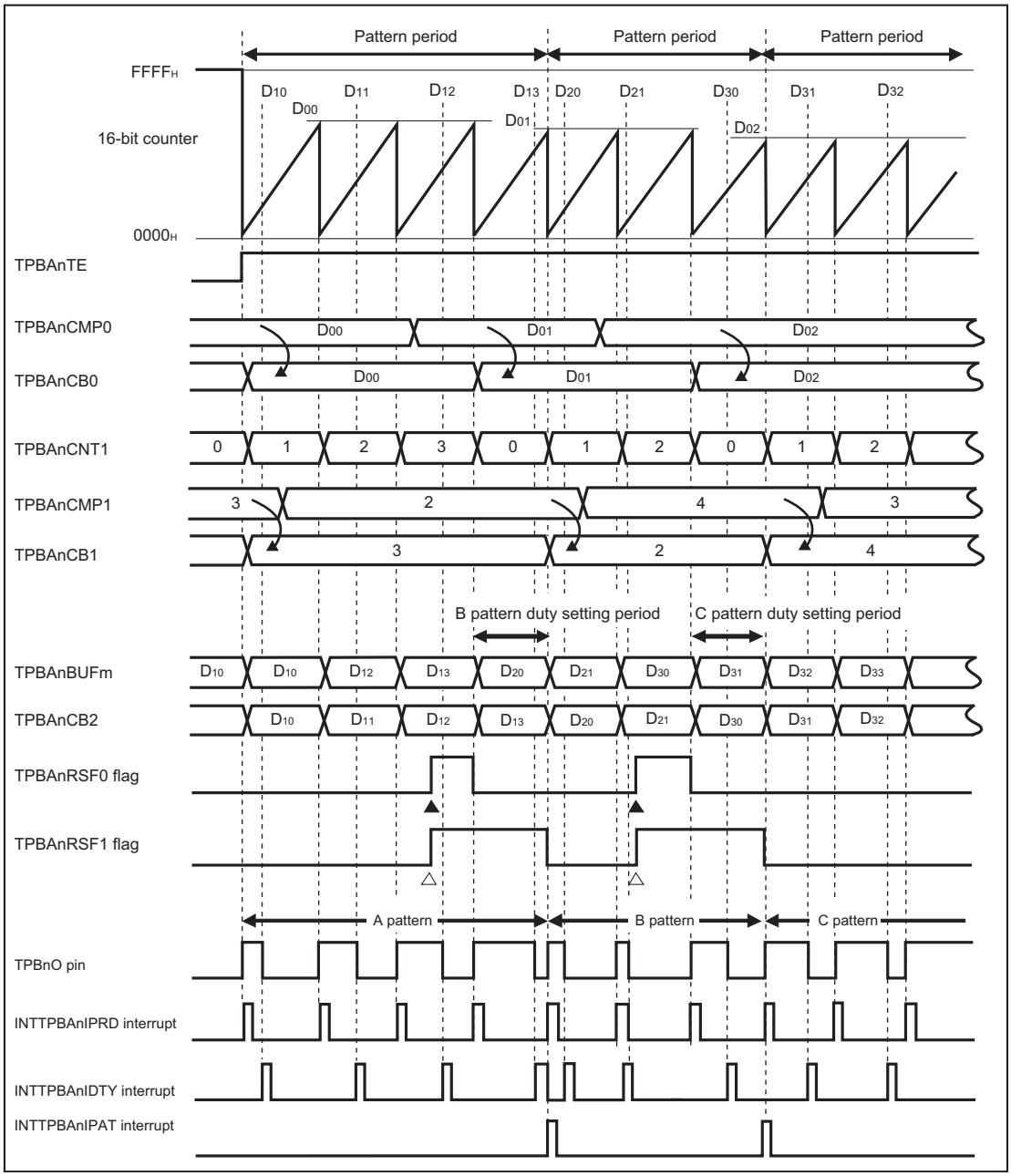


Figure 16-5 Example of Basic Timing (2/2)

When a period-matched detection interrupt is used as a trigger of the TPBAnCMP0 and TPBAnTOL reload timing (TPBAnIRDM.TPBAnRDM0 = 1 and TPBAnTOL = 0)

Note 1. PWM duty cycle = (TPBAnBUFm) / (TPBAnCMP0 set value + 1)
PWM period = (TPBAnCMP0 set value + 1) × (count clock period)

Note 2. ▲ Writing 1 to TPBAnRDT0

Note 3. Δ: Writing 1 to TPBAnRDT1

Section 17 OS Timer (OSTM)

This section contains a generic description of the OS timer.

17.1 OSTM Features

Instances This product has the following number of instances of the OS timer.

Table 17-1 Instances of OS timer

OS Timer	
Instances	2
Name	OSTMn

Instances index n Throughout this section, the individual instances of the OS timer are identified by the index "n" (n = 0, 1), for example OSTMnTO for the OS timer n output register.

Register address All OS timer register addresses are given as address offsets from the individual base addresses <OSTMn_base0>.or <OSTMn_base1>.
The <OSTMn_base> addresses of each OSTMn are listed in the following table.

Table 17-2 Register Base Addresses

OSTMn	<OSTMn_base0> Address	<OSTMn_base1> Address
OSTM0	FF80 0000_H	FFFF $C000_H$
OSTM1	FF80 1000_H	FFFF $C100_H$

Clock supply All OS timers provide one clock input.

Table 17-3 OSTM Clock Supply

OSTMn	OSTMn Clock	Connected to
OSTM0	PCLK	Clock controller
OSTM1		

Interrupts The OS timers can generate the following interrupt requests.

Table 17-4 OSTMn Interrupt Requests

OSTMn Signals	Function	Connected to
OSTM0TINT	OSTM0 interrupt	Interrupt controller INTOSTM0
OSTM1TINT	OSTM1 interrupt	Interrupt controller INTOSTM1

I/O signals The I/O signals of the OS timers are listed in the following table.

Table 17-5 OSTMn I/O Signals

OSTNn Signals	Function	Connected to
OSTM0TTOUT	OS timer output	OSTM0O
OSTM1TTOUT		OSTM1O

Table 17-6 OSTMn Signal Processing

OSTNn Signals	Function	Connected to
OSTMnTCKE	Enabling the counter clock	The clock signal selected by the setting of the IC0CKSELn register
OSTMnTSST	Synchronous counting start trigger	PIC function

Note 1. The counter clock supplied as the OSTMnTCKE signal depends on the settings for TAUA and TAUJ in the IC0CKSELn register. For details, refer to section 17.3.1, Clock Signal to Drive Counting.

Note 2. If you wish to use the synchronous counting start trigger (OSTMnTSST), refer to section 26.4.1, Simultaneous Start Trigger Function.

17.2 Functional Overview

Features summary

The OS timer has the following features.

- Two operating modes
  - Interval timer mode
  - Free-running comparison mode
- Two output modes
  - Software control mode
  - Timer-output toggling mode
- Synchronized starting of the timers is possible

The following block diagram shows the main components of the OS timer.

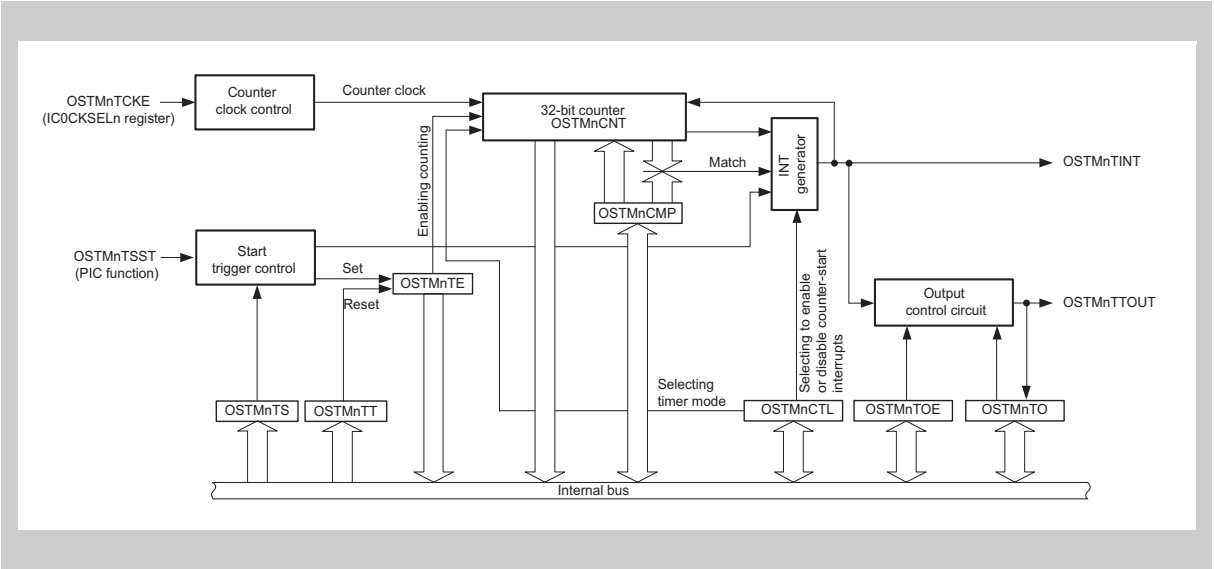


Figure 17-1 Block Diagram of the OS Timer

17.3 Functional Description

Each OS timer is a 32-bit timer/counter. The settings for operating mode specify the direction of counting (up or down) and the generation of interrupt requests.

17.3.1 Clock Signal to Drive Counting

The clock for counting by the OS timer is defined by the level of the OSTMnTCKE input signal.

- The OSTMnTCKE signal becoming 1 (at the high level) while IC0KSELn.IC0TMEN0 = 0 selects counting of cycles of PCLK.
- If IC0KSELn.IC0TMEN0 is set to 1, counting of cycles of the selected clock signal from TAUA or TAUJ is selected.

This is illustrated in the following figures.

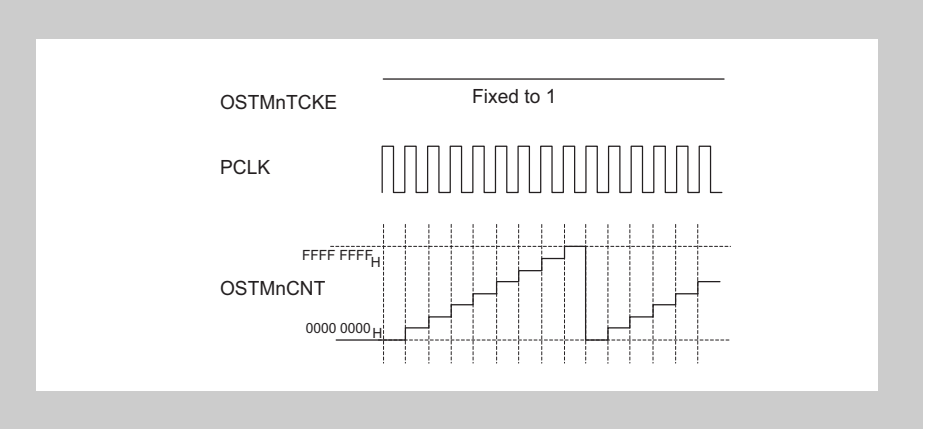


Figure 17-2 Counter Operation with the High Level on OSTMnTCKE (IC0KSELn.IC0TMEN0 = 0)

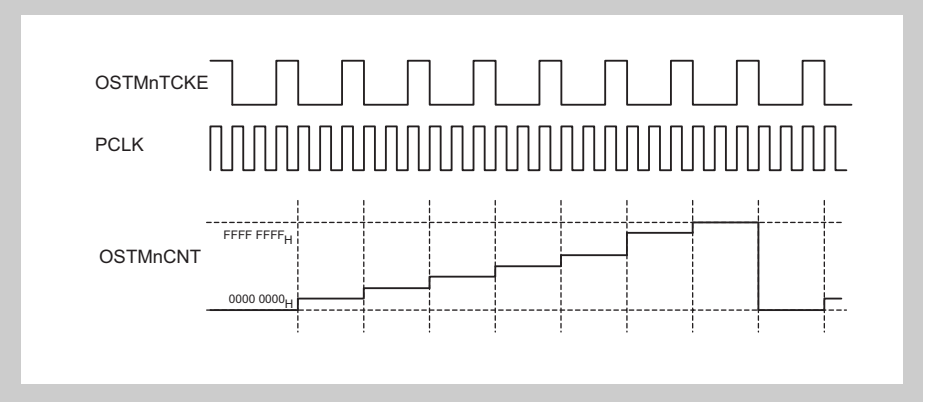


Figure 17-3 Counter Operation with OSTMnTCKE Set to the High Level in Every 3rd Cycle of PCLK (IC0KSELn.IC0TMEN0 = 1)

Each OSTM is capable of intermittent operation with an external signal (OSTMnTCKE) disabling and enabling the clock for counting.

The counter-clock-enable signal can be selected by the IC0CKSELn register from among 40 timer-output channels:16 from each of TAUA0 and TAUA1, and 4 from each of TAUJ0 and TAUJ1.

Caution 1. Select a counter-clock-enable signal for OSTMn while the operation of OSTMn is stopped (the OSTMnTE.OSTMnTE bit is 0).

Caution 2. After using the IC0CKSELn.IC0CKSELn[13:0] bits to select the counter-clock-enable signal for OSTMn, set the IC0CKSELn.IC0TMENn bit to 1.

Caution 3. If TAUAn or TAUJn is selected as the source of a counter-clock-enable signal for OSTMn (the IC0CKSELn.IC0TMENn bit is 1), do not change the operation of TAUAn or TAUJn while the OSTMn is operating (the OSTMnTE.OSTMnTE bit is 1).

[Setting procedure]

(1) Confirm that the OSTMnTE.OSTMnTE bit is 0 (OSTMn operation is stopped).

(2) Set the IC0CKSELn.IC0CKSELn[13:0] bits to select a counter-clock-enable signal for the OSTMn.

(3) Set the IC0CKSELn.IC0TMENn bit to 1.

(4) Enable OSTMn operation by setting the OSTMnTS.OSTMnTS bit to 1.

17.3.2 Output Modes

The OS timer has the following output modes. The mode is selected by the setting of the OSTMnTOE.OSTMnTOE bit.

- Software control mode (the OSTMnTOE.OSTMnTOE bit is 0)

  The level on the OSTMnTTOUT signal corresponds to the setting of the OSTMnTO.OSTMnTO bit.

- Timer-output toggling mode (the OSTMnTOE.OSTMnTOE bit is 1)

  The OSTMnTTOUT output is toggled each time an OSTMnTINT request is generated.

Both output modes are illustrated in the following figure.

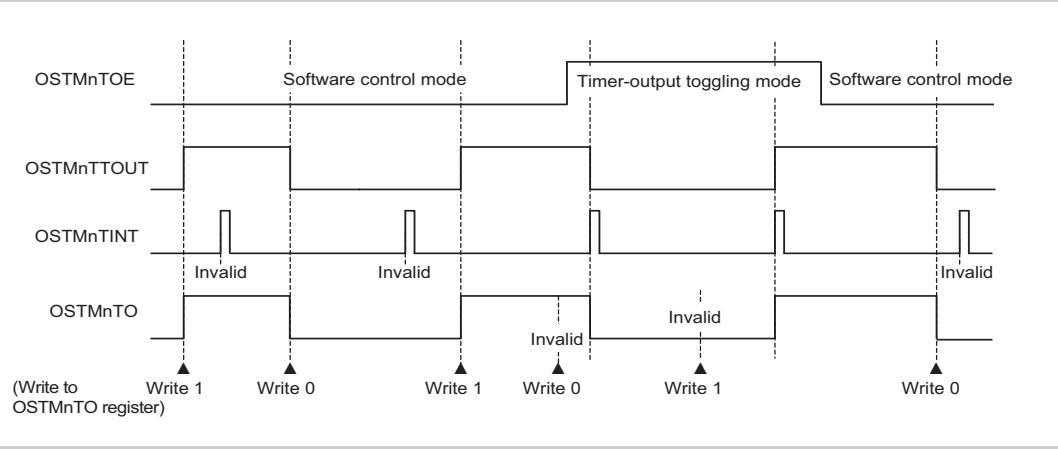


Figure 17-4 Timing Diagram of Output Modes

The above timing diagram shows the following operations.

- In software control mode, the level of the OSTMnTTOUT output changes in accord with the value set in the OSTMnTO.OSTMnTO bit.
- In timer-output toggling mode, the value of the OSTMnTO.OSTMnTO bit and level of the OSTMnTTOUT output are toggled each time an OSTMnTINT interrupt request is generated.

17.3.3 Interrupt Request Generation

An OSTMnTINT interrupt request is generated whenever the counter reaches 0000 0000_H (in interval timer mode) or matches the comparison value (in free-running comparison mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

Since OSTMnTINT triggers toggling of the OSTMnTTOUT output in timer-output toggling mode (OSTMnTOE.OSTMnTOE is 1), the setting of the OSTMnCTL.OSTMnMD0 bit also affects the output (OSTMnTTOUT).

This is illustrated in the following figure.

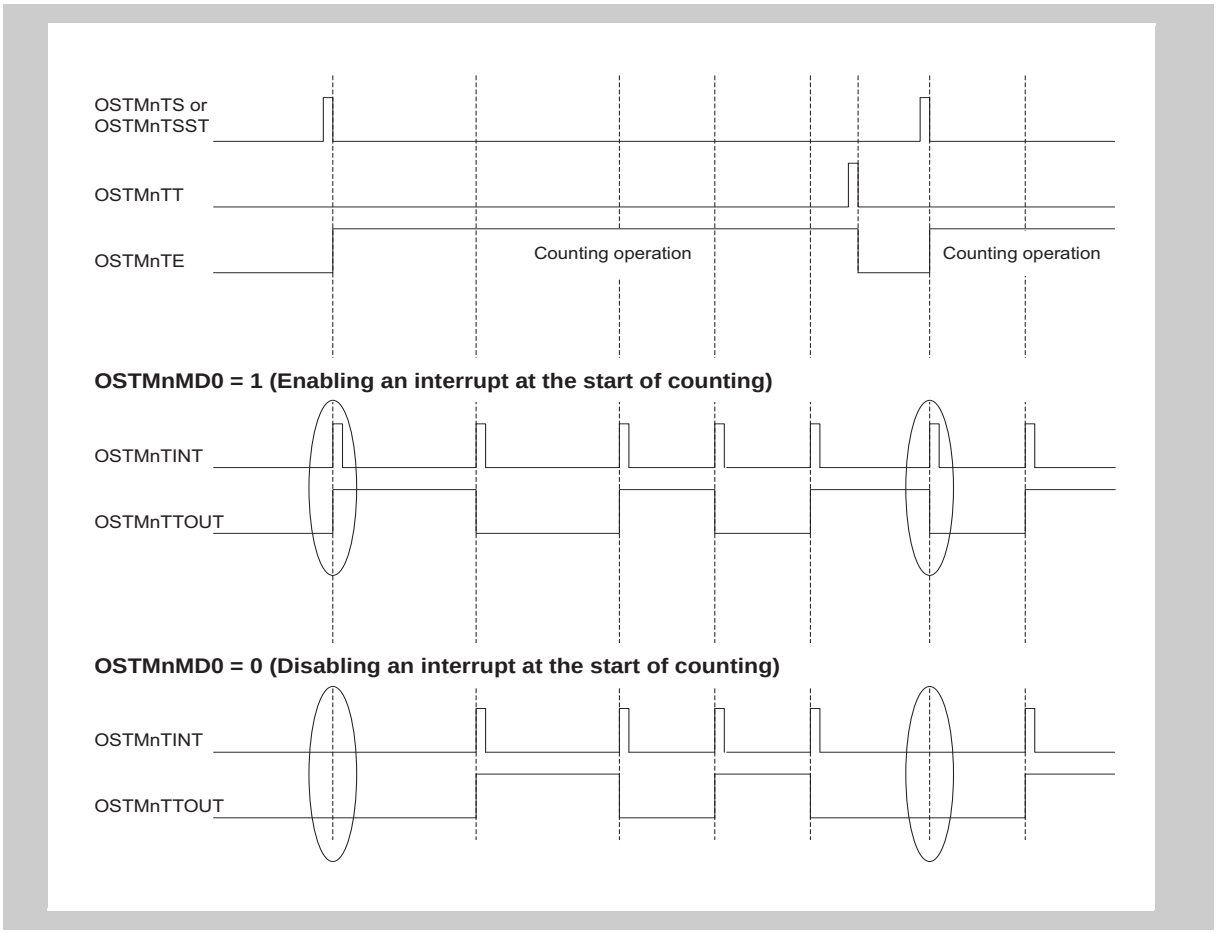


Figure 17-5 Generating an Interrupt When Counting Starts

17.3.4 Starting and Stopping the Timer

The OS timer is started and stopped as follows.

Starting the timer

The timer is started in either of the following ways:

- setting the OSTMnTS.OSTMnTS bit to 1
- placing the high level on the OSTMnTSST signal (when starting synchronous counting).

Status bit OSTMnTE.OSTMnTE is set to 1.

The counter starts to count up or down in accord with the settings for operating mode. For details, refer to section 17.3.5, Interval Timer Mode and section 17.3.6, Free-Running Comparison Mode.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the timer.

This also clears the OSTMnTE.OSTMnTE status flag.

When the counter is stopped, values in the OSTMnTO and OSTMnCNT registers and the level of the OSTMnTTOUT output are retained until further counting operations start.

Synchronous start

The OSTMnTSST signal output from the PIC module can be used to start multiple timers at the same time.

Refer to section 26, Peripheral Interconnection (PIC).

Initialization

To initialize interval timer mode and free-running comparison mode after reset release, take the following steps.

1. Set the value where the down-counter is to start counting or the value for comparison in the OSTMnCMP register.
2. In case of output (OSTMnTTOUT) toggling:
   – initialize the OSTMnTO register during operation in software control mode (OSTMnTOE.OSTMnTOE = 0) and
   – select timer-output toggling (OSTMnTOE.OSTMnTOE = 1).
3. Set the OSTMnCTL.OSTMnMD1 bit to select interval timer mode or free-running comparison mode.
4. Select the interrupt mode for the start of counting (OSTMnCTL.OSTMnMD0 = 1 or 0).

17.3.5 Interval Timer Mode

Select the interval timer mode when an OS timer is to be used as a reference timer for generating interrupt requests at a fixed interval.

(1) Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMnTINT interrupt request is generated when the counter underflows (reaches 0000 0000_H).

Select interval timer mode by setting OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next 0000 0000_H is reached. Then the counter continues with the new value.

Periods of OSTMnTINT and OSTMnTTOUT output

The periods of OSTMnTINT and OSTMnTTOUT output are as follows.

- OSTMnTINT generation period = counter-clock period × (OSTMnCMP + 1)
- OSTMnTTOUT output period = OSTMnTINT generation period × 2

The following figure shows the basic operation of the OS timer in interval timer mode with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT output in timer-output toggling mode (OSTMnTOE.OSTMnTOE = 1).

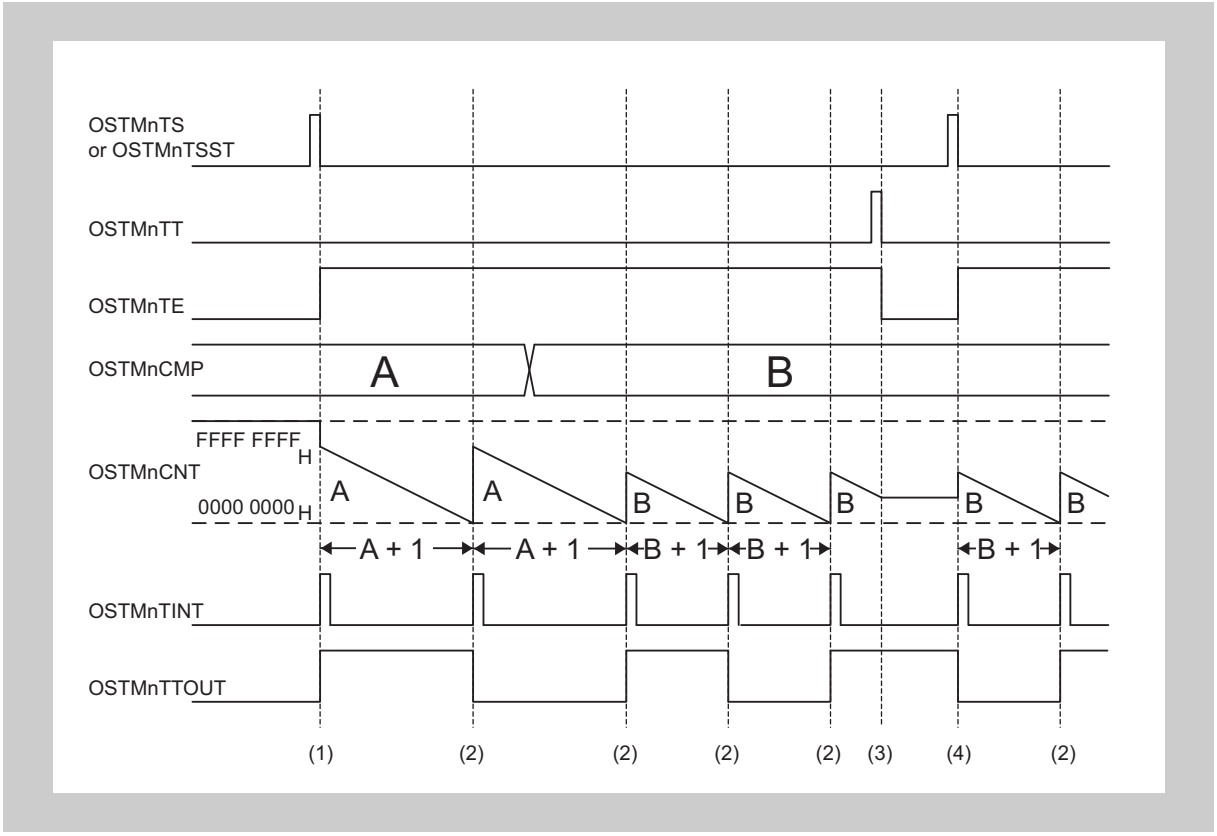


Figure 17-6 Timing Diagram of OS Timer in Interval Timer Mode

The above timing diagram shows the following operations.

1. The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST is high (if a synchronous start trigger is in use). The counter starts counting-down from the value of OSTMnCMP.
   If OSTMnCTL.OSTMnMD0 is 1, OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT output is toggled. The OSTMnCNT register contains the current value as the counter.
2. When the counter reaches 0000 0000H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output is toggled. The counter loads the new initial value from OSTMnCMP and continues counting down.
3. When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
4. When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST is high if the synchronous start trigger is in use), the counter loads the new initial value from OSTMnCMP and continues counting down.

Forced restart The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 or by setting the OSTMnTSST signal at the high level (if the synchronous start trigger is in use) during counting.

The counter loads the initial value from the OSTMnCMP register and continues counting down.

The following figure shows a forced restart of the OS timer in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT output in timer-output toggling mode (OSTMnTOE.OSTMnTOE = 1).

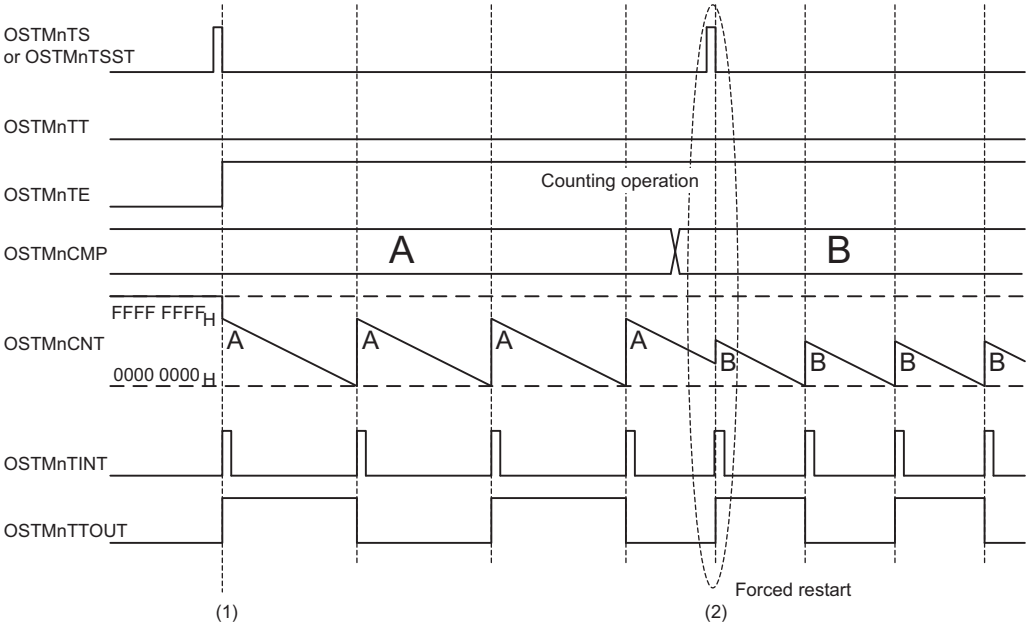


Figure 17-7 Timing Diagram for a Forced Restart in Interval Timer Mode

Operations shown in the above timing diagram are as follows.

1. The counter is started and stopped as shown in and described under *figure17-6 Timing Diagram of OS Timer in Interval Timer Mode*.
2. Writing OSTMnTS.OSTMnTS = 1 or placing the high level on OSTMnTSST (if the synchronous start trigger is in use) forcibly restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1). The counter immediately loads the start value from the OSTMnCMP register and continues counting down. When OSTMnCTL.OSTMnMD0 = 1, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output is toggled.

(2) Operation when OSTMnCMP = 0000 0000$_H$

When OSTMnCMP = 0000 0000$_H$, the OS timer behaves as follows.

- When PCLK is selected as the counter clock, the OSTMnTINT interrupt request signal stays fixed to the high level from the start of counting to the end of counting. Accordingly, output of the OSTMnTINT interrupt is not possible when OSTMnCMP = 0000 0000$_H$. However, the timer output (OSTMnTTOUT) is available in this situation. When the timer output (OSTMnTTOUT) is in timer-output toggling mode, the output is toggled every time a cycle of the clock is counted.

The following figure shows operations of the OS timer when OSTMnCMP = 0000 0000$_H$, counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1), and output on OSTMnTTOUT is in timer-output toggling mode (OSTMnTOE.OSTMnTOE = 1).

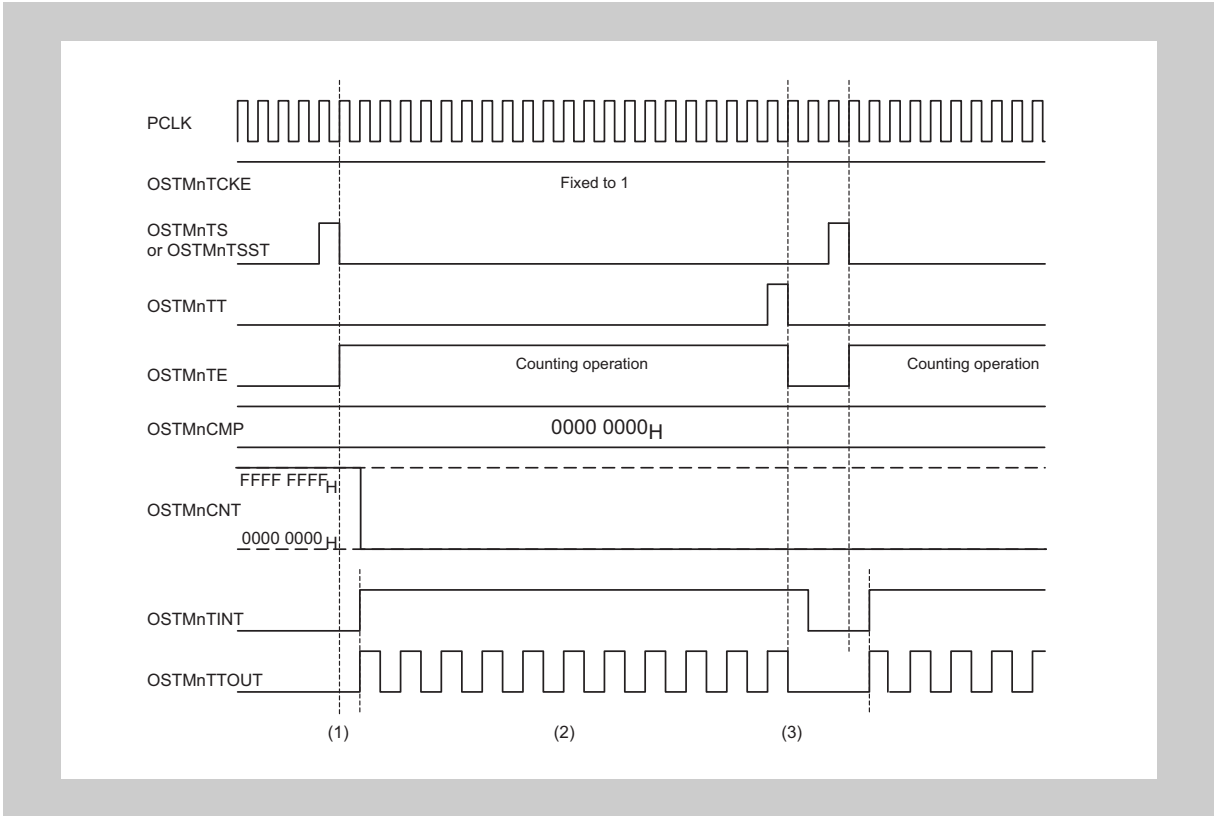


Figure 17-8 Timing Diagram for Operations When OSTMnCMP = 0000 0000$_H$ in Interval Timer Mode (PCLK Is Selected as the Counter Clock, i.e. OSTMnTCKE Is High)

The timing diagram above shows the following.

1. The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000$_H$ is retained in OSTMnCMP.
2. Toggling of the OSTMnTTOUT output starts on generation of the OSTMnTINT interrupt request (OSTMnTINT is fixed to the high level because PCLK is selected as the counter clock in the figure).
3. After the counter stops, the OSTMnTINT interrupt request signal is deasserted and the level of the OSTMnTTOUT output is retained.

17.3.6 Free-Running Comparison Mode

(1) Basic Operation in Free-Running Comparison Mode

In free-running comparison mode, the counter counts up from 0000 0000_H to FFFF $FFFF_H$. An OSTMnTINT interrupt request is output when the current value of the counter matches the value of the OSTMnCMP register. The free-running comparison mode is selected by setting the OSTMnCTL.OSTMnMD1 bit to 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of the OS timer in free-running comparison mode with the start of counting enabled (OSTMnTS.OSTMnTS = 1) and OSTMnTTOUT output in timer-output toggling mode (OSTMnTOE.OSTMnTOE = 1).

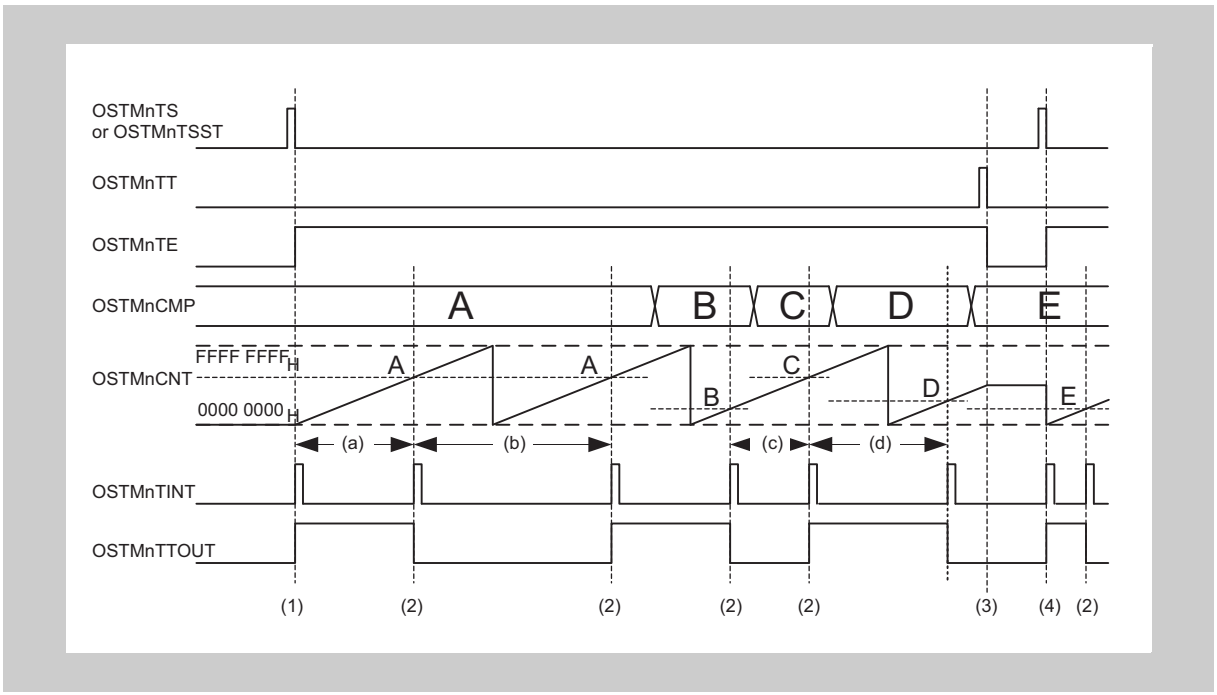


Figure 17-9 Timing Diagram of OS Timer in Free-Running Comparison Mode

The timing diagram above shows the following.

1. The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST is high if the synchronous start trigger is in use. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000_H to FFFF $FFFF_H$. The OSTMnCNT register is the counter, so it contains the current value.
2. When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output is toggled.
3. When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
4. Counting by the counter restarts from 0000 0000_H when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST is high if the synchronous start trigger is in use.

The OSTMnTINT interrupt-generation period depends on the timing for updating of the OSTMnCMP register and the counter value at the time. For the formula to calculate the period for generation of the OSTMnTINT interrupt, refer to the table below.

Table 17-7 OSTMnTINT Generation Timing

Old Value for Comparison	New Value for Comparison	Counter Value at the Time of Rewriting	Period of OSTMnTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × counter clock period	(a)
A	A	No rewriting	(FFFF FFFF$_H$ + 1) × counter clock period	(b)
B	C > B	B < counter value < C	(C - B) × counter clock period	(c)
C	D < C	Counter value > D, C	(FFFF FFFF$_H$ - C + D + 1) × counter clock period	(d)

Note A, B, C and D indicate the values of OSTMnCMP over the intervals thus labeled in *figure17-9 Timing Diagram of OS Timer in Free-Running Comparison Mode*.

Forced restart Forced restarting does not proceed during counting even if the OSTMnTS.OSTMnTS bit is set, or if OSTMnTSST is high when the synchronous start trigger is in use.

The counter ignores the attempted setting and continues counting.

(2) Operation when OSTMnCMP = 0000 0000_H

The following figure shows the operation of the OS timer when OSTMnCMP = 0000 0000_H, counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1), and OSTMnTTOUT is in timer-output toggling mode (OSTMnTOE.OSTMnTOE = 1).

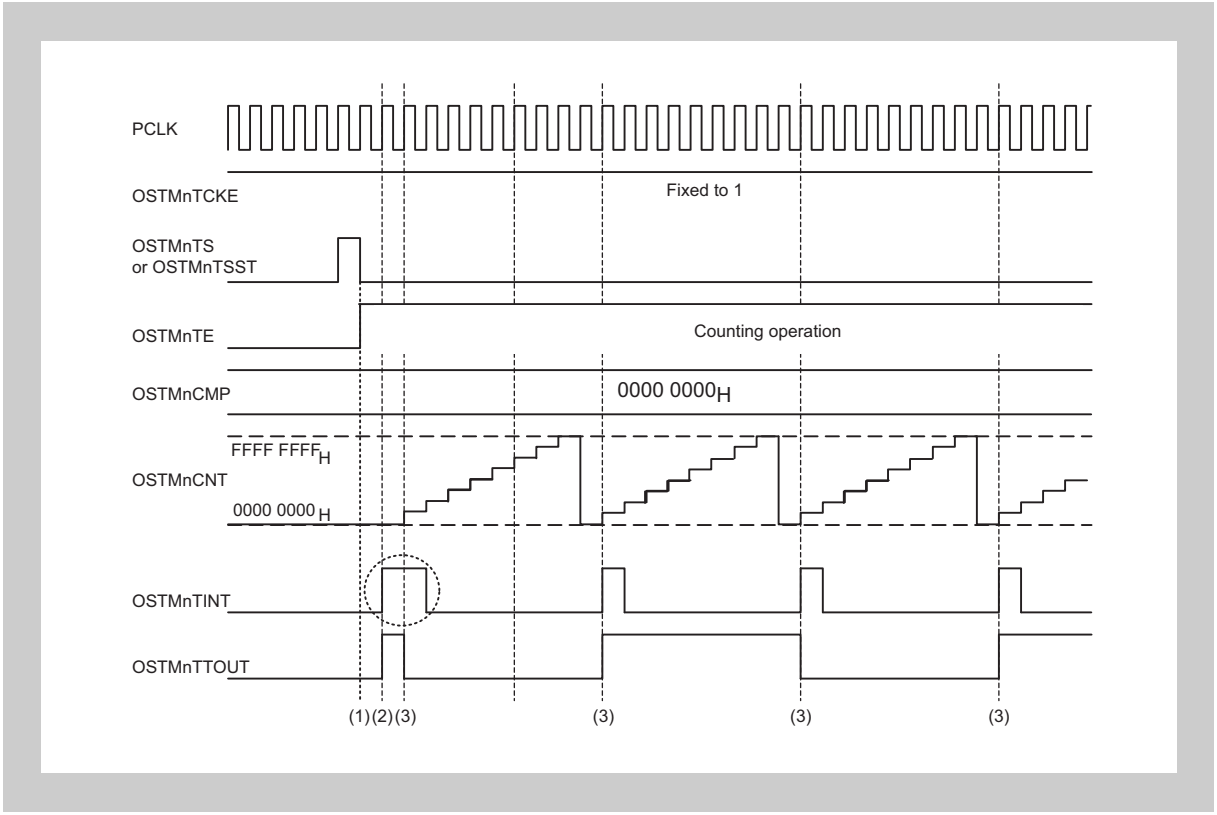


Figure 17-10 Timing Diagram when OSTMnCMP = 0000 0000_H in Free-Running Comparison Mode with PCLK Selected as the Counter Clock (OSTMnTCKE Is High)

The above timing diagram shows the following operations.

1. Once the counter starts, it counts up from 0000 0000_H to FFFF $FFFF_H$.
2. An OSTMnTINT interrupt request is generated when counting starts (and the OSTMnTTOUT output is toggled).
3. The OSTMnTTOUT output is toggled. A comparison interrupt is only generated if the current value of the counter matches that in the OSTMnCMP register.

Note When PCLK is selected as the counter clock (OSTMnTCKE is high) and the value for comparison is 0 (OSTMnCMP = 0000 0000_H), OSTMnTINT initially stays at the high level over two cycles of PCLK, and one interrupt request is generated for every rising edge of OSTMnTINT.

17.4 Registers

This section describes all registers of the OS timer.

17.4.1 OS Timer Registers Overview

The OS timer is controlled and operated by the following registers.

Table 17-8 List of OS Timer Registers

Register Name	Symbol	Address
OSTM compare register	OSTMnCMP	<OSTMn_base1>
OSTM counter register	OSTMnCNT	<OSTMn_base1> + 4_H
OSTM output register	OSTMnTO	<OSTMn_base1> + 8_H
OSTM output enable register	OSTMnTOE	<OSTMn_base1> + C_H
OSTM count enable status register	OSTMnTE	<OSTMn_base1> + 10_H
OSTM count start trigger register	OSTMnTS	<OSTMn_base1> + 14_H
OSTM count stop trigger register	OSTMnTT	<OSTMn_base1> + 18_H
OSTM control register	OSTMnCTL	<OSTMn_base0> + 20_H
OSTM input clock select function register 0	IC0CKSEL0	$FF83F000_H$
OSTM input clock select function register 1	IC0CKSEL1	$FF83F004_H$

17.4.2 OS Timer Registers in Detail

(1) OSTMnCMP - OSTM Compare Register

Depending on the mode of operation, this register holds the start value for the down-counter or the value for comparison with that of the counter.

Access This register is readable/writable in 32-bit units.

Address <OSTMn_base1>

Initial value 0000 0000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTMnCMP[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTMnCMP[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-9 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP[31:0]	• In interval timer mode: start value of the down-counter • In free-running comparison mode: value for comparison

(2) OSTMnCNT - OSTM Counter Register

This register indicates the counter value of the timer.

Access This register is readable in 32-bit units.

Address <OSTMn_base1> + 4_H

Initial value The initial value depends on the operating mode of the OS timer. Refer to table 17-11, Correspondence between Operating Mode, Counting Direction and Initial Value. This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTMnCNT[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTMnCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17-10 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT[31:0]	32-bit counter value

The following table shows the correspondence between operating mode, counting direction and initial value. The initial value is the value read from the counter after a change to the operating mode.

Table 17-11 Correspondence between Operating Mode, Counting Direction and Initial Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Initial value
Interval timer mode	0*	Down	FFFF $FFFF_H$
Free-running comparison mode	1	Up	0000 0000_H

Note * Value after reset

(3) OSTMnTO - OSTM Output Register

This register is for specifying or reading of the level of the OSTMnTTOUT output signal.

Access This register is readable and writable in 8-bit units. Writing can only proceed when the software control mode is selected (OSTMnTOE.OSTMnTOE = 0).

Address <OSTMn_base1> + 8_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTO
R	R	R	R	R	R	R	R/W

Table 17-12 OSTMnTO Register Contents

Bit Position	Bit Name	Function
0	OSTMnTO	For specifying or reading the level of the OSTMnTTOUT output signal 0: Low level 1: High level

(4) OSTMnTOE - OSTM Output Enable Register

This register specifies OSTMnTTOUT output mode.

Access This register is readable/writable in 8-bit units.

Address <OSTMn_base1> + C_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMn TOE
R	R	R	R	R	R	R	R/W

Table 17-13 OSTMnTOE Register Contents

Bit Position	Bit Name	Function
0	OSTMnTOE	This bit specifies the OSTMnTTOUT output mode. 0: Software control mode: The level corresponding to the setting of OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Timer-output toggling mode: OSTMnTTOUT output is toggled whenever an OSTMnTINT interrupt request is generated.

(5) OSTMnTE - OSTM Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access This register is readable in 8-bit units.

Address <OSTMn_base1> + 10_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTE
R	R	R	R	R	R	R	R

Table 17-14 OSTMnTE Register Contents

Bit Position	Bit Name	Function
0	OSTMnTE	This bit indicates whether the counter is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1, or OSTMnTSST being at the high level if synchronous starting is in use. Setting OSTMnTT.OSTMnTT to 1 re-sets this bit to 0.

Note When OSTMnTE = 0, the counter retains its value.

If the counter is restarted, it

- restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- restarts counting up from the counter value 0000 0000_H if it is in free-running comparison mode.

(6) OSTMnTS - OSTM Count Start Trigger Register

This register starts the counter.

Access This register is writable in 8-bit units. It is always read as 00_H.

Address <OSTMn_base1> + 14_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTS
R	R	R	R	R	R	R	W

Table 17-15 OSTMnTS Register Contents

Bit Position	Bit Name	Function
0	OSTMnTS	This bit starts the counter. 0: No function 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. • In free-running comparison mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

(7) OSTMnTT - OSTM Count Stop Trigger Register

This register stops the counter.

Access This register is writable in 8-bit units. It is always read as 00_H.

Address <OSTMn_base1> + 18_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTT
R	R	R	R	R	R	R	W

Table 17-16 OSTMnTT Register Contents

Bit Position	Bit Name	Function
0	OSTMnTT	Stops the counter. 0: No function 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit. When the counter has stopped, this bit returns to 0.

(8) OSTMnCTL - OSTM Control Register

This register specifies the operating mode for the counter and controls the generation of OSTMnTINT interrupt requests when counting starts.

Access This register is readable/writable in 8-bit units. Writing to this register is only possible if the counter is disabled (OSTMnTE.OSTMnTE = 0).

Address <OSTMn_base0> + 20_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OSTMnMD1	OSTMnMD0
R	R	R	R	R	R	R/W	R/W

Table 17-17 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
1	OSTMnMD1	Specifies the operating mode for the counter. 0: Interval timer mode 1: Free-running comparison mode
0	OSTMnMD0	Controls the generation of OSTMnTINT interrupt requests at the start of counting. 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled.

(9) IC0CKSEL0 - OSTM Input Clock Select Function Register 0

This register is used to select the clock-enable signal for the counter clock of OSTM0.

Access This register is readable/writable in 16-bit units.

Address $FF83F000_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC0TMEN0	0	IC0TMSEL01	IC0TMSEL00	IC0CKSEL031	IC0CKSEL030	IC0CKSEL021	IC0CKSEL020	IC0CKSEL013	IC0CKSEL012	IC0CKSEL011	IC0CKSEL010	IC0CKSEL003	IC0CKSEL002	IC0CKSEL001	IC0CKSEL000
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-18 IC0CKSEL0 Register Contents

Bit Position	Bit Name	Function
15	IC0TMEN0	Selects output of the counter-clock-enable signal. **IC0TMEN0** / **Description** 0: Input the high level as the counter-clock-enable signal for the OSTM0 (PCLK is selected as the counter clock). 1: Input the counter-clock-enable signal selected by IC0TMSEL00 and IC0TMSEL01 to OSTM0.
13,12	IC0TMSEL00, IC0TMSEL01	According to the setting, these bits select the timer signal indicated below as the counter-clock-enable signal (from among the counter-clock-enable signals selected by bits 11 to 0). **IC0TMSEL01** / **IC0TMSEL00** / **Description** 0 / 0 / TAUA0 0 / 1 / TAUA1 1 / 0 / TAUJ0 1 / 1 / TAUJ1
11,10	IC0CKSEL030, IC0CKSEL031	Select a counter-clock-enable signal for input to OSTM0. **IC0CKSEL031** / **IC0CKSEL030** / **Description** 0 / 0 / TAUJ1 timer channel 0 0 / 1 / TAUJ1 timer channel 1 1 / 0 / TAUJ1 timer channel 2 1 / 1 / TAUJ1 timer channel 3
9,8	IC0CKSEL020, IC0CKSEL021	Select a counter-clock-enable signal for input to OSTM0. **IC0CKSEL021** / **IC0CKSEL020** / **Description** 0 / 0 / TAUJ0 timer channel 0 0 / 1 / TAUJ0 timer channel 1 1 / 0 / TAUJ0 timer channel 2 1 / 1 / TAUJ0 timer channel 3
7 to 4 3 to 0	IC0CKSEL010 to IC0CKSEL013 IC0CKSEL000 to IC0CKSEL003	Select a counter-clock-enable signal for input to OSTM0. x = 0, 1 **IC0CKSEL0x3** / **IC0CKSEL0x2** / **IC0CKSEL0x1** / **IC0CKSEL0x0** / **Description** 0 / 0 / 0 / 0 / TAUAx timer channel 0 0 / 0 / 0 / 1 / TAUAx timer channel 1 … / … / … / … / … 1 / 1 / 1 / 0 / TAUAx timer channel 14 1 / 1 / 1 / 1 / TAUAx timer channel 15

Caution 1. Select a counter-clock-enable signal for OSTMn while the operation of OSTMn is stopped (the OSTMnTE.OSTMnTE bit is 0).

Caution 2. After using the IC0CKSELn.IC0CKSELn[13:0] bits to select the counter-clock-enable signal for OSTMn, set the IC0CKSELn.IC0TMENn bit to 1.

Caution 3. If TAUAn or TAUJn is selected as the source of a counter-clock-enable signal for OSTMn (the IC0CKSELn.IC0TMENn bit is 1), do not change the operation of TAUAn or TAUJn while the OSTMn is operating (the OSTMnTE.OSTMnTE bit is 1).

[Setting procedure]

(1) Confirm that the OSTMnTE.OSTMnTE bit is 0 (OSTMn operation is stopped).

(2) Set the IC0CKSELn.IC0CKSELn[13:0] bits to select a counter-clock-enable signal for the OSTMn.

(3) Set the IC0CKSELn.IC0TMENn bit to 1.

(4) Enable OSTMn operation by setting the OSTMnTS.OSTMnTS bit to 1.

(10) IC0CKSEL1 - OSTM Input Clock Select Function Register 1

This register is used to select the clock-enable signal for the counter clock of OSTM1.

Access This register is readable/writable in 16-bit units.

Address $FF83F004_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC0TMEN1	0	IC0TMSEL11	IC0TMSEL10	IC0CKSEL131	IC0CKSEL130	IC0CKSEL121	IC0CKSEL120	IC0CKSEL113	IC0CKSEL112	IC0CKSEL111	IC0CKSEL110	IC0CKSEL103	IC0CKSEL102	IC0CKSEL101	IC0CKSEL100
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-19 IC0CKSEL1 Register Contents

Bit Position	Bit Name	Function
15	IC0TMEN1	Selects output of the counter-clock-enable signal. **IC0TMEN1** / **Description** 0: Input the high level as the counter-clock-enable signal for the OSTM1. 1: Input the counter-clock-enable signal selected by IC0TMSEL10 and IC0TMSEL11 to OSTM1.
13,12	IC0TMSEL10, IC0TMSEL11	According to the setting, these bits select the timer signal indicated below as the counter-clock-enable signal (from among the counter-clock-enable signals selected by bits 11 to 0) . **IC0TMSEL11** / **IC0TMSEL10** / **Description** 0 / 0 / TAUA0 0 / 1 / TAUA1 1 / 0 / TAUJ0 1 / 1 / TAUJ1
11,10	IC0CKSEL130, IC0CKSEL131	Select a counter-clock-enable signal for input to OSTM1. **IC0CKSEL131** / **IC0CKSEL130** / **Description** 0 / 0 / TAUJ1 timer channel 0 0 / 1 / TAUJ1 timer channel 1 1 / 0 / TAUJ1 timer channel 2 1 / 1 / TAUJ1 timer channel 3
9,8	IC0CKSEL120, IC0CKSEL121	Select a counter-clock-enable signal for input to OSTM1. **IC0CKSEL121** / **IC0CKSEL120** / **Description** 0 / 0 / TAUJ0 timer channel 0 0 / 1 / TAUJ0 timer channel 1 1 / 0 / TAUJ0 timer channel 2 1 / 1 / TAUJ0 timer channel 3
7 to 4 3 to 0	IC0CKSEL110 to IC0CKSEL113 IC0CKSEL100 to IC0CKSEL103	Select a counter-clock-enable signal for input to OSTM1. x = 0, 1 **IC0CKSEL1x3** / **IC0CKSEL1x2** / **IC0CKSEL1x1** / **IC0CKSEL1x0** / **Description** 0 / 0 / 0 / 0 / TAUAx timer channel 0 0 / 0 / 0 / 1 / TAUAx timer channel 1 … / … / … / … / … 1 / 1 / 1 / 0 / TAUAx timer channel 14 1 / 1 / 1 / 1 / TAUAx timer channel 15

Caution 1. Select a counter-clock-enable signal for OSTMn while the operation of OSTMn is stopped (the OSTMnTE.OSTMnTE bit is 0).

Caution 2. After using the IC0CKSELn.IC0CKSELn[13:0] bits to select the counter-clock-enable signal for OSTMn, set the IC0CKSELn.IC0TMENn bit to 1.

Caution 3. If TAUAn or TAUJn is selected as the source of a counter-clock-enable signal for OSTMn (the IC0CKSELn.IC0TMENn bit is 1), do not change the operation of TAUAn or TAUJn while the OSTMn is operating (the OSTMnTE.OSTMnTE bit is 1).

[Setting procedure]

(1) Confirm that the OSTMnTE.OSTMnTE bit is 0 (OSTMn operation is stopped).

(2) Set the IC0CKSELn.IC0CKSELn[13:0] bits to select a counter-clock-enable signal for the OSTMn.

(3) Set the IC0CKSELn.IC0TMENn bit to 1.

(4) Enable OSTMn operation by setting the OSTMnTS.OSTMnTS bit to 1.

Section 18 Encoder Timer (ENCA)

18.1 ENCA Features

Instances

This product has two encoder timers.

Table 18-1 Instances of ENCA

Encoder Timer	
Instances	2
Name	ENCA0, ENCA1

Instances index n

Throughout this section, both of the individual encoder timers are indicated by the index "n" (where n = 0 and 1). For example, ENCAnCTL indicates both ENCAn control registers.

Meaning of both edges

In this section, "both edges" refers to both rising and falling edges.

Register addresses

All ENCAn register addresses are given as address offsets to the individual base address <ENCAn_base>.
The base address <ENCAn_base> of each ENCAn is listed in the following table:

Table 18-2 Register Base Addresses <ENCAn_base>

ENCAn Instance	<ENCAn_base0> Address	<ENCAn_base1> Address
ENCA0	FF81 9000_H	FFFF $D800_H$
ENCA1	FF81 $A000_H$	FFFF $D900_H$

Clock supply

The following clock input is provided to each ENCAn. That is, each ENCAn is connected to the PCLK.

Table 18-3 ENCAn Clock Supply

ENCAn Instance	Clock Supply
ENCA0, ENCA1	PCLK

Interrupt requests

Interrupt requests from ENCAn are listed in the following table:

Table 18-4 List of ENCAn Interrupt Requests

ENCAn Signals	Function	Connected to
INTENCAnIOV	Overflow interrupt	Interrupt controller, DMA, DTS
INTENCAnIUD	Underflow interrupt	Interrupt controller, DMA, DTS
INTENCAnI0	Compare match 0 or capture 0 interrupt	Interrupt controller, DMA, DTS
INTENCAnI1	Compare match 1 or capture 1 interrupt	Interrupt controller, DMA, DTS
INTENCAnIEC	Interrupt to indicate clearing due to clearing input from the encoder	Interrupt controller, DMA, DTS

Input signals

Input signals of ENCAn are listed in the following table:

Table 18-5 List of ENCAn Input Signals

ENCAn Signals	Function	Connected to
ENCAnI0	ENCAn capture trigger input 0	Port
ENCAnI1	ENCAn capture trigger input 1	Port
ENCAnE0	ENCAn encoder input 0	Port
ENCAnE1	ENCAn encoder input 1	Port
ENCAnEC	ENCAn encoder clearing input	Port
ENCAnTSST	ENCAn simultaneous start trigger input	PIC module (input)

18.2 Functional Overview

Features summary

- Generates a counter control signal from the encoder input signal and executes counting in synchronization with PCLK.
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture compare registers that can be set separately for capture operation and for compare operation
- Interrupt masking function for masking the output of interrupt signals as a result of a match in judgment by comparison
- Function for loading the value of the capture-compare register to the counter when the latter underflows
- An input signal from an encoder can be applied as the clearing condition for the timer counter.
- Edge or level detection is selectable for the encoder input signal that provides the clearing condition for the timer counter.
- Five interrupt outputs: 2 capture-compare interrupts, 1 counter-clear interrupt, 1 overflow interrupt, and 1 underflow interrupt.

18.2.1 Block Diagram

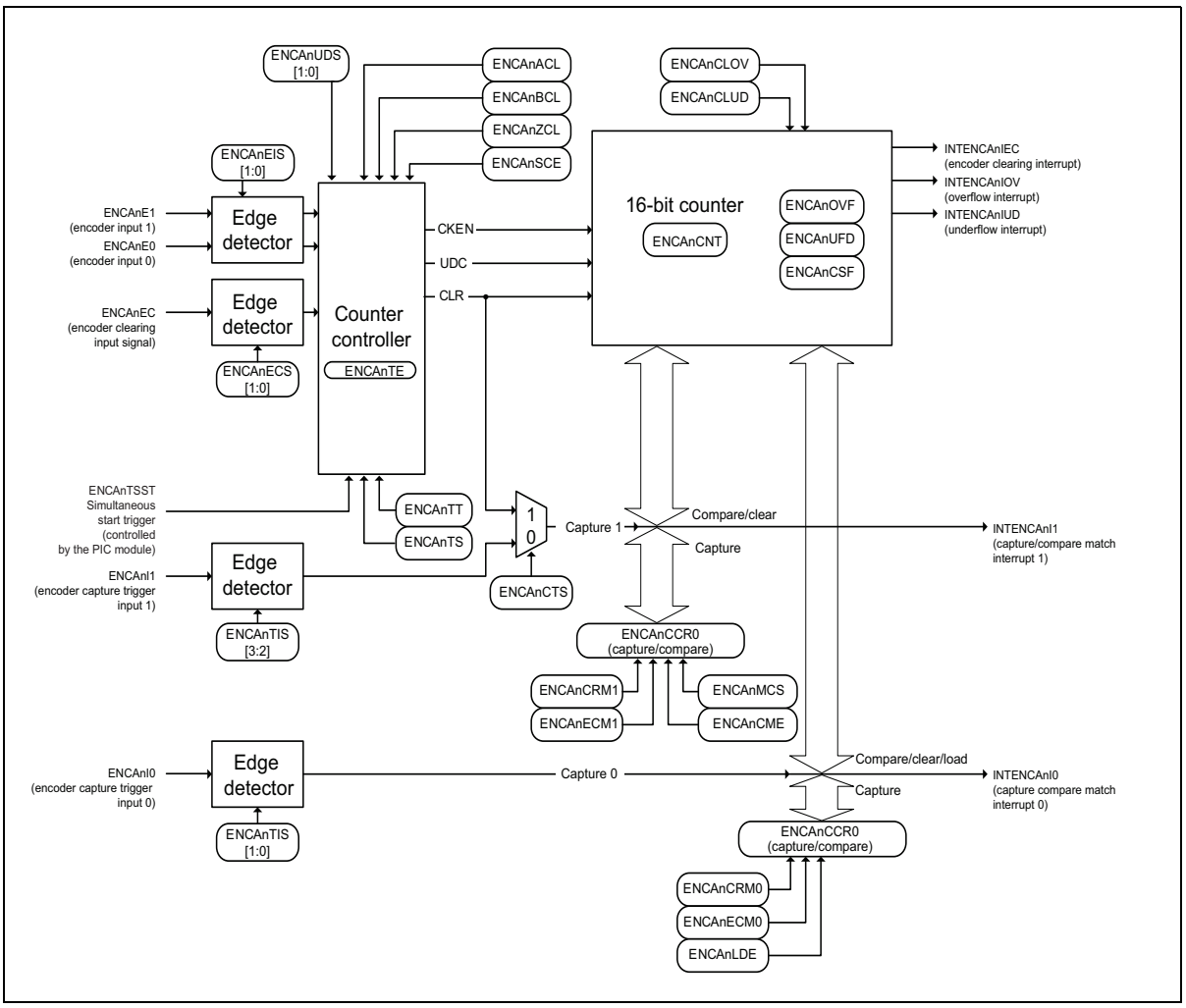


Figure 18-1 Block Diagram of Encoder Timer

18.3 ENCA Control Registers

The ENCAn is controlled and operated by the following registers:

Table 18-6 ENCAn Register Overview

Register Function	Name	Address
ENCAn capture/compare register 0	ENCAnCCR0	<ENCAn_base1>
ENCAn capture/compare register 1	ENCAnCCR1	<ENCAn_base1> + 04_H
ENCAn counter register	ENCAnCNT	<ENCAn_base1> + 08_H
ENCAn status flag register	ENCAnFLG	<ENCAn_base1> + $0C_H$
ENCAn status flag clear register	ENCAnFGC	<ENCAn_base1> + 10_H
ENCAn timer enable status register	ENCAnTE	<ENCAn_base1> + 14_H
ENCAn timer start trigger register	ENCAnTS	<ENCAn_base1> + 18_H
ENCAn timer stop trigger register	ENCAnTT	<ENCAn_base1> + $1C_H$
ENCAn I/O control register 0	ENCAnIOC0	<ENCAn_base1> + 20_H
ENCAn control register	ENCAnCTL	<ENCAn_base0> + 40_H
ENCAn I/O control register 1	ENCAnIOC1	<ENCAn_base0> + 44_H

<ENCAn_base> The base address <ENCAn_base> for each ENCAn is defined in section 18.1, under "Register addresses".

(1) ENCAnCTL – ENCA Control Register

This register is a 16-bit register that controls operation of the encoder timer.

Access This register can be read/written in 16-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base0> + 40_H

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8
ENCAn CME	ENCAn MCS	0	0	0	0	ENCAn CRM1	ENCAn CRM0
R/W	R/W	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
ENCAn CTS	0	0	ENCAn LDE	ENCAn ECM1	ENCAn ECM0	ENCAn UDS[1:0]	
R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 18-7 ENCAnCTL Register Contents (1/3)

Bit Position	Bit Name	Function
15	ENCAnCME	This bit is used to enable or disable masking of compare-match interrupts when comparison is in use. 0: Disables masking. 1: Enables masking. Caution 1. The setting of the ENCAnCME bit is only valid when ENCAnCRM1 = 0. Caution 2. While the ENCAnCME bit is set to 1, setting ENCAnECM1 to 1 is prohibited.
14	ENCAnMCS	This bit is used to select the trigger for canceling masking of compare match interrupt detection when the compare function is used. 0: Masking of compare match interrupt detection is canceled when the ENCAnCCR1 register is written. 1: Masking of compare match interrupt detection is canceled when one of the following three operations is performed. - Timer counter clearing in response to the clearing input signal from the encoder. - Timer counter clearing upon compare match between ENCAnCNT and ENCAnCCR0 when ENCAnECM0 = 1 - Loading from ENCAnCCR0 to timer counter upon underflow detection when ENCAnLDE = 1 Caution: The setting of the ENCAnMCS bit is only valid when ENCAnCRM1 = 0.
9	ENCAnCRM1	This bit selects how the ENCAnCCR1 register is to be used. 0: Selects the use of ENCAnCCR1 as a comparison register. 1: Selects the use of ENCAnCCR1 as a capture register.
8	ENCAnCRM0	This bit selects how the ENCAnCCR0 register is to be used. 0: Selects the use of ENCAnCCR0 as a comparison register. 1: Selects the use of ENCAnCCR0 as a capture register.

Table 18-7 ENCAnCTL Register Contents (2/3)

Bit Position	Bit Name	Function
7	ENCAnCTS	This bit selects the trigger for capture by the ENCAnCCR1 register. 0: The ENCAnI1 input is the capture trigger. 1: The ENCAnEC input is the capture trigger. Caution: The setting of the ENCAnCTS bit is only valid when ENCAnCRM1 = 1.
4	ENCAnLDE	This bit is used to enable or disable the loading of settings of the ENCAnCCR0 register to the counter when the counter underflows. 0: Disable loading to the counter. 1. Enable loading to the counter. Caution 1. The setting of the ENCAnLDE bit is only valid when ENCAnCRM0 = 0. Caution 2. When ENCAnCRM0 = 1, the value in the ENCAnCCR0 register is not loaded to the counter when the counter underflows, regardless of the value of the ENCAnLDE bit.
3	ENCAnECM1	This bit is used to select or deselect counter-clearing upon matches between the values in the counter and in register ENCAnCCR1. 0: The counter is not cleared to 0000_H. 1: The counter is cleared to 0000_H if the next step in counting is counting down. Caution: The setting of the ENCAnECM1 bit is only valid when ENCAnCRM1 = 0.
2	ENCAnECM0	This bit is used to select or deselect counter-clearing upon matches between the values in the counter and in register ENCAnCCR0. 0: The counter is not cleared to 0000_H. 1: The counter is cleared to 0000_H if the next step in counting is counting up. Caution: The setting of the ENCAnECM0 bit is only valid when ENCAnCRM0 = 0.

Table 18-7 ENCAnCTL Register Contents (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>ENCAnUDS1,
ENCAnUDS0</td><td>Along with the inputs on the ENCAnE0 and ENCAnE1 pins, these bits control whether counting is up or down.
<table>
<tr><th>ENCAn UDS1</th><th>ENCAn UDS1</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>Upon detection of a valid edge in the ENCAnE0 input signal from the encoder, counter operation depends on the state of the ENCAnE1 input signal from the encoder.
Counting will be
• down if ENCAnE1 is "high" and
• up if ENCAnE1 is "low".</td></tr>
<tr><td>0</td><td>1</td><td>• Detection of a valid edge in the ENCAnE0 encoder input causes counting up.
• Detection of a valid edge in the ENCAnE1 encoder input causes counting down.</td></tr>
<tr><td>1</td><td>0</td><td>• A rising edge of the ENCAnE0 encoder input causes counting down.
• A falling edge of the ENCAnE0 encoder input causes counting up.
However, counting in either direction only proceeds when ENCAnE1 is low.</td></tr>
<tr><td>1</td><td>1</td><td>Both edges are detected in encoder inputs ENCAnE0 and ENCAnE1.
The combination of detected edge and input level determines the counting operation.</td></tr>
</table>
</td></tr>
</table>

(2) ENCAnIOC0 – ENCA I/O Control Register 0

This is an 8-bit register for use to control the valid edge in the input of capture triggers 0 and 1 (i.e. input signals ENCAnI0 and ENCAnI1).

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

Address <ENCAn_base1> + 20_H

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	ENCAn TIS[3:2]		ENCAn TIS[1:0]	
R	R	R	R	R/W	R/W	R/W	R/W

Table 18-8 ENCAnIOC0 Register Contents

Bit Position	Bit Name	Function
3, 2	ENCAnTIS3, ENCAnTIS2	Selects the valid edge of capture trigger 1 (ENCAnI1). **ENCAn TIS3** / **ENCAn TIS2** / **Operation** 0 / 0 / Neither edge is detected. 0 / 1 / Rising edges are detected. 1 / 0 / Falling edges are detected. 1 / 1 / Both edges are detected. Caution: Settings in these bits are only valid when ENCAnCTL.ENCAnCRM1 = 1 and ENCAnCTS = 0. The settings are invalid with any other settings of ENCAnCRM1 and ENCAnCTS.
1, 0	ENCAnTIS1, ENCAnTIS0	Selects the valid edge of capture trigger 0 (ENCAnI0). **ENCAn TIS1** / **ENCAn TIS0** / **Operation** 0 / 0 / Neither edge is detected. 0 / 1 / Rising edges are detected. 1 / 0 / Falling edges are detected. 1 / 1 / Both edges are detected. Caution: Settings in these bits are only valid when ENCAnCTL.ENCAnCRM0 = 1.

(3) ENCAnIOC1 – ENCA I/O Control Register 1

This is an 8-bit register that controls the clearing condition and valid edges for the input and clearing input signals from the encoder (ENCAnE0, ENCAnE1 and ENCAnEC).

Access This register can be read/written in 8-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base0> + 44_H

Initial value 00_H

A reset from any source will initialize the bits

7	6	5	4	3	2	1	0
ENCAn SCE	ENCAn ZCL	ENCAn BCL	ENCAn ACL	ENCAn ECS[1:0]		ENCAn EIS[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18-9 ENCAnIOC1 Register Contents (1/2)

Bit Position	Bit Name	Function
7	ENCAnSCE	This bit selects the method for clearing of the encoder counter. 0: Detection of any valid edge in ENCAnEC causes clearing. 1: Detection of the level of ENCAnEC selected as the clearing condition causes clearing. When ENCAnSCE = 0, settings of the ENCAnZCL, ENCAnACL, and ENCAnBCL bits become ineffective, and settings of the ENCAnECS1 and ENCAnECS0 bits become effective. The counter is cleared to 0000_H when ENCAnSCE = 1 and the clearing conditions set in the ENCAnZCL, ENCAnACL, and ENCAnBCL bits are all matched (settings of the ENCAnECS1 and ENCAnECS0 bits become ineffective). Caution: When ENCAnSCE = 1, set bits ENCAnUDS1 and ENCAnUDS0 to 10B or 11B.
6	ENCAnZCL	This bit is used to set the active level for clearing in the ENCAnEC encoder-clearing input signal. 0: Clearing condition: Low level 1: Clearing condition: High level Caution: This bit is valid when ENCAnSCE = 1.
5	ENCAnBCL	This bit is used to set the active level for clearing in the ENCAnE1 encoder input signal. 0: Clearing condition: Low level 1: Clearing condition: High level Caution: This bit is valid when ENCAnSCE = 1.
4	ENCAnACL	This bit is used to set the active level for clearing in the ENCAnE0 encoder input signal. 0: Clearing condition: Low level 1: Clearing condition: High level Caution: This bit is valid when ENCAnSCE = 1.

Table 18-9 ENCAnIOC1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>ENCAnECS1,
ENCAnECS0</td><td>These bits set the valid edge of the encoder-clearing input signal (ENCAnEC).

<table><tr><th>ENCAn ECS1</th><th>ENCAn ECS0</th><th>Operation</th></tr><tr><td>0</td><td>0</td><td>Neither edge is detected.</td></tr><tr><td>0</td><td>1</td><td>Rising edges are detected.</td></tr><tr><td>1</td><td>0</td><td>Falling edges are detected.</td></tr><tr><td>1</td><td>1</td><td>Both edges are detected.</td></tr></table>Caution: These bits are valid when ENCAnSCE bit =0.</td></tr>
<tr><td>1, 0</td><td>ENCAnEIS1,
ENCAnEIS0</td><td>These bits set the valid edge for input signals from the encoder (signals on the ENCAnE0 and ENCAnE1 pins).

<table><tr><th>ENCAn EIS1</th><th>ENCAn EIS0</th><th>Operation</th></tr><tr><td>0</td><td>0</td><td>Neither edge is detected.</td></tr><tr><td>0</td><td>1</td><td>Rising edges are detected.</td></tr><tr><td>1</td><td>0</td><td>Falling edges are detected.</td></tr><tr><td>1</td><td>1</td><td>Both edges are detected.</td></tr></table>Caution: These bits are valid when the ENCAnUDS1 and ENCAnUDS0 bits are 00B or 01 and invalid when the ENCAnUDS1 and ENCAnUDS0 are 10B or 11B.</td></tr>
</table>

(4) ENCAnFLG – ENCA Status Flag Register

This 8-bit flag register indicates the state of the timer counter for the encoder.

Access This register is read-only and readable in 8-bit units.

Address <ENCAn_base1> + $0C_H$

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	ENCAn CSF	ENCAn UDF	ENCAn OVF
R	R	R	R	R	R	R	R

Table 18-10 ENCAnFLG Register Contents

Bit Position	Bit Name	Function
2	ENCAnCSF	This bit is a flag that indicates whether the timer counter is counting up or down. 0: Counting up 1: Counting down Caution: This bit is cleared at the start of counting operations.
1	ENCAnUDF	This bit is a flag that indicates if an underflow has or has not occurred during timer-counter operations. The value of the 16-bit counter changing from 0000_H to $FFFF_H$ is an underflow of the counter, and an underflow interrupt (INTENCAnIUD) is also generated if the ENCAnUDF flag is set to 1. Setting condition An underflow during encoder timer-counter operations causes setting of this flag to 1. Clearing condition Writing 1 to the ENCAnCLUD bit of the ENCAnFGC register, setting the EnCAnTS bit to 1 while the ENCAnTE bit = 0, or input of the synchronous start trigger signal (ENCAnTSST signal) at the high level causes clearing of this flag to 0. Caution: This bit is cleared at the start of counting operations.
0	ENCAnOVF	This bit is a flag that indicates if an overflow has or has not occurred during timer-counter operations. The value of the 16-bit counter changing from $FFFF_H$ to 0000_H is an overflow of the counter, and an overflow interrupt (INTENCAnIOV) is also generated if the ENCAnOVF flag is set to 1. Setting condition An overflow during encoder timer-counter operations causes setting of this flag to 1. Clearing condition Writing 1 to the ENCAnCLOV bit of the ENCAnFGC register, setting the ENCAnTS bit to 1 while the ENCAnTE bit = 0, or input of the synchronous start trigger signal (ENCAnTSST signal) at the high level causes clearing of this flag to 0. Caution: This bit is cleared at the start of counting operations.

(5) ENCAnFGC – ENCA Status Flag Clear Register

This register is an 8-bit register that controls clearing of the timer counter status flags in the ENCAnFLG register.

Access This register can be written in 8-bit units.
This register always returns 00_H when read.

Address <ENCAn_base1> + 10_H

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	ENCAn CLUD	ENCAn CLOV
R	R	R	R	R	R	W	W

Table 18-11 ENCAnFGC Register Contents

Bit Position	Bit Name	Function
1	ENCAnCLUD	This bit clears the underflow flag. 0: Invalid 1: Clears ENCAnUDF of the ENCAnFLG register (clears underflow detection).
0	ENCAnCLOV	This bit clears the overflow flag. 0: Invalid 1: Clears ENCAnOVF of the ENCAnFLG register (clears overflow detection).

(6) ENCAnCCR0 – ENCA Capture/Compare Register 0

This register is a 16-bit register that is used in both capture and comparison roles (i.e. to hold captured values or values for use in comparison).

Access This register can be read/written in 16-bit units.
This register can be written in 16-bit units when used as a comparison register (ENCAnCTL.ENCAnCRM0 = 0). When used as a capture register (ENCAnCTL.ENCAnCRM0 = 1), writing to this register during operation is invalid.

Address <ENCAn_base1> + 00_H

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCCR0[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18-12 ENCAnCCR0 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAn CCR0[15:0]	If ENCAnCTL.ENCAnCRM0 = 0, ENCAnCCR0 is comparison register. Set the value to be compared with the timer counter value. If ENCAnCTL.ENCAnCRM0 = 1, ENCAnCCR0 is capture register. The captured timer counter value is stored. Note: Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAnCTL.ENCAnLDE setting. For details, refer to the description of the ENCAnLDE bit in ENCA control register ENCAnCTL.

(7) ENCAnCCR1 – ENCA Capture/Compare Register 1

This register is a 16-bit register that is used in both capture and comparison roles (i.e. to hold captured values or values for use in comparison).

Access This register can be read/written in 16-bit units.
This register can be written in 16-bit units when used as a comparison register (ENCAnCTL.ENCAnCRM0 = 0). When used as a capture register (ENCAnCTL.ENCAnCRM0 = 1), writing to this register during operation is invalid.

Address <ENCAn_base1> + 04_H

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCCR1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18-13 ENCAnCCR1 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAn CCR1[15:0]	If ENCAnCTL.ENCAnCRM1 = 0, ENCAnCCR1 is comparison register. Set the value to be compared with the timer counter value. If ENCAnCTL.ENCAnCRM1 = 1, ENCAnCCR1 is capture register. The captured timer counter value is stored. Note: During capture operation, the trigger for capturing to this register differs according to the ENCAnCTL.ENCAnCTS setting. For details, refer to the description of the ENCAnCTS bit in ENCA control register ENCAnCTL.

(8) ENCAnCNT – ENCA Counter Register

This register is a 16-bit timer counter.

Access This register can be read/written in 16-bit units.
This register can be written only when the counting operation is stopped.
Writing to this register during counting operation is invalid.

Address <ENCAn_base1> + 08_H

Initial value 0000_H

A reset from any source will initialize the bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCNT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18-14 ENCAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCNT [15:0]	Operation of counter register

ENCAnTE	Counter Status	Operation
0	Stopped (initial setting)	Set a given value to timer counter register.
0→1	Starting operation	Start counting up/down from the set given value.
1	In operation	Counting up/down
1→0	Stopped	Counting stopped (retains its value before counting stopped).

Note: ENCAnTE state refers to the state of the ENCAnTE bit in the ENCAnTE register.

(9) ENCAnTE – ENCA Timer Enable Status Register

This 8-bit register indicates the state of operation of the encoder timer.

Access This register is read-only and readable in 8-bit units.

Address <ENCAn_base1> + 14_H

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTE
R	R	R	R	R	R	R	R

Table 18-15 ENCAnTE Register Contents

Bit Position	Bit Name	Function
0	ENCAnTE	This is a status bit that indicates the operation enabled/stopped status of the encoder timer. 0: Operation stopped status 1: Operation enabled status Clearing condition This bit is cleared to 0 when 1 is written to ENCAnTT.ENCAnTT. Setting condition Writing of 1 to the ENCAnTS.ENCAnTS bit or the input of a high level of the simultaneous start trigger signal (ENCAnTSST) causes setting of this bit to 1.

(10) ENCAnTS – ENCA Timer Start Trigger Register

This 8-bit register controls starting of the encoder timer.

Access This register can be read/written in 8-bit units.
This register always returns 00_H when read. Writing to this register is only valid when ENCAnTE.ENCAnTE = 0.

Address <ENCAn_base1> + 18_H

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTS
R	R	R	R	R	R	R	W

Table 18-16 ENCAnTS Register Contents

Bit Position	Bit Name	Function
0	ENCAnTS	This is the trigger bit for placing the encoder timer in the operation-enabled state. 0: Disabled 1: Operation is enabled (setting the ENCAnTE.ENCAnTE bit to 1 when the ENCAnTS bit to 1).

(11) ENCAnTT – ENCA Timer Stop Trigger Register

This 8-bit register controls stopping of the encoder timer.

Access This register can be read/written in 8-bit units.
This register always returns 00_H when read.

Address <ENCAn_base1> + $1C_H$

Initial value 00_H

A reset from any source will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTT
R	R	R	R	R	R	R	W

Table 18-17 ENCAnTT Register Contents

Bit Position	Bit Name	Function
0	ENCAnTT	This is the trigger bit for stopping ENCAn operations. 0: Disabled 1: Operation is stopped (setting the ENCAnTE.ENCAnTE bit to 0 when the ENCAnTT bit to 1).

18.4 Functional Description

The encoder inputs and encoder-clearing input (ENCAnE0, ENCAnE1, and ENCAnEC) control the counter of ENCAn by causing it to count up or down or by clearing it.
The ENCAnCCR0 and ENCAnCCR1 registers are available for use as comparison registers or as capture registers.

18.4.1 Timer Counter Operation

Operations of the encoder timer are described below.

The figure below shows the basic operations. For detailed descriptions of the individual types of operation, refer to the corresponding passages.

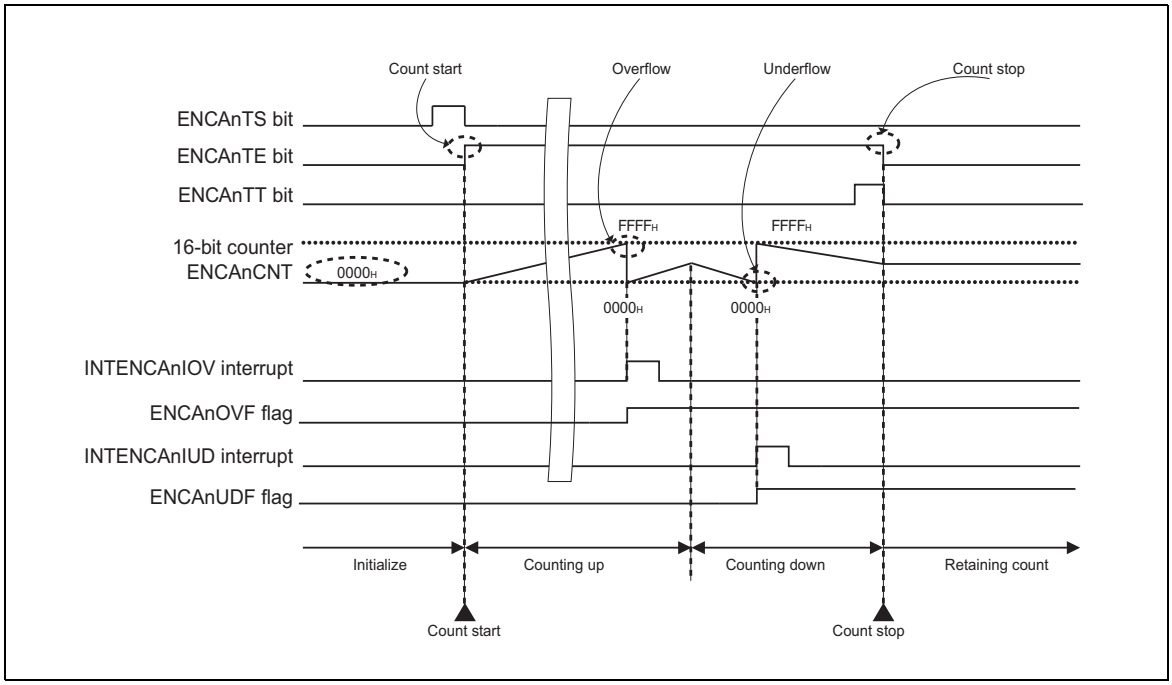


Figure 18-2 Timer Counter Initial Value Setting/Start/Stop

(1) Timer Counter Initial Value Setting

The initial value of the ENCAn counter register (ENCAnCNT) can be set in the counter operation stopped status (ENCAnTE.ENCAnTE = 0).

Combinations of compare/capture in ENCAnCCR0 and ENCAnCCR1 are listed in the table below.

Condition for Use	ENCAnCCR0 Register	ENCAnCCR1 Register
Only comparison is in use (encoder comparison mode).	Comparison function	Comparison function
Comparison and capture are in use (encoder capture and comparison mode)	Comparison function	Capture function
Capture and capture are in use (encoder capture mode).	Capture function	Capture function

(2) Timer Counter Startup

Writing "1" to the timer start trigger bit (ENCAnTS.ENCAnTS) causes setting of the timer status enable bit (ENCAnTE.ENCAnTE) to "1" (enables counting), and counting proceeds upon detection of a valid edge in the input signal from the encoder.

(3) Overflow Operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. When an overflow occurs, an overflow interrupt (INTENCAnIOV) is output, and the overflow flag (ENCAnOVF) is set to "1". The overflow flag (ENCAnOVF) is cleared to "0" when "1" is set to the overflow clear bit (ENCAnCLOV).

Operations in the generation of an overflow and clearing of the overflow flag are described below.

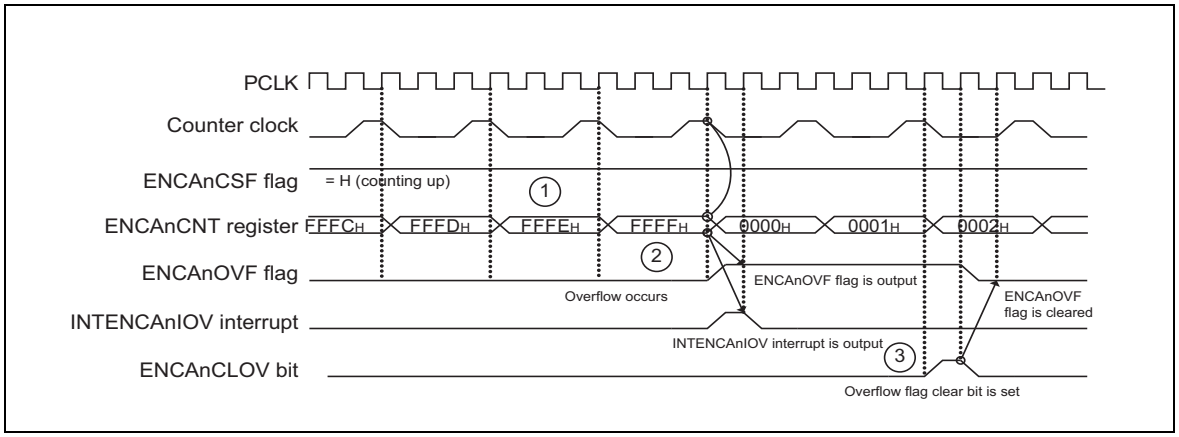


Figure 18-3 Generation of an Overflow and Settings to Clear the Overflow Flag

1. From label 1 in the figure, the counter value increases from $FFFE_H$ to $FFFF_H$.
2. From label 2, the counter overflows when its value changes from $FFFF_H$ to 0000_H. At the same time, an overflow interrupt is output and the overflow flag is set to 1.
3. Label 3 indicates the method for clearing of the overflow flag; i.e. the overflow flag is cleared to 0 by setting ENCAnFGC.ENCAnCLOV to 1. Alternatively, the overflow flag may be cleared by setting ENCAnTS.ENCAnTS to 1 while ENCAnTE.ENCAnTE = 0, or by placing the signal input as ENCAnTSST (simultaneous start trigger input) at the high level.

(4) Underflow Operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. When an underflow occurs, an underflow interrupt (INTENCAnIUD) is output, and the underflow flag (ENCAnUDF) is set to "1". The underflow flag (ENCAnUDF) is cleared to "0" when "1" is set to the underflow clear bit (ENCAnCLUD).

Operations in the generation of an underflow and clearing of the underflow flag are described below.

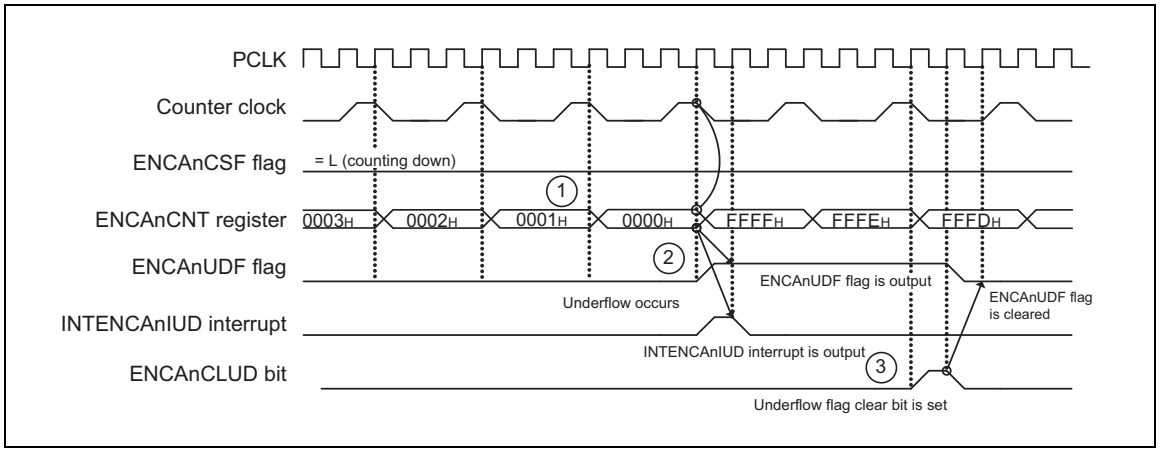


Figure 18-4 Generation of an Underflow and Settings to Clear the Underflow Flag

1. From label 1 in the figure, the counter value decreases from 0001_H to 0000_H.
2. From label 2, the counter underflows when its value changes from 0000_H to $FFFF_H$. At the same time, an underflow interrupt is output, setting the underflow flag to 1.
3. Label 3 indicates the method for clearing of the underflow flag; i.e. the underflow flag is cleared to 0 by setting ENCAnFGC.ENCAnCLUD to 1. Alternatively, the underflow flag may be cleared by setting ENCAnTS.ENCAnTS to 1 while ENCAnTE.ENCAnTE = 0, or by placing the signal input as ENCAnTSST (synchronous start trigger) at the high level.

(5) Timer Counter Stop

Writing "1" to the timer stop trigger bit (ENCAnTT.ENCAnTT) causes clearing of the timer status enable bit (ENCAnTE) to "0" and stops the counter. At this time, the timer counter is not reset to 0000_H but retains its value from before counting stopped.

18.4.2 Up/Down Control of Timer Counter

The setting of the ENCAnCTL.ENCAnUDS[1:0] bits and the phase of the encoder inputs (ENCAnE0 and ENCAnE1) control counting up and down by the timer.

(1) When ENCAnCTL.ENCAnUDS[1:0] = 00B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		Signal on the ENCAnE0 Pin	Signal on the ENCAnE1 Pin	Counting Operation
0	0	Rising edge	High level	Counting down
		Falling edge		
		Both edges		
		Rising edge	Low level	Counting up
		Falling edge		
		Both edges		

The valid edge specification for ENCAnE0 is made by setting ENCAnIOC1.ENCAnEIS[1:0].

"Counting operation" indicates that counting up or down proceeds when the valid edges of the signals on ENCAnE0 and ENCAnE1 match.

The timing chart below shows counting operation when ENCAnCTL.ENCAnUDS[1:0] = 00B.

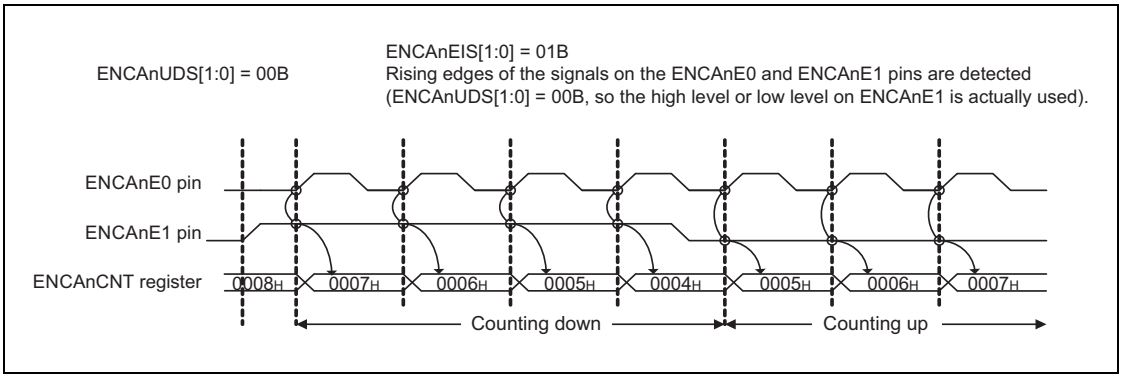


Figure 18-5 Counting Operation when ENCAnCTL.ENCAnUDS[1:0] = 00B

(2) When ENCAnCTL.ENCAnUDS[1:0] = 01B

ENCAnUDS1	ENCAnUDS0	Description of Operation		
		Signal on the ENCAnE0 Pin	Signal on the ENCAnE1 Pin	Counting Operation
0	1	Low level	Rising edge	Counting down
			Falling edge	
			Both edges	
		High level	Rising edge	
			Falling edge	
			Both edges	
		Rising edge	Low level	Counting up
		Falling edge		
		Both edges		
		Rising edge	High level	
		Falling edge		
		Both edges		
		Simultaneous input		Retaining

The valid edge specification for ENCAnE0 and ENCAnE1 is made by setting ENCAnIOC1.ENCAnEIS[1:0].

"Counting operation" indicates that counting up or down proceeds when the valid edges of the signals on ENCAnE0 and ENCAnE1 match. When valid edges coincide, the count is retained.

The timing chart below shows counting operation when ENCAnCTL.ENCAnUDS[1:0] = 01B.

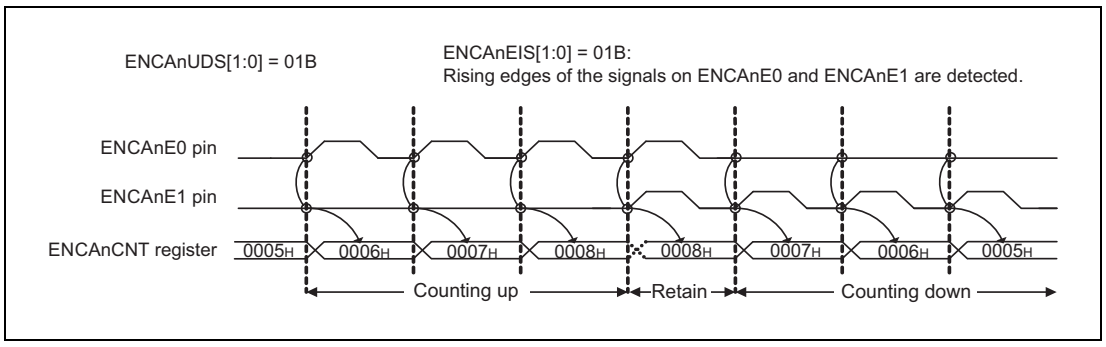


Figure 18-6 Counting Operation of ENCAnCTL.ENCAnUDS[1:0] = 01B

(3) When ENCAnCTL.ENCAnUDS[1:0] = 10B

ENCAnUDS1	ENCAnUDS0	Description of Operation		
		Signal on the ENCAnE0 Pin	Signal on the ENCAnE1 Pin	Counting Operation
1	0	Rising edge	Low level	Counting down
		Rising edge	Falling edge	
		Falling edge	Low level	Counting up
		Falling edge	Falling edge	
		Low level	Rising edge	Retaining
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

The valid edge specification for ENCAnE0 and ENCAnE1 (setting of ENCAnIOC1.ENCAnEIS[1:0] bits) is ineffective.

The timing chart below shows counting operation when ENCAnCTL.ENCAnUDS[1:0] = 10B.

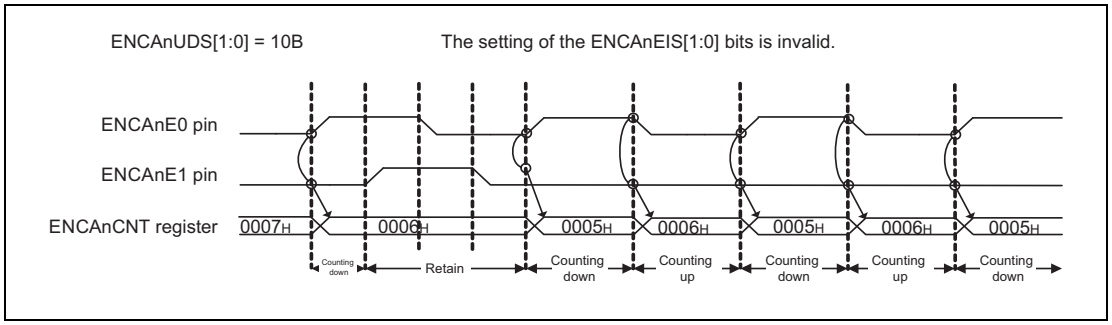


Figure 18-7 Counting Operation of ENCAnCTL.ENCAnUDS[1:0] = 10B

(4) When ENCAnCTL.ENCAnUDS[1:0] = 11B

ENCAnUDS1	ENCAnUDS0	Description of Operation		
		Signal on the ENCAnE0 Pin	Signal on the ENCAnE1 Pin	Counting Operation
1	1	Low level	Falling edge	Counting down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Counting up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Retaining

The valid edge specification for ENCAnE0 and ENCAnE1 (setting of ENCAnIOC1.ENCAnEIS[1:0] bits) is ineffective.

When the valid edges of the signals on ENCAnE0 and ENCAnE1 coincide, the count is retained.

The timing chart below shows counting operation when ENCAnCTL.ENCAnUDS[1:0] = 11B.

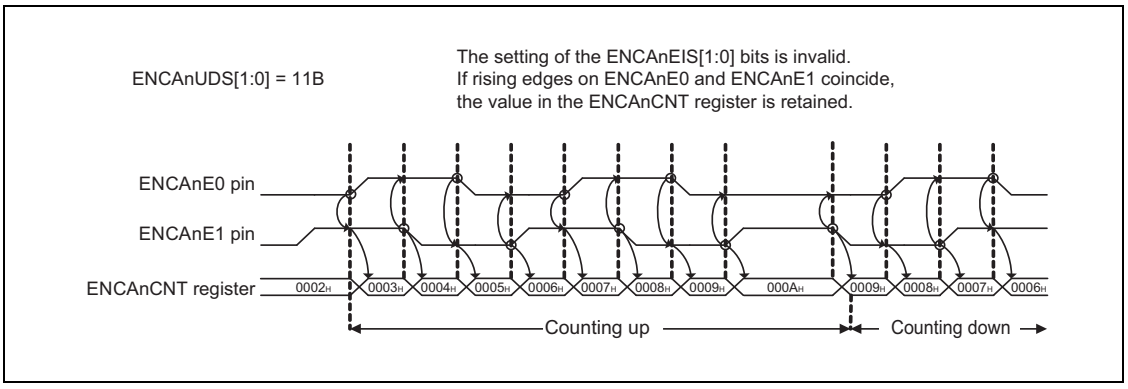


Figure 18-8 Counting Operation of ENCAnCTL.ENCAnUDS[1:0] = 11B

18.4.3 Control of Timer Counter Clearing

The following conditions lead to clearing of the timer counter.

- Detection of a valid edge of the encoder clearing input signal (signal on the ENCAnEC pin)
- Level detection of the encoder input and encoder clearing input signals (signals on the ENCAnE0, ENCAnE1, and ENCAnEC pins)

Clearing refers to clearing in accord with the settings of the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS1 and ENCAnECS0 bits in the ENCAnIOC1 register.

Condition for Clearing	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS1, ENCAnECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

(1)-1 Clearing on Detection of a Valid Edge of the Encoder-Clearing Input Signal (when ENCAnIOC1.ENCAnSCE = 0).

- Upon detection of the valid edge of ENCAnEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCAnEC is specified by the setting of the ENCAnECS1 and ENCAnECS0 bits in the ENCAnIOC1 register.
- The settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits in the ENCAnIOC1 register are ineffective.
- A clear interrupt signal (INTENCAnIEC) is output simultaneously with timer counter clearing.

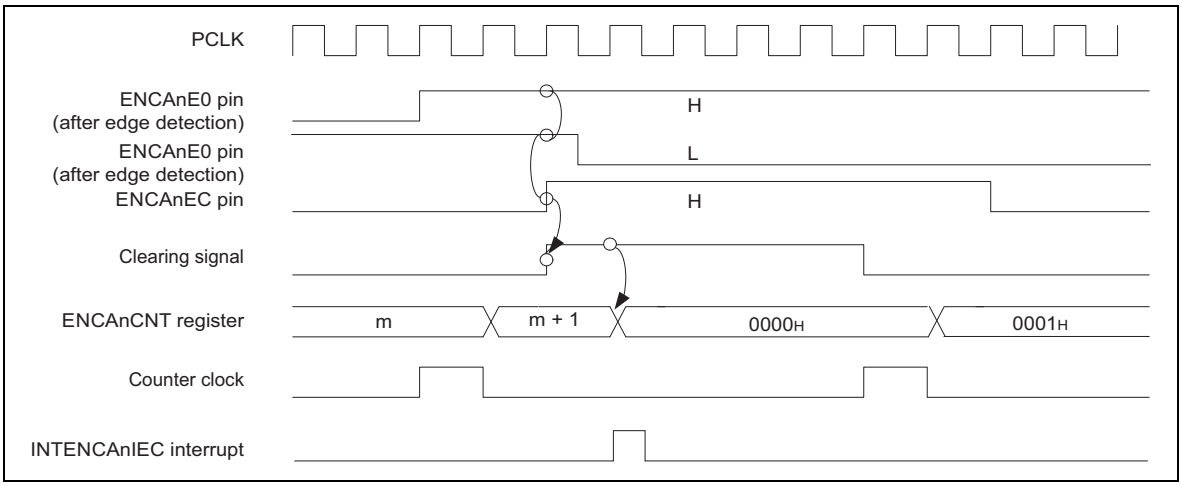


Figure 18-9 Operations to Clear the Counter in Response to Encoder Clearing Input (ENCAnEC)

(1)-2 Operations for Counter Clearing and Capture in Response to Encoder Clearing Input (ENCAnEC)

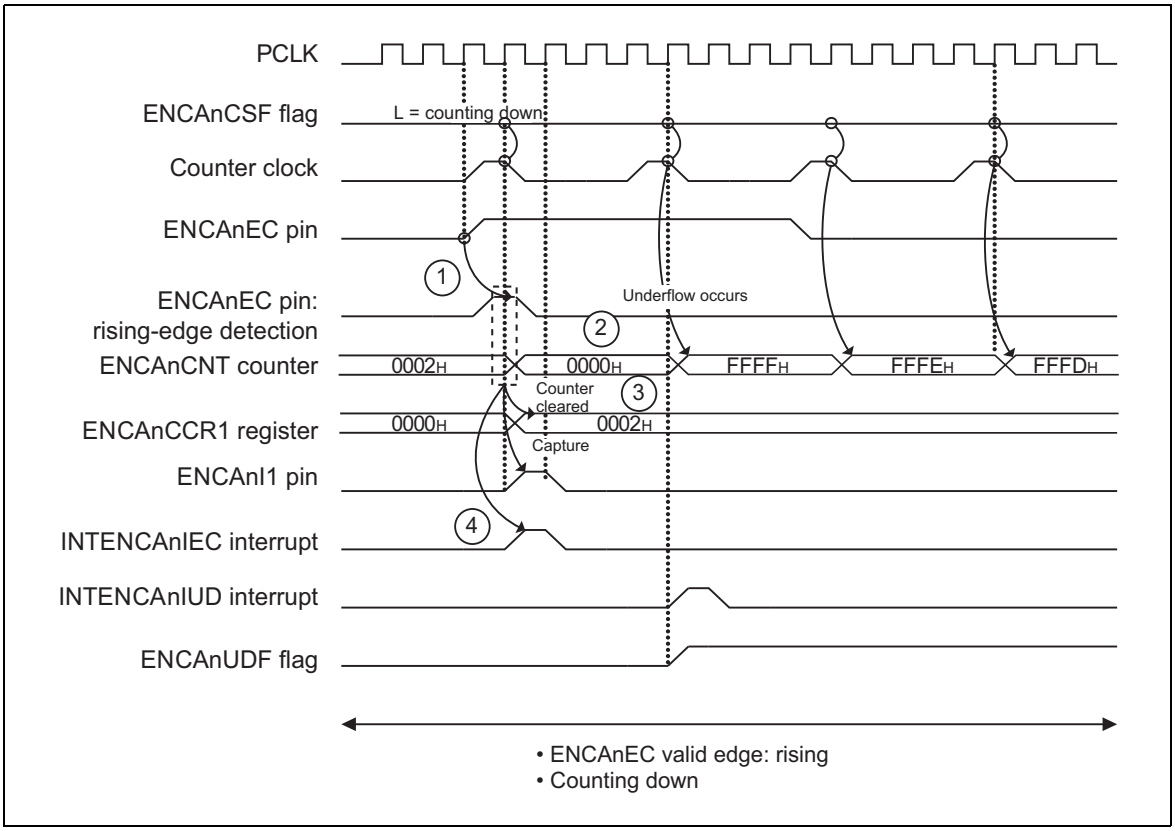


Figure 18-10 Timing of Counter Clearing and Capture Operations in Response to Encoder Clearing Input (ENCAnEC)

<Setting condition>

ENCAnCTL.ENCAnCRM1 = 1: selects the ENCAnCCR1 register for capture.
ENCAnCTL.ENCAnCTS = 1: selects the ENCAnEC pin input as capture trigger input.
ENCAnIOC1.ENCAnECS[1:0] = 01B: selects detection of the rising edge on the ENCAnEC pin input

1. Capture proceeds on the rising edge of the trigger signal input on the ENCAnEC pin.
2. Clearing of the counter's value to 0000_H in response to the input on the ENCAnEC pin proceeds.
3. The pre-clearing value from the counter (0002_H) is captured in the ENCAnCCR1 register on the rising edge of the input signal on the ENCAnEC pin.
4. The cleared interrupt (INTENCAnIEC) and capture interrupt (INTENCAnI1) in response to the input signal on the ENCAnEC pin are output at the same time.

(2) Clearing in Response to Detection of the Input and Clearing Input Signals from the Encoder

Clearing is in accord with the settings of the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS1 and ENCAnECS0 bits in the ENCAnIOC1 register (when ENCAnIOC1.ENCAnSCE = 1).

- Clearing levels on the ENCAnEC, ENCAnE1, and ENCAnE0 pins as specified by the settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits in the ENCAnIOC1 register are detected and the timer counter is cleared to 0000_H in synchronization with the operating clock.
- The setting of the ENCAnIOC1.ENCAnECS[1:0] bits is ineffective.
- The encoder cleared interrupt signal (INTENCAnIEC) is output simultaneously with the timer counter clearing.

Conditions for clearing of the timer counter in accord with the setting of the ENCAnZCL, ENCAnBCL and ENCAnACL bits in the ENCAnIOC1 register are given in the table below.

Counter Clearing Condition Settings			Level Input on the Encoder Pin		
ENCAnZCL	**ENCAnBCL**	**ENCAnACL**	**ENCAnEC**	**ENCAnE1**	**ENCAnE0**
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

(a) When Input on the ENCAnEC Pin Follows Input on the ENCAnE1 Pin during Counting up (When ENCAnIOC1.ENCAnACL = 1, ENCAnIOC1.ENCAnBCL = 0, ENCAnIOC1.ENCAnZCL = 1, and ENCAnCTL.ENCAnUDS[1:0] = 11B)

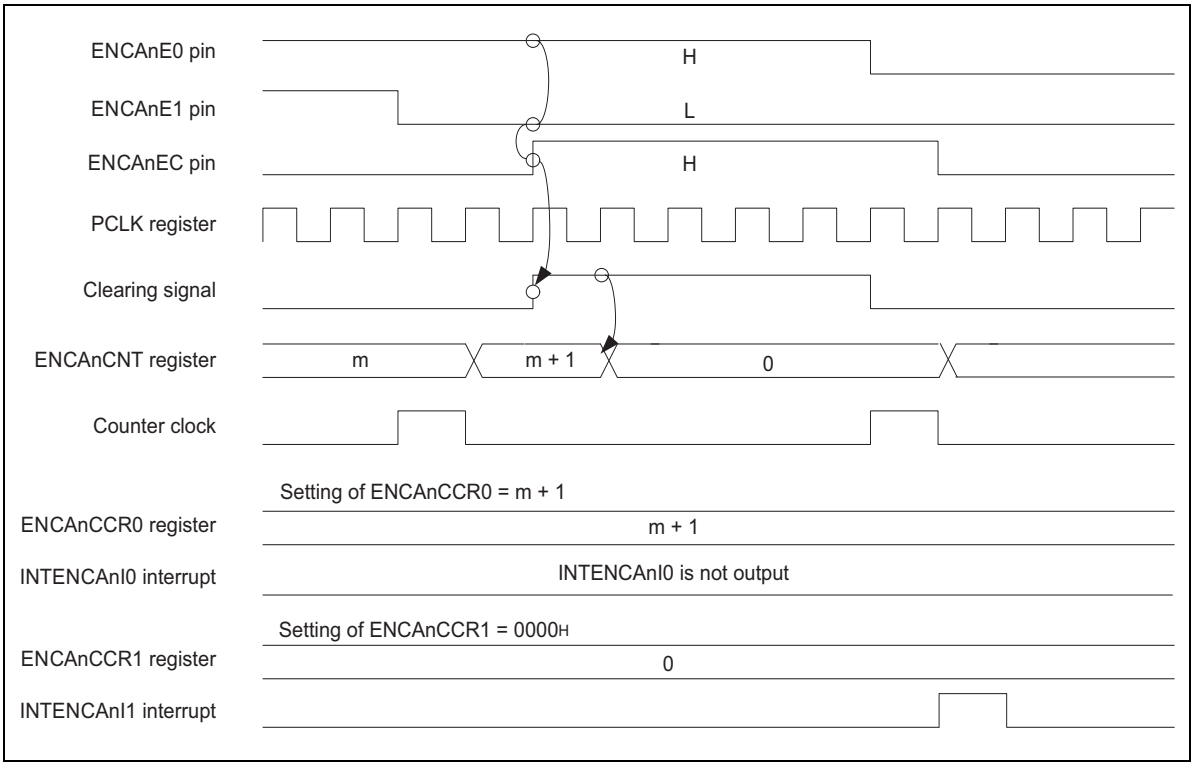


Figure 18-11 Timing of Clearing when Input on the ENCAnEC Pin Follows Input on the ENCAnE1 Pin during Counting up

(b) When the Timing of Input on the ENCAnEC and ENCAnE1 Pins Coincides during Counting up (When ENCAnIOC1.ENCAnACL = 1, ENCAnIOC1.ENCAnBCL = 0, ENCAnIOC1.ENCAnZCL = 1, and ENCAnCTL.ENCAnUDS[1:0] = 11B)

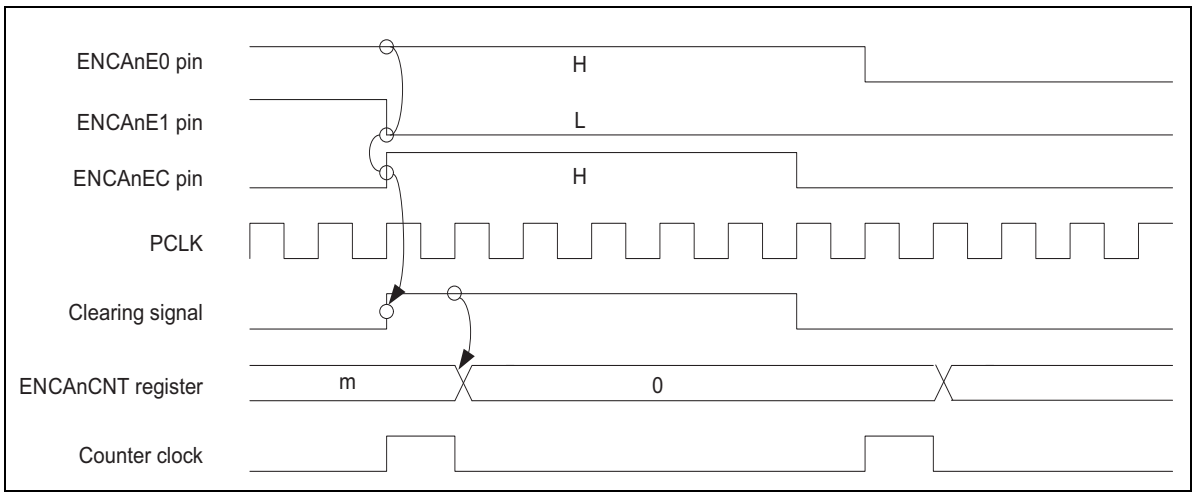


Figure 18-12 Timing of Clearing when Input on the ENCAnEC and ENCAnE1 Pins Coincides during Counting up

(c) When Input on the ENCAnEC Pin Precedes Input on the ENCAnE1 Pin during Counting up (When ENCAnIOC1.ENCAnACL = 1, ENCAnIOC1.ENCAnBCL = 0, ENCAnIOC1.ENCAnZCL = 1, and ENCAnCTL.ENCAnUDS[1:0] = 11B)

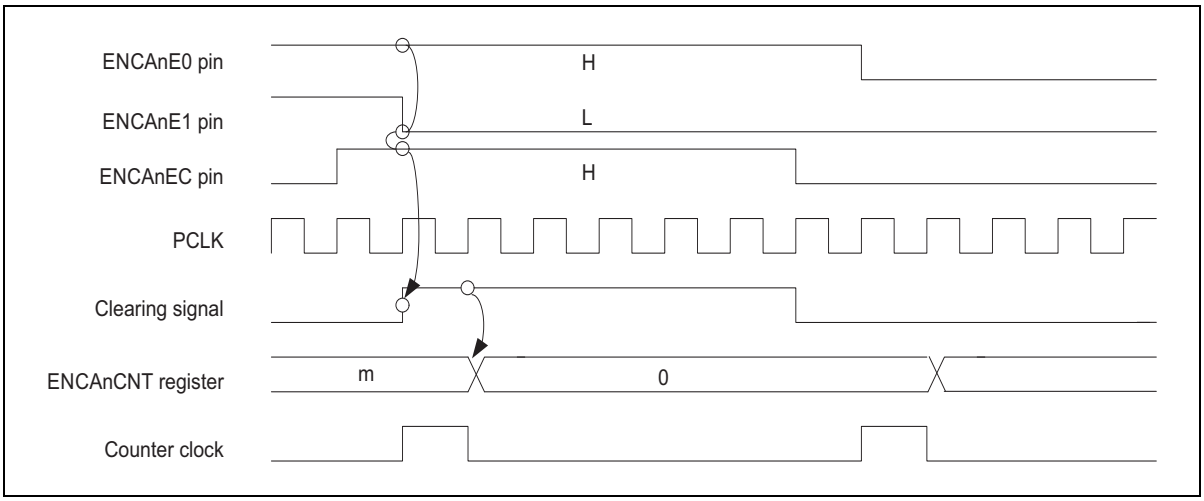


Figure 18-13 Timing of Clearing when Input on the ENCAnEC Pin Precedes Input on the ENCAnE1 Pin during Counting up

(d) When Input on the ENCAnEC Pin Follows Input on the ENCAnE1 Pin during Counting down (When ENCAnIOC1.ENCAnACL = 1, ENCAnIOC1.ENCAnBCL = 0, ENCAnIOC1.ENCAnZCL = 1, and ENCAnCTL.ENCAnUDS[1:0] = 11B)

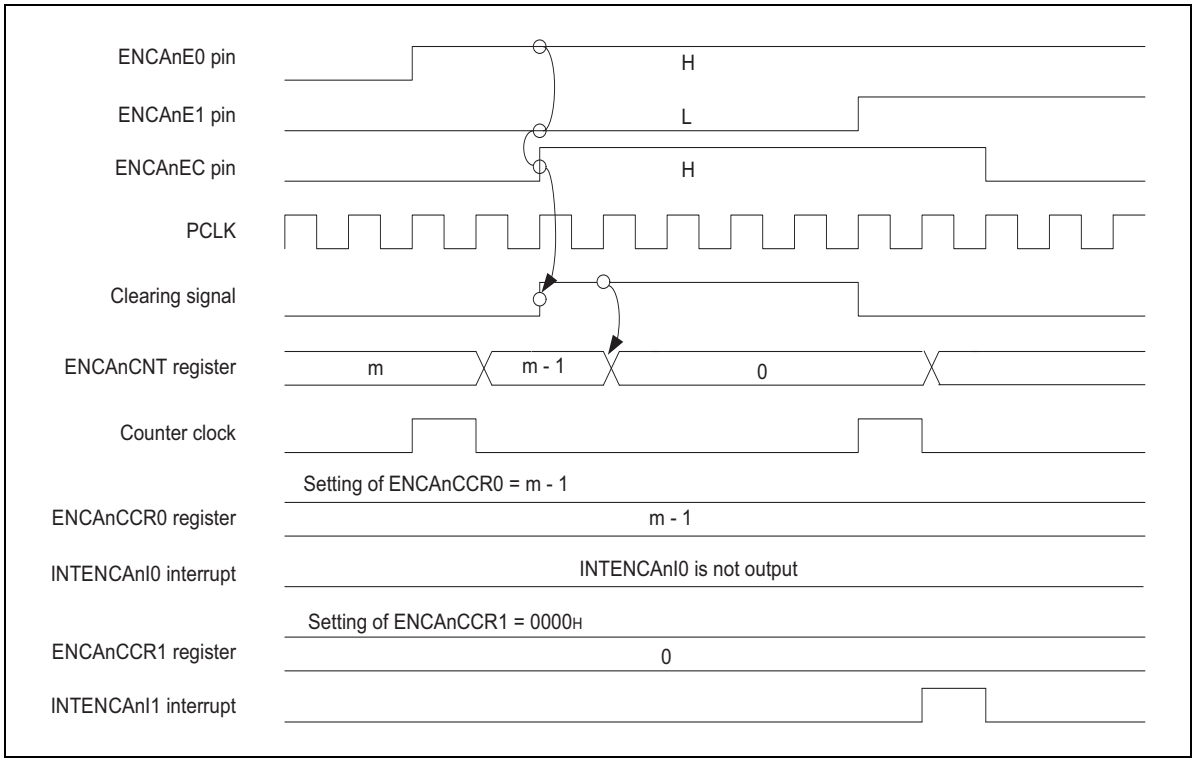


Figure 18-14 Timing of Clearing when Input on the ENCAnEC Pin Follows Input on the ENCAnE1 Pin during Counting down

18.4.4 Detailed Description of the ENCAnCCR0 Register

(1) Compare Function

- When ENCAnCTL.ENCAnCRM0 = 0, the ENCAnCCR0 register functions as a dedicated comparison register.
- A compare 0 match interrupt (INTENCAnI0) is output when the values in the timer counter and the ENCAnCCR0 register match.
- When ENCAnCTL.ENCAnECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock (PCLK) upon a compare match if further counting is to be counting up.

ENCAnCCR0 Function	Compare Match Clear Control	Next Counting Operation	Timer Counter Clearing Upon Compare Match with ENCAnCCR0
ENCAnCRM0	ENCAnECM0		
0 (Compare)	0	Counting up	Does not clear timer counter (continues counting operation).
		Counting down	
	1	Counting up	Clears timer counter to 0000_H.
		Counting down	Does not clear timer counter (continues counting operation).

When ENCAnCTL. ENCAnLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (INTENCAnIUD) is output.

(2) Capture Function

- When ENCAnCTL.ENCAnCRM0 = 1, the ENCAnCCR0 register functions as a dedicated capture register.
- On detection of the valid edge of capture trigger input 0 (ENCAnI0), the value of the timer counter is stored in the ENCAnCCR0 register.
- A capture 0 interrupt (INTENCAnI0) is output during a capture operation.

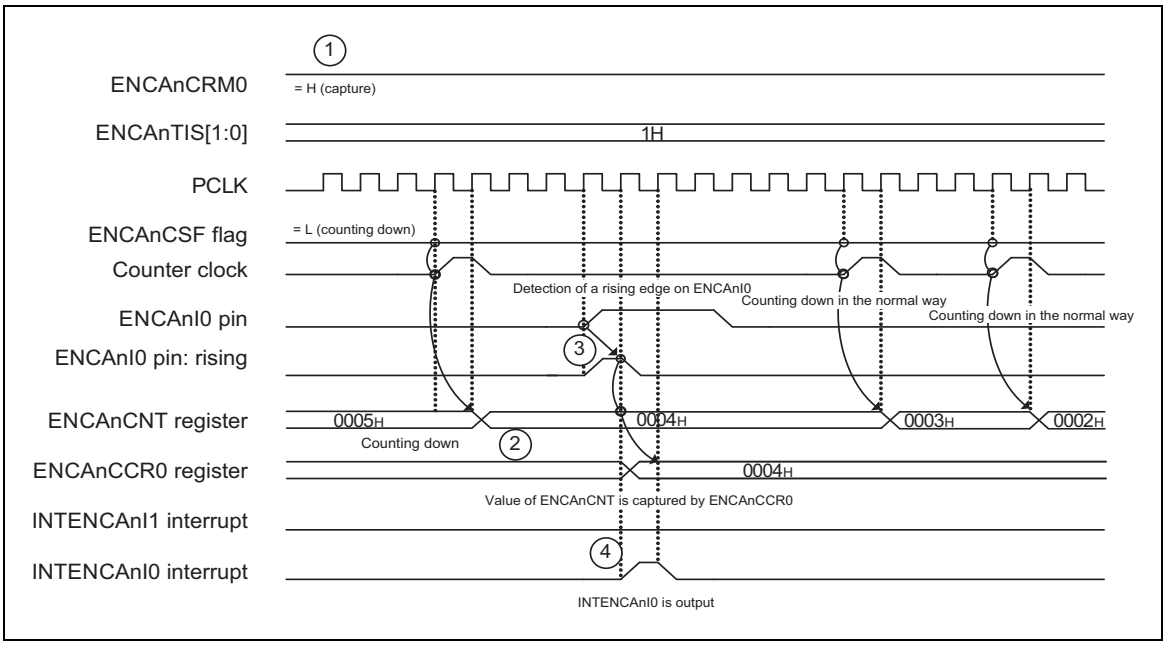


Figure 18-15 Capture Operation between Pulses of the Counter Clock (ENCAnCCR0)

1. The following values are set at label 1: ENCAnCRM0 = 1, ENCAnTIS1 and ENCAnTIS0 = 01B.
2. Counting down proceeds at label 2.
3. At label 3, a rising edge of the input signal on the ENCAnI0 pin is detected, and the value reached by the counter is captured in the ENCAnCCR0 register.
4. At label 4, the interrupt signal (INTENCAnI0) that corresponds to capture to the ENCAnCCR0 register is output.

18.4.5 Detailed Description of the ENCAnCCR1 Register

(1) Compare Function

- When ENCAnCTL.ENCAnCRM1 = 0, the ENCAnCCR1 register functions as a dedicated comparison register.
- A compare 1 match interrupt (INTENCAnI1) is output when the values in the timer counter and the ENCAnCCR1 register match.
- When ENCAnCTL.ENCAnECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock (PCLK) upon a compare match if further counting is to be counting down.

ENCAnCCR1 Function	Compare Match Clear Control	Next Counting Operation	Timer Counter Clearing upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnECM1		
0 (Compare)	0	Counting up	Does not clear timer counter (continues count operation).
		Counting down	
	1	Counting up	Does not clear timer counter (continues count operation).
		Counting down	Clears timer counter to 0000_H.

Compare match interrupt detection mask function

- When ENCAnCTL.ENCAnCME = 1, masking of compare 1 match interrupt detection is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAnCTL.ENCAnMCS = 0, writing to the ENCAnCCR1 register disables masking of compare 1 match interrupt detection.
- When ENCAnCTL.ENCAnMCS = 1, clearing of the timer counter due to input on the ENCAnEC pin or a match between the value in the ENCAnCCR0 register and the timer counter disables masking of compare 1 match interrupt detection.
- When ENCAnCTL.ENCAnMCS = 1 and ENCAnCTL.ENCAnLDE = 1, loading of the value from the ENCAnCCR0 register to the timer counter upon detection of an underflow disables masking of compare 1 match interrupt detection.
- Setting ENCAnCTL.ENCAnECM1 = 1 is prohibited while compare 1 match interrupt detection masking is enabled.

ENCAnCCR1 Function	Compare 1 Match Interrupt Masking	Interrupt Mask Cancel Trigger		Compare 1 Match Interrupt Output upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnCME	ENCAnMCS	Underflow Occurrence upon ENCAnLDE = 0	
0 (Compare)	0 (Masking disabled)	– (Setting invalid)	–	Outputs compare 1 match interrupt upon each compare match.
	1 (Masking enabled)	0 (Writing to ENCAnCCR1)	Occurred (Loading of ENCAnCCR0 to timer counter)	Outputs compare 1 match interrupt once, upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)
		1 (Timer counter clearing)		

(2) Capture Function

- When ENCAnCTL.ENCAnCRM1 =1, the ENCAnCCR1 register functions as a dedicated capture register.

The capture operation to ENCAnCCR1 is shown in the timing chart in section 18.6, (4) Timing of Basic Encoder Operation 4 (Encoder Capture Mode).

Operations corresponding to the settings of ENCAnCTL.ENCAnCTS are listed in the table below.

ENCAnCCR1 Function	Capture Trigger Selection	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
ENCAnCRM1	ENCAnCTS			
1 (Capture)	0	Capture trigger 1 input (ENCAnI1)	Does not clear timer counter.	(1) Capture 1 interrupt (INTENCAnI1)
	1	Encoder clearing input (ENCAnEC)	Clears timer counter.	(1) Capture 1 interrupt (INTENCAnI1) (2) Encoder clear interrupt (INTENCAnIEC)

The basic operations of the capture function are described below.

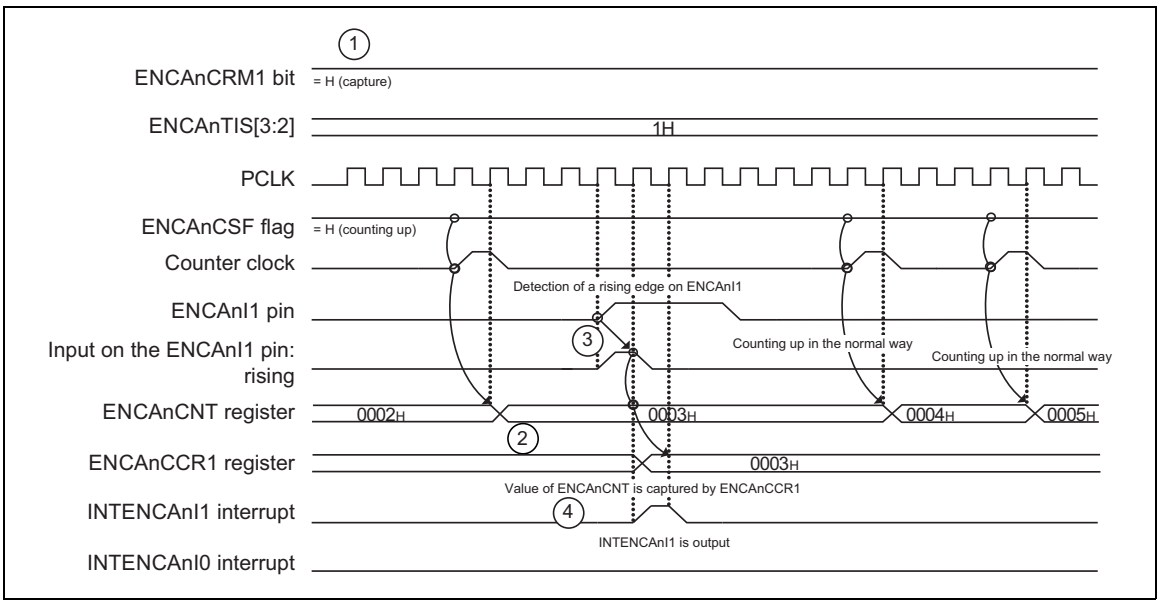


Figure 18-16 Capture between Pulses of the Counter Clock (ENCAnCCR1)

1. The following values are set at label 1: ENCAnCRM1 = 1, ENCAnTIS3 and ENCAnTIS2 = 01B.
2. Counting up proceeds at label 2.
3. At label 3, a rising edge of the input signal on the ENCAnI1 pin is detected, and the value reached by the counter is captured in the ENCAnCCR1 register.
4. At label 4, the interrupt signal (INTENCAnI1) that corresponds to capture to the ENCAnCCR1 register is output.

(3) Timer Counter Clearing upon Comparison Register Match

Clearing of the timer counter upon a match between its value and the setting of ENCAnCCR0/1, in accord with the settings of the ENCAnECM1 and ENCAnECM0 bits in the ENCAnCTL register, is detailed in the following table.

ENCAnECM1 and ENCAnECM0	Next Counting Operation	Timer Counter Clearing upon Compare Match with ENCAnCCR1	Timer Counter Clearing upon Compare Match with ENCAnCCR0
00	Counting up	Does not clear timer counter (continues counting operation).	Does not clear timer counter (continues counting operation).
	Counting down	Does not clear timer counter (continues counting operation).	Does not clear timer counter (continues counting operation).
01	Counting up	Does not clear timer counter (continues counting operation).	Clears timer counter to 0000_H.
	Counting down	Does not clear timer counter (continues counting operation).	Does not clear timer counter (continues counting operation).
10	Counting up	Does not clear timer counter (continues counting operation).	Does not clear timer counter (continues counting operation).
	Counting down	Clears timer counter to 0000_H.	Does not clear timer counter (continues counting operation).
11	Counting up	Does not clear timer counter (continues counting operation).	Clears timer counter to 0000_H.
	Counting down	Clears timer counter to 0000_H.	Does not clear timer counter (continues counting operation).

18.4.6 Timer Counter Operation Start/Stop

(1) Starting the Timer in a Single Encoder-Timer Configuration

Operation can be started by setting ENCAnTE.ENCAnTS = 1.

(2) Starting the Timers in a Dual Encoder-Timer Configuration

This product has two encoder timers, for which independent or synchronized operation is selectable.

In the case of independent operation, operation is started by setting the ENCAnTS bits in the respective ENCAnTE registers to 1.

Setting up input of the simultaneous start trigger (ENCAnTSST signal) to set the enable status bit (ENCAnTE) to 1 causes the trigger to enable counting. When edges are detected in the input signal from the encoder, counting can be executed by the two timers in synchronization.

Input of the simultaneous start trigger (ENCAnTSST) has the same effect on the encoder-timers as setting the ENCAnTS bit in the ENCAnTS register to 1.

(3) Stopping an Encoder Timer

The timer is stopped by setting ENCAnTT.ENCAnTT = 1 and ENCAnTE.ENCAnTE = 0.

In a single encoder-timer configuration, setting the ENCAnTT.ENCAnTT bit to 1 changes the value of the ENCAnTE bit to 0 and thus stops the timer.

(4) Stopping the Timers in a Dual Encoder-Timer Configuration

Setting the ENCAnTT bit in the ENCAnTT register of a given encoder timer to 1 causes the ENCAnTE bit in the ENCAnTE register to be set to 0.

Writing to the ENCAnTT bits in the ENCAnTT registers of the individual encoder timers will lead to each encoder timer stopping with different timing, creating a possible margin of error in the held counter values. Therefore, when resuming operation after it has been stopped, the value in the counter must be re-set or corrected as in the examples below.

Example 1: Remaking the Counter Setting Prior to Restarting Operation

Condition:	Input on the ENCAnEC pin while dual encoder timers are operating with common ENCAnE0 and ENCAnE1 pins.
Restart procedure:	Execute simultaneous restarting by setting the encoder timers to the same value. Setting the timers to the same value eliminates errors in the counted value due to operation being stopped and allows restarting of the operation.

Example 2: Correcting the Counter Setting Prior to Restarting Operation

Condition:	Input on the ENCAnEC pin while dual encoder timers are operating with separate ENCAnE0 and ENCAnE1 pins.
Restart procedure:	Execute simultaneous restarting by calculating the difference between the values of the counters in the respective encoder timers and compensate for the difference in setting the counters for simultaneous restarting. Since the information of the CPU includes the information on the differences between the values of the counters of the encoder timers, operation can be restarted by correcting the error in the counter value that arose when operation was stopped, i.e. by calculating the difference and setting corrected values.

(5) Example of Connection in a Dual Encoder-Timer Configuration

To simultaneously operate the counters of both encoder timer, make the same settings in the ENCAnUDS[1:0] bits in the ENCAnIOC1 register and the ENCAnCTL register.

When using ENCAnCCR0 as the comparison register for both encoder timers, set the same value in the ENCAnCCR0, ENCAnECM0, and ENCAnLDE registers of both encoder timers.

When using ENCAnCCR1 as the comparison register for both encoder timers, set the same value in the ENCAnECM1 registers of both encoder timers.

Since the values are separate if they are not the same, synchronized operation will not be possible.

An example where two encoder timers are installed is shown below.

This setting example assumes the use of the ENCAnCCR0 register as a comparison register, and the ENCAnCCR1 register as a capture register.

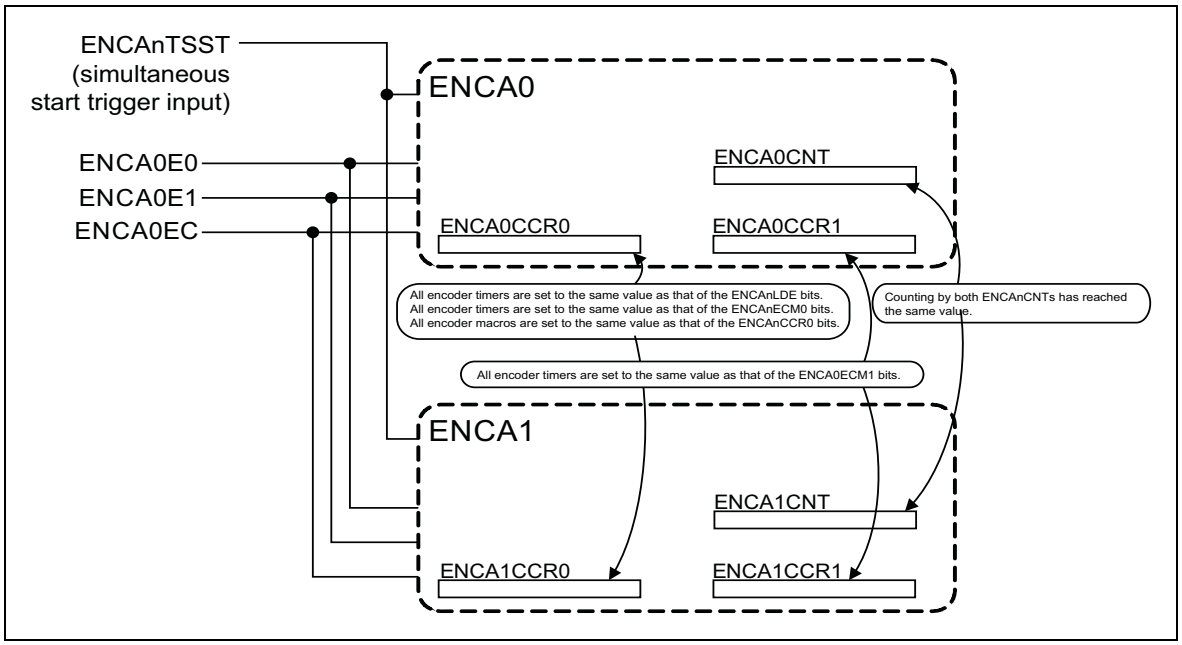


Figure 18-17 Example of Connection in a Dual Encoder-Timer Configuration

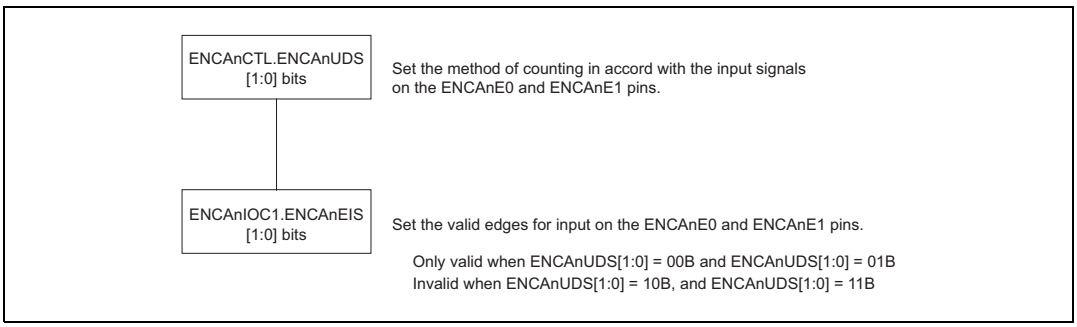
18.5 Setting Sequences

18.5.1 Encoder Timer Setting Procedure

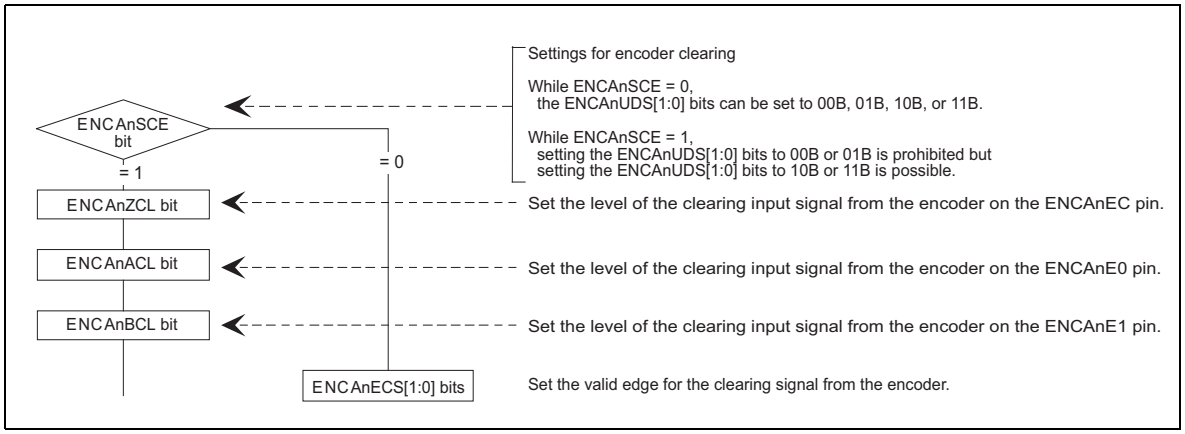
The encoder timer setting procedure is described below.

	Action	State of Setting
Initial setting	Reset release	Power-on state, encoder-timer operation stopped state (writing to the individual registers is possible)
Initial settings of encoder timer	Perform the following initial settings. • Setting for counter • Setting for clearing counter • Setting for ENCAnCCR0 register • Setting for ENCAnCCR1 register	Counting operations are stopped in this state. The value of the ENCAnTE bit indicating the state of operation is 0.
	Perform the initial settings for counter. • Set any 16-bit value to ENCAnCNT register. (When, after setting this register, the ENCAnTS bit is set to "1", the counter operation starts from the set count value.)	The set value is set as the initial value of the counter register.
Operation start	Perform the counter operation start setting. • Set the ENCAnTS bit to "1".	Counting is started in this state. The value of the ENCAnTE bit indicating the state of operation is 1, and the counter clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. • ENCAnCCR0 register setting • ENCAnCCR1 register setting • ENCAnIOC0 register setting	Count proceeds from the initial setting, and is up or down in accord with the setting of the ENCAnE0 and ENCAnE1 pins.
Operation stop	Perform the counter operation stop setting during operation. • Set the ENCAnTT bit to "1".	Counting operations are stopped in this state. The value of the ENCAnTE bit indicating the state of operation is 0.
Encoder timer stop	Reset	The setting registers are initialized.

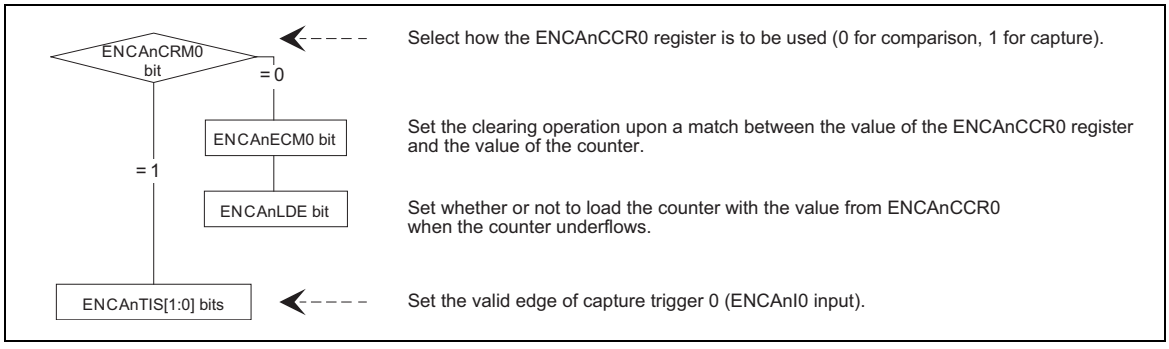
(1) Initial Settings for Counter



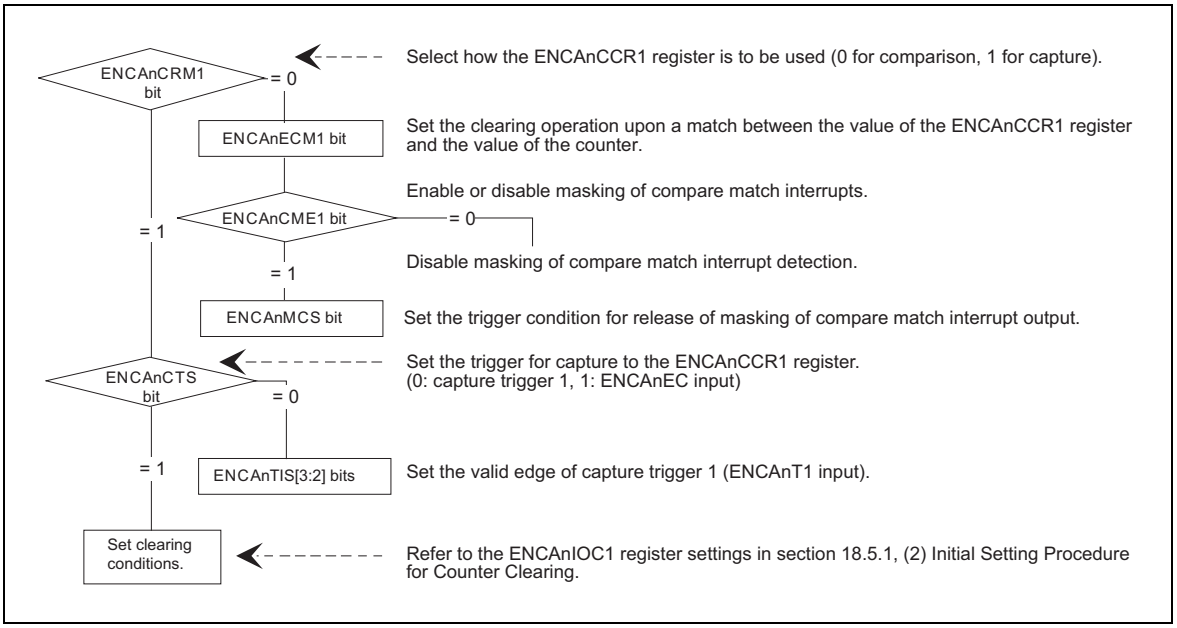
(2) Initial Settings for Counter Clearing



(3) Initial Settings for the ENCAnCCR0 Register



(4) Settings for the ENCAnCCR1 Register



18.6 Timing Charts for Encoder Operations

(1) Timing of Basic Encoder Operation 1 (Encoder Comparison Mode 1)

<Setting condition>

- ENCAnCTL.ENCAnCRM[1:0] = 00B:
  Comparison is selected as the function of the ENCAnCCR0 and ENCAnCCR1 registers.
- ENCAnCTL.ENCAnECM[1:0] = 01B:
  If counting after a match between the values in the counter and the ENCAnCCR0 register is upwards, the counter is cleared.
- ENCAnCTL.ENCAnLDE = 1:
  When the counter underflows, it is loaded with the value from the ENCAnCCR0 register.

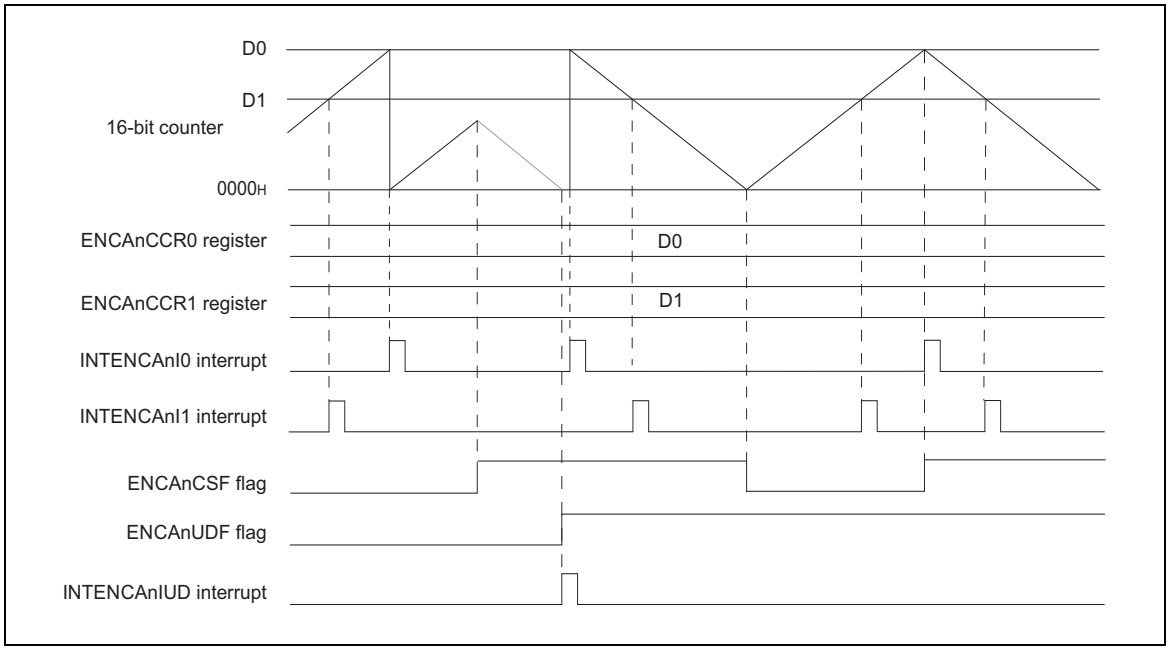


Figure 18-18 Timing of Basic Encoder Operation 1 (Encoder Comparison Mode 1)

A compare match interrupt (INTENCAnI0) is generated when the values of the counter and register ENCAnCCR0 (D0) match.
If further counting is upwards, the counter is cleared to 0000_H because ENCAnECM0 = 1.

A compare match interrupt (INTENCAnI1) is generated when the values of the counter and register ENCAnCCR1 (D1) match.
Counter clearing due to a match with ENCAnCCR1 does not proceed because ENCAnECM1 = 0.

An underflow interrupt (INTENCAnIUD) is generated when the counter underflows.
ENCAnLDE = 1, so the counter is loaded with the value from the ENCAnCCR0 register (D0) when the counter underflows.

ENCAnLDE = 1 and ENCAnECM[1:0] = 01B, so counting is from 0000_H to the setting of the ENCAnCCR0 register.

(2) Timing of Basic Encoder Operation 2 (Encoder Comparison Mode 2)

<Setting condition>

- ENCAnCTL.ENCAnCRM[1:0] = 00B:
  Comparison is selected as the function of the ENCAnCCR0 and ENCAnCCR1 registers.
- ENCAnCTL.ENCAnECM[1:0] = 00B:
  The counter is not cleared on a match between its value and that of the ENCAnCCR0 register.
- ENCAnCTL.ENCAnLDE = 0:
  The counter is not loaded with the value from the ENCAnCCR0 register.

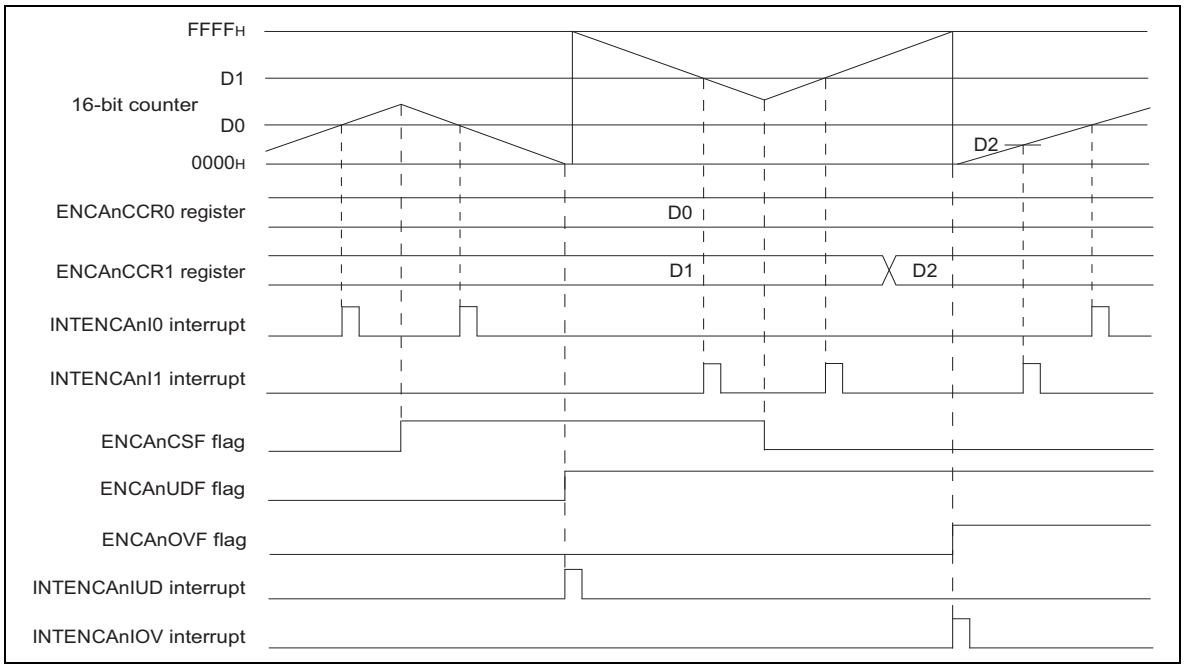


Figure 18-19 Timing of Basic Encoder Operation 2 (Encoder Comparison Mode 2)

A compare match interrupt (INTENCAnI0) is generated when the values of the counter and register ENCAnCCR0 (D0) match.
Counter clearing due to matching with ENCAnCCR0 does not proceed because ENCAnECM0 = 0.

A compare match interrupt (INTENCAnI1) is generated when the values of the counter and register ENCAnCCR1 (D1 and D2) match.
Counter clearing due to matching with ENCAnCCR1 does not proceed because ENCAnECM1 = 0.

Overflow interrupts (INTENCAnIOV) or underflow interrupts (INTENCAnIUD) are generated in response to overflows or underflows of the counter.

(3) Timing of Basic Encoder Operation 3 (Encoder Comparison Mode 3)

<Setting condition>

- ENCAnCTL.ENCAnCRM[1:0] = 00B:
  Comparison is selected as the function of the ENCAnCCR0 and ENCAnCCR1 registers.
- ENCAnCTL.ENCAnECM[1:0] = 11B:
  If counting after a match between the values in the counter and the ENCAnCCR0 register is upwards, the counter is cleared.
  If further counting after a match between the values in the counter and the ENCAnCCR1 register is downwards, the counter is cleared.
- ENCAnCTL.ENCAnLDE = 0
  The counter is not loaded with the value from the ENCAnCCR0 register.

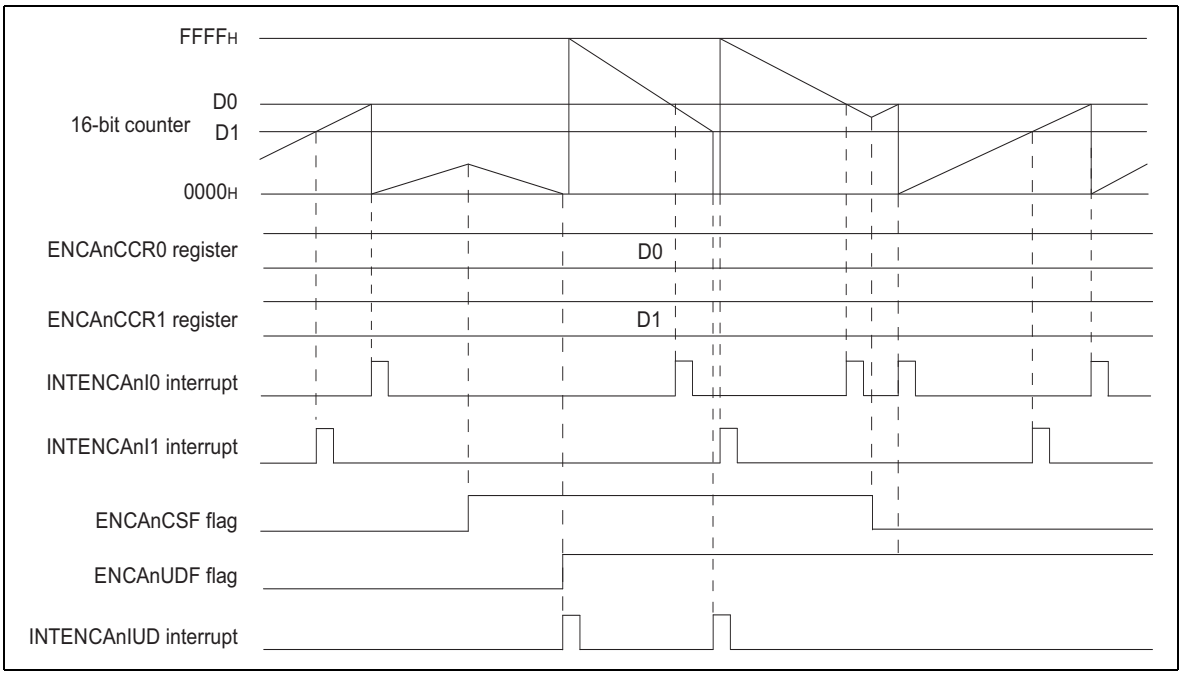


Figure 18-20 Timing of Basic Encoder Operation 3 (Encoder Comparison Mode 3)

A compare match interrupt (INTENCAnI0) is generated when the values of the counter and register ENCAnCCR0 (D0) match.
If further counting is upwards, the counter is cleared to 0000_H because ENCAnECM0 = 1.

A compare match interrupt (INTENCAnI1) is generated when the values of the counter and register ENCAnCCR1 (D1) match.
If further counting is downwards, the counter is cleared to 0000_H because ENCAnECM1 = 1.

(4) Timing of Basic Encoder Operation 4 (Encoder Capture Mode)

<Setting condition>

- ENCAnCTL.ENCAnCRM[1:0] = 11B:
  Capture is selected as the function of the ENCAnCCR0 and ENCAnCCR1 registers.
- ENCAnCTL.ENCAnECM[1:0] = 00B:
  The counter is not cleared on a match between its value and that of the ENCAnCCR0 register.
- ENCAnCTL.ENCAnLDE = 0:
  The setting from the ENCAnCCR0 register is not loaded to the counter.
- ENCAnIOC1.ENCAnSCE = 0, ENCAnECS[1:0] = 00B:
  Input on the ENCAnEC pin does not lead to edge detection.
- ENCAnIOC0.ENCAnTIS[3:2] = 01B:
  Selects detection of rising edges of the signal on the ENCAnI1 pin.
- ENCAnIOC0.ENCAnTIS[1:0] = 01B:
  Selects detection of rising edges of the signal on the ENCAnI0 pin.

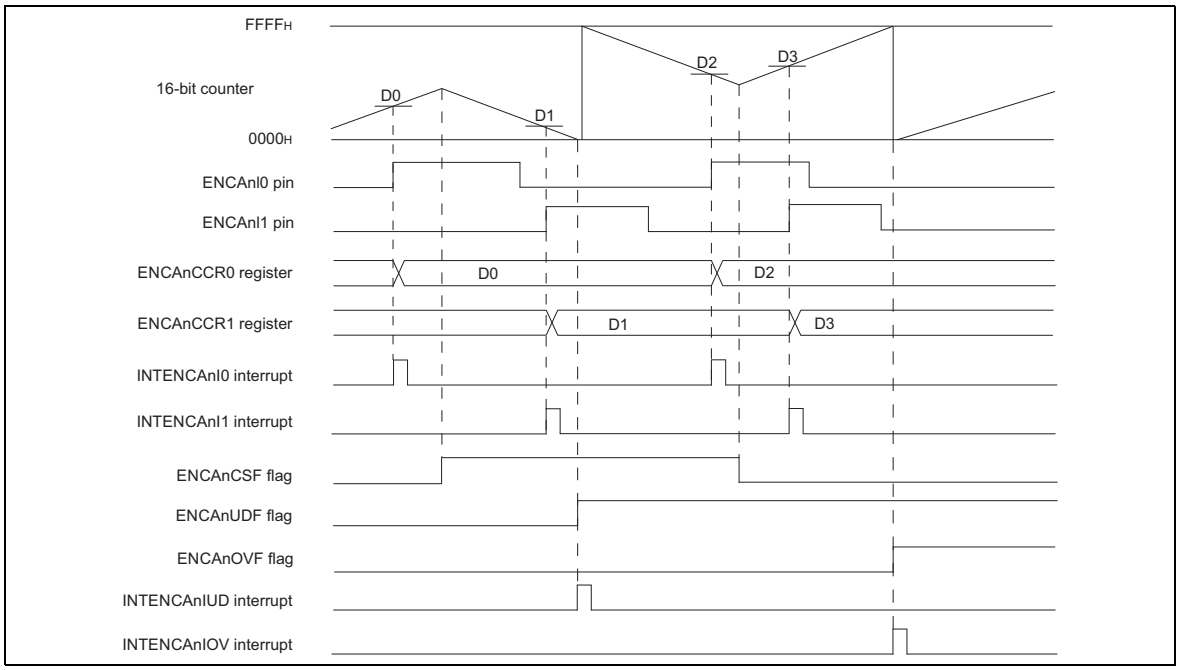


Figure 18-21 Timing of Basic Encoder Operation 4 (Encoder Capture Mode)

Detection of a rising edge on the ENCAnI0 pin leads to storage of the counter value in the capture register (ENCANCCR0) and the generation of a capture interrupt (INTENCAnI0).

Detection of a rising edge on the ENCAnI1 pin leads to storage of the counter value in the capture register (ENCANCCR1) and the generation of a capture interrupt (INTENCAnI1).

Overflow interrupts (INTENCAnIOV) or underflow interrupts (INTENCAnIUD) are generated in response to overflows or underflows of the counter.

(5) Timing of Basic Encoder Operation 5 (Encoder Capture and Comparison Mode)

<Setting condition>

- ENCAnCTL.ENCAnCRM[1:0] = 10B:
  Select the comparison function for the ENCAnCCR0 register and the capture function for the ENCAnCCR1 register.
- ENCAnCTL.ENCAnECM[1:0] = 01B:
  The counter is cleared when its value matches that of the ENCAnCCR0 register.
- ENCAnCTL.ENCAnLDE = 1:
  When the counter underflows, it is loaded with the value from the ENCAnCCR0 register.
- ENCAnIOC1.ENCAnSCE = 0, ENCAnECS[1:0] = 00B
- ENCAnIOC0.ENCAnTIS[3:2] = 11B
  Selects detection of both edges of the signal on the ENCAnI1 pin.

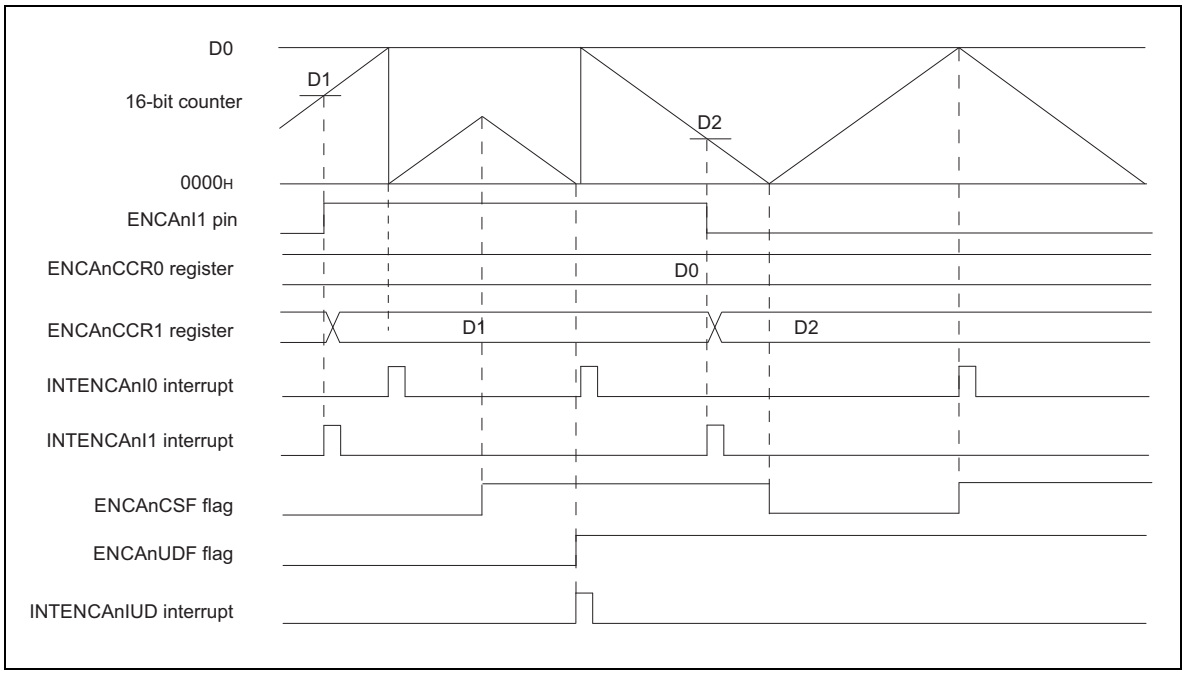


Figure 18-22 Timing of Basic Encoder Operation 5 (Encoder Capture and Comparison Mode)

A compare match interrupt (INTENCAnI0) is generated when the values of the counter and register ENCAnCCR0 (D0) match.
If further counting is upwards, the counter is cleared to 0000_H because ENCAnECM0 = 1.

Detection of both edges on the ENCAnI1 pin leads to storage of the counter value in the capture register (ENCANCCR1) and the generation of a capture interrupt (INTENCAnI1).

An underflow interrupt (INTENCAnIUD) is generated when the counter underflows.
ENCAnLDE = 1, so the counter is loaded with the value from the ENCAnCCR0 register (D0) when the counter underflows

ENCAnLDE = 1 and ENCAnECM[1:0] = 01B, so counting is from 0000_H to the setting of the ENCAnCCR0 register.

Section 19 Timer Option Module (TAPA)

The timer option module is described in this section.

19.1 Features of the Timer Option Module

Instances This product has the following number of instances of timer option modules.

Table 19-1 Instances of Timer Option Modules

Timer Option	
Instances	4
Name	TAPAn

Instances index n Throughout this section, the individual instances of the timer option function are identified by the index “n” (n = 0 to 3). For example, TAPAnFLG indicates the TAPAn flag registers.

Register addresses All TAPAn register addresses are given as address offsets from the given base address, <TAPAn_base0> or <TAPAn_base1>.
Each of the TAPAn base addresses is listed in the following table.

Table 19-2 Register Base Address

TAPAn	<TAPAn_base0> Address	<TAPAn_base1> Address
TAPA0	FF81 5000$_H$	FFFF D400$_H$
TAPA1	FF81 6000$_H$	FFFF D500$_H$
TAPA2	FF81 7000$_H$	FFFF D600$_H$
TAPA3	FF81 8000$_H$	FFFF D700$_H$

Clock supply The following clock signal is input to the timer option.

Table 19-3 TAPA Clock Supply

TAPAn	TAPAn Clock	Connected to
TAPA0	PCLK	Clock controller
TAPA1		
TAPA2		
TAPA3		

Interrupt The timer option module can generate the following interrupt requests.

Table 19-4 TAPAn Interrupt Requests

TAPAn Signals	Function	Connected to
TAPA0		
TAPA0TIPEK0	TAPA0 peak interrupt	INTTAPA0IPEK0 signal for the interrupt controller
TAPA0TIPEK1	TAPA0 peak interrupt	-
TAPA0TIVLY0	TAPA0 valley interrupt	INTTAPA0IVLY0 signal for the interrupt controller
TAPA0TIVLY1	TAPA0 valley interrupt	-
TAPA1		
TAPA1TIPEK0	TAPA1 peak interrupt	INTTAPA1IPEK0 signal for the interrupt controller
TAPA1TIPEK1	TAPA1 peak interrupt	-
TAPA1TIVLY0	TAPA1 valley interrupt	INTTAPA1IVLY0 signal for the interrupt controller
TAPA1TIVLY1	TAPA1 valley interrupt	-
TAPA2		
TAPA2TIPEK0	TAPA2 peak interrupt	-
TAPA2TIPEK1	TAPA2 peak interrupt	-
TAPA2TIVLY0	TAPA2 valley interrupt	-
TAPA2TIVLY1	TAPA2 valley interrupt	-
TAPA3		
TAPA3TIPEK0	TAPA3 peak interrupt	-
TAPA3TIPEK1	TAPA3 peak interrupt	-
TAPA3TIVLY0	TAPA3 valley interrupt	-
TAPA3TIVLY1	TAPA3 valley interrupt	-

Internal signals The internal signal connections of the timer option module are listed in the following table.

Table 19-5 TAPAn Internal Signal (1/3)

TAPAn Signals	Function	Connected to
TAPA0		
TAPA0THASIN	Asynchronous Hi-Z control input signal	PIC
TAPA0TSIM0	TAU-A master channel 0 interrupt input signal	PIC
TAPA0TSIM1	TAU-A master channel 0 interrupt input signal	-
TAPA0TUDCM0	TAU-A master channel 0 up/down input signal	PIC
TAPA0TUDCM1	TAU-A master channel 0 up/down input signal	-
TAPA0TCDENS0	TAU-A slave channel 0 match detection input signal	PIC
TAPA0TCDENS1	TAU-A slave channel 1 match detection input signal	PIC
TAPA0TCDENM0	TAU-A master channel 0 cycle detection input signal	-
TAPA0TCDENM1	TAU-A master channel 1 cycle detection input signal	-
TAPA0TOEM0	TAU-A master channel 0 timer enabled input signal	-
TAPA0TOEM1	TAU-A master channel 1 timer enabled input signal	-
TAPA0TADOUT0	A/D converter trigger output signal 0	ADCA0
TAPA0TADOUT1	A/D converter trigger output signal 1	ADCA0
TAPA1		
TAPA1THASIN	Asynchronous Hi-Z control input signal	PIC
TAPA1TSIM0	TAU-A master channel 0 interrupt input signal	PIC
TAPA1TSIM1	TAU-A master channel 0 interrupt input signal	-
TAPA1TUDCM0	TAU-A master channel 0 up/down input signal	PIC
TAPA1TUDCM1	TAU-A master channel 0 up/down input signal	-
TAPA1TCDENS0	TAU-A slave channel 0 match detection input signal	PIC
TAPA1TCDENS1	TAU-A slave channel 1 match detection input signal	PIC
TAPA1TCDENM0	TAU-A master channel 0 cycle detection input signal	-

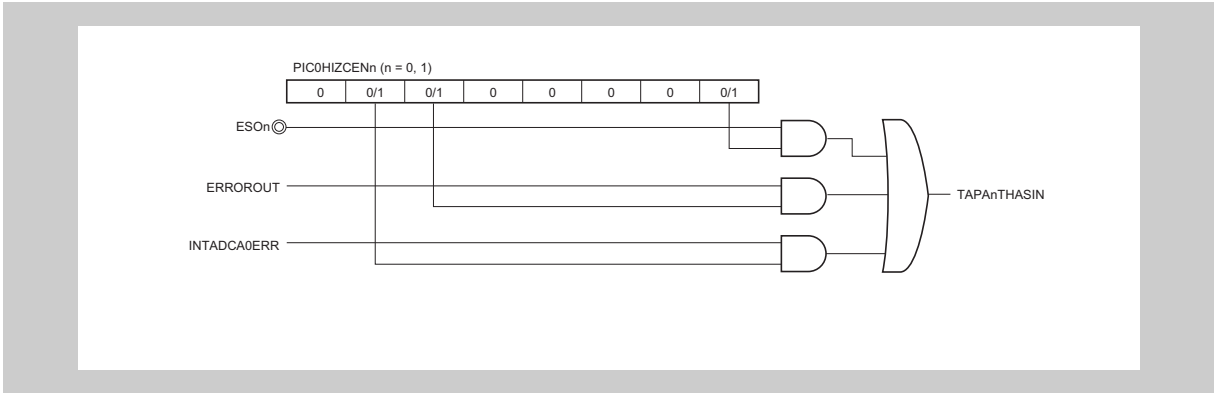
Table 19-5 TAPAn Internal Signal (2/3)

TAPAn Signals	Function	Connected to
TAPA1TCDENM1	TAU-A master channel 1 cycle detection input signal	-
TAPA1TOEM0	TAU-A master channel 0 timer enabled input signal	-
TAPA1TOEM1	TAU-A master channel 1 timer enabled input signal	-
TAPA1TADOUT0	A/D converter trigger output signal 0	ADCA0
TAPA1TADOUT1	A/D converter trigger output signal 1	ADCA0
TAPA2		
TAPA2THASIN	Asynchronous Hi-Z control input signal	PIC
TAPA2TSIM0	TAU-A master channel 0 interrupt input signal	-
TAPA2TSIM1	TAU-A master channel 0 interrupt input signal	-
TAPA2TUDCM0	TAU-A master channel 0 up/down input signal	-
TAPA2TUDCM1	TAU-A master channel 0 up/down input signal 0	-
TAPA2TCDENS0	TAU-A slave channel 0 match detection input signal 0	-
TAPA2TCDENS1	TAU-A slave channel 1 match detection input signal	-
TAPA2TCDENM0	TAU-A master channel 0 cycle detection input signal	-
TAPA2TCDENM1	TAU-A master channel 1 cycle detection input signal	-
TAPA2TOEM0	TAU-A master channel 0 timer enabled input signal	-
TAPA2TOEM1	TAU-A master channel 1 timer enabled input signal	-
TAPA2TADOUT0	A/D converter trigger output signal 0	-
TAPA2TADOUT1	A/D converter trigger output signal 1	-
TAPA3		
TAPA3THASIN	Asynchronous Hi-Z control input signal	PIC
TAPA3TSIM0	TAU-A master channel 0 interrupt input signal	-
TAPA3TSIM1	TAU-A master channel 0 interrupt input signal	-
TAPA3TUDCM0	TAU-A master channel 0 up/down input signal	-
TAPA3TUDCM1	TAU-A master channel 0 up/down input signal	-

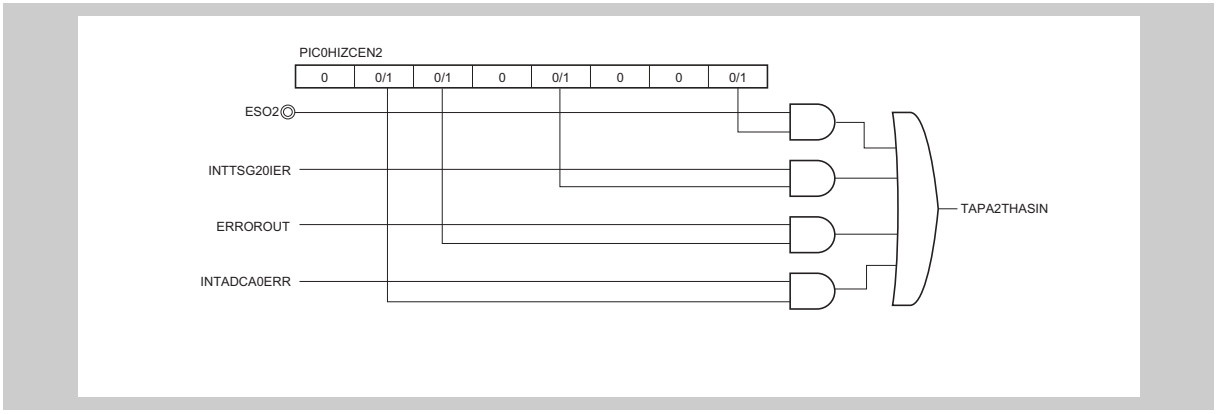
Table 19-5 TAPAn Internal Signal (3/3)

TAPAn Signals	Function	Connected to
TAPA3TCDENS0	TAU-A slave channel 0 match detection input signal	-
TAPA3TCDENS1	TAU-A slave channel 1 match detection input signal	-
TAPA3TCDENM0	TAU-A master channel 0 cycle detection input signal	-
TAPA3TCDENM1	TAU-A master channel 1 cycle detection input signal	-
TAPA3TOEM0	TAU-A master channel 0 timer enabled input signal	-
TAPA3TOEM1	TAU-A master channel 1 timer enabled input signal	-
TAPA3TADOUT0	A/D converter trigger output signal 0	-
TAPA3TADOUT1	A/D converter trigger output signal 1	-

(a) Hi-Z Control Signal for TAPA0 and TAPA1 (TAPA0THASIN and TAPA1THASIN)



(b) Hi-Z Control Signal for TAPA2 (TAPA2THASIN)



(c) Hi-Z Control Signal for TAPA3 (TAPA3THASIN)

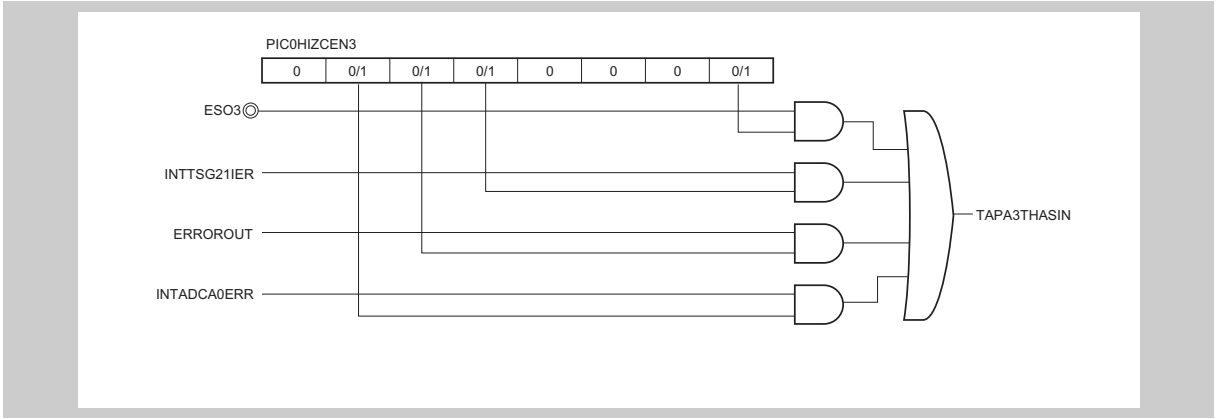


Figure 19-1 Hi-Z Control Signals for TAPA0 to TAPA3 (TAPAnTHASIN)

(1) Hi-Z Output Control Register 0 (PIC0HIZCEN0)

This register selects input signals to control Hi-Z output.

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

Address FF81 C0B4$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN06	PIC0HIZ CEN05	0	0	0	0	PIC0HIZ CEN00
R	R/W	R/W	R	R	R	R	R/W

Table 19-6 Contents of the PIC0HIZCEN0 Register

Bit Position	Bit Name	Function
6	PIC0HIZCEN06	Enables and disables Hi-Z output control by the INTADCA0TERR interrupt signal. 0: Disabled 1: Enabled
5	PIC0HIZCEN05	Enables and disables Hi-Z output control by the ERROROUT signal. 0: Disabled 1: Enabled
0	PIC0HIZCEN00	Enables and disables Hi-Z output control by input to the ESO0 pin. 0: Disabled 1: Enabled

Caution 1. Set the PIC0HIZCEN0 register before enabling asynchronous Hi-Z control.

Caution 2. When the ERROROUT signal is selected, select the rising edge (TAPA0CTL0.TAPA0DCN, TAPA0DCP = 01) as the valid edge.

Note Set undefined bits in PIC0HIZCEN0 to 0.

(2) Hi-Z Output Control Register 1 (PIC0HIZCEN1)

This register selects input signals to control Hi-Z output.

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

Address FF81 C0B8$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN16	PIC0HIZ CEN15	0	0	0	0	PIC0HIZ CEN10
R	R/W	R/W	R	R	R	R	R/W

Table 19-7 Contents of the PIC0HIZCEN1 Register

Bit Position	Bit Name	Function
6	PIC0HIZCEN16	Enables and disables Hi-Z output control by the INTADCA0TERR interrupt signal. 0: Disabled 1: Enabled
5	PIC0HIZCEN15	Enables and disables Hi-Z output control by the ERROROUT signal. 0: Disabled 1: Enabled
0	PIC0HIZCEN10	Enables and disables Hi-Z output control by input to the ESO1 pin. 0: Disabled 1: Enabled

Caution 1. Set the PIC0HIZCEN1 register before enabling asynchronous Hi-Z control.

Caution 2. When the ERROROUT signal is selected, select the rising edge (TAPA1CTL0.TAPA1DCN, TAPA1DCP = 01) as the valid edge.

Note Set the undefined bits in PIC0HIZCEN1 to 0.

(3) Hi-Z Output Control Register 2 (PIC0HIZCEN2)

This register selects input signals to control Hi-Z output.

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

Address FF81 C0DC$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN26	PIC0HIZ CEN25	0	PIC0HIZ CEN23	0	0	PIC0HIZ CEN20
R	R/W	R/W	R	R/W	R	R	R/W

Table 19-8 Contents of the PIC0HIZCEN2 Register

Bit Position	Bit Name	Function
6	PIC0HIZCEN26	Enables and disables Hi-Z output control by the INTADCA0TERR interrupt signal. 0: Disabled 1: Enabled
5	PIC0HIZCEN25	Enables and disables Hi-Z output control by the ERROROUT signal. 0: Disabled 1: Enabled
3	PIC0HIZCEN23	Enables and disables Hi-Z output control by the INTTSG20IER interrupt signal. 0: Disabled 1: Enabled
0	PIC0HIZCEN20	Enables and disables Hi-Z output control by input to the ESO2 pin. 0: Disabled 1: Enabled

Caution 1. Set the PIC0HIZCEN2 register before enabling asynchronous Hi-Z control.

Caution 2. When the ERROROUT signal is selected, select the rising edge (TAPA2CTL0.TAPA2DCN, TAPA2DCP = 01) as the valid edge.

Note Set the undefined bits in PIC0HIZCEN2 to 0.

(4) Hi-Z Output Control Register 3 (PIC0HIZCEN3)

This register selects input signals to control Hi-Z output.

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

Address FF81 C0E4$_H$

Initial value 00$_H$

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN36	PIC0HIZ CEN35	PIC0HIZ CEN34	0	0	0	PIC0HIZ CEN30
R	R/W	R/W	R/W	R	R	R	R/W

Table 19-9 Contents of the PIC0HIZCEN3 Register

Bit Position	Bit Name	Function
6	PIC0HIZCEN36	Enables and disables Hi-Z output control by the INTADCA0TERR interrupt signal. 0: Disabled 1: Enabled
5	PIC0HIZCEN35	Enables and disables Hi-Z output control by the ERROROUT signal. 0: Disabled 1: Enabled
4	PIC0HIZCEN34	Enables and disables Hi-Z output control by the INTTSG21IER interrupt signal. 0: Disabled 1: Enabled
0	PIC0HIZCEN30	Enables and disables Hi-Z output control by input to the ESO3 pin. 0: Disabled 1: Enabled

Caution 1. Set the PIC0HIZCEN3 register before enabling asynchronous Hi-Z control.

Caution 2. When the ERROROUT signal is selected, select the rising edge (TAPA3CTL0.TAPA3DCN, TAPA3DCP = 01) as the valid edge.

Note Set the undefined bits in PIC0HIZCEN3 to 0.

I/O signals I/O signals for the timer option are listed in the following table.

Table 19-10 I/O Signals of TAPAn

TAPAn Signals	Function	Connected to
TAPA0		
$\overline{\text{TAPA0THZOUT0}}$	Hi-Z control signal 0 (phase U)	Hi-Z control of phase U output signal from TAUA0 (TOUT10/ TOUT11)
$\overline{\text{TAPA0THZOUT1}}$	Hi-Z control signal 1 (phase V)	Hi-Z control of phase V output signal from TAUA0 (TOUT12/ TOUT13)
$\overline{\text{TAPA0THZOUT2}}$	Hi-Z control signal 2 (phase W)	Hi-Z control of phase W output signal from TAUA0 (TOUT14/ TOUT15)
TAPA1		
$\overline{\text{TAPA1THZOUT0}}$	Hi-Z control signal 0 (phase U)	Hi-Z control of phase U output signal from TAUA1 (TOUT10/ TOUT11)
$\overline{\text{TAPA1THZOUT1}}$	Hi-Z control signal 1 (phase V)	Hi-Z control of phase V output signal from TAUA1 (TOUT12/ TOUT13)
$\overline{\text{TAPA1THZOUT2}}$	Hi-Z control signal 2 (phase W)	Hi-Z control of phase W output signal from TAUA1 (TOUT14/ TOUT15)
TAPA2		
$\overline{\text{TAPA2THZOUT0}}$	Hi-Z control signal 0	Hi-Z control of TSG20O1 to TSG20O6 pins of TSG20
TAPA3		
$\overline{\text{TAPA3THZOUT0}}$	Hi-Z control signal 0	Hi-Z control of TSG21O1 to TSG21O6 pins of TSG21

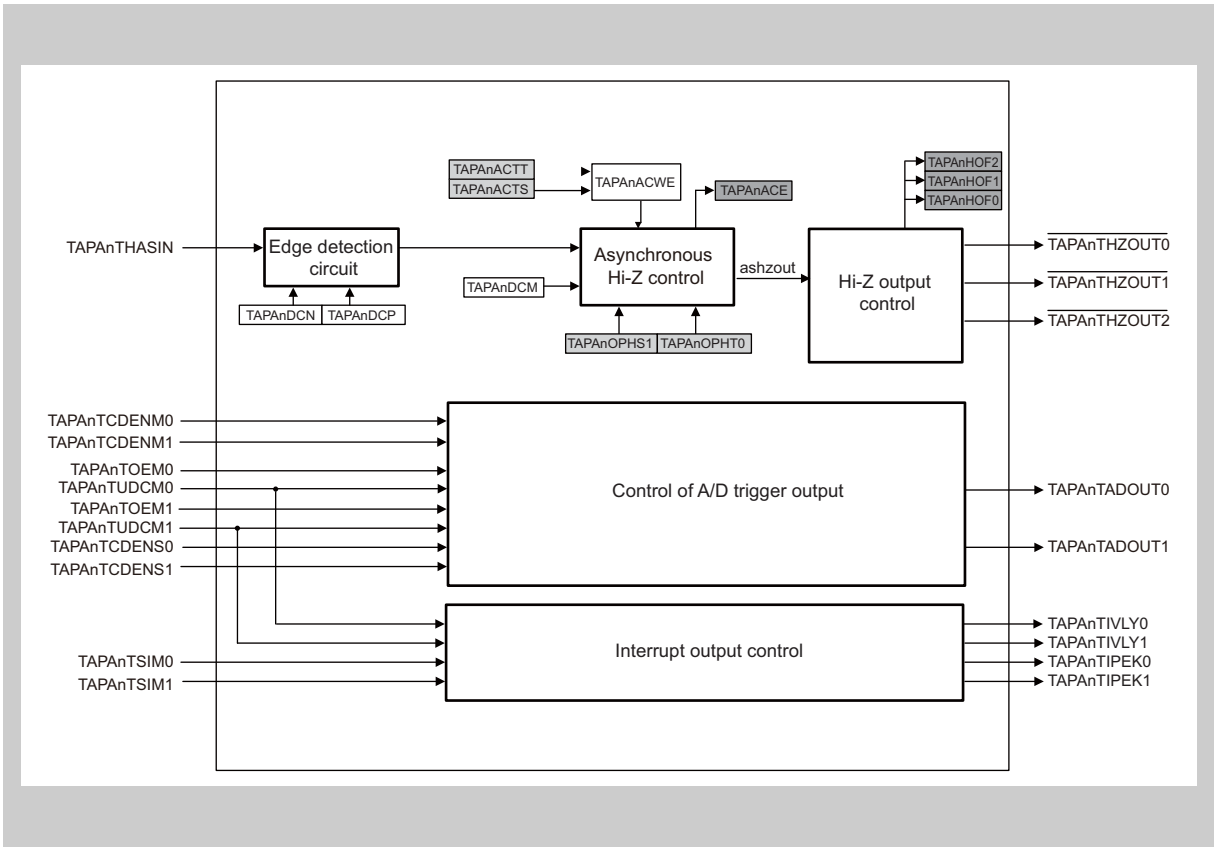
19.2 Functional Overview

Functional overview

The timer option module (TAPA) is for use with the timer array unit A (TAUA) and TSG2 modules. It provides optional functions for the timers.

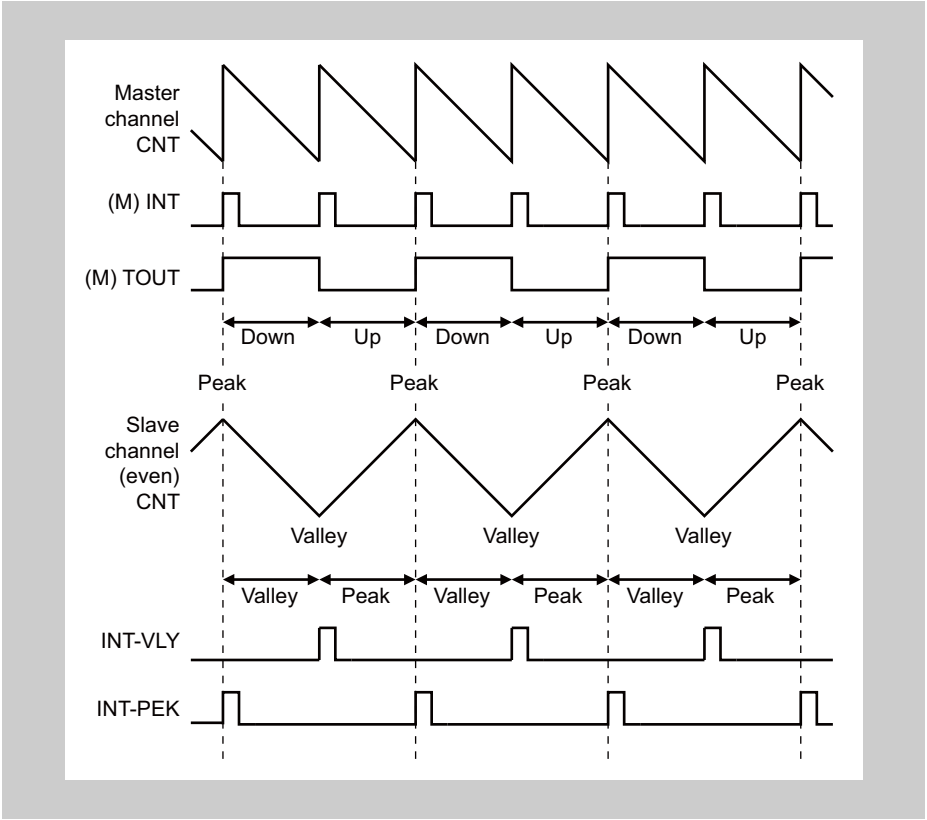
- Asynchronous Hi-Z control to deal with pin input
- Output of the INTn signal from the TAUA as a peak or valley interrupt is selectable.
- The INTn signal output by the TAUA provides the basis for the output of two conversion-trigger signals for the A/D converter.

19.2.1 Block Diagram



19.2.2 Peak and Valley of Timer Counter, and Peak and Valley Interrupts

In this document, the period from a TAUA down status (counting-down status) to generation of an interrupt by the master channel is defined as the valley period, and an interrupt generated to indicate this period is defined as a valley interrupt (INT-VLY). The period from a TAUA up status (counting-up status) to generation of an interrupt from the master channel is defined as the peak period, and an interrupt generated to indicate this period is defined as a peak interrupt (INT-PEK).



19.3 Registers

TAPAn is controlled and operated by means of the following registers.

19.3.1 Registers Overview

Table 19-11 List of Control Registers

Register Name	Symbol	Address
TAPAn control register 0	TAPAnCTL0	<TAPAn_base0> + 20_H
TAPAn control register 1 (n = 0, 1) *	TAPAnCTL1 (n = 0, 1) *	<TAPAn_base0> + 24_H (n = 0, 1) *
TAPAn flag register	TAPAnFLG	<TAPAn_base1> + 00_H
TAPAn asynchronous control write enable register	TAPAnACWE	<TAPAn_base1> + 04_H
TAPAn asynchronous control start trigger register	TAPAnACTS	<TAPAn_base1> + 08_H
TAPAn asynchronous control stop trigger register	TAPAnACTT	<TAPAn_base1> + $0C_H$
TAPAn Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base1> + 14_H
TAPAn Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base1> + 18_H

Note * TAPA2 and TAPA3 do not have TAPAn control register 1 (TAPAnCTL1) because their functions do not include A/D converter trigger selection.

<TAPAn_base> The base addresses <TAPAn_base> of the TAPAn are defined in the first part of this section under the heading “Register addresses”.

19.3.2 Registers Details

(1) TAPAnCTL0 - TAPAn Control Register 0

Control register 0 is used to control Hi-Z.

A value in this register can only be rewritten when TAPAnFLG.TAPAnASCE = 0, and TAUAnTE of the corresponding master channel of TAUA = 0.

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

Address <TAPAn_base0> + 20_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	TAPAnDCM	TAPAnDCN	TAPAnDCP	0	0
R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 19-12 Contents of the TAPAnCTL0 Register

Bit Position	Bit Name	Function
4	TAPAnDCM	Clearing condition specification bit This control bit specifies the condition for clearing of the Hi-Z control outputs. 0: Manipulation of TAPAnOPHS0 and TAPAnOPHT0 is enabled regardless of the TAPAnTHASIN signal input level. 1: Manipulation of TAPAnOPHT0 is disabled when the TAPAnTHASIN input signal is at the active level. Manipulation of TAPAnOPHT0 is enabled when the TAPAnTHASIN signal input is inactive.
3, 2	TAPAnDCN, TAPAnDCP	Hi-Z input edge selection bits These control bits specify the valid edge of TAPAnTHASIN.

TAPAn DCN	TAPAn DCP	Description
0	0	Does not detect valid edges
0	1	Detects a rising edge as the valid edge (active level = high)
1	0	Detects a falling edge as the valid edge (active level = low)
1	1	Setting prohibited

(2) TAPAnCTL1 - TAPAn Control Register 1

This register selects a trigger for the A/D converter.

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

Address <TAPAn_base0> + 24_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	TAPAn ATS3	TAPAn ATS2	TAPAn ATS1	TAPAn ATS0
R	R	R	R	R/W	R/W	R/W	R/W

Table 19-13 Contents of the TAPAnCTL1 Register

Bit Position	Bit Name	Function
3, 2	TAPAn ATS[3:2]	A/D converter trigger 1 select bits These control bits specify the signal for output as A/D converter conversion trigger output 1 (TAPAnTADOUT1).
1, 0	TAPAn ATS[1:0]	A/D converter trigger 0 select bits These control bits specify the signal for output as A/D converter conversion trigger output 0 (TAPAnTADOUT0).

TAPAn ATS3	TAPAn ATS2	Description
0	0	Outputs INT while the master channel is in the down state.
0	1	Outputs INT while the master channel is in the up state.
1	0	Outputs INT while the master channel is in the down/up state.
1	1	Outputs INT and a valley interrupt of the master channel (TAPAnTIVLY0) while the master channel is in the down/up state.

TAPAn ATS1	TAPAn ATS0	Description
0	0	Outputs INT while the master channel is in the down state.
0	1	Outputs INT while the master channel is in the up state.
1	0	Outputs INT while the master channel is in the down/up state.
1	1	Outputs INT and a valley interrupt of the master channel (TAPAnTIVLY0) while the master channel is in the down/up state.

(3) TAPAnFLG - TAPAn Flag Register

This flag register is used to control Hi-Z

Access This register can only be read in 16-bit units.

Address <TAPAn_base1> + 00_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	TAPAn HOF2	TAPAn HOF1	TAPAn HOF0	0	0	0	0	0	0	0	TAPAn ACE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19-14 Contents of the TAPAnFLG Register

Bit Position	Bit Name	Function
10 to 8	TAPAn HOFm	HZOUTm output monitor bit (m = 0 to 2) This bit monitors the TAPAnTHZOUTm output. 0: The TAPAnTHZOUTm output is at the low level. 1: The TAPAnTHZOUTm output is at the high level.
0	TAPAn ACE	Asynchronous Hi-Z control enable bit This bit indicates the state of asynchronous Hi-Z control (TAPAnTHASIN). 0: Asynchronous Hi-Z control is stopped. 1: Asynchronous Hi-Z control is enabled. The conditions for setting and clearing this bit are as follows. Clearing condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1. Setting condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1.

(4) TAPAnACWE - TAPAn Asynchronous Control Write Enable Register

This register enables writing for asynchronous Hi-Z control.

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

Address <TAPAn_base1> + 04_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn ACWE
R	R	R	R	R	R	R	R/W

Table 19-15 Contents of the TAPAnACWE Register

Bit Position	Bit Name	Function
0	TAPAn ACWE	Asynchronous control write enable bit This is a write-enable bit for asynchronous Hi-Z control. After 1 has been written to this bit, it is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

(5) TAPAnACTS - Asynchronous Control Start Trigger Register

This register enables the start trigger for asynchronous Hi-Z control.

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

Address <TAPAn_base1> + 08_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn ACTS
R	R	R	R	R	R	R	W

Table 19-16 Contents of the TAPAnACTS Register

Bit Position	Bit Name	Function
0	TAPAn ACTS	Asynchronous control start trigger bit This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is only valid when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Enables asynchronous Hi-Z control if TAPAnACE = 1.

(6) TAPAnACTT - TAPAn Asynchronous Control Stop Trigger Register

This register enables the stop trigger for asynchronous Hi-Z control.

Access This register can be read/written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base1> + $0C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn ACTT
R	R	R	R	R	R	R	W

Table 19-17 Contents of the TAPAnACTT

Bit Position	Bit Name	Function
0	TAPAn ACTT	Asynchronous control stop trigger bit This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is only valid when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Stops asynchronous Hi-Z control if TAPAnACE = 0.

(7) TAPAnOPHS - TAPAn Hi-Z Start Trigger Register

This register sets the start trigger for a Hi-Z control signal ($\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ or $\overline{\text{TAPAnTHZOUT2}}$).

Access This register can be read/written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base1> + 14_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn OPHS0
R	R	R	R	R	R	R	W

Table 19-18 Contents of the TAPAnOPHS Register

Bit Position	Bit Name	Function
0	TAPAn OPHS0	Hi-Z control signal start trigger 0 bit This bit sets the start trigger for a Hi-Z control signal. 0: The read value is 0 and writing 0 to this bit is ignored (no function). 1: Sets the corresponding Hi-Z control signal ($\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ or $\overline{\text{TAPAnTHZOUT2}}$) to the low level.

(8) TAPAnOPHT - TAPAn Hi-Z Stop Trigger Register

This register sets the stop trigger for a Hi-Z control signal ($\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ or $\overline{\text{TAPAnTHZOUT2}}$).

Access This register can be read/written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base1> + 18_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn OPHT0
R	R	R	R	R	R	R	W

Table 19-19 Contents of the TAPAnOPHT Register

Bit Position	Bit Name	Function
0	TAPAn OPHT0	Hi-Z control signal stop trigger 0 bit This bit sets the stop trigger for a Hi-Z control signal. 0: The read value is 0 and writing 0 to this bit is ignored (no function). 1: Sets a Hi-Z control signal ($\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ or $\overline{\text{TAPAnTHZOUT2}}$) to the high level.

19.4 Basic Functions

19.4.1 Hi-Z Control Functions

(1) Purpose of Hi-Z Control Function

Abnormal operation in timer motor-control under CPU control leads to rotation of the externally connected motor also becoming abnormal. In such a case, this function forcibly sets the motor control output to the Hi-Z state, independently of control by the CPU.

(2) Overview of Hi-Z Control Function

The following method is available for controlling Hi-Z states

- Asynchronous input Hi-Z control for pin inputs
  - Controls the Hi-Z control output signals (TAPAnTHZOUT0 (phase U), TAPAnTHZOUT1 (phase V) and TAPAnTHZOUT2 (phase W) asynchronously.

(3) Hi-Z Control and Its Operation

Function	Operation
Asynchronous Hi-Z control corresponding to pin input	This function detects analog inputs that are asynchronous and forcibly stops TOUTn output from the corresponding timer module (TAUA, TSG2) in response. Device outputs become Hi-Z while TAPAnTHASIN is active and until software sends a stop request.

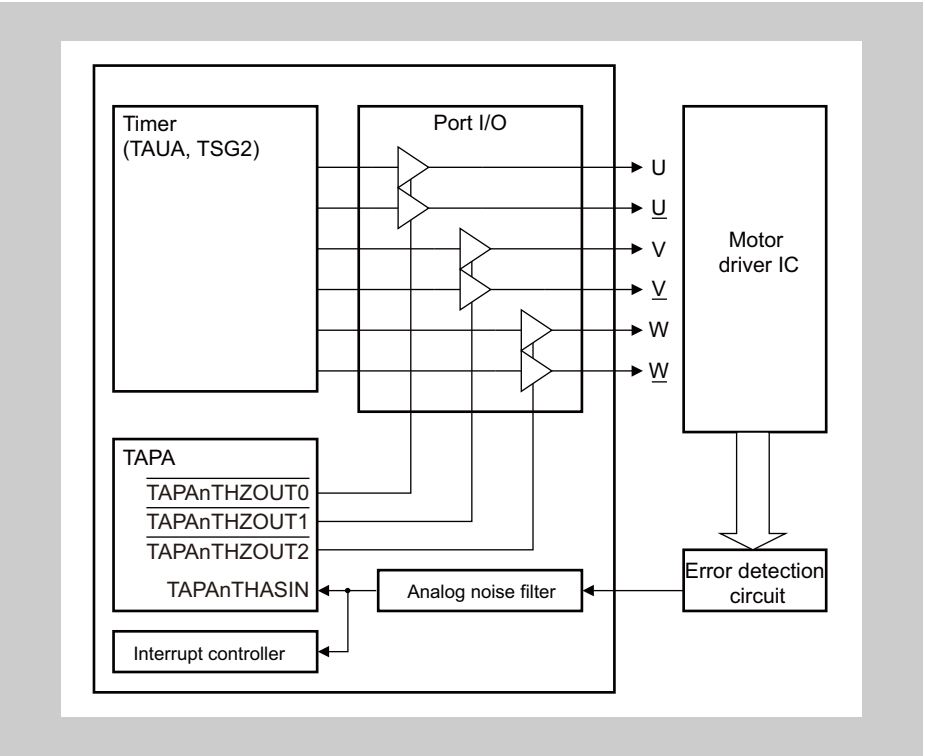
19.4.2 Asynchronous Hi-Z Control for Pin Inputs

(1) An Example of System Configuration of Asynchronous Hi-Z Control for Pin Inputs

When an external error detection signal is received, an interrupt is generated and, at the same time, the motor-driving signal output is set to the Hi-Z state.

This module assumes that microcontroller operation may hang when an error occurs. To handle such situations, external error detection signals are continuously processed so that the motor-driving signal can be set to the Hi-Z state even if no clock signal is being supplied.

This module only detects an error as an edge of the error-detection signal. A fixed output level is not detected as an error (the signal has no edge).

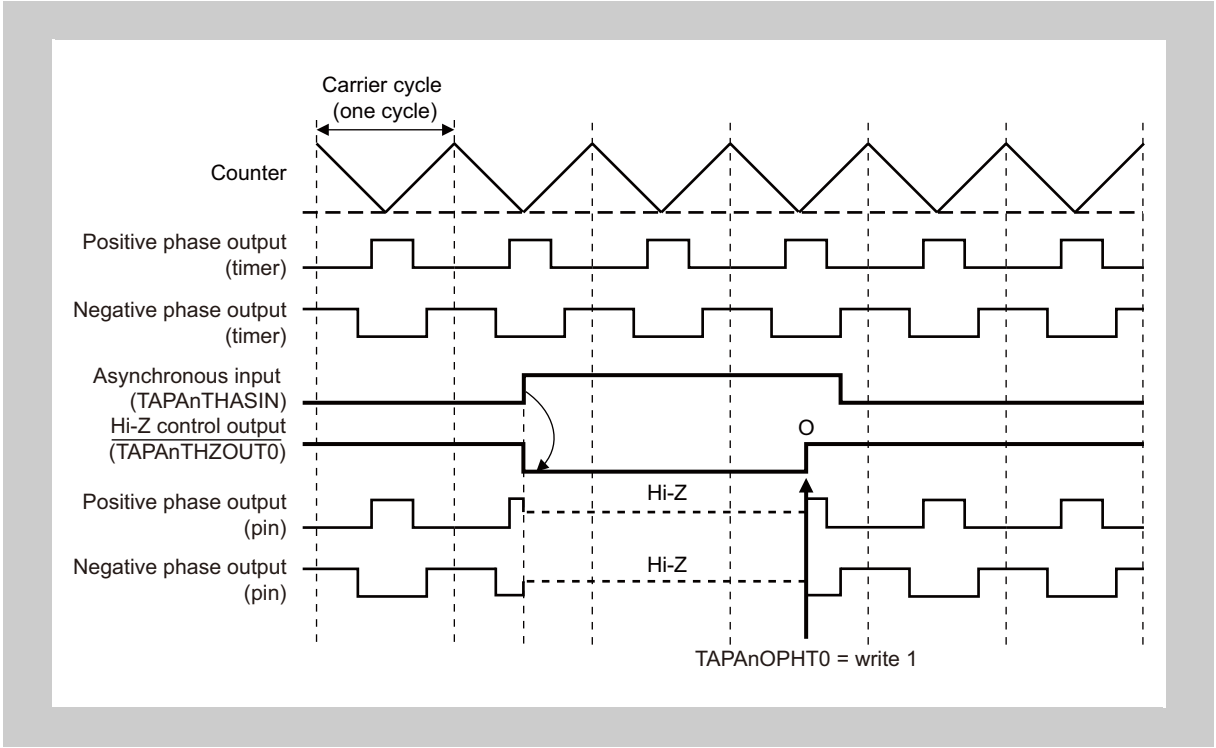


(2) Basic Operation in Hi-Z Control for Asynchronous Inputs

(a) Hi-Z Control when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 0, and TAPAnDCN = 0

$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level on detection of a valid edge of the asynchronous input (TAPAnTHASIN).

Output is forcibly stopped (by port control for Hi-Z output) as long as the $\overline{\text{TAPAnTHZOUT0}}$ output is at the low level.



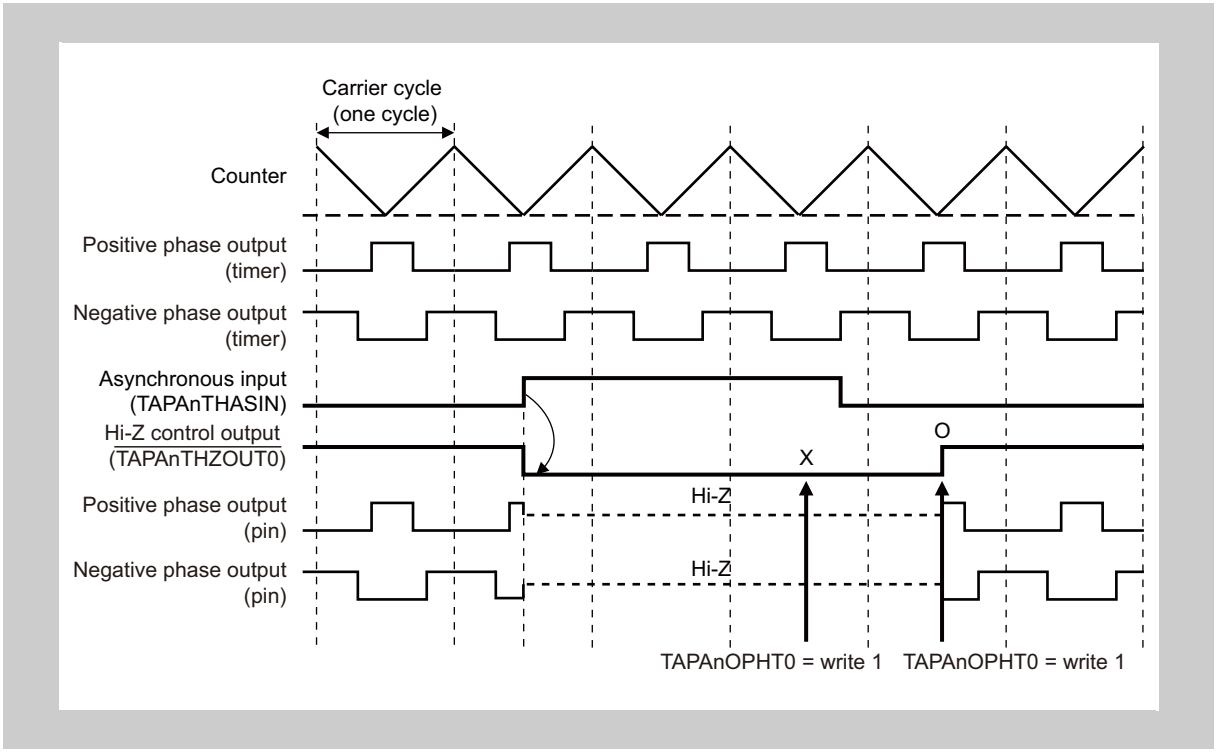
$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level in response to detection of a rising edge of the asynchronous input (TAPAnTHASIN).

$\overline{\text{TAPAnTHZOUT0}}$ goes to the high level in response to writing 1 to Hi-Z stop trigger 0 (TAPAnOPHT0), regardless of the level of TAPAnTHASIN.

(b) Hi-Z Control when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, TAPAnDCN = 0

$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level in response to detection of a valid edge of the asynchronous input signal (TAPAnTHASIN).

Output is forcibly stopped (by port control for Hi-Z output) as long as the $\overline{\text{TAPAnTHZOUT0}}$ output is at the low level.



$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level in response to detection of a rising edge of the asynchronous input signal (TAPAnTHASIN).

Writing of 1 to Hi-Z stop trigger 0 (TAPAnOPHT0) is ignored as long as the asynchronous input signal (TAPAnTHASIN) is at the active level (high because TAPAnDCP = 1).

After the asynchronous input signal (TAPAnTHASIN) is switched to the inactive level (low because TAPAnDCP = 1), $\overline{\text{TAPAnTHZOUT0}}$ goes to the high level when 1 is written to Hi-Z stop trigger 0 (TAPAnOPHT0).

(3) Software Operations for Asynchronous Input Hi-Z Control

This module allows software control of the output of Hi-Z control signals.

Hi-Z start trigger 0 (TAPAnOPHS0) and Hi-Z stop trigger 0 (TAPAnOPHT0) are used to control TAPAnTHZOUT0, TAPAnTHZOUT1, and TAPAnTHZOUT2.

(a) Operation of the Hi-Z Start Trigger (TAPAnOPHS)

TAPAn DCM	Operation
0/1	Writing 1 to the TAPAnOPHS0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1 and TAPAnTHZOUT2 signals at the low level.

(b) Operation of the Hi-Z Stop Trigger (TAPAnOPHT) during Hi-Z Control in Response to Asynchronous Input

The Hi-Z stop trigger operates as follows.

TAPAn DCM	Operation
0	Writing 1 to the TAPAnOPHT0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1 and TAPAnTHZOUT2 signals at the high level.
1	If TAPAnTHASIN is at the inactive level, writing 1 to the TAPAnOPHT0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1 and TAPAnTHZOUT2 signals at the high level. If TAPAnTHASIN is at the active level, writing of 1 to the TAPAnOPHT0 bit is ignored.

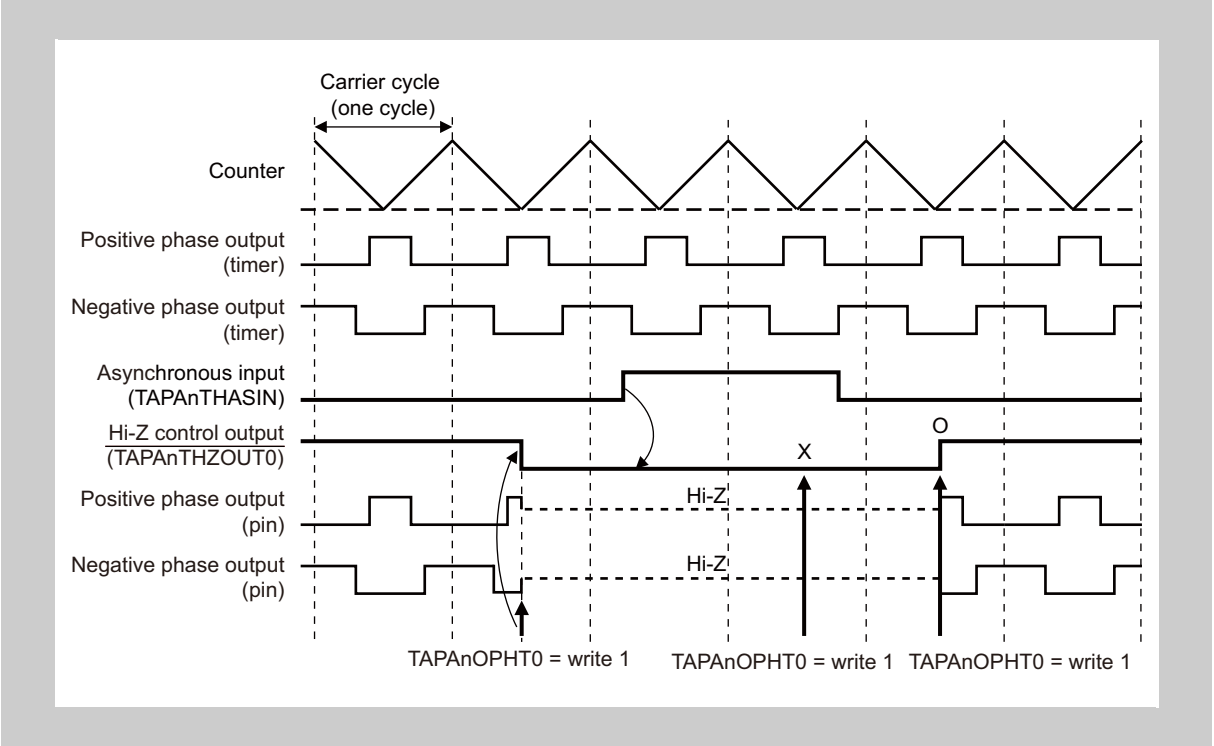


Figure 19-2 Operation of $\overline{\text{TAPAnTHZOUT0}}$ when TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0

$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level in response to the writing of 1 to TAPAnOPHS0.

After that, $\overline{\text{TAPAnTHZOUT0}}$ remains at the low level even if a rising edge of TAPAnTHASIN is detected.

Writing to TAPAnOPHT0 is ignored as long as TAPAnTHASIN is at the active level (high when TAPAnDCN = 0 and TAPAnDCP = 1).

After a falling edge of TAPAnTHASIN is detected, $\overline{\text{TAPAnTHZOUT0}}$ goes to the high level when 1 is written to TAPAnOPHT0 while TAPAnTHASIN is at the inactive level (low when DCN = 0 and DCP = 1).

(4) Operating Procedure Example for Hi-Z Control in Response to Asynchronous Input

An example of the operating procedure for Hi-Z control in response to asynchronous input is as follows (the table only covers settings for the timer option module because this operation does not depend on timer operations).

	Operation	State of TAPA
Initial settings	Setting in the TAPAnCTL0 register. Set TAPAnDCP and TAPAnDCN (input edge selection). Set TAPAnDCM (clearing mode selection).	Hi-Z control in response to asynchronous input is stopped (TAPAnFLG.TAPAnACE = 0).
Starting operation	Setting in the TAPAnACWE register: Set the TAPAnACWE bit to 1. Setting in the TAPAnACTS register: Set the TAPAnACTS bit to 1.	Writing to the TAPAnASTS bit is enabled. TAPAnFLG.TAPAnACE = 1, enabling Hi-Z control in response to asynchronous input.
During operation	To start Hi-Z control of an output from the timer: - Control is by the TAPAnOPHS0 bit of TAPA (if TAPAnDCM = 0) - Control is by the Hi-Z input signal (TAPAnTHASIN) for TAPA (if TAPAnDCM = 1) To stop Hi-Z control of output from the timer: - Control is by the TAPAnOPHT0 bit of TAPA (if TAPAnDCM = 0) - TAPAnOPHT0 is used if the Hi-Z input signal for TAPA (TAPAnTHASIN) is at the inactive level (if TAPAnDCM = 1) The state of TAPA operations can be read from the TAPAnFLG register at all times.	On detection of input of the starting edge of the Hi-Z input signal (TAPAnTHASIN) or setting of the start trigger bit (TAPAnOPHS0 = 1), the Hi-Z controller switches the $\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ and $\overline{\text{TAPAnTHZOUT2}}$ pins to low-level output. In accord with the operating mode settings in TAPAnDCM, the Hi-Z controller switches the $\overline{\text{TAPAnTHZOUT0}}$, $\overline{\text{TAPAnTHZOUT1}}$ and $\overline{\text{TAPAnTHZOUT2}}$ pins to high-level outputs in response to setting of the stop trigger bit (TAPAnOPHT0 = 1).
Stopping operation	Setting in the TAPAnACWE register: Set the TAPAnACWE bit to 1. Setting in the TAPAnACTT register. Set the TAPAnACTT bit to 1.	Writing to the TAPAnASTT bit is enabled. TAPAnFLG.ACE = 0, stopping asynchronous Hi-Z control

Restart operation

19.4.3 Selection of INT Signal Output

(1) Configuration of INT Signal Output Selection

This function outputs peak or valley interrupts by using the interrupt signal (TAPAnTSIM0) and up/down input signal (TAPAnTUDCM0) output from the triangle-wave carrier-cycle generation channel (master) of the TAUA module.

The TAPAnTSIM0 signal from the master that arrives while the TAPAnTUDCM0 signal from the master channel of the TAUA is at the high level is regarded as a peak interrupt and leads to output of the TAPAnTIPEK0 signal.

The TAPAnTSIM0 signal from the master that arrives while the TAPAnTUDCM0 signal from the master channel of the TAUA is at the low level is regarded as a valley interrupt and leads to output of the TAPAnTIVLY0 signal.

The PIC0REG2n0 register selects the use of the TAUA channel 0, 2, or 8 for the corresponding TAPAnTSIM0 and TAPAnTUDCM0 signals.

Timer I/O control register 2n0 (PIC0REG2n0)

Access Readable/writable in 32-bit units.

Address FF81 C08C$_H$ (n = 0), FF81 C09C$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	PIC0REG 2n025	PIC0REG 2n024
R	R	R	R	R	R	R/W	R/W

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
25, 24	PIC0REG2n025, PIC0REG2n024	These bits select the channel of TAUA to be used for TAPAnTSIM0 and TAPAnTUDCM0. 00: Not selected 01: TAUAn channel 0 is selected. 10: TAUAn channel 2 is selected. 11: TAUAn channel 8 is selected.

Caution Bits defined as 0 may also be defined in section 26, Peripheral Interconnection (PIC). In that case, the definition of the bit for the relevant function becomes applicable.

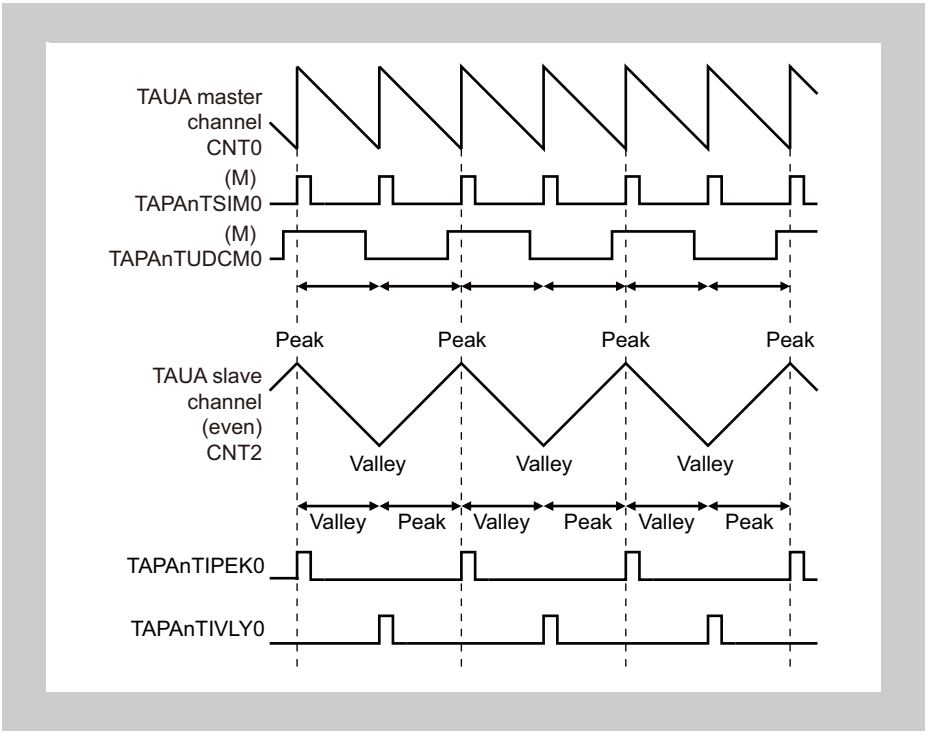


Figure 19-3 Example of Operations in Response to Master Channel 0 of TAUA

TAUA

- The master channel of TAUA generates a triangle-wave carrier-cycle.

  The TAUA module outputs the TAPAnTSIM0 signal and toggles the up/down signal on TAPAnTUDCM0 per half cycle of this carrier wave.

TAPA

- The TAPA module handles the TAPAnTSIM0 signal arriving while TAPAnTUDCM0 is high as a peak interrupt and outputs the TAPAnTIPEK0 signal in response.
- The TAPA module handles the TAPAnTSIM0 signal arriving while TAPAnTUDCM0 is low as a valley interrupt and outputs the TAPAnTIVLY0 signal in response.

Caution Combined circuits handle output of the TAPAnTIPEK0/TAPAnTIVLY0 signals and operate regardless of the operating mode.

When the TAPAnTIPEK0/TAPAnTIVLY0 signals are not in use, they must be controlled by using the interrupt mask bits as described under section 4.3, Interrupt Controller Control Registers.

(2) Operating Procedure for Selecting INT Signal Output

The procedure for selecting INT signal output is as follows.

	Operation	States of the TAUA and TAPA Modules
Initial settings	The TAPA module does not require initial settings.	TAUA and TAPA are stopped.
	Initialize the TAUA module. Determine the timer operating mode.	
Starting operation	Start the TAUA module.	TAUA starts counting.
During operation	TAUA runs in accord with the settings for the various functions.	The INT signal output selector outputs a peak interrupt (TAPAnTIPEK0) or a valley interrupt (TAPAnTIVLY0) for control cycle 0. This is based on interrupt input (TAPAnTSIM0) and up/down input (TAPAnTUDCM0) from TAUA.
Stopping operation	Stop the TAUA module.	TAUA stops counting.

Restart operation

19.4.4 Selecting a Trigger to Start Conversion by the A/D Converter

The TAPA is capable of producing triggers to start conversion by the A/D converter (TAPAnTADOUT0 and TAPAnTADOUT1). A trigger signal is produced from the INT and TOUT signals from the triangle-wave carrier-cycle generation channel (master) of the TAUA and the INT signal output from the channel selected for operation with the trigger to start conversion by the A/D converter.

(1) Configuration for Selecting the Triggers for Conversion by the A/D Converter

Table 19-20 Signals Used in Generating the TAPAnTADOUT Signals

Output Signal	Up/Down Input	Slave Match Detection Signal	Valley Interrupt Signal
TAPAnTADOUT0	TAPAnTUDCM0	TAPAnTCDENS0	TAPAnTIVLY0
TAPAnTADOUT1	TAPAnTUDCM0	TAPAnTCDENS1	TAPAnTIVLY0

Note

- For the up/down input and valley interrupt signals, refer to section 19.4.3, Selection of INT Signal Output.
- The slave match detection signal selects a slave channel according to the settings of the following registers.

For details on the individual registers, refer to section 26.4.2.3 (1) A/D Converter Trigger Output Control Register 4nj (PIC0ADTEN4nj).

TAPAnCDENS0: PIC0ADTEN4n1
TAPAnCDENS1: PIC0ADTEN4n2

Table 19-21 Operation of TAPAnCTL1.TAPAnATS[1:0] and TAPAnTADOUT0

TAPAn ATS1	TAPAn ATS0	Description
0	0	The INT signal from slave 0 is output as TAPAnTADOUT0 while master 0 of the TAUA module is in the down state.
0	1	The INT signal from slave 0 is output as TAPAnTADOUT0 while master 0 of the TAUA module is in the up state.
1	0	The INT signal from slave 0 of TAUA is output as TAPAnTADOUT0.
1	1	The INT and TAPAnTIVLY0 (valley interrupt signal 0) signals from slave 0 of TAUA are output as TAPAnTADOUT0.

Table 19-22 Operation of TAPAnCTL1.TAPAnATS[3:2] and TAPAnTADOUT1

TAPAn ATS3	TAPAn ATS2	Function
0	0	The INT signal from slave 1 is output as TAPAnTADOUT1 while master 0 of the TAUA module is in the down state.
0	1	The INT signal from slave 1 is output as TAPAnTADOUT1 while master 0 of the TAUA module is in the up state.
1	0	The INT signal from slave 1 of TAUA is output as TAPAnTADOUT1.
1	1	The INT and TAPAnTIVLY0 (valley interrupt signal 0) signals from slave 1 of TAUA are output as TAPAnTADOUT1.

(2) Waveforms in Control of A/D Converter Trigger Output in Triangle-Wave PWM Mode

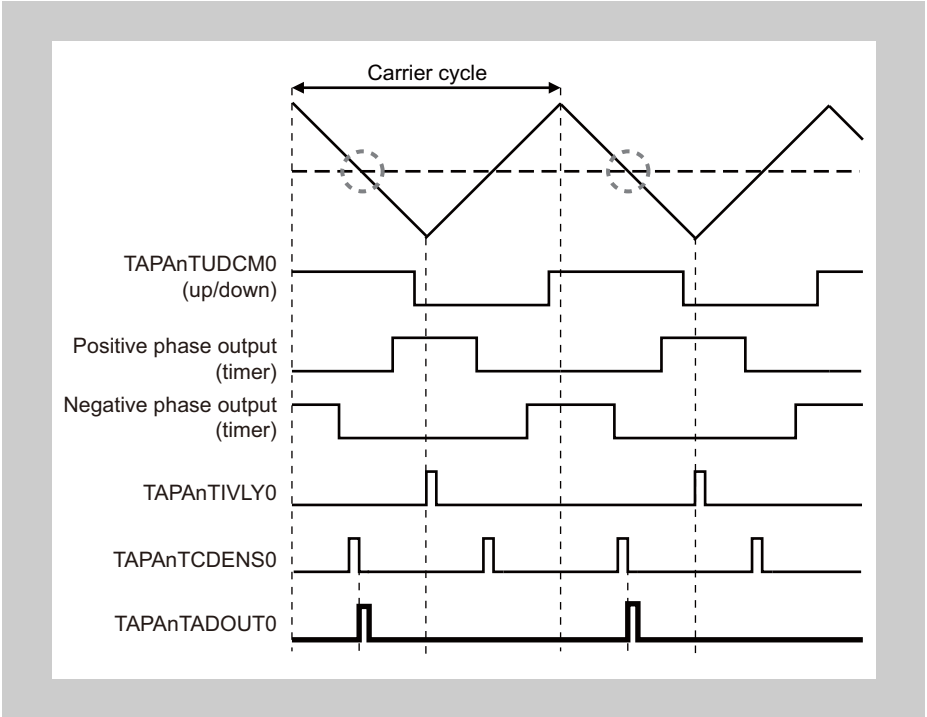


Figure 19-4 TAPAnATS = {0, 0}: INT Output while Master Channel is in the Down State

An INT signal from the slave while the master is in the down state is output as a trigger to start conversion by the A/D converter.

An INT signal from the slave while the master is in the up state is not output.

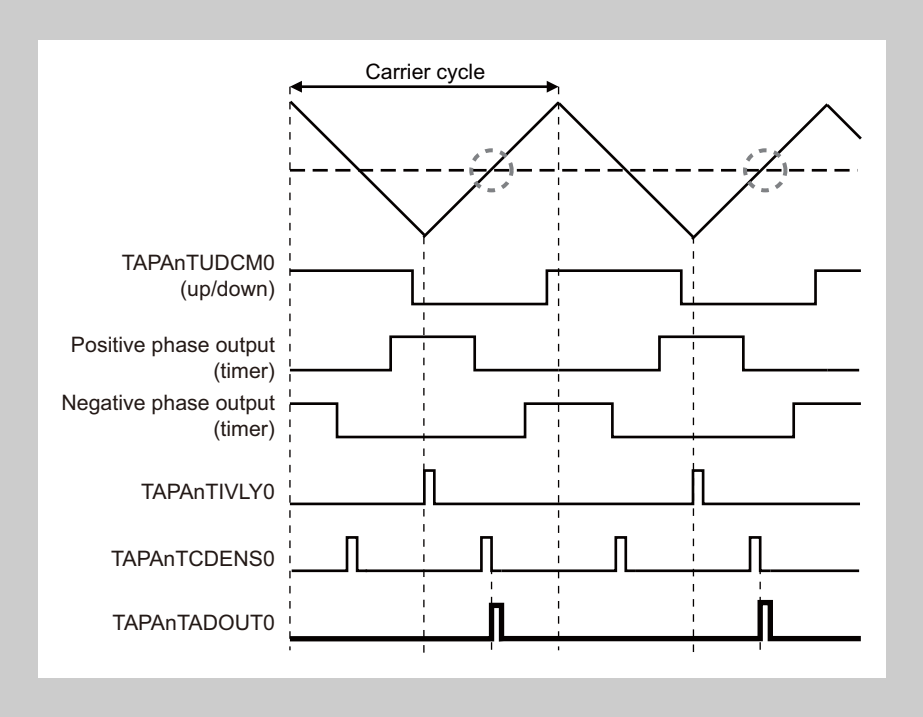


Figure 19-5 TAPAnATS = {0,1}: INT Output while Master Channel is in the Up State

An INT signal from the slave while the master is in the up state is output as a trigger to start conversion by the A/D converter.

An INT signal from the slave while the master is in the down state is not output.

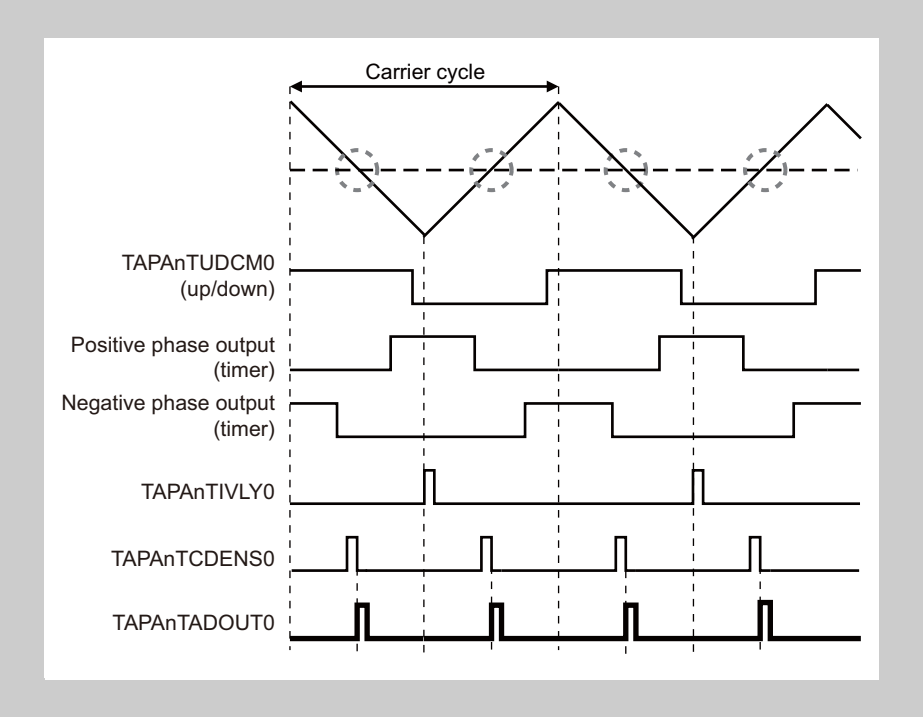


Figure 19-6 TAPAnATS = {1, 0}: INT Output while Master Channel is in either the Down or Up State

An INT signal from the slave at any time is output as a trigger to start conversion by the A/D converter.

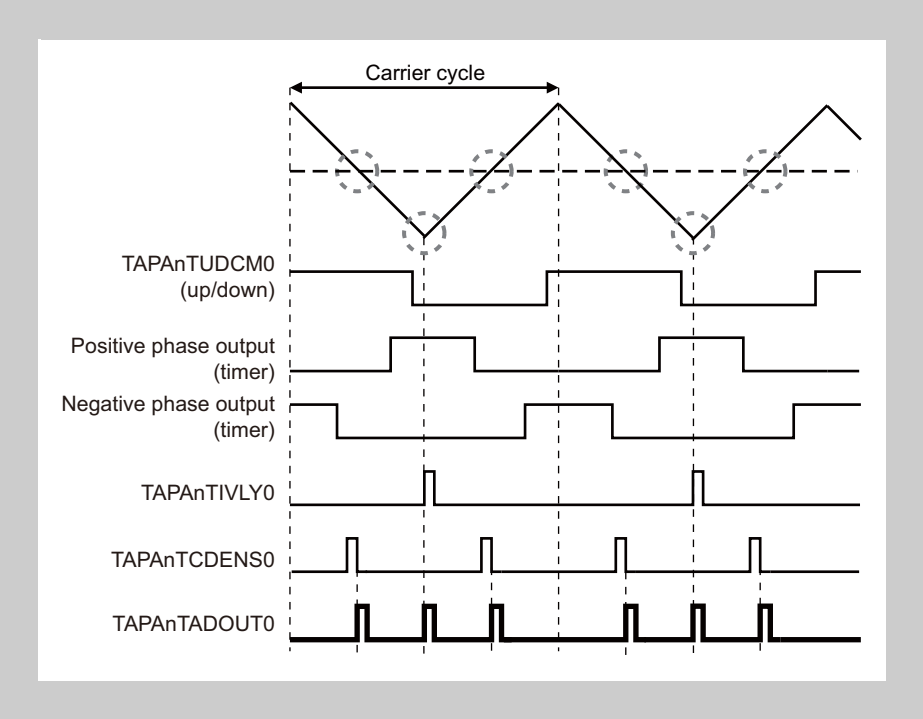


Figure 19-7 TAPAnATS = {1, 1}: Output of INT and Valley Interrupt while Master Channel is in either the Down or Up State

An INT signal or valley interrupt signal from the slave is output as a trigger to start conversion by the A/D converter.

(3) Operating Procedure for Setting a Trigger to Start Conversion by the A/D Converter

The operating procedure for setting a trigger to start conversion by the A/D converter is as follows.

	Operation	States of TAUA and TAPA
Initial settings	Initialize TAUA. Determine the timer's operating mode.	TAUA and TAPA are stopped.
	Setting in the TAPAnCTL1 register. Set TAPAnATS[1:0] bits (for TAPAnTADOUT0). Set TAPAnATS[3:2] (for TAPAnTADOUT1).	
Start operation	Start the TAUA module.	TAUA starts counting.
During operation	TAUA runs in accord with the settings for the various functions.	The A/D converter conversion trigger selector outputs TAPAnTADOUT0 in accord with the settings of TAPAnATS[1:0] or TAPAnTADOUT1 in accord with the settings of TAPAnATS[3:2], based on the slave match detection signal (TAPAnTCDENS1 or TAPAnTCDENS0) and up/down input (TAPAnTUDCM0) from the TAUA and the valley interrupt signal (TAPAnTIVLY0) generated by TAPA.
Stopping operation	Stopping the TAUA module.	TAUA stops counting.

Restart operation

Section 20 CAN Controller (FCN)

This product features on-chip CAN (Controller Area Network) controller that complies with the CAN protocol as standardized in ISO 11898.

This section contains a generic description of the CAN controller (FCN).

The first part describes instances, register base addresses, input/output signal names, etc.

20.1 FCN Features

Instances

This product has two instances of the CAN controller.

Table 20-1 Instances of FCN

CAN Controller	
Instance	2
Name	FCN0, FCN1

Instances index n

Throughout this section, the instance of a CAN controller is identified by the index “n” (n = 0, 1), for example, FCN0GMCLCTL for the FCN0 control register.

Table 20-2 Message Buffers of FCNn

FCNn Instance	Number m of Message Buffers
FCN0	64
FCN1	64

Message buffers index m

Throughout this section, the FCN message buffer registers are identified by “m” (m: 000 to 063), for example FCNnMmDAT4B for FCN instance n, message data byte 4 of message buffer register m.

Register address

All CAN controller register addresses are given as address offsets to the individual base address <FCNn_base>.

The <FCNn_base> address of each FCNn is given in Table 20-3.

Table 20-3 Register Base Addresses <FCNn_base>

FCNn Instance	<FCNn_base> Address
FCN0	FF48 0000_H
FCN1	FF4A 0000_H

Clock supply The following clock is input to the CAN controller.

Table 20-4 FCNn Cock Supply

FCNn	FCNn Reference Clock (f_{CAN})	Connected to
FCN0, FCN1	PCLK	Clock controller

Interrupts The interrupts of the CAN controllers are listed in Table 20-5.

Table 20-5 Interrupts of the CAN Controller

FCNn Signals	Function	Connected to
INT0ERR	FCN0 error detection	INTFCN0ERR
INT0REC	FCN0 reception completion	INTFCN0REC
INT0TRX	FCN0 transmission completion	INTFCN0TRX
INT1ERR	FCN1 error detection	INTFCN1ERR
INT1REC	FCN1 reception completion	INTFCN1REC
INT1TRX	FCN1 transmission completion	INTFCN1TRX
INT0WUP	FCN0 sleep wake-up/transmission abortion	INTFCN0WUP
INT1WUP	FCN1 sleep wake-up/transmission abortion	INTFCN1WUP

I/O signals The I/O signals of the CAN controllers are listed in the table below.

Table 20-6 I/O Signals of the CAN Controller

FCNn Signals	Function	Connected to
CANRXD0	FCN0 CAN bus receive input	Port FCN0RX
CANTXD0	FCN0 CAN bus transmit output	Port FCN0TX
CANRXD1	FCN1 CAN bus receive input	Port FCN1RX
CANTXD1	FCN1 CAN bus transmit output	Port FCN1TX

20.2 Features

- Compliant with ISO 11898
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max.
- 64 message buffers per channel
- Receive/transmit history list function
  (with enable flag for each message buffer individually)
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of 8 patterns is possible for each channel, applicable for data and remote frames
- Data bit time, communication baud rate, and sample point can be controlled by FCN module bit-rate prescaler register (FCNnCMBRPRS) and bit-rate register (FCNnCMBTCTL)
  - As an example, the following sample-point configurations can be configured:
    66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
  - Baud rates in the range of 10 kbps up to 1 Mbps can be configured.
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer.
  - A transmission request can be aborted by clearing the dedicated transmit-request flag of the concerned message buffer. Supported by transmission abort interrupt, on successful abortion.
  - Automatic block transmission operating mode (ABT)
  - Time stamp function of FCN channels 0 and 1, collaborated with timers capture channels
  - Centralized global data new bit monitor register, collecting all data new flags to be read from one location

20.2.1 Overview of Functions

table 20-7, Overview of Functions, presents an overview of the CAN controller functions.

Table 20-7 Overview of Functions

Function	Description
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (FCN's minimum reference clock input = 16 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	• 64 message buffers per channel • 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 8 patterns is possible for each channel, applicable for data and remote frames • A receive completion interrupt, which can be enabled or disabled for each message buffer, is generated every 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, with enable flag for each message buffer individually • Centralized global data new bit monitor register
Message transmission	• Unique ID can be set to each message buffer. • Transmit completion interrupt can be enabled or disabled for each message buffer. • Transmission abort interrupt and -flag for each message buffer (only one transmission of any buffer can be aborted at a time) • Message buffer number 0 to 63 specified as the transmit message buffer can be set for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as "ABT")). • Transmit history list function, with enable flag for each message buffer individually
Remote frame processing	• Remote frame processing by transmit message buffer • Remote frame processing by receive message buffer, when applying one of the 8 masks
Time stamp function	• The time stamp function can be set for a message reception when a 16-bit timer is used in combination. • Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected).
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
Release from bus-off state	• Forced release from bus-off possible by software • No automatic release from bus-off (software must re-enable).
Power save mode	• CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

20.2.2 Configuration

The CAN controller is composed of the following four blocks.

- Peripheral bus interface
  This functional block provides a peripheral bus interface and means of transmitting and receiving messages between the FCN module and the host CPU.
- MCM (Message Control Module)
  This functional block controls access to the CAN protocol layer and to the CAN RAM within the FCN module.
- CAN protocol layer
  This functional block is involved in the operation of the CAN protocol and its related settings.
- CAN RAM
  This is the CAN memory functional block, which is used to store message IDs, message data, etc.

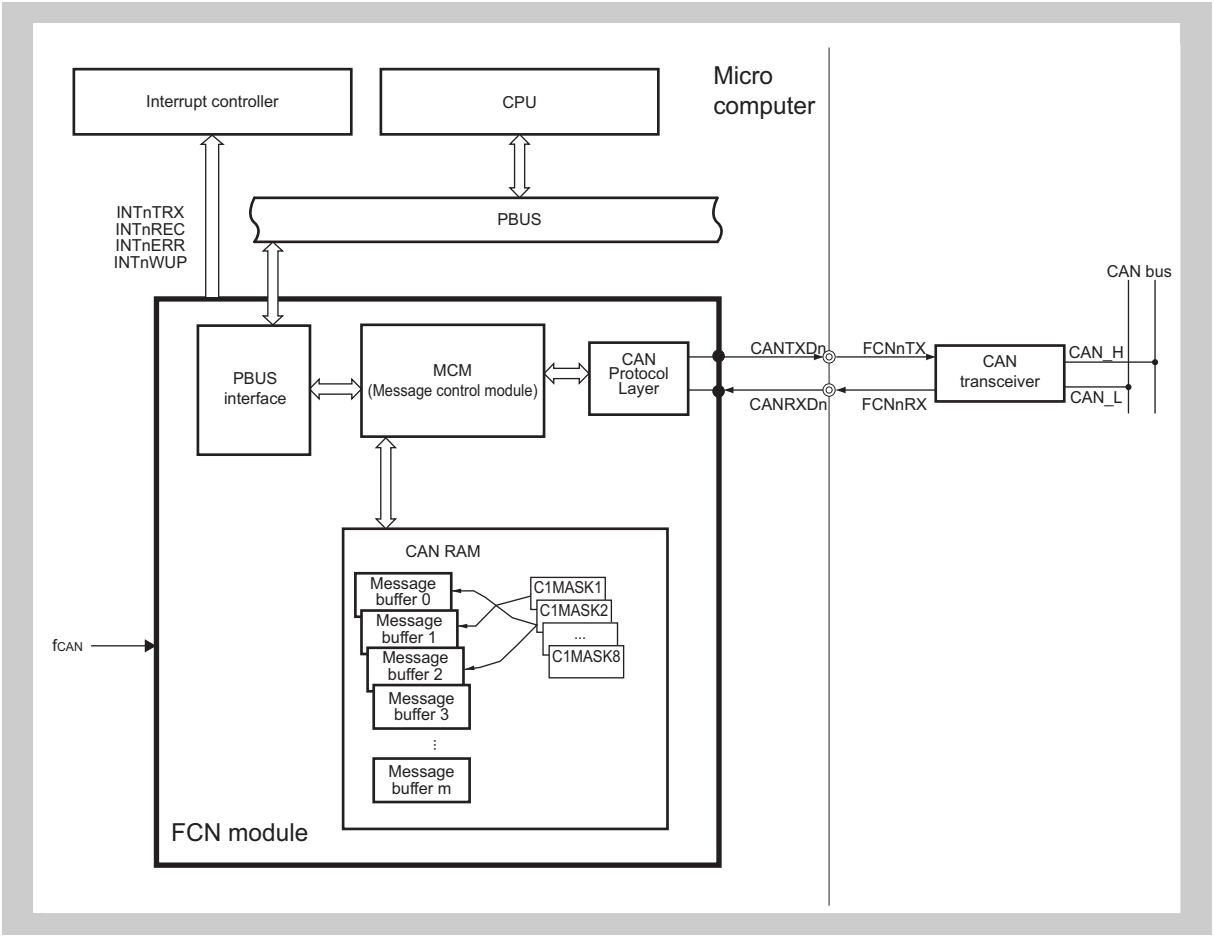


Figure 20-1 Block Diagram of the CAN Controller

20.3 Internal Registers of FCN

20.3.1 CAN Controller Configuration

Table 20-8 List of FCN Registers (1/2)

Item	Register Name
FCNn global registers	FCNn global control register (FCNnGMCLCTL)
	FCNn global clock selection register (FCNnGMCSPRE)
	FCNn global automatic block transmission control register (FCNnGMABCTL)
	FCNn global automatic block transmission delay setting register (FCNnGMADCTL)
	FCNn global data new bit monitor registers (FCNnDNBMRX0, FCNnDNBMRX1)
FCNn module registers	FCNn module mask 1 registers (FCNnCMMKCTL01H, FCNnCMMKCTL02H, FCNnCMMKCTL01W)
	FCNn module mask 2 registers (FCNnCMMKCTL03H, FCNnCMMKCTL04H, FCNnCMMKCTL03W)
	FCNn module mask 3 registers (FCNnCMMKCTL05H, FCNnCMMKCTL06H, FCNnCMMKCTL05W)
	FCNn module mask 4 registers (FCNnCMMKCTL07H, FCNnCMMKCTL08H, FCNnCMMKCTL07W)
	FCNn module mask 5 registers (FCNnCMMKCTL09H, FCNnCMMKCTL10H, FCNnCMMKCTL09W)
	FCNn module mask 6 registers (FCNnCMMKCTL11H, FCNnCMMKCTL12H, FCNnCMMKCTL11W)
	FCNn module mask 7 registers (FCNnCMMKCTL13H, FCNnCMMKCTL14H, FCNnCMMKCTL13W)
	FCNn module mask 8 registers (FCNnCMMKCTL15H, FCNnCMMKCTL16H, FCNnCMMKCTL15W)
	FCNn module control register (FCNnCMCLCTL)
	FCNn module last error information register (FCNnCMLCSTR)
	FCNn module information register (FCNnCMINSTR)
	FCNn module error counter register (FCNnCMERCNT)
	FCNn module interrupt enable register (FCNnCMIECTL)
	FCNn module interrupt status register (FCNnCMISCTL)
	FCNn module bit-rate prescaler register (FCNnCMBRPRS)
	FCNn module bit-rate register (FCNnCMBTCTL)
	FCNn module last in-pointer register (FCNnCMLISTR)
	FCNn module receive history list register (FCNnCMRGRX)
	FCNn module last out-pointer register (FCNnCMLOSTR)
	FCNn module transmit history list register (FCNnCMTGTX)
	FCNn module time stamp register (FCNnCMTSCTL)

Table 20-8 List of FCN Registers (2/2)

Item	Register Name
FCNn message buffer registers	FCNn message data byte 0 to 3 registers m (FCNnMmDAT0W, FCNnMmDAT0H, FCNnMmDAT2H, FCNnMmDAT0B, FCNnMmDAT1B, FCNnMmDAT2B, FCNnMmDAT3B)
	FCNn message data byte 4 to 7 registers m (FCNnMmDAT4W, FCNnMmDAT4H, FCNnMmDAT6H, FCNnMmDAT4B, FCNnMmDAT5B, FCNnMmDAT6B, FCNnMmDAT7B)
	FCNn message data length register m (FCNnMmDTLGB)
	FCNn message configuration register m (FCNnMmSTRB)
	FCNn message ID registers m (FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W)
	FCNn message control register m (FCNnMmCTL)

20.3.2 Overview of CAN Controller Registers

Note All register addresses are given as offsets to the base address <FCNn_base>. The <FCNn_base> addresses of the registers are defined in the first part of this section under the keyword “Register address”.

(1) FCNn Global and Module Registers

Table 20-9 FCNn Global and Module Registers (1/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
$0\ 0008_H$	FCNn global clock selection register	FCNnGMCSPRE	R/W	8	$0F_H$
$0\ 0020_H$	FCNn global automatic block transmission delay register	FCNnGMADCTL	R/W	8	00_H
$0\ 8000_H$	FCNn global control register	FCNnGMCLCTL	R/W	16	$00X0_H$ *1
$0\ 8018_H$	FCNn global automatic block transmission register	FCNnGMABCTL	R/W	16	0000_H
$1\ 00C0_H$	FCNn global data new bit monitor register 0	FCNnDNBMRX0	R	32	*2
$1\ 00D0_H$	FCNn global data new bit monitor register 1	FCNnDNBMRX1	R	32	*2
$0\ 8300_H$	FCNn module mask 1 register	FCNnCMMKCTL01H	R/W	16	*2
$0\ 8308_H$		FCNnCMMKCTL02H			
$1\ 0300_H$		FCNnCMMKCTL01W		32	
$0\ 8310_H$	FCNn module mask 2 register	FCNnCMMKCTL03H	R/W	16	*2
$0\ 8318_H$		FCNnCMMKCTL04H			
$1\ 0310_H$		FCNnCMMKCTL03W		32	
$0\ 8320_H$	FCNn module mask 3 register	FCNnCMMKCTL05H	R/W	16	*2
$0\ 8328_H$		FCNnCMMKCTL06H			
$1\ 0320_H$		FCNnCMMKCTL05W		32	
$0\ 8330_H$	FCNn module mask 4 register	FCNnCMMKCTL07H	R/W	16	*2
$0\ 8338_H$		FCNnCMMKCTL08H			
$1\ 0330_H$		FCNnCMMKCTL07W		32	

Table 20-9 FCNn Global and Module Registers (2/2)

Address Offset	Register Name	Symbol	R/W	Access Bit	After Reset
0 8340$_H$	FCNn module mask 5 register	FCNnCMMKCTL09H	R/W	16	*2
0 8348$_H$		FCNnCMMKCTL10H			
1 0340$_H$		FCNnCMMKCTL09W		32	
0 8350$_H$	FCNn module mask 6 register	FCNnCMMKCTL11H	R/W	16	*2
0 8358$_H$		FCNnCMMKCTL12H			
1 0350$_H$		FCNnCMMKCTL11W		32	
0 8360$_H$	FCNn module mask 7 register	FCNnCMMKCTL13H	R/W	16	*2
0 8368$_H$		FCNnCMMKCTL14H			
1 0360$_H$		FCNnCMMKCTL13W		32	
0 8370$_H$	FCNn module mask 8 register	FCNnCMMKCTL15H	R/W	16	*2
0 8378$_H$		FCNnCMMKCTL16H			
1 0370$_H$		FCNnCMMKCTL15W		32	
0 0248$_H$	FCNn module last error information register	FCNnCMLCSTR	R/W	8	00$_H$
0 024C$_H$	FCNn module information register	FCNnCMINSTR	R	8	00$_H$
0 0268$_H$	FCNn module bit-rate prescaler register	FCNnCMBRPRS	R/W	8	FF$_H$
0 0278$_H$	FCNn module last in-pointer register	FCNnCMLISTR	R	8	Undefined
0 0288$_H$	FCNn module last out-pointer register	FCNnCMLOSTR	R	8	Undefined
0 8240$_H$	FCNn module control register	FCNnCMCLCTL	R/W	16	0000$_H$
0 8250$_H$	FCNn module error counter register	FCNnCMERCNT	R	16	0000$_H$
0 8258$_H$	FCNn module interrupt enable register	FCNnCMIECTL	R/W	16	0000$_H$
0 8260$_H$	FCNn module interrupt status register	FCNnCMISCTL	R/W	16	0000$_H$
0 8270$_H$	FCNn module bit-rate register	FCNnCMBTCTL	R/W	16	370F$_H$
0 8280$_H$	FCNn module receive history list register	FCNnCMRGRX	R/W	16	xx02$_H$
0 8290$_H$	FCNn module transmit history list register	FCNnCMTGTX	R/W	16	xx02$_H$
0 8298$_H$	FCNn module time stamp register	FCNnCMTSCTL	R/W	16	0000$_H$

Note 1. The initial value depends on FCNnGMCLCTL.FCNnGMCLECCF, which indicates error detections when reading from message buffer RAM. For details, see the description on the FCNnGMCLCTL register.

Note 2. The value after a reset is 0000$_H$ or 00000000$_H$.

20.3.3 Register Bit Configuration

Table 20-10 FCN Global Register Bit Configuration

Address Offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
$0\ 8000_H$	FCNnGMCLCTL (W)	0	0	FCNnGMCLCLMB	0	0	0	0	FCNnGMCLCLOM
		0	0	0	FCNnGMCLSESR	0	0	FCNnGMCLSEDE	FCNnGMCLSEOM
	FCNnGMCLCTL (R)	0	0	FCNnGMCLECCF	FCNnGMCLSORF	0	0	FCNnGMCLESDE	FCNnGMCLPWOM
		FCNnGMCLSSMO	0	0	0	0	0	0	0
$0\ 0008_H$	FCNnGMCSPRE	0	0	0	0	FCNnGMCSPRSC[3:0]			
$0\ 8018_H$	FCNnGMABCTL (W)	0	0	0	0	0	0	0	FCNnGMABCLAT
		0	0	0	0	0	0	FCNnGMABSEAC	FCNnGMABSEAT
	FCNnGMABCTL (R)	0	0	0	0	0	0	FCNnGMABCLRF	FCNnGMABABTT
		0	0	0	0	0	0	0	0
$0\ 0020_H$	FCNnGMADCTL	0	0	0	0	FCNnGMADSSAD[3:0]			
$1\ 00C0_H$	FCNnDNBMRX0 (R)	FCNnDNBMSSDN[7:0]							
		FCNnDNBMSSDN[15:8]							
		FCNnDNBMSSDN[23:16]							
		FCNnDNBMSSDN[31:24]							
$1\ 00D0_H$	FCNnDNBMRX1 (R)	FCNnDNBMSSDN[39:32]							
		FCNnDNBMSSDN[47:40]							
		FCNnDNBMSSDN[55:48]							
		FCNnDNBMSSDN[63:56]							

Table 20-11 FCN Module Mask Control 16-Bit Registers Bit Configuration (1/2)

Address Offset	Symbol	Bit 15	Bit 14	Bit 13	Bits 12 to 0
$0\ 8300_H$	FCNnCMMKCTL01H	FCNnCMMKSSID[15:0]			
$0\ 8308_H$	FCNnCMMKCTL02H	0	0	0	FCNnCMMKSSID[28:16]
$0\ 8310_H$	FCNnCMMKCTL03H	FCNnCMMKSSID[15:0]			
$0\ 8318_H$	FCNnCMMKCTL04H	0	0	0	FCNnCMMKSSID[28:16]
$0\ 8320_H$	FCNnCMMKCTL05H	FCNnCMMKSSID[15:0]			
$0\ 8328_H$	FCNnCMMKCTL06H	0	0	0	FCNnCMMKSSID[28:16]
$0\ 8330_H$	FCNnCMMKCTL07H	FCNnCMMKSSID[15:0]			
$0\ 8338_H$	FCNnCMMKCTL08H	0	0	0	FCNnCMMKSSID[28:16]
$0\ 8340_H$	FCNnCMMKCTL09H	FCNnCMMKSSID[15:0]			
$0\ 8348_H$	FCNnCMMKCTL10H	0	0	0	FCNnCMMKSSID[28:16]

Table 20-11 FCN Module Mask Control 16-Bit Registers Bit Configuration (2/2)

Address Offset	Symbol	Bit 15	Bit 14	Bit 13	Bits 12 to 0
0 8350$_H$	FCNnCMMKCTL11H	FCNnCMMKSSID[15:0]			
0 8358$_H$	FCNnCMMKCTL12H	0	0	0	FCNnCMMKSSID[28:16]
0 8360$_H$	FCNnCMMKCTL13H	FCNnCMMKSSID[15:0]			
0 8368$_H$	FCNnCMMKCTL14H	0	0	0	FCNnCMMKSSID[28:16]
0 8370$_H$	FCNnCMMKCTL15H	FCNnCMMKSSID[15:0]			
0 8378$_H$	FCNnCMMKCTL16H	0	0	0	FCNnCMMKSSID[28:16]

Table 20-12 FCN Module Mask Control 32-Bit Registers Bit Configuration

Address Offset	Symbol	Bit 31	Bit 30	Bit 29	Bits 28 to 0
1 0300$_H$	FCNnCMMKCTL01W	0	0	0	FCNnCMMKSSID[28:0]
1 0310$_H$	FCNnCMMKCTL03W	0	0	0	FCNnCMMKSSID[28:0]
1 0320$_H$	FCNnCMMKCTL05W	0	0	0	FCNnCMMKSSID[28:0]
1 0330$_H$	FCNnCMMKCTL07W	0	0	0	FCNnCMMKSSID[28:0]
1 0340$_H$	FCNnCMMKCTL09W	0	0	0	FCNnCMMKSSID[28:0]
1 0350$_H$	FCNnCMMKCTL11W	0	0	0	FCNnCMMKSSID[28:0]
1 0360$_H$	FCNnCMMKCTL13W	0	0	0	FCNnCMMKSSID[28:0]
1 0370$_H$	FCNnCMMKCTL15W	0	0	0	FCNnCMMKSSID[28:0]

Table 20-13 FCN Module Register Bit Configuration (1/2)

Address Offset	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
0 8240$_H$	FCNnCM CLCTL (W)	0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCMCLCLPS[1:0]		FCNnCMCLCLOP[2:0]		
		FCNnCM CLSERC	FCNnCM CLSEAL	0	FCNnCMCLSEPS[1:0]		FCNnCMCLSEOP[2:0]		
	FCNnCM CLCTL (R)	FCNnCM CLERCF	FCNnCM CLALBF	FCNnCM CLVALF	FCNnCMCLMDPF [1:0]		FCNnCMCLMDOF[2:0]		
		0	0	0	0	0	0	FCNnCM CLSSRS	FCNnCM CLSSTS
0 0248$_H$	FCNnCM LCSTR (W)	0	0	0	0	0	0	0	0
	FCNnCM LCSTR (R)	0	0	0	0	0	FCNnCMLCSSLC[2:0]		
0 024CH	FCNnCM INSTR	0	0	0	FCNnCMI NBOFF	FCNnCMINSSTE[1:0]		FCNnCMINSSRE[1:0]	
0 8250$_H$	FCNnCM ERCNT	FCNnCMERTECF[7:0]							
		FCNnCM ERRPSF	FCNnCMERRECF[6:0]						
0 8258$_H$	FCNnCM IECTL (W)	0	FCNnCMIECLIE[6:0]						
		0	FCNnCMIESEIE[6:0]						
	FCNnCM IECTL (R)	0	FCNnCMIEINTF[6:0]						
		0	0	0	0	0	0	0	0
0 8260$_H$	FCNnCM ISCTL (W)	0	FCNnCMISCLTS[6:0]						
		0	0	0	0	0	0	0	0
	FCNnCM ISCTL (R)	0	FCNnCMISITSF[6:0]						
		0	0	0	0	0	0	0	0
0 0268$_H$	FCNnCM BRPRS	FCNnCMBRPRS[7:0]							
0 8270$_H$	FCNnCM BTCTL	0	0	0	0	FCNnCMBTS1LG[3:0]			
		0	0	FCNnCMBTJWLG [1:0]		0	FCNnCMBTS2LG[2:0]		
0 0278$_H$	FCNnCM LISTR	FCNnCMLISSLR[7:0]							
0 8280$_H$	FCNnCM RGRX (W)	0	0	0	0	0	0	0	FCNnCM RGCLRV
		0	0	0	0	0	0	0	0
	FCNnCM RGRX (R)	0	0	0	0	0	0	FCNnCM RGSSPM	FCNnCM RGRVFF
		FCNnCMRDSSPT[7:0]							
0 0288$_H$	FCNnCM LOSTR	FCNnCMLOSSLT[7:0]							
0 8290$_H$	FCNnCM TGTX (W)	0	0	0	0	0	0	0	FCNnCM TGCLTV
		0	0	0	0	0	0	0	0
	FCNnCM TGTX (R)	0	0	0	0	0	0	FCNnCM TGSSPM	FCNnCM TGTVFF
		FCNnCMTGSSPT[7:0]							

Table 20-13 FCN Module Register Bit Configuration (2/2)

Address Offset	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
$0\ 8298_H$	FCNnCM TSCTL (W)	0	0	0	0	0	FCNnCM TSCLLK	FCNnCM TSCLSL	FCNnCM TSCLTS
		0	0	0	0	0	FCNnCM TSSELK	FCNnCM TSSESL	FCNnCM TSSETS
	FCNnCM TSCTL (R)	0	0	0	0	0	FCNnCM TSLOKE	FCNnCM TSSELE	FCNnCM TSTSGE
		0	0	0	0	0	0	0	0

Table 20-14 FCN Message Buffer Register Bit Configuration (1/2)

Address Offset	Symbol	Bit 7/15/ 31/23	Bit 6/14/ 30/22	Bit 5/13/ 29/21	Bit 4/12/ 28/20	Bit 3/11/ 27/19	Bit 2/10/ 26/18	Bit 1/9/ 25/17	Bit 0/8/ 24/16
$1\ 1000_H + m \times 40_H$	FCNnMm DAT0W	FCNnMmSSD0[7:0]							
		FCNnMmSSD1[7:0]							
		FCNnMmSSD2[7:0]							
		FCNnMmSSD3[7:0]							
$0\ 9000_H + m \times 40_H$	FCNnMm DAT0H	FCNnMmSSD0[7:0]							
		FCNnMmSSD1[7:0]							
$0\ 1000_H + m \times 40_H$	FCNnMm DAT0B	FCNnMmSSD0[7:0]							
$0\ 1004_H + m \times 40_H$	FCNnMm DAT1B	FCNnMmSSD1[7:0]							
$0\ 9008_H + m \times 40_H$	FCNnMm DAT2H	FCNnMmSSD2[7:0]							
		FCNnMmSSD3[7:0]							
$0\ 1008_H + m \times 40_H$	FCNnMm DAT2B	FCNnMmSSD2[7:0]							
$0\ 100C_H + m \times 40_H$	FCNnMm DAT3B	FCNnMmSSD3[7:0]							
$1\ 1010_H + m \times 40_H$	FCNnMm DAT4W	FCNnMmSSD4[7:0]							
		FCNnMmSSD5[7:0]							
		FCNnMmSSD6[7:0]							
		FCNnMmSSD7[7:0]							
$0\ 9010_H + m \times 40_H$	FCNnMm DAT4H	FCNnMmSSD4[7:0]							
		FCNnMmSSD5[7:0]							
$0\ 1010_H + m \times 40_H$	FCNnMm DAT4B	FCNnMmSSD4[7:0]							
$0\ 1014_H + m \times 40_H$	FCNnMm DAT5B	FCNnMmSSD5[7:0]							
$0\ 9018_H + m \times 40_H$	FCNnMm DAT6H	FCNnMmSSD6[7:0]							
		FCNnMmSSD7[7:0]							
$0\ 1018_H + m \times 40_H$	FCNnMm DAT6B	FCNnMmSSD6[7:0]							
$0\ 101C_H + m \times 40_H$	FCNnMm DAT7B	FCNnMmSSD7[7:0]							
$0\ 1020_H + m \times 40_H$	FCNnMm DTLGB	0				FCNnMmDTLG[3:0]			
$0\ 1024_H + m \times 40_H$	FCNnMm STRB	FCNnMm SSOW	FCNnMmSSMT[3:0]				FCNnMm SSRT	0	FCNnMm SSAM

Table 20-14 FCN Message Buffer Register Bit Configuration (2/2)

Address Offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
$0\ 9028_H + m \times 40_H$	FCNnMm MID0H	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
$0\ 9030_H + m \times 40_H$	FCNnMm MID1H	FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
$1\ 1028_H + m \times 40_H$	FCNnMm MID0W	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
		FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
$0\ 9038_H + m \times 40_H$	FCNnMmCTL (W)	0	FCNnMm CLNH	0	FCNnMm CLMW	FCNnMm CLIE	FCNnMm CLDN	FCNnMm CLTR	FCNnMm CLRY
		0	FCNnMm SENH	0	0	FCNnMm SEIE	0	FCNnMm CSETR	FCNnMm SERY
	FCNnMmCTL (R)	0	FCNnMm NHMF	0	FCNnMm MOWF	FCNnMm IENF	FCNnMm DTNF	FCNnMm TRQF	FCNnMm RDYF
		0	0	FCNnMm MUCF	0	0	0	FCNnMm TCPF	0

20.4 Bit Set/Clear Functions

The FCN control registers include registers whose bits can be set or cleared via the CPU and via the CAN controller. These register bits cannot be changed directly by the CPU by any bit manipulation instructions, such as SET1, CLR1, and NOT1. Instead, a special bit-set/bit-clear mechanism is used.

All registers where bit manipulation operations are prohibited are organized in such a way that all bits allowed for changing by the CPU are located in the lower byte (RWx in the register layout below), while in the upper byte either no or read-only information is located (ROx in the register layout below).

The registers can be read in the usual way getting all 16 data bits in their

current setting, as described in the register description.

For setting or clearing any of the lower 8 bits, the following mechanism is implemented:

When writing 16-bit data to the register address,

Bit clear

- Each of the lower 8 data bits (CLx in the register layout below) indicates whether the corresponding register bit RWx should be
  - cleared, i.e. set to 0: if CLx = 1, the corresponding RWx is cleared to 0
  - remain unchanged: if CLx = 0, the corresponding RWx does not change

Bit set

- Each of the upper 8 data bits (SEx in the register layout below) indicates whether the corresponding register bit should be
  - set, i.e. set to 1: if SEx = 1, the corresponding RWx is set to 1
  - remain unchanged: if SEx = 0, the corresponding RWx does not change

Register layout for read access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO7	RO6	RO5	RO4	RO3	RO2	RO1	RO0	RW7	RW6	RW5	RW4	RW3	RW2	RW1	RW0
Changing by the CPU not possible								Bits for CPU manipulation via SE7 to SE0 and CL7 to CL0							

Register layout for write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	CL7	CL6	CL5	CL4	CL3	CL2	CL1	CL0
SEx = 1 sets the corresponding RW7 to RW0								CLx = 1 clears the corresponding RW7 to RW0							

The following table denotes the operations applied to the RWx bits.

Table 20-15 Bit Set/Clear Operation

CLx	SEx	Operation on RWx
0	0	No change of RWx
0	1	RWx set to 1
1	0	RWx cleared to 0
1	1	No change of RWx

Example The following shows an example.

The register with the content 1883_H shall be changed so that

- bit 3 shall be set to 1: SE3 = 1
- bit 1 shall be cleared to 0: CL1 = 1

Register read before bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	**0**	0	**1**	1
May hold any value, here 18_H								RW7 to RW0: 83_H							

Register write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0
SE3 = 1: 08_H								CL1 = 1: 02_H							

Register read after bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	**1**	0	**0**	1
May hold any value, here 18_H								RW7 to RW0: 89_H							

20.5 Control Registers

20.5.1 FCN Global Registers

(1) FCNnGMCLCTL - FCNn Global Control Register

This register is used to control the operation of the FCN module.

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

Address <FCNn_base> + 0 8000_H

Initial value $00x0_H$

This register is initialized by various types of reset.

(a) FCNnGMCLCTL Read

15	14	13	12	11	10	9	8
FCNnGMCLSSMO	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	FCNnGMCLECCF *1	FCNnGMCLSORF*1	0	0	FCNnGMCLESDE	FCNnGMCLPWOM

Note 1. Initial values are as follows:

- No error detection after software reset: 0000_H
- No error detection during software reset: 0010_H
- Error detection after software reset: 0020_H
- Error detection during software reset: 0030_H

FCNnGMCLSSMO	Enabling Access to Message Buffer Register and Transmit/Receive History Registers of FCN Module
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.

Caution 1. While FCNnGMCLCTL.FCNnGMCLSSMO is cleared to 0, software access to the message buffer registers (i.e. all registers with name prefix FCNnMm...), or registers related to transmit history or receive history (FCNnCMLOSTR, FCNnCMTGTX, FCNnCMLISTR, and FCNnCMRGRX) is disabled.

Caution 2. FCNnGMCLCTL.FCNnGMCLSSMO is read-only. Even if 1 is written while it is 0, its value does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Note FCNnGMCLCTL.FCNnGMCLSSMO is cleared to 0 when the FCN module enters FCN sleep mode/FCN stop mode, or when FCNnGMCLCTL.FCNnGMCLPWOM is cleared to 0.

FCNnGMCLSSMO is set to 1 when FCN sleep mode/FCN stop mode is released, or when FCNnGMCLCTL.FCNnGMCLPWOM is set to 1.

FCNnGMCLECCF	Message Buffer RAM Read Error Detect
0	Not detect error for reading from message buffer RAM.
1	Detect error for reading from message buffer RAM.

Note 1. FCNnGMCLCTL.FCNnGMCLECCF is set to 1 in case of detecting a memory error when reading from the message buffer RAM during the software reset process. Once FCNnGMCLECCF is set to 1, it keeps the level until it is cleared to 0.

Note 2. Use this bit only to check a memory error after software reset.

Note 3. It is impossible to clear FCNnGMCLECCF to 0 while FCNnGMCLCTL.FCNnGMCLSORF is set to 1 (software reset is ongoing).

FCNnGMCLSORF	Software Reset Execution Status
0	No software reset
1	Software reset is ongoing

Note 1. While a software reset is ongoing (FCNnGMCLCTL.FCNnGMCLSORF is set to 1), it is impossible to set FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.EFSD.
It is possible to set start a software reset by setting FCNnGMCLCTL.FCNnGMCLSESR to 1 while FCNnGMCLCTL.FCNnGMCLPWOM is cleared to 0.

Note 2. When FCNnGMCLCTL.FCNnGMCLSORF is set to 1, the initialization of message buffer RAM starts. It is possible to detect error during initializing message buffer RAM, if FCNnGMCLCTL.FCNnGMCLECCF is cleared before setting FCNnGMCLSORF.

Note 3. When FCNnGMCLCTL.FCNnGMCLSORF is set to 1 again while it has been set to 1, the software reset procedure does not restart, but continues.

Note 4. After release of the hardware reset, FCNnGMCLCTL.FCNnGMCLSORF is automatically set to 1 and initialization of message buffer RAM starts.

Note 5. It is impossible that clearing FCNnGMCLCTL.FCNnGMCLPWOM to 0 and setting FCNnGMCLCTL.FCNnGMCLSORF to 1 are done at the same time.

Note 6. If a hardware reset occurs while FCNnGMCLCTL.FCNnGMCLSORF = 1, the software reset procedure is stopped (aborted), and the hardware reset starts.

FCNnGMCLESDE	Enabling Forced Shut Down
0	Forced shut down disabled.
1	Forced shut down of FCNnGMCLCTL.FCNnGMCLPWOM bit = 0 enabled.

Caution To request a forced shut down, FCNnGMCLCTL.FCNnGMCLPWOM must be cleared to 0 in a sub-sequent, immediately following access after FCNnGMCLCTL.FCNnGMCLESDE has been set to 1. If any access to another register (including reading the FCNnGMCLCTL register) is executed without clearing FCNnGMCLPWOM immediately after FCNnGMCLESDE has been set to 1, FCNnGMCLESDE is forcibly cleared to 0, and the forced shut down request is invalid.

FCNnGMCLPWOM	Global Operating Mode
0	FCN module is disabled to operate.
1	FCN module is enabled to operate.

Caution FCNnGMCLCTL.FCNnGMCLPWOM can be cleared only in initialization mode or immediately after FCNnGMCLCTL.FCNnGMCLESDE is set (forced shut down).

(b) FCNnGMCLCTL Write

15	14	13	12	11	10	9	8
0	0	0	FCNnGMCLSESR	0	0	FCNnGMCLSESD	FCNnGMCLSEOM

7	6	5	4	3	2	1	0
0	0	FCNnGMCLCLMB	0	0	0	0	FCNnGMCLCLOM

FCNnGMCLSESR	Software Reset Start
0	No changes.
1	Start software reset.

FCNnGMCLSESD	FCNnGMCLSESD Setting
0	FCNnGMCLESDE bit remains unchanged.
1	FCNnGMCLESDE bit set to 1.

FCNnGMCLSEOM	FCNnGMCLCLOM	FCNnGMCLPWOM Setting
0	1	FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.
1	0	FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.
Other than above		FCNnGMCLCTL.FCNnGMCLPWOM bit remains unchanged.

Caution Set FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.FCNnGMCLESDE bits always separately.

FCNnGMCLCLMB	FCNnGMCLCTL.FCNnGMCLECCF Clear
0	No change in FCNnGMCLCTL.FCNnGMCLECCF bit.
1	FCNnGMCLCTL.FCNnGMCLECCF bit cleared to 0.

(2) FCNnGMCSPRE - FCNn Global Clock Selection Register

This register is used to select the FCN module system clock.

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

Address <FCNn_base> + 0008_H

Initial value $0F_H$

This register is initialized by various types of reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMCSPRSC[3:0]			

FCNnGMCSPRSC[3:0]	FCN Module System Clock (f_{CANMOD})
0000_B	$f_{CAN}/1$
0001_B	$f_{CAN}/2$
0010_B	$f_{CAN}/3$
0011_B	$f_{CAN}/4$
0100_B	$f_{CAN}/5$
0101_B	$f_{CAN}/6$
0110_B	$f_{CAN}/7$
0111_B	$f_{CAN}/8$
1000_B	$f_{CAN}/9$
1001_B	$f_{CAN}/10$
1010_B	$f_{CAN}/11$
1011_B	$f_{CAN}/12$
1100_B	$f_{CAN}/13$
1101_B	$f_{CAN}/14$
1110_B	$f_{CAN}/15$
1111_B	$f_{CAN}/16$ (default value)

Note f_{CAN} = FCN reference clock

(3) FCNnGMABCTL - FCNn Global Automatic Block Transmission Control Register

This register is used to control the automatic block transmission (ABT) operation.

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

Address <FCNn_base> + $0\ 8018_H$

Initial value 0000_H

This register is initialized by various types of reset.

(a) FCNnGMABCTL Read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnGMABCLRF	FCNnGMABABTT

FCNnGMABCLRF	Automatic Block Transmission Engine Clear Status
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Note Start clearing the automatic transmission engine by setting FCNnGMABCTL.FCNnGMABCLAT to 1 while FCNnGMABCTL.FCNnGMABCLRF is 0.
The operation is not guaranteed if FCNnGMABCLRF is set to 1 while FCNnGMABCLRF = 1.

FCNnGMABABTT	Automatic Block Transmission Status
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

(b) FCNnGMABCTL Write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnGMABSEAC	FCNnGMABSEAT

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnGMABCLAT

Note When the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC to 1, FCNnGMABCLRF is automatically set, and cleared to 0 as soon as the requested clearing processing is completed.

Caution 1. Before changing normal operating mode with ABT to initialization mode, be sure to set the FCNnGMABCTL register to the default value (0000_H) and confirm the FCNnGMABCTL register is surely initialized to the default value (0000_H).

Caution 2. Do not start automatic block transmission in initialization mode. If automatic block transmission is started in initialization mode, the operation is not guaranteed after the CAN controller has entered normal operating mode with ABT.

Caution 3. Do not start automatic block transmission while FCNnCMCLCTL.FCNnCMCLSSTS is set to 1 (transmission in progress). Confirm FCNnCMCLSSTS = 0 directly in advance before starting automatic block transmission.

FCNnGMABSEAC	Automatic Block Transmission Engine Clear Request
0	The automatic block transmission engine is in idle status 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 FCNnGMABCTL.FCNnGMABABTT to 1.

FCNnGMABSEAT	FCNnGMABCLAT	Automatic Block Transmission Start
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change of FCNnGMABCTL.FCNnGMABABTT

(4) FCNnGMADCTL - FCNn Global Automatic Block Transmission Delay Register

This register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in normal operating mode with ABT.

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

Address <FCNn_base> + 0020_H

Initial value 00_H

This register is initialized by various types of reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMADSSAD[3:0]			

FCNnGMADSSAD[3:0]	Data Frame Interval during Automatic Block Transmission in DBT Unit*1
0000_B	0 DBT (default value)
0001_B	2^5 DBT
0010_B	2^6 DBT
0011_B	2^7 DBT
0100_B	2^8 DBT
0101_B	2^9 DBT
0110_B	2^{10} DBT
0111_B	2^{11} DBT
1000_B	2^{12} DBT
Other than above	Setting prohibited

Note 1. Unit: Data bit time (DBT)

Caution 1. Do not change the contents of the FCNnGMADCTL register while FCNnGMABCTL.FCNnGMABCLRF = 1 (clearing of ABT in progress).

Caution 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 is made.

(5) FCNnDNBMRXk - FCNn Global Data New Bit Monitor Registers (k = 0, 1)

These registers are used to read data new flags globally for several message buffers at a time.

Access These registers can be read in 32-bit units.

Address FCNnDNBMRX0: <FCNn_base> + 1 $00C0_H$
FCNnDNBMRX1: <FCNn_base> + 1 $00D0_H$

Initial value 00000000_H

31	30	29	28	27	26	25	24
FCNnDNBMSSDN[31:24]							

23	22	21	20	19	18	17	16
FCNnDNBMSSDN[23:16]							

15	14	13	12	11	10	9	8
FCNnDNBMSSDN[15:08]							

7	6	5	4	3	2	1	0
FCNnDNBMSSDN[07:00]							

FCNnDNBMSSDN[31:0]	Message Buffer Data New
0	No remote or data frame has been stored into the message buffer.
1	A remote or data frame has been stored into the message buffer.

20.5.2 FCN Module Registers

(1) FCNnCMMKCTLaH - FCNn Module Mask Control Register

These registers are used to extend the number of receivable messages into the same message buffer by masking part of the identifier (ID) comparison of a message and invalidating the ID of the masked part.

Two 16-bit registers FCNnCMMKCTLaH (a = 01 to 16) can also be accessed via a single 32-bit access to the registers FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15).

Access The FCNnCMMKCTLaH registers can be read/written in 16-bit units.
The FCNnCMMKCTLaW registers can be read/written in 32-bit units.

Address FCNnCMMKCTL01H: <FCNn_base> + 0 8300_H
FCNnCMMKCTL02H: <FCNn_base> + 0 8308_H
FCNnCMMKCTL03H: <FCNn_base> + 0 8310_H
FCNnCMMKCTL04H: <FCNn_base> + 0 8318_H
FCNnCMMKCTL05H: <FCNn_base> + 0 8320_H
FCNnCMMKCTL06H: <FCNn_base> + 0 8328_H
FCNnCMMKCTL07H: <FCNn_base> + 0 8330_H
FCNnCMMKCTL08H: <FCNn_base> + 0 8338_H
FCNnCMMKCTL09H: <FCNn_base> + 0 8340_H
FCNnCMMKCTL10H: <FCNn_base> + 0 8348_H
FCNnCMMKCTL11H: <FCNn_base> + 0 8350_H
FCNnCMMKCTL12H: <FCNn_base> + 0 8358_H
FCNnCMMKCTL13H: <FCNn_base> + 0 8360_H
FCNnCMMKCTL14H: <FCNn_base> + 0 8368_H
FCNnCMMKCTL15H: <FCNn_base> + 0 8370_H
FCNnCMMKCTL16H: <FCNn_base> + 0 8378_H

FCNnCMMKCTL01W: <FCNn_base> + 1 0300_H
FCNnCMMKCTL03W: <FCNn_base> + 1 0310_H
FCNnCMMKCTL05W: <FCNn_base> + 1 0320_H
FCNnCMMKCTL07W: <FCNn_base> + 1 0330_H
FCNnCMMKCTL09W: <FCNn_base> + 1 0340_H
FCNnCMMKCTL11W: <FCNn_base> + 1 0350_H
FCNnCMMKCTL13W: <FCNn_base> + 1 0360_H
FCNnCMMKCTL15W: <FCNn_base> + 1 0370_H

Initial value 0000_H

FCNnCMMKCTLaH (a = 01, 03, 05, 07, 09, 11, 13, 15)

15	14	13	12	11	10	9	8
FCNnCMMKSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnCMMKSSID[7:0]							

(a) FCNnCMMKCTLaH (a = 02, 04, 06, 08, 10, 12, 14, 16)

15	14	13	12	11	10	9	8
0	0	0	FCNnCMMKSSID[28:24]				

7	6	5	4	3	2	1	0
FCNnCMMKSSID[23:16]							

(b) FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15)

31	30	29	28	27	26	25	24
0	0	0	FCNnCMMKSSID[28:24]				

23	22	21	20	19	18	17	16
FCNnCMMKSSID[23:16]							

15	14	13	12	11	10	9	8
FCNnCMMKSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnCMMKSSID[7:0]							

FCNnCMMKSSID[i] *1	Mask Pattern Setting of ID
0	The ID bit **i** of the message buffer m set by FCNnMmSSID[**i**] are compared with the ID bits of the received message frame.
1	The ID bit **i** of the message buffer m set by FCNnMmSSID[**i**] are not compared with the ID bits of the received message frame (they are masked).

Note 1. i = [28:0]

Note Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, FCNnCMMKSSID[17:0] are ignored. Therefore, only FCNnCMMKSSID[28:18] of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(2) FCNnCMCLCTL - FCNn Module Control Register

This register is used to control operating mode of the FCN module.

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

Address <FCNn_base> + $0\ 8240_H$

Initial value 0000_H

This register is initialized by various types of reset.

(a) FCNnCMCLCTL Read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnCMCLSSRS	FCNnCMCLSSTS

7	6	5	4	3	2	1	0
FCNnCMCLERCF	FCNnCMCLALBF	FCNnCMCLVALF	FCNnCMCLMDPF[1:0]		FCNnCMCLMDOF[2:0]		

FCNnCMCLSSRS	Reception Status
0	Reception is stopped.
1	Reception is in progress.

Note 1. FCNnCMCLSSRS 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

Note 2. FCNnCMCLSSRS is cleared to 0 under the following conditions (timing).

- When a recessive level is detected at the second bit of the inter-frame space
- On transition to initialization mode at the first bit of the inter-frame space

FCNnCMCLSSTS	Transmission Status
0	Transmission is stopped.
1	Transmission is in progress.

Note 1. FCNnCMCLSSTS is set to 1 under the following condition (timing).

- The SOF bit of a transmit frame is detected

Note 2. FCNnCMCLSSTS 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 inter-frame space
- On transition to initialization mode at the first bit of the inter-frame space

FCNnCMCLERCF	Error Counter Clear
0	The FCNnCMERCNT and FCNnCMLCSTR registers are not cleared in initialization mode.
1	The FCNnCMERCNT and FCNnCMLCSTR registers are cleared in initialization mode.

Caution The FCNnCMCLERCF bit is used to clear the error counter FCNnCMERCNT and information register FCNnCMLCSTR for re-initialization or forced recovery from the bus-off state. The error counter and the information register can be cleared (by setting FCNnCMCLERCF to 1) under the following conditions.

- In initialization mode during bus-off period
- In initialization mode after the FCN module is started (by setting FCNnGMCLPWOM to 1 from the FCNnGMCLPWOM = 0 state)
- In initialization mode after all transmission requests are cleared according to the transmission abort processing in Figure 20-24 in operating mode (In normal operating mode with ABT, clear all transmission requests according to the transmission abort processing in Figure 20-25.)

Note 1. When the FCNnCMERCNT and FCNnCMLCSTR registers have been cleared, FCNnCMCLERCF is also cleared to 0 automatically.
Note 2. FCNnCMCLERCF can be set to 1 at the same time as a request to change initialization mode to an operating mode is made.
Note 3. FCNnCMCLERCF is read-only in FCN sleep mode or FCN stop mode.
Note 4. The error counter is also cleared by normal shutdown or forced shutdown of the CAN controller.

FCNnCMCLALBF	Setting Operation in Case of Arbitration Loss
0	Re-transmission is not executed in case of an arbitration loss in single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in single-shot mode.

Note FCNnCMCLALBF is valid only in single-shot mode.

FCNnCMCLVALF	Valid Receive Message Frame Detection
0	A valid message frame has not been received since FCNnCMCLVALF was last cleared to 0.
1	A valid message frame has been received since FCNnCMCLVALF was last cleared to 0.

Note 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).
Note 2. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in normal mode and the other in receive-only mode, FCNnCMCLVALF is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.
Note 3. To clear FCNnCMCLVALF, set FCNnCMCLCLVL to 1 first and confirm that FCNnCMCLVALF is cleared. If it is not cleared, perform clearing processing again.

FCNnCMCLMDPF[1:0]	Power Save Mode
00_B	No power save mode is selected.
01_B	FCN sleep mode
10_B	Setting prohibited
11_B	FCN stop mode

Caution 1. Transition to and from FCN stop mode must be made via FCN sleep mode. A request for direct transition to and from FCN stop mode is ignored.

Caution 2. The FCNnGMCLSSMO flag of FCNnGMCLCTL must be checked after releasing a power save mode prior to access the message buffers again.

Caution 3. FCN sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading FCNnCMCLMDPF[1:0].

Caution 4. Do not use power save mode in combination with the operating mode change. Make these accesses in different steps.

Note When initialization mode transitions to any communication mode, the FCN module checks the CAN bus idle period and then participates in communication. The FCN module can transition to sleep mode before checking the idle period, but the wake-up condition is always a change from recessive level to dominant level.

FCNnCMCLMDOF[2:0]	Operating Mode
000_B	No operating mode is selected (FCN module is in initialization mode).
001_B	Normal operating mode
010_B	Normal operating mode with automatic block transmission function (normal operating mode with ABT)
011_B	Receive-only mode
100_B	Single-shot mode
101_B	Self-test mode
Other than above	Setting prohibited

Caution 1. Transition to initialization mode or power save mode may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.

Caution 2. If initialization mode is set during reception in operating mode, the last reception in which the FCNnMmCTL.FCNnMmDTNF bit in the message buffer is set may occur. A transition back to operating mode also clears the receive history list. Therefore, confirm that initialization mode is reached by reading operating mode. Before restarting operating mode, clear all set FCNnMmCTL.FCNnMmDTNF bits in all valid receive message buffer.

Note FCNnCMCLMDOF[2:0] are read-only in FCN sleep mode or FCN stop mode.

(b) FCNnCMCLCTL Write

15	14	13	12	11	10	9	8
FCNnCM CLSERC	FCNnCM CLSEAL	0	FCNnCM CLSEPS[1:0]		FCNnCM CLSEOP[2:0]		

7	6	5	4	3	2	1	0
0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCM CLCLPS[1:0]		FCNnCM CLCLOP[2:0]		

FCNnCMCLSERC	Setting FCNnCMCLERCF
1	FCNnCMCLERCF is set to 1.
Other than above	FCNnCMCLERCF is not changed.

FCNnCMCLSEAL	FCNnCMCLCLAL	Setting FCNnCMCLALBF
0	1	FCNnCMCLALBF is cleared to 0.
1	0	FCNnCMCLALBF is set to 1.
Other than above		FCNnCMCLALBF is not changed.

FCNnCMCLCLVL	Setting FCNnCMCLVALF
0	FCNnCMCLVALF is not changed.
1	FCNnCMCLVALF is cleared to 0.

FCNnCMCLSEPS0	FCNnCMCLCLPS0	Setting FCNnCMCLMDPF0
0	1	FCNnCMCLMDPF0 is cleared to 0.
1	0	FCNnCMCLMDPF0 is set to 1.
Other than above		FCNnCMCLMDPF0 is not changed.

FCNnCMCLSEPS1	FCNnCMCLCLPS1	Setting FCNnCMCLMDPF1
0	1	FCNnCMCLMDPF1 is cleared to 0.
1	0	FCNnCMCLMDPF1 is set to 1.
Other than above		FCNnCMCLMDPF1 is not changed.

FCNnCMCLSEOP0	FCNnCMCLCLOP0	Setting FCNnCMCLMDOF0
0	1	FCNnCMCLMDOF0 is cleared to 0.
1	0	FCNnCMCLMDOF0 is set to 1.
Other than above		FCNnCMCLMDOF0 is not changed.

FCNnCMCLSEOP1	FCNnCMCLCLOP1	Setting FCNnCMCLMDOF1
0	1	FCNnCMCLMDOF1 is cleared to 0.
1	0	FCNnCMCLMDOF1 is set to 1.
Other than above		FCNnCMCLMDOF1 is not changed.

FCNnCMCLSEOP2	FCNnCMCLCLOP2	Setting FCNnCMCLMDOF2
0	1	FCNnCMCLMDOF2 is cleared to 0.
1	0	FCNnCMCLMDOF2 is set to 1.
Other than above		FCNnCMCLMDOF2 is not changed.

(3) FCNnCMCLCSTR - FCNn Module Last Error Information Register

This register provides the error information of the CAN protocol.

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

Address <FCNn_base> + $0\ 0248_H$

Initial value 00_H

This register is initialized by various types of reset

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMLCSSLC[2:0]		

Note 1. The contents of the FCNnCMLCSTR register are not cleared when the FCN module changes from an operating mode to initialization mode.

Note 2. If an attempt is made to write a value other than 00_H to the FCNnCMLCSTR register by software, the access is ignored.

FCNnCMLCSSLC[2:0]	Last FCN Protocol Error Information
000_B	No error
001_B	Stuff error
010_B	Form error
011_B	ACK error
100_B	Bit error (The FCN 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.)
101_B	Bit error (The FCN 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.)
110_B	CRC error
111_B	Undefined

(4) FCNnCMINSTR - FCNn Module Information Register

This register indicates the status of the FCN module.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + $024C_H$

Initial value 00_H

This register is initialized by various types of reset.

7	6	5	4	3	2	1	0
0	0	0	FCNnCMINBOFF	FCNnCMINSSTE[1:0]		FCNnCMINSSRE[1:0]	

FCNnCMINBOFF	Bus-off State
0	Not bus-off state (transmit error counter is 255 or less). (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter is more than 255). (The value of the transmit counter is 256 or more.)

FCNnCMINSSTE[1:0]	Transmission Error Counter Status
00_B	The value of the transmission error counter is less than that of the warning level (less than 96).
01_B	The value of the transmission error counter is in the range of the warning level (96 to 127).
10_B	Setting prohibited
11_B	The value of the transmission error counter is in the range of the error passive or bus-off status (128 or more).

FCNnCMINSSRE[1:0]	Reception Error Counter Status
00_B	The value of the reception error counter is less than that of the warning level (less than 96).
01_B	The value of the reception error counter is in the range of the warning level (96 to 127).
10_B	Setting prohibited
11_B	The value of the reception error counter is in the range of the error passive status (128 or more).

(5) FCNnCMERCNT - FCNn Module Error Counter Register

This register indicates the count value of the transmission/reception error counter.

Access This register is read-only in 16-bit units.

Address <FCNn_base> + $0\ 8250_H$

Initial value 0000_H

This register is initialized by various types of reset.

15	14	13	12	11	10	9	8
FCNnCMERRPSF	FCNnCMERRECF[6:0]						

7	6	5	4	3	2	1	0
FCNnCMERTECF[7:0]							

FCNnCMERRPSF	Reception Error Passive Status
0	The reception error counter is not in the error passive range (less than 128).
1	The reception error counter is in the error passive range (128 or more).

FCNnCMERRECF[6:0]	Reception Error Counter
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.

Note FCNnCMERRECF[6:0] are invalid in the reception error passive state (FCNnCMINSTR.FCNnCMINSSRE[1:0] = 11_B).

FCNnCMERTECF[7:0]	Transmission Error Counter
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.

Note FCNnCMERTECF[7:0] are invalid in the bus-off state (FCNnCMINSTR.FCNnCMINBOFF = 1).

(6) FCNnCMIECTL - FCNn Module Interrupt Enable Register

This register is used to enable or disable the interrupts of the FCN module.

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

Address <FCNn_base> + $0\ 8258_H$

Initial value 0000_H

This register is initialized by various types of reset.

(a) FCNnCMIECTL Read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMIEINTF[6:0]						

FCNnCMIEINTF[6:0]	FCN Module Interrupt Enable
0	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is disabled.
1	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is enabled.

(b) FCNnCMIECTL Write

15	14	13	12	11	10	9	8
0	FCNnCMIESEIE[6:0]						

7	6	5	4	3	2	1	0
0	FCNnCMIECLIE[6:0]						

FCNnCMIESEIE[6:0]	FCNnCMIECLIE[6:0]	Setting FCNnCMIEINTF[6:0]
0	1	FCNnCMIEINTF[6:0] bits are cleared to 0.
1	0	FCNnCMIEINTF[6:0] bits are set to 1.
Other than above		FCNnCMIEINTF[6:0] bits are not changed.

(7) FCNnCMISCTL - FCNn Module Interrupt Status Register

This register indicates the interrupt status of the FCN module.

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

Address <FCNn_base> + $0\ 8260_H$

Initial value 0000_H

This register is initialized by various types of reset.

(a) FCNnCMISCTL Read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMISITSF[6:0]						

FCNnCMISITSF[6:0]	FCN Interrupt Status
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt Status Bit	Related Interrupt Source Event
FCNnCMISITSF6	FCN module transmission abort interrupt status bit
FCNnCMISITSF5	Wake-up interrupt from FCN sleep mode*1
FCNnCMISITSF4	Arbitration loss interrupt
FCNnCMISITSF3	CAN protocol error interrupt
FCNnCMISITSF2	CAN error status interrupt
FCNnCMISITSF1	Interrupt on completion of reception of valid message frame to message buffer m
FCNnCMISITSF0	Interrupt on normal completion of transmission of message frame from message buffer m

Note 1. FCNnCMISITSF5 is set only when the FCN module is woken up from FCN sleep mode by a CAN bus operation. It is not set when FCN sleep mode has been released by software.

(b) FCNnCMISCTL Write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	FCNnCMISCLTS[6:0]						

FCNnCMISCLTS[6:0]	Clearing FCNnCMISITSF[6:0]
0	FCNnCMISITSF[6:0] bits are not changed.
1	FCNnCMISITSF[6:0] bits are cleared to 0.

Caution Clear the status bits of this register by software, when the confirmation of each status is necessary in the interrupt processing because these bits are not cleared automatically.

(8) FCNnCMBRPRS - FCNn Module Bit-Rate Prescaler Register

This register is used to select the CAN protocol layer reference system clock (f_{TQ}). The communication baud rate is set according to the FCNnCMBTCTL register.

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

Address <FCNn_base> + $0\ 0268_H$

Initial value FF_H

This register is initialized by various types of reset.

7	6	5	4	3	2	1	0
FCNnCMBRPRS[7:0]							

FCNnCMBRPRS [7:0]	CAN Protocol Layer Reference System Clock (f_{TQ})
0	$f_{CANMOD}/1$
1	$f_{CANMOD}/2$
n	$f_{CANMOD}/(n+1)$
:	:
255	$f_{CANMOD}/256$ (default value)

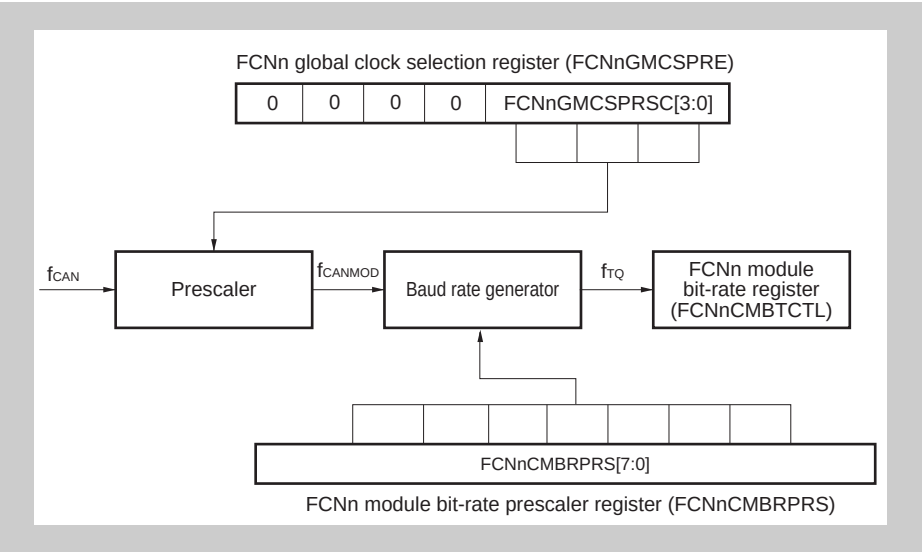


Figure 20-2 FCN Module Clock

Note f_{CAN}: FCN reference clock

f_{CANMOD}: FCN module system clock

f_{TQ}: CAN protocol layer reference system clock

Caution FCNnCMBRPRS can be write-accessed only in initialization mode.

(9) FCNnCMBTCTL - FCNn Module Bit-Rate Register

This register is used to control the data bit time of the communication baud rate.

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

Address <FCNn_base> + $0\ 8270_H$

Initial value $0370F_H$

This register is initialized by various types of reset.

15	14	13	12	11	10	9	8
0	0	FCNnCM BTJWLG[1:0]		0	FCNnCM BTS2LG[2:0]		

7	6	5	4	3	2	1	0
0	0	0	0	FCNnCMBTS1LG[3:0]			

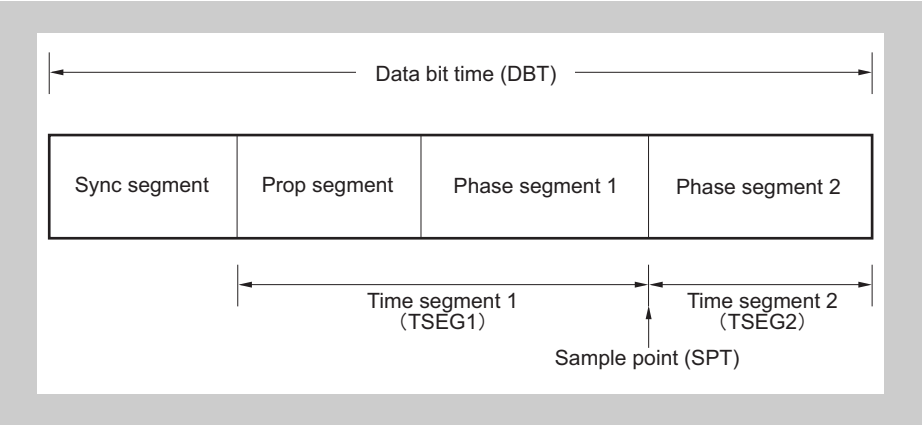


Figure 20-3 Data Bit Time

FCNnCMBTJWLG[1:0]	Length of Synchronization Jump Width (SJW)
00_B	$1T_Q$
01_B	$2T_Q$
10_B	$3T_Q$
11_B	$4T_Q$ (default value)

FCNnCMBTS2LG[2:0]	Length of Time Segment 2 (TSEG2)
000_B	$1T_Q$
001_B	$2T_Q$
010_B	$3T_Q$
011_B	$4T_Q$
100_B	$5T_Q$
101_B	$6T_Q$
110_B	$7T_Q$
111_B	$8T_Q$ (default value)

FCNnCMBTS1LG[3:0]	Length of Time Segment 1 (TSEG1)
0000_B	Setting prohibited
0001_B	$2T_Q$ *1
0010_B	$3T_Q$ *1
0011_B	$4T_Q$
0100_B	$5T_Q$
0101_B	$6T_Q$
0110_B	$7T_Q$
0111_B	$8T_Q$
1000_B	$9T_Q$
1001_B	$10T_Q$
1010_B	$11T_Q$
1011_B	$12T_Q$
1100_B	$13T_Q$
1101_B	$14T_Q$
1110_B	$15T_Q$
1111_B	$16T_Q$ (default value)

Note 1. This setting must not be made when the FCNnCMBRPRS register = 00_H.

Note $T_Q = 1/f_{TQ}$ (f_{TQ}: CAN protocol layer reference system clock)

(10) FCNnCMLISTR - FCNn Module Last In-Pointer Register

This register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + $0\ 0278_H$

Initial value Undefined

7	6	5	4	3	2	1	0
FCNnCMLISSLR[7:0]							

FCNnCMLISSLR [7:0]	Last In-pointer Register of Receive History List
0 to 63	When the FCNnCMLISTR register is read, the contents of the element indexed by the last in-pointer (FCNnCMLISSLR[7:0]) 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.

Note The read value of FCNnCMLISTR is undefined if a data frame or a remote frame has never been stored in the message buffer. Therefore, if FCNnCMRGRX.FCNnCMRGSSPM is set to 1 after the FCN module has changed from initialization mode to an operating mode, the read value of FCNnCMLISTR is undefined.

(11) FCNnCMRGRX - FCNn Module Receive History List Register

This register is used to read the receive history list (RHL).

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

Address <FCNn_base> + $0\ 8280_H$

Initial value $xx02_H$

This register is initialized by various types of reset.

(a) FCNnCMRGRX Read

15	14	13	12	11	10	9	8
FCNnCMRGSSPT[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCMRGSSPM	FCNnCMRGRVFF

FCNnCMRGSSPT[7:0]	Receive History List Read Pointer
0 to 63	When FCNnCMRGRX is read, the contents of the element indexed by the read pointer (FCNnCMRGRX.FCNnCMRGSSPT) 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.

FCNnCMRGSSPM*1	Receive History List Pointer
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.

Note 1. The read value of FCNnCMRGSSPT[7:0] is invalid when FCNnCMTGSSPM = 1.

FCNnCMRGRVFF*1	Receive History List Overflow
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the RHL last time (i.e. read FCNnCMRGRX). The first **(i-1)** entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored, because all buffer numbers are stored at position **(i)**, when FCNnCMRGRVFF is set. Thus, the sequence of receptions cannot be recovered completely now.

Note 1. If FCNnCMRGRVFF is set, FCNnCMRGSSPM is no longer cleared on message storage, but FCNnCMRGSSPM is still set if all entries of FCNnCMRGRX have been read by software.

(b) FCNnCMRGRX Write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMRGCLRV

FCNnCMRGCLRV	Clearing FCNnCMRGRVFF
0	FCNnCMRGRVFF bit is not changed.
1	FCNnCMRGRVFF bit is cleared to 0.

(12) FCNnCMLOSTR - FCNn Module Last Out-Pointer Register

This register indicates the number of the message buffer, from which a data frame or a remote frame was transmitted last.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + $0\ 0288_H$

Initial value Undefined

7	6	5	4	3	2	1	0
FCNnCMLOSSLT[7:0]							

FCNnCMLOSSLT [7:0]	Last Out-pointer of Transmit History List
0 to 63	When FCNnCMLOSSLT is read, the contents of the element indexed by the last out-pointer (FCNnCMLOSTR[7:0]) 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.

Caution The value read from the FCNnCMLOSTR register is undefined if a data frame or remote frame has never been transmitted from the message buffer.

(13) FCNnCMTGTX - FCNn Module Transmit History List Register

This register is used to read the transmit history list (THL).

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

Address <FCNn_base> + $0\ 8290_H$

Initial value $xx02_H$

This register is initialized by various types of reset.

(a) FCNnCMTGTX Read

15	14	13	12	11	10	9	8
FCNnCMTGSSPT[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCMTGSSPM	FCNnCMTGTVFF

FCNnCMTGSSPT[7:0]	Transmit History List Read Pointer
0 to 63	When FCNnCMTGTX is read, the contents of the element indexed by the read pointer (FCNnCMTGSSPT[7:0]) 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.

FCNnCMTGSSPM*1	Transmit History Pointer
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.

Note 1. The read value of FCNnCMTGSSPT[7:0] is invalid when FCNnCMTGSSPM = 1.

FCNnCMTGTVFF *1	Transmit History List Overflow
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the THL last time (e.g, read FCNnCMTGTX). The first **(i-1)** entries are sequentially stored while the last entry is overwritten whenever a new message is completed, because all message buffer numbers are stored at position **(i)**, when FCNnCMTGTVFF is set. Thus, the sequence of transmission cannot be recovered completely.

Note 1. If FCNnCMTGTVFF is set, FCNnCMTGSSPM is no longer cleared on message transmission, but FCNnCMTGSSPM is still set, if all entries of FCNnCMTGTX are read by software.

Note Transmission from the following message buffers is not recorded to the transmit history list in normal operating mode with ABT.

- 0 to 16

(b) FCNnCMTGTX Write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMTGCLTV

FCNnCMTGCLTV	Setting FCNnCMTGTVFF
0	FCNnCMTGTVFF bit is not changed.
1	FCNnCMTGTVFF bit is cleared to 0.

(14) FCNnCMTSCTL - FCNn Module Time Stamp Register

This register is used to control the time stamp function.

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

Address <FCNn_base> + $0\ 8298_H$

Initial value 0000_H

This register is initialized by various types of reset.

(a) FCNnCMTSCTL Read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMTSLOKE	FCNnCMTSSELE	FCNnCMTSTSGE

Note The lock function of the time stamp function must not be used when the FCN module is in normal operating mode with ABT.

FCNnCMTSLOKE	Time Stamp Lock Function Enable
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0. *1

Note 1. FCNnCMTSTSGE is automatically cleared to 0.

FCNnCMTSSELE	Time Stamp Capture Event Selection
0	The time capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

FCNnCMTSTSGE	Setting TSOUT Operation
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

(b) FCNnCMTSCTL Write

15	14	13	12	11	10	9	8
0	0	0	0	0	FCNnCMTSSELK	FCNnCMTSSESL	FCNnCMTSSETS

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMTSCLLK	FCNnCMTSCLSL	FCNnCMTSCLTS

FCNnCMTSSELK	FCNnCMTSCLLK	Setting FCNnCMTSLOKE
0	1	FCNnCMTSLOKE is cleared to 0.
1	0	FCNnCMTSLOKE is set to 1.
Other than above		FCNnCMTSLOKE is not changed.

FCNnCMTSSESL	FCNnCMTSCLSL	Setting FCNnCMTSSELE
0	1	FCNnCMTSSELE is cleared to 0.
1	0	FCNnCMTSSELE is set to 1.
Other than above		FCNnCMTSSELE is not changed.

FCNnCMTSSETS	FCNnCMTSCLTS	Setting FCNnCMTSTSGE
0	1	FCNnCMTSTSGE is cleared to 0.
1	0	FCNnCMTSTSGE is set to 1.
Other than above		FCNnCMTSTSGE is not changed.

20.5.3 FCN Message Buffer Registers

(1) FCNnMmDATxB/H/W - FCNn Message Data Byte Registers

These registers are used to store the data of transmit/receive messages.

Access The FCNnMmDATzW registers can be read/written in 32-bit units.
The FCNnMmDATyH registers can be read/written in 16-bit units.
The FCNnMmDATxB registers can be read/written in 8-bit units.

Address
FCNnMmDAT0B: <FCNn_base> + $0\ 1000_H$ + m × 40_H
FCNnMmDAT1B: <FCNn_base> + $0\ 1004_H$ + m × 40_H
FCNnMmDAT2B: <FCNn_base> + $0\ 1008_H$ + m × 40_H
FCNnMmDAT3B: <FCNn_base> + $0\ 100C_H$ + m × 40_H
FCNnMmDAT4B: <FCNn_base> + $0\ 1010_H$ + m × 40_H
FCNnMmDAT5B: <FCNn_base> + $0\ 1014_H$ + m × 40_H
FCNnMmDAT6B: <FCNn_base> + $0\ 1018_H$ + m × 40_H
FCNnMmDAT7B: <FCNn_base> + $0\ 101C_H$ + m × 40_H

FCNnMmDAT0H: <FCNn_base> + $0\ 9000_H$ + m × 40_H
FCNnMmDAT2H: <FCNn_base> + $0\ 9008_H$ + m × 40_H
FCNnMmDAT4H: <FCNn_base> + $0\ 9010_H$ + m × 40_H
FCNnMmDAT6H: <FCNn_base> + $0\ 9018_H$ + m × 40_H

FCNnMmDAT0W: <FCNn_base> + $1\ 1000_H$ + m × 40_H
FCNnMmDAT4W: <FCNn_base> + $1\ 1010_H$ + m × 40_H

Initial value 00_H

(a) FCNnMmDATxB (x = 0 to 7)

7	6	5	4	3	2	1	0
FCNnMmSSD0, FCNnMmSSD1, FCNnMmSSD2, FCNnMmSSD3, FCNnMmSSD4, FCNnMmSSD5, FCNnMmSSD6, FCNnMmSSD7							

(b) FCNnMmDATyH (y = 0, 2, 4, 6)

15	14	13	12	11	10	9	8
FCNnMmSSD0, FCNnMmSSD2, FCNnMmSSD4, FCNnMmSSD6							

7	6	5	4	3	2	1	0
FCNnMmSSD1, FCNnMmSSD3, FCNnMmSSD5, FCNnMmSSD7							

(c) FCNnMmDATzW (z = 0, 4)

31	30	29	28	27	26	25	24
FCNnMmSSD0, FCNnMmSSD4							

23	22	21	20	19	18	17	16
FCNnMmSSD1, FCNnMmSSD5							

15	14	13	12	11	10	9	8
FCNnMmSSD2, FCNnMmSSD6							

7	6	5	4	3	2	1	0
FCNnMmSSD3, FCNnMmSSD7							

(2) FCNnMmDTLGB - FCNn Message Data Length Register m

This register is used to set the number of bytes of the data field of a message buffer.

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

Address <FCNn_base> + $0\ 1020_H + m \times 40_H$

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	FCNnMmDTLG[3:0]			

FCNnMmDTLG[3:0]	Data Length of Transmit/Receive Message
0000_B	0 bytes
0001_B	1 byte
0010_B	2 bytes
0011_B	3 bytes
0100_B	4 bytes
0101_B	5 bytes
0110_B	6 bytes
0111_B	7 bytes
1000_B	8 bytes
1001_B	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set FCNnMmDTLG[3:0] value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.) *
1010_B	
1011_B	
1100_B	
1101_B	
1110_B	
1111_B	

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 FCNnMmDTLG[3:0] (However, 8 bytes if the set value is 8 or more)	Set value of FCNnMmDTLGB.FCNnMmDTLG[3:0] bits
Remote frame	0 bytes	

Caution 1. Be sure to set bits 7 to 4 to 0000_B.

Caution 2. Receive data is stored in as many FCNnMmDATxB register as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. The FCNnMmDATxB register in which no data is stored is undefined.

Caution 3. On reception, FCNnMmDTLGB is updated according to the received frame.

(3) FCNnMmSTRB - FCNn Message Configuration Register m

This register is used to specify the type of the message buffer and to set a mask.

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

Address <FCNn_base> + 0 1024_H + m × 40_H

Initial value 00_H

7	6	5	4	3	2	1	0
FCNnMmSSOW	FCNnMmSSMT[3:0]				FCNnMmSSRT	0	FCNnMmSSAM

FCNnMmSSOW	Overwrite Control
0	The message buffer that has already received a data frame*1 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*1 is overwritten by a newly received data frame.

Note 1. The “message buffer that has already received a data frame” is a receive message buffer whose FCNnMmCTL.FCNnMmDTNF bit has been set to 1.

Note Reception and storage of remote frames in the transmit message buffer do not depend on the settings of FCNnMmCTL.FCNnMmSSOW and FCNnMmCTL.FCNnMmDTNF. A remote frame that meets other conditions (matching of ID, FCNnMmSTRB.FCNnMmSSRT = 0, and FCNnMmCTL.FCNnMmTRQF = 0) is always received and stored in the corresponding message buffer (with interrupt generation, FCNnMmDTNF flag setting, FCNnMmDTLBG.FCNnMmDTLG[3:0] bits update, and receive history list recording).

FCNnMmSSRT	Remote Frame Request
0	Transmit or receive a data frame.
1	Transmit or receive a remote frame.

FCNnMmSTRB.FCNnMmSSRT specifies the type of message frame that is transmitted or received from/to a message buffer.

Note 1. If the message buffer is defined as a transmit message buffer, and a remote frame shall be received into it, the FCNnMmSSRT bit must be cleared.

Note 2. Even if a valid remote frame has been received in a transmit message buffer, the FCNnMmSSRT bit of the transmit message buffer that has received the frame remains cleared to 0.

Note 3. Even if a remote frame whose ID matches has been received from the CAN bus, if the FCNnMmSSRT bit of a transmit message buffer is set to 1 (to transmit a remote frame), that remote frame is not stored in this transmit message buffer.

Note 4. If the message buffer is defined as a receive message buffer, the FCNnMmSSRT bit must be set, in order to receive remote frames instead of data frames.

FCNnMmSSMT[3:0]	Message Buffer Type Setting
0000_B	Transmit message buffer
0001_B	Receive message buffer (no mask setting)
0010_B	Receive message buffer (mask 1 set)
0011_B	Receive message buffer (mask 2 set)
0100_B	Receive message buffer (mask 3 set)
0101_B	Receive message buffer (mask 4 set)
0110_B	Receive message buffer (mask 5 set)
0111_B	Receive message buffer (mask 6 set)
1000_B	Receive message buffer (mask 7 set)
1001_B	Receive message buffer (mask 8 set)
Other than above	Setting prohibited

Note The setting of FCNnMmSSMT is also valid to select masks in conjunction with remote frame reception. To receive remote frames in receive message buffers, the FCNnMmSSRT flag of the message buffer must be set.

FCNnMmSSAM	Message Buffer Assignment
0	Message buffer not used.
1	Message buffer used.

Caution Be sure to write 0 to bit 1.

(4) FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W - FCNn Message ID Register m

These registers are used to set an identifier (ID).

Access FCNnMmMID0H and FCNnMmMID1H can be read/written in 16-bit units. FCNnMmMID0W can be read/written in 32-bit units.

Address FCNnMmMID0H: <FCNn_base> + $0\ 9028_H + m \times 40_H$
FCNnMmMID1H: <FCNn_base> + $0\ 9030_H + m \times 40_H$

FCNnMmMID0W: <FCNn_base> + $1\ 1028_H + m \times 40_H$

Initial value 0000_H

(a) FCNnMmMID0H

15	14	13	12	11	10	9	8
FCNnMmSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnMmSSID[7:0]							

(b) FCNnMmMID1H

15	14	13	12	11	10	9	8
FCNnMmSSIE	0	0	FCNnMmSSID[28:24]				

7	6	5	4	3	2	1	0
FCNnMmSSID[23:16]							

(c) FCNnCMmMID0W

31	30	29	28	27	26	25	24
FCNnMmSSIE	0	0	FCNnMmSSID[28:24]				

23	22	21	20	19	18	17	16
FCNnMmSSID[23:16]							

15	14	13	12	11	10	9	8
FCNnMmSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnMmSSID[7:0]							

FCNnMmSSIE	Format Mode Specification
0	Standard format mode (FCNnMmSSID[28:18]: 11 bits, FCNnMmSSID[17:0] are not used)
1	Extended format mode (FCNnMmSSID[28:0]: 29 bits)

FCNnMmSSID[28:0]	Message ID
FCNnMmSSID[28:18]	Standard ID value of 11 bits (when FCNnMmSSIE = 0)
FCNnMmSSID[28:0]	Extended ID value of 29 bits (when FCNnMmSSIE = 1)

Caution 1. Be sure to write 0 to bits 14 and 13 of FCNnMmMID1H, and bits 30 and 29 of FCNnMmMID0W register.

Caution 2. Be sure to align the ID value with the specified bit position in this register. Note that for standard ID, the ID value must be shifted to fit into FCNnMmSSID[28:18] bit positions.

(5) FCNnMmCTL - FCNn Message Control Register m

This register is used to control the operation of the message buffer.

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

Address <FCNn_base> + $0\ 9038_H$ + m × 40_H

Initial value 0000_H

(a) FCNnMmCTL Read

15	14	13	12	11	10	9	8
0	0	FCNnMmMUCF	0	0	0	FCNnMmTCPF	0

7	6	5	4	3	2	1	0
0	FCNnMmNHMF	0	FCNnMmMOWF	FCNnMmIENF	FCNnMmDTNF	FCNnMmTRQF	FCNnMmRDYF

FCNnMmMUCF	Message Buffer Data Being Updated
0	The FCN module is not updating (receiving and storing) the message buffer.
1	The FCN module is updating (receiving and storing) the message buffer.

FCNnMmTCPF*1	Transmission Completion Flag
0	Transmission failed.*2
1	Transmission completed.

Note 1. FCNnMmTCPF is cleared if FCNnMmRDYF is changed or FCNnMmTRQF is set.

Note 2. This indicates a successful transmission abort, if this was requested by the application by clearing the FCNnMmTRQF flag.

FCNnMmNHMF	History Mask Flag *1
0	Update of receive/transmit history list registers FCNnCMRGRX/ FCNnCMRGTX is not masked.
1	Update of receive/transmit history list registers FCNnCMRGRX/ FCNnCMRGTX is masked.

Note 1. When masked, the history lists are not updated upon reception or transmission completion activity on this message buffer.

FCNnMmMOWF	Message Buffer Overwrite Status
0	The message buffer is not overwritten by a newly received data or remote frame.
1	The message buffer is overwritten by a newly received data or remote frame.

Note This bit is not set to 1 even if a remote frame is received and stored in the transmit message buffer of FCNnMmDTNF = 1 .

FCNnMmIENF	Message Buffer Interrupt Request Enable
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt and abort interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt and abort interrupt enabled.

Caution Be sure to set FCNnMmIENF and CNnMmRDYF separately.

FCNnMmDTNF	Message Buffer Data Update
0	No new data frame or remote frame has been stored in the message buffer.
1	A new data frame or remote frame has been stored in the message buffer.

Caution Do not set FCNnMmDTNF to 1 by software. Be sure to write 0 to bit 10.

FCNnMmTRQF	Message Buffer Transmission Request
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Caution 1. Do not set FCNnMmTRQE and FCNnMmRDYE at the same time.
Set FCNnMmRDYF = 1 and then set FCNnMmTRQF = 1.
Caution 2. Do not set FCNnMmTRQF to 1 for other than the transmit message buffer (buffer of FCNnMmSSMT[3:0] are not 4’b0000 or FCNnMmSSAM = 0).

FCNnMmRDYF	Message Buffer Ready
0	The message buffer can be written by software. The FCN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to FCNnMmRDYF, FCNnMmTRQF, FCNnMmDTNF, and FCNnMmMOWF). The FCN module can write to the message buffer.

Caution 1. Be sure to set FCNnMmIEnF and FCNnMmRDYF separately.
Caution 2. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time.
Set FCNnMmRDYF = 1 and then set CNnMmTRQF = 1.
Caution 3. Do not clear CNnMMRDYF to 0 during message transmission. To redefine the message buffer, perform the transmission abort processing to clear FCNnMmRDYF.
Caution 4. Clearing of FCNnMmRDYF may take some time, depending on activity of the

CAN controller. Repeat the clearing access until reading of FCNnMmRDYF confirms that the bit has been cleared.

Caution 5. Make sure that FCNnMmRDYF is cleared before writing to another FCN message buffer registers by checking the status of FCNnMmRDYF.

(b) FCNnMmCTL Write

15	14	13	12	11	10	9	8
0	FCNnMmSENH	0	0	FCNnMmSEIE	0	FCNnMmSETR	FCNnMmSERY

7	6	5	4	3	2	1	0
0	FCNnMmCLNH	0	FCNnMmCLMW	FCNnMmCLIE	FCNnMmCLDN	FCNnMmCLTR	FCNnMmCLRY

FCNnMmSENH	FCNnMmCLNH	Setting FCNnMmNHMF
0	1	FCNnMmNHMF is cleared to 0.
1	0	FCNnMmNHMF is set to 1.
Other than above		FCNnMmNHMF is not changed.

FCNnMmCLMW	Setting FCNnMmMOWF
0	FCNnMmMOWF is not changed.
1	FCNnMmMOWF is cleared to 0.

FCNnMmSEIE	FCNnMmCLIE	Setting FCNnMmIENF
0	1	FCNnMmIENF is cleared to 0.
1	0	FCNnMmIENF is set to 1.
Other than above		FCNnMmIENF is not changed.

Caution Be sure to set FCNnMmIENF and FCNnMmRDYF separately.

FCNnMmCLDN	Setting FCNnMmDTNF
0	FCNnMmDTNF is not changed.
1	FCNnMmDTNF is cleared to 0.

Note If FCNnMmDTNF is cleared by the completion of ID field receiving, the message buffer participates in the search to store the receiving frame.

Caution Do not set FCNnMmDTNF to 1 by software. Be sure to write 0 to bit 10.

FCNnMmSETR	FCNnMmCLTR	Setting FCNnMmTRQF
0	1	FCNnMmTRQF is cleared to 0.
1	0	FCNnMmTRQF is set to 1.
Other than above		FCNnMmTRQF is not changed.

Caution Do not set FCNnMmTRQF and FCNnMmRDYF at the same time. Set FCNnMmRDYF = 1, and then set FCNnMmTRQF = 1.

FCNnMmSERY	FCNnMmCLRY	Setting FCNnMmRDYF
0	1	FCNnMmRDYF is cleared to 0.
1	0	FCNnMmRDYF is set to 1.
Other than above		FCNnMmRDYF is not changed.

Caution 1. Be sure to set FCNnMmIENF and FCNnMmRDYF separately.

Caution 2. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1, and then set FCNnMmTRQF = 1.

Caution 3. Do not clear FCNnMmRDYF to 0 during message transmission. To redefine the message buffer, perform the transmission abort processing to clear FCNnMmRDYF.

Caution 4. Clearing of FCNnMmRDYF may take some time, depending on activity of the CAN controller. Repeat the clearing access until reading of FCNnMmRDYF confirms that the bit has been cleared.

Caution 5. Make sure that FCNnMmRDYF is cleared before writing to another message buffer register by checking the status of FCNnMmRDYF.

20.6 CAN Controller Initialization

20.6.1 Initialization of FCN Module

To enable FCN module operation, the FCN module system clock needs to be determined by setting FCNnGMCSPRE.FCNnGMCSPRSC[3:0] by software. Do not change the setting of the FCN module system clock after FCN module operation is enabled.

The FCN module is enabled by setting FCNnGMCLCTL.FCNnGMCLPWOM.

For the procedure of initializing the FCN module, see section 20.14, Operation of the CAN Controller.

20.6.2 Redefinition of Message Buffer

Redefining a message buffer is to change the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) Redefining Message Buffer in Initialization Mode

Place the FCN module in initialization mode once and then change the ID and control information of the message buffer in initialization mode. After the ID and control information are changed, set the FCN module to an operating mode.

(2) Redefining Message Buffer during Reception

Perform redefinition as shown in *figure20-17 Message Buffer Redefinition during Reception*.

(3) Redefining 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 (see the description in section 20.8.4, (1) Transmission Abort Process except for in Normal Operating Mode with Automatic Block Transmission (ABT) and ((2) Transmission Abort Process in Normal Operating Mode with Automatic Block Transmission (ABT)). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After the transmit message buffer is redefined, 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.

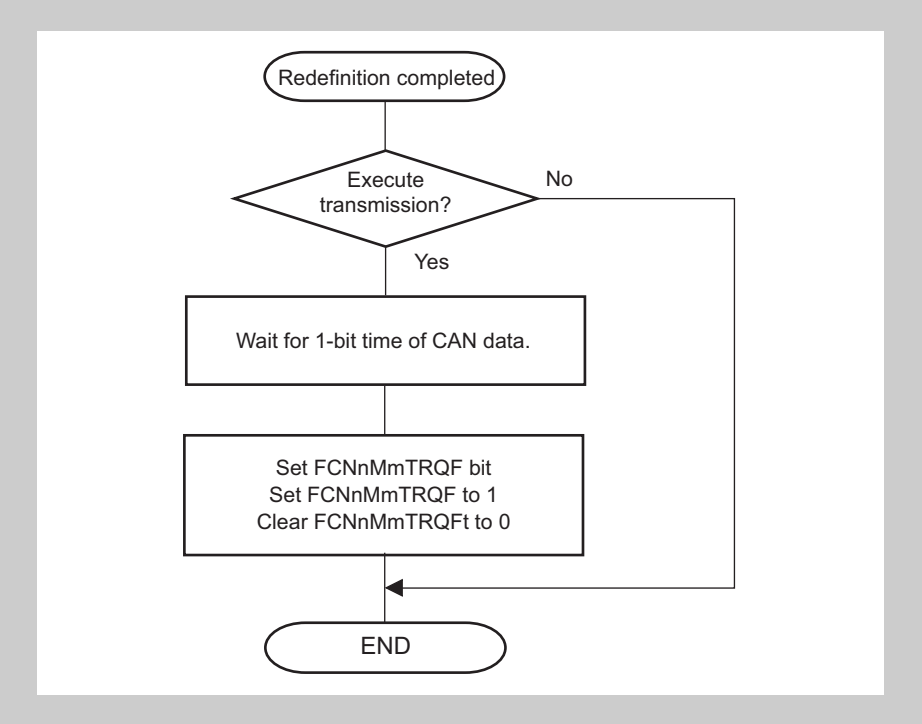


Figure 20-4 Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition

Caution 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 *figure20-17 Message Buffer Redefinition during Reception*, 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.

Caution 2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and FCNnMmSTRB.FCNnMmSSRT 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 *figure20-4 Setting Transmission Request (FCNnMmCTL.FCNnMmTRQF) to Transmit Message Buffer after Redefinition* to Transmit Message Buffer after Redefinition, is not observed, a message with an ID having the highest priority may not be transmitted after redefinition.

20.6.3 Transition from Initialization Mode to Operating Mode

The FCN module can be switched to the following operating modes.

- Normal operating mode
- Normal operating mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

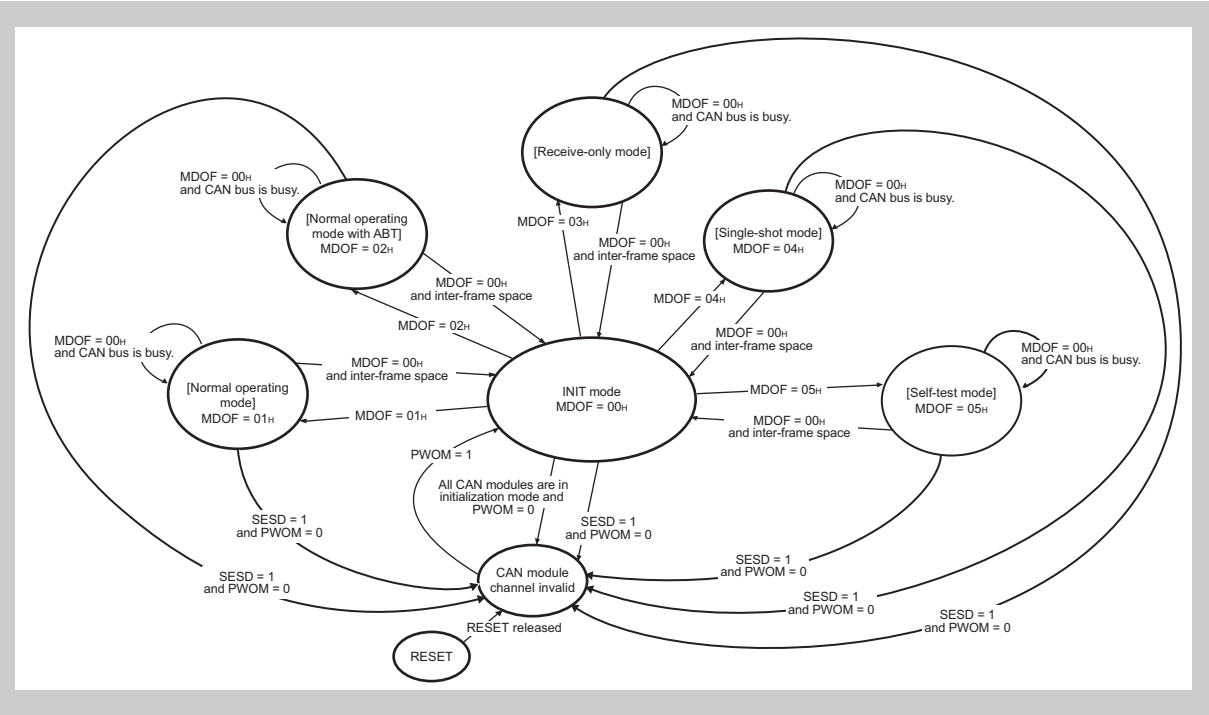


Figure 20-5 Transition to Operating Modes

Note The following abbreviations are used in the figure above.

- MDOF = FCNnCMCLCTL.FCNnCMCLMDOF[2:0]
- PWOM = FCNnGMCLCTL.FCNnGMCLPWOM
- SESD = FCNnGMCLCTL.FCNnGMCLSESD

The transition from initialization mode to an operating mode is controlled by FCNnCM.FCNnCMCLMDOF[2:0].

Changing from one operating mode to another requires shifting to initialization mode in between. Do not change one operating mode to another directly. Otherwise, the operation will not be guaranteed.

Requests for transition from an operating mode to initialization mode are held pending when the CAN bus is not in the inter-frame space (i.e., frame reception or transmission is in progress), and the FCN module enters initialization mode at the first bit in the inter-frame space (the values of FCNnCMCLCTL.FCNnCMCLMDOF[2:0] are changed to 000_B). After issuing a request to change the mode to initialization mode, read FCNnCMCLCTL.FCNnCMCLMDOF[2:0] until their values become 000_B to confirm that the module has entered initialization mode (see *figure20-14 Re-initialization without Software Reset Function*).

20.7 Reception of Message

20.7.1 Message Reception

In all operating modes, the entire message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (FCNnMmSTRB.FCNnMmSSAM = 1)
- Set as a receive message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B to 1001_B)
- Ready for reception
  (FCNnMmCTL.FCNnMmRDYF = 1)

When two or more message buffers of the FCN module are able to receive a message, the message is stored according to the priority described below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to store a message (i.e., when FCNnMmCTL.FCNnMmDTNF = 1 indicating that a message has already been received, but rewriting is disabled because FCNnMmSTRB.FCNnMmSSOW = 0). In this case, the message is not actually stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Table 20-16 MBRB Priorities

Priority	Storing Condition if Same ID is Set	
1 (high)	Unmasked message buffer	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
2	Message buffer linked to mask 1	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
3	Message buffer linked to mask 2	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
...	...	...
9 (low)	Message buffer linked to mask 8	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1

20.7.2 Receive Data Read

To keep data consistency when reading message buffers, read data according to the procedures in *figure20-30 Reception via Interrupt (Using FCNnCMLISTR Register)*, to *figure20-33 Reception via Software Polling*.

During message reception, the FCN module sets FCNnMmCTL.FCNnMmDTNF two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, FCNnMmCTL.FCNnMmMUCF of the message buffer is set (see *figure20-6 Reception Timing*).

The receive history list is also updated just before the storage process. In addition, during storage process (FCNnMmCTL.FCNnMmMUCF = 1), FCNnMmCTL.FCNnMmRDYF of the message buffer is locked to avoid any coincidental data write by the CPU. Note that the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

Caution To securely store a message in the message buffer, the FCNnMmCTL.FCNnMmDTNF bit of the buffer must be cleared before the message search processing is started (before the frame ID is output to the bus). This is the 15th of CAN bits after EOF of the previous frame at shortest. To securely receive CAN frames that continuously appear on the bus, it is recommended that one more message buffers for frame reception be used.

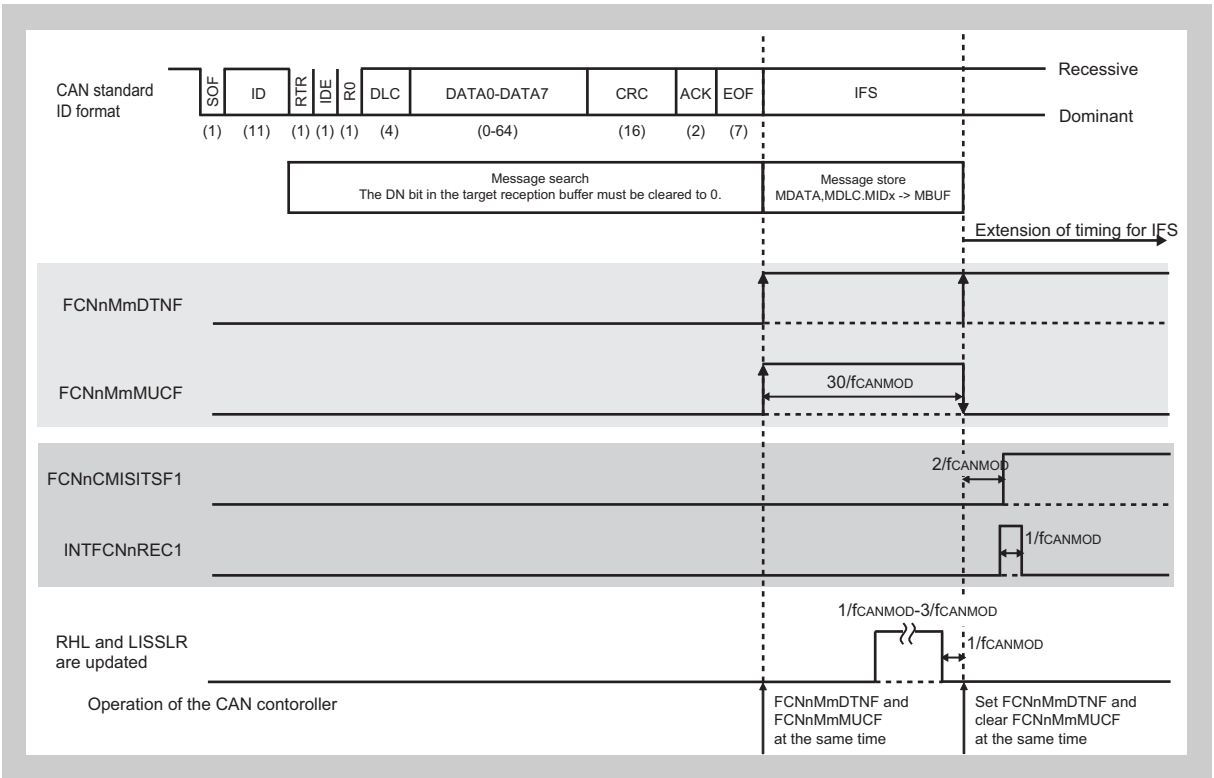


Figure 20-6 Reception Timing

20.7.3 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 47 messages, the last in-message pointer FCNnCMLISSLR[7:0] with the corresponding FCNnCMLISTR register and the receive history list get pointer FCNnCMRGSSPT with the corresponding FCNnCMRGRX register.

The RHL is undefined immediately after the transition of the FCN module from initialization mode to one of operating modes.

The FCNnCMLISTR register holds the contents of the RHL element indicated by the value of the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer minus 1. By reading the FCNnCMLISTR register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The FCNnCMLISSLR[7:0] pointer is used as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Each 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 FCNnCMLISSLR[7:0] pointer. Each time recording to the RHL has been completed, the FCNnCMLISSLR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame is chronologically recorded.

For message buffers where the FCNnMmCTL.FCNnMmNHMF flag is set, no entry in the receive history list is recorded.

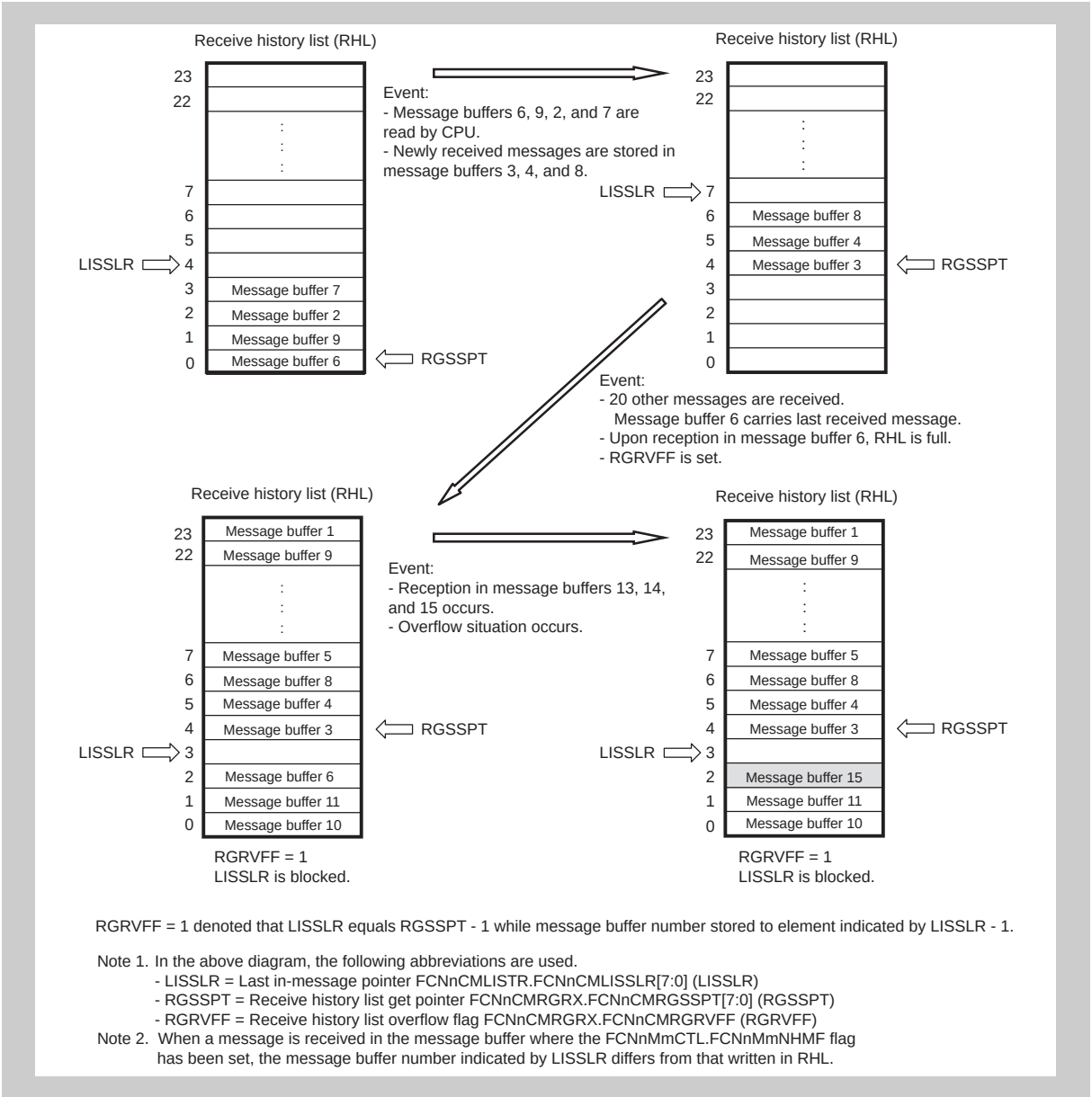
The FCNnCMRGRX.FCNnCMRGSSPT pointer is used as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the FCNnCMRGRX register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the FCNnCMRGRX register, the FCNnCMRGSSPT pointer is automatically incremented.

If the FCNnCMRGRX.FCNnCMRGSSPT pointer value matches the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer value, FCNnCMRGRX.FCNnCMRGSSPM (receive history list pointer match) 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 FCNnCMLISSLR[7:0] pointer is incremented and because its value no longer matches the FCNnCMRGSSPT pointer value, FCNnCMRGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer is incremented and matches the value of the FCNnCMRGRX.FCNnCMRGSSPT pointer minus 1, FCNnCMRGRX.FCNnCMRGRVFF (receive history list overflow) 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 newly received message. In this case, after FCNnCMRGRVFF has been set to 1, the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However, messages are not lost and can be located by CPU search in message buffer memory with the help of FCNnMmCTL.FCNnMmDTNF, or by reading the FCN global registers FCNnDNBMRX[3:0].

Caution Even if the receive history list is in the overflow state (FCNnCMRGRX.FCNnCMRGRVFF is set), the receive history list contents can still be read until the history list is empty (indicated by FCNnCMRGRX.FCNnCMRGSSPM flag set). Nevertheless, the receive history list remains in the overflow state, until FCNnCMRGRVFF is cleared by software. Unless FCNnCMRGRVFF is cleared, the FCNnCMRGSSPM flag is also not updated (cleared) upon a message storage of a newly received frame. This may lead to the situation that FCNnCMRGSSPM indicates an empty receive history list, although a reception has taken place, while the receive history list is in the overflow state (FCNnCMRGRVFF and FCNnCMRGSSPM are set).

As long as the RHL still has free entries, the sequence of reception is maintained. If more reception occurs before the RHL is read by the host processor, complete sequence of receptions cannot be recovered.



RGRVFF = 1 denoted that LISSLR equals RGSSPT - 1 while message buffer number stored to element indicated by LISSLR - 1.

Note 1. In the above diagram, the following abbreviations are used.
- LISSLR = Last in-message pointer FCNnCMLISTR.FCNnCMLISSLR[7:0] (LISSLR)
- RGSSPT = Receive history list get pointer FCNnCMRGRX.FCNnCMRGSSPT[7:0] (RGSSPT)
- RGRVFF = Receive history list overflow flag FCNnCMRGRX.FCNnCMRGRVFF (RGRVFF)

Note 2. When a message is received in the message buffer where the FCNnMmCTL.FCNnMmNHMF flag has been set, the message buffer number indicated by LISSLR differs from that written in RHL.

Figure 20-7 Receive History List

20.7.4 Mask Function

For any message buffer, which is used for reception, the assignment to one of eight global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is enabled, 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 to ID22 are 1, are to be stored in message buffer 14. The setting 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

(2) Identifier to be Configured in Message Buffer 14 (Example) (Using FCNnCMMKCTL14 Register)

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				

Note 1. ID with the ID27 to ID25 bits cleared to 0 and the ID24 to ID22 bits set to 1 is registered (initialized) in message buffer 14.

Note 2. Message buffer 14 is set as a standard format identifier that is linked to mask 1 (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0010_B).

(3) Mask setting for FCN module 1 (mask 1) (example) (using CAN1 address mask 1 register FCNnCMMKCTL01)

FCNnCMMKSSID[..]

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
1	0	0	0	0	1	0	1	1	1	1
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
1	1	1	1	1	1	1	1	1	1	1
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
1	1	1	1	1	1	1				

1: Not compared (masked)
0: Compared

Clear FCNnCMMKSSID[27:24] and FCNnCMMKSSID[22] to 0, and set FCNnCMMKSSID[28], FCNnCMMKSSID[23], and FCNnCMMKSSID[21:0] to 1.

20.7.5 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. These message buffers can be allocated anywhere in the message buffer memory and need not be adjacent with each other.

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 the ID of the message buffers is received, it is stored in message buffer 10. At this point, FCNnMmCTL.FCNnMmDTNF 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 FCNnMmCTL.FCNnMmIENF 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. FCNnMmIENF in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and FCNnMmIENF in message buffer k-1 is set to 1 (interrupts enabled). In that 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 FCNnMmIENF of message buffers 0 to (k-3) and setting FCNnMmIENF 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.

Caution 1. MBRB can be configured for each message buffer type.Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different is empty, the received message is not stored in that message buffer, but instead discarded.

Caution 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.

Caution 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.

Caution 4. Regarding 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 to be a matching ID and the buffer that has this ID is treated as the storage destination of a message.

Caution 5. For the priority between MBRBs, see table 20-16, MBRB Priorities.

20.7.6 Remote Frame Reception

In all operating modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers that meet the following conditions (1 and 2, condition 1 has priority on reception acceptance). If condition 1 is not met, the remaining message buffers are searched, whether condition 2 could be met.

- Condition 1:
  Set as a transmit message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B)
  - Used as a message buffer
    (FCNnMmSTRB.FCNnMmSSAM = 1)
  - Ready for reception
    (FCNnMmCTL.FCNnMmRDYF = 1)
  - Set for data frame message type
    (FCNnMmSTRB.FCNnMmSSRT = 0)
  - Transmission request is not set.
    (FCNnMmCTL.FCNnMmTRQF = 0)
- Condition 2:
  Set as a receive message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B ... 1001_B)
  - Used as a message buffer
    (FCNnMmSTRB.FCNnMmSSAM = 1)
  - Ready for reception
    (FCNnMmCTL.FCNnMmRDYF = 1)
  - Set for remote frame message type
    (FCNnMmSTRB.FCNnMmSSRT = 1)
  - Buffer is ready to store a message.
    (FCNnMmCTL.FCNnMmDTNF = 0, or FCNnMmSTRB.FCNnMmSSOW = 1 and FCNnMmCTL.FCNnMmDTNF = 1)

Upon reception 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 FCNnMmDTLG[3:0] bits in the FCNnMmDTLGB register store the received DLC value.
- In reception to the transmit message buffer, the FCNnMmDAT0B to FCNnMmDAT7B registers in the data area are not updated (data before reception is saved).
- FCNnMmCTL.FCNnMmDTNF is set to 1.
- FCNnCMISCTL.FCNnCMISITSF1 is set to 1 (if FCNnMmCTL.FCNnMmIENF of the message buffer that receives and stores the frame is set to 1).
- The receive completion interrupt (INTCnREC) is output (if FCNnMmCTL.FCNnMmIENF of the message buffer that receives and stores the frame is set to 1 and if FCNnCMIECTL.FCNnCMIESEIE1 is set to 1).
- The message buffer number is recorded in the receive history list, if the FCNnMmCTL.FCNnMmNHMF flag is not set.

Caution When a transmit message buffer is found for receiving and storing a remote frame, overwrite control by FCNnMmSTRB.FCNnMmSSOW of the message buffer and FCNnMmCTL.FCNnMmDTNF are not checked. The setting of FCNnMmSSOW is ignored, and FCNnMmDTNF is set in any case.

Note 1. If two or more transmit message buffers have 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.

Note 2. If transmit and receive message buffers are found, which can receive a remote frame matching with its ID, either masked or unmasked, the remote frame is stored in the transmit message buffer.

Note 3. If there are receive message buffers that meet remote frame receiving conditions, the reception priority is identical to the data frame reception priority.

Note 4. If a receive message buffer is found to match for a remote frame reception, and selected for storage, but this receive message buffer does not allow the storage, because FCNnMmDTNF is set and FCNnMmSSOW is not set, the remote frame is not stored at all.

20.8 Transmission of Message

20.8.1 Message Transmission

A message buffer with its FCNnMmCTL.FCNnMmTRQF bit set to 1 participates in the search for the most high-prioritized message when the following conditions are met. This processing is valid for all operating modes.

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Set as a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B)
- Ready for transmission (FCNnMmCTL.FCNnMmRDYF = 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 FCN 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).

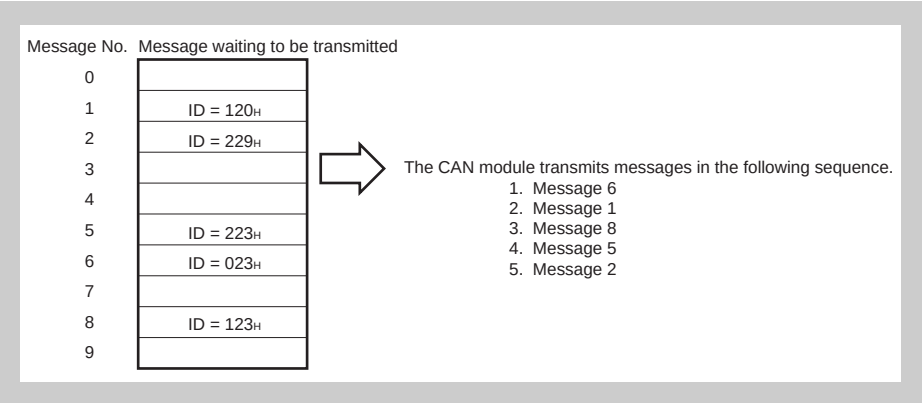


Figure 20-8 Example of Message Processing

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 FCNnMmCTL.FCNnMmTRQF 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. To solve this priority inversion problem, the software can perform a transmission abort request for the lower priority message. The priority order is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of upper 11 bits of ID[ID28 to ID18]:	The message frame with the lowest value represented by the upper 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the upper 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 (FCNnMmSTRB.FCNnMmSSRT 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 (message buffer identifier register FCNnMmMID... bit FCNnMmSSIE 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 two or more transmission-pending extended ID message frames have equal values in the upper 11 bits of the ID and the same frame type (equal FCNnMmSTRB.FCNnMmSSRT 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.

Note 1. If the automatic block transmission request bit FCNnGMABCTL.FCNnGMABABTT is set to 1 in the normal operating mode with ABT, FCNnMmCTL.FCNnMmTRQF is set to 1 only for one message buffer in the ABT message buffer group.

If ABT mode is triggered by FCNnGMABCTL.FCNnGMABSEAT = 1, one FCNnMmCTL.FCNnMmTRQF is set to 1 in the ABT area (FCN: 0 to 15 of 64 message buffers). Beyond this transmit request, the application can request transmissions (set FCNnMmTRQF to 1) for other transmit message buffers that do not belong to the ABT area. In that case, an interval arbitration process (transmission search) evaluates all transmit message buffers with FCNnMmTRQF set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are two or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted first.

Upon successful transmission of a message frame, the following operations are performed.

- The FCNnMmCTL.FCNnMmTRQF flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1 (if the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTCnTRX is output (if FCNnCMIECTL.FCNnCMIESEIE0 is set to 1 and the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1).

Note 2. When changing the contents of a transmit buffer, the FCNnMmCTL.FCNnMmRDYF flag of this buffer must be cleared before updating the buffer contents. Since the FCNnMmRDYF flag may be locked temporarily during internal transfer operation, the status of the FCNnMmRDYF flag must be checked by software after it is changed.

20.8.2 Transmit History List Function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been sent. The THL consists of storage elements equivalent to up to 15 messages, the last out-message pointer FCNnCMLOSTR[7:0] with the corresponding FCNnCMLOSTR register, and the transmit history list get pointer FCNnCMTGSSPT[7:0] with the corresponding FCNnCMTGTX register.

The THL is undefined immediately after the transition of the FCN module from initialization mode to one of operating modes.

The FCNnCMLOSTR register holds the contents of the THL element indicated by the value of the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer minus 1. By reading the FCNnCMLOSTR register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The FCNnCMLOSTR[7:0] 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 FCNnCMLOSTR[7:0] pointer. Each time recording to the THL has been completed, the FCNnCMLOSTR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

For message buffers where the FCNnMmCTL.FCNnMmNHMF flag is set, no entry in the history lists is recorded.

The FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer is used 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 FCNnCMTGTX 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 FCNnCMTGTX register, the FCNnCMTGSSPT[7:0] pointer is automatically incremented.

If the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer value matches the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer value, FCNnCMTGTX.FCNnCMTGSSPM (transmit history list pointer match) is set to1. 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 FCNnCMLOSTR[7:0] pointer is incremented and because its value no longer matches the value of the FCNnCMTGSSPT[7:0] pointer, FCNnCMTGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer is incremented and matches the value of the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer minus 1, FCNnCMTGTX.FCNnCMTGTVFF (transmit history list overflow) 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 message buffer number that transmitted its message afterwards. In that case, after FCNnCMTGTVFF has been set to 1, the recorded message buffer numbers in the THL do not completely reflect the chronological order. However, the CPU can identify the numbers of message buffers that have completed transmission by searching all transmit buffers (before re-setting transmission).

Regardless of the FCNnCMTGTX.FCNnCMTGTVFF value, 14 (64 message buffers) or 30 (128 message buffers) transmit message buffer numbers are stored in the THL.

Caution If the transmit history list is in the overflow condition (FCNnCMTGTX.FCNnCMTGTVFF is set), reading the transmit history list contents is still possible, until the history list is empty (indicated by FCNnCMTGTX.FCNnCMTGSSPM flag set). Nevertheless, the transmit history list remains in the overflow condition until FCNnCMTGTVFF is cleared by software. If FCNnCMTGTVFF is not cleared, the FCNnCMTGTX.FCNnCMTGSSPM flag is not updated (cleared) upon successful transmission of a new message. This may lead to the situation, that FCNnCMTGSSPM indicates an empty transmit history list, although a successful transmission has taken place, while the history list is in the overflow state (FCNnCMTGTVFF and FCNnCMTGSSPM are set).

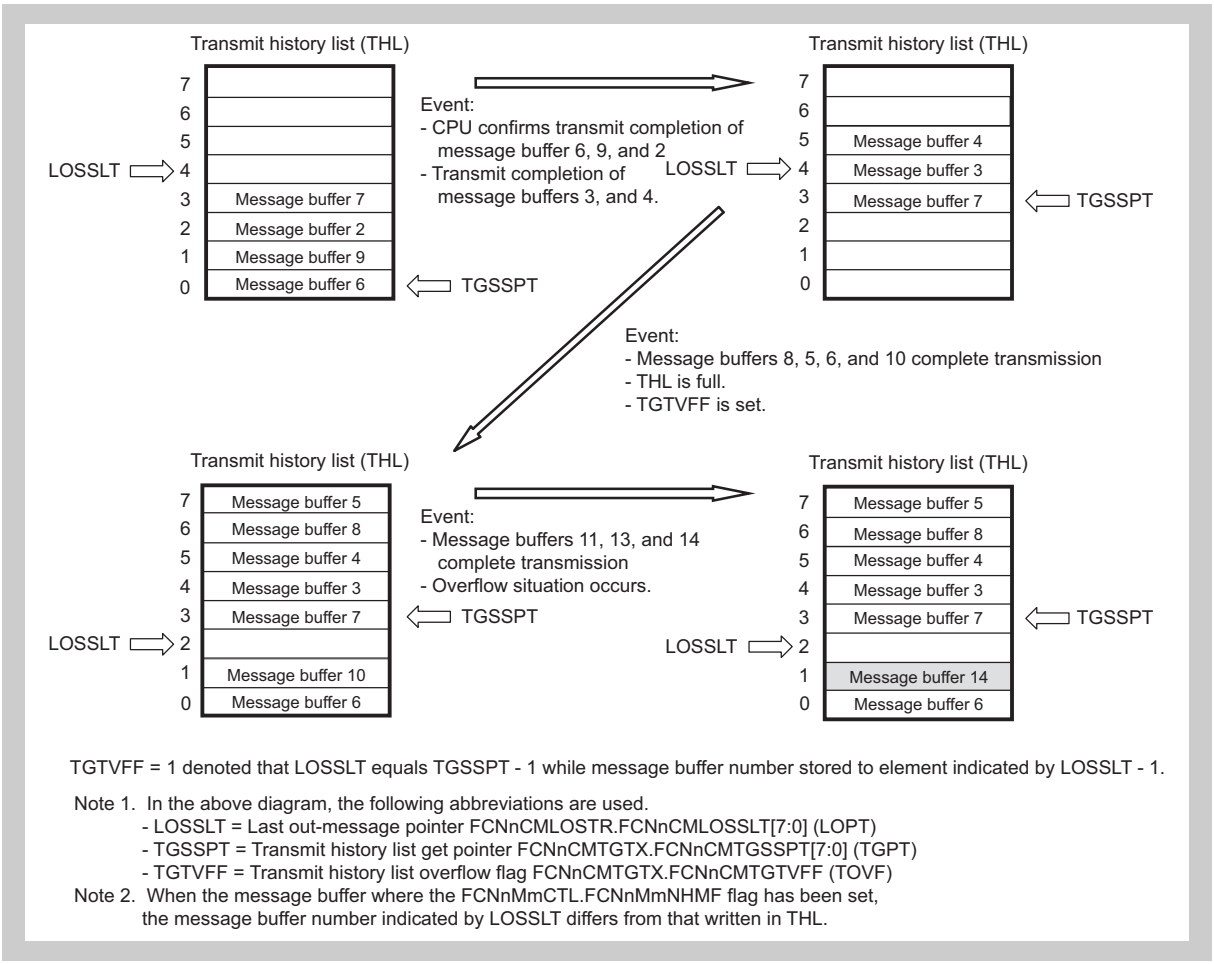


Figure 20-9 Transmit History List

20.8.3 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 16. They are always assigned sequentially from the lowest message buffer number.

By setting FCNnCM.FCNnCMCLMDOF[2:0] to 010_B, “normal operating 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 FCNnMmSTRB.FCNnMmSSAM to 1 in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting the FCNnMmSTRB.FCNnMmSSMT[3:0] bits to 0000_B. Be sure to set the same ID for the message buffers for ABT 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 FCNnMmMID0H and FCNnMmMID1H or FCNnMmMID0W registers. Set the FCN message buffer data byte registers before issuing a transmission request using the ABT function.

After initialization of message buffers for ABT is completed, FCNnMmCTL.FCNnMmRDYF needs to be set to 1. In ABT mode, FCNnMmCTL.FCNnMmTRQF does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set FCNnGMABCTL.FCNnGMABSEAT to 1. Automatic block transmission is then started. When ABT is started, FCNnMmCTL.FCNnMmTRQF in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 is completed, the FCNnMmTRQF of the next message buffer, message buffer 1, is automatically set. In this way, transmission is performed successively.

A delay time can be inserted by the program in the interval in which the transmission request FCNnMmCTL.FCNnMmTRQF is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the FCNnGMADCTL register. The unit of the delay time is DBT (data bit time). DBT depends on the settings of the FCNnCMBRPRS and FCNnCMBTCTL registers.

Among transmit objects in the ABT area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from the last message buffer 7 has been completed, FCNnGMABCTL.FCNnGMABABTT is automatically cleared to 0 and the ABT operation is completed.

If FCNnMmCTL.FCNnMmRDYF of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is suspended, and FCNnGMABCTL.FCNnGMABABTT is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting FCNnMmRDYF and FCNnGMABABTT to 1 by software. To not resume transmission from the message buffer where ABT stopped, set the FCNnGMABCTL.FCNnGMABCLRF bit to 1 while ABT mode is stopped and FCNnGMABABTT is cleared to 0 to reset the internal ABT engine. In this case, transmission starts from message buffer 0 if FCNnGMABCTL.FCNnGMABSEAC is cleared to 0 and then FFCNnGMABABTT 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, FCNnMmCTL.FCNnMmIENF of each message buffer except the last message buffer needs to be cleared to 0.

If a transmit message buffer other than those used by the ABT function is assigned to a transmit message buffer, the message to be transmitted next 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 that are not used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Caution 1. Set FCNnGMABCTL.FCNnGMABSEAC to 1 while FCNnGMABCTL.FCNnGMABABTT is cleared to 0 in order to resume ABT operation at buffer No. 0. If FCNnGMABSEAC is set to 1 while FCNnGMABABTT is set to 1, the subsequent operation is not guaranteed.

Caution 2. If the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC to 1, FCNnGMABSEAC is automatically cleared immediately after the processing of the clearing request is completed.

Caution 3. Do not trigger automatic block transmission in initialization mode. If FCNnGMABCTL.FCNnGMABSEAT is set in initialization mode, the proper operation is not guaranteed after the mode is changed from initialization mode to ABT mode.

Caution 4. Do not set FCNnMmCTL.FCNnMmTRQF of the ABT message buffers to 1 by software in normal operating mode with ABT. Otherwise, the operation is not guaranteed.

Caution 5. The FCNnGMADCTL register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of FCNnMmCTL.FCNnMmTRQF 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 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 ABT messages.

Caution 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 (FCNnGMADCTL = 00_H), messages other than ABT messages may be transmitted regardless of the difference in priority from the ABT message.

Caution 7. Do not clear FCNnMmCTL.FCNnMmRDYF to 0 when FCNnGMABCTL.FCNnGMABABTT = 1.

Caution 8. If a message is received from another node while normal operating mode with ABT is active, the transmit message from the ABT area may be transmitted with a delay of one frame although the FCNnGMADCTL register is set to 00_H.

20.8.4 Transmission Abort Process

(1) Transmission Abort Process except for in Normal Operating Mode with Automatic Block Transmission (ABT)

The user can clear FCNnMmCTL.FCNnMmTRQF to 0 to abort a transmission request. FCNnMmTRQF will be cleared immediately when the abort is successful. Whether the transmission was successfully aborted or not can be checked by using FCNnCMCLCTL.FCNnCMCLSSTS and the FCNnCMTGTX register or the FCNnMmCTL.FCNnMmTCPF flag, which indicate the transmission status on the CAN bus (for details, see the procedure in *figure20-24 Transmission Abort Processing (Except Normal Operating Mode with ABT)*).

(2) Transmission Abort Process in Normal Operating Mode with Automatic Block Transmission (ABT)

The user can clear FCNnGMABCTL.FCNnGMABABTT to 0 to abort a transmission request. After checking FCNnGMABABTT = 0, clear FCNnMmCTL.FCNnMmTRQF to 0. FCNnMmTRQF will be cleared immediately if the abort is successful. Whether the transmission was successfully aborted or not can be checked by using FCNnCMCLCTL.FCNnCMCLSSTS and the FCNnCMTGTX register or the FCNnMmCTL.FCNnMmTCPF flag, which indicate the transmission status on the CAN bus (for details, see the procedure in *figure20-25 Transmission Abort Processing (Normal Operating Mode with ABT) - Retransmission Option for Aborted Message*).

(3) Transmission Abort Process for ABT Transmission in Normal Operating Mode with Automatic Block Transmission (ABT)

To abort ABT that is already started, clear FCNnGMABCTL.FCNnGMABCLAT to 0. In this case, FCNnGMABCTL.FCNnGMABABTT remains 1 until the transmission is completed (successfully or not) if an ABT message is currently being transmitted, and is cleared to 0 as soon as transmission is completed. This aborts ABT.

If the last transmission (before ABT is aborted) is successful, the internal ABT pointer of the normal operating mode with ABT points to the next message buffer to be transmitted.

In the case of a failure in transmission, the position of the internal ABT pointer depends on the status of FCNnMmCTL.FCNnMmTRQF in the last transmitted message buffer. If FCNnMmTRQF is set to 1 when clearing FCNnGMABCTL.FCNnGMABABTT is requested, the internal ABT pointer points to the last transmitted message buffer. (For details, see the procedure in *figure20-26 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT)*.) If FCNnMmTRQF is cleared to 0 when clearing FCNnGMABCTL.FCNnGMABABTT is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area. (For details, see the procedure in *figure20-27 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT) with Transmission Abort Interrupt Flag*.)

Caution Be sure to abort ABT by clearing FCNnGMABCTL.FCNnGMABCLAT to 0. The operation is not guaranteed if aborting transmission is requested by clearing FCNnMmCTL.FCNnMmRDYF.

When normal operating mode with ABT is resumed after ABT has been aborted and FCNnGMABCTL.FCNnGMABSEAT is set to 1, the ABT message buffer to be transmitted next is determined from the following table.

Status of FCNnMmCTL.FCNnMmTRQF of ABT Message Buffer	Abort of ABT after Successful Transmission	Abort of ABT after Erroneous Transmission
Set to 1	Next message buffer in the ABT area *1	Same message buffer in the ABT area
Cleared to 0	Next message buffer in the ABT area *1	Next message buffer in the ABT area *1

Note 1. 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 highest ABT message buffer is in progress is regarded as completion of ABT, rather than abort, if transmission of this message buffer has been successfully completed, even if FCNnGMABCTL.FCNnGMABABTT is cleared to 0. If FCNnMmCTL.FCNnMmRDYF 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 FCNnGMABABTT is set to 1, and ABT ends immediately.

20.8.5 Remote Frame Transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted by using FCNnMmSTRB.FCNnMmSSRT. Setting FCNnMmSSRT to 1 sets remote frame transmission.

20.9 Power Save Modes

20.9.1 FCN Sleep Mode

FCN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The FCN module can enter FCN sleep mode from all operating modes. Release of FCN sleep mode returns the FCN module to exactly the same operating mode from which FCN sleep mode was entered.

In FCN sleep mode, the FCN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering FCN Sleep Mode

The CPU issues an FCN sleep mode transition request by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B.

This transition request is acknowledged only under the following conditions.

1. The FCN module is already in one of the following operating modes.
   – Normal operating mode
   – Normal operating mode with ABT
   – Receive-only mode
   – Single-shot mode
   – Self-test mode
   – FCN stop mode in all above operating modes
2. The CAN bus state is bus idle (the fourth bit in the inter-frame space is recessive).
   If the CAN bus is fixed to dominant, the request for transition to FCN sleep mode is held pending. Also the transition from FCN stop mode to FCN sleep mode is independent of the CAN bus state.
3. No transmission request is pending.
4. Do not use power save mode in combination with the operating mode change. Make these accesses in different steps.

Note If a sleep mode request is pending and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been completed. This may cause the FCN to enter sleep mode while the CPU is executing the reception interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as receive history list registers by using the FCNnGMCLSSMO flag, if sleep mode is used.

If none of the above conditions is met, the FCN module will operate as follows.

- If FCN sleep mode is requested from initialization mode, the FCN sleep mode transition request is ignored and the FCN module remains in initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when a transition to FCN sleep mode is requested from one of operating modes, immediate transition to FCN sleep mode is not possible. In this case, FCN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the fourth bit in the inter-frame space is recessive). During the time period from the FCN sleep mode request to successful transition, FCNnCMCLCTL.FCNnCMCLMDPF[1:0]

remain 00_B. When the module has entered FCN sleep mode, the FCNnCMCLMDPF[1:0] bits are set to 01_B.

- If a request for transition to initialization mode and a request for transition to FCN sleep mode are made at the same time while the FCN module is in one of operating modes, the request for initialization mode is enabled. The FCN module enters initialization mode at a predetermined timing. At this time, the FCN sleep mode request is not held pending and is ignored.
- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for transition to initialization mode has priority over the sleep mode request. The sleep mode request is cancelled when the transition to initialization mode is requested. While a request for transition to initialization mode is pending, a subsequent request for sleep mode request is cancelled right at the point in time where it was issued.

(2) FCN Sleep Mode Status

The FCN module is in the following state after it enters FCN sleep mode.

- The internal operating clock stops and the power consumption is minimized.
- The function to detect the falling edge of the FCN reception pin (CRXDn) remains in effect to wake up the FCN module from the CAN bus.
- To wake up the FCN module from the CPU, data can be set for FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- FCNnDNBMRX registers cannot be read.
- A request for transition to initialization mode is not acknowledged and is ignored.

(3) Releasing FCN Sleep Mode

FCN sleep mode is released by the following events.

- Setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 00_B by the CPU
- A falling edge of the FCN reception pin CRXDn (i.e. the CAN bus level shifts from recessive to dominant)

Caution Even if the falling edge belongs to the SOF of a receive message, this message will not be received or stored. If the CPU stops supplying the clock to the FCN module while the FCN module is in sleep mode, FCN sleep mode will not be released and FCNnCMCLMDPF[1:0] will remain 01_B unless the clock to the FCN module is supplied again. In addition to this, the receive message will not be received after that.

After releasing sleep mode, the FCN module returns operating mode from which FCN sleep mode was requested. After that, FCNnCMCLCTL.FCNnCMCLCTL[1:0] must be reset to 00_B by software. If FCN sleep mode is released by a change in the CAN bus state,

FCNnCMISCTL.FCNnCMISITSF5 is set to 1 regardless of FCNnCMIECTL.FCNnCMIEINTF[6:0]. After the FCN module is released from FCN sleep mode, it participates in the CAN bus communication again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until FCNnGMCLCTL.FCNnGMCLSSMO = 1, before accessing message buffers again.

When a request for transition to initialization mode is made while the FCN module is in FCN sleep mode, the request is ignored. The FCN module has to be released from sleep mode first by software before entering initialization mode.

Caution 1. Note that the release of FCN sleep mode by CAN bus event, and thus the wake-up interrupt may happen at any time, even right after requesting transition to sleep mode, if a CAN bus event occurs.

Caution 2. Always reset the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] bits to 00_B, when waking up from FCN sleep mode before accessing any other registers of the FCN module.

Caution 3. Always clear the interrupt flag FCNnCMISCTL.FCNnCMISITSF5, when waking up from FCN sleep mode.

20.9.2 FCN Stop Mode

FCN stop mode can be used to set the CAN controller to stand-by mode to reduce power consumption. The FCN module can enter FCN stop mode only from FCN sleep mode. Releasing FCN stop mode causes the FCN module to enter FCN sleep mode.

FCN stop mode can be released (entering FCN sleep mode) only by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 01_B. FCN stop mode cannot be released by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering FCN Stop Mode

A FCN stop mode transition request is issued by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 11_B.

A FCN stop mode transition request is only acknowledged while the FCN module is in FCN sleep mode. In all other modes, the request is ignored.

Caution To set the FCN module to FCN stop mode, the module must be in FCN sleep mode. To confirm that the module is in sleep mode, check that the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B, and then request transition to FCN stop mode. If a bus change occurs at the FCN reception pin CRXDn while this process is being performed, FCN sleep mode is automatically released. In this case, the FCN stop mode transition request is not acknowledged.

(2) FCN Stop Mode Status

The FCN module is in the following state after it enters FCN stop mode.

- The internal operating clock stops and the power consumption is minimized.
- To wake up the FCN module from the CPU, data can be set to FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- FCNnDNBMRX registers cannot be read.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing FCN Stop Mode

FCN stop mode can be released only by writing 01_B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0]. After releasing FCN stop mode, the FCN module enters FCN sleep mode.

When transition to initialization mode is requested while the FCN module is in FCN stop mode, the request is ignored. The CPU has to release stop mode and then release the FCN sleep mode before entering initialization mode. It is impossible to enter another operating mode directly from FCN stop mode not entering FCN sleep mode. Such request is ignored.

20.9.3 Example of Using Power Save Modes

In some application systems, it may be necessary to place the CPU in a power save mode to reduce power consumption. By using power save mode specific to the FCN module and power save mode specific to the CPU in combination, the CPU can be woken up from the power save status by the CAN bus.

The following shows an example for using power save modes.

- First, put the FCN module in FCN sleep mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B).
  After successfully confirming this state by reading the sleep mode status, put the CPU in power save mode. Disable interrupts to the CPU, while processing additional tasks after the FCN module has entered sleep mode to avoid the FCN wake-up interrupt from being acknowledged.
  If an edge transition from recessive to dominant is detected at the FCN reception pin CRXDn in this state, FCNnCMISCTL.FCNnCMISITSF5 in the FCN module is set to 1. When FCNnCNIECTL.FCNnCMIEINT5 is set to 1, a wake-up interrupt (INTCnWUP) is generated.
  The FCN module is automatically released from FCN sleep mode (FCNnCMCLMDPF[1:0] = 00_B) and returns to normal operating mode.
- The CPU, in response to INTnWUP, can release its own power save mode and return to normal operating mode.

  To further reduce the power consumption of the CPU, the internal clock - including that of the FCN module - may be stopped. In this case, the operating clock supplied to the FCN module stops after the FCN module has entered FCN sleep mode. Then, the CPU enters a power save mode in which the clock supplied to the CPU is stopped.
- If an edge transition from recessive to dominant is detected at the FCN reception pin CRXDn in this state, the FCN module can set FCNnCMISCTL.FCNnCMISITSF5 to 1 and generate a wake-up interrupt INTnWUP even if it is not supplied with the clock.
- However, other functions do not operate because clock supply to the FCN module is stopped, and the module remains in FCN sleep mode.
- The CPU, in response to INTnWUP,
  - releases its power save mode,
  - resumes supply of the internal clocks - including the clock to the FCN module - after the oscillation stabilization time has elapsed, and
  - starts instruction execution.
- When clock supply is resumed, the FCN module is immediately released from FCN sleep mode and returns to normal operating mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 00_B).

20.10 Interrupt Function

The FCN module provides seven different interrupt sources.

The occurrence of these interrupt sources is stored in the interrupt status register. Four separate interrupt request signals are generated from the seven 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 the interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 20-17 List of FCN Module Interrupt Sources

No.	Interrupt Status FCNnCMISCTL.	Interrupt Enable FCNnCMIECTL.*[1]	Interrupt Request Signal	Description of Interrupt Source
1	FCNnCMISITSF0	FCNnCMIESEIE0	INTnTRX	Message frame successfully transmitted from message buffer m
2	FCNnCMISITSF1	FCNnCMIESEIE1	INTnREC	Valid message frame reception in message buffer m
3	FCNnCMISITSF2	FCNnCMIESEIE2	INTnERR	FCN module error state interrupt • This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.
4	FCNnCMISITSF3	FCNnCMIESEIE3		FCN module protocol error interrupt • This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
5	FCNnCMISITSF4	FCNnCMIESEIE4		FCN module arbitration loss interrupt
6	FCNnCMISITSF5	FCNnCMIESEIE5	INTnWUP	FCN module wake-up interrupt from FCN sleep mode • This interrupt is generated when the FCN module is woken up from FCN sleep mode because a falling edge is detected at the FCN reception pin (CAN bus transition from recessive to dominant).
7	FCNnCMISITSF6	FCNnCMIESEIE6		FCN module transmission abort interrupt status • This interrupt is generated when transmission was successfully aborted (aborted message was not sent).

Note 1. The message buffer interrupt enable bit FCNnMmCTL.FCNnMmIENF of the corresponding message buffer must be set to 1 for that message buffer to participate in the interrupt generation process.

20.11 Diagnosis Functions and Special Operating Modes

The FCN 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.

20.11.1 Receive-only Mode

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 of the FCN module changes until “valid reception” is detected, so that the baud rates in the module match (“valid reception” means that 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). A valid reception event is indicated by setting FCNnCMCLCTL.FCNnCMCLVALF to 1.

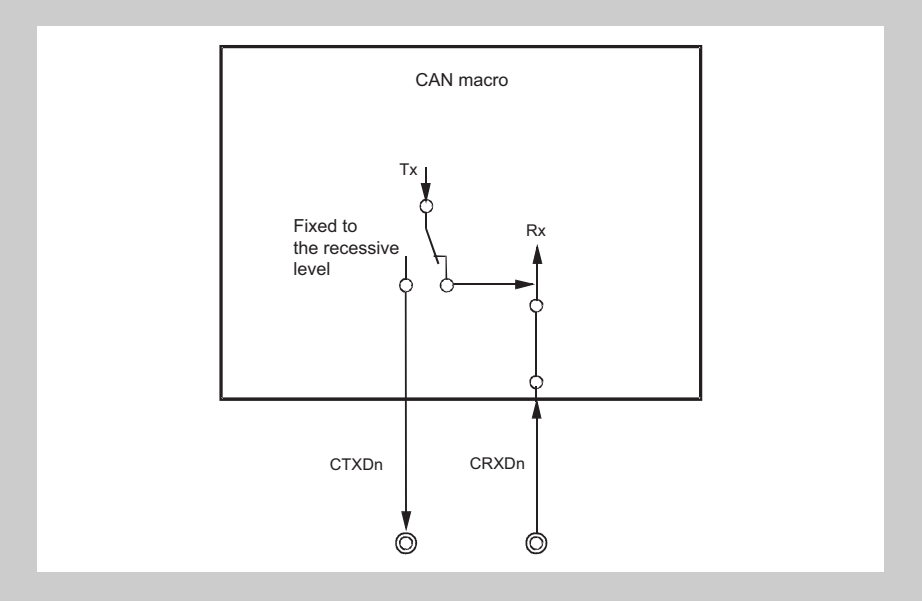


Figure 20-10 FCN Module Pin Connection in Receive-only Mode

In receive-only mode, no message frames can be transmitted from the FCN module to the CAN bus. Transmit requests issued to message buffers defined as transmit message buffers are held pending.

In receive-only mode, the FCN transmission pin CTXDn of the FCN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the FCN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since the FCN module cannot send data, the transmission error counter the FCNnCMERCNT.TEC7 to FCNnCMERCNT.TEC0 bits are never updated. Therefore, the FCN module in receive-only mode does not enter the bus-off state.

Furthermore, in receive-only mode, ACK is not returned to the CAN bus upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. No overload frame can 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 receive-only mode, no ACK is transmitted to the CAN bus. Due to the missing ACK, the transmitting node transmits 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). After the 17th message frame is transmitted, the transmitting node generates a passive error flag. The receiving node in receive-only mode detects a valid message frame for the first time at this point, and the FCNnCMCLCTL.FCNnCMCLVALF bit is set to 1 for the first time.

20.11.2 Single-shot Mode

In single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other operations of single shot-mode are identical to normal operating mode. Features of single-shot mode cannot be used in combination with normal operating mode with ABT.

Single-shot mode disables the re-transmission of an aborted message frame according to the setting of FCNnCMCLCTL.FCNnCMCLALBF. When FCNnCMCLALBF is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If FCNnCMCLALBF is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, FCNnMmCTL.FCNnMmTRQF 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 a message frame
- Error occurrence while sending a message frame

Arbitration loss and error occurrence events can be distinguished by checking FCNnCMISCTL.FCNnCMISITSF4 and FCNnCMISCTL.FCNnCMISITSF3 respectively, and the type of the error can be identified by reading FCNnCMLCSTR.FCN0CMLCSSLC[2:0].

Upon successful transmission of a message frame, the transmit completion interrupt bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1. When FCNnCMIECTL.FCNnCMIEINTF0 is set to 1 at this time, an interrupt request signal is output.

Single-shot mode can be used for emulating time-triggered communication methods (e.g. TTCAN level 1).

Caution FCNnCMCLCTL.FCNnCMCLALBF is only valid in single-shot mode. It does not affect the re-transmission upon occurrence of arbitration loss in other operating modes.

20.11.3 Self-test Mode

In self-test mode, message frame transmission and reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In self-test mode, the FCN module is completely disconnected from the CAN bus and transmission and reception are internally looped back. The FCN transmission pin CTXDn is fixed to the recessive level.

If the falling edge on the FCN reception pin CRXDn is detected after the FCN module has entered FCN sleep mode from self-test mode, however, the module is released from the FCN sleep mode in the same manner as the other operating modes. To keep the module in FCN sleep mode, use the FCN reception pin CRXDn as a port pin.

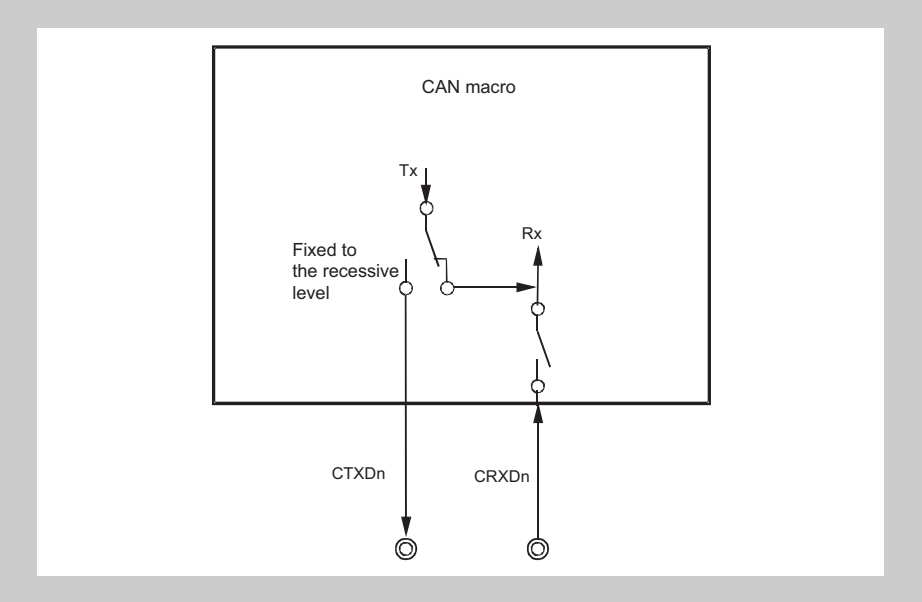


Figure 20-11 FCN Module Pin Connection in Self-test Mode

20.11.4 Receive/Transmit Operations in Each Operating Mode

The following table shows outline of the receive/transmit operations in each operating mode.

Table 20-18 Outline of Receive/Transmit Operations in Each Operating Mode

Operating Mode	Transmission of Data/Remote Frame	Transmission of ACK	Transmission of Error/ Overload Frame	Transmission Retry	Automatic Block Transmission (ABT)	FCNnCMCLVALF Setting	Store Data to Message Buffer
Initialization mode	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
Normal operating mode	Enabled	Enabled	Enabled	Enabled	Disabled	Enabled	Enabled
Normal operating mode with ABT	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
Receive-only mode	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
Single-shot mode	Enabled	Enabled	Enabled	Disabled *1	Disabled	Enabled	Enabled
Self-test mode	Enabled *2	Enabled *2	Enabled *2	Enabled*2	Disabled	Enabled *2	Enabled *2

Note 1. When the arbitration loss occurs, control of re-transmission is possible by FCNnCMCLCTL.FCNnCMCLALBF.

Note 2. Signals to be generated are not output to outside, but remain in the FCN module.

20.12 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 nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies depending on nodes).

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.

20.12.1 Time Stamp Function

The CAN controller supports the capturing of timer values triggered by a specific frame. This product uses the CAN controller and 1ch and 3ch of TAUJ0 in combination. (For details, see section 26.4.12, CAN Time Stamp Function.)

The 1ch and 3ch of TAUJ0 capture timer values according to the capturing trigger signal (TSOUT) that is output when a data frame from the CAN controller is received. 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. The TSOUT signal can be selected from the following two event sources and is specified by FCNnCMTSCTL.FCNnCMTSSELE.

- SOF event (start of frame)
  (FCNnCMTSCTL.FCNnCMTSSELE = 0)
- EOF event (last bit of end of frame)
  (FCNnCMTSCTL.FCNnCMTSSELE = 1)

The TSOUT signal is enabled by setting FCNnCMTSCTL.FCNnCMTSTSGE to 1.

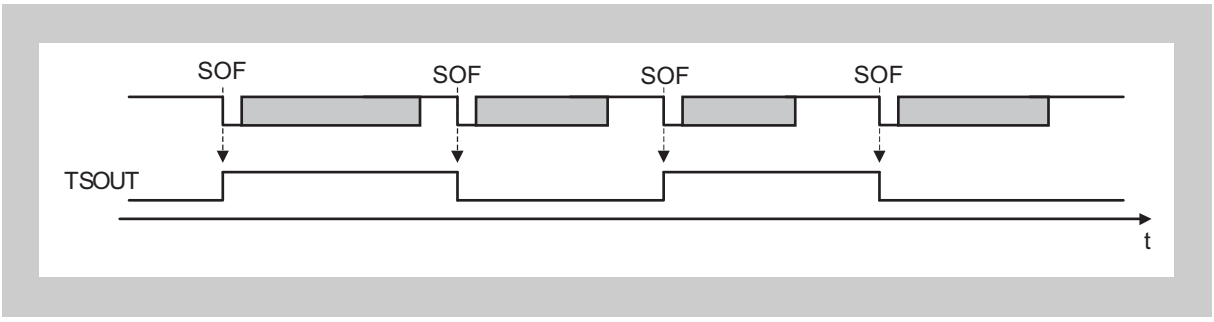


Figure 20-12 Timing Chart of Capture Signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in *figure20-12 Timing Chart of Capture Signal TSOUT*, the SOF is used as the trigger event source). To capture a timer value by using the TSOUT signal, 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 FCNnCMTSLOKE bit of the FCNnCMTSCTL register. When FCNnCMTSLOKE is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. When FCNnCMTSLOKE is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle stops as FCNnCMTSCTL.FCNnCMTSTSGE

is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by the TSOUT signal, 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 the FCNnCMTSLOKE bit stops toggle of the TSOUT signal by receiving a data frame in message buffer 0. Toggle of the TSOUT signal does not stop when a data frame is received in a message buffer other than message buffer 0.
A data frame cannot be received in message buffer 0 when the FCN module is in normal operating mode with ABT, because message buffer 0 must be set as a transmit message buffer.
In this operating mode, therefore, the function to stop toggle of the TSOUT signal by the FCNnCMTSLOKE bit cannot be used.

20.13 Baud Rate Settings

20.13.1 Baud Rate Setting Conditions

Make sure that the settings are within the range of limit values shown below for ensuring correct operation of the CAN controller.

- 5 TQ ≤ SPT (sampling point) ≤ 17 TQ

  SPT = TSEG1 + 1
- 8 TQ ≤ DBT (data bit time) ≤ 25 TQ

  DBT = TSEG1 + TSEG2 + 1 TQ = TSEG2 + SPT
- 1 TQ ≤ SJW (synchronization jump width) ≤ 4 TQ

  SJW ≤ DBT - SPT
- 4 ≤ TSEG1 ≤ 16
- 1 ≤ TSEG2 ≤ 8

Note 1. TQ = $1/f_{TQ}$ (f_{TQ}: CAN protocol layer reference system clock)

Note 2. The values TSEG1, TSEG2, and SJW are defined by the following register bits.

TSEG1 = FCNnCMBTCTL.FCNnCMBTS1LG[3:0] + 1

TSEG2 = FCNnCMBTCTL.FCNnCMBTS2LG[2:0] + 1

SJW = FCNnCMBTCTL.FCNnCMBTJWLG[1:0] + 1

table 20-19, Settable Bit-Rate Combinations, shows combinations of bit rates that meet the above conditions.

Table 20-19 Settable Bit-Rate Combinations (1/3)

Valid Bit-Rate Setting					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [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

Table 20-19 Settable Bit-Rate Combinations (2/3)

Valid Bit-Rate Setting					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8
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

Table 20-19 Settable Bit-Rate Combinations (3/3)

Valid Bit-Rate Setting					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
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*1	1	2	2	2	0011	001	71.4
7*1	1	4	1	1	0100	000	85.7
6*1	1	1	2	2	0010	001	66.7
6*1	1	3	1	1	0011	000	83.3
5*1	1	2	1	1	0010	000	80.0
4*1	1	1	1	1	0001	000	75.0

Note 1. Setting with a DBT value of 7 or less is valid only when the FCNnCMBRPRS register value is not 00_H.

Caution The values in table 20-19, Settable Bit-Rate Combinations, do not guarantee the operation of the network system. Thoroughly check the effects of set values on the network system, considering oscillation errors and delays of the CAN bus and CAN transceiver.

20.13.2 Typical Examples of Baud Rate Settings

table 20-20, Typical Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) , and table 20-21, Typical Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz), show typical examples of baud rate settings.

Table 20-20 Typical Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Baud Rate Setting Value (unit: kbps)	Division Ratio of FCNnCMBRPRS	FCNnCMBRPRS Setting Value	Valid Bit-Rate Setting (Unit: TQ)					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
			DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
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

Table 20-20 Typical Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Baud Rate Setting Value (unit: kbps)	Division Ratio of FCNnCMBRPRS	FCNnCMBRPRS Setting Value	Valid Bit-Rate Setting (Unit: TQ)					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
			DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
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 1. The values in table 20-21, Typical Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz), do not guarantee the operation of the network system. Thoroughly check the effects of set values on the network system, considering oscillation errors and delays of the CAN bus and CAN transceiver.

Caution 2. Baud rates higher than 500 kbit/s are not allowed when f_{CANMOD} is 8 MHz or less.

Table 20-21 Typical Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Baud Rate Setting Value (unit: kbps)	Division Ratio of FCNnCMBRPRS	FCNnCMBRPRS Setting Value	Valid Bit-Rate Setting (Unit: TQ)					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
			DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
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

Table 20-21 Typical Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

Baud Rate Setting Value (unit: kbps)	Division Ratio of FCNnCMBRPRS	FCNnCMBRPRS Setting Value	Valid Bit-Rate Setting (Unit: TQ)					FCNnCMBTCTL Setting Value		Sampling Point (Unit: %)
			DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	FCNnCMBTS1LG [3:0]	FCNnCMBTS2LG [2:0]	
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 20-21, Typical Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz), do not guarantee the operation of the network system. Thoroughly check the effects of set values on the network system, considering oscillation errors and delays of the CAN bus and CAN transceiver.

20.14 Operation of the CAN Controller

This section provides recommended processing procedures required to operate the FCN. Develop a program referring to recommended processing procedures in this section.

20.14.1 Initialization

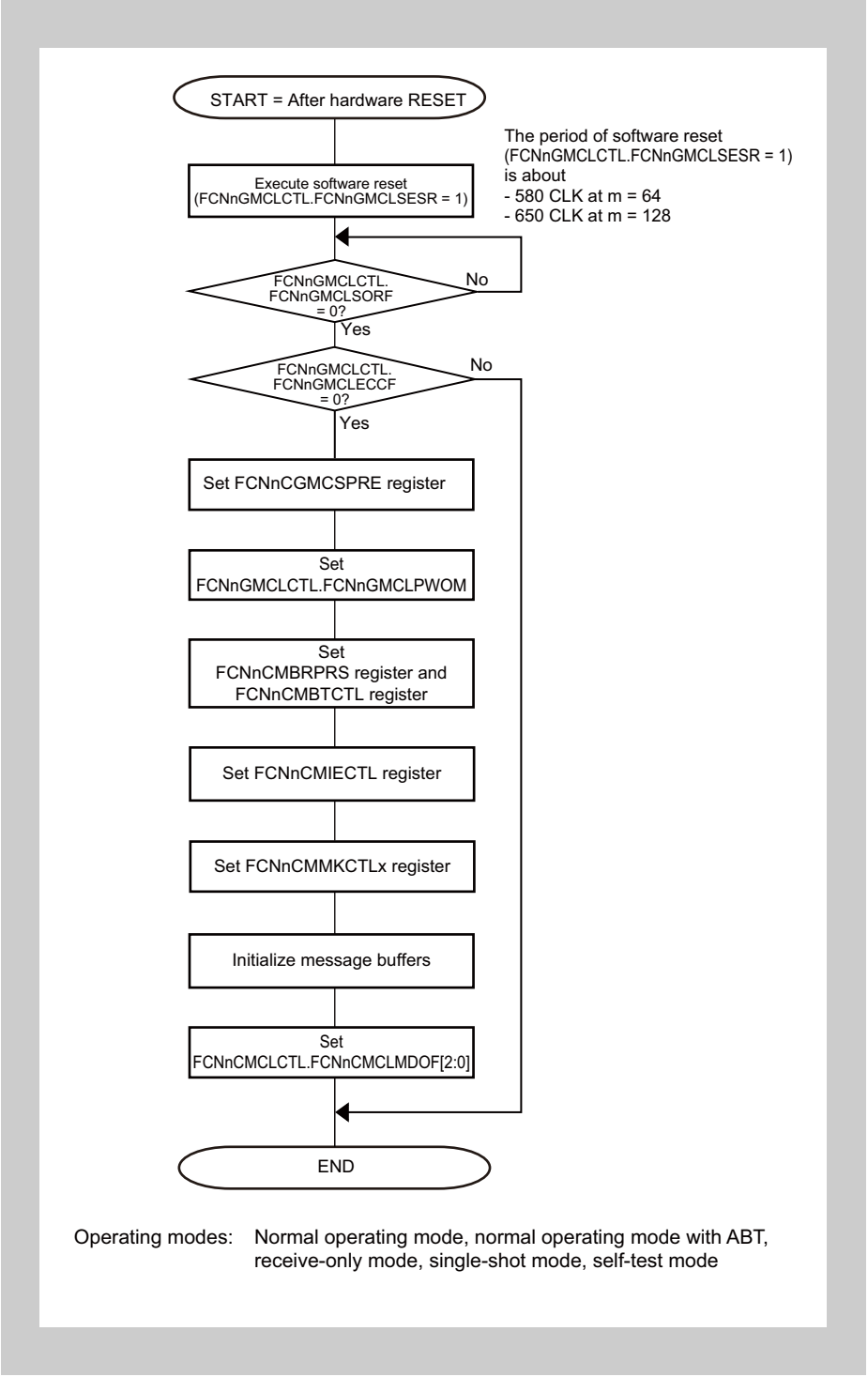


Figure 20-13 Initialization

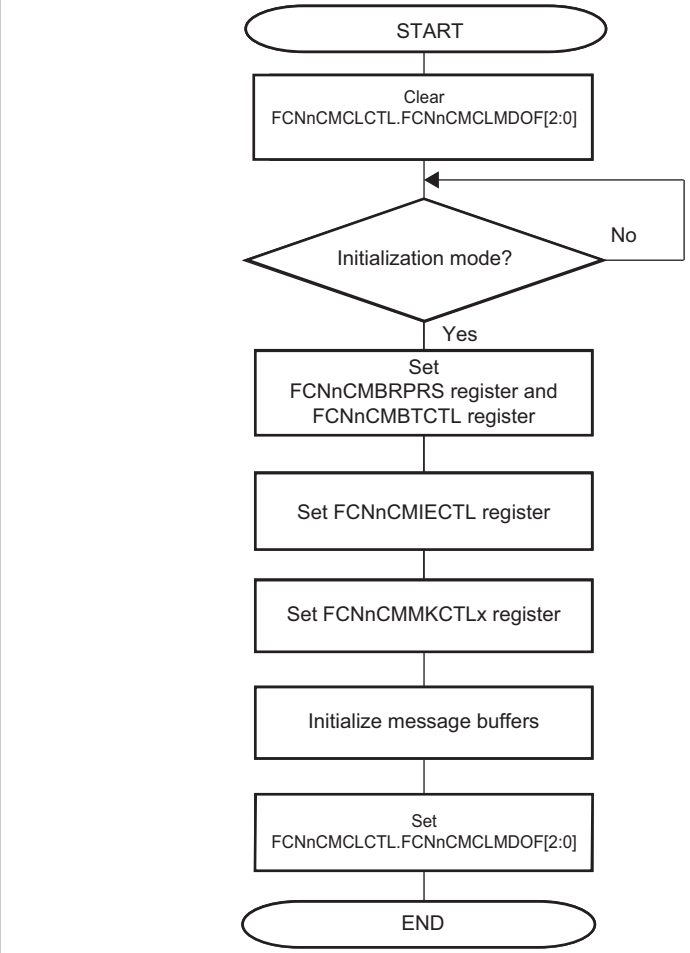


Figure 20-14 Re-initialization without Software Reset Function

Caution When clearing the error counter (setting CNnCMCLERCF) during re-initialization, clear it in either of the following states.

- In initialization mode after the FCN module has started (setting FCNnGMCLPWOM from FCNnGMCLPWOM = 0)
- In initialization mode after all transmit requests have been cleared according to the transmission abort processing in Figure 20-24 in operating mode (In normal operating mode with ABT, clear all transmit requests according to the transmission abort processing in Figure 20-25.)

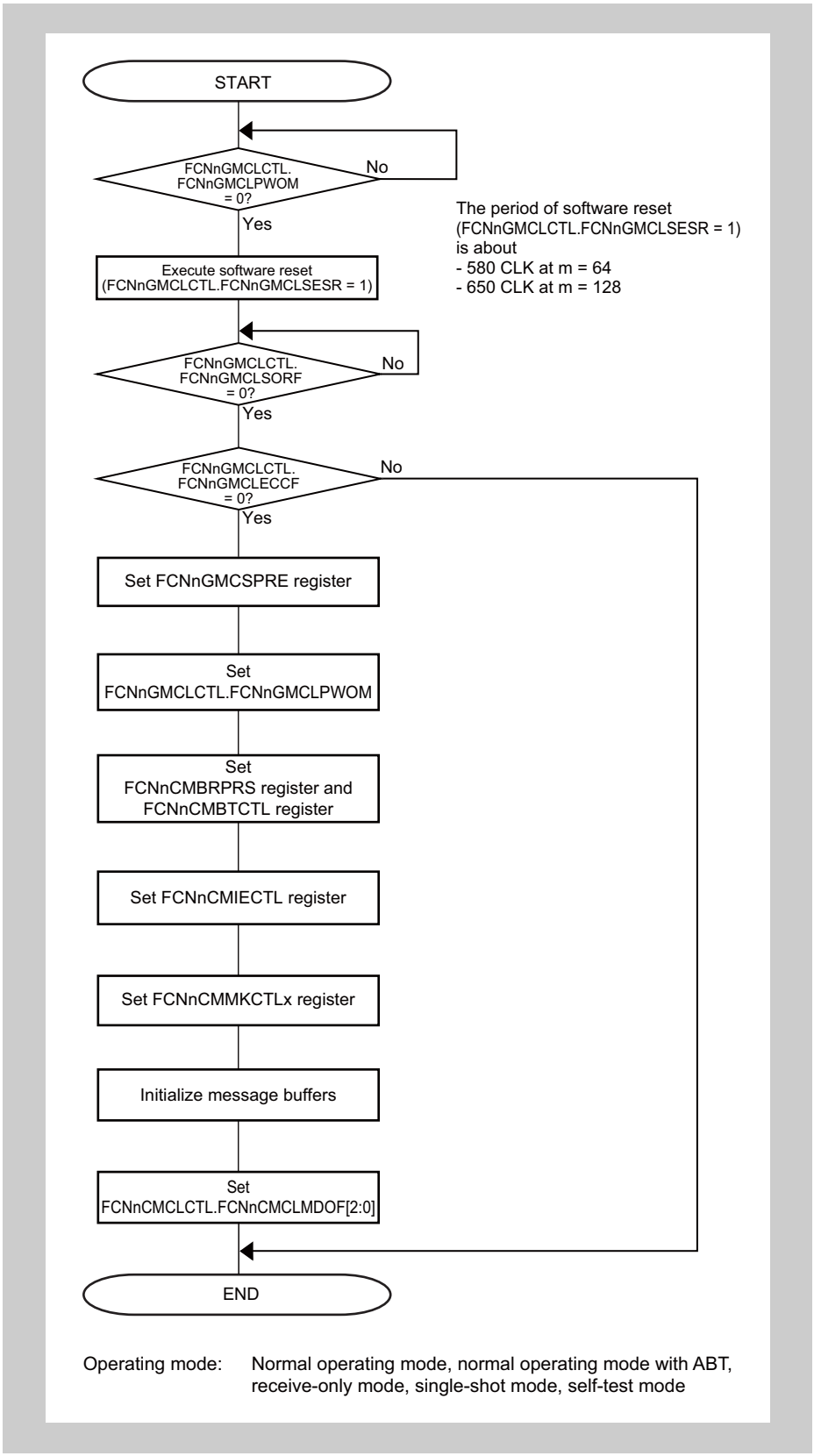


Figure 20-15 Re-initialization with Software Reset Function

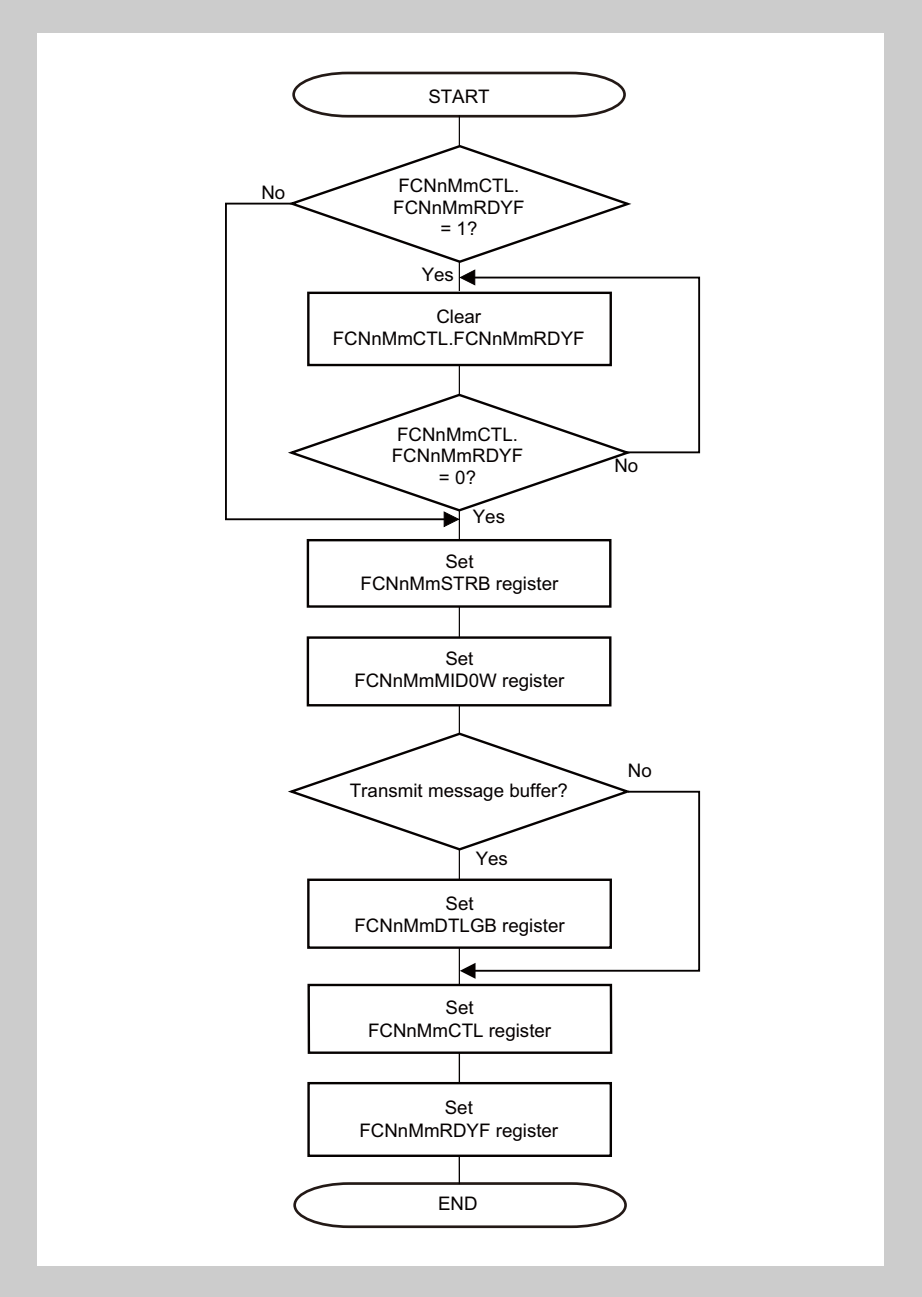


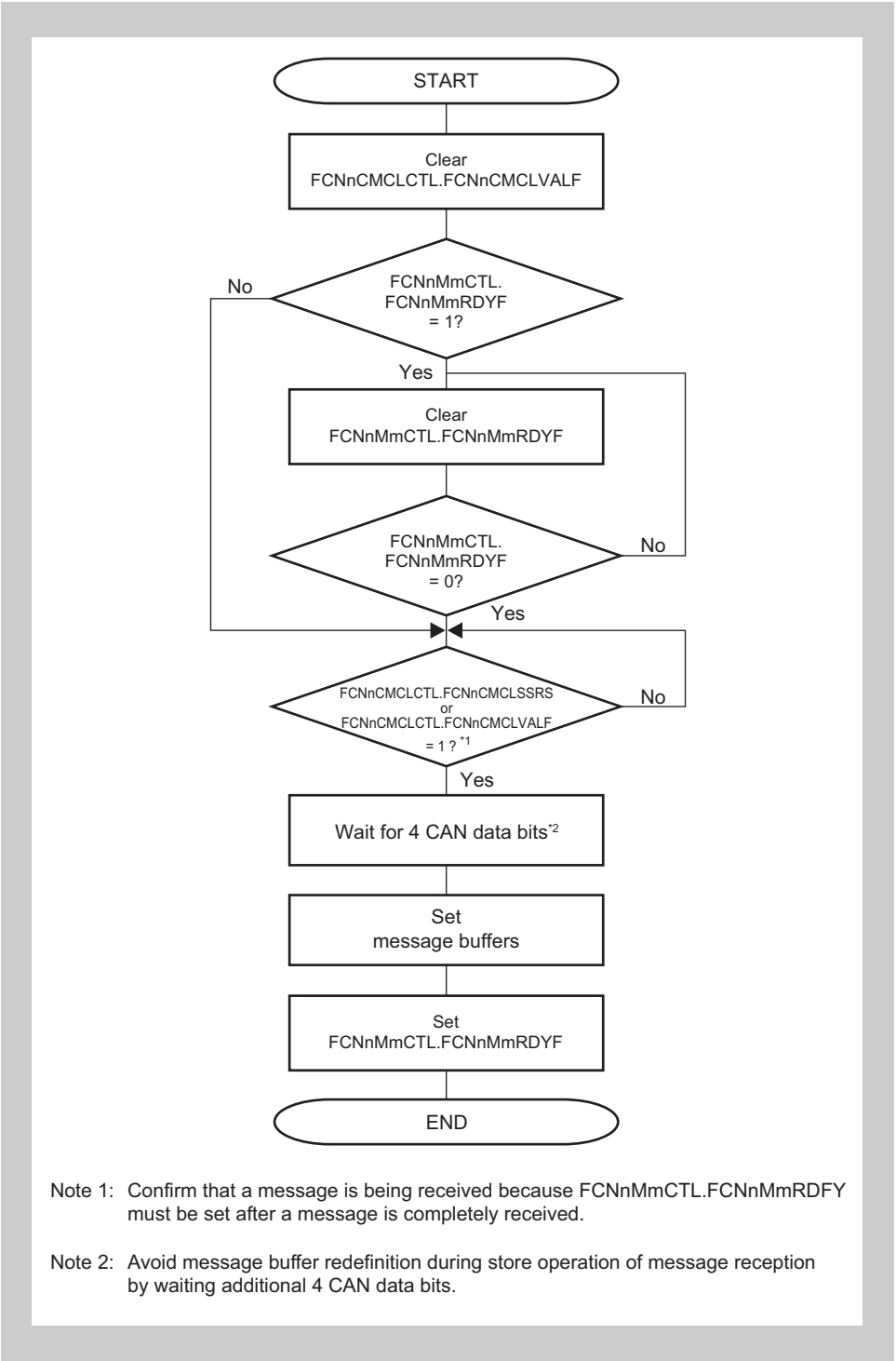
Figure 20-16 Message Buffer Initialization

Caution 1. Before a message buffer is initialized, FCNnMmCTL.FCNnMmRDYF must be cleared.

Caution 2. Make the following settings for message buffers not used by the application.

- Clear the FCNnMmRDYF, FCNnMmTRQF, and FCNnMmDTNF bits of the FCNnMmCTL register to 0.
- Clear FCNnMmSTRB.FCNnMmSSAM to 0.

figure20-17 Message Buffer Redefinition during Reception, shows the processing for a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B to 1000_B).



Note 1: Confirm that a message is being received because FCNnMmCTL.FCNnMmRDFY must be set after a message is completely received.

Note 2: Avoid message buffer redefinition during store operation of message reception by waiting additional 4 CAN data bits.

Figure 20-17 Message Buffer Redefinition during Reception

figure20-18 Message Buffer Redefinition during Transmission, shows the processing for a transmit message buffer during transmission (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B).

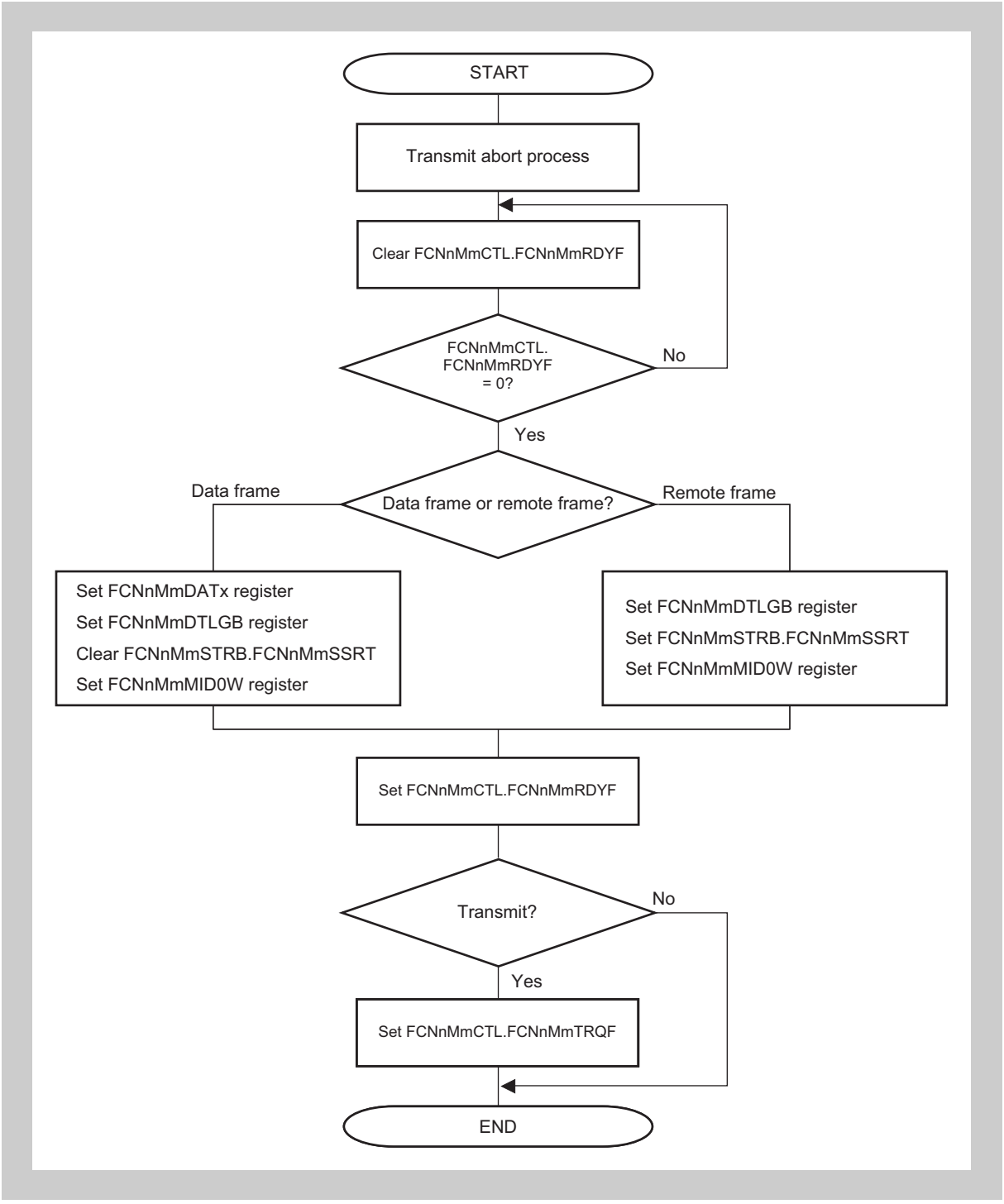


Figure 20-18 Message Buffer Redefinition during Transmission

20.14.2 Message Transmission

figure20-19 Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B).

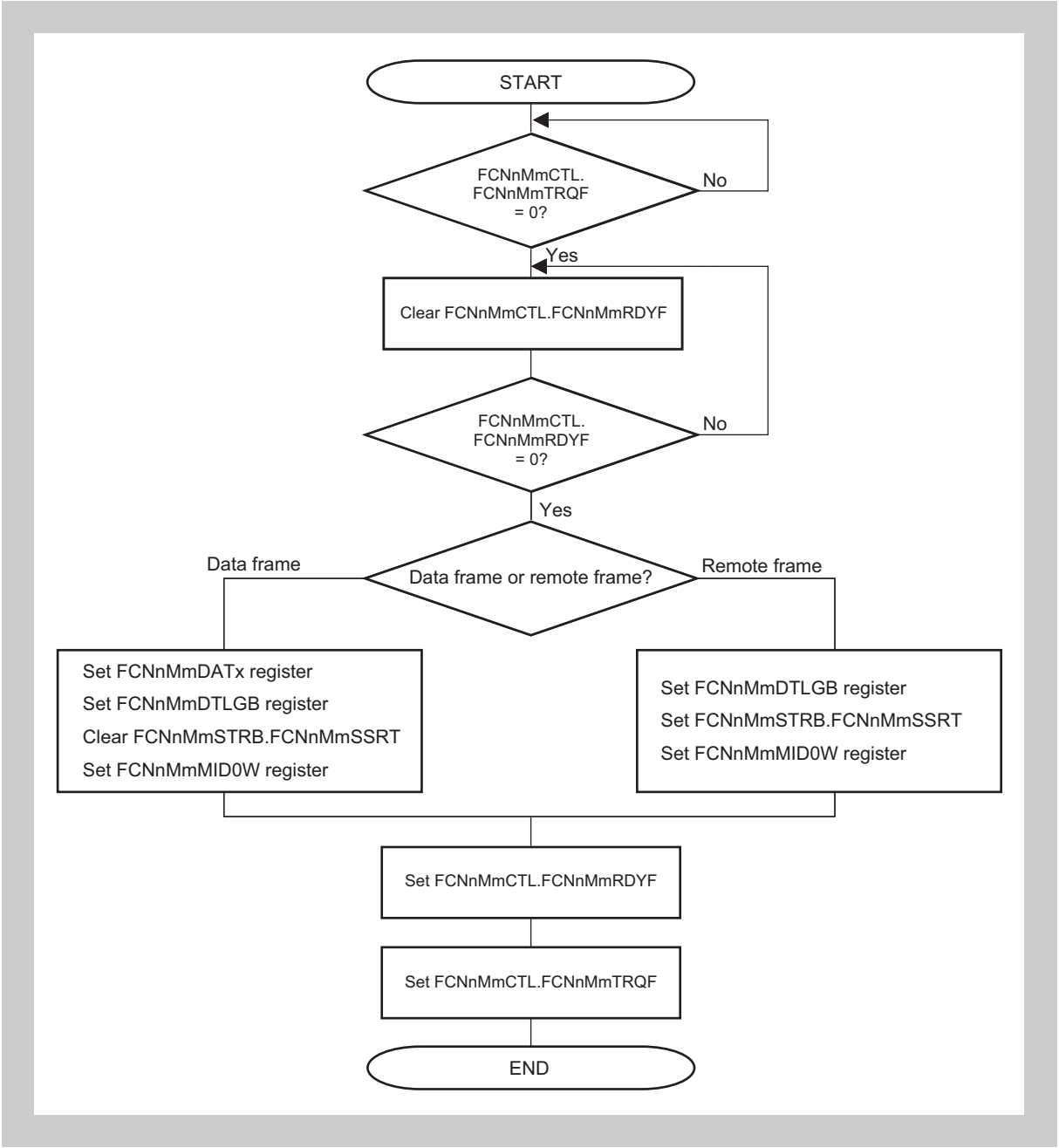


Figure 20-19 Message Transmit Processing

Caution 1. FCNnMmCTL.FCNnMmRDYF must be set before setting FCNnMmCTL.FCNnMmTRQF.

Caution 2. Do not set FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF at the same time.

figure20-20 ABT Message Transmit Processing, shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B).

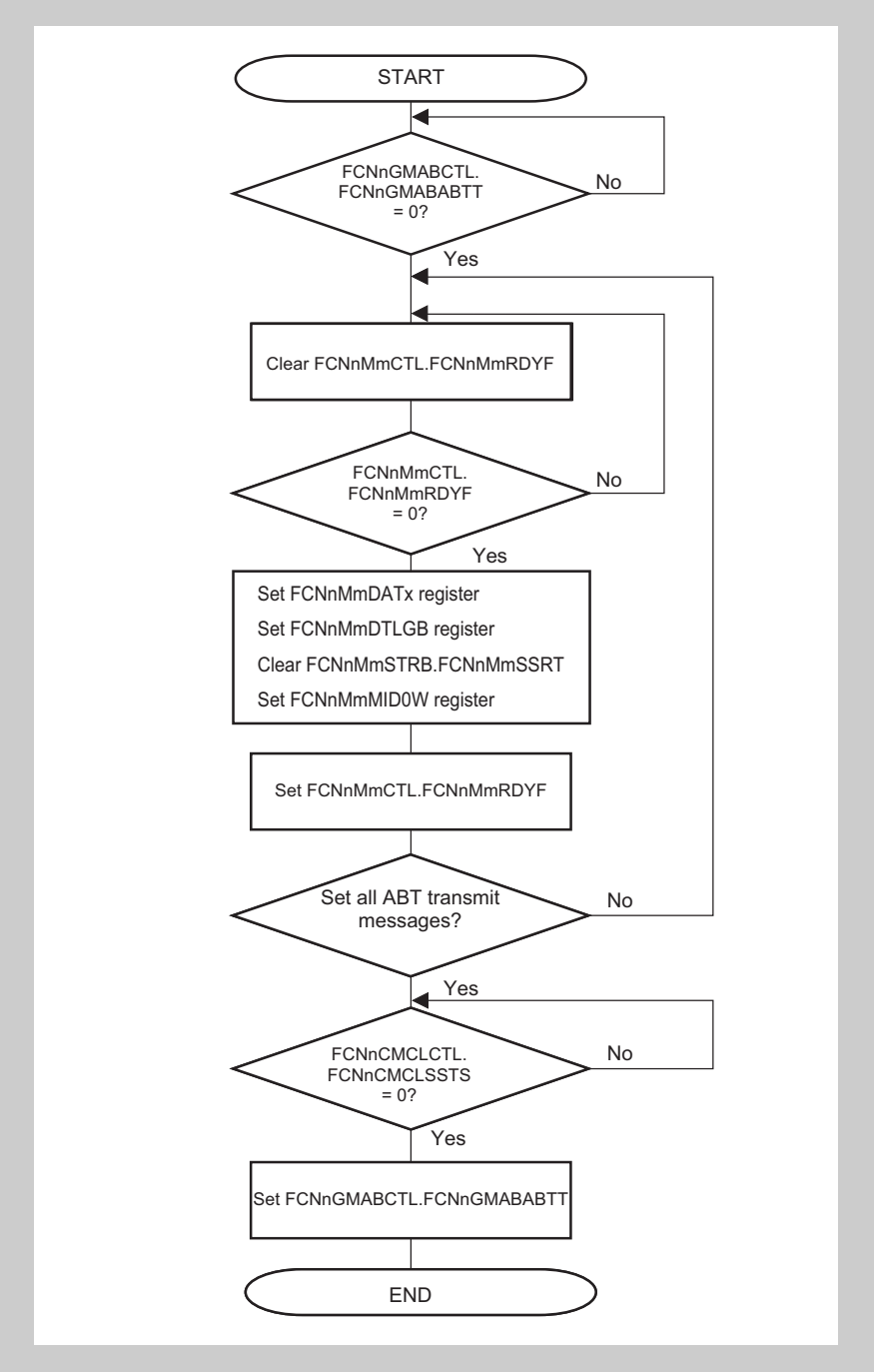


Figure 20-20 ABT Message Transmit Processing

Note This processing (normal operating mode with ABT) can only be applied to message buffers usable with ABT mode. For message buffers other than ABT message buffers, see *figure20-19 Message Transmit Processing*.

Caution FCNnCMCLCTL.FCNnCMCLSSTS must be cleared to 0 before setting FCNnGMABCTL.FCNnGMABSEAT to 1. FCNnCMCLCTL.FCNnCMCLSSTS must be checked first and then FCNnGMABCTL.FCNnGMABSEAT must be set to 1.

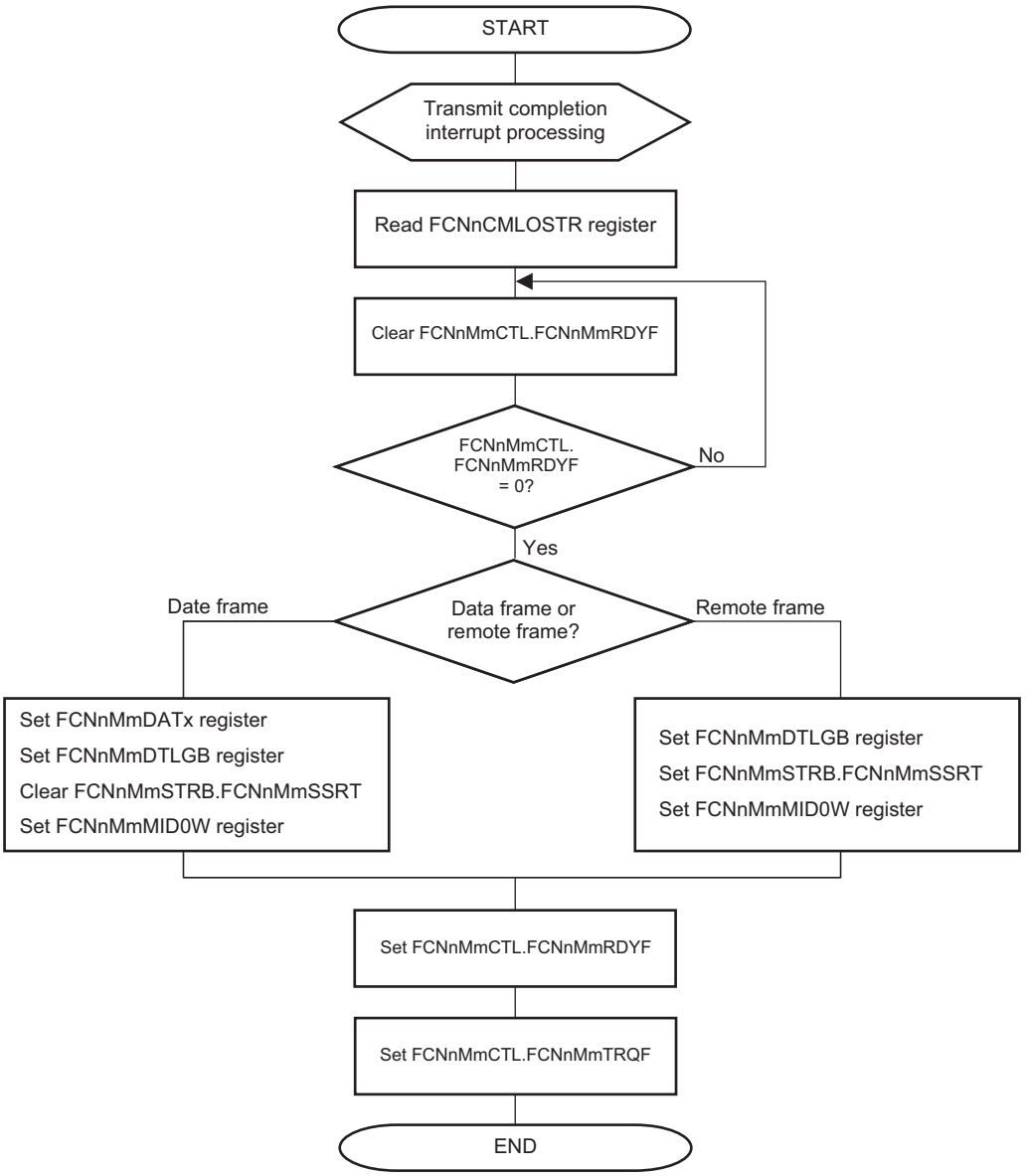


Figure 20-21 Transmission via Interrupt (Using FCNnCMLOSTR Register)

Caution 1. FCNnMmCTL.FCNnMmRDYF must be set before setting FCNnMmCTL.FCNnMmTRQF.

Caution 2. Do not set FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF at the same time.

Note Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as transmit history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing transmission interrupts.

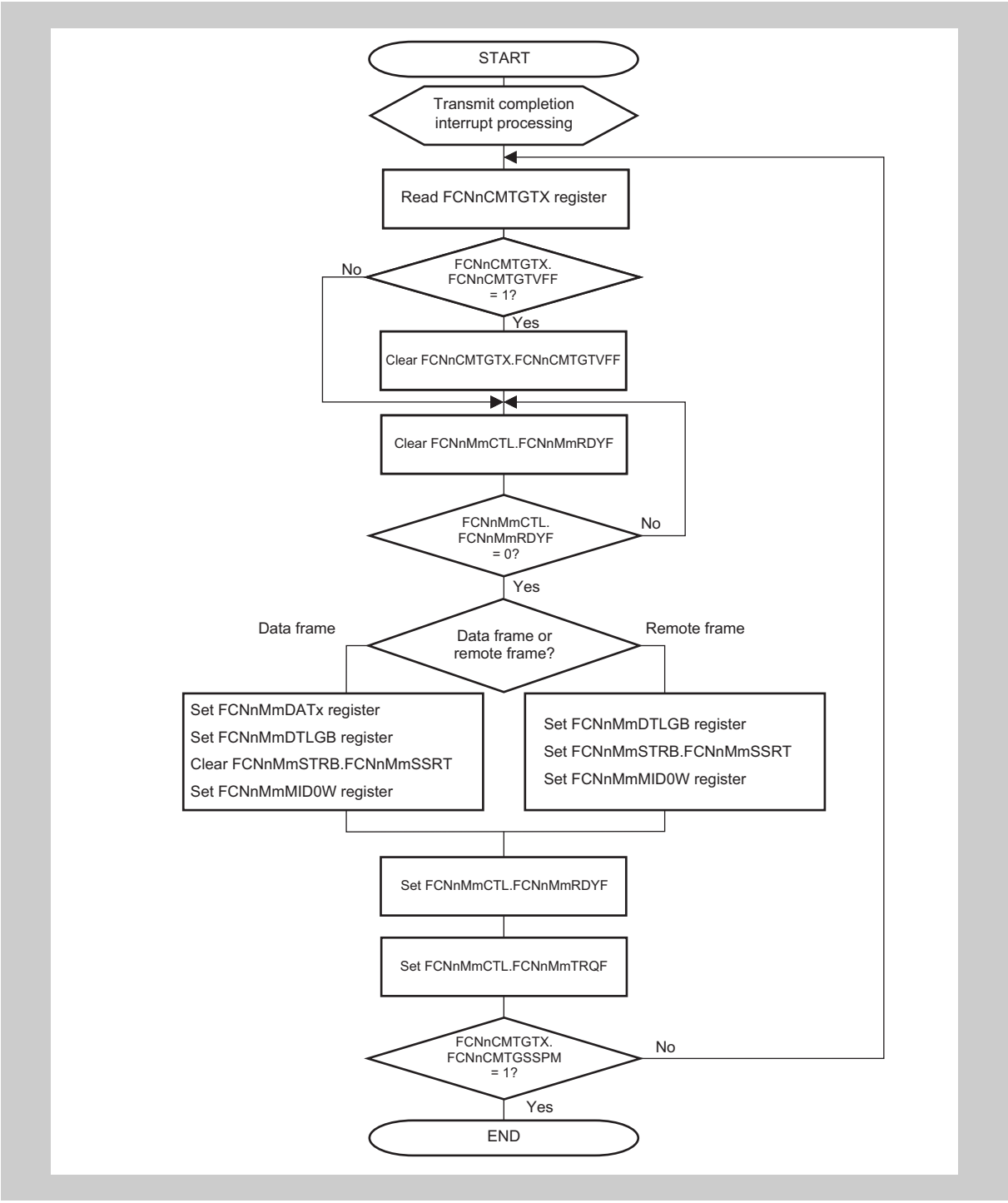


Figure 20-22 Transmission via Interrupt (Using FCNnCMTGTX Register)

Caution 1. FCNnMmCTL.FCNnMmRDYF must be set before setting FCNnMmCTL.FCNnMmTRQF.

Caution 2. Do not set FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF at the same time.

Note 1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as transmit history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing transmission interrupts.

Note 2. Once FCNnCMTGTX.FCNnCMTGTVFF is set, the transmit history list is inconsistent. Consider to examine all configured transmit buffers to check completed transmissions.

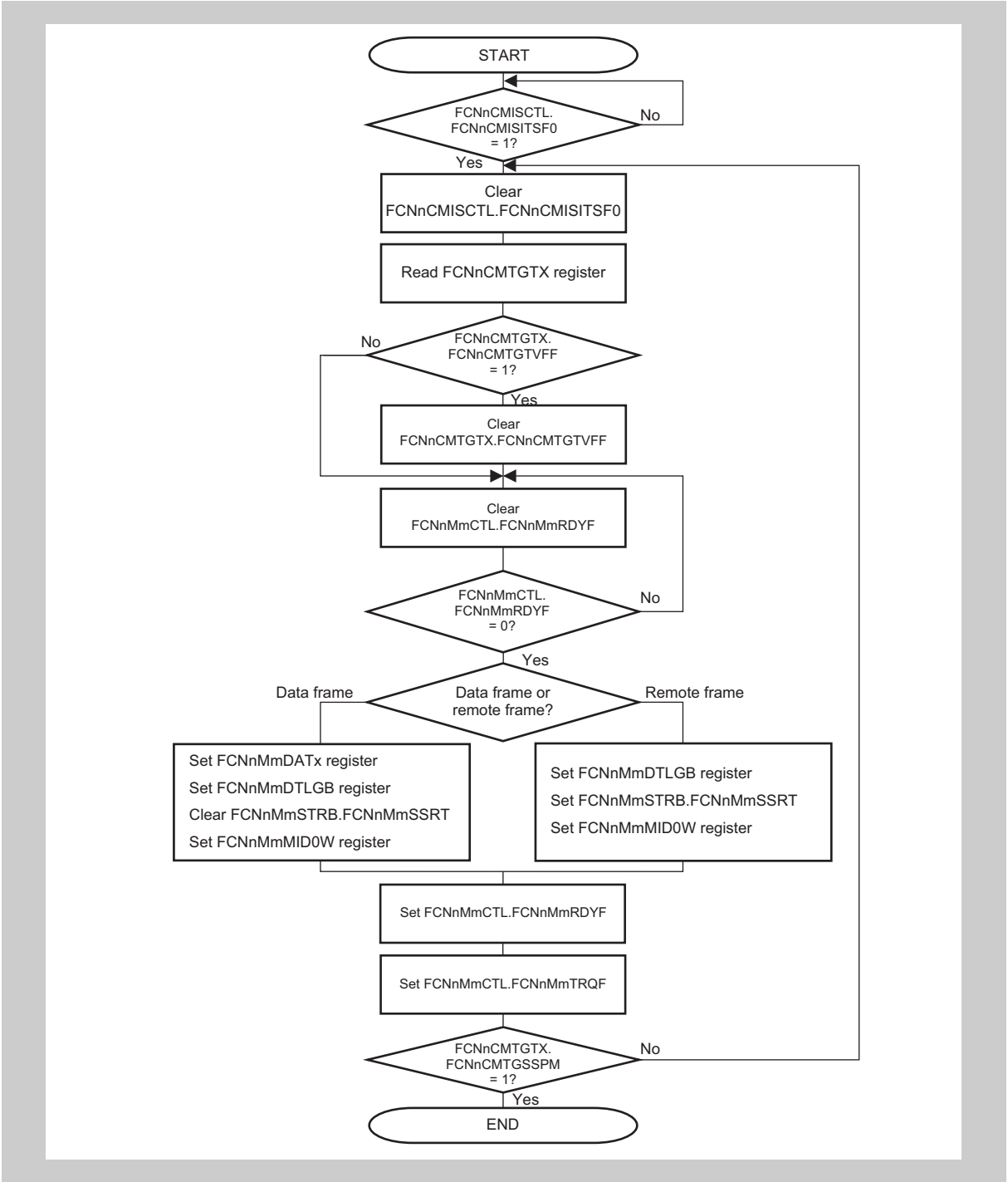


Figure 20-23 Transmission via Software Polling

Caution 1. FCNnMmCTL.FCNnMmRDYF must be set before setting FCNnMmCTL.FCNnMmTRQF.

Caution 2. Do not set FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF at the same time.

Note 1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as transmit history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing transmission interrupts.

Note 2. Once FCNnCMTGTX.FCNnCMTGTVFF is set, the transmit history list is inconsistent. Consider to examine all configured transmit buffers to check completed transmissions.

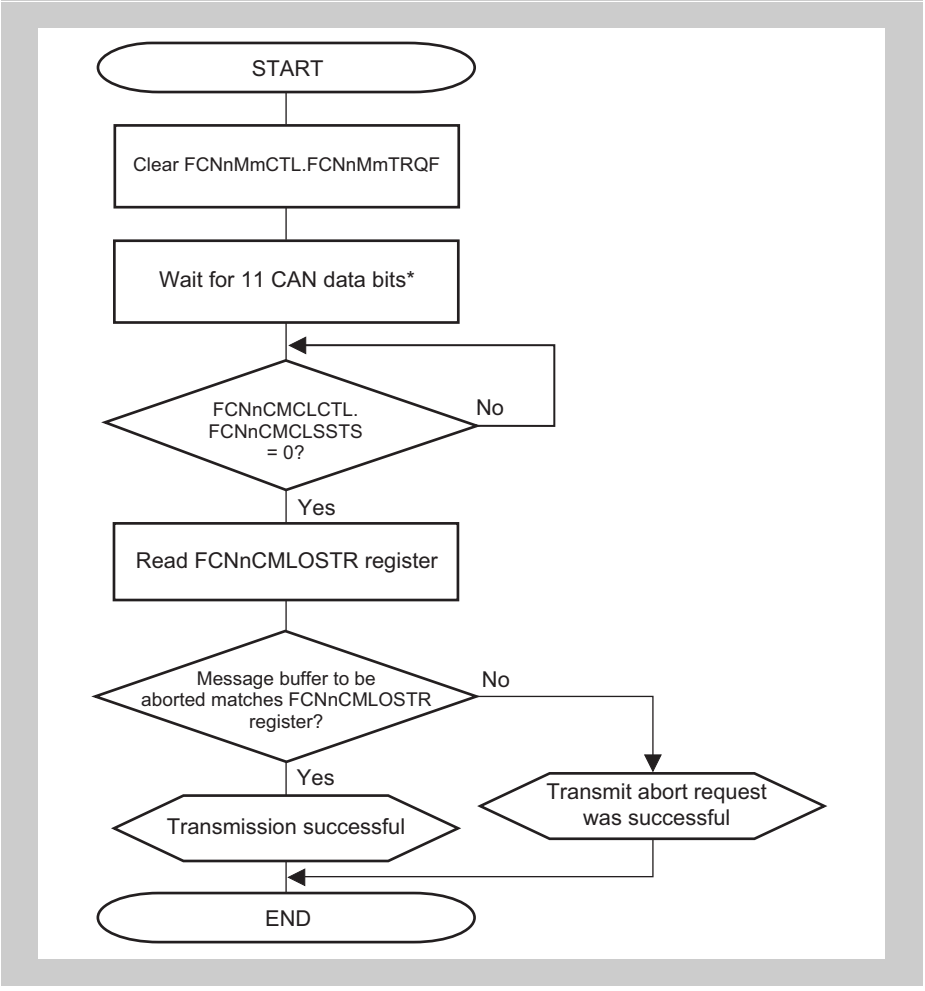


Figure 20-24 Transmission Abort Processing (Except Normal Operating Mode with ABT)

Note * A transmission request to the protocol layer may have been acknowledged during a period of 11 bits (3-bit next inter-frame space + 8-bit transmission abort processing). Therefore, transmission may not be aborted and started even if FCNnMmCTL.FCNnMmTRQF is cleared.

Caution 1. To request for aborting transmission, clear FCNnMmCTL.FCNnMmTRQF, but not FCNnMmCTL.FCNnMmRDYF.

Caution 2. Before making a sleep mode transition request, confirm that no transmission request that uses this processing is remaining.

Caution 3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.

Caution 4. Do not execute any new transmission request including transmission in other message buffers while transmission abort processing is in progress.

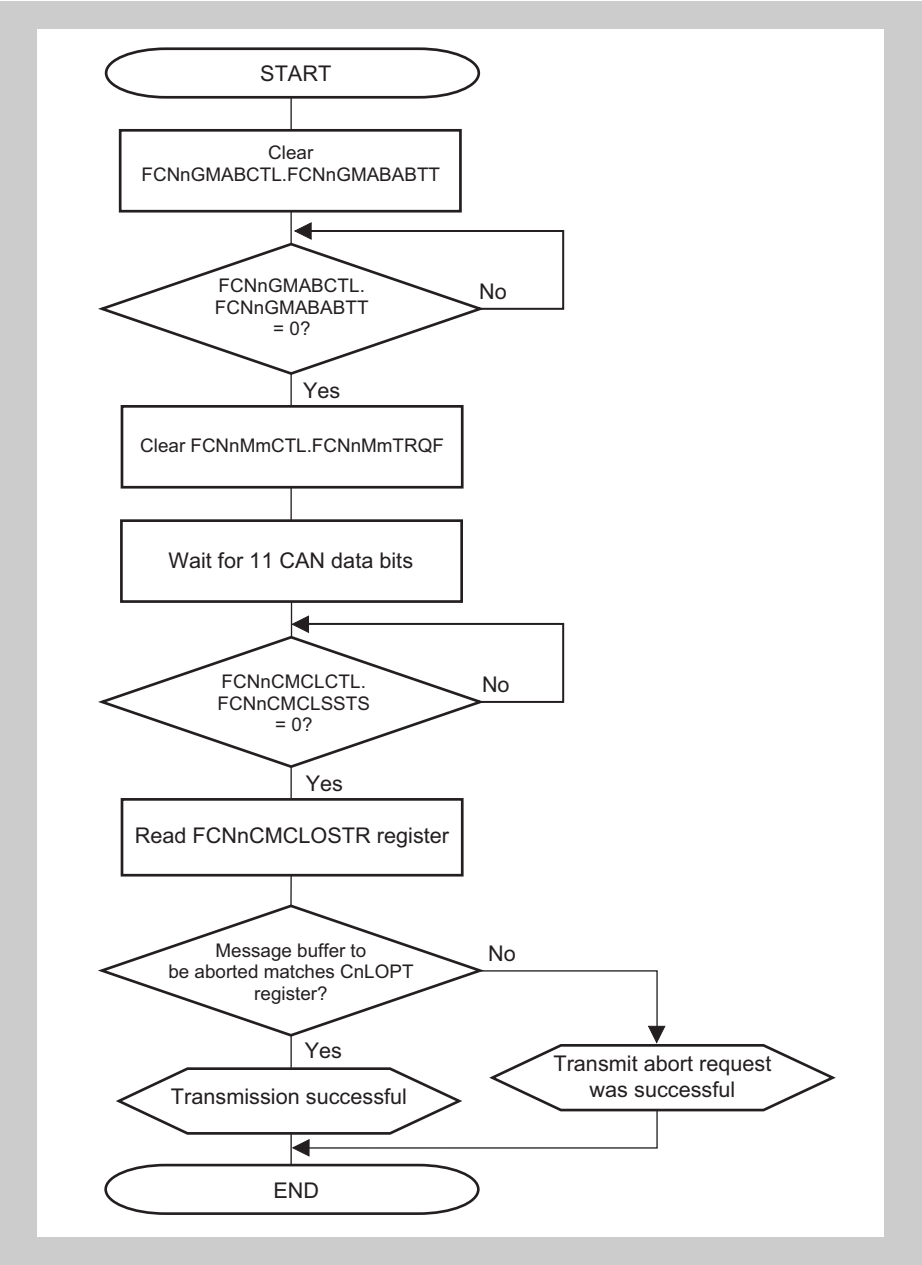


Figure 20-25 Transmission Abort Processing (Normal Operating Mode with ABT) - Retransmission Option for Aborted Message

Caution 1. To request for aborting transmission, clear FCNnMmCTL.FCNnMmTRQF, but not FCNnMmCTL.FCNnMmRDYF.

Caution 2. Before making a sleep mode transition request, confirm that no transmission request that uses this processing is remaining.

Caution 3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.

Caution 4. Do not execute any new transmission request including transmission in other message buffers while transmission abort processing is in progress.

figure20-26 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT), shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

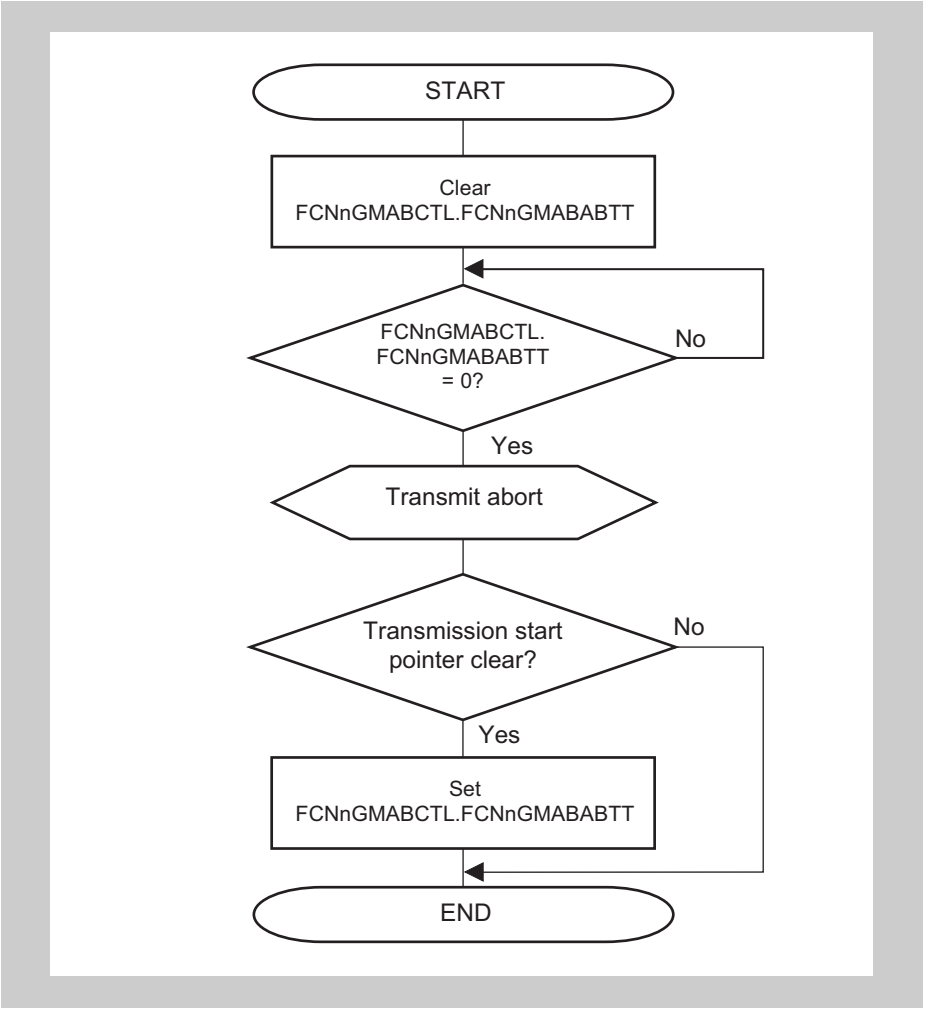


Figure 20-26 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT)

Caution 1. Do not set any transmission request while ABT transmission abort processing is in progress.

Caution 2. Make a FCN sleep mode/FCN stop mode transition request after FCNnGMABCTL.FCNnGMABABTT is cleared (after ABT mode is aborted) following the procedure shown in *figure20-26 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT)*, or *figure20-27 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT) with Transmission Abort Interrupt Flag* with Transmission Abort Interrupt Flag. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *figure20-24 Transmission Abort Processing (Except Normal Operating Mode with ABT)*.

figure20-27 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT) with Transmission Abort Interrupt Flag with Transmission Abort Interrupt Flag, shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

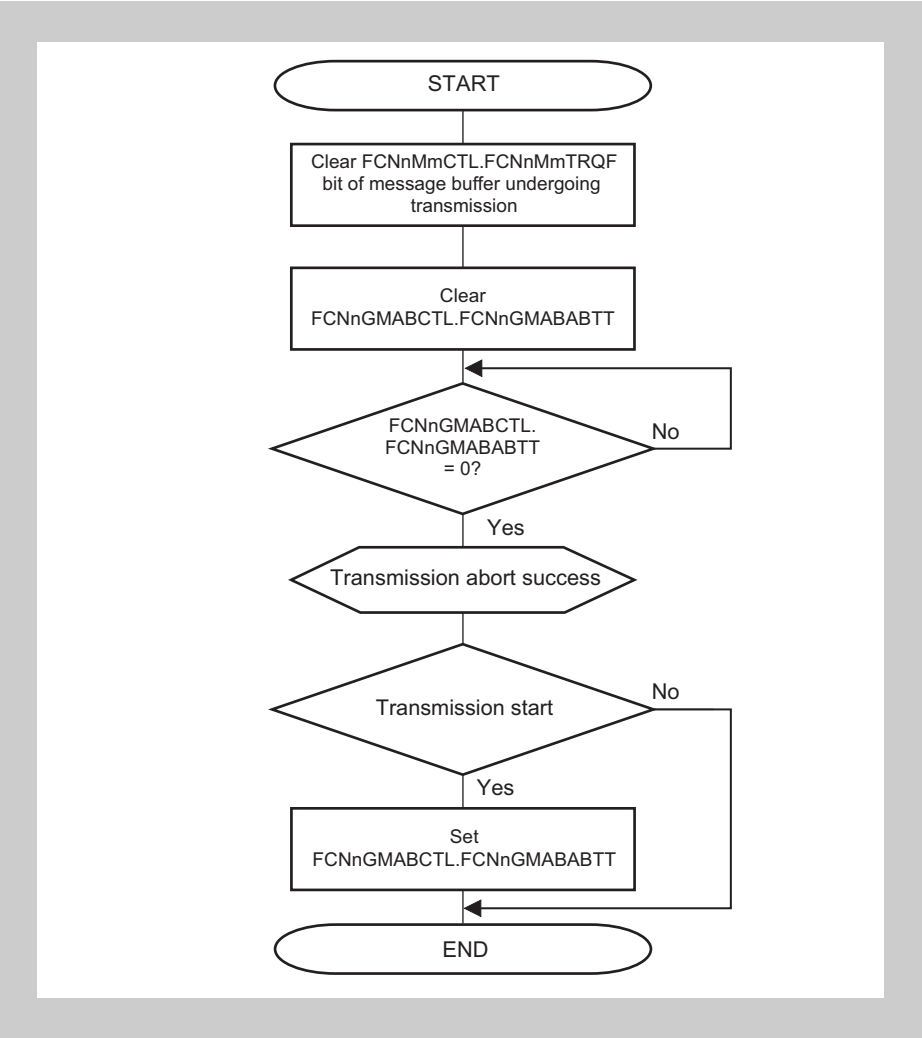


Figure 20-27 ABT Transmission Request Abort Processing (Normal Operating Mode with ABT) with Transmission Abort Interrupt Flag

Caution 1. Do not set any transmission request while ABT transmission abort processing is in progress.

Caution 2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG is cleared (after ABT mode is aborted) following the above processing. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in the flowchart for aborting a transmission request in an operating mode other than ABT mode.

Note There is the case that all ABT is transmitted completely even if transmission abort interrupt is generated. Then, it is possible to know which message is finished transmission. *figure20-28 Transmission Abort Processing with Transmission Abort Interrupt and Transmission Completion Flag*, shows the processing using the transmission abort function (transmission abort interrupt).

figure20-28 Transmission Abort Processing with Transmission Abort Interrupt and Transmission Completion Flag and Transmission Completion Flag, shows the processing using the transmission abort function (transmission abort interrupt).

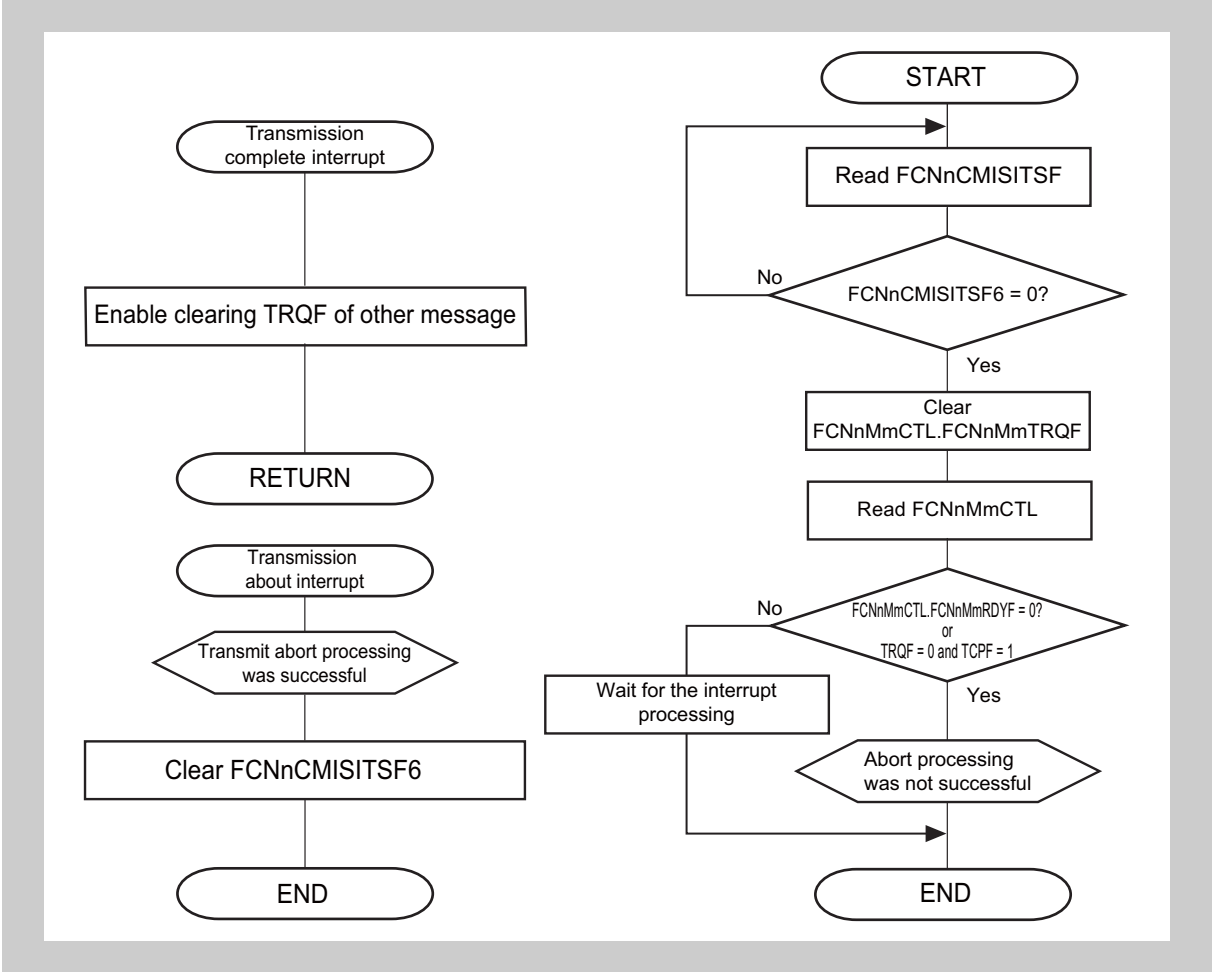


Figure 20-28 Transmission Abort Processing with Transmission Abort Interrupt and Transmission Completion Flag

Note Decision of FCNnMmRDYF = 0 is based on that FCNnMmRDYF is cleared during transmission completion processing triggered by an interrupt.

Caution 1. Perform transmission abort processing by clearing FCNnMmTRQF, not by clearing FCNnMmRDYF.

Caution 2. Before making a sleep mode transition request, confirm that there is no transmission request left using this flowchart.

Caution 3. Avoid the target message from being updated (setting FCNnMmRDYF and FCNnMmTRQF) while transmission abort processing is in progress due to the transmission completion processing.

Caution 4. Do not clear FCNnMmTRQF of other message buffers while transmission abort processing is in progress.

Caution 5. When setting an ID with lower priority than the previous ID after transmission abort processing, clear FCNnMmTRQF and then make a transmission request with a delay of at least one-frame time period.

Caution 6. While a transmission abort interrupt is in use, do not enable the wake-up interrupt by RXONLY-CH (only DIAG version).

Caution 7. Be sure to read FCNnMmTRQF and FCNnMmTCPF at the same time.

Note Since the transmission request to the protocol layer has already been acknowledged, transmission may be started during a period of 11 bits (3-bit next inter-frame space + 8-bit transmission abort processing) even if the TRQ bit is cleared.

figure20-29 Transmission Abort Processing with Transmission Completion Flag, shows the processing using the transmission abort function (transmission completion flag FCNnMmTCPF).

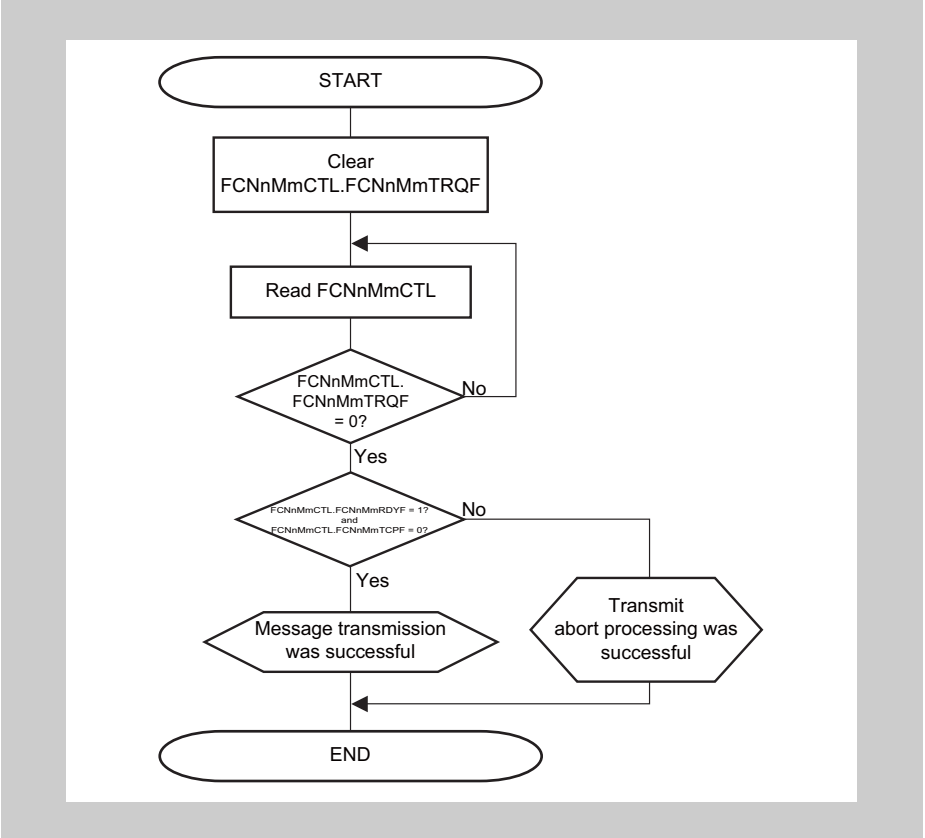


Figure 20-29 Transmission Abort Processing with Transmission Completion Flag

Note Decision of FCNnMmRDYF=1 is based on that FCNnMmRDYF is cleared during transmission completion processing triggered by an interrupt.

Caution 1. Perform transmission abort processing by clearing FCNnMmTRQF, not by clearing FCNnMmRDYF.

Caution 2. Before making a sleep mode transition request, confirm that there is no transmission request left using this flowchart.

Caution 3. Avoid the target message from being updated (setting FCNnMmRDYF and FCNnMmTRQF) while transmission abort processing is in progress due to the transmission completion processing.

Caution 4. When setting an ID with lower priority than the previous ID after transmission abort processing, clear FCNnMmTRQF and then make a transmission request with a delay of at least one-frame time period.

Caution 5. Be sure to read FCNnMmTRQF and FCNnMmTCPF at the same time.

Note Since the transmission request to the protocol layer has already been acknowledged, transmission may be started during a period of 11 bits (3-bit next inter-frame space + 8-bit transmission abort processing) even if the TRQ bit is cleared.

20.14.3 Message Reception

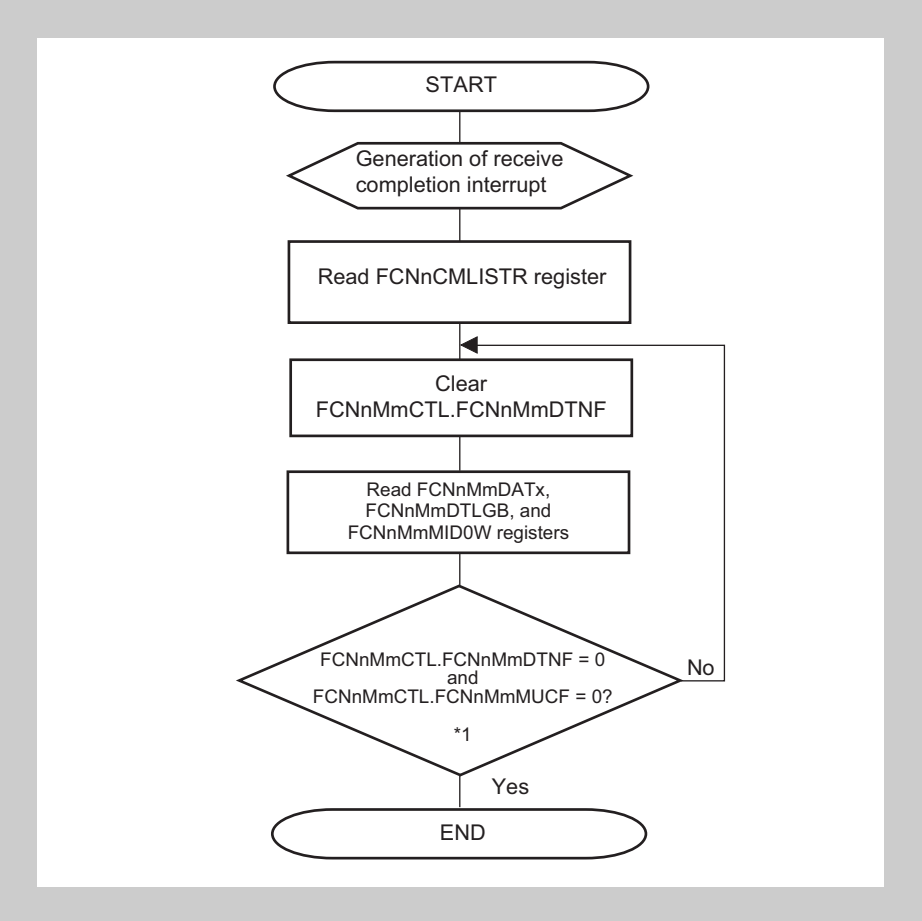


Figure 20-30 Reception via Interrupt (Using FCNnCMLISTR Register)

Note 1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits in one read access.

Note 2. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as receive history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing reception interrupts.

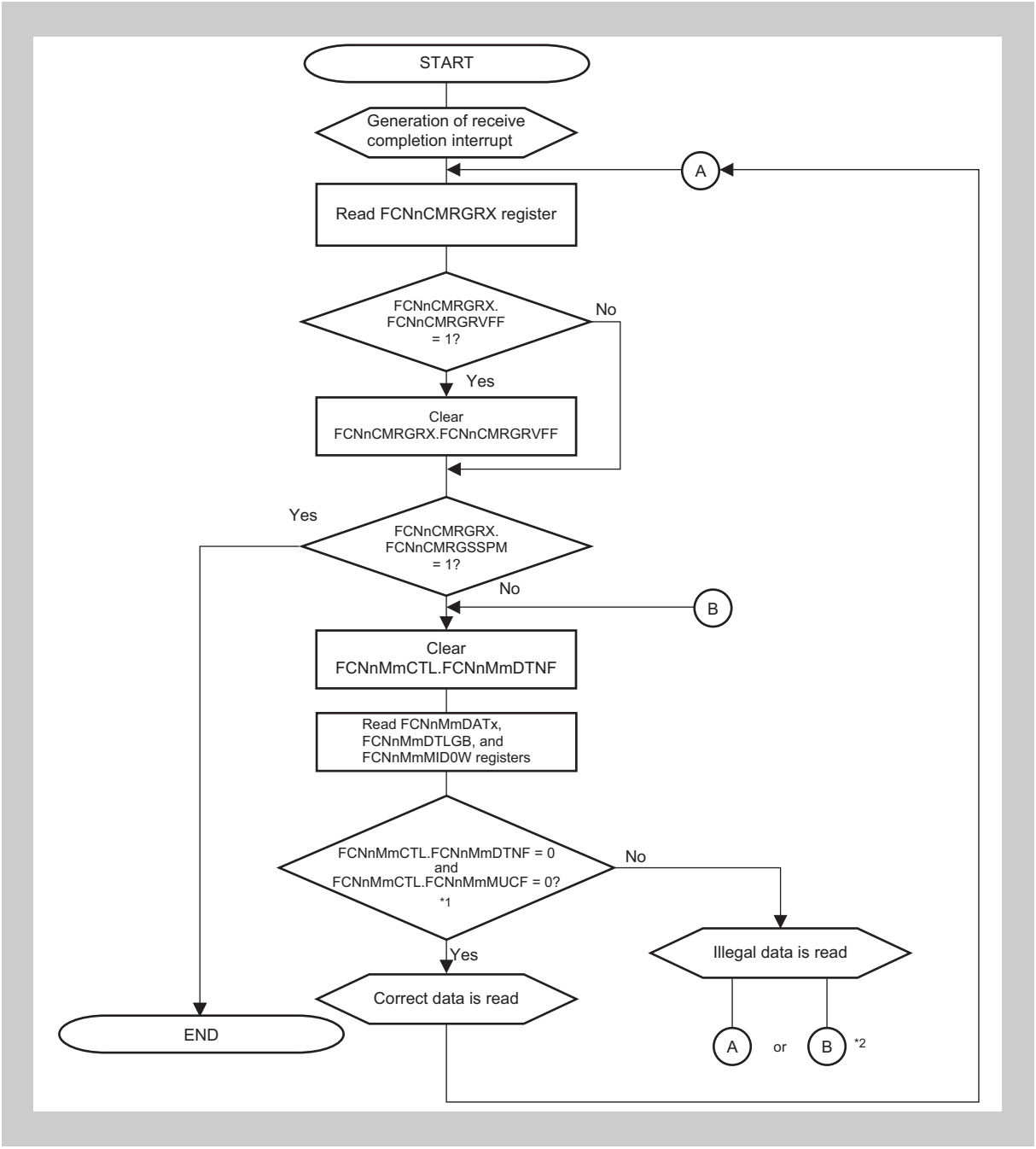


Figure 20-31 Reception via Interrupt (Using FCNnCMRGRX Register)

Note 1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits in one read access.

Note 2. Depending of the processing target of the application, two ways are possible:

- Way A: The message is not processed within this pass, but with the next pass, depending on the latest timing at which the message is processed with the next reception interrupt. Other messages are processed earlier.
- Way B: The message is processed within this pass, and the loop enters the waiting state with this message. Other messages are processed later.

Note 3. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as receive history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing reception interrupts.

Note 4. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list is inconsistent. Consider to examine all configured receive buffers to check reception.

Note 5. *figure20-32 Reception via Interrupt (Using FCNnCMRGRX Register), Alternative Way*, Alternative Way, is available as an alternative of *figure20-31 Reception via Interrupt (Using FCNnCMRGRX Register)*.

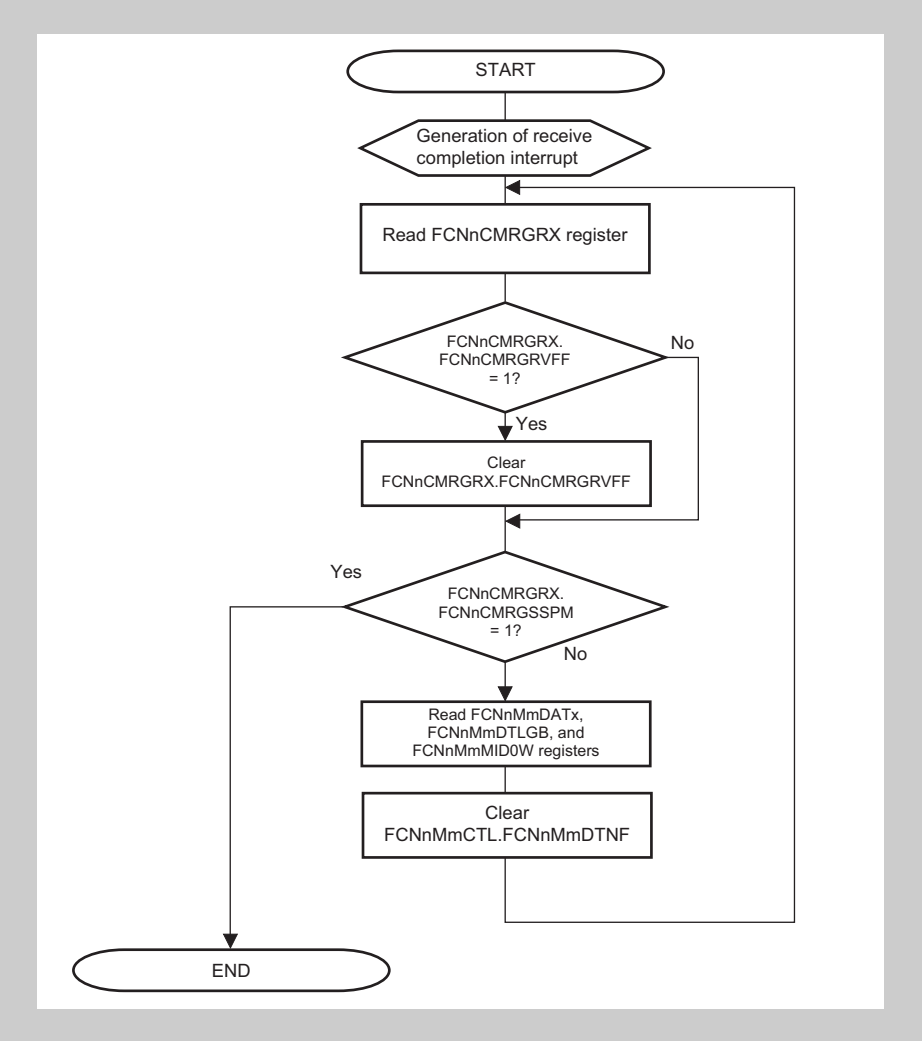


Figure 20-32 Reception via Interrupt (Using FCNnCMRGRX Register), Alternative Way

Note 1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as receive history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.
It is recommended that all sleep mode requests be cancelled before processing reception interrupts.

Note 2. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list is inconsistent. Consider to examine all configured receive buffers to check reception.

Note 3. This flow will not provide most recently received data for the application. However, interrupt load is reduced because of less processing amount.

Note 4. The overwrite function (FCNnMmSTRB.FCNnMmSSOW = 1) must not be used in this flow. Using this function may lose data consistency.

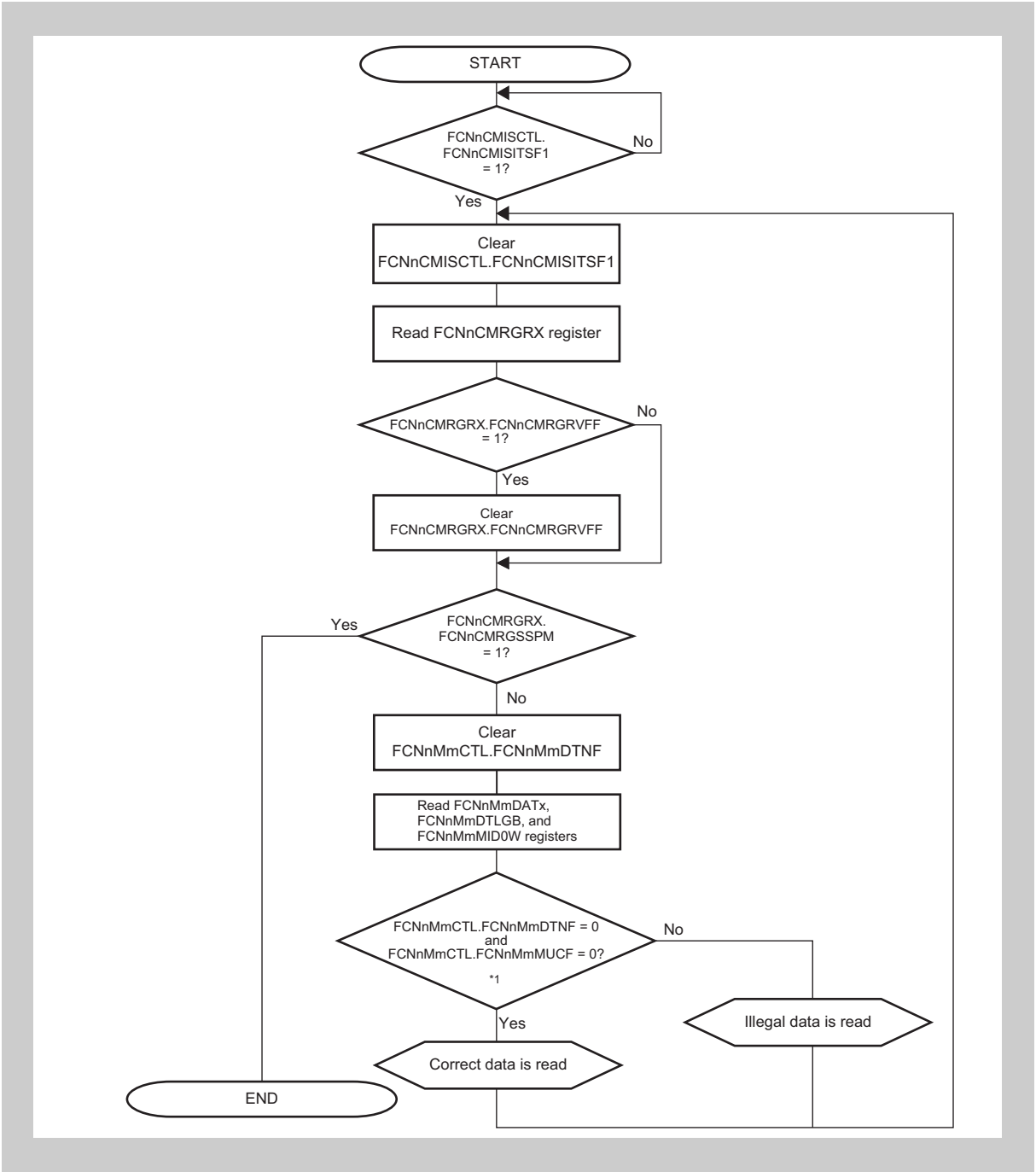


Figure 20-33 Reception via Software Polling

Note 1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits using one read access.

Note 2. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as receive history list registers of the FCN module, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, re-set FCNnGMCLSSMO, discard actions and results of the processing, and then perform processing again.

Note 3. Once FCNnCMRGRX.FCNnCMRGRVFF is set, the receive history list is inconsistent. Consider to examine all configured receive buffers to check reception.

20.14.4 Power Save Modes

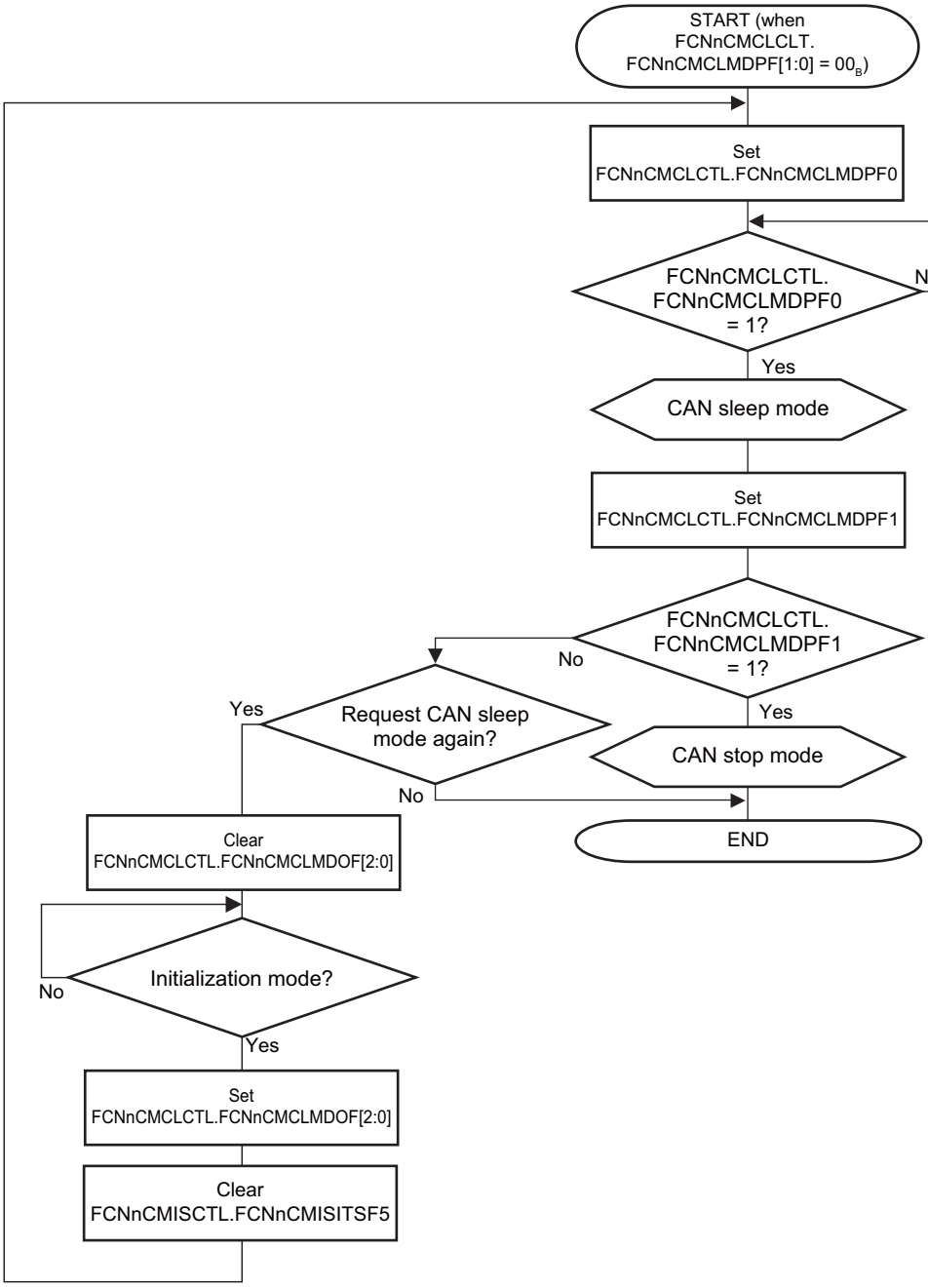


Figure 20-34 Setting FCN Sleep Mode and Stop Mode

Caution To abort transmission before making a request for transition to FCN sleep mode, perform transmission abort processing according to previously given flowcharts.

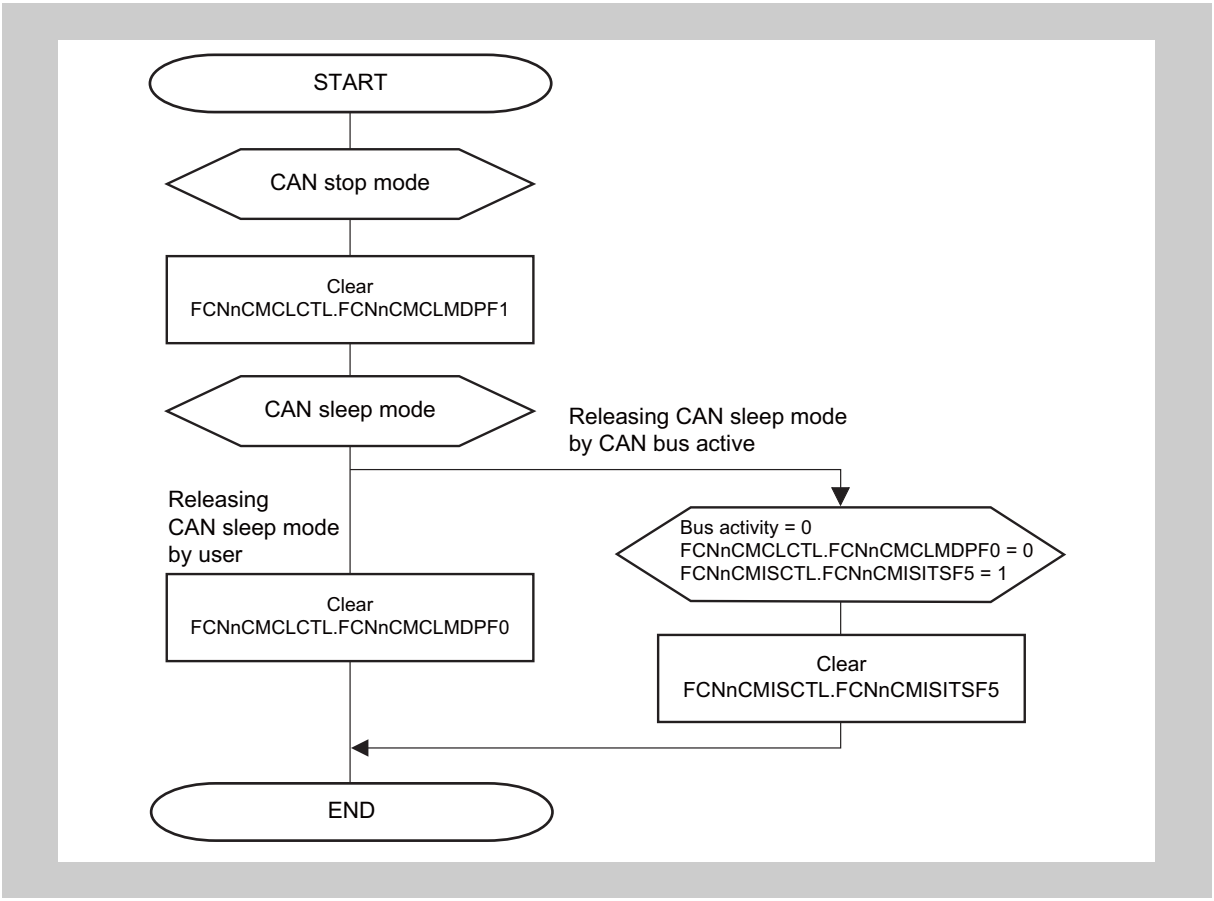


Figure 20-35 Releasing FCN Sleep Mode and Stop Mode

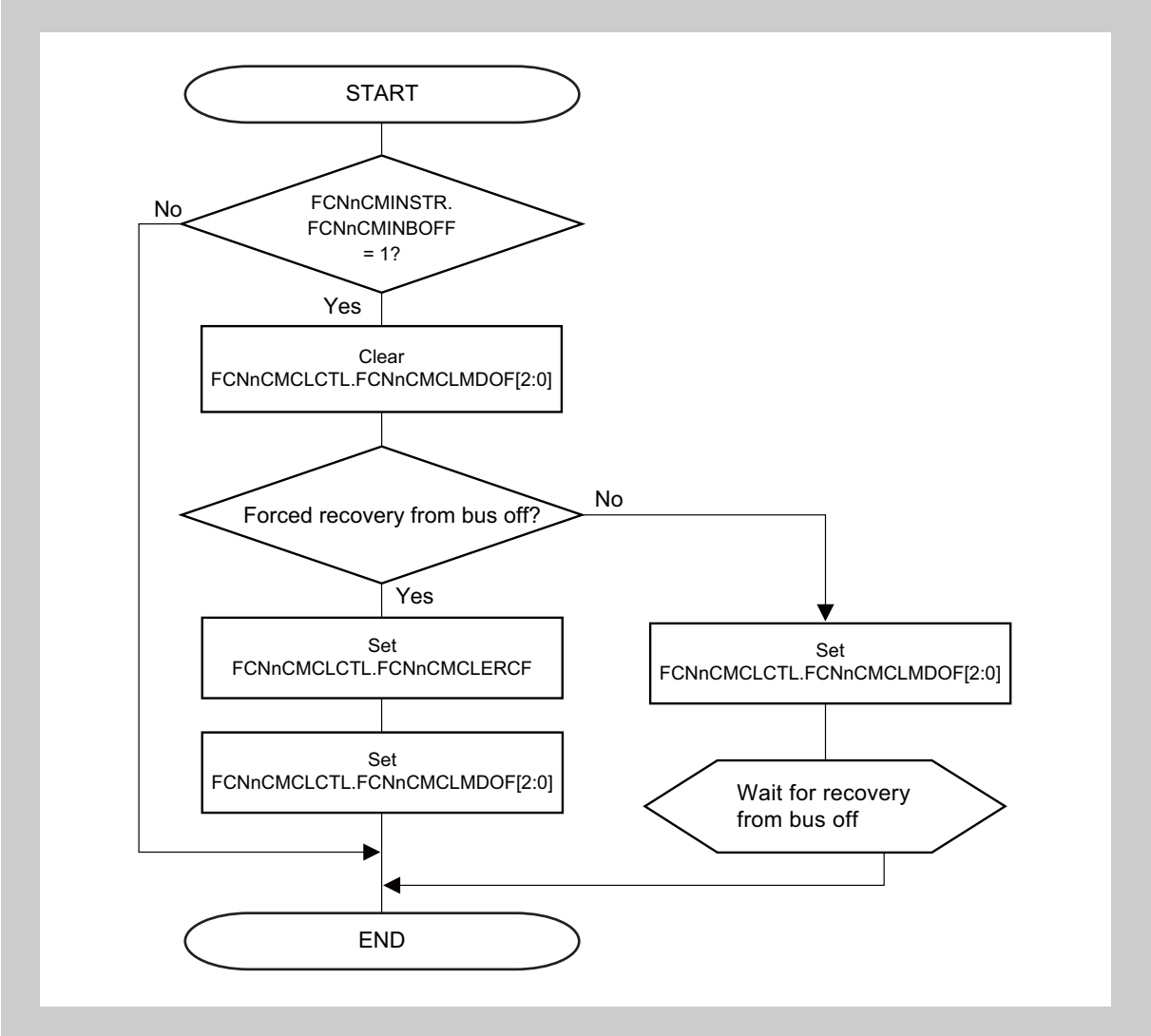


Figure 20-36 Recovery from Bus-off

Caution When the transmission from initialization mode to any operating mode is requested in the bus-off recovery sequence and bus-off recovery sequence is re-executed, the reception error counter is cleared.
Therefore, it is necessary to detect 11 consecutive recessive-level bits 128 times again on the bus.

Note Operating mode: Normal operating mode, normal operating mode with ABT, receive-only mode, single-shot mode, self-test mode

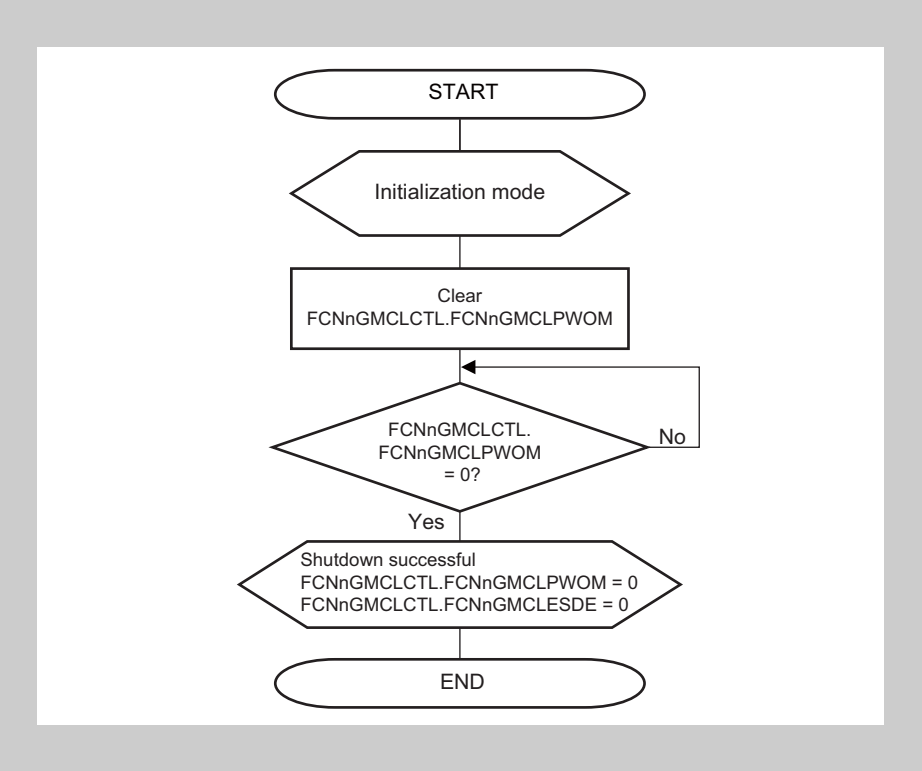


Figure 20-37 Normal Shutdown Processing

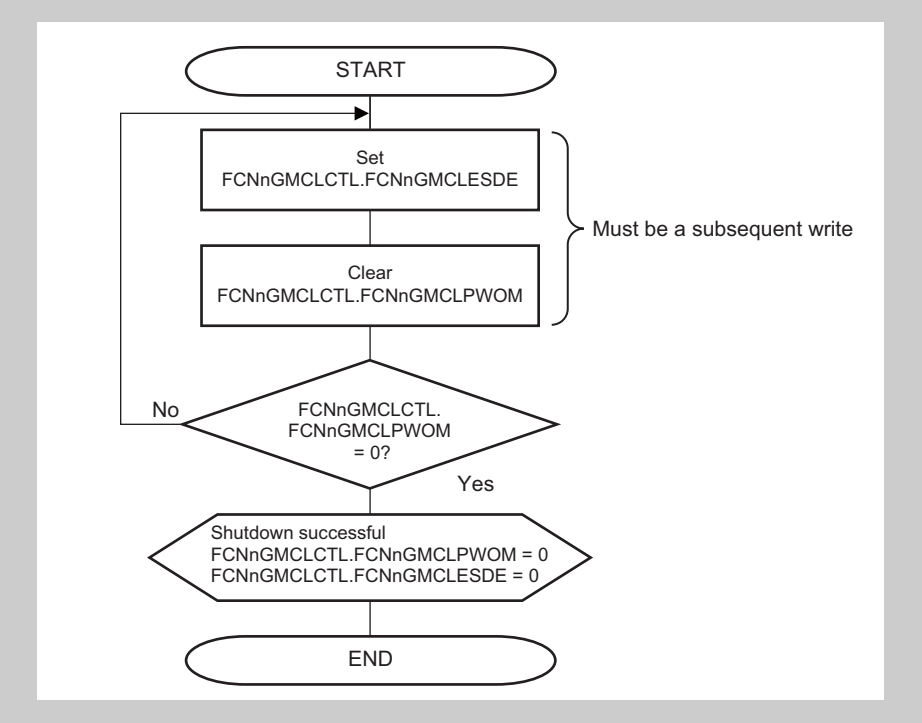


Figure 20-38 Forced Shutdown Processing

Caution Do not make read access or write access to any registers by software during a time period from the setting of the FCNnGMCLESDE bit until clearing of the FCNnGMCLPWOM bit.

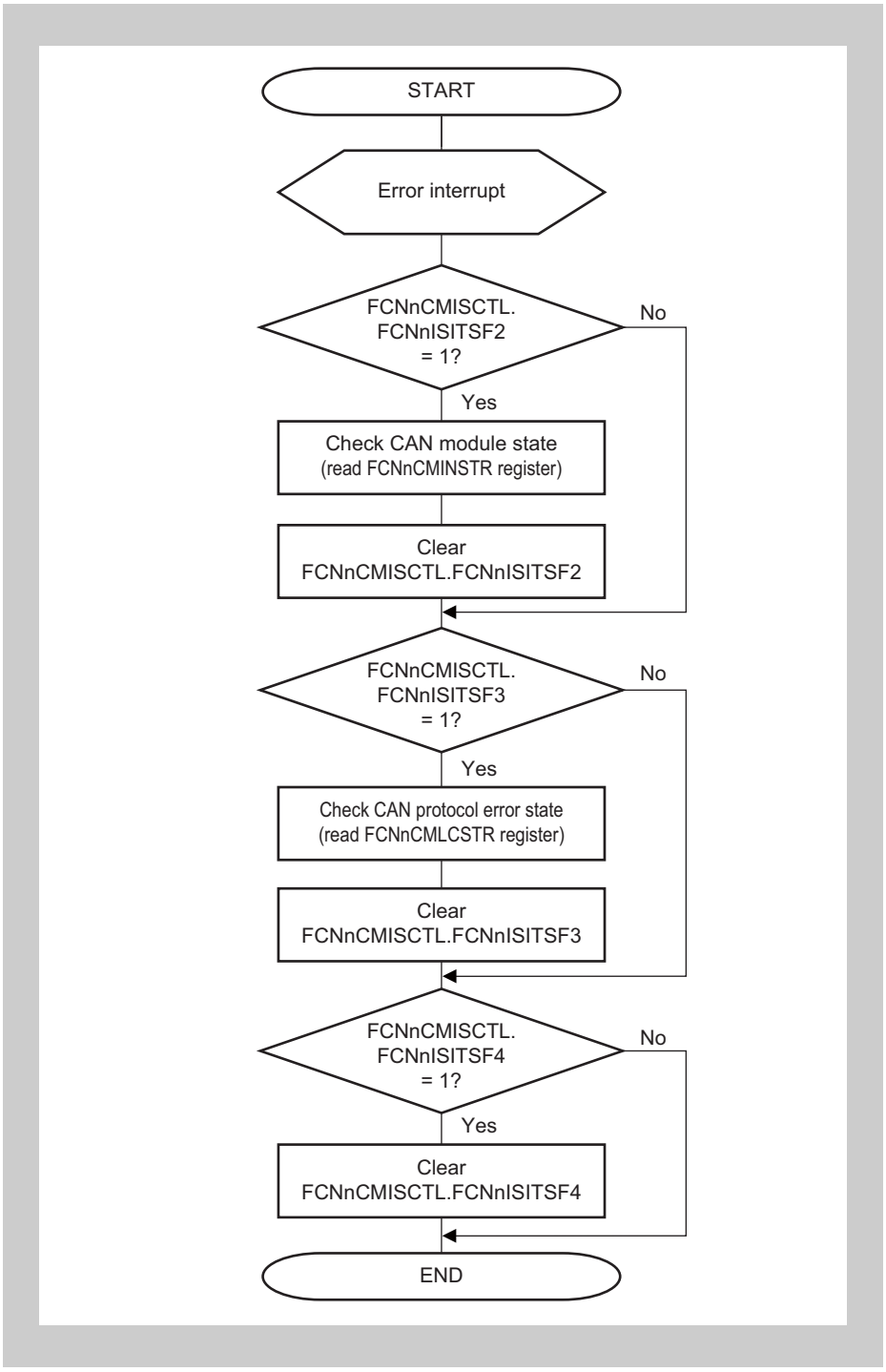


Figure 20-39 Error Handling

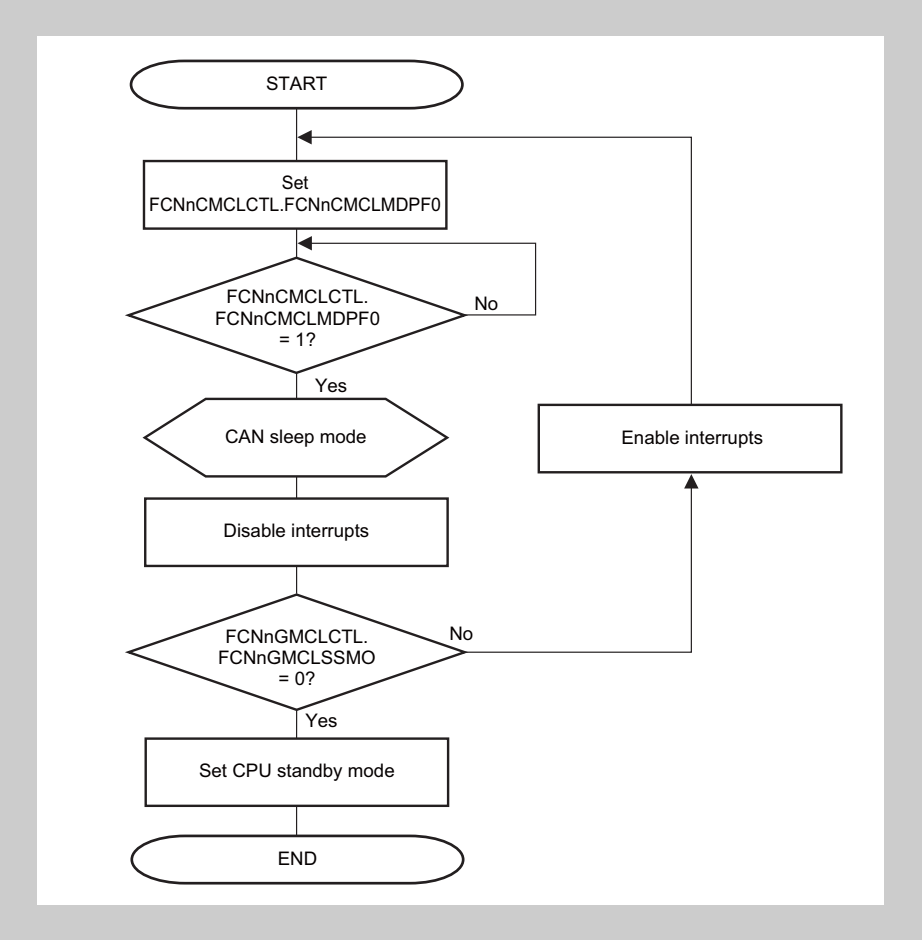


Figure 20-40 Setting CPU Standby (from FCN Sleep Mode)

Note 1. Before setting the CPU to CPU standby mode, check whether the CPU is in FCN sleep mode.
However, after FCN sleep mode is checked, FCN sleep mode may be cancelled by wake-up from the CAN bus until the CPU is set to CPU standby mode.

Note 2. A wake-up condition may occur on the CAN bus during a time period from checking of FCNnGMCLSSMO = 0 until CPU standby mode is set. In that case, the CAN module releases sleep mode, the FCNnCMISITSF5 bit is set, and a wake-up interrupt is generated if it is enabled.

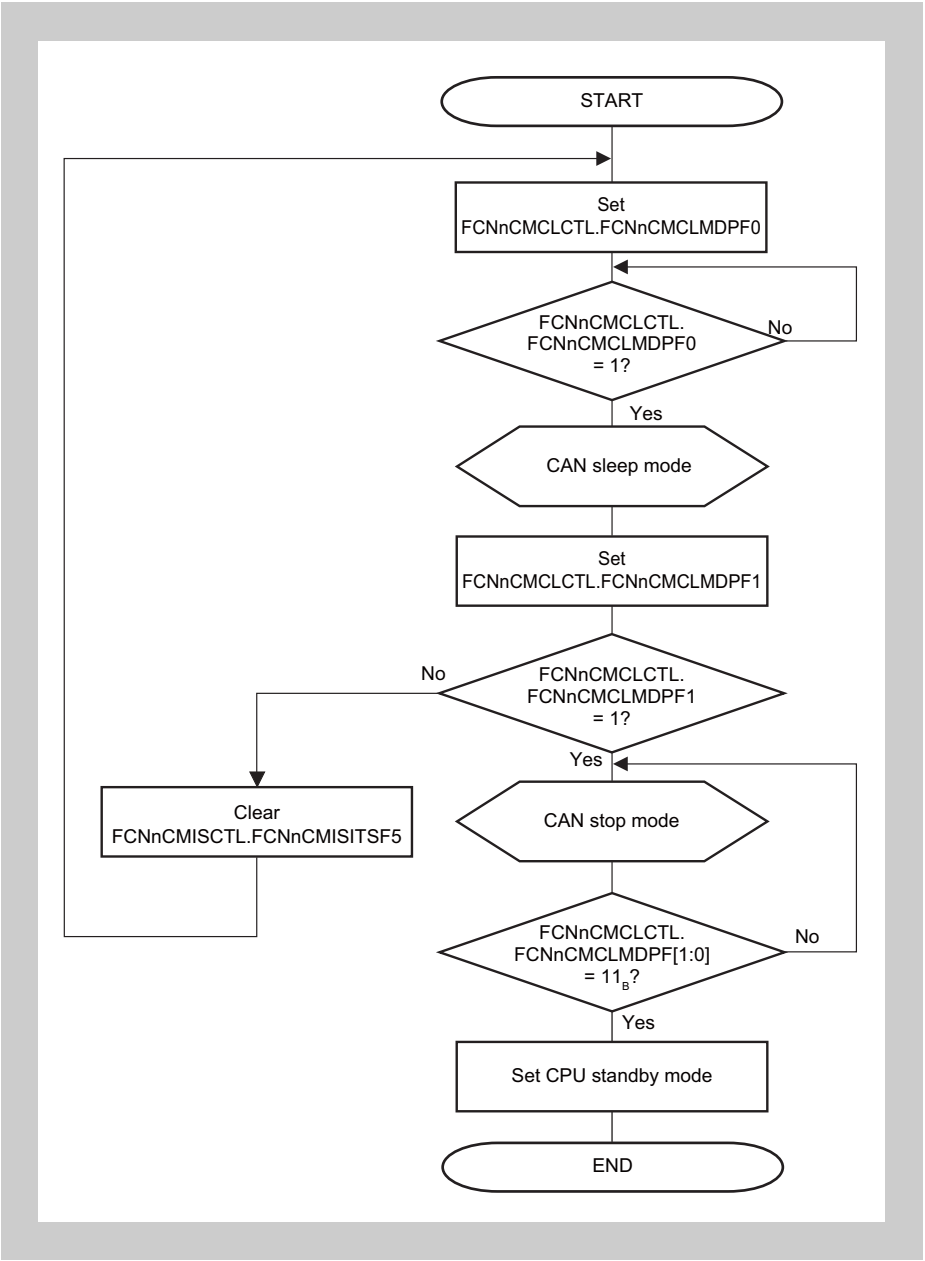


Figure 20-41 Setting CPU Standby (from FCN Stop Mode)

Caution FCN stop mode can be released only by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 01_B, and is not released by a change in the FCN bus state.

Section 21 Clocked Serial Interface G (CSIG)

This section contains a general description of the clocked serial interface G (CSIG).

21.1 CSIG Features

Instances

This product has three instances of the clocked serial interface G blocks.

Table 21-1 Instances of CSIG

Clocked Serial Interface G	
Instances	3
Name	CSIG0 to CSIG2

Instances index n

Throughout this section, the individual instances of a clocked serial interface G is identified by the index “n” (n = 0 to 2), for example, CSIGnCTL0 for the CSIGn control register 0.

Register addresses

All CSIGn register addresses are given as address offsets to the individual base addresses <CSIGn_base0> and <CSIGn_base1>.

The base address <CSIGn_base> of each CSIGn is listed in the following table:

Table 21-2 Register Base Addresses <CSIGn_base0> and <CSIGn_base1>

CSIGn Instance	<CSIGn_base0> Address	<CSIGn_base1> Address
CSIG0	FF70 0000_H	FFFF $E400_H$
CSIG1	FF71 0000_H	FFFF $E500_H$
CSIG2	FF72 0000_H	FFFF $E600_H$

Clock supply

The following clock is input on CSIGn.

Table 21-3 CSIGn Clock Supply

Channel of CSIGn	CSIGn Clock	Connected to
CSIG0 to CSIG2	PCLK	Clock controller

Interrupt The clocked serial interface G can generate the following interrupt requests:

Table 21-4 CSIGn Interrupt Requests

CSIGn Signals	Function	Connected to
CSIGnTIC	Communication status interrupt	Interrupt controller INTCSIGnIC DMA, DTS
CSIGnTIR	Reception status interrupt	Interrupt controller INTCSIGnIR DMA, DTS
CSIGnTIRE	Reception error interrupt	Interrupt controller INTCSIGnIRE

I/O signals The I/O signals of the clocked serial interface G are listed in the following table.

Table 21-5 CSIGn I/O Signals

CSIGn Signals	Function	Connected to
CSIGnTSCK	Serial clock signal	Port CSIGnSC
CSIGnTSI	Serial data input signal	Port CSIGnSI
CSIGnTSO	Serial data output signal	Port CSIGnSO
CSIGnTRYI	Handshake input signal	Port CSIGnRYI
CSIGnTRYO	Handshake output signal	Port CSIGnRYO

Port Groups Use CSIGn pins from the groups listed below.

Table 21-6 Port Groups for CSIGn

CSIGn Channel	Pin Name	Group		
		1	2	3
CSIG0	CSIG0SC	P0_2	P0_7	P1_9
	CSIG0SO	P0_1	P0_6	P1_8
	CSIG0SI	P0_0	P0_5	P1_7
	CSIG0RYI	-	-	P2_0
	CSIG0RYO	-	-	P4_4
CSIG1	CSIG1SC	P3_4	P4_4	-
	CSIG1SO	P3_3	P4_3	-
	CSIG1SI	P3_2	P4_2	-
	CSIG1RYI	P2_1	-	-
	CSIG1RYO	P4_5	-	-
CSIG2	CSIG2SC	P1_3	P4_3	P6_2
	CSIG2SO	P1_2	P4_2	P6_1
	CSIG2SI	P1_1	P4_1	P6_0
	CSIG2RYI	-	P3_7	-
	CSIG2RYO	-	P1_7	-

21.2 Functional Overview

Features summary

- Three-line clock-synchronous serial communications
- Master mode and slave mode selectable
- Built-in baud rate generator
- Maximum transmission speed:
  - In master mode: 8 Mbps
  - In slave mode: 6.6 Mbps (but no more than PCLK/6 Mbps)
- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- EDL (extended data length) function for transferring data with more than16 bits
- Three selectable transfer modes:
  - Transmit-only mode
  - Receive-only mode
  - Transmit/receive mode
- Built-in handshake function
- Separate transmit and receive buffers (two 16-bit registers)
- Error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (CSIGnTIC, CSIGnTIR, CSIGnTIRE)
- LBM (loop-back mode) function for self test

The block diagram shows the main components of the CSIG.

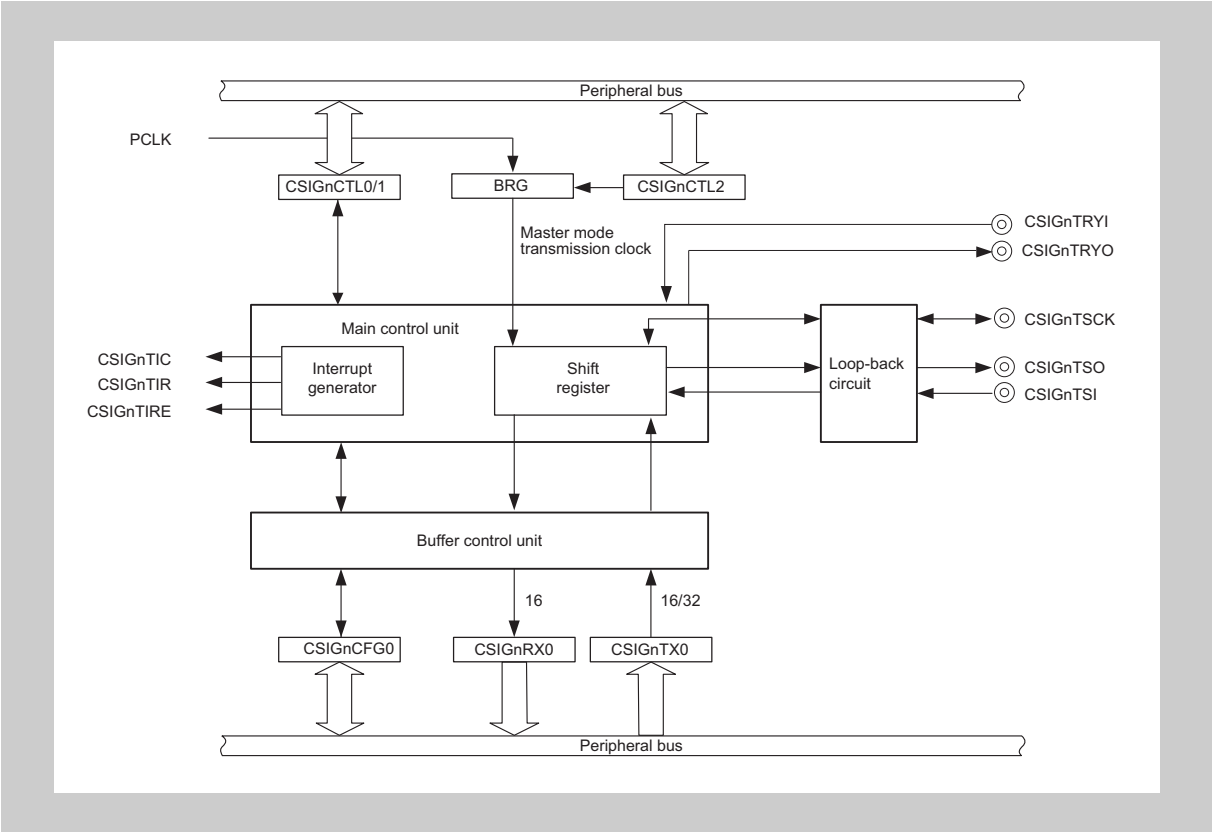


Figure 21-1 CSIG Block Diagram

In master mode, the serial communications clock (CSIGnTSCK) is generated by the internal baud rate generator (BRG). In slave mode, the module supplies the serial communications clock through CSIGnTSCK.

21.3 Functional Description

The clocked serial interface G module uses three signals for communication.

- Serial communications clock CSIGnTSCK (output in master mode, and input in slave mode)
- Data output signal CSIGnTSO
- Data input signal CSIGnTSI

Data transmission is bit-wise and serial and synchronous to the transmission clock.

The registers used for CSIG settings are listed in the following table.

Register	Function
CSIGnCTL0	Supplies or stops supply of the operating clock signal and enables or disables data transmission and reception.
CSIGnCTL1	Controls options like interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master or slave mode. In master mode, this also selects the rate of the signal produced by the baud rate generator.
CSIGnCFG0	Configures the communication protocol

21.3.1 Master/Slave Mode

CSIG operation in master mode or in slave mode depends on the setting of CSIGnCTL2.CSIGnPRS[2:0].

(1) Master Mode

In master mode, the serial communications clock is generated by the internal baud rate generator (BRG) and provided by signal CSIGnTSCK.

Setting a value other than 111_B in CSIGnCTL2.CSIGnPRS[2:0] and a value other than 0 in CSIGnCTL2.CSIGnBRG[11:0] makes the settings for the BRG effective.

The default level of CSIGnTSCK depends on the clock phase selection bit: it is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for 8 data bits, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first.

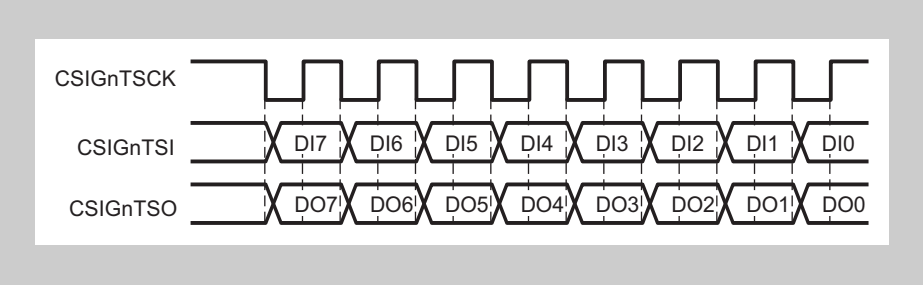


Figure 21-2 Transmit/Receive in Master Mode

(2) Slave Mode

In slave mode, another device is the communication master. The external clock is received by signal CSIGnTSCK. Transmit/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

Note When using slave mode, disable the baud rate generator (BRG) by clearing bits CSIGnCTL2.CSIGnBRS[11:0].

The example below shows the communication in slave mode for 8 data bits, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

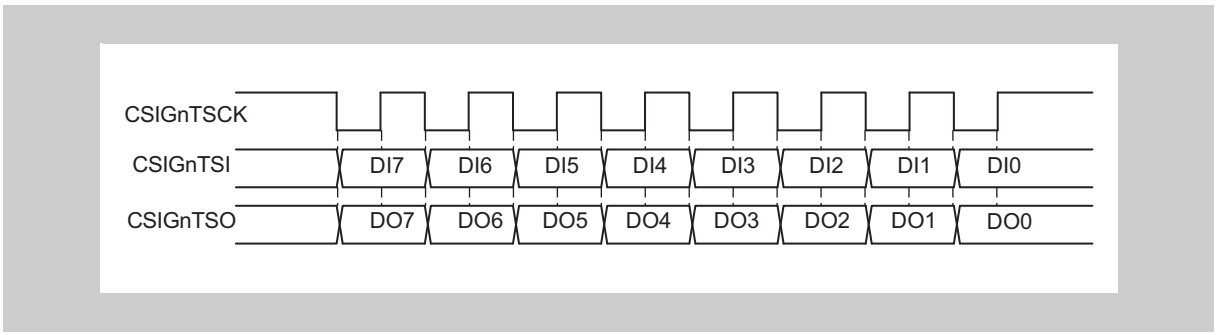


Figure 21-3 Transmit/Receive in Slave Mode

21.3.2 Master/Slave Connections

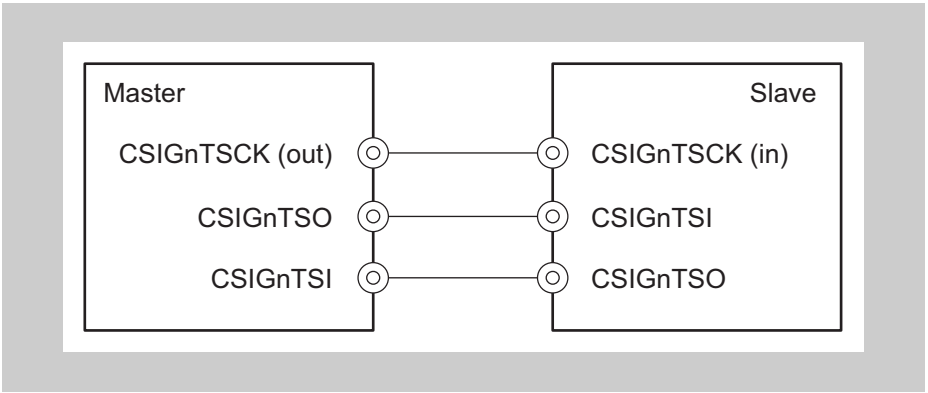


Figure 21-4 Simple Master/Slave Connection

21.3.3 Selection of Serial Communications Clock

In master mode, the baud rate for the serial communications clock is set by the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

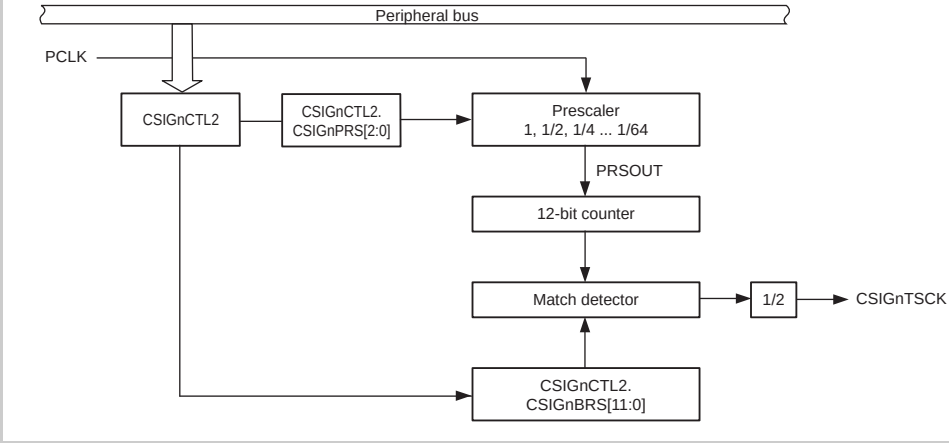


Figure 21-5 BRG Block Diagram

Clearing CSIGnCTL2.CSIGnBRS[11:0] disables the BRG and stops output on all CSIGnTSCKx pins.

Baud rate calculation

The baud rate is calculated as follows:

$$CSIGnTSCKx = PCLK / (2^m \times k \times 2)$$

Where,

m = CSIGnCTL2.CSIGnPRS[2:0] = 0 to 6

k = CSIGnCTL2.CSIGnBRS[11:0] = 1 to 4095

Baud rate upper/ lower limits

When setting the baud rate, note the following points.

- The baud rate must be no greater than 8.0 Mbps in master mode.
- The baud rate must be no greater than 6.6 Mbps in slave mode (and no greater than PCLK/6 Mbps). For operation with an external master, check that the baud rate from the master is in this range.
- Minimum baud rate in both modes is PCLK/524160.

21.3.4 Data Transfer Modes

(1) Trasmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 puts the CSIG in transmit-only mode. After setting CSIGnCTL0.CSIGnTXE = 1, transmission starts when transmit data is written in the CSIGnTX0W or CSIGnTX0H register.

In master mode, transmission starts when transmit data is written in the CSIGnTX0W or CSIGnTX0H register.

In slave mode, transmission starts on clock input to CSIGnTSCK after the settings indicated above.

Caution In case transmit-only mode has been entered after any reception mode, the data in the CSIGnRX0 buffer becomes undefined after completion of the first transmission. Consequently the reception register CSIGnRX0 has to be read before changing to transmit-only mode.

(2) Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in receive-only mode.

In master mode, once the CSIGnCTL0.CSIGnRXE is set to 1, reading from the CSIGnRX0 register causes transfer to start. As long as CSIGnBCTL0.CSIGnSCE = 1, reading from the receive data register (CSIGnRX0) triggers further reception.

In slave mode, reception starts when the communication clock CSIGTSCK from the master is received.

Note In receive-only mode, avoid overwriting of data by reading previously received data from the reception register (CSIGnRX0) before new data arrive.

Moreover, the communication start bit CSIGnBCTL0.CSIGnSCE has to be set to 1 and then set back to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Before starting the first receive operation, read CSIGnRX0 (dummy data) with CSIGnBCTL0.CSIGnSCE = 1.
2. Wait for the reception interrupt CSIGnTIR.
3. Read CSIGnRX0 (received data).
   If further data reception is to continue from step 2, continue to read all received data every time reception is completed.
   Before reading the last of the received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0. For details, refer to section 21.4 *chapter(6) CSIGnBCTL0 - CSIG Receive-Only Mode Control Register 0*.

(3) Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when transmit data is written to the CSIGnTX0W or CSIGnTX0H register.

21.3.5 Data length Selection

(1) Data Length from 7 to 16 Bits

Transmission data length is selectable from 7 to 16 bits using the CSIGnDLS[3:0] bits in the CSIGnCFG0 register. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

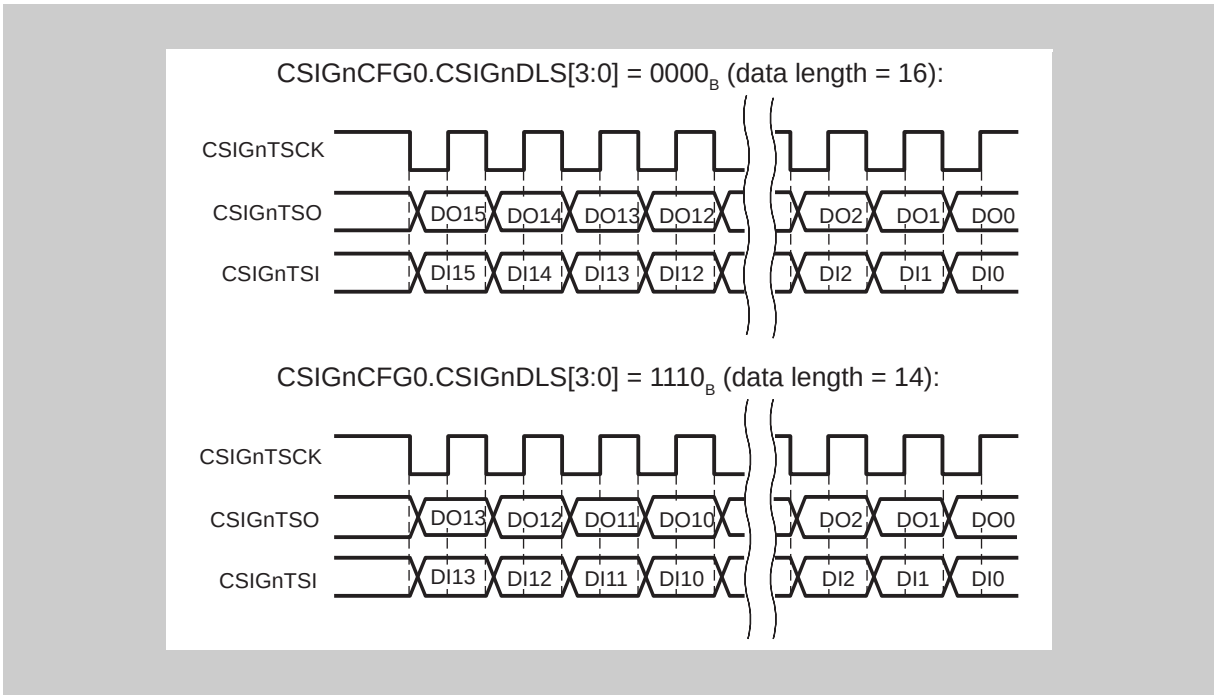


Figure 21-6 Data Length Select Function

(2) Data Length Greater than 16 Bits.

If the data to be transmitted/received exceed 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting the CSIGnCTL1.CSIGnEDLE bit to 1.

The EDL function works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, a string of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The remainder defines the "data length" that has to be specified in the CSIGnCFG0.CSIGnDLS[3:0] bits.
- If a 16-bit block is to be transmitted, CSIGnTX0W.CSIGnEDL must be set to 1. In this case, data written to CSIGnTX0W are transmitted as 16-bit data regardless of setting of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- The transfer is complete after the data with the specified data length (the remainder with CSIGnTX0W.CSIGnEDL = 0) has been sent.

Example Example for sending 40-bit data, for example the string $123456789A_H$:

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8_D.
- To send the string $123456789A_H$ with MSB first, write the following sequence to CSIGnTX0W:
  - $2000\ 1234_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $2000\ 5678_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $0000\ 009A_H$ (CSIGnTX0W.CSIGnEDL = 0)

The following figure shows the timing.

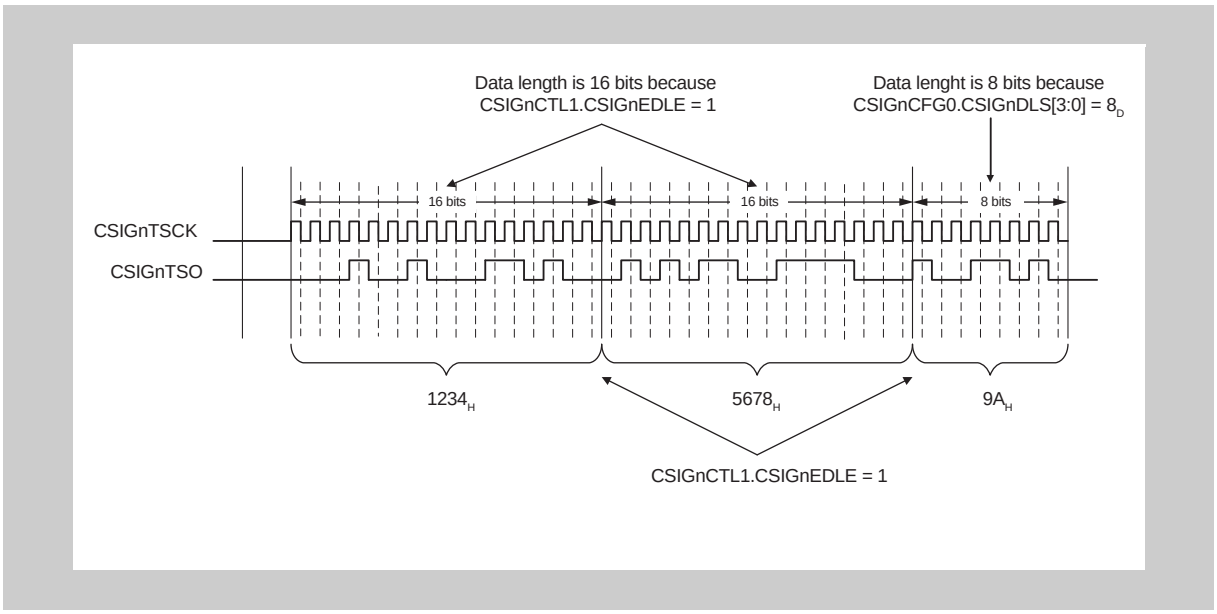


Figure 21-7 EDL Timing Diagram

Note 1. It is not possible to send two consecutive data with a data length of less than 7 bits.

Note 2. If parity is enabled, the parity bit is added after the last bit.

Note 3. When sending data in LSB first mode/MSB first mode, write to the CSIGnTX0W register according to the relevant sequence below (when transmission data is 123456_H).

- CSIGnCFG0.CSIGnDIR = 1: LSB first
  CSIGnTX0W = $2000\ 3456_H$ (CSIGnEDL = 1)
  CSIGnTX0W = $0000\ 0012_H$ (CSIGnEDL = 0)
- CSIGnCFG0.CSIGnDIR = 0: MSB first
  CSIGnTX0W = $2000\ 1234_H$ (CSIGnEDL = 1)
  CSIGnTX0W = $0000\ 0056_H$ (CSIGnEDL = 0)

Note 4. Use of the EDL function in receive-only mode during slave operation is prohibited.

Note 5. Setting a data length of less than 7 bits is only possible with the EDL function is in use.

21.3.6 Serial Data Direction Selection

The serial data direction is selectable using the CSIGnDIR bit in the CSIGnCFG0 register. The examples below show the communication for 8-bit data (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B):

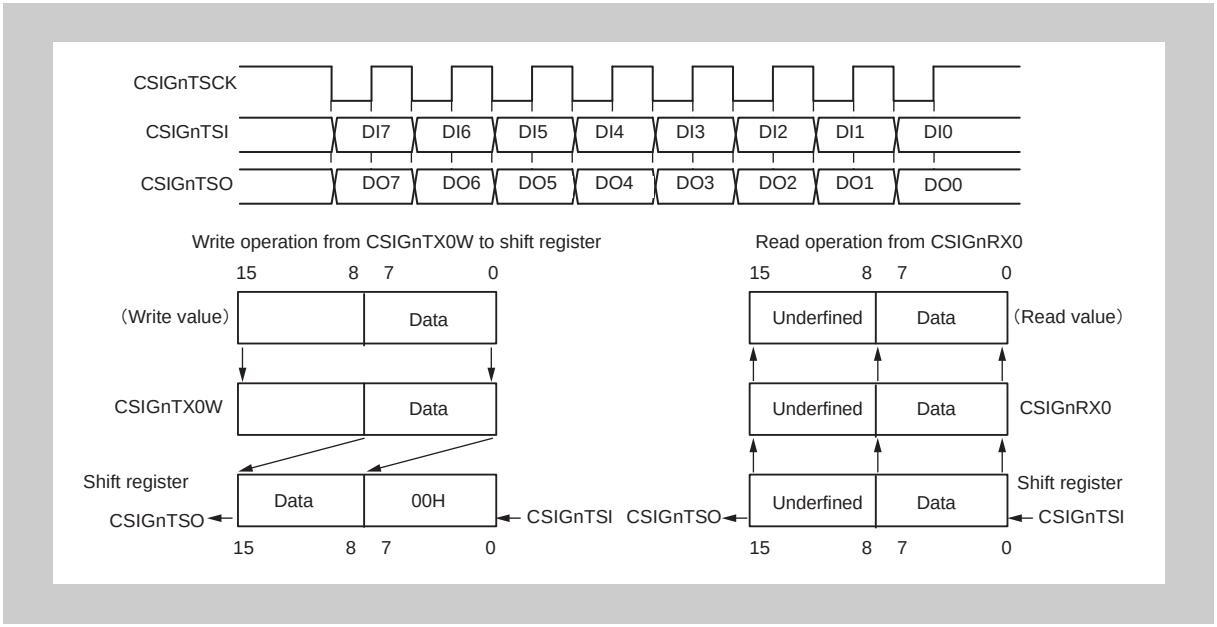


Figure 21-8 Serial Data Direction Select Function - MSB First (CSIGnCFG0.CSIGnDIR = 0)

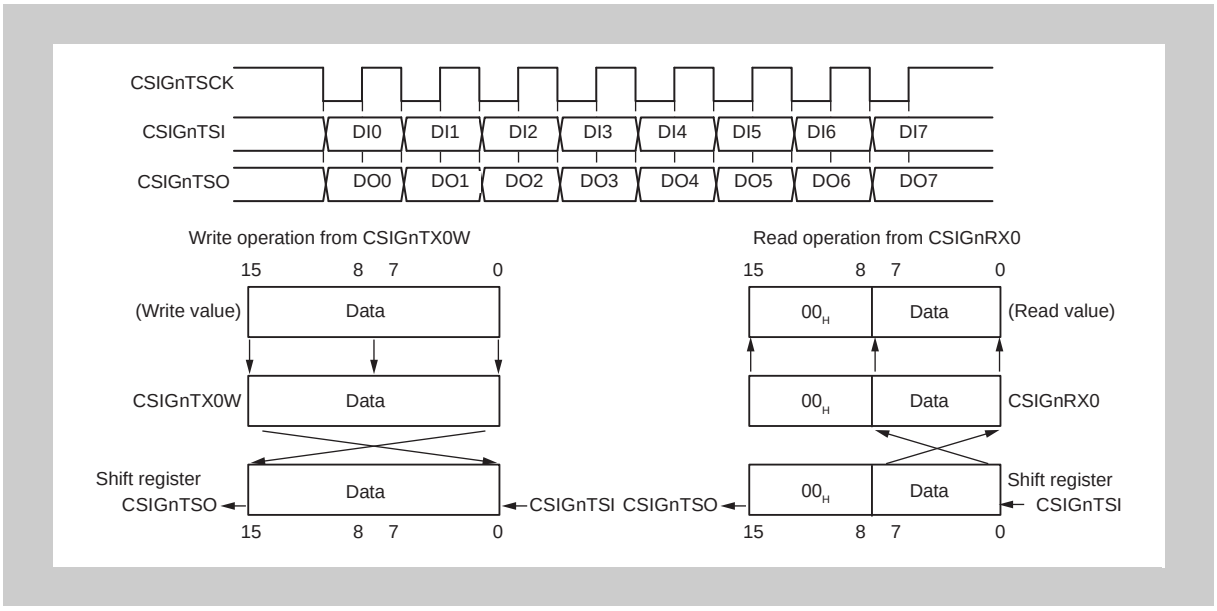


Figure 21-9 Serial Data Direction Select Function - LSB First (CSIGnCFG0.CSIGnDIR = 1)

21.3.7 Communication in Slave Mode

The following figure shows the communication signals and timings in slave mode.

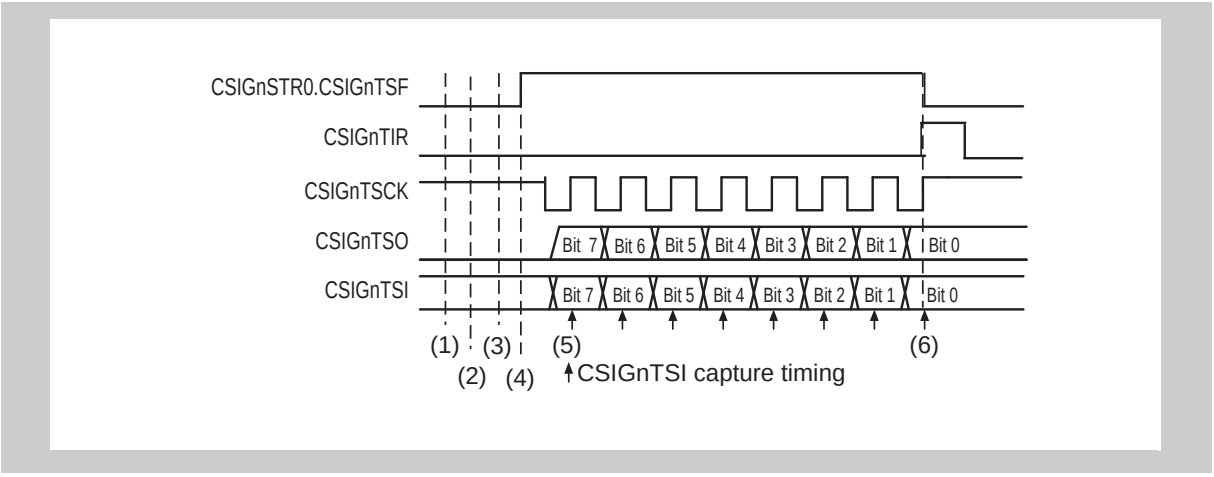


Figure 21-10 Receive/Transmit Communication Timing in Slave Mode

1. CSIG is put into slave mode by setting CSIGnCTL2.CSIGnPRS[2:0] = 111_B.
2. CSIGnCTL1.CSIGnCKR and CSIGnCFG0.CSIGnDAP are 0. Data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B). Data direction is MSB first (CSIGnCFG0.CSIGnDIR = 0).
3. When CSIG is set to transmit/receive mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, and CSIGnCTL0.CSIGnRXE = 1), communication start is enabled.
4. The transfer status flag (CSIGnSTR0.CSIGnTSF) is automatically set when data for transfer are written to the transmission register (CSIGnTX0W or CSIGnTX0H).
5. When an external clock signal is detected as the CSIGnTSCK signal, the slave immediately starts transferring data through the CSIGnTSO pin and capturing data from the CSIGnTSI pin.
6. Reception of the last of the data is complete.
7. Received data are read.

21.3.8 CSIG Interrupts

CSIG can generate the following interrupts:

- CSIGnTIC
- CSIGnTIR
- CSIGnTIRE

(1) CSIGnTIC (Communication Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (8-bit data length), and CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing).

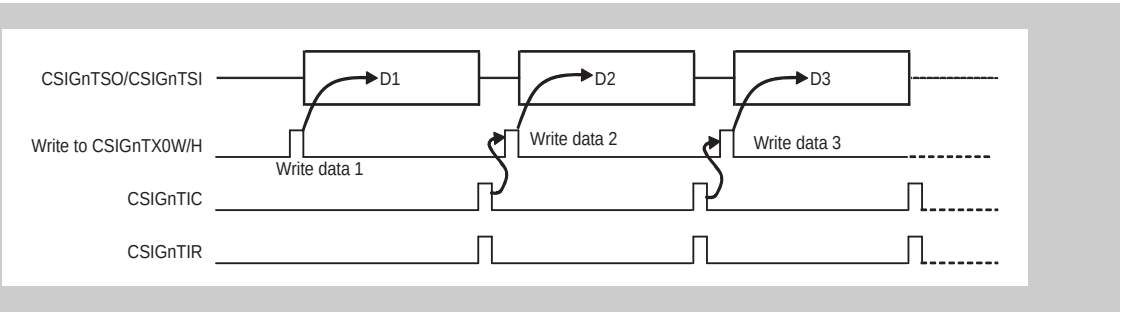


Figure 21-11 Generation of CSIGnTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)

However, CSIGnTIC can also be set up to occur when the CSIGnTX0 or CSIGnTX0H register is free for the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

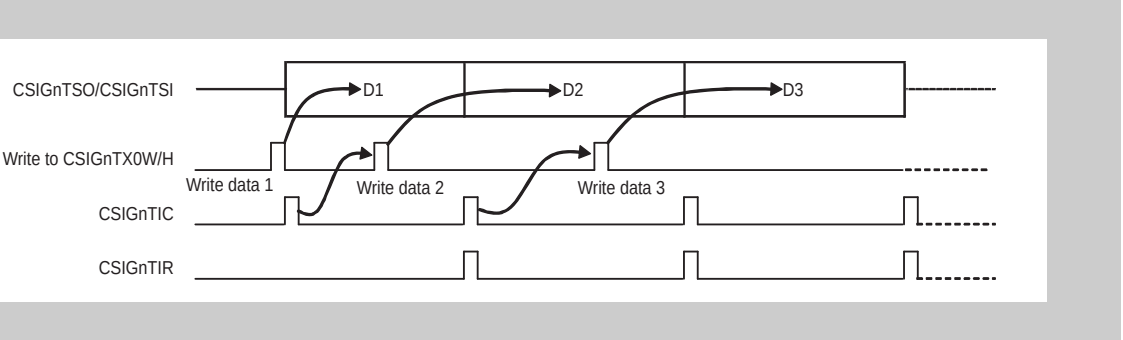


Figure 21-12 Generation of CSIGnTIC at the Beginning of Communication

(2) CSIGnTIR (Reception Interrupt)

This interrupt is generated in receive-only mode or transmit/receive mode after data has been received and is available in the reception register.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (8-bit data length).

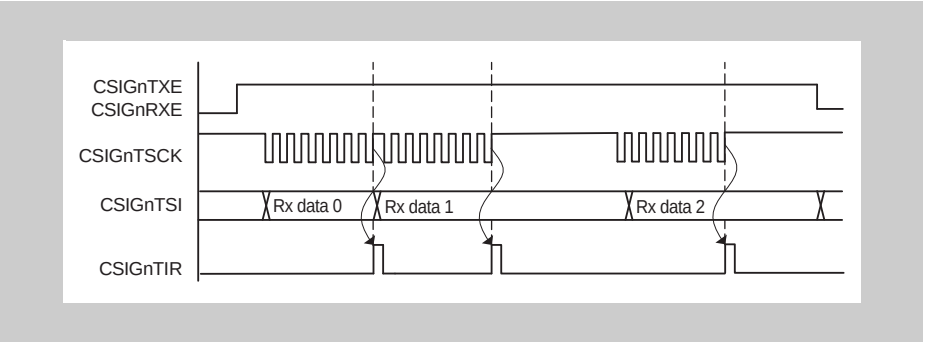


Figure 21-13 Generation of CSIGnTIR

(3) CSIGnTIRE (Reception Error Interrupt)

This interrupt is generated whenever the following errors are detected.

- Parity error
- Data consistency error
- Overrun error

Table 21-7 Data Error Types

Error Type	Communication Status after Error Interrupt
Parity error	Interrupt is generated and communication continues
Data consistency error	Interrupt is generated and communication continues
Overrun error	Interrupt is generated and communication is stopped

The type of error that caused the generation of CSIGnTIRE is indicated in register CSIGnSTR0.

For details on the various error types, refer to section 21.3.11, Error Detection.

(4) All Interrupts Delay

In master mode, delaying all interrupts from the master by half a cycle of the transmission clock (CSIGnTSCK) is possible. This function is not available in slave mode.

Set the CSIGnCTL1.CSIGnSIT bit to 1 to specify this delay.

The following settings are assumed in the figure below showing an example of using the interrupt delay: CSIGnCTL1.CSIGnSIT = 1 (enabling interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phases), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (8-bit data length).

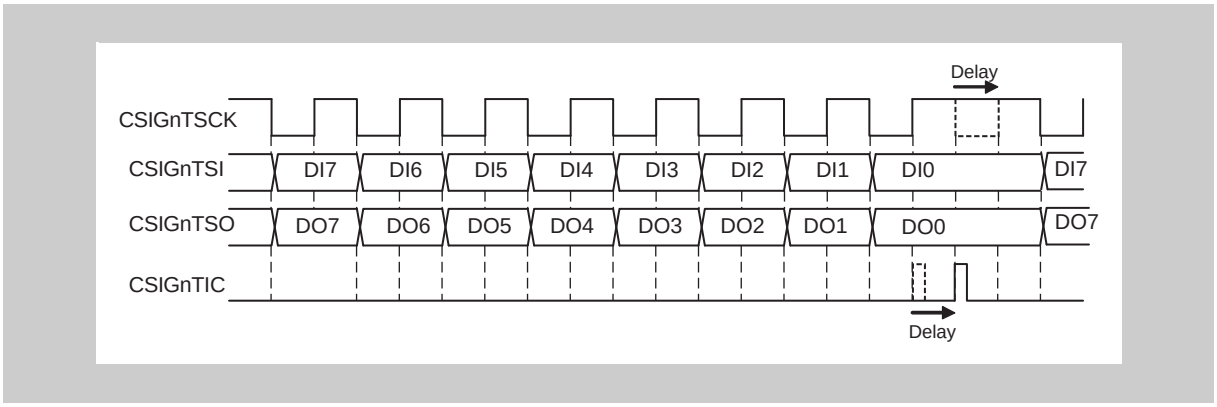


Figure 21-14 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

21.3.9 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIGnCTL1.CSIGnHSE.

For handshake, the signals CSIGnTRYI/CSIGnTRYO are used.

The timing depends on the data phase selection bit CSIGnCFG0.CSIGnDAP.

(1) Slave Mode

When CSIGnCTL1.CSIGnHSE = 1, the slave outputs CSIGnTRYO = 0 when it is busy. This happens when previously received data is still in the CSIGnRX0 register, and new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume 8-bit data length.

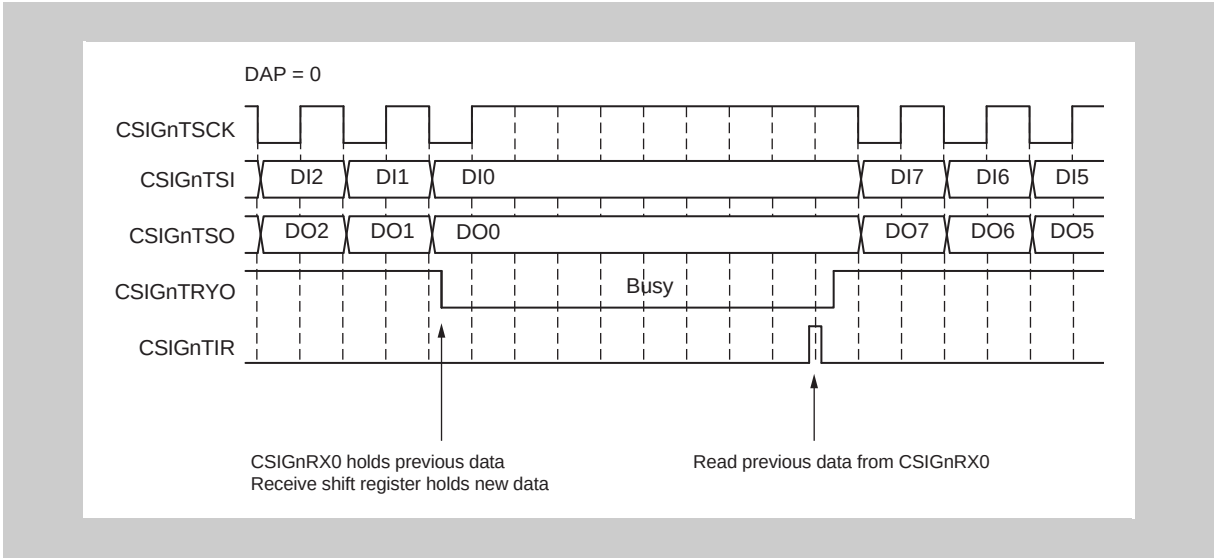


Figure 21-15 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

As long as the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGnTRYO to high (“ready”) as soon as the reception register CSIGnRX0 has been read.

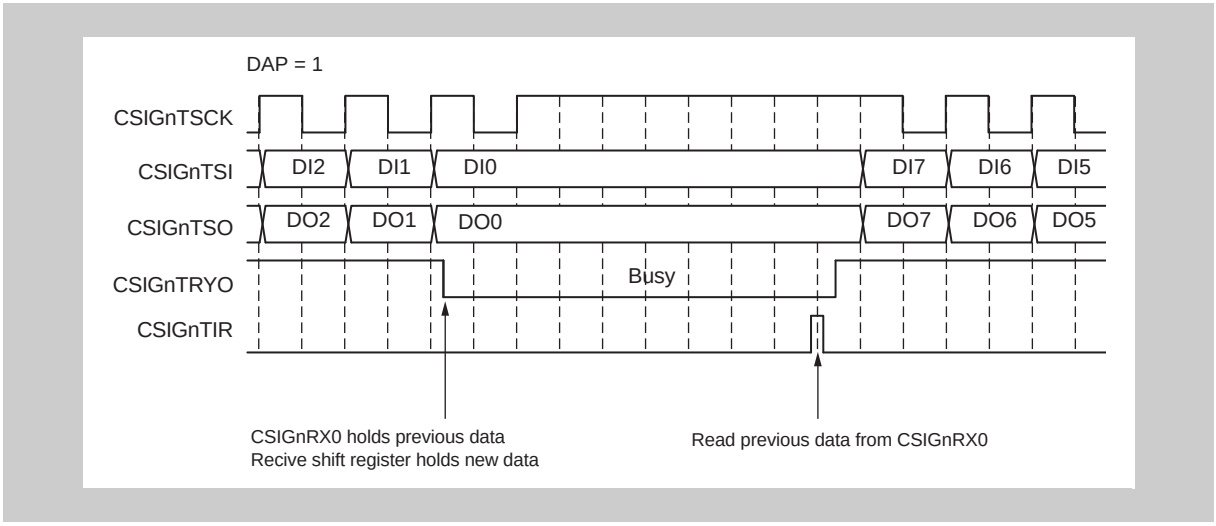


Figure 21-16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

(2) Master Mode

When the master detects CSIGnTRYI = 0, the following transfer is put on hold, and the master goes into wait status. It suspends the clock at CSIGnTSCK.

The CSIGnTRYI level is checked at each half clock cycle of CSIGnTSCK.

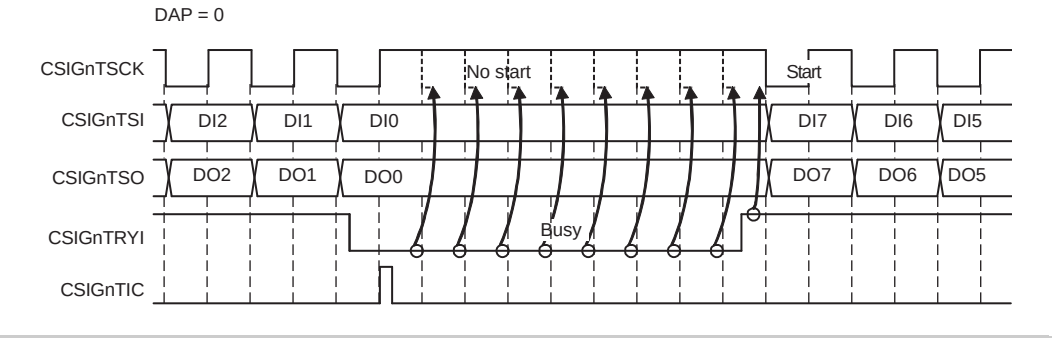


Figure 21-17 Master's Response to CSIGnTRYO Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

A low level from the slave on the CSIGnTRYI pin of the master while data transfer is in progress causes the master to suspend generation of the serial clock after the current transfer is complete.

The master resumes the transfer as soon as the level on the CSIGnTRYI is high (indicating that the slave is “ready”).

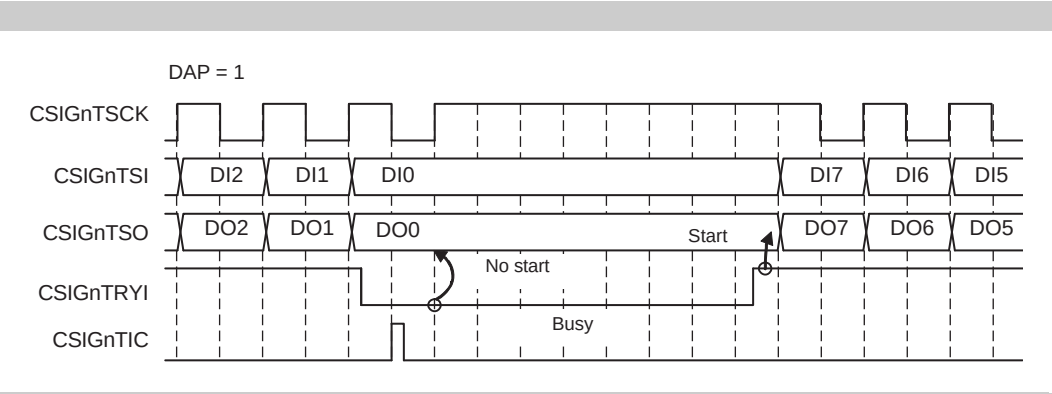


Figure 21-18 Master's Response to CSIGnTRYO Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

Caution CSIGnTRYI must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will be finished.

21.3.10 Loop-Back Mode

Loop-back mode is a special mode for self-testing. This feature is only available in master mode.

During operation in this mode, the transmission and reception signal lines are internally connected, as shown in the figures below. The CSIGnTSCK, CSIGnTSO, and CSIGnTSI signals are disconnected from the port pins. In addition, the CSIGnTSO output is fixed to the low level, and CSIGnTSCK is set to the inactive state. The rest of CSIG works as in normal operation.

In order to test the CSIG, set the CSIGnCTL1.CSIGnLBM bit to 1, carry out normal transfer operations, and then check that the received data is the same as the transmitted data.

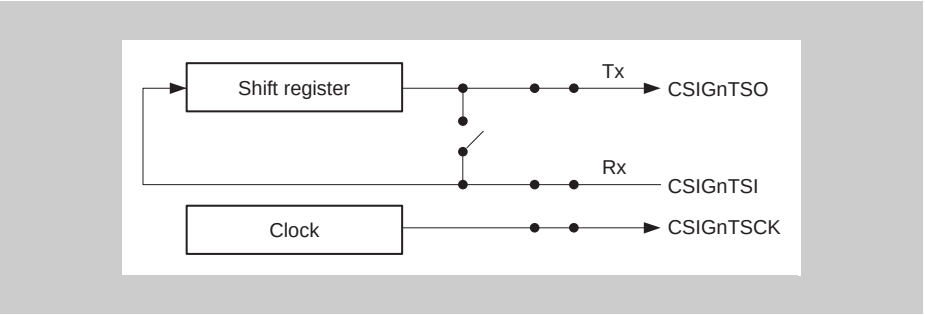


Figure 21-19 Normal Operation

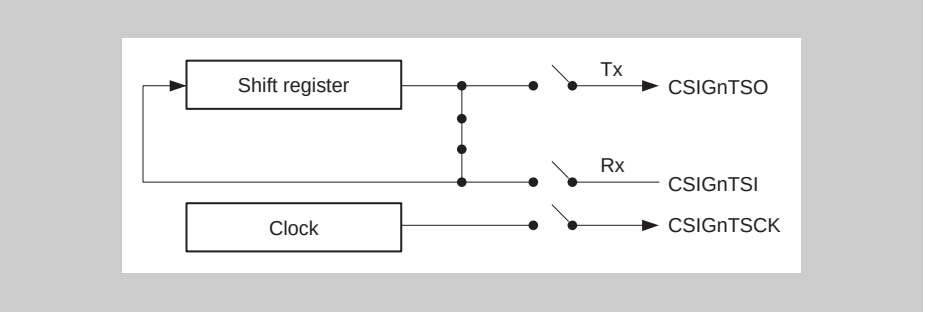


Figure 21-20 Operation in Loop-Back Mode

21.3.11 Error Detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error

Error checking can be individually enabled/disabled for each type.

If one of these errors is detected, the CSIGnTIRE interrupt signal is generated and the flag corresponding to the detected error is set.

(1) Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent to the output pin is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIGnCTL1.CSIGnDCS bit. It is not active if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When data consistency checking is being applied, data transferred from the CSIGnTX0W or CSIGnTX0H register to the shift register are also copied to a separate register. In addition, the physical levels on CSIGnTSO are written into their own shift register.

After completion of the transmission, the transmitted data is compared with the original transmission data.

A mismatch is considered a data consistency error, in which case the following steps proceed:

- The CSIGnTIRE interrupt signal is generated.
- The data consistency error flag (CSIGnSTR0.CSIGnDCE) is set.

Caution For data consistency checking, set the bit in the PIPCn register to which the CSIGnTSO pin is assigned to 0, and set the relevant bit in the PBDCn register to 1.

The function is illustrated in the following block diagram.

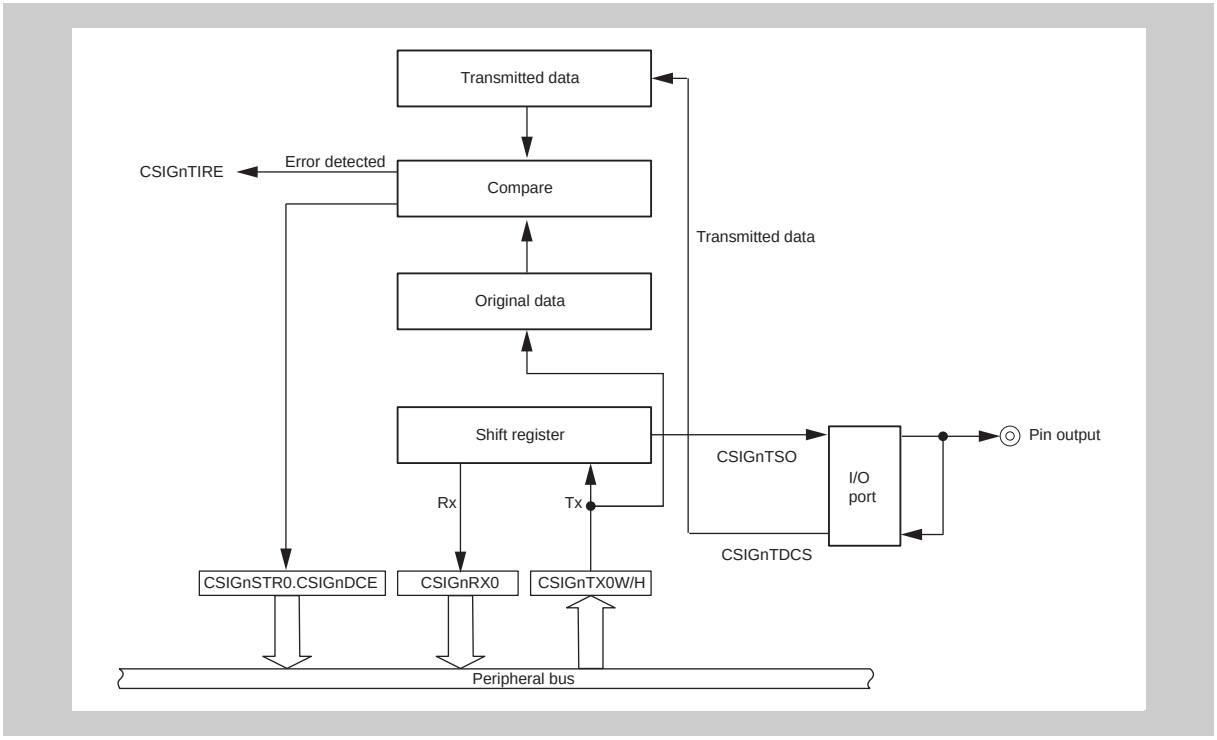


Figure 21-21 Functional Block Diagram of the Data Consistency Check

(2) Parity Check

CSIG can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIGnCFG0.CSIGnPS[1:0].

Parity check is enabled if CSIGnCFG0.CSIGnPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt CSIGnTIRE is generated.
- The parity error flag (CSIGnSTR0.CSIGnPE) is set.

The following figure shows an example.

Data length is 8 bits. The data transmitted is 05_H and 35_H. Parity type is odd.

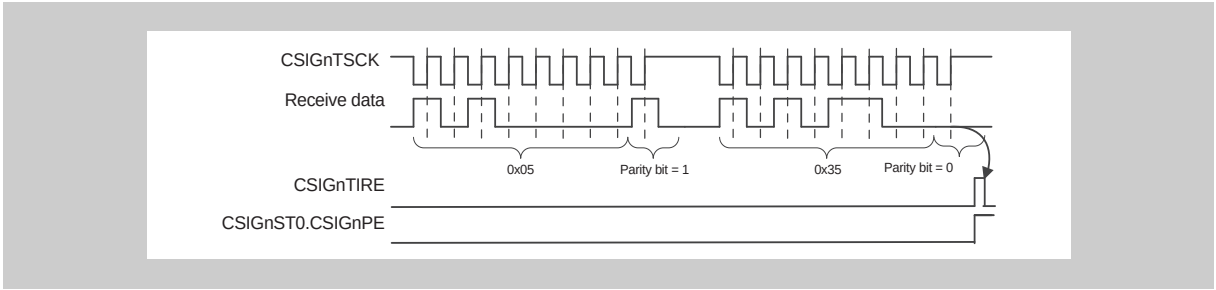


Figure 21-22 Parity Check Example

The parity bit for the first of the data to be received is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit for the subsequently received data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

(3) Overrun Error

This error occurs when new data are received in the reception register (CSIGnRX0) but previously received data have not been read.

The overrun error is not generated if data reception is disabled (CSIGnCTL0.CSIGnRXE = 0) or in master mode.

If overrun occurs:

- Interrupt CSIGnTIRE is generated
- The overrun error flag (CSIGnSTR0.CSIGnOVE) is set.
- Communication is stopped

The following figure illustrates the function.

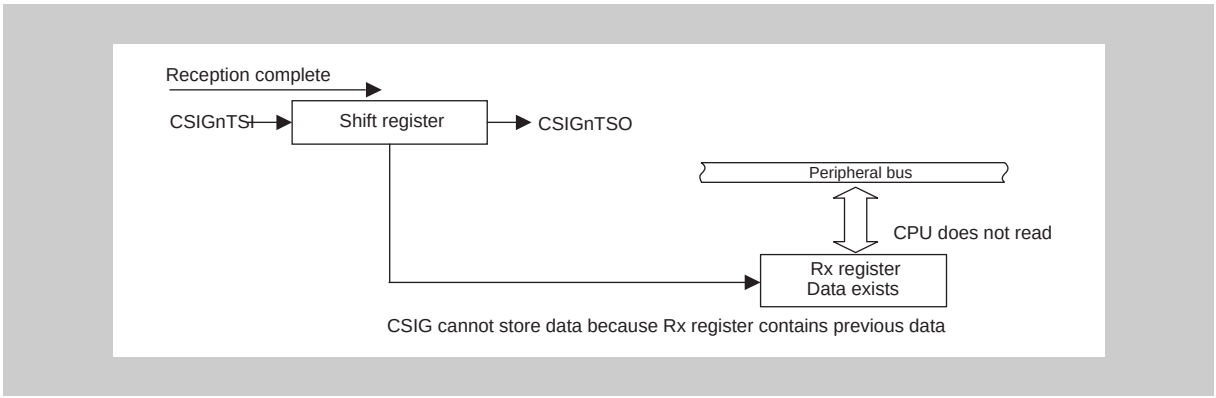


Figure 21-23 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 was not read
- Rx data 4 was received, but cannot be stored

Thus an overrun error occurs.

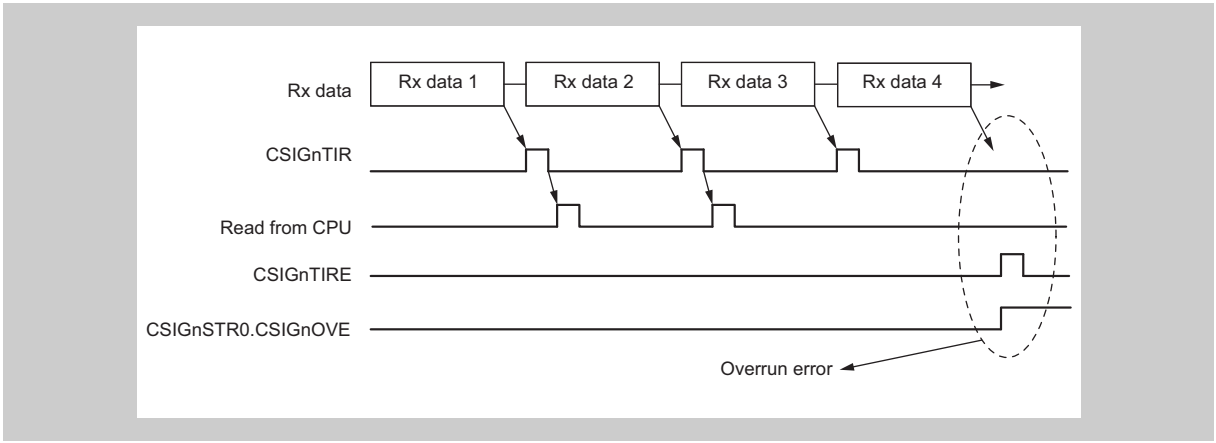


Figure 21-24 Overrun Error Detection - Example

Note Overrun errors from trying to overwrite received data in slave mode can be avoided through handshaking.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception register and is ready again.

Refer to section 21.3.9, Handshake Function for details.

21.4 CSIG Control Registers

The CSIGn is controlled and operated by means of the following registers.

Table 21-8 CSIGn Register Overview

Register name	Symbol	Address
Control register 0	CSIGnCTL0	<CSIGn_base1> + 00_H
Control register 1	CSIGnCTL1	<CSIGn_base0> + 10_H
Control register 2	CSIGnCTL2	<CSIGn_base0> + 14_H
Status register 0	CSIGnSTR0	<CSIGn_base1> + 04_H
Status clear register 0	CSIGnSTCR0	<CSIGn_base1> + 08_H
Receive-only mode control register 0	CSIGnBCTL0	<CSIGn_base1> + 80_H
Configuration register 0	CSIGnCFG0	<CSIGn_base0> + 1010_H
Transmission register 0 for word access	CSIGnTX0W	<CSIGn_base1> + 84_H
Transmission register 0 for half word access	CSIGnTX0H	<CSIGn_base1> + 88_H
Reception register 0	CSIGnRX0	<CSIGn_base1> + $8C_H$

<CSIGn_base> The base addresses <CSIGn_base0> and <CSIGn_base1> of the CSIGn are defined in section 21.1 under the key word "Register addresses".

(1) CSIGnCTL0 - CSIG Control Register 0

This register controls the operation clock and enables/disables transmission and reception.

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

Address <CSIGn_base1> + 00_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
CSIGn PWR	CSIGn TXE	CSIGn RXE	0	0	0	0	CSIGn MBS
R/W	R/W	R/W	R	R	R	R	R/W

Table 21-9 CSIGnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGnPWR	Controls operation clock: 0: Stop operation clock 1: Provide operation clock Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. No clock is provided to internal circuits. If CSIGnPWR is cleared during communication, ongoing communication is aborted. A restart of the communication is then required.
6	CSIGnTXE	Enables/disables transmission: 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception: 0: Reception disabled 1: Reception enabled
0	CSIGnMBS	Be sure to set this bit to 1 (the initial value is 0).

Caution 1. Do not modify CSIGnRXE or CSIGnTXE while CSIGnPWR = 0. However, both bits can be modified in the same write operation when setting CSIGnPWR = 1.

Caution 2. Do not modify CSIGnRXE or CSIGnTXE while a data transmission is pending or ongoing, i.e. if CSIGnSTR0.CSIGnTSF = 1.

Caution 3. Set the CSIGnMBS bit to 1 before activating CSIG (before or at the same time as CSIGnPWR is set to 1).

(2) CSIGnCTL1 - CSIG Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It enables/disables extended data length control, data consistency check, loop-back mode, and handshake function.

Access This register can be read/written in 32-bit units.

Address <CSIGn_base0> + 10_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIGn CKR	CSIGn SLIT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CSIGn EDLE	0	CSIGn DCS	0	CSIGn LBM	CSIGn SIT	CSIGn HSE	0
R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R

Caution Changing the contents of this register is only permitted when CSIGnCTL0.CSIGnPWR = 0.

Table 21-10 CSIGnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
17	CSIGnCKR	Selects the phase of the CSIGnTSCK clock signal. 0: Default level of CSIGnTSCK is high 1: Default level of CSIGnTSCK is low For the setting example, refer to the CSIGnDAP bit in table 21-15, CSIGnCFG0 Register Contents.
16	CSIGnSLIT	Selects the timing of interrupt CSIGnTIC: 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt is generated when CSIGnTX0W or CSIGnTX0H is free for further data. For details, refer to section 21.3.8 *chapter(1) CSIGnTIC (Communication Interrupt)*.
7	CSIGnEDLE	Enables/disables extended data length (EDL) mode: 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to section 21.3.5 *chapter(2) Data Length Greater than 16 Bits.*.
5	CSIGnDCS	Enables/disables data consistency check: 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to section 21.3.11 *chapter(1) Data Consistency Check*.

Table 21-10 CSIGnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
3	CSIGnLBM	Controls loop-back mode (LBM): 0: Loop-back mode deactivated 1: Loop-back mode activated For details, refer to section 21.3.10, Loop-Back Mode. Caution: Set the bit to 0 in slave mode.
2	CSIGnSIT	Selects interrupt delay mode: 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, refer to section 21.3.8, CSIG Interrupts.
1	CSIGnHSE	Enables/disables handshake mode: 0: Handshake function disabled 1: Handshake function enabled For details, refer to section 21.3.9, Handshake Function.

(3) CSIGnCTL2 - CSIG Control Register 2

This register selects the communication clock.

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

Address <CSIGn_base0> + 14_H

Initial value $E000_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGn PRS[2:0]			0	CSIGn BRS[11:0]											
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Changing the contents of this register is only permitted when CSIGnCTL0.CSIGnPWR = 0.

Table 21-11 CSIGnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIGnPRS [2:0]	Selects the value of the prescaler:
11 to 0	CSIGnBRS [11:0]	Selects the baud rate:

Selects the value of the prescaler:

CSIGn PRS2	CSIGn PRS1	CSIGn PRS0	Prescaler Output (PRSOUT)
0	0	0	PCLK (master mode)
0	0	1	PCLK/2 (master mode)
0	1	0	PCLK/4 (master mode)
0	1	1	PCLK/8 (master mode)
1	0	0	PCLK/16 (master mode)
1	0	1	PCLK/32 (master mode)
1	1	0	PCLK/64 (master mode)
1	1	1	External clock via CSIGnTSCK (slave mode)

Selects the baud rate:

CSIGnBRS[11:0]	Baud Rate at CSIGnTSCK
0	BRG is stopped
1	PCLK/($2^m \times 1 \times 2$)
2	PCLK/($2^m \times 2 \times 2$)
3	PCLK/($2^m \times 3 \times 2$)
4	PCLK/($2^m \times 4 \times 2$)
...	...
4095	PCLK/($2^m \times 4095 \times 2$)

m = CSIGnPRS[2:0] = 0 to 6
The setting is enabled in master mode.

(4) CSIGnSTR0 - CSIG Status Register 0

This register indicates the status of the CSIG.

Access This register can be read in 32-bit units.

However, the CSIGnDCE, CSIGnPE and CSIGnOVE bits are writable when CSIGnCTL0.PWR is 0.

Address <CSIGn_base1> + 04_H

Initial value $0000\ 0010_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CSIGn TSF	0	0	1	CSIGn DCE	0	CSIGn PE	CSIGn OVE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21-12 CSIGnSTR0 Register Contents (1/2)

Bit Position	Bit Name	Function
7	CSIGnTSF	Transfer status flag: 0: Idle state 1: Transmission is in progress or being prepared Conditions for setting and clearing the bit are as follows:

Master Mode	Set by	Cleared by
Transmit-only mode	Writing to the CSIGnTX0W or CSIGnTX0H register	Within 0.5 clock from the last edge of CSIGnTSCK
Transmit/Receive mode		
Receive-only mode	Reading from the CSIGnRX0 register	

Slave Mode	Set by	Cleared by
Transmit-only mode	Writing to the CSIGnTX0W or CSIGnTX0H register	Within 0.5 clock from the last edge of CSIGnTSCK
Transmit/Receive mode		
Receive-only mode	Input clock on the CSIGnTSCK pin	

Table 21-12 CSIGnSTR0 Register Contents (2/2)

Bit Position	Bit Name	Function
3	CSIGnDCE	Data consistency check error flag: 0: No data consistency error detected 1: Data consistency error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnDCEC. This bit is writable when CSIGnCTL0.CSIGnPWR is 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When setting to 1 due to data consistency error detection and clearing to 0 by CSIGnSTCR0.CSIGnDCEC occur simultaneously, setting to 1 due to data consistency error detection takes precedence over clearing.
1	CSIGnPE	Parity error flag: 0: No parity error detected 1: Parity error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnPEC. This bit is writable when CSIGnCTL0.CSIGnPWR is 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When setting to 1 due to parity error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnPEC occur simultaneously, setting to 1 due to parity error detection takes precedence over clearing.
0	CSIGnOVE	Overrun error flag: 0: No overrun error detected 1: Overrun error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnOVEC. This bit is writable when CSIGnCTL0.CSIGnPWR is 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence over clearing.

(5) CSIGnSTCR0 - CSIG Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access This register can be read/written in 16-bit units. The value is always read as 0000_H.

Address <CSIGn_base1> + 08_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	CSIGn DCEC	0	CSIGn PEC	CSIGn OVEC
R	R	R	R	R	R	R	R	R	R	R	R	W	R	W	W

Table 21-13 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
3	CSIGnDCEC	Controls the operation of clearing the data consistency check error flag (CSIGnSTR0.CSIGnDCE) 0: No operation. Read value is always 0. 1: Clear data consistency error flag (CSIGnSTR0.CSIGnDCE).
1	CSIGnPEC	Controls the operation of clearing the parity error flag (CSIGnSTR0.CSIGnPE). 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIGnSTR0.CSIGnPE).
0	CSIGnOVEC	Controls the operation of clearing the overrun error flag (CSIGnSTR0.CSIGnOVE). 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIGnSTR0.CSIGnOVE).

(6) CSIGnBCTL0 - CSIG Receive-Only Mode Control Register 0

This register enables/disables the data transfer in receive-only mode.

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

Address <CSIGn_base1> + 80_H

Initial value 01_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CSIGnSCE
R	R	R	R	R	R	R	R/W

Table 21-14 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
0	CSIGnSCE	Disables/enables next data reception start by reading CSIGnRX0: 0: Next reception disabled 1: Next reception enabled For details, refer to section 21.3.4 *chapter(2) Receive-Only Mode*.

Note Setting of the CSIGnSCE bit is ineffective in transmit-only and transmit/receive modes.

Caution To stop subsequent reception, the CSIGnSCE bit has to be controlled as follows.

- If CSIGnSLIT is 0, clear the bit before the last data reception (reading from the CSIGnRX0 register).
- If CSIGnSLIT is 1, clear the bit immediately after reception of all but the last of the data (reading from the CSIGnRX0 register) and at least one clock cycle of CSIGnTSCK before generation of the CSIGnTIR interrupt for completion of the last reception.

(7) CSIGnCFG0 - CSIG Configuration Register 0

This register configures the communication protocol - data length, parity, transfer direction, clock phase, and data phase.

Access This register can be read/written in 32-bit units.

Address <CSIGn_base0> + 1010_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	CSIGn PS[1:0]		CSIGn DLS[3:0]				0	0	0	0	0	CSIGn DIR	0	CSIGn DAP
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Caution Changing the contents of this register is only permitted when CSIGnCTL0.CSIGnPWR = 0.

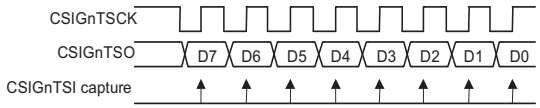
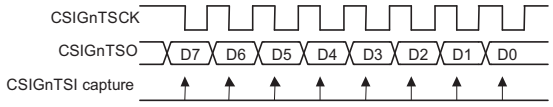
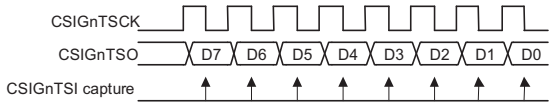
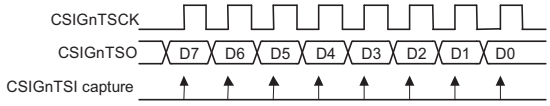
Table 21-15 CSIGnCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function
29 to 28	CSIGnPS [1:0]	Specifies parity: CSIGn PS1 / CSIGn PS0 / Transmission / Reception 0 / 0 / No parity transmitted / No parity judged. 0 / 1 / Add parity bit fixed to 0 / Parity bit is received but not judged. 1 / 0 / Add odd parity / Odd parity judged. 1 / 1 / Add even parity / Even parity judged.
27 to 24	CSIGnDLS [3:0]	Specifies data length: 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **Caution:** The extended data length (EDL) function has to be used to set data lengths from 1 to 6 bits (refer to section 21.3.5 *chapter(2) Data Length Greater than 16 Bits.* Transmitting twice consecutively with a data length of less than 7 bits is forbidden.
18	CSIGnDIR	Selects the orientation of transferred data: 0: Data is transmitted/received with MSB first 1: Data is transmitted/received with LSB first

Table 21-15 CSIGnCFG0 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>16</td><td>CSIGnDAP</td><td>Data phase selection bit

The timing of data transmission and reception according to the combination of CSIGnCTL1.CSIGnCKR and this bit is as follows.

• CSIGnCTL1.CSIGnCKR = 0
<table>
<tr><th>CSIGn DAP</th><th>Clock and Data Phase Selection</th></tr>
<tr><td>0</td><td> 
</td></tr>
<tr><td>1</td><td> 
</td></tr>
</table>
• CSIGnCTL1.CSIGnCKR = 1
<table>
<tr><th>CSIGn DAP</th><th>Clock and Data Phase Selection</th></tr>
<tr><td>0</td><td> 
</td></tr>
<tr><td>1</td><td> 
</td></tr>
</table></td></tr>
</table>

(8) CSIGnTX0W - CSIG Transmission Register 0 for Word Access

This register stores the transmission data. It has to be used when the extended data length function is enabled (CSIGnCTL1.SCIGnEDLE = 1).

Access This register can be read/written in 32-bit units.

Address <CSIGn_base1> + 84_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	CSIGn EDL	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnTX[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Writing to this register is forbidden when CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 0.

Table 21-16 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
29	CSIGnEDL	Specifies the extended data length configuration: 0: Normal operation 1: Extended data length activated The data is transmitted as 16-bit data. Caution: This bit can only be set if CSIGnCTL1.CSIGnEDLE = 1. It is automatically cleared if CSIGnCTL1.CSIGnEDLE is cleared.
15 to 0	CSIGn TX[15:0]	Data to be transmitted

(9) CSIGnTX0H - CSIG Transmission Register 0 for Half Word Access

This register stores the transmission data. It can be used when the extended data length function is disabled (CSIGnCTL1.CSIGnEDLE = 0).

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

Address <CSIGn_base1> + 88_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnTX[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Writing to this register is forbidden when CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 0.

Table 21-17 CSIGnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnTX [15:0]	Data to be transmitted

(10) CSIGnRX0 - CSIG reception register 0

This register stores the received data.

Access This register can be read in 16-bit units.

Address <CSIGn_base1> + $8C_H$

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Caution 1. This register is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0.

Caution 2. Reading from this register is forbidden when CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 0. Values read from the register while settings are as follows are undefined.

- CSIGnPWR = 0, CSIGnTXE = 1, CSIGnRXE = 1
- CSIGnPWR = 1, CSIGnTXE = 0, CSIGnRXE = 0

Caution 3. Reading from this register is possible when CSIGnCTL0.CSIGnPWR = 1

Caution 4. Writing to this register is possible when CSIGnCTL0.CSIGnPWR = 0.

Table 21-18 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX [15:0]	Received data

21.5 Operating Procedure Example

A transmit/receive example in master mode is described in this section. The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- MSB is transmitted first (CSIGnCFG0.CSIGnDIR = 0)
- CSIGnTIC interrupt is generated at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase: CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0
- The number of data packets is 10 (0 to 9)

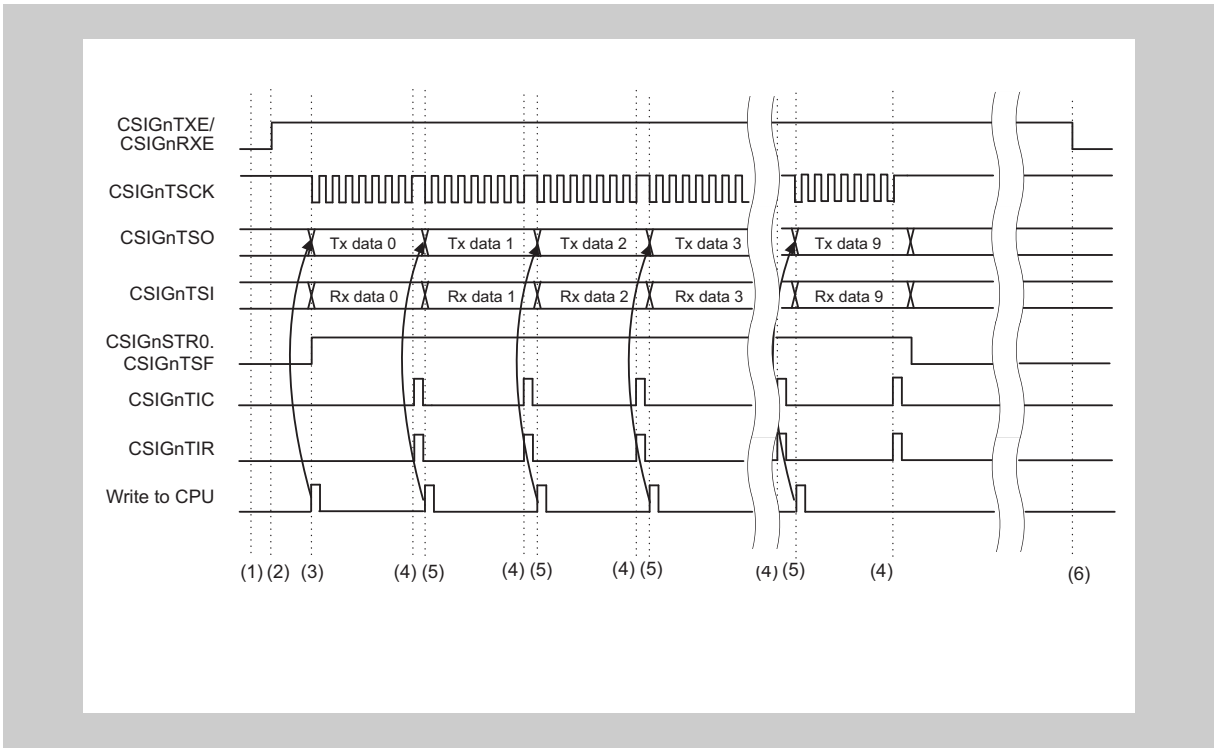


Figure 21-25 Communication in Master Mode

Procedure:

1. Configure each function in the CSIGnCTL1 register, communication speed in the CSIGnCTL2 register, and communication protocol in the CSIGnCFG0 register, respectively.
2. In the CSIGnCTL0 register, set bits CSIGnPWR = 1 (enabling the clock), CSIGnTXE = 1 (enabling transmission), and CSIGnRXE = 1 (enabling reception). The data output pin CSIGnTSO is now enabled.
3. Write the first packet of data to be sent to the transmission register (CSIGnTX0W or CSIGnTX0H). Transmission starts automatically.
4. After every packet that has been transmitted, the interrupts CSIGnTIC and CSIGnTIR are generated. CSIGnTIC indicates that the next packet can be written to CSIGnTX0W or CSIGnTX0H. CSIGnTIR indicates that the reception register CSIGnRX0 must be read.

   In this example, CPU write and DMA write are equivalent.
5. Write the next data to CSIGnTX0W or CSIGnTX0H.
6. No more write action is required after completion of packet 9.
7. To finally disable the transmit/receive operation, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE.

Section 22 Queue-Equipped Three-Line Serial Interface H (CSIHn)

This section contains a generic description of the queue-equipped three-line serial interface H (CSIHn).

22.1 CSIHn Features

Instances

This product has following number of instances of the queue-equipped three-line serial interface H.

Table 22-1 Instances of CSIHn

Queue-Equipped Three-Line Serial Interface H (CSIHn)	
Instances	2*1
Name	CSIH0, CSIH1*2

Note 1. The μPD70F3505A only has 1 instance.
Note 2. The μPD70F3505A does not support this.

Instances index n

Throughout this section, the individual instances of a queue-equipped three-line serial interface H is identified by the index “n” (n = 0, 1), for example, CSIHnCTL0 for the CSIHn control register 0.

Chip select index x

The queue-equipped three-line serial interface H has 4 (CSIH1) to 8 (CSIH0) chip select signals. Throughout this section, the individual chip select signals are identified by the index “x” (x = 0 to 3 (CSIH1), 0 to 7 (CSIH0)), thus a certain chip select signal is denoted as CSx.

The number of chip selects for each instance of CSIHn is given in the following table.

CSIHn Instance	Chip Select Signal
CSIH0	x = CS0 to CS7
CSIH1	x = CS0 to CS3

Register addresses

All CSIHn register addresses are given as address offsets to the individual base address <CSIHn_base>.

The base address <CSIHn_base> of each CSIHn is listed in the following table.

Table 22-2 Register Base Addresses <CSIHn_base>

CSIHn Instance	<CSIHn_base0> Address	<CSIHn_base1> Address
CSIH0	FF6C 0000$_H$	FFFF E000$_H$
CSIH1	FF6D 0000$_H$	FFFF E100$_H$

Clock supply

The following clock signal is input to CSIHn.

Table 22-3 CSIHn Clock Supply

CSIHn Instance	CSIHn Clock	Connected to
CSIH0, CSIH1	PCLK	Clock controller

Interrupts The queue-equipped three-line serial interface H can generate the following interrupt.

Table 22-4 CSIHn Interrupt Requests

CSIHn Signals	Function	Connected to
CSIHnTIC	Communication status interrupt	Interrupt controller INTCSIHnIC DMA, DTS
CSIHnTIR	Reception status interrupt	Interrupt controller INTCSIHnIR DMA, DTS
CSIHTnTRE	Communication error interrupt	Interrupt controller INTCSIHnRE
CSIHnTIJC	Job completion interrupt	Interrupt controller INTCSIHnIJC DMA, DTS

Internal signals The internal signal connections of the queue-equipped three-line serial interface H are listed in the following table.

Table 22-5 CSIHn Internal Signal Connections

CSIHn Signal	Function	Connected to
CSIHnBPCLK	Clock input for CSx prescaler	CSx clock generation CSIHnTSCKx

I/O signals The I/O signals of the queue-equipped three-line serial interface H are listed in the following table.

Table 22-6 CSIHn I/O Signals

CSIHn Signal	Function	Connected to
CSIHnTSCK	Serial clock signal	Port CSIHnSC
CSIHnTSI	Serial data input signal	Port CSIHnSI
CSIHnTSO	Serial data output signal	Port CSIHnSO
CSIHnTSHSG	Handshake signal (ready signal)	Port CSIHnRYI
	Handshake signal (busy signal)	Port CSIHnRYO
CSIH0TCSSn [7:0]	Chip select signals	Port CSIH0CS[7:0]
CSIH1TCSSn [3:0]		Port CSIH1CS[3:0]

Port Groups Use CSIHn pins from the groups listed below.

Table 22-7 Port Groups for CSIHn

CSIHn Channel	Pin Name	Group		
		1	2	
CSIH0	CSIH0SC	P1_6	P5_8	
	CSIH0SO	P1_5	P5_7	
	CSIH0SI	P1_4	P5_6	
	CSIH0RYI*1	-	P5_9	P5_1
	CSIH0RYO*2	-	P5_10	P5_2
	CSIH0CS0	P2_5	P5_1	P7_0
	CSIH0CS1	P2_6	P5_2	P7_1
	CSIH0CS2	P2_7	P5_3	P7_2
	CSIH0CS3	P3_0	P5_4	P7_3
	CSIH0CS4	-	-	P7_4
	CSIH0CS5	-	-	P7_5
	CSIH0CS6	-	-	P7_6
	CSIH0CS7	-	-	P7_7
CSIH1	CSIH1SC	P5_11		-
	CSIH1SO	P5_10		-
	CSIH1SI	P5_9		-
	CSIH1RYI	P5_4		-
	CSIH1RYO	P5_5		-
	CSIH1CS0	P5_1	P7_4	-
	CSIH1CS1	P5_2	P7_5	-
	CSIH1CS2	P5_3	P7_6	-
	CSIH1CS3	P5_4	P7_7	-

Note 1. When P5_1 is used instead of P5_9, it cannot be used for CSIH0CS0 at the same time.
Note 2. When P5_2 is used instead of P5_10, it cannot be used for CSIH0CS1 at the same time.

22.2 Functional Overview

Features summary

- Three-line serial synchronous data transfer
- Master mode and slave mode selectable
- Multiple slaves configuration plus RCB (recessive configuration for broadcasting) thanks to 4 to 8 configurable chip select output signals
- Built-in baud rate generator
- Maximum transmission speed
  - In master mode: 8 Mbps
  - In slave mode: 6.6 Mbps (but no more than PCLK/6 Mbps)
- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- EDL (extended data length) function for transferring data with more than 16 bits
- Three selectable transfer modes
  - Transmit-only mode
  - Receive-only mode
  - Transmit/receive mode
- Built-in handshake function
- Error detection (data consistency check, parity, time-out, overflow, overrun)
- Full support of job concept
- 128 words I/O buffer memory
- Memory mode selectable (FIFO, double-buffer, transmit-only buffer, direct access)
- Four different interrupt request signals (CSIHnTIC, CSIHnTIR, CSIHnTIRE, CSIHnTIJC)
- LBM (loop-back mode) function for self test

The block diagram shows the main components of the CSIHn.

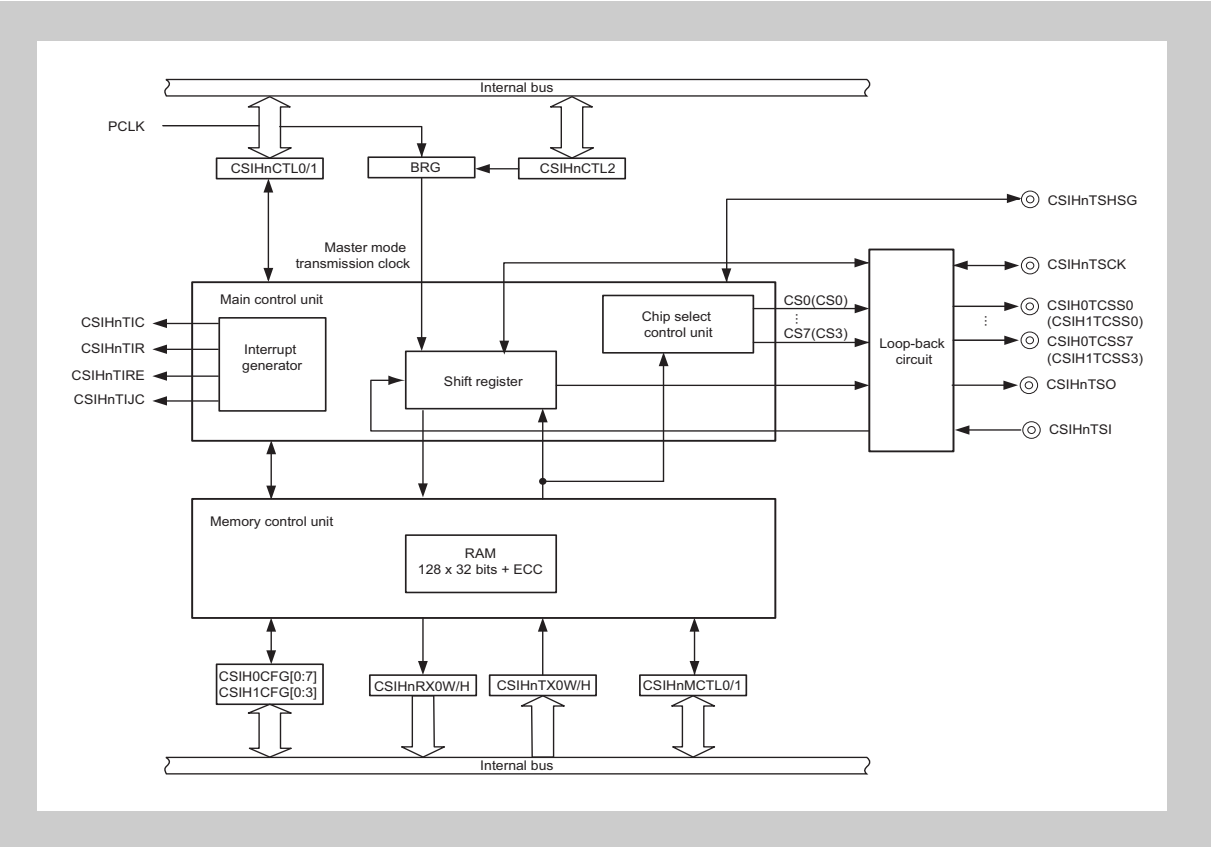


Figure 22-1 CSIHn Block Diagram

In master mode, the transmission clock CSIHnTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is received from an external source.

The built-in memory can be configured as FIFO, double-buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop-back circuit disconnects the CSIHn completely from the ports and supports internal self test.

Note This section describes the following modes.

- The “operating mode” separates between master and slave mode. In this context, only a master can control and communicate with several slaves (for details, see section 22.3.1, Operating Modes (Master/Slave)).
- The “job mode” is related to the Autosar job concept (for details, see section 22.3.3 (3) The Job Concept).
- The “memory mode” takes the various configurations of the associated buffer memory into account (for details, see section 22.3.6, CSIHn Buffer Memory).
- The “data transfer mode” specifies the kind of the communication - transmit only, receive only, or both (for details, see section 22.3.7, Data Transfer Modes).

22.3 Functional Description

The queue-equipped three-line serial interface H uses three signals for communication:

- Serial clock: CSIHnTSCK (output in master mode, input in slave mode)
- Data output signal: CSIHnTSO
- Data input signal: CSIHnTSI

Additional signals are available for external control and monitoring.

- Handshake signal: CSIHnTSHSG
- Chip-select signal: CSIH0TCSS[7:0], CSIH1TCSS[3:0]

Data transmission is bit-wise and serial and synchronous to the serial clock.

The most important registers for setting up the CSIHn are:

Register	Function
CSIHnCTL0	Enables/disables operation (PCLK) clock, data transmission, and data reception. Defines end-of-job behavior and enables/ disables buffering (bypass of the buffer).
CSIHnCTL1	Controls options such as interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master/slave mode. In master mode, selects the baud rate of internal generator.
CSIHnMCTL0	Selects memory mode and specifies time-out.
CSIHnMCTL1	Controls the memory in FIFO mode.
CSIHnMCTL2	Controls the memory in double-buffer and transmit-only-buffer modes.
CSIHnCFGx	Registers to configure the communication protocol for each chip select signal.

22.3.1 Operating Modes (Master/Slave)

For a particular CSIHn module, the master or slave mode determines the source of the serial clock.

(1) Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIHnTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to anything but 111_B. In master mode, CSIHnCTL2.CSIHnBRS becomes effective.

Chip select signals In master mode, one or several chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to address one or several of the slaves. Only a selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see section 22.3.3, Chip Selection (CS) Features.

Clock defaults The default level of CSIHnTSCK depends on the clock phase selection bit. It is high when CSIHnCFGx.CSIHnCKPx = 0, and is low when CSIHnCFGx.CSIHnCKPx = 1.

(2) Slave Mode

In slave mode, another device is the master of communications and thus provides the serial clock. Normal transmission or reception starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIHnCTL2.CSIHnPRS[2:0] to 111_B.

The transmission protocol settings in the CSIHnCFG0 register are still relevant in slave mode.

- CSIHnCFG0.CSIHnPS0[1:0]: Parity usage
- CSIHnCFG0.CSIHnDLS0[3:0]: Data length selection
- CSIHnCFGx.CSIHnDIRx: Data direction
- CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx: Clock and data phase

Note When using slave mode, minimize power consumption by clearing bits CSIHnCTL2.CSIHnBRS[11:0] and thus disabling the baud rate generator (BRG). If timeout errors are to be used, however, set the BRG to a value other than 0.

The example below shows the communication in master mode for 8 data bits, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

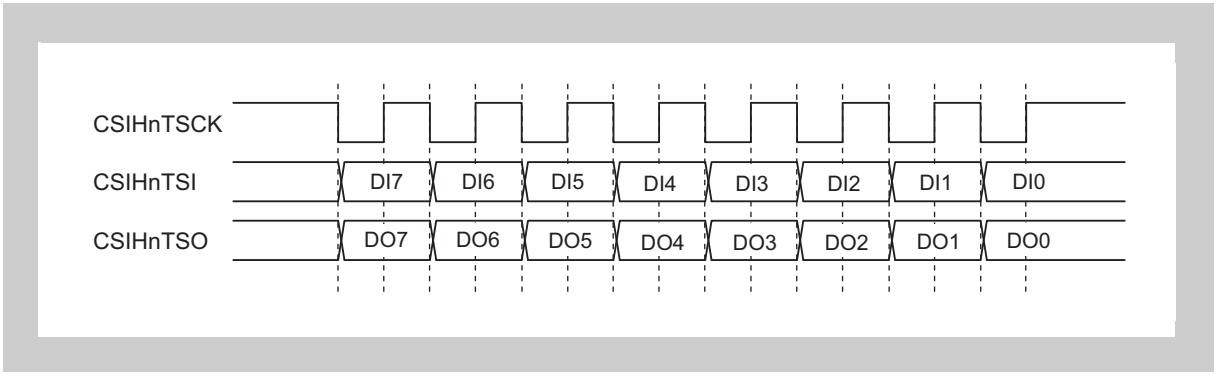


Figure 22-2 Transmit/Receive in Master Mode

Note In slave mode, the signal on CSIHnTSCK in the above figure acts as the input clock.

22.3.2 Master/Slave Connections

(1) One Master and One Slave

The following figure illustrates the connections between one master and one slave.

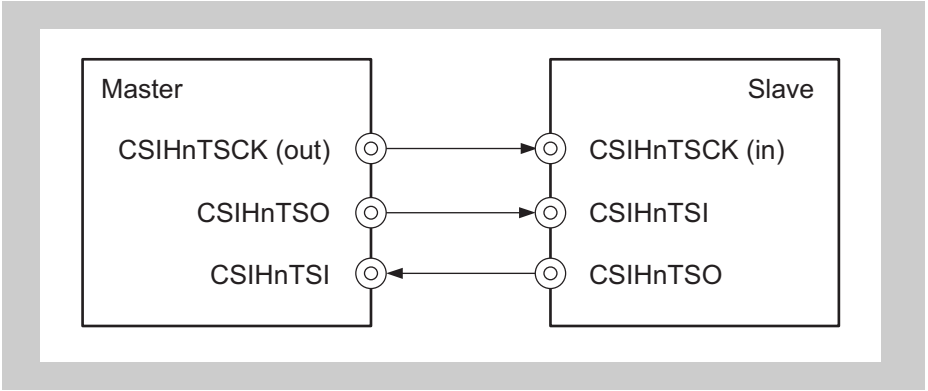


Figure 22-3 Direct Master/Slave Connection

(2) One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master provides one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input of the slave.

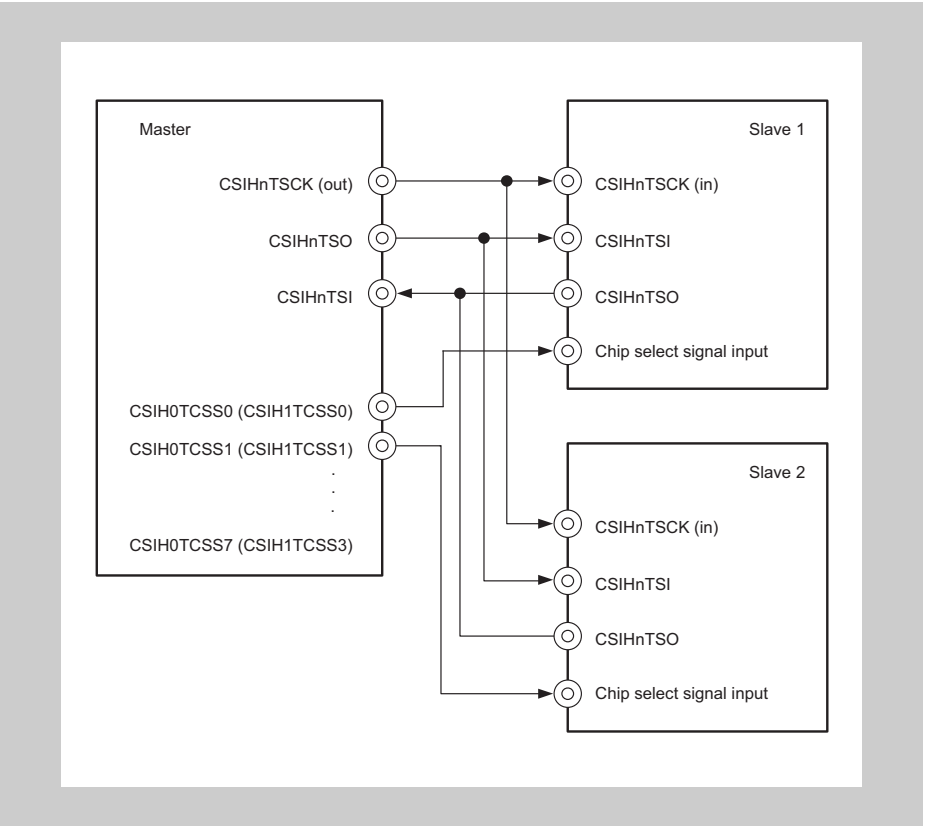


Figure 22-4 Master to Multiple Slaves Connection

By default, the chip select level is active low. That means, a CSIHn slave is selected (enabled) when its chip select signal has low level. However, to adapt the chip select signal (CS) to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, its output CSIHnTSO is set to input mode in order to avoid interference with the output of another slave that was selected.

22.3.3 Chip Selection (CS) Features

The chip select signals CSIHnTCSSx can be used by the master to select one or several slaves for communication.

(1) Configuration Registers

The parameters for each chip select signal CSIHnTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.
- Transfer direction: MSB first or LSB first.
- Parity usage: Odd, even, fixed to 0, or none.
- Clock phase and data phase.

Additional parameters for each chip select and only available in master mode are:

- Prescaler selection of the baud rate generator separately for each chip select
- Chip select priority: Separates between “dominant” and “recessive” chip select signals. The priority applies if two or more chip selects with different configurations are simultaneously activated for message broadcasting. In this case, the configuration that is set as dominant is used.

  The principle is also called “recessive configuration for broadcasting (RCB)”.

Caution It is forbidden to specify several chip select signals as dominant with different configurations unless all dominant chip selects have the same configuration.

- Chip select timing
  - Setup time T_{setup}: The time from setting the chip select (CSx) signal active to starting data output.
  - Inter-data time T_{inter}: The time between data while the same chip select (CSx) signal is active.
  - Hold time T_{hold}: Hold time of the CSx active level before changing the chip select (CSx).
  - Idle time T_{idle}: Inactive time after terminating a CSx signal or after every data transfer to the same CSx.

The various chip select (CSx) timings are illustrated in the following figure. The example shows the default - the CS1 and CS2 signals are active low. But the active level can be specified individually for each chip select (CSx).

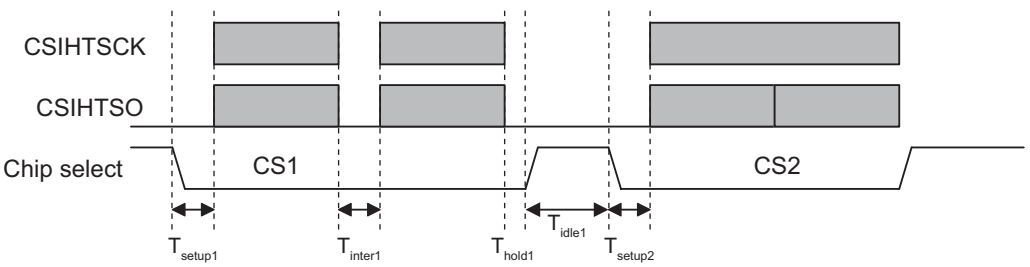


Figure 22-5 Chip Select Timings

Note that each chip select (CSx) can have a different value for setup, inter-data gap, hold, and idle periods.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIHnTX0W.CSIHnCS[7:0].

The reception register CSIHnRX0W.CSIHnCS[7:0] indicates the chip select signal associated with the received data.

(2) Chip Select (CS) Example

The following figure shows an example of two consecutive transmissions.

The first communication uses CS0 to address one single slave. The second communication (that communicates in communication setting at the dominant side) enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: lower priority”, the priority of CS1 to “dominant: higher priority”.

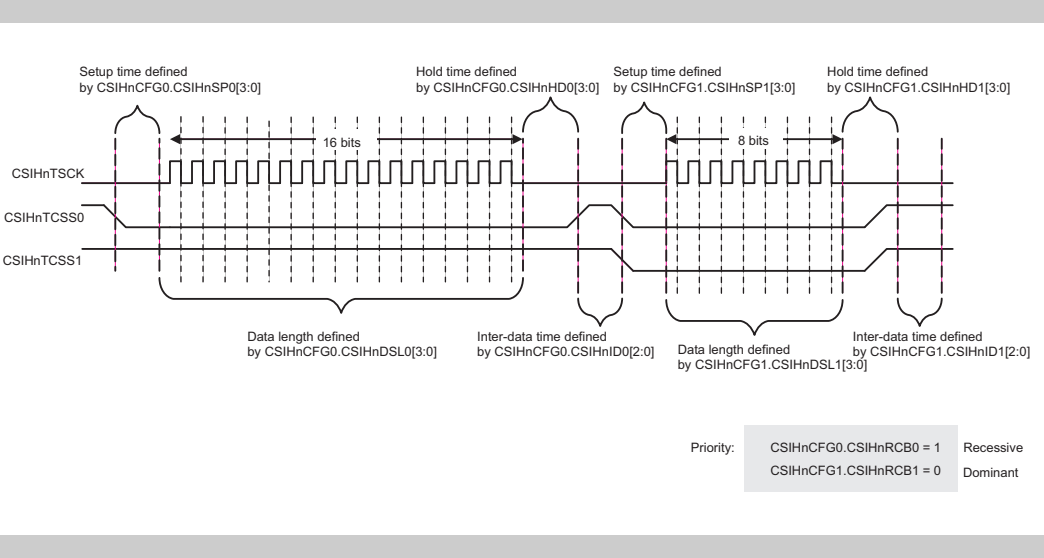


Figure 22-6 Chip Select and RCB Example

(3) The Job Concept

In terms of CSIHn, a job consists of a number of data that are transferred.

Job mode enable

Job mode is only enabled in master mode.

The job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIHn is disabled by CSIHnCTL0.CSIHnPWR = 0.

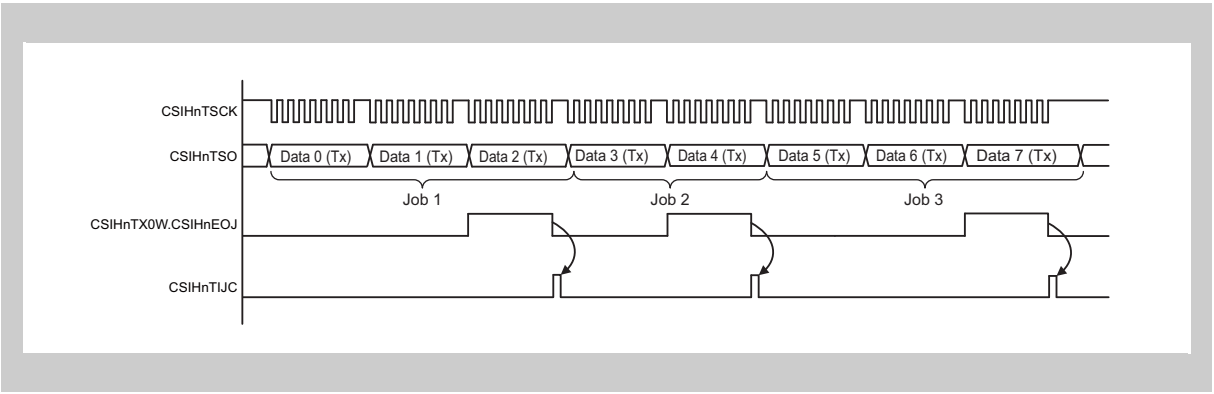


Figure 22-7 Job Examples

A job ends when a data with the end-of-job bit set, i.e. with CSIHnTX0W.CSIHnEOJ = 1.

A communication stop can be specified to occur after a job has finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until a data is sent, for which the CSIHnEOJ bit was set. After this data is sent, the communication is stopped and the end-of job-interrupt CSIHnTIJC is generated.

22.3.4 Chip Select Timing Details

(1) Changing the Clock Phase

When CSIHnCTL0.CSIHnPWR = 0, the serial clock level specified by CSIHnCFGx.CSIHnCKPx may be changed during an idle time. The minimum value of an idle time is a half period (0.5 SCK) of serial clock (CSIHnTSCK).

If the idle time is set to 0.5 serial clock periods (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data are sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one period of CSIHnTSCK.

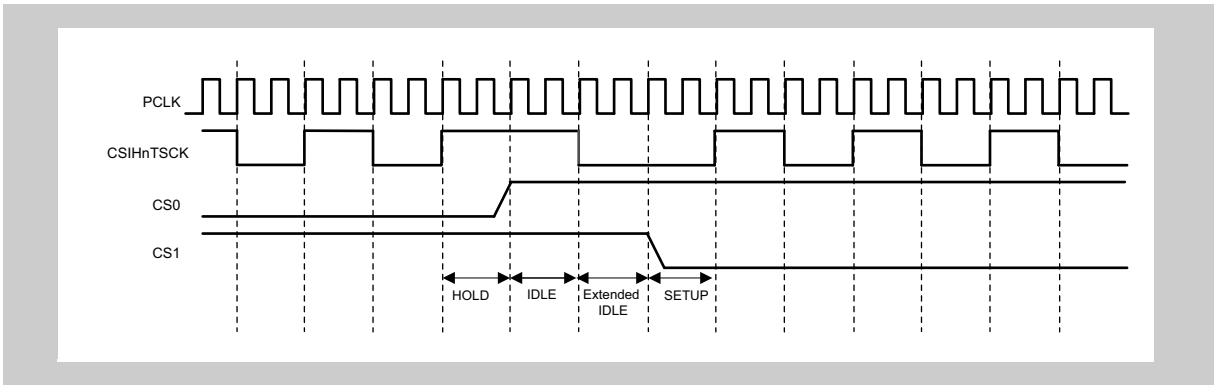


Figure 22-8 Data Phase Timing with PCLK/4, HOLD = SETUP = 0 (0.5 SCK), IDLE = 0 (0.5 SCK), CKP0 = 0 (CS0) → CKP1 = 1 (CS1)

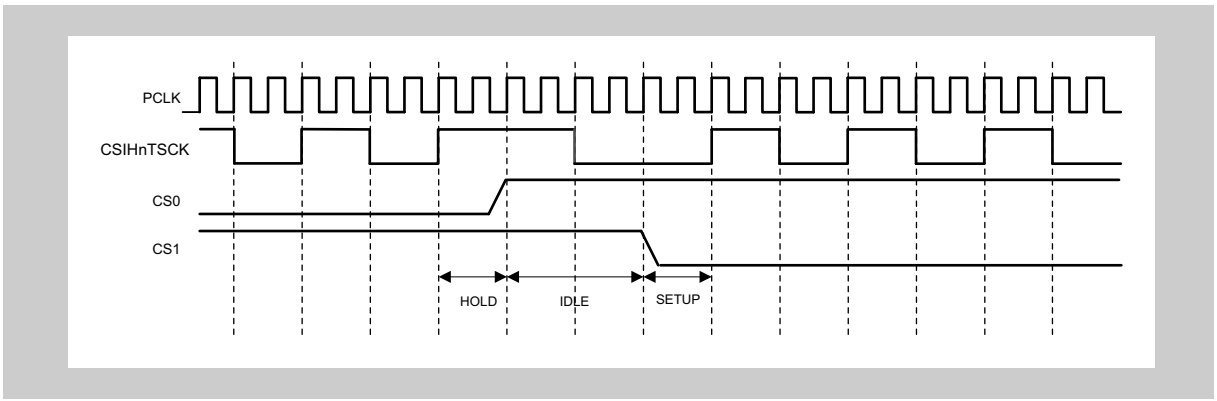


Figure 22-9 Data Phase Timing with PCLK/4, HOLD = SETUP = 0 (0.5 SCK), IDLE = 1 (1 SCK), CKP0 = 0 (CS0) → CKP1 = 1 (CS1)

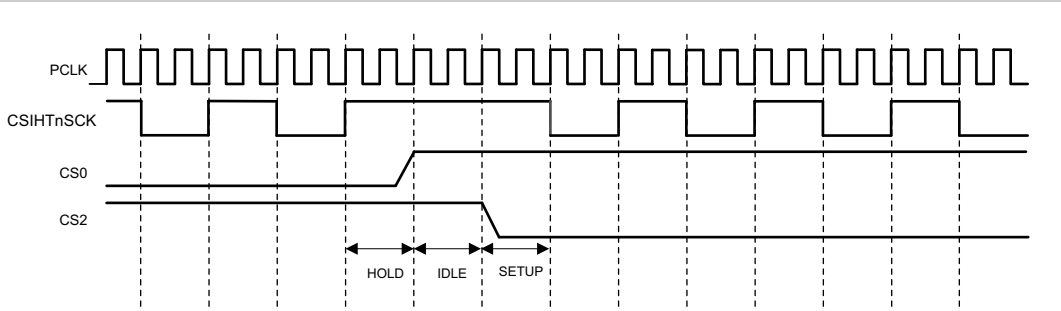


Figure 22-10 Data Phase Timing with PCLK/4, HOLD = SETUP = 0 (0.5 SCK), IDLE = 0 (0.5 SCK), CKP0 = 0 (CS0) → CKP2 = 0 (CS2)

(2) Changing the Data Phase

The bit CSIHnCFGx.CSIHnDAPx defines the phase of the data bits relative to the clock.

If CSIHnCFGx.CSIHnDAPx = 0, the serial clock CSIHnTSCK holds its level after the last bit of a data is transferred.

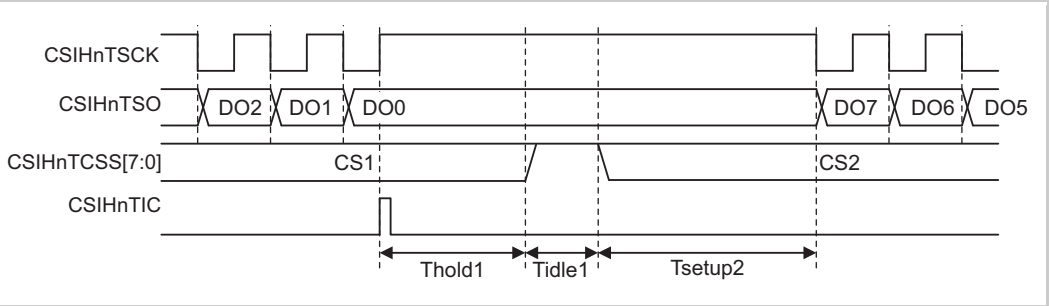


Figure 22-11 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

If the normal clock phase changes between two consecutive chip selects, the serial clock CSIHnTSCK changes its level after the last bit of the first data is transferred.

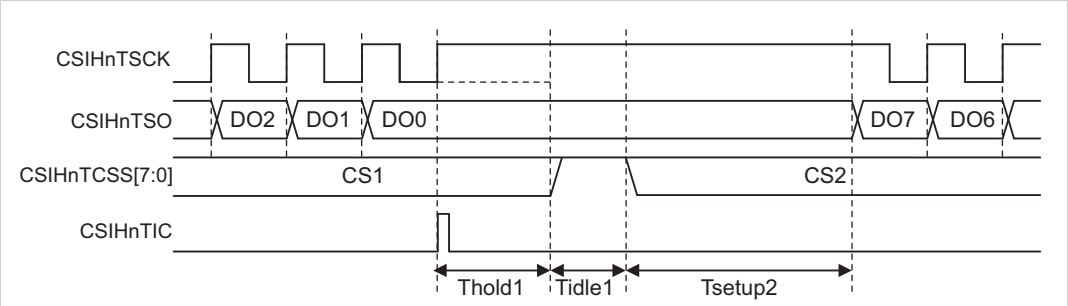


Figure 22-12 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 1 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

Note that if CSIHnCFGx.CSIHnIDx[2:0] = 0 (T_{idle1} = 0.5 CSIHnTSCK periods), the minimum idle time of one CSIHnTSCK period is automatically inserted.

22.3.5 Serial Clock Selection

In master mode, the transmission baud rate is selectable using

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnCTL2.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnPSCLx[1:0]

While the settings in the CSIHnCTL2 register determine the transmission base clock CSIHnBPCLK, a chip select dedicated prescaler, controlled by CSIHnCFGx.CSIHnPSCLx[1:0], allows to generate different baud rates for different chip select signals.

The following figure shows a block diagram of the baud rate generator.

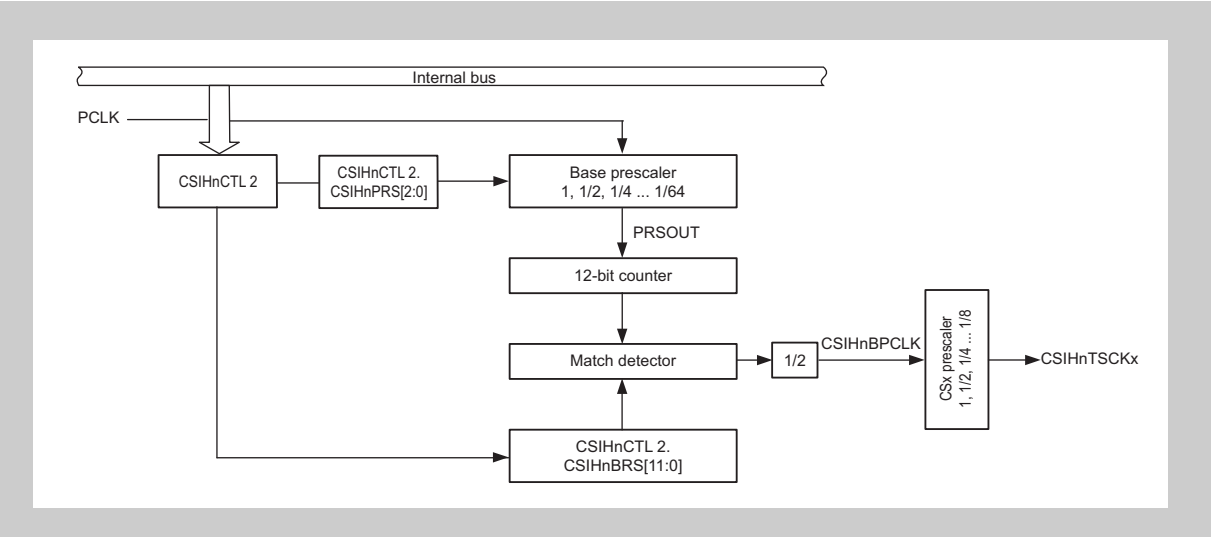


Figure 22-13 Baud Rate Generator Block Diagram

Clearing CSIHnCTL2.CSIHnBRS[11:0] disables the baud rate generator, and thus all CSIHnTSCKx are stopped.

Baud rate calculation

The baud rate is calculated as:

$$CSIHnTSCKx = PCLK/(2^m \times k \times 2 \times 2^j)$$

where

m = CSIHnCTL2.CSIHnPRS[2:0] = 0 to 6

k = CSIHnCTL2.CSIHnBRS[11:0] = 1 to 4095

j = CSIHnCFGx.CSIHnPSCLx[1:0] = 0 to 3

Baud rate upper/ lower limits

When setting the baud rate, note the following points.

- The baud rate must be no greater than 8.0 Mbps in master mode.
- The baud rate must be no greater than 6.6 Mbps in slave mode (and no greater than PCLK/6 Mbps). For operation with an external master, check that the baud rate from the master is in this range.
- Minimum baud rate in both modes is PCLK/524160.

22.3.6 CSIHn Buffer Memory

The CSIHn has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word is comprised of 32-bit data plus 7-bit ECC.

The following configurations are available:

Mode	CSIHnCTL0. CSIHnMBS	CSIHnMCTL0. CSIHnMMS[1:0]
FIFO mode	0	00_B
Double-buffer mode		01_B
Transmit-only buffer mode		10_B
Direct access mode	1	X

(1) FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided the FIFO is not full.

Data to be transmitted is stored to the FIFO memory. Transmission and reception occur simultaneously - one bit is sent, one bit is received. That means, received data overwrites the transmitted data in the FIFO.

The CSIHn automatically updates the respective FIFO memory pointers when a data package is committed, sent, or received.

Pointer Description	Control Bits*	Range
Number of untransmitted data	CSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of receivable words	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address of data to be transmitted	CSIHnMRWP0.CSIHnTRWA[6:0]	0 to 127
Address of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0 to 127

Note: * Pointers are automatically incremented after each round of reading or writing. The CSIHn status register contains two FIFO status flags.

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

Bit CSIHnSTCR0.CSIHnPCT must be set for entry to this mode. Setting this bit resets all FIFO pointers and flags.

(2) Double-Buffer Mode

In this mode, the memory is divided into two parts of equal size - this means 64 words for transmit data and 64 words for received data. In double-buffer mode, the respective buffer pointers indicate:

Pointer Description	Pointer*	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0 to 63
Address of data read from CSIHnRX0W/H	CSIHnMRWP0.CSIHnRRA[6:0]	0 to 63

Note: * Both pointers are automatically incremented after each read/write.

(3) Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the buffer pointer indicates:

Pointer Description	Pointer	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0 to 127

(4) Direct Access Mode

In direct access mode, the CSIHn memory is completely bypassed.

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

22.3.7 Data Transfer Modes

(1) Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode.

Master mode

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of double-buffer or transmit-only buffer mode, transmission starts when bit CSIHnMCTL2.CSIHnBTST is set.

Slave mode

- In slave mode, transmission starts on input of the serial clock after the above setting is made.

(2) Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIHn in receive-only mode.

Master mode

In master mode, the start of reception depends on the memory mode.

- In case of FIFO or direct access mode, reception starts when dummy data is written in the CSIHnTX0W or CSIHnTX0H register.
- In case of double-buffer mode, reception starts when bit CSIHnMCTL2.CSIHnBTST is set.

Slave mode

- In slave mode, reception starts as soon as the serial clock CSIHnTSCK from the master is received. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

(3) Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIHn in transmit/receive mode.

The start of the communication (transmission and reception) depends on the memory mode.

Master mode

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of double-buffer or transmit-only buffer mode, communication starts when bit CSIHnMCTL2.CSIHnBTST is set.

Slave mode

- In slave mode, transmission starts on input of the serial clock after the above setting is made.

(4) Summary

The following table provides a summary. It shows how the data transfer is started in the various memory, operating, and transfer modes.

Table 22-8 Start of Data Transfer

Memory and Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
FIFO, direct access	Master	Writing to CSIHnTX0W or CSIHnTX0H register	Writing to CSIHnTX0W or CSIHnTX0H register
	Slave		A clock signal is incoming from the master
Transmit-only buffer, double-buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	
	Slave	CSIHnMCTL2.CSIHnBTST = 1 and a clock signal is incoming from the master	

Caution Be sure to set the CSIHnTX0W register before starting data transfer in memory mode, operation mode, or transfer mode (except in reception-only slave operation). For details, refer to section 22 (11) CSIHnTX0W — CSIHn Transmission Register 0 for Word Access.

22.3.8 Data Length Selection

(1) Data Length between 7 and 16 Bits

The length of a data is selectable for each chip select signal from 7 to 16 bits using CSIHnCFGx.CSIHnDLSx[3:0]. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000_B)

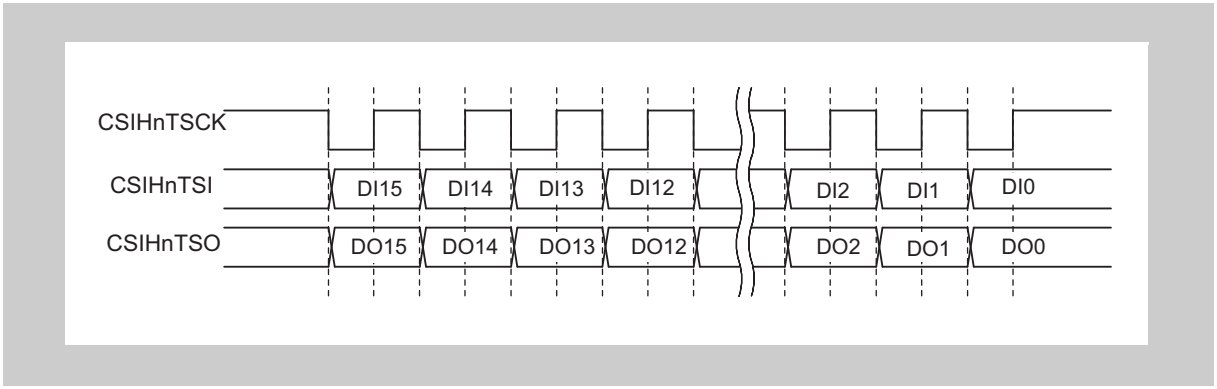


Figure 22-14 16-Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B)

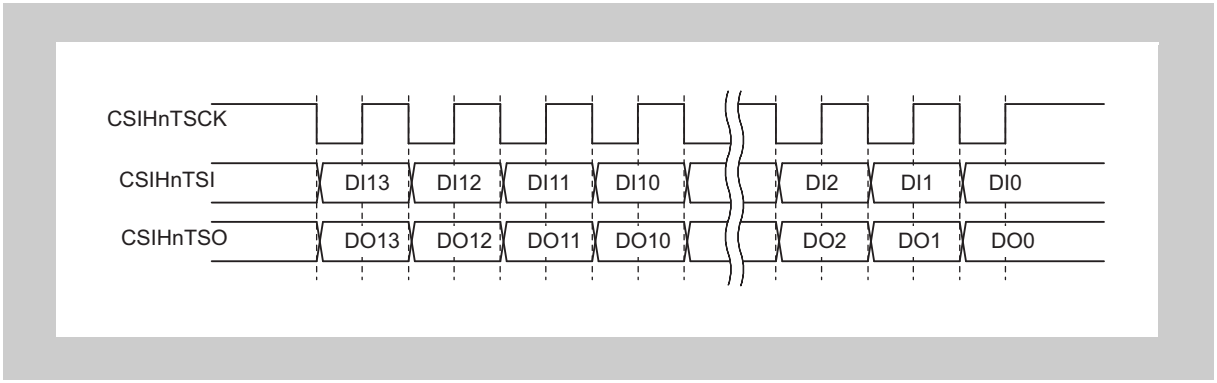


Figure 22-15 14-Bit Data Length, MSB First

(2) Data Length Greater than 16 Bits

If the length of the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

EDL is enabled by setting bit CSIHnCTL1.CSIHnEDLE to 1.

EDL works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The remainder defines the “data length” that has to be specified in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFG0.CSIHnDLS[3:0] bits.
- The transfer is complete after a block with the specified data length (the remainder - with CSIHnTX0W.CSIHnEDL = 0) has been sent.

Example Example for sending 40-bit data, for example the string $123456789A_H$

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIHnCFGx.CSIHnDLSx[3:0] = 8_D
- To send the string $123456789A_H$ with MSB first, write the following sequence to CSIHnTX0W.
  - $2000\ 1234_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $2000\ 5678_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $0000\ 009A_H$ (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

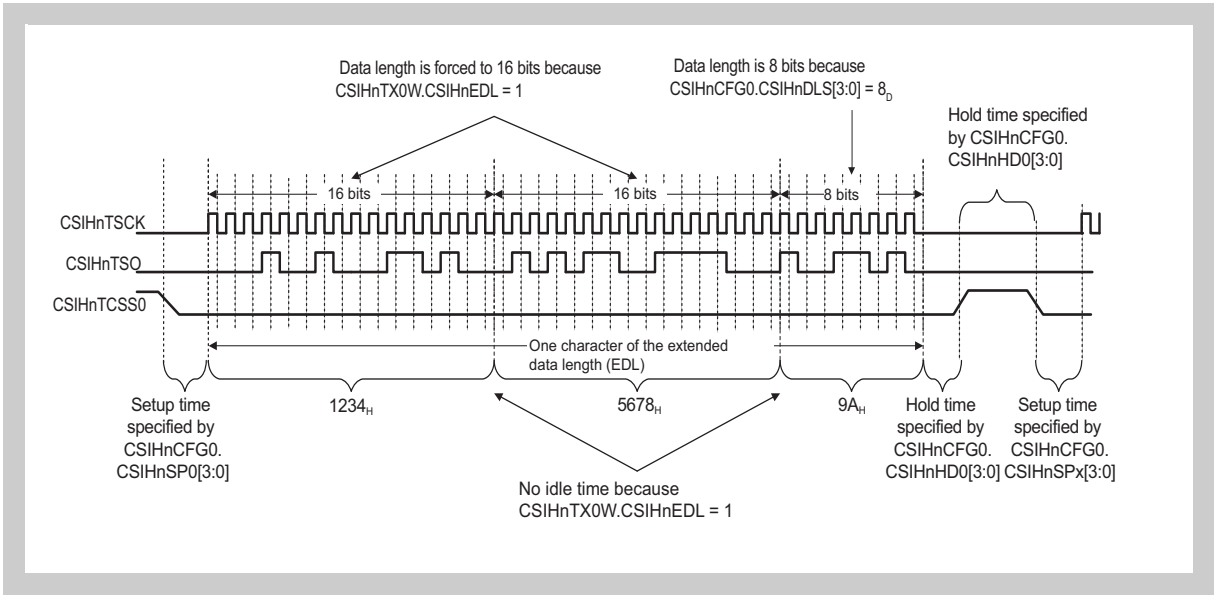


Figure 22-16 EDL Timing Diagram

Note 1. Data lengths settings less than 7 are only permitted in combination with EDL mode.

Note 2. It is not possible to send two consecutive data with a data length of less than 7 bits.

Note 3. If parity is enabled, the parity bit is added after the last bit.

Note 4. To consider the data direction, pay attention to the following example.

Data to be sent: 123456_H

MSB first:
Set CSIHnCFGx.CSIHnDIR = 0
Write CSIHnTX0W = 2000 1234_H (EDL bit = 1)
Write CSIHnTX0W = 0000 0056_H (EDL bit = 0)

LSB first:
Set CSIHnCFGx.CSIHnDIR = 1
Write CSIHnTX0W = 2000 3456_H (EDL bit = 1)
Write CSIHnTX0W = 0000 0012_H (EDL bit = 0)

Note 5. Using this in slave mode is prohibited.

22.3.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show the communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

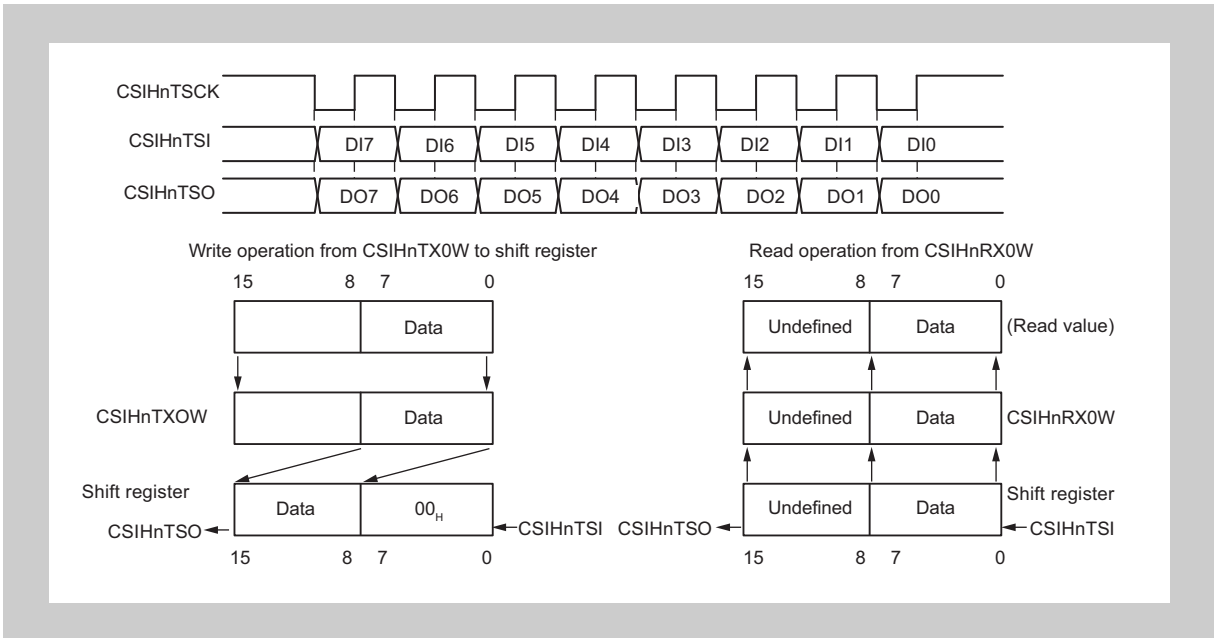


Figure 22-17 Serial Data Direction Select Function — MSB First (CSIHnDIRx = 0)

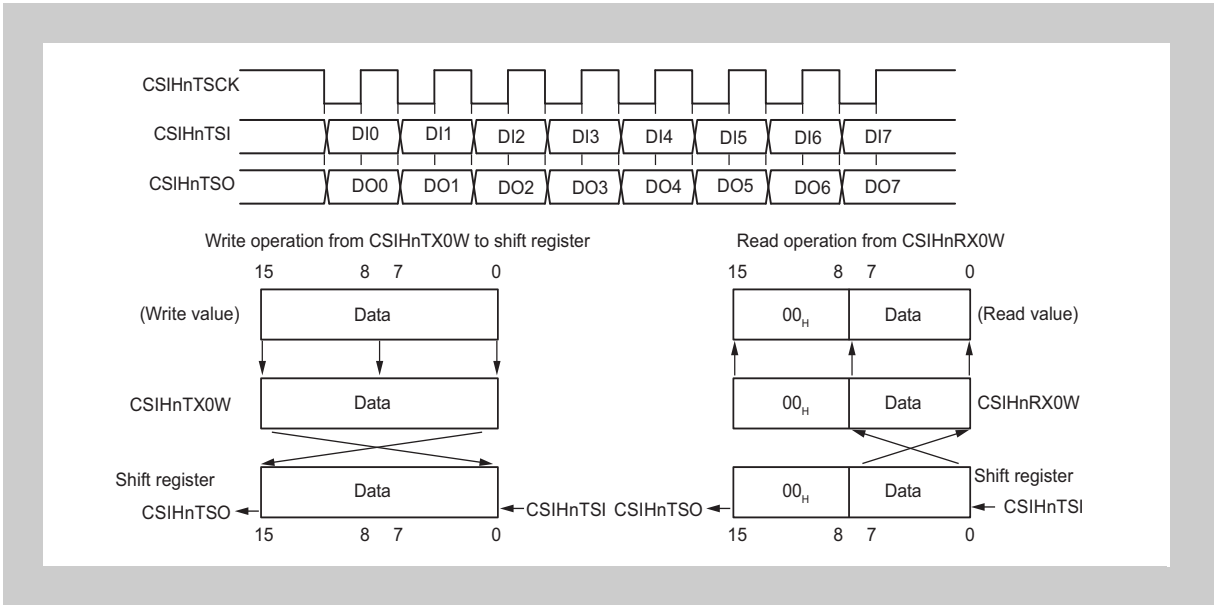


Figure 22-18 Serial Data Direction Select Function — LSB First (CSIHnDIRx = 1)

22.3.10 CSIHn Interrupt Requests

CSIHn can generate the following interrupt requests:

- CSIHnTIC (communication interrupt)
- CSIHnTIR (communication interrupt)
- CSIHnTIRE (error interrupt)
- CSIHnTIJC (job completion interrupt)

(1) CSIHnTIC (Communication Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the following conditions.

Table 22-9 CSIHnTIC Interrupt Generation

Memory Mode	Master and Slave	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	This interrupt occurs when transmission data is about to be missing in the FIFO, indicating to the application that new data should be added. CSIHnTIC is generated, if the number of data to be sent remaining in the FIFO CSIHnSTR0.CSIHnSPF[7:0] equals CSIHnMCTL1.CSIHnFES[6:0]. However, if a job is aborted* while job mode is enabled (CSIHnCTL1.CSIHnJE = 1), a CSIHnTIJC interrupt will be generated instead of a CSIHnTIC interrupt.	
Transmit-only buffer, double-buffer	End of communication (as specified in CSIHnMCTL2.CSIHnND[7:0]).	End of communication, (as specified in register CSIHnMCTL2) or data with CSIHnTX0W.CSIHnCIRE = 1 was sent. However, if data with CSIHnTX0W.CSIHnCIRE = 1 and job abortion* is sent, a CSIHnTIJC interrupt will be generated instead of a CSIHnTIC interrupt.
Direct access	Happens after every data transfer.	Happens after every data transfer, except when the communication was aborted*.

Note: * Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

CSIHnTIC in direct access mode

The following example shows the CSIHnTIC behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access memory mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

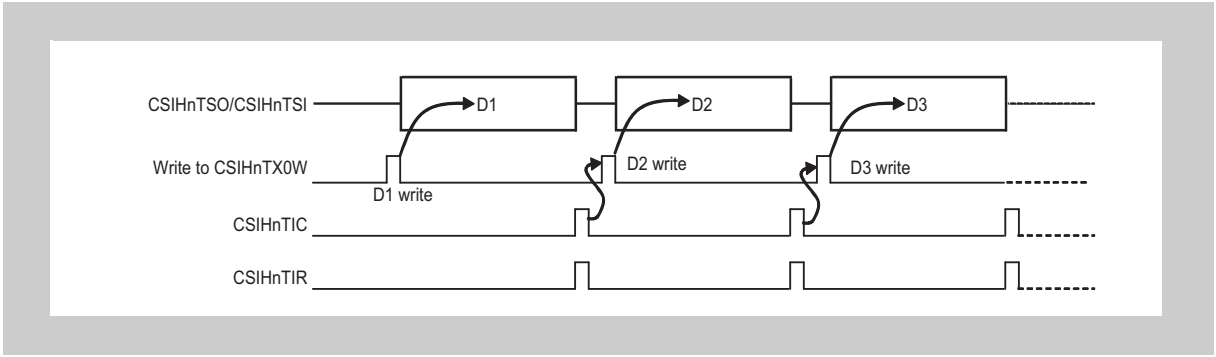


Figure 22-19 Generation of CSIHnTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTX0W.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then CSIHnTIC is replaced by the job completion interrupt CSIHnTIJC.

CSIHnTIC can also be set up to occur as soon as the CSIHnTX0W register is free for the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

Note This mode allows faster data transfer but is only available in direct access memory mode.

The effect is illustrated in the figure below.

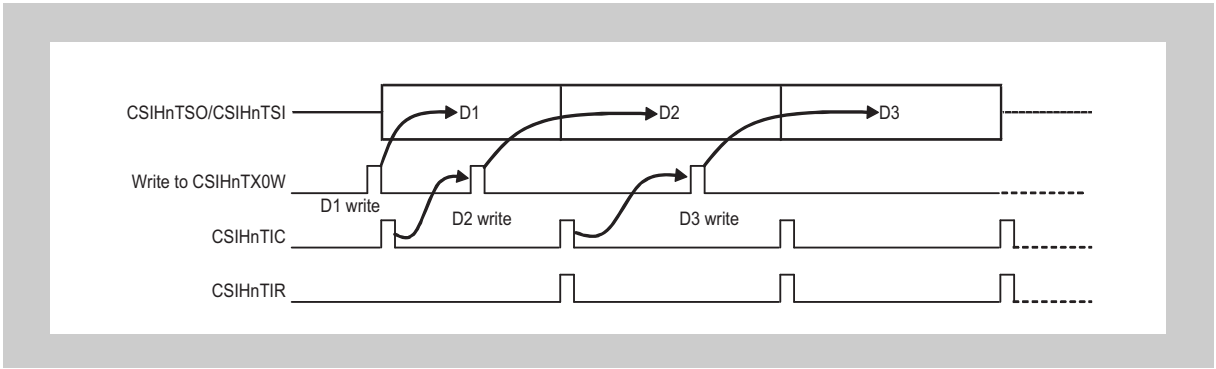


Figure 22-20 Immediate Generation of CSIHnTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

CSIHnTIC in FIFO mode

The following example shows the CSIHnTIC behavior in FIFO mode.

The following example assumes:

- Master mode
- FIFO memory mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

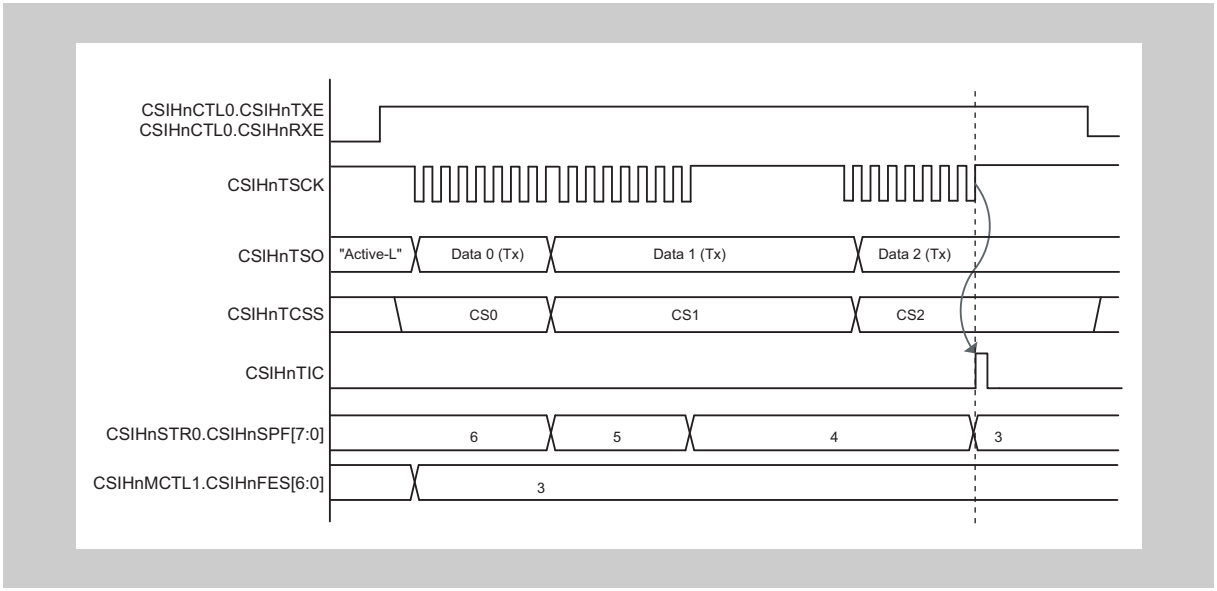


Figure 22-21 Generation of CSIHnTIC in FIFO Memory Mode

The condition for “FIFO empty” is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, minimum stock is 3 data.

CSIHnSTR0.CSIHnSPF[7:0] indicates the number of untransmitted data. When both match, the interrupt CSIHnTIC occurs.

CSIHnTIC in job mode

The following example shows the CSIHnTIC behavior in job mode.

The following example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

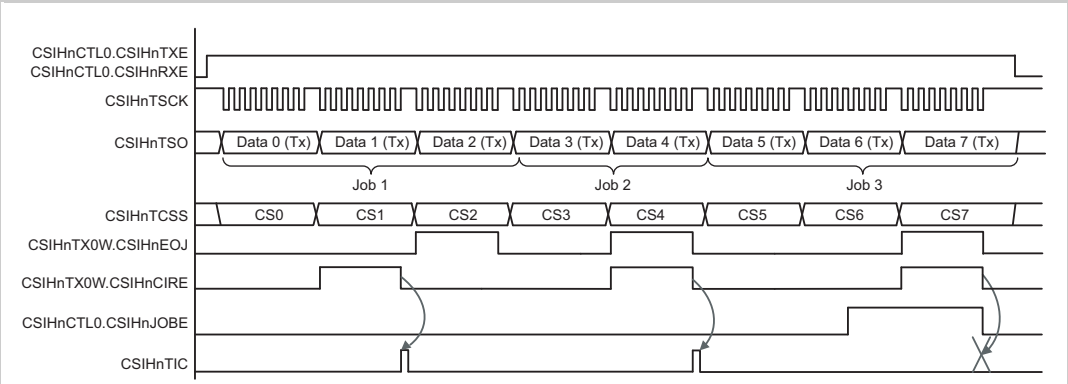


Figure 22-22 Generation of CSIHnTIC in Job Mode

The rules for generating CSIHnTIC in job mode are:

Table 22-10 Generation of CSIHnTIC in Job Mode (CSIHnCTL1.CSIHnJE = 1) (1/2)

Memory Mode	CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	CSIHnTIC
FIFO mode	0	0	Generated
	0	1	Generated
	1	0	CSIHnCTL0.CSIHnJOBE = 0: Generated
			CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt CSIHnTIJC
	1	1	CSIHnCTL0.CSIHnJOBE = 0: Generated
			CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt CSIHnTIJC
Double-buffer mode, transmit-only buffer mode	0	0	Not generated
	0	1	Generated
	1	0	CSIHnCTL0.CSIHnJOBE= 0: Not generated
			CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt CSIHnTIJC
	1	1	CSIHnCTL0.CSIHnJOBE = 0: generated
			CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt CSIHnTIJC

Table 22-10 Generation of CSIHnTIC in Job Mode (CSIHnCTL1.CSIHnJE = 1) (2/2)

Memory Mode	CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	CSIHnTIC
Direct access mode	0	-	Generated
	1	-	CSIHnCTL0.CSIHnJOBE = 0: Generated
			CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt CSIHnTIJC

(2) CSIHnTIR (Reception Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the following conditions.

Table 22-11 CSIHnTIR Interrupt Generation

Memory Mode	Master and Slave	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt occurs when the FIFO buffer is almost full with received data, indicating to the application that the FIFO must be emptied. CSIHnTIR is generated, if (128 - CSIHnSTR0.CSIHnSRP[7:0]) equals CSIHnMCTL1.CSIHnFFS[6:0].	
Double-buffer	Occurs after communication is ended (specified by CSIHnMCTL2. CSIHnND[7:0]) and when CSIHnCTL0.CSIHnRXE = 1.	Occurs after every received data when CSIHnCTL0.CSIHnRXE = 1.
Transmit-only buffer/ direct access	Occurs after every received data when CSIHnCTL0.CSIHnRXE = 1	

In transmit-only or double-buffer mode, this interrupt is generated in receive-only and transmit/receive mode after each data has been received.

CSIHnTIR in direct access mode

The following example shows the CSIHnTIR behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

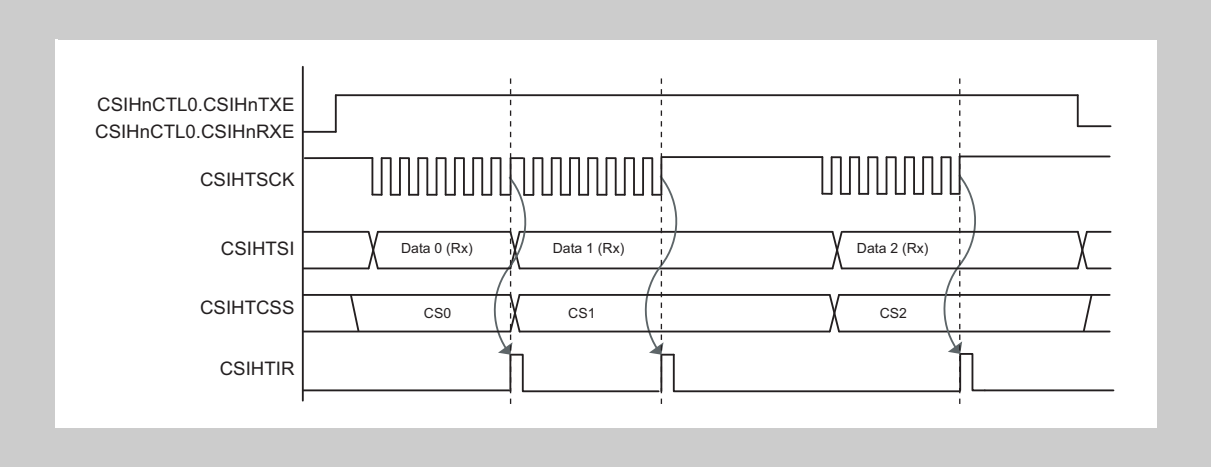


Figure 22-23 Generation of CSIHnTIR in Direct Access Memory Mode

CSIHnTIR in double-buffer mode

The following example shows the CSIHnTIR behavior in double-buffer mode.

The following example assumes:

- Master mode
- Transmit-only-buffer or double-buffer mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

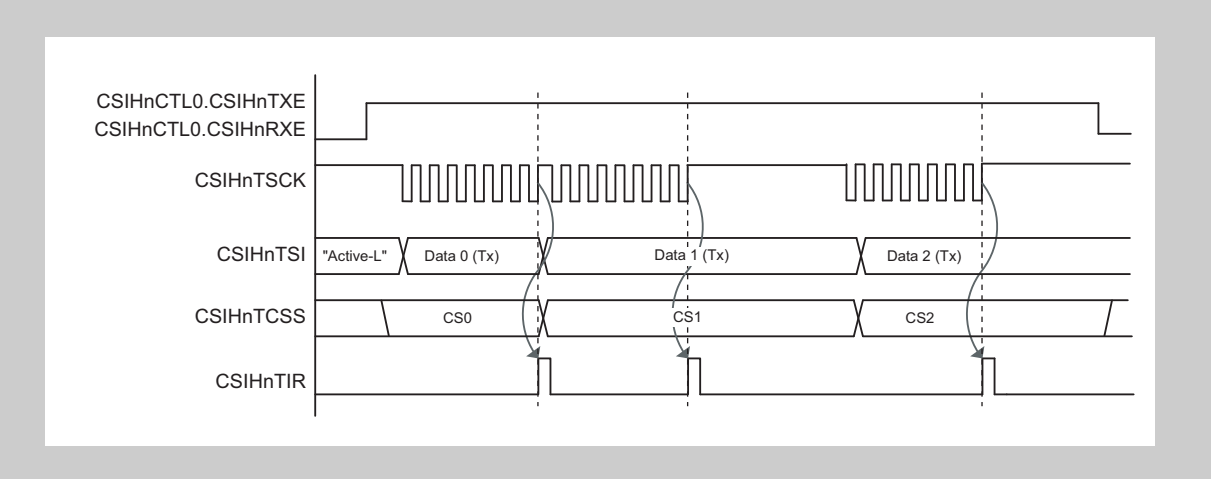


Figure 22-24 Generation of CSIHnTIR in Transmit-Only-Buffer or Double-Buffer Mode

(3) CSIHnTIRE (Reception Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 22-12 Data Error Types

Error Type	Communication Status after Error Interrupt	Comment
FIFO overflow error	Communication continues even after the interrupt is generated	The data written to the FIFO is lost, but previously started communications are continued.
Parity error	Communication continues even after the interrupt is generated	-
Data consistency error	Communication continues even after the interrupt is generated	-
Time-out error	Communication continues even after the interrupt is generated	-
Overrun error	If CSIHnCTL1.CSIHnHSE = 0 (no handshake), communication continues after the interrupt is generated (i.e. it does not stop).	This error is generated when the number of received data is 0 and the CPU attempts to read reception data.
	If CSIHnCTL1.CSIHnHSE = 1 (with handshake), communication stops due to the handshake. An interrupt is not generated so an overrun error will not occur.	This error is generated when data was received, but FIFO is full. In transmit-only buffer mode or direct access mode, reception data is left in the CSIHnRX0 register.

The type of error that caused the generation of CSIHnTIRE is flagged in register CSIHnSTR0.

Additionally a parity and data consistency error flag is attached to the reception data in CSIHnRX0W.

For details, about the various error types, refer to section 22, Error Detection.

(4) CSIHnTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs - for details, see section 22 3 The Job Concept. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, CSIHnTIJC is not generated.

This interrupt is generated according to the following conditions.

Table 22-13 CSIHnTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not generated	Indicates that the communication has stopped at the end of a job after a job abortion* was triggered
Transmit-only buffer		
Double-buffer		
Direct access		

Note: * Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

(5) All Interrupts Delay

In master mode, all interrupts generated by the master can be delayed by one half period of the serial clock CSIHnTSCK. This is not possible in slave mode.

The delay is specified by setting bit CSIHnCTL1.CSIHnSIT = 1.

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (normal clock and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

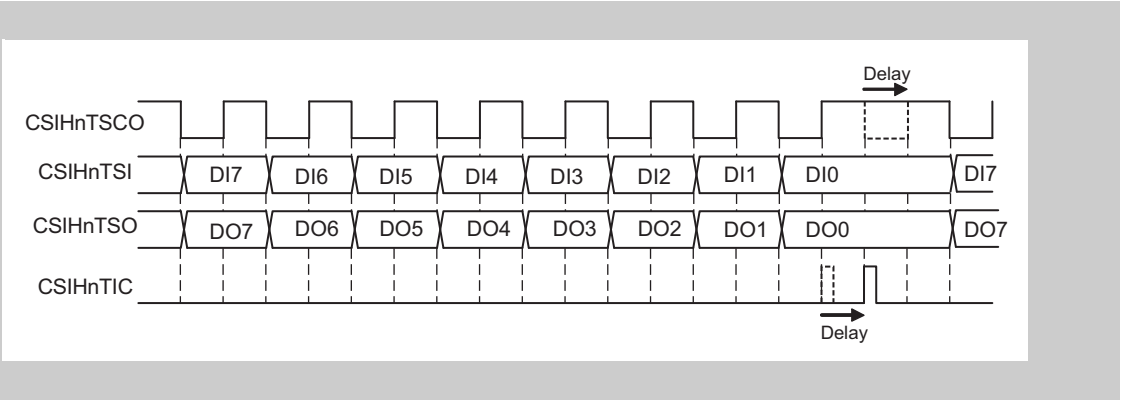


Figure 22-25 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds one half period delay to the serial clock. This delays also the end of the present chip select signal.

22.3.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIHnCTL1.CSIHnHSE. For handshake, the signal CSIHnTSHSG is used. If handshaking is disabled (CSIHnCTL1.CSIHnHSE = 0), the low level is output as the CSIHnTSHSG signal (i.e. on the CSIHnRYO pin).

The timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

The following describes operation when handshaking is enabled (CSIHnCTL1.CSIHnHSE = 1).

(1) Slave Mode

When CSIHnCTL1.CSIHnHSE = 1, the slave outputs CSIHnTSHSG = 0 (i.e. on the CSIHnRYO pin) when it is busy. States of transmission and reception by a slave in the respective modes when it is busy are given below.

Table 22-14 When Not Ready to Transmit

Memory Mode	Operation CSIHnCTL0.CSIHnTXE = 1
FIFO mode	There are no data for subsequent transmission (CSIHnSTR0.CSIHnEMF = 1).
Direct access mode	There are no data for subsequent transmission.
Double-buffer mode	CSIHnMCTL2.CSIHnBTST is not set to 1.
Transmit-only buffer mode	

Table 22-15 When Received Data are Left in the CSIHnRXW/H Register and the Transfer of Data for Transmission from the Shift Register to the CSIHnRXW/H Register was not Possible

Memory Mode	Operation CSIHnCTL0.CSIHnRXE = 1
FIFO	FIFO buffer is full (CSIHnSTR0.CSIHnFLF = 1).
Direct access mode	CSIHnRX0W/H register is full.
Double-buffer mode	-
Transmit-only buffer mode	CSIHnRX0W/H register is full.

Note: If a low level of the CSIHnTSHSG signal ("busy" on the CSIHnRYI pin) is detected in master mode, the next transmission is put on hold and the interface enters a state of waiting. In this state, the level of the CSIHnTSHSG signal (the CSIHnRYI pin) is checked every half-cycle of the CSIHnTCSS0 clock. However, output of the chip-select signals (CSIHnTCSSx: x = 0 to 7 for CSIH0 and x = 0 to 3 for CSIH1) will have already started and is maintained while the interface is waiting.

Caution 1. Make sure that the master side only detects CSIHnTSHSG signals (the CSIHnRYO pin) from the slave selected by CSIHnTCSSx (x = 0 to 7 for CSIH0 and x = 0 to 3 for CSIH1).

Caution 2. Even if the master detects the active level from the slave during transmission, the next transmission does not proceed until the current transmission is completed.

Descriptions of the FIFO and direct-access modes follow.

1. Memory Mode is FIFO Mode
   The slave is in transmit-only or transmit/receive mode but has no transmission data in its buffer. This status is indicated by the flag CSIHnSTR0.CSIHnEMF.

   The following examples assume a data length of 8 bits.

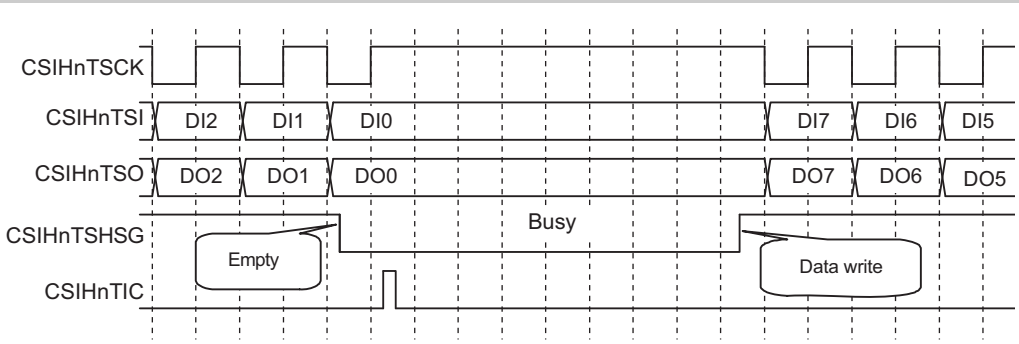


Figure 22-26 Busy Signal from Slave (FIFO Mode, CSIHnCFGx.CSIHnDAPx = 0)

The slave sets CSIHnTSHSG to high ("ready") as soon as new transmission data is written to the FIFO.

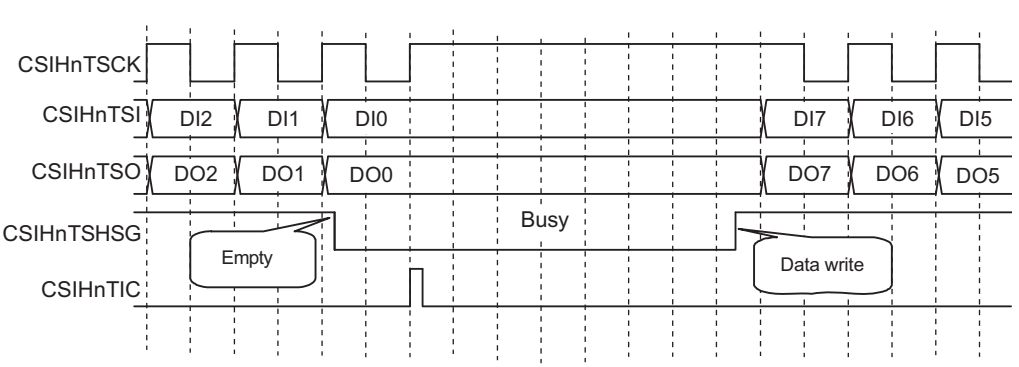


Figure 22-27 Busy Signal from Slave (FIFO Mode, CSIHnCFGx.CSIHnDAPx = 1)

2. Memory Mode is Direct Access Mode

The slave is in receive-only or transmit/receive mode but previously received data is still in the CSIHnRX0W register, and new data cannot be copied from the shift register to CSIHnRX0W (CSIHnRX0W full condition).

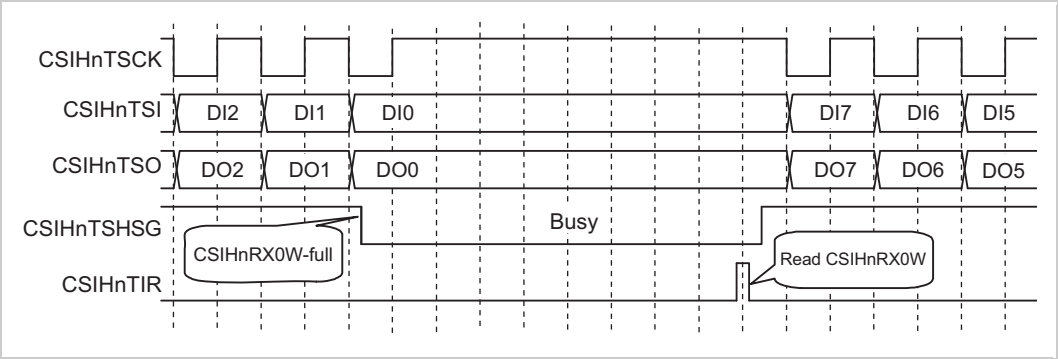


Figure 22-28 Busy Signal from Slave (Direct Access, CSIHnCFGx.CSIHnDAPx = 0)

The slave sets CSIHTSHSG to high (“ready”) as soon as the reception register CSIHnRX0W has been read.

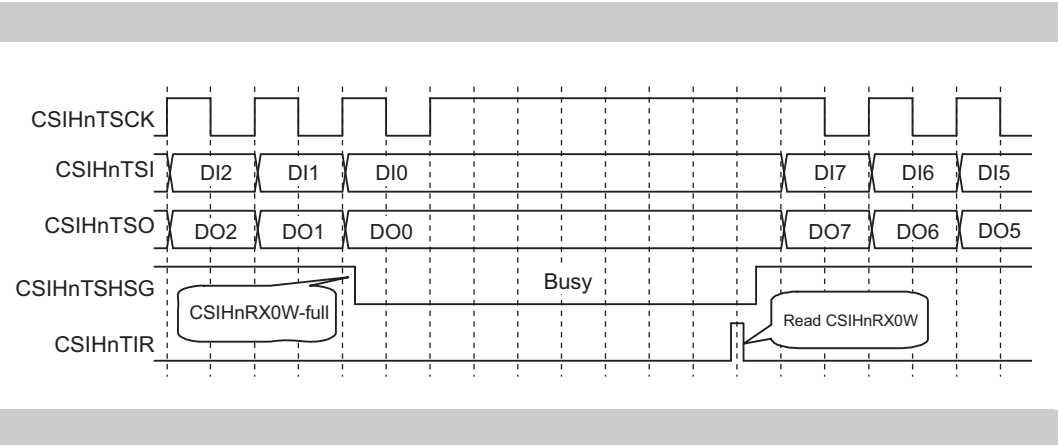


Figure 22-29 Busy Signal from Slave (Direct Access, CSIHnCFGx.CSIHnDAPx = 1)

(2) Master Mode

When the master detects CSIHnTSHSG = 0, the following transfer is put on hold, and the master goes into wait status. It suspends the clock CSIHnTSCK.

The CSIHnTSHSG level is checked at each half clock cycle of CSIHnTSCK.

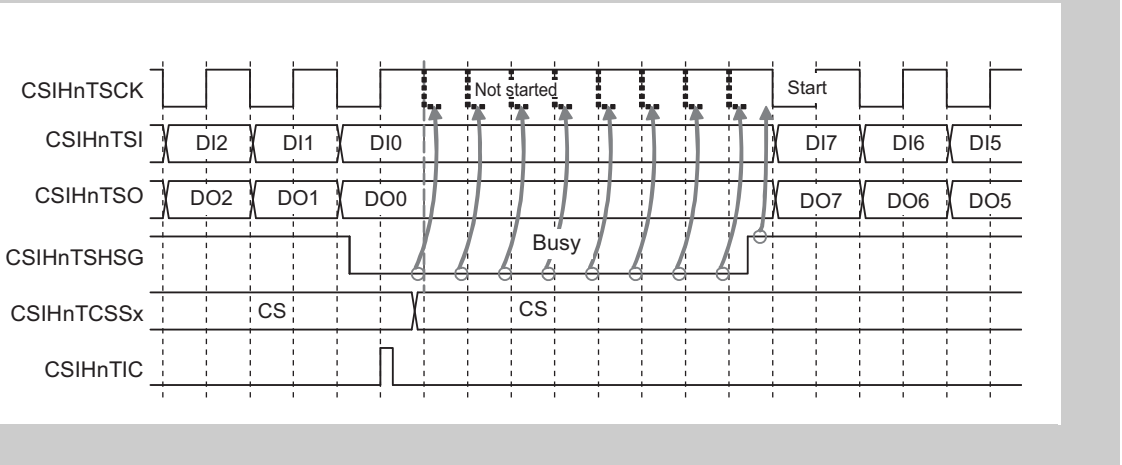


Figure 22-30 Master's Reaction on CSIHnTSHSG (CSIHnCFGx.CSIHnDAPx = 0)

CSIHnTSHSG must be pulled down by the slave before the next transfer starts. If the CSIHnTSHSG signal is pulled down while a data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIHnTSHSG becomes high (slave is “ready”).

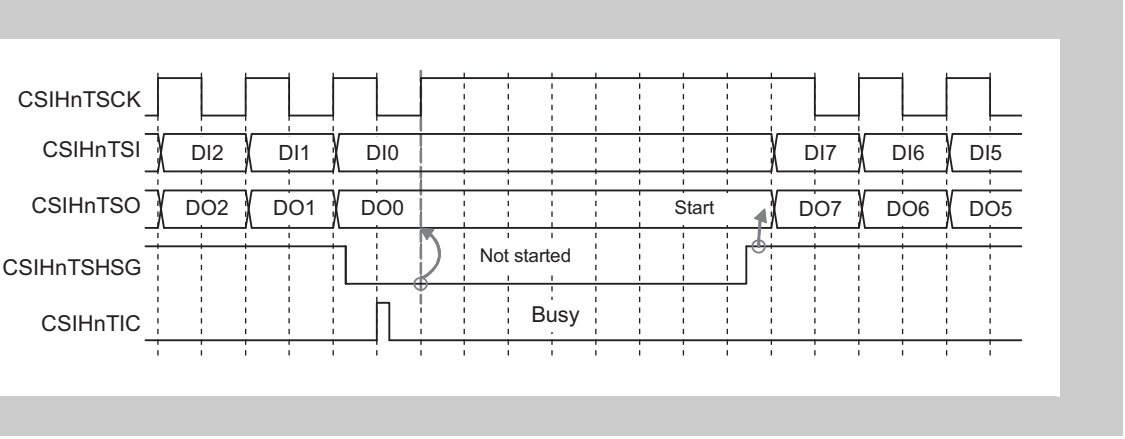


Figure 22-31 Master's Reaction on CSIHnTSHSG (CSIHnCFGx.CSIHnDAPx = 1)

Caution If multiple slaves are connected, the master must only detect the CSIHnTSHSG signal of the slave it has selected for communication.

22.3.12 Error Detection

The CSIHn can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency, and time-out errors can be enabled/disabled individually.

If one of these errors is detected, the interrupt request CSIHnTIRE is generated and the corresponding flags are set.

(1) Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output pin is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by bit CSIHnCTL1.CSIHnDCS. It is not active if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels at CSIHnTSO are written into an own shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error.

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnDCE is set.

Additionally, CISHnRX0W.CSIHnTDCE is set with the corresponding data.

Caution If you wish to apply data consistency checking, set the bits allocated to the CSIHnTSO pin in the corresponding PIPCn and PBDCn registers to 0 and 1, respectively.

The function of data consistency checking is illustrated in the following block diagram.

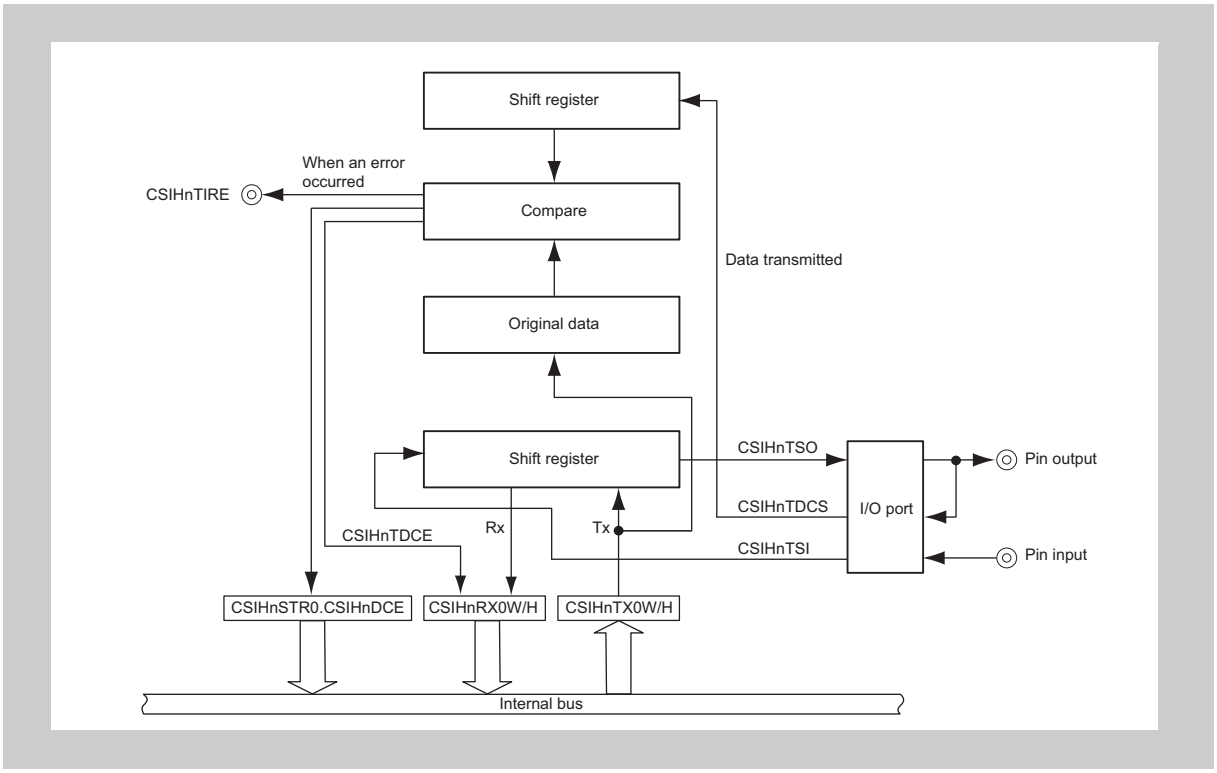


Figure 22-32 Functional Block Diagram of the Data Consistency Check

(2) Parity Check

CSIHn can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

The parity bit is checked after a reception is complete. In case of parity error:

- Interrupt CSIHnTIRE is generated.
- Parity error flag (CSIHnSTR0.CSIHnPE) is set.

Additionally, CISHnRX0W.CSIHnRPE is set with the corresponding data.

The following figure shows an example.

- Data length is 8 bits.
- The data transmitted is 05_H and 35_H
- Data direction is LSB first.
- Parity type is odd.

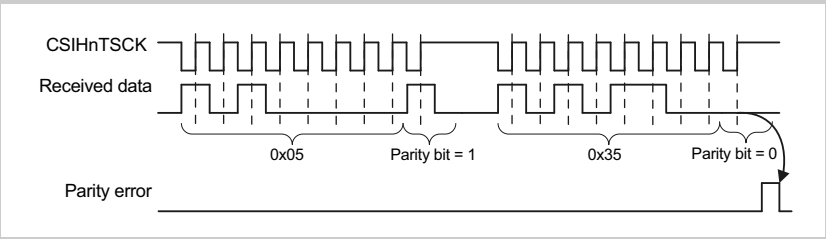


Figure 22-33 Parity Check Example

For the first data received, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the next data received, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

(3) Time-Out Error

Time-out errors checks are only possible in slave FIFO mode.

This error occurs when received data in the FIFO is not read or no data from CSIHnTSI is received in the FIFO within a certain time.

The time is defined in CSIHnMCTL0.CSIHnTO[4:0] in multiples of 8 times the serial clock CSIHnSCK. Time-out error occurs when the specified time is exceeded. However, the above error checking is invalid with CSIHnMCTL0. CSIHnTO[4:0]=00000_B and does not generate time-out error.

A dedicated time-out counter measures the time between the last and the next read operation.

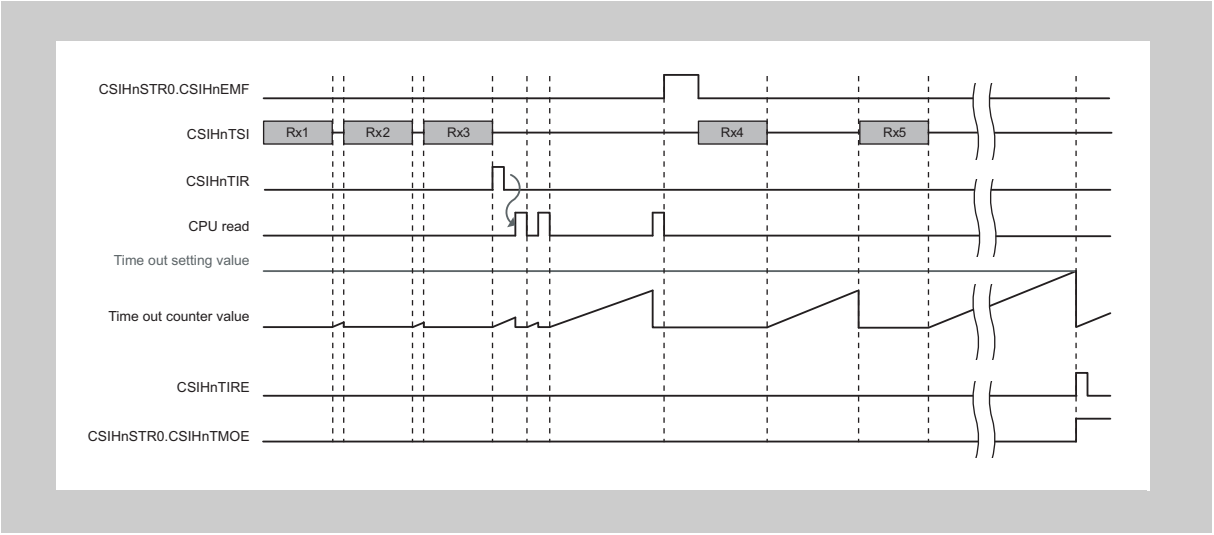


Figure 22-34 Time-Out Check Functional Timing Diagram

The counter (re-)starts after every data reception and after every read operation, if data is still available in the FIFO. It does not start when the buffer is marked as empty (CSIHnSTR0.CSIHnEMF = 1).

The counter is reset at the following timing:

- When time-out error was detected.
- When the data is cleared by CSIHnSTCR0.CSIHnPCT.
- Every time a CPU read is made.
- Every time a new data is received.

In case of time-out error:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnTMOE is set.

(4) Overflow Error

Overflow errors may occur in FIFO mode. They occur when transmission data is written to the CSIHnTX0W register or CSIHnTX0H register while the FIFO buffer is filled with transmission and reception data.

Example 100 data have been transmitted. That means, the FIFO contains 100 received data. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data to the FIFO. However, only 10 received data have been read up to now, 90 are still in the FIFO.

In this case, only 38 cells are available for new transmission data. When the CPU tries to write the 39th data, an overflow error occurs.

This is illustrated in the following figure.

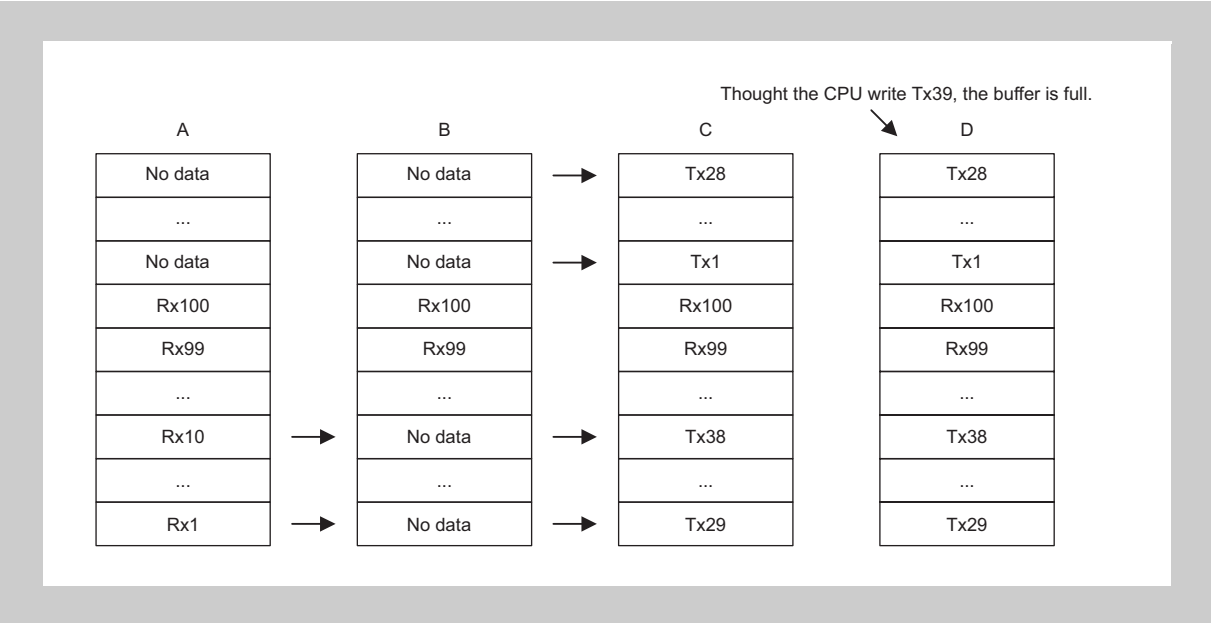


Figure 22-35 FIFO Overflow

The 39th and subsequent data are discarded. The following figure shows the overflow timing.

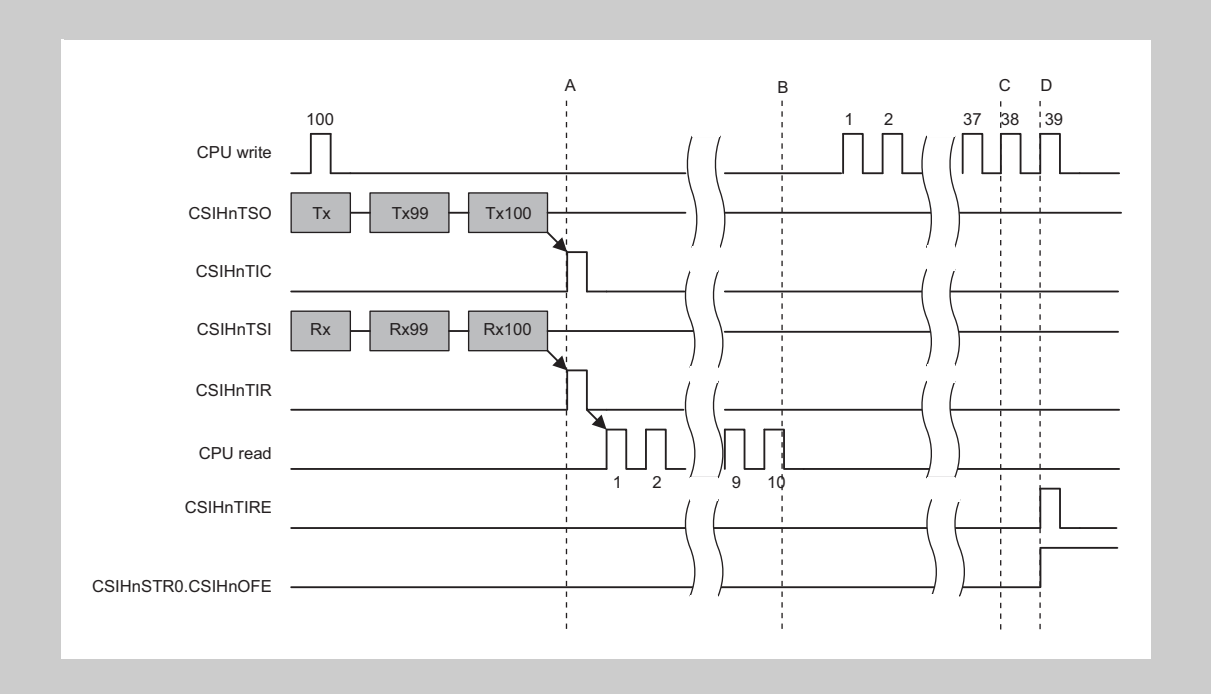


Figure 22-36 FIFO Overflow Timing

In case of overflow error:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOFE is set.

(5) Overrun Error

Overrun errors may occur in direct-access, transmit-only buffer, and FIFO modes. They will not occur in double-buffer mode. Overrun errors will not be generated if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

Direct access/ transmit-only buffer modes

In direct access and transmit-only buffer mode, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W. This happens when CSIHnRX0W was not read and therefore contains previous reception data.

The following figure illustrates the function.

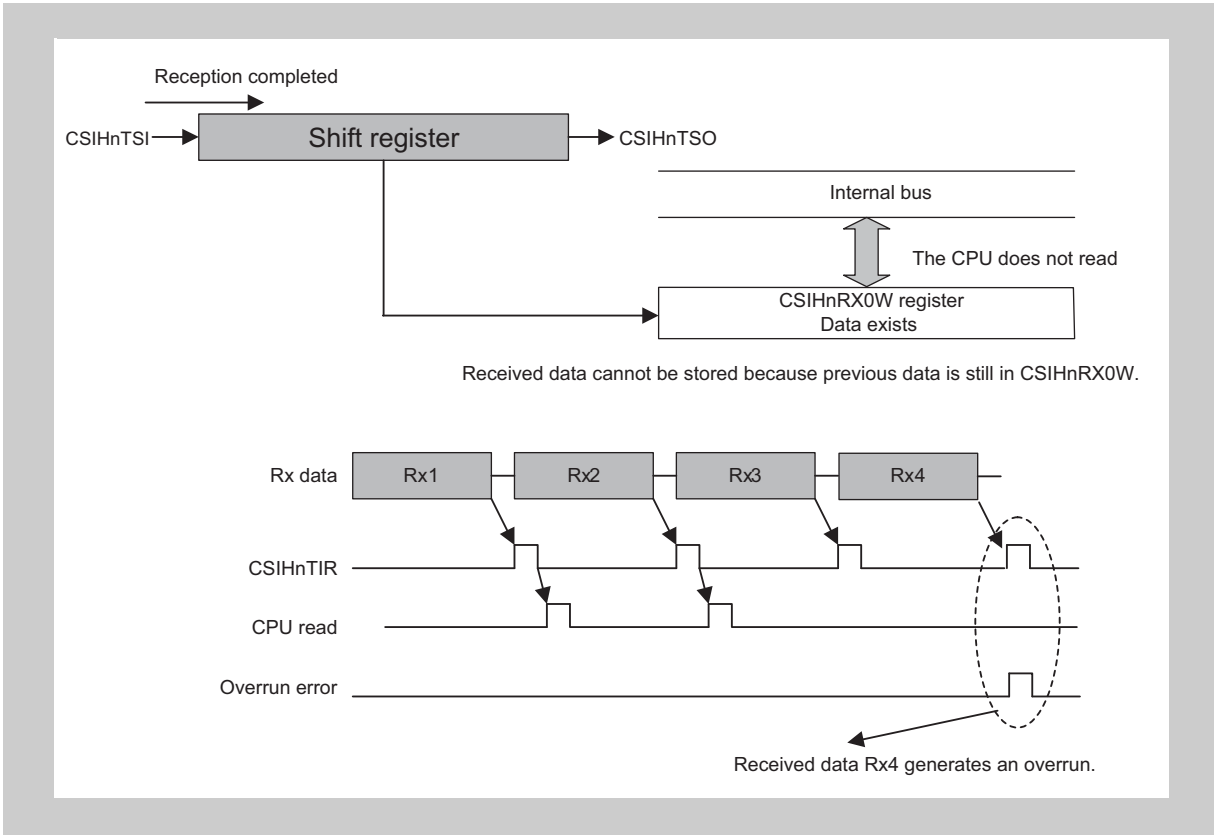


Figure 22-37 Overrun Error Detection in Direct Access and Transmit-Only Buffer Modes

Note Overrun errors do not occur for a master during direct-access or transmit-only buffer mode operation.

FIFO mode In FIFO mode, this error occurs if:

1. Newly received data cannot be transferred from the shift register to the FIFO because the FIFO is full.

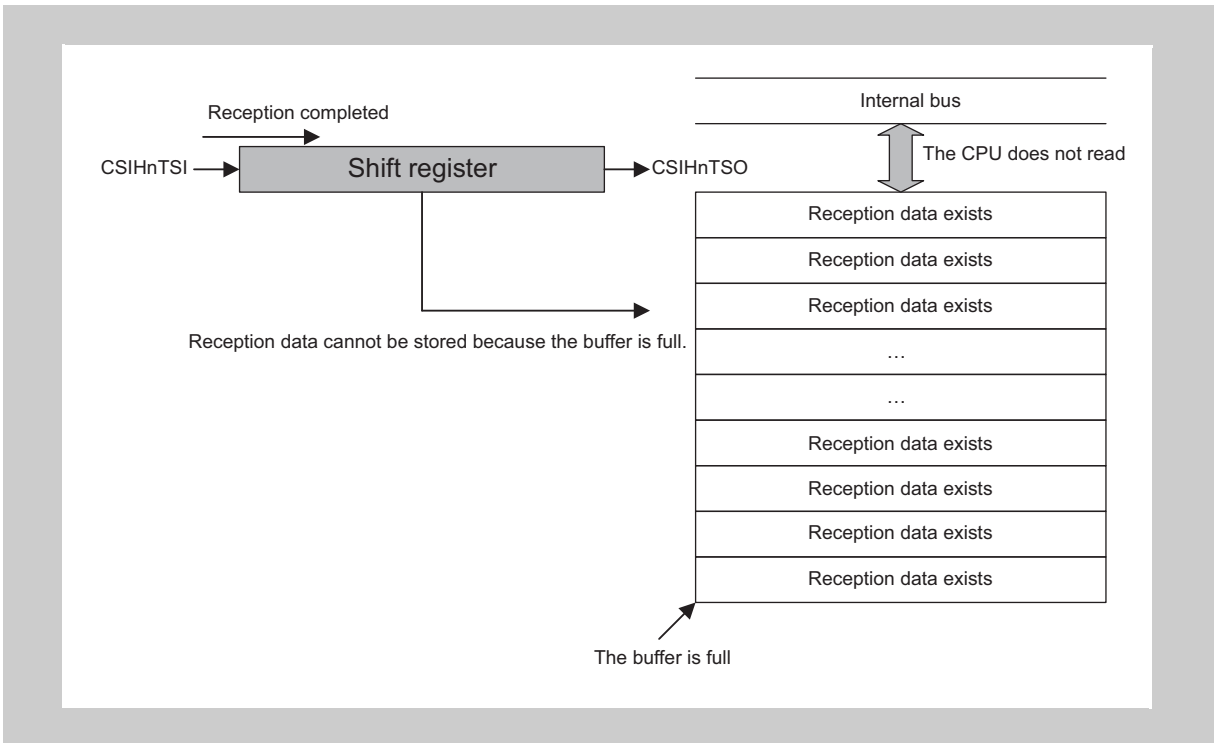


Figure 22-38 Overrun Error Detection in FIFO Mode (FIFO Full)

2. The CPU attempts to read non existing reception data.

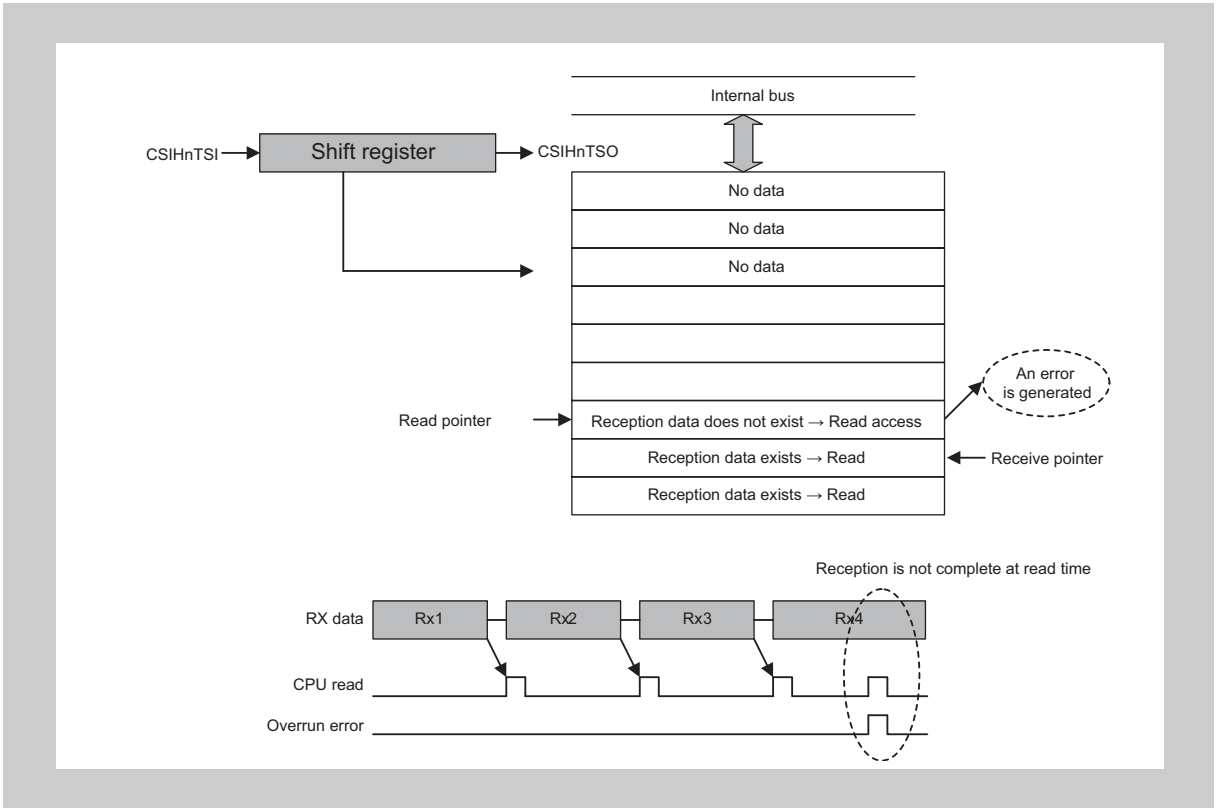


Figure 22-39 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOVE is set.
- The received value is written over the contents of the CSIHnRX0W register.
- Communications continue (except when the CPU has tried to read non-existing data).

Note An overrun error can be avoided in slave mode by using the handshake. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception register and is ready again.

For details, see section 22, Handshake Function.

22.3.13 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active, the transmit and receive signals are internally connected, as shown in the figures below.

The signals CSIHnTSCK, CSIHnTSO, and CSIHnTSI are disconnected from the ports. In addition, the CSIHnTSO output level is fixed to low, and CSIHnTSCK is set to the inactive level, as defined by CSIHnCFGx.CSIHnCKPx. The rest of CSIHn works as in normal operation.

In order to test the CSIHn, set CSIHnCTL1.CSIHnLBM to 1 and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

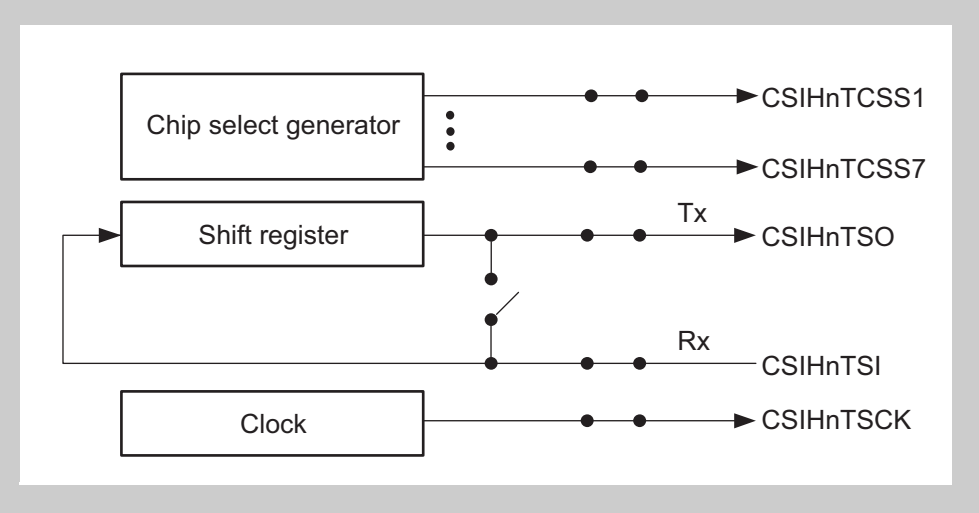


Figure 22-40 Normal Operation (CSIHnCTL1.CSIHnLBM = 0)

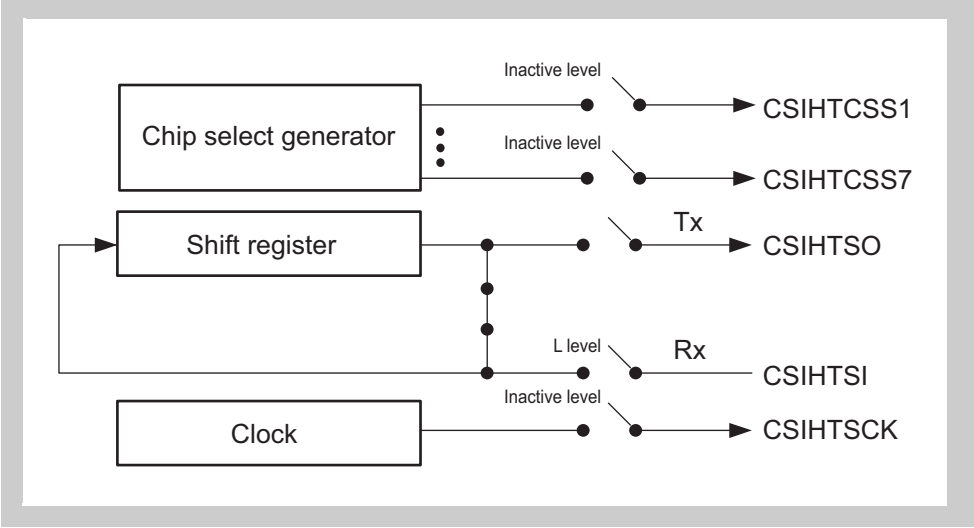


Figure 22-41 Loop-Back Operation (CSIHnCTL1.CSIHnLBM = 1)

22.4 CSIHn Control Registers

The CSIHn is controlled and operated by means of the following registers.

Table 22-16 CSIH Register Overview

Register Name	Symbol	Address
Control register 0	CSIHnCTL0	<CSIHn_base1> + 0000_H
Control register 1	CSIHnCTL1	<CSIHn_base0> + 0010_H
Control register 2	CSIHnCTL2	<CSIHn_base0> + 0014_H
Status register 0	CSIHnSTR0	<CSIHn_base1> + 0004_H
Status clear register 0	CSIHnSTCR0	<CSIHn_base1> + 0008_H
Memory control register 0	CSIHnMCTL0	<CSIHn_base0> + 1040_H
Memory control register 1	CSIHnMCTL1	<CSIHn_base1> + 0080_H
Memory control register 2	CSIHnMCTL2	<CSIHn_base1> + 0084_H
Transmission register 0 for word access	CSIHnTX0W	<CSIHn_base1> + 0088_H
Transmission register 0 for half word access	CSIHnTX0H	<CSIHn_base1> + $008C_H$
Reception register 0 for word access	CSIHnRX0W	<CSIHn_base1> + 0090_H
Reception register 0 for half word access	CSIHnRX0H	<CSIHn_base1> + 0094_H
Memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base1> + 0098_H
Configuration register 0	CSIHnCFG0	<CSIHn_base0> + 1044_H
Configuration register 1	CSIHnCFG1	<CSIHn_base0> + 1048_H
Configuration register 2	CSIHnCFG2	<CSIHn_base0> + $104C_H$
Configuration register 3	CSIHnCFG3	<CSIHn_base0> + 1050_H
Configuration register 4	CSIH0CFG4	<CSIH0_base0> + 1054_H
Configuration register 5	CSIH0CFG5	<CSIH0_base0> + 1058_H
Configuration register 6	CSIH0CFG6	<CSIH0_base0> + $105C_H$
Configuration register 7	CSIH0CFG7	<CSIH0_base0> + 1060_H

<CSIHn_base> The base addresses <CSIHn_base0> and <CSIHn_base1> of the CSIHn are defined in section 22 under the key word “Register addresses”.

22.4.1 CSIH Registers Details

(1) CSIHnCTL0 — CSIHn Control Register 0

This register controls the CSIHn. It enables and disables the operating clock, transmission and reception, and sections of memory for use in transmission and reception. It can also be used to forcibly stop transfer at the end of the current job.

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

Address <CSIHn_base1> + 0000_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
CSIHn PWR	CSIHn TXE	CSIHn RXE	0	0	0	CSIHn JOBE	CSIHn MBS
R/W	R/W	R/W	R	R	R	R/W	R/W

Table 22-17 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock 1: Provides operation clock Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets the CSIHn to standby state. No clock is provided to internal circuits. **Caution: If CSIHnPWR is cleared while transfer is in progress, transfer is immediately aborted. In such cases, transfer must be restarted.**
6	CSIHnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIHnRXE	Enables/disables reception. 0: Reception disabled 1: Reception enabled
1	CSIHnJOBE	Stops the communication at the end of the current job. 0: Communication stop is not required 1: Communication stop This is a trigger bit that can be used to abort an ongoing job. In FIFO mode, the next communication should then be started after clearing the pointers by setting CSIHnSTCR0.CSIHnPCT = 1. **Caution: CSIHnJOBE is only valid when CSIHnCTL1.CSIHnJE = 1. Modification of this bit in slave mode is prohibited. This bit is always read as 0.**
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIHn memory is used for transmission and/or reception data. 1: Direct access mode CSIHn memory is bypassed. **Caution: Modify this bit at the same time as the transition CSIHnPWR = 0 → 1 in slave mode.**

Caution 1. Do not modify any of these bits while CSIHnPWR = 0. These bits can be modified with the same write operation when setting CSIHnPWR = 1.

Caution 2. Do not modify CSIHnTXE, CSIHnRXE, or CSIHnMBS while a data transmission is pending or going on, i.e. if CSIHnSTR0.CSIHnTSF = 1.

(2) CSIHnCTL1 — CSIHn Control Register 1

This register controls the CSIHn. It specifies the clock phase, interrupt timing, and the interrupt delay mode. It enables or disables extended data length control, data consistency checking, loop-back mode, handshaking, and job mode. It sets the active output level on each chip-select line and the behavior of the chip-select signals after the final data transfer.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base0> + 0010_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHnCKR	CSIHnSLIT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnCSL7	CSIHnCSL6	CSIHnCSL5	CSIHnCSL4	CSIHnCSL3	CSIHnCSL2	CSIHnCSL1	CSIHnCSL0	CSIHnEDLE	CSIHnJE	CSIHnDCS	CSIHnCSRI	CSIHnLBM	CSIHnSIT	CSIHnHSE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 22-18 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
17	CSIHnCKR	Selects the clock phase of CSIHnTSCK. 0: Default level of CSIHnTSCK is high. 1: Default level of CSIHnTSCK is low. Caution: This bit is only valid when the chip-select signal is not in use. Fix it to 0 when the chip-select signal is in use.
16	CSIHnSLIT	Selects the timing of interrupt CSIHnTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt is generated as soon as new data can be written to register CSIHnTX0 (works only in direct access mode). For details, refer to section 22 (1) CSIHnTIC (Communication Interrupt).
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHnTCSSx). 0: Chip select is active low 1: Chip select is active high For details, refer to section 22.3.3, Chip Selection (CS) Features.
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to section 22 (2) Data Length Greater than 16 Bits.
6	CSIHnJE	Enables/disables job mode. 0: Job mode disabled 1: Job mode enabled For details, refer to section 22 (3) The Job Concept. Caution: This bit should be 0 in slave mode.
5	CSIHnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to section 22 (1) Data Consistency Check.

Table 22-18 CSIHnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
4	CSIHnCSRI	Defines chip select behavior after last data transfer. 0: Chip select holds active level 1: Chip select returns to inactive level
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated For details, refer to section 22, Loop-Back Mode. Caution: This bit should be 0 in slave mode.
2	CSIHnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, refer to section 22 (5) All Interrupts Delay.
1	CSIHnHSE	Enables/disables handshake mode. 0: Handshake function disabled 1: Handshake function enabled For details, refer to section 22.3.11, Handshake Function.

Caution Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

(3) CSIHnCTL2 — CSIHn Control Register 2

This register controls the CSIHn. This register is used to select the operating mode and prescaler, and to specify the baud rate.

For details, refer to section 22.3.5, Serial Clock Selection.

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

Address <CSIHn_base0> + 0014_H

Initial value $E000_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnPRS[2:0]			0	CSIHnBRS[11:0]											
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

Table 22-19 CSIHnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIHnPRS [2:0]	Selects the operating mode and the value of the prescaler.
11 to 0	CSIHnBRS [11:0]	Selects the baud rate.

CSIHn PRS2	CSIHn PRS1	CSIHn PRS0	Prescaler output (PRSOUT)
0	0	0	PCLK (master mode)
0	0	1	PCLK/2 (master mode)
0	1	0	PCLK/4 (master mode)
0	1	1	PCLK/8 (master mode)
1	0	0	PCLK/16 (master mode)
1	0	1	PCLK/32 (master mode)
1	1	0	PCLK/64 (master mode)
1	1	1	External clock via CSIHnTSCK (slave mode)

CSIHnBRS[11:0]	Baud rate at CSIHnBRS[11:0] CSIHnTSCK
0	BRG is stopped
1	PCLK/($2^m \times 1 \times 2$)
2	PCLK/($2^m \times 2 \times 2$)
3	PCLK/($2^m \times 3 \times 2$)
4	PCLK/($2^m \times 4 \times 2$)
...	...
4095	PCLK/($2^m \times 4095 \times 2$)

Settings of m = CSIHnCTL2.CSIHnPRS[2:0] = 0 to 6
In slave mode, these bits are used in clock setting for the time-out error function.

(4) CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of the CSIHn.

Access This register can be read in 32-bit units.

However, bits CSIHnTMOE, CSIHnOFE, CSIHnDCE, CSIHnPE, and CSIHnOVE are writable when CSIHnCTL0.CSIHnPWR = 0.

Address <CSIHn_base1> + 0004_H

Initial value $0000\ 0010_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnSRP[7:0]								CSIHnSPF[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTMOE	CSIHnOFE	0	0	0	0	0	0	CSIHnTSF	0	CSIHnFLF	CSIHnEMF	CSIHnDCE	0	CSIHnPE	CSIHnOVE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22-20 CSIHnSTR0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	CSIHnSRP [7:0]	Indicates the number of received words in FIFO mode. **CSIHnSRP [7:0]** / **Description** 00_H — Number of received words (0 to 128_D) 80_H > 80_H — Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In double-buffer or transmit-only buffer mode, these bits are fixed to 0 because the number of data is controlled by CSIHnMCTL2.CSIHnND[6:0]. These bits are fixed to 0 in direct access mode because there is no pointer.
23 to 16	CSIHnSPF [7:0]	Indicates the number of untransmitted data in FIFO mode. **CSIHnSPF [7:0]** / **Description** 00_H — Number of untransmitted data (0 to 128_D) 80_H > 80_H — Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In double-buffer or transmit-only buffer mode, these bits are fixed to 0 because the number of data is controlled by CSIHnMCTL2.CSIHnND[6:0]. These bits are fixed to 0 in direct access mode because there is no pointer.

Table 22-20 CSIHnSTR0 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOE</td><td>Time-out error flag
Indicates whether a time-out error was detected.
0: No time-out error detected
1: Time-out error detected
For details, refer to section 22.3.12 (3) Time-Out Error.
This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.
This bit is writable when CSIHnSTCR0.CSIHnPWR = 0.
This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
If setting to 1 due to detection of a time-out error and clearing to 0 by setting CSIHnSTCR0.CSIHnTMOEC coincide, the former (setting to 1 due to detection of the time-out error) takes precedence.</td></tr>
<tr><td>14</td><td>CSIHnOFE</td><td>Overflow error flag
Indicates whether an overflow error was detected.
0: No overflow error detected
1: Overflow error detected
For details, see section 22.3.12 (4) Overflow Error.
This bit is cleared by CSIHnSTCR0.CSIHnOFEC.
This bit is writable when CSIHnSTCR0.CSIHnPWR = 0.
This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
When 129 transmission data are written to Tx register while CSIHnPWR = 0, an overfolow error is generated.
If setting to 1 due to detection of a overflow error and clearing to 0 by setting CSIHnSTCR0.CSIHnOFEC coincide, the former (setting to 1 due to detection of the overfolow error) takes precedence.</td></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer status flag
0: Idle state
1: Transmission is in progress or being prepared
Setting and clearing of this bit is as follows.
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Set by</th><th rowspan="2">Cleared by</th></tr>
<tr><th>Direct Access/ FIFO</th><th>Double-Buffer/ Transmit-Only Buffer</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="3">Write to CSIHnTX0W register or CSIHnTX0H register</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST = 1</td><td rowspan="3">Within 0.5 clock from the edge of the last CSIHnTSCK</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave Mode</th><th colspan="2">Set by</th><th rowspan="2">Cleared by</th></tr>
<tr><th>Direct Access/ FIFO</th><th>Double-Buffer/ Transmit-Only Buffer</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="2">Write to CSIHnTX0W register or CSIHnTX0H register</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST = 1</td><td rowspan="3">Within 0.5 clock from the edge of the last CSIHnTSCK</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td><td>CSIHnTSCK clock input</td></tr>
</table>
This bit is cleared by CSIHnSTCR0.CSIHnPCT.</td></tr>
</table>

Table 22-20 CSIHnSTR0 Register Contents (3/3)

Bit Position	Bit Name	Function
5	CSIHnFLF	FIFO full flag: Indicates a completely filled FIFO buffer. 0: FIFO buffer is not full 1: FIFO buffer is full The FIFO buffer may become full of untransmitted data or of received data. This bit is cleared by CSIHnSTCR0.CSIHnPCT.
4	CSIHnEMF	FIFO empty flag: Indicates an empty FIFO buffer. 0: FIFO buffer is not empty 1: FIFO buffer is empty This bit is cleared by CSIHnSTCR0.CSIHnPCT.
3	CSIHnDCE	Data consistency check error flag 0: No data consistency error detected 1: Data consistency error detected This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. This bit is writable when CSIHnSTCR0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If setting to 1 due to detection of a data consistency error and clearing to 0 by setting CSIHnSTCR0.CSIHnDCEC coincide, the former (setting to 1 due to detection of the data consistency error) takes precedence.
1	CSIHnPE	Parity error flag 0: No parity error detected 1: Parity error detected This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. This bit is writable when CSIHnSTCR0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If setting to 1 due to detection of a parity error and clearing to 0 by setting CSIHnSTCR0.CSIHnPEC coincide, the former (setting to 1 due to detection of the parity error) takes precedence.
0	CSIHnOVE	Overrun error flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. This bit is writable when CSIHnSTCR0.CSIHnPWR = 0 This bit is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. This bit is fixed to 0 in double-buffer mode. If setting to 1 due to detection of an overrun error and clearing to 0 by setting CSIHnSTCR0.CSIHnOVEC coincide, the former (setting to 1 due to detection of the overrun error) takes precedence.

Table 22-21 Bit States by Memory Mode

Bit Name	Bit Position	Direct Access	FIFO	Transmit-Only Buffer	Double-Buffer
CSIHnSRP [7:0]	31 to 24	Fixed to 0	Number of received data	Fixed to 0	Fixed to 0
CSIHnSPF [7:0]	23 to 16	Fixed to 0	Number of untransmitted data	Fixed to 0	Fixed to 0
CSIHn TMOE	15	Fixed to 0	0: Error not detected 1: Error detected	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: Error not detected 1: Error detected	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Transmission is in progress or being prepared			
CSIHnFLF	5	Fixed to 0	0: Not full 1: Full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: Not empty 1: Empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: Error not detected 1: Error detected			
CSIHnPE	1	0: Error not detected 1: Error detected			
CSIHnOVE	0	0: Error not detected 1: Error detected			Fixed to 0

(5) CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

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

The value is always read as 0000_H.

Address <CSIHn_base1> + 0008_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn TMOEC	CSIHn OFEC	0	0	0	0	0	CSIHn PCT	0	0	0	0	CSIHn DCEC	0	CSIHn PEC	CSIHn OVEC
W	W	R	R	R	R	R	W	R	R	R	R	W	R	W	W

Table 22-22 CSIHnSTCR0 Register Contents

Bit Position	Bit Name	Function
15	CSIHnTMOEC	Time-out error flag (CSIHnSTR0.CSIHnTMOE bit) clear command 0: No operation. Read value is always 0. 1: Clears the time-out error flag (CSIHnSTR0.CSIHnTMOE).
14	CSIHnOFEC	Overflow error flag (CSIHnSTR0.CSIHnOFE bit) clear command 0: No operation. Read value is always 0. 1: Clears the overflow error flag (CSIHnSTR0.CSIHnOFE).
8	CSIHnPCT	FIFO pointer control 0: No operation. Read value is always 0. 1: Clears all of the FIFO buffer pointers listed below if operation is in double-buffer, transmit-only buffer, or FIFO mode. - CSIHnMRWP0.CSIHnTRWA[6:0], - CSIHnMRWP0.CSIHnRRA[6:0], and - CSIHnMCTL2.CSIHnSOP[6:0] In FIFO mode only, setting the bit to 1 also clears the status bits listed below. - CSIHnSTR0.CSIHnSPF[7:0], - CSIHnSTR0.CSIHnSRP[7:0], - CSIHnSTR0.CSIHnFLF, and - CSIHnSTR0.CSIHnTSF and sets the CSIHnSTR0.CSIHnEMF bit to 1 (FIFO empty). Caution: If this bit is set to 1 during communication, on-going communication is aborted.
3	CSIHnDCEC	Data consistency error flag (CSIHnSTR0.CSIHnDCE bit) clear command 0: No operation. Read value is always 0. 1: Clears the data consistency error flag (CSIHnSTR0.CSIHnDCE).
1	CSIHnPEC	Parity error flag (CSIHnSTR0.CSIHnPE bit) clear command 0: No operation. Read value is always 0. 1: Clear the parity error flag (CSIHnSTR0.CSIHnPE).
0	CSIHnOVEC	Overrun error flag (CSIHnSTR0.CSIHnOVE bit) clear command 0: No operation. Read value is always 0. 1: Clear the overrun error flag (CSIHnSTR0.CSIHnOVE).

(6) CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

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

Address <CSIHn_base0> + 1040_H

Initial value $001F_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CSIHn MMS[1:0]		0	0	0	CSIHnTO[4:0]				
R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22-23 CSIHnMCTL0 Register Contents

Bit Position	Bit Name	Function
9, 8	CSIHn MMS[1:0]	Selects the memory mode. **CSIHn MMS1** / **CSIHn MMS0** / **Description** 0 / 0 / FIFO mode 0 / 1 / Double-buffer mode 1 / 0 / Transmit-only-buffer mode 1 / 1 / Prohibited After any change to the memory mode, set CSIHnSTCR0.CSIHnPCT to 1. Caution: Changing the memory mode is only permitted when CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.
4 to 0	CSIHn TO[4:0]	Selects the time-out setting. **CSIHnTO[4:0]** / **Description** 00000_B / No time-out detection 00001_B / Time-out is (1 × 8 × BRG output clocks) 00010_B / Time-out is (2 × 8 × BRG output clocks) ... / 11111_B / Time-out is (31 × 8 × BRG output clocks) Caution: Changing the time-out is only permitted when CSIHnCTL0.CSIHnPWR = 0. In master mode, set these bits to 0. In double-buffer, transmit-only buffer, and direct-access modes, set these bits to 0. For details, on time-out detection, see also section 22.3.12 (3) Time-Out Error.

(7) CSIHnMCTL1 — CSIH Memory Control Register 1

This register selects the conditions to generate the interrupt requests CSIHnTIC and CSIHnTIR in FIFO mode.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base1> +0080_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	CSIHnFES[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnFFS[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note Writing during communication is possible.

Table 22-24 CSIHnMCTL1 Register Contents

Bit Position	Bit Name	Function
22 to 16	CSIHnFES [6:0]	Selects the condition for the generation of CSIHnTIC interrupt requests in FIFO mode when the FIFO is empty. A CSIHnTIC interrupt request is generated when the number of untransmitted data in the FIFO buffer equals CSIHnMCTL1.CSIHnFES[6:0], i.e. when CSIHnSTR0.CSIHnSPF[7:0] is decremented to match CSIHnFES[6:0].
6 to 0	CSIHnFFS [6:0]	Selects the condition for the generation of CSIHnTIR interrupt requests in FIFO mode when the FIFO is full. A CSIHnTIR interrupt request is generated when (128 - CSIHnSTR0.CSIHnSRP[7:0]) equals CSIHnMCTL1.CSIHnFFS[6:0].

(8) CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory and initiates a communication start in double-buffer or transmit-only buffer mode.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base1> + 0084_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnBTST	0	0	0	0	0	0	0	CSIHnND[7:0]							
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnSOP[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-25 CSIHnMCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	CSIHnBTST	Provides a start trigger for buffer transfer. 0: No operation 1: Starts the transfer command The read value is always 0. Caution: This bit can only be used in double-buffer mode and transmit-only buffer mode. In direct-access mode and FIFO mode, this bit is disabled.
23 to 16	CSIHnND[7:0]	Specifies the number of data for the double-buffer mode and transmit-only buffer mode.

CSIHnND [7:0]	Double-Buffer Mode	Transmit-only-Buffer Mode	Direct-Access Mode	FIFO Mode
00_H	Send 0 data	Send 0 data	No influence	No influence
01_H	Send 1 data	Send 1 data	No influence	No influence
...	...	...	No influence	No influence
$3F_H$	Send 63 data	Send 63 data	No influence	No influence
40_H	Send 64 data	Send 64 data	No influence	No influence
...	Prohibited	...	No influence	No influence
$7F_H$	Prohibited	Send 127 data	No influence	No influence
80_H	Prohibited	Send 128 data	No influence	No influence
> 80_H	Undefined			

The value is automatically decremented after each data transfer.
In reference to these bits during transfer, the remaining number of data is read.
This field is not decremented in response to transfer in direct-access mode.

Table 22-25 CSIHnMCTL2 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 0</td><td>CSIHn SOP[6:0]</td><td>Selects the pointer of the data to be sent.
<table>
<tr><th>CSIHn SOP[6:0]</th><th>Double-Buffer Mode</th><th>Transmit-only-Buffer Mode</th><th>Direct Access Mode</th><th>FIFO Mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>No influence</td><td>0000_H</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>No influence</td><td>0004_H</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>No influence</td><td>...</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No influence</td><td>$00FC_H$</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>No influence</td><td>0100_H</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>No influence</td><td>...</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>No influence</td><td>$01FC_H$</td></tr>
</table>
These bits are cleared by hardware when communication is forced to stop by CSIHnCTL0.PWR = 0 or CSIHnSTR0.CSIHnPCT.
Note: In FIFO mode, these bits indicate the send address. This field is not incremented in response to transfer in direct-access mode.</td></tr>
</table>

Caution 1. Writing to this register when CSIHnSTR0.CSIHnTSF = 1 (i.e. during transfer) is prohibited.

Caution 2. Writing to this register when CSIHnCTL0.CSIHnPWR = 0 is prohibited.

Caution 3. Writing to this register when CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 0 is prohibited.

Caution 4. Writing to this register during operation in direct-access or FIFO mode is prohibited.

(9) CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the double-buffer or transmit-only buffer.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base1> + 0098_H

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	CSIHnRRA[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnTRWA[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-26 CSIHnMRWP0 Register Contents (1/2)

Bit Position	Bit Name	Function
22 to 16	CSIHn RRA[6:0]	Selects the read pointer of the Rx buffer. **CSIHn RRA[6:0]** / **Double-Buffer Mode** / **Transmit-only-Buffer Mode** / **Direct Access Mode** / **FIFO Mode** 00_H / 0000_H / No influence / No influence / 0000_H 01_H / 0004_H / No influence / No influence / 0004_H ... / ... / No influence / No influence / ... $3F_H$ / $00FC_H$ / No influence / No influence / $00FC_H$ 40_H / Prohibited / No influence / No influence / 0100_H ... / Prohibited / No influence / No influence / ... $7F_H$ / Prohibited / No influence / No influence / $01FC_H$ These bits are automatically incremented when the reception data is read. If an overrun error is generated while reading the CSIHnRX0W or CSIHnRX0H register, the read pointer is not incremented. These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1. In FIFO mode, these bits indicate read address of the reception data. In direct access mode or transmit-only buffer mode, these bits are not incremented.

Table 22-26 CSIHnMRWP0 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 0</td><td>CSIHn TRWA[6:0]</td><td>Selects the read/write pointer of the Tx buffer.
<table>
<tr><th>CSIHn TRWA[6:0]</th><th>Double-Buffer Mode</th><th>Transmit-only-Buffer Mode</th><th>Direct Access Mode</th><th>FIFO Mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>No influence</td><td>0000_H</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>No influence</td><td>0004_H</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>No influence</td><td>...</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No influence</td><td>$00FC_H$</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>No influence</td><td>0100_H</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>No influence</td><td>...</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>No influence</td><td>$01FC_H$</td></tr>
</table>
These bits are automatically incremented when the transmission data is written or read.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In FIFO mode, these bits indicate the read and write addresses of the transmission data. In direct-access mode, these bits are not incremented.</td></tr>
</table>

Caution 1. Writing during the communication is possible.

Caution 2. Writing to this register during operation in FIFO mode or direct-access mode is forbidden.

(10) CSIHnCFGx — CSIHn Configuration Register X

These 8 registers (4 registers for CSIH1) specify for each chip select signal CSIHnTCSSx prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock and data phase, idle enforcement configuration, idle timing, hold timing, inter-data delay timing, and setup timing.

Slave mode The transmission protocol setting of the CSIHnCFG0 register are valid in slave mode.

- CSIHnPS0: Parity usage
- CSIHnDLS0: Data length selection
- CSIHnDIR0: Data direction
- CSIHnCKP0, CSIHnDAP0: Clock and data phase

All other register settings are disregarded.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base0> + 1044_H (CSIHnCFG0) + xxx4

Initial value $0000\ 0000_H$

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHn PSCLx[1:0]		CSIHn PSx[1:0]		CSIHnDLSx[3:0]				0	0	0	0	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Write access is only possible while CSIHnCTL0.CSIHnPWR = 0.

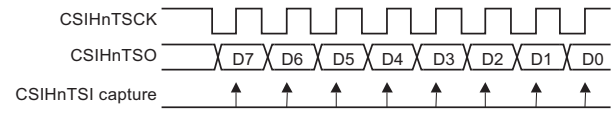
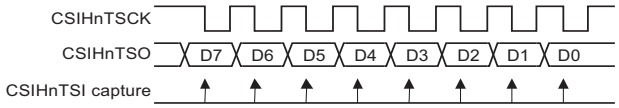
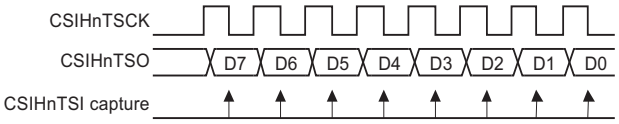
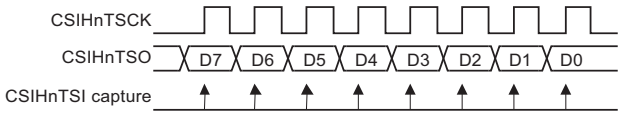
Table 22-27 CSIHnCFGx Register Contents (1/4)

Bit Position	Bit Name	Function
31, 30	CSIHn PSCLx[1:0]	Selects the prescaler for chip select signal x. **CSIHn PSCLx1** / **CSIHn PSCLx0** / **Prescaler Output** 0 / 0 / CSIHnBPCLK 0 / 1 / CSIHnBPCLK/2 1 / 0 / CSIHnBPCLK/4 1 / 1 / CSIHnBPCLK/8 These bits are only available in master mode.

Table 22-27 CSIHnCFGx Register Contents (2/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>29, 28</td><td>CSIHnPSx [1:0]</td><td>Selects the parity for chip select signal x for transmission and reception.
<table>
<tr><th>CSIHn PSx1</th><th>CSIHn PSx0</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>No parity transmitted</td><td>No parity judged</td></tr>
<tr><td>0</td><td>1</td><td>Add parity bit fixed to 0</td><td>Parity bit is received but not judged</td></tr>
<tr><td>1</td><td>0</td><td>Add odd parity</td><td>Odd parity judged</td></tr>
<tr><td>1</td><td>1</td><td>Add even parity</td><td>Even parity judged</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIHn DLSx[3:0]</td><td>Selects the data length for chip select signal x.
<table>
<tr><th>CSIHn DLSx[3:0]</th><th>Data Length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
Note: Data length between 1 bit and 6 bits requires that EDL function is used (see also section 22.3.7 (2) Data Length Greater than 16 Bits. In addition, it is forbidden to transmit two consecutive data with a length of less than 7 bits.</td></tr>
<tr><td>19</td><td>CSIHn RCBx</td><td>Selects the recessive configuration for broadcasting for chip select signal x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see section 22.3.3 (1) Configuration Registers.</td></tr>
<tr><td>18</td><td>CSIHn DIRx</td><td>Selects the serial data direction for chip select signal x.
0: Data is transmitted/received with MSB first
1: Data is transmitted/received with LSB first
For details, see section 22.3.9, Serial Data Direction Selection.</td></tr>
</table>

Table 22-27 CSIHnCFGx Register Contents (3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17, 16</td><td>CSIHnCKPx,
CSIHnDAPx</td><td>Selects the clock and data phase for chip select signal x.
The timing of data transfer is as follows.
<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock and Data Phase Selection</th></tr>
<tr><td>0</td><td>0</td><td> 
</td></tr>
<tr><td>0</td><td>1</td><td> 
</td></tr>
<tr><td>1</td><td>0</td><td> 
</td></tr>
<tr><td>1</td><td>1</td><td> 
</td></tr>
</table>
Caution: When chip selection is not performed, fix CSIHnCKPx bit to 0, and use the setting of CSIHnCKPx bit instead in CSIHnCTL1.CSIHnCKR bit.</td></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the idle enforcement configuration for chip select signal x.
0: If the chip select value remains the same, the chip select signal stays active. If a different chip select value is defined, chip select signal x becomes idle.
1: An idle state is inserted after every transfer of slave data corresponding to chip select signal x.

This bit is only available in master mode.</td></tr>
<tr><td>14 to 12</td><td>CSIHnIDx [2:0]</td><td>Selects the idle time for chip select signal x.
<table>
<tr><th>CSIHnDLSx[3:0]</th><th>Idle Timing</th></tr>
<tr><td>000_B</td><td>0.5 serial clock periods</td></tr>
<tr><td>001_B</td><td>1 serial clock period</td></tr>
<tr><td>010_B</td><td>1.5 serial clock periods</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5)</td></tr>
<tr><td>111_B</td><td>8.5 serial clock periods</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

Table 22-27 CSIHnCFGx Register Contents (4/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>CSIHnHDx [3:0]</td><td>Selects the hold time for chip select signal x in transmission clock periods.
<table>
<tr><th>CSIHn INx[3:0]</th><th>Hold Timing with CSIHnCTL1.CSIHnSIT = 0</th><th>Hold Timing with CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000_B</td><td>0.5</td><td>1</td></tr>
<tr><td>0001_B</td><td>1</td><td>1.5</td></tr>
<tr><td>0010_B</td><td>1.5</td><td>2</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td><td>... (3, 4, 5, 7, 9, 10, 11, 12, 13, 15, 17, 19)</td></tr>
<tr><td>1111_B</td><td>20.5</td><td>21</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>7 to 4</td><td>CSIHnINx [3:0]</td><td>Selects the inter-data delay time for chip select signal x in transmission clock periods.
<table>
<tr><th>CSIHn INx[3:0]</th><th>Hold Timing with CSIHnCTL1.CSIHnSIT = 0</th><th>Hold Timing with CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000_B</td><td>0</td><td>0.5</td></tr>
<tr><td>0001_B</td><td>0.5</td><td>1</td></tr>
<tr><td>0010_B</td><td>1</td><td>1.5</td></tr>
<tr><td>0011_B</td><td>2</td><td>2.5</td></tr>
<tr><td>...</td><td>... (3, 4, 6, 8, 9, 10, 11, 12, 14, 16, 18)</td><td>... (3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5 12.5, 14.5, 16.5, 18.5)</td></tr>
<tr><td>1111_B</td><td>20</td><td>20.5</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>3 to 0</td><td>CSIHnSPx [3:0]</td><td>Selects the setup time for chip select signal x in transmission clock periods.
<table>
<tr><th>CSIHn SPx[3:0]</th><th>Setup Delay</th></tr>
<tr><td>0000_B</td><td>0.5</td></tr>
<tr><td>0001_B</td><td>1</td></tr>
<tr><td>0010_B</td><td>1.5</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td></tr>
<tr><td>1111_B</td><td>20.5</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

(11) CSIHnTX0W — CSIHn Transmission Register 0 for Word Access

This register stores the transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base1> + 0088_H

Initial value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnCIRE	CSIHnEOJ	CSIHnEDL	0	0	0	0	0	CSIHnCS[7:0]							
R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTX[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-28 CSIHnTX0W Register Contents (1/2)

Bit Position	Bit Name	Function
31	CSIHnCIRE	Enables the communication interrupt request CSIHnTIC in double-buffer or transmit-only buffer mode or the job completion interrupt CSIHnTIJC request in FIFO mode. 0: No interrupt requested. 1: Interrupt requested. Generates interrupt CSIHnTIC or CSIHnTIJC after transmission. For details, see section 22.3.10 (1) CSIHnTIC (Communication Interrupt) and section 22.3.10 (4) CSIHnTIJC (Job Completion Interrupt). Caution: This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
30	CSIHnEOJ	Specifies the end of a job. 0: No end-of-job data. Job is continued. 1: End-of-job data. Caution: This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Extended data length activated The associated data is transmitted as of 16 bits. No inter-data delay or idle timing will be inserted after the data is transmitted. Caution: This bit is only available if job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and CSIHnCTL1.CSIHnEDLE = 1.

Table 22-28 CSIHnTX0W Register Contents (2/2)

Bit Position	Bit Name	Function
23 to 16	CSIHnCSx	Activates one or several chip select signals. 0: Chip select signal x is activated for the associated transmission 1: Chip select signal x is deactivated for the associated transmission If CSIHnTX0W.CSIHnCS[7:0] = FF_H, the data is sent with the configuration of register CSIHnCFG0. In this case, no chip select signal is activated. Caution: If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same configuration. If the CSIHnCSx bits are written in slave mode, FE_H is written to these bits.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

Caution 1. While communicating in FIFO mode, read access to these bits are prohibited.

Caution 2. In FIFO mode at CSIHnCTL0.CSIHnPWR = 0, write/read access to these bits are prohibited.

Caution 3. In direct access mode at CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 0, write access to these bits are prohibited.

(12) CSIHnTX0H — CSIHn Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

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

Address <CSIHn_base1> + $008C_H$

Initial value Undefined

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTX[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-29 CSIHnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data

Caution 1. While communicating in FIFO mode, read access to these bits are prohibited.

Caution 2. In FIFO mode at CSIHnCTL0.CSIHnPWR = 0, write/read access to these bits are prohibited.

Caution 3. In direct access mode at CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 0, write access to these bits are prohibited.

(13) CSIHnRX0W — CSIHn Reception Register 0 for Word Access

This register stores the received data.

Access This register can be read in 32-bit units.

Address <CSIHn_base1> + 0090_H

Initial value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	CSIHnRPE	CSIHnTDCE	CSIHnCS[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22-30 CSIHnRX0W Register Contents

Bit Position	Bit Name	Function
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error on the associated reception data detected 1: Parity error on the associated reception data detected
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. 0: No consistency error on the associated transmission data detected 1: Consistency error on the associated transmission data detected
23 to 16	CSIHnCSx	Indicates which chip select signal was activated. 0: Chip select signal x was activated for the associated transmission 1: Chip select signal x was deactivated for the associated transmission
15 to 0	CSIHnRX[15:0]	Stores the reception data. Read the previously received data by one serial-clock cycle before generation of the interrupt.

Caution 1. This register is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Caution 2. This register is writable while CSIHnCTL0.CSIHnPWR = 0.

Caution 3. In FIFO mode at CSIHnCTL0.CSIHnPWR = 0, write/read access to these bits are prohibited.

(14) CSIHnRX0H — CSIHn Reception Register 0 for Half Word Access

This register stores the reception data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

Access This register can be read in 16-bit units.

Address <CSIHn_base1> + 0094_H

Initial value Undefined

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22-31 CSIHnRX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIHnRX[15:0]	Stores the reception data.

Caution 1. This register is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Caution 2. This register is writable while CSIHnCTL0.CSIHnPWR = 0.

Caution 3. In FIFO mode at CSIHnCTL0.CSIHnPWR = 0, write/read access to these bits are prohibited.

22.5 Operating Procedures

The following examples and procedures are sorted according to the memory mode:

- Direct access
- Transmit-only buffer
- Double-buffer
- FIFO

22.5.1 Procedures in Direct Access Mode

Two examples for a master are given, one with job mode disabled, one with job mode enabled.

(1) Transmit/Receive in Master Mode, Job Mode Disabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

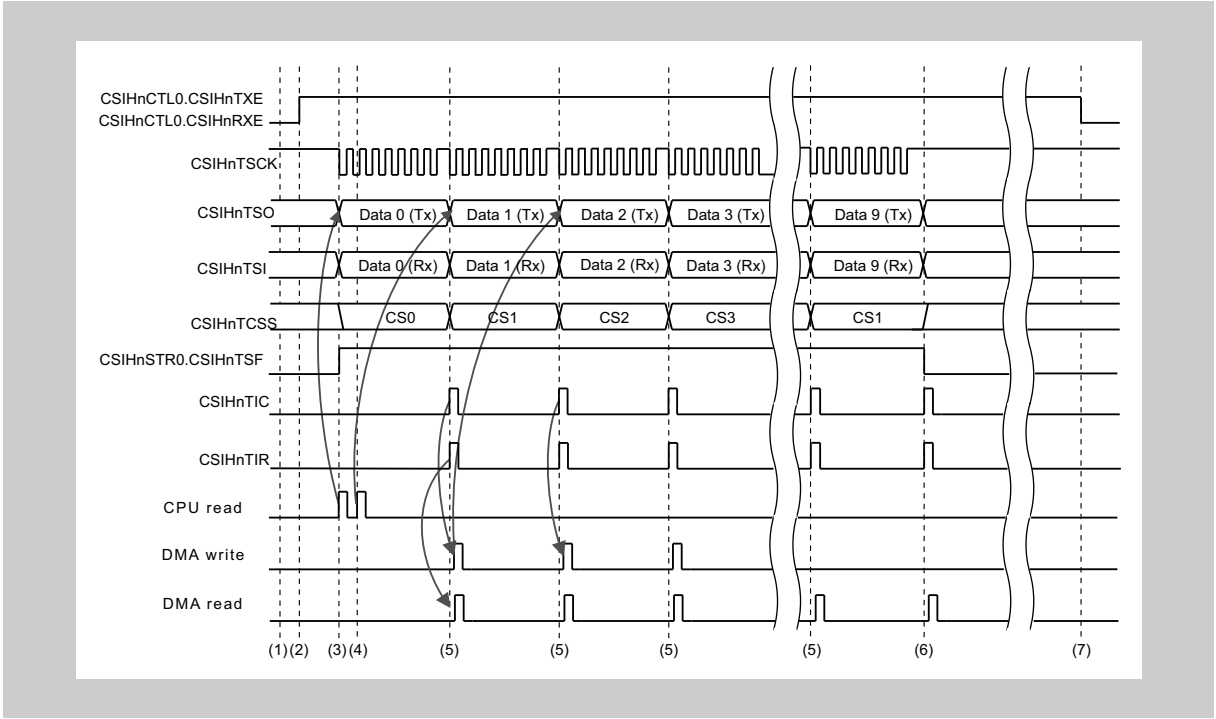


Figure 22-42 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the register CSIHnCFGx. This example uses the chip select signals CS0 to CS3.
2. In the CSIHnCTL0 register, set bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission), CSIHnRXE = 1 (enable reception), and CSIHnMBS = 1 (direct access mode).

   The CSIHnTSO output signal is enabled.
3. Write the first data to be sent to the transmission register CSIHnTX0W. Within the same write operation activate CS0. Transmission starts automatically when the first data is available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data that has been transmitted, the interrupts CSIHnTIC and CSIHnTIR are generated:
   - CSIHnTIC indicates that the next data can be written to CSIHnTX0W.
   - CSIHnTIR indicates that the reception register CSIHnRX0W must be read.
6. No more write action is required after completion of Data 8. Data 9 (the last data) has been written in advance.

   However, the reception register CSIHnRX0W must be read after completion of Data 8 and Data 9.
7. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE.

(2) Transmit/Receive in Master Mode, Job Mode Enabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Two jobs, each with three data are sent
- Normal CSIHnTIC interrupt timing (CSHICTL1.CSIHnSLIT = 0)

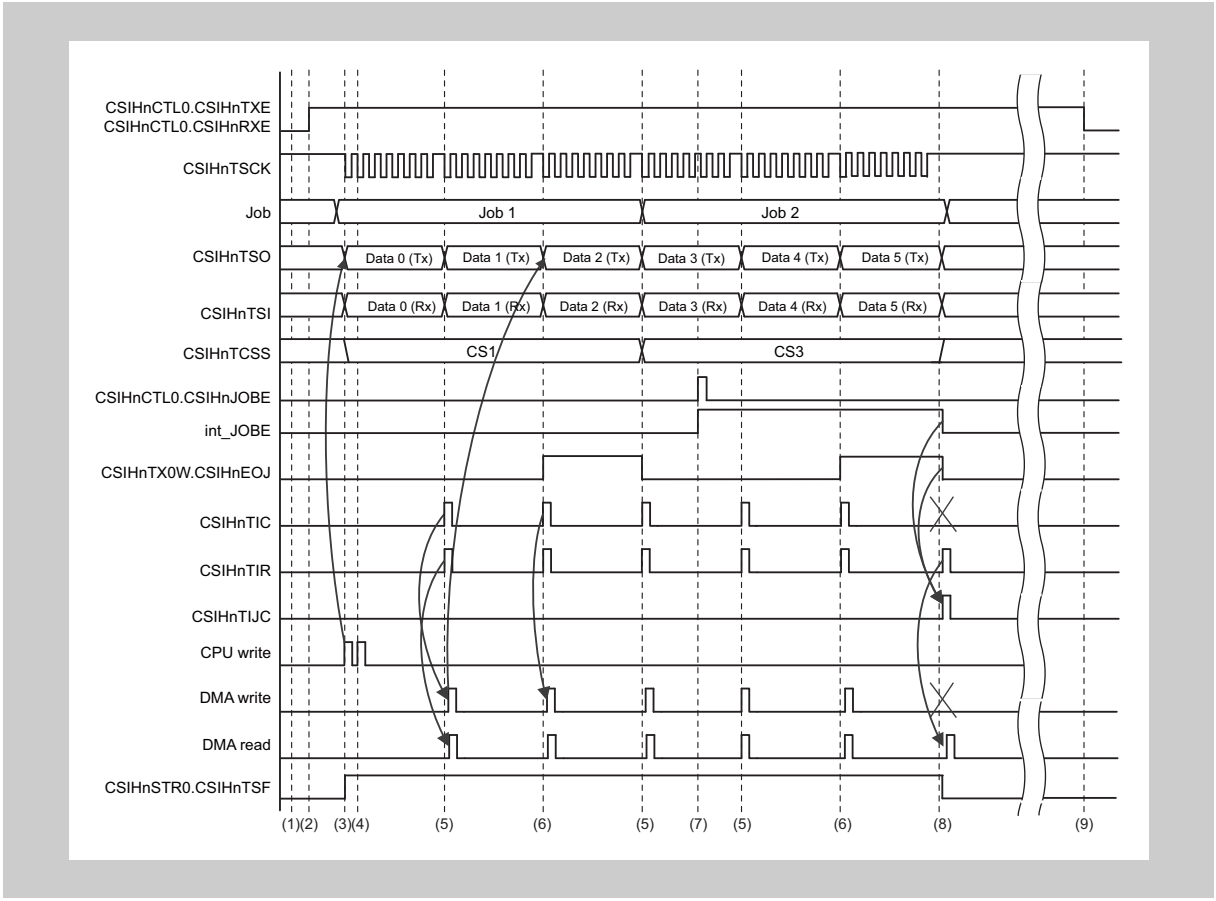


Figure 22-43 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Note The label “int_JOBE” in the above timing diagram indicates the internal signal representing bit CSIHnJOBE.

Procedure:

1. Configure the communication protocol in the register CSIHnCFGx This example uses the chip select signals CS1 to CS3.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission), CSIHnRXE = 1 (enable the reception) and CSIHnMBS = 1 (select direct access mode).

   This will enable the CSIHnTSO output signal.
3. Write the first data to be sent to the transmission register CSIHnTX0W. Transmission starts automatically when the first data is available.

   The flag CSIHnSTR0.CSIHnTSF signals that a communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data that has been transmitted, the interrupt requests CSIHnTIC and CSIHnTIR are generated:
   - CSIHnTIC indicates that the next data can be written to CSIHnTX0W.
   - CSIHnTIR indicates that the reception register CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 specifies that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIHnCTL0.CSIHnJOBE = 1, the communication is forced to stop at the end of the current job (Job 2).
8. After the forced stop of communication, the interrupt request CSIHnTIC is replaced by CSIHnTIJC. CSIHnTIR is generated as usual.

   The interrupt request CSIHnTIJC indicates a forced stop of communication at the end of the current job.

   The interrupt request CSIHnTIC is not generated. Additionally, the transmission data available in register CSIHnTX0W is not sent.
9. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE. Alternatively, a new transmission can be started from step 3.

22.5.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master are given, one with job mode disabled, one with job mode enabled.

(1) Transmit/Receive in Master Mode, Job Mode Disabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0,CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)

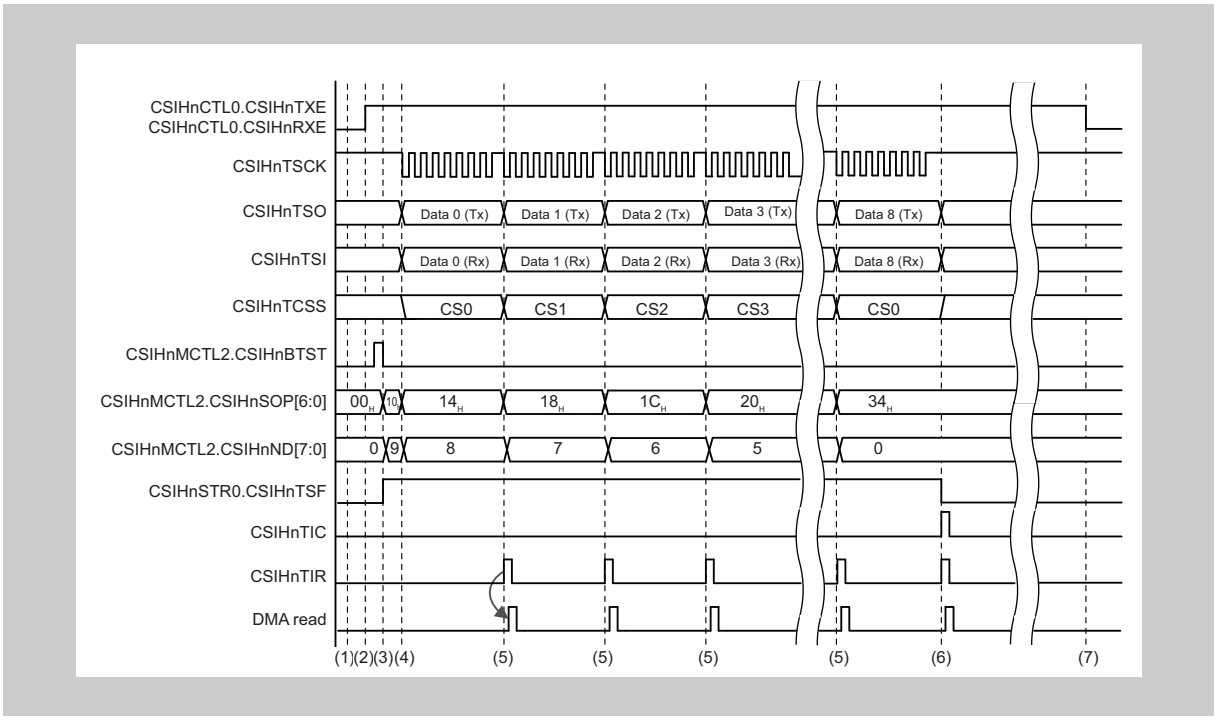


Figure 22-44 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

Note The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the registers CSIHnCFGx. This example uses the chip select signals CS0 to CS3.
   Specify the transfer and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission) and CSIHnRXE = 1 (enable reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
3. Configure the send pointer and the number of data by setting bits CSIHnMCTL2.SCIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented and bits CSIHnMCTL2.CSIHnND[7:0] decremented after each data. Transmission/reception is started.
5. After every data that has been transmitted, the interrupt request CSIHnTIR is generated. CSIHnTIR indicates that the reception register CSIHnRX0W must be read.
6. When all transmissions are complete, the interrupt request CSIHnTIC is generated.
7. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE.

(2) Transmit/Receive in Master Mode, Job Mode Enabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)

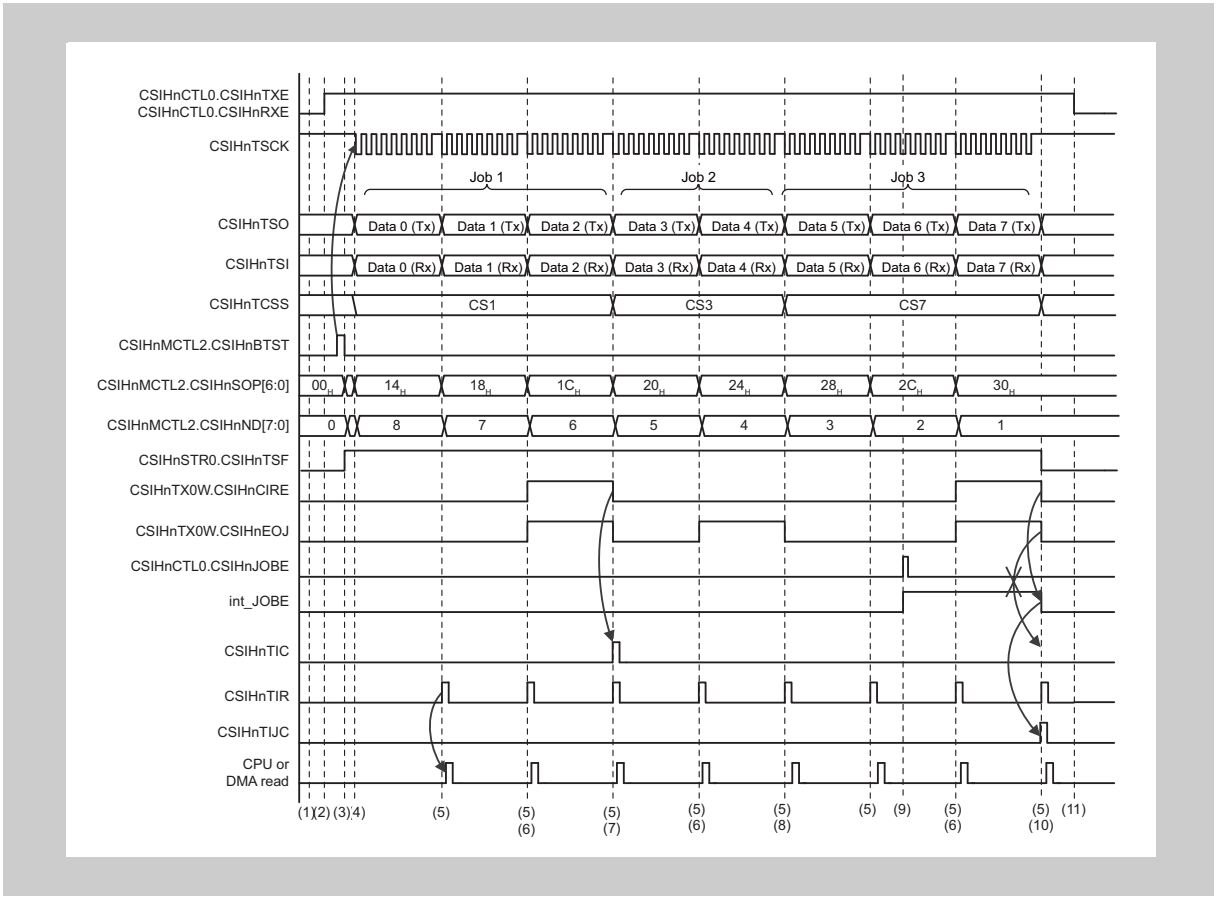


Figure 22-45 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

Note The label “int_JOBE” in the above timing diagram indicates the internal signal representing bit CSIHnJOBE. The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the registers CSIHnCFGx. This example uses the chip select signals CS1, CS3, and CS7.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission), and CSIHnRXE = 1 (enable reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
3. Configure the send pointer and the number of data by setting bits CSIHnMCTL2.SCIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented and CSIHnMCTL2.CSIHnND[7:0] decremented after each data.
5. After every data that has been transmitted, the interrupt request CSIHnTIR is generated. CSIHnTIR indicates that the reception register CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 specifies that the last data of the current job is sent.
7. The interrupt request CSIHnTIC is generated. CSIHnTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. The interrupt request CSIHnTIC is not generated, because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. By setting CSIHnCTL0.CSIHnJOBE = 1, the communication is forced to stop at the end of Job 3.
10. After the forced stop of communication, the interrupt requests CSIHnTIJC and CSIHnTIR are generated at the end of Job 3.

    The interrupt request CSIHnTIJC indicates a forced stop of communication at the end of the current job.

    The interrupt request CSIHnTIC is not generated because the interrupt request CSIHnTIJC is generated instead. Moreover, any usable data for transmission inside the CSIHnTX0W register are not transmitted.
11. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE.

22.5.3 Procedures in Double-Buffer Mode

Four examples are given, two for a master and two for a slave with job mode disabled and enabled.

(1) Transmit/Receive in Master Mode, Job Mode Disabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)

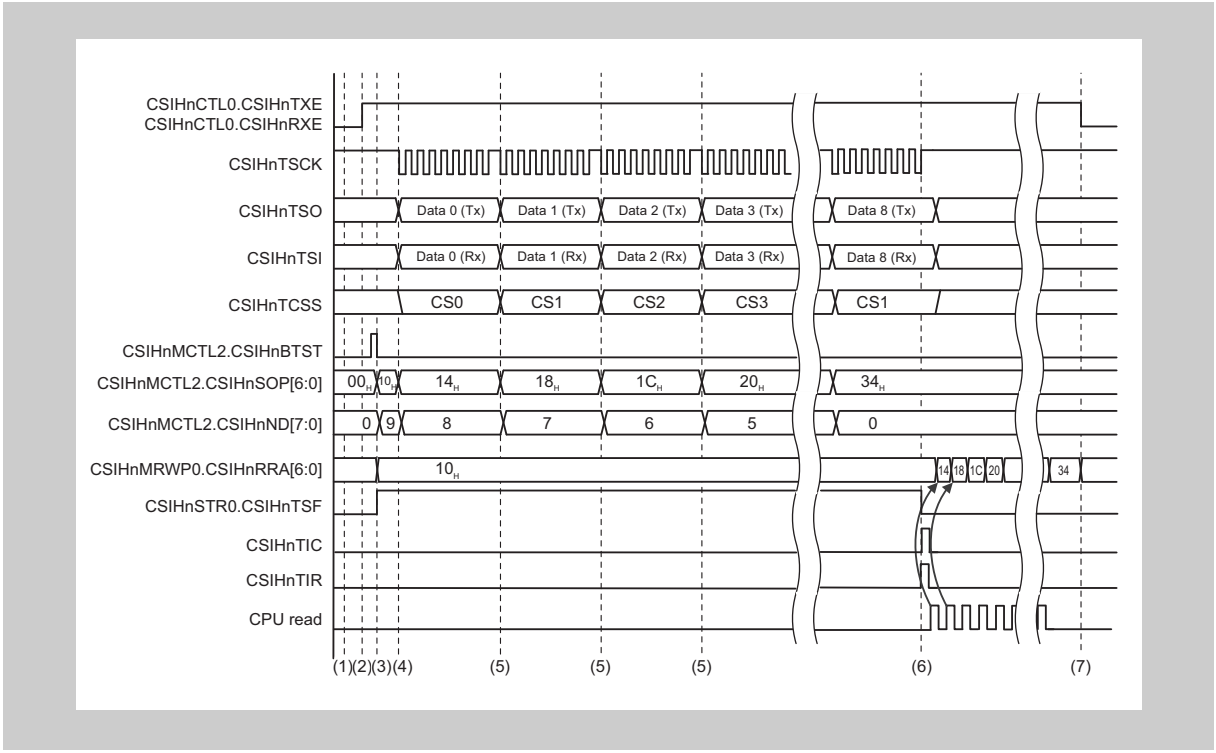


Figure 22-46 Master in Double-Buffer Mode, CSIHnCTL1.CSIHnJE = 0

Note The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the register CSIHnCFGx. The example uses the chip select signals CS0 to CS3.
   Specify the job mode disabled and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Set the memory mode by setting CSIHnMTCL0.CSIHnMMS[1:0] = 01_B (double-buffer mode).
   Set bit CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission) and CSIHnRXE = 1 (enable the reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
3. Configure the communication by setting CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Trigger the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started.
5. Bits CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented and bits CSIHnMCTL2.CSIHnND[7:0] decremented after each data. This is repeated until the last data is transmitted/received.
6. When the last data is transmitted/received, the interrupt requests CSIHnTIC and CSIHnTIR are generated.

   The CPU starts to read the received data from the Rx buffer. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after the reading of every data.
7. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE

(2) Transmit/Receive in Master Mode, Job Mode Enabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1)
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H)
- The transfer start address is 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H)

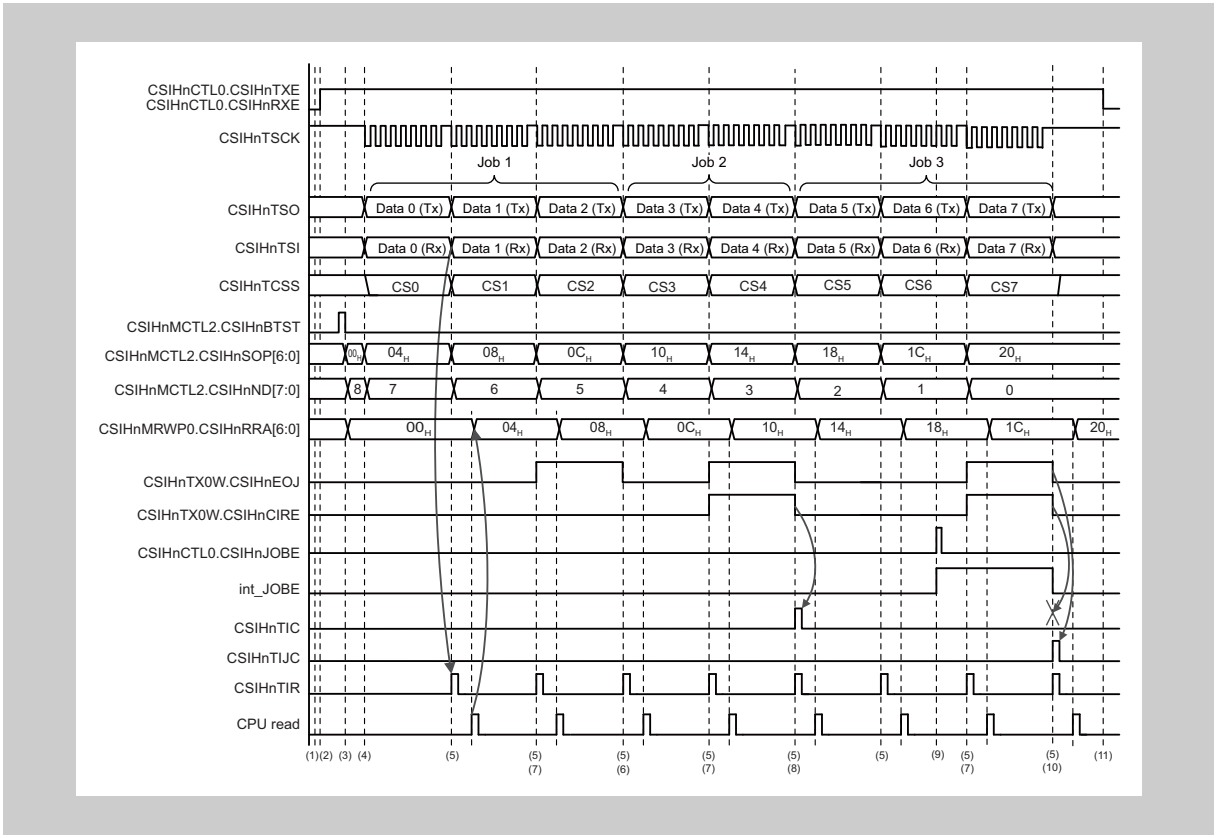


Figure 22-47 Master in Double-Buffer Mode, CSIHnCTL1.CSIHnJE = 1

Note The label “int_JOBE” in the above timing diagram indicates the internal signal representing bit CSIHnJOBE. The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the register CSIHnCFGx. The example uses the chip select signals CS0 to CS7.
   Specify the job mode enable and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Set the memory mode by setting CSIHnMTCL0.CSIHnMMS[1:0] = 01_B (double-buffer mode).
   Set bit CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission), and CSIHnRXE = 1 (enable reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
3. Configure the communication by setting bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started.
5. Bits CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented and bits CSIHnMCTL2.CSIHnND[7:0] decremented after each data. This is repeated until the last data is transmitted/received. The interrupt request CSIHnTIR is generated after every data.
6. The interrupt request CSIHnTIC is not generated, because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
7. CSIHnTX0W.CSIHnEOJ = 1 indicates that the data are the last for the job.
8. The interrupt request CSIHnTIC is generated. CSIHnTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
9. By setting CSIHnCTL0.CSIHnJOBE = 1, the communication is forced to stop at the end of Job 3.
10. After the forced stop of communication, the interrupt requests CSIHnTIJC and CSIHnTIR are generated at the end of Job 3.

    The interrupt request CSIHnTIJC indicates a forced stop of communication at the end of the current job.
    The interrupt request CSIHnTIC is not generated because the interrupt request CSIHnTIJC is generated instead. Additionally, the transmission data available in register CSIHnTX0W is not sent.
11. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE.

(3) Transmit/Receive in Slave Mode, Job Mode Disabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)

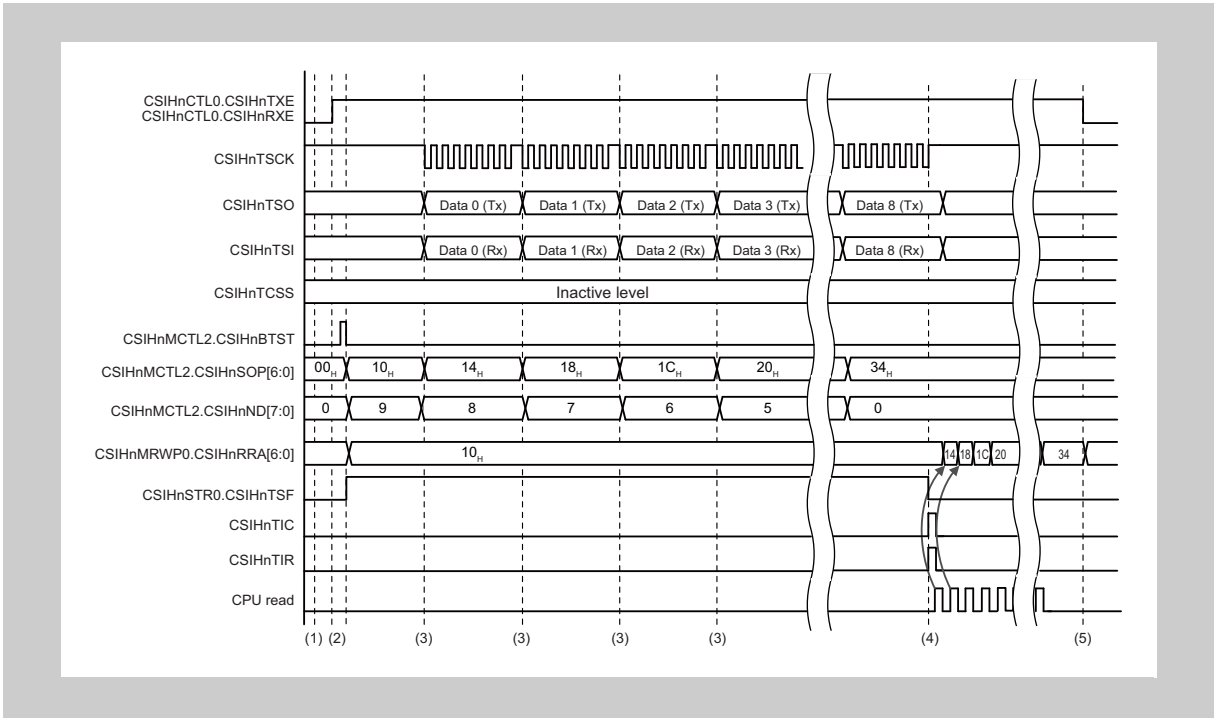


Figure 22-48 Slave in Double-buffer Mode, CSIHnCTL1.CSIHnJE = 0

Note The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in register CSIHnCFG0.
   Specify job mode disabled and slave mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Set CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
2. In the CSIHnCTL0 register set the bits CSIHnPWR = 1 (enable the clock), CSIHnTXE = 1 (enable transmission) and CSIHnRXE = 1 (enable reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.

   Specify the transfer start address by setting CSIHnMCTL2.CSIHnSOP[6:0] and the number of data by setting CSIHnMCTL2.CSIHnND[7:0]. Enable the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
   Transmission is started when the input clock from the master is received.
3. Bits CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented and bits CSIHnMCTL2.CSIHnND[7:0] decremented after each data. This is repeated until the last data is transmitted/received.

   CSIHnTIC and CSIHnTIR interrupt requests are not generated because data are being transmitted from or stored in the buffers.
4. When the last data is transmitted/received, the interrupt requests CSIHnTIC and CSIHnTIR are generated.

   The CPU starts to read the received data that is stored in the Rx buffer. The start address of the read access is specified in bits CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after the reading of every data.
5. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE.

22.5.4 Procedures in FIFO Mode

Two examples for a master are given, one with job mode disabled, one with job mode enabled.

(1) Transmit/Receive in Master Mode, Job Mode Disabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)

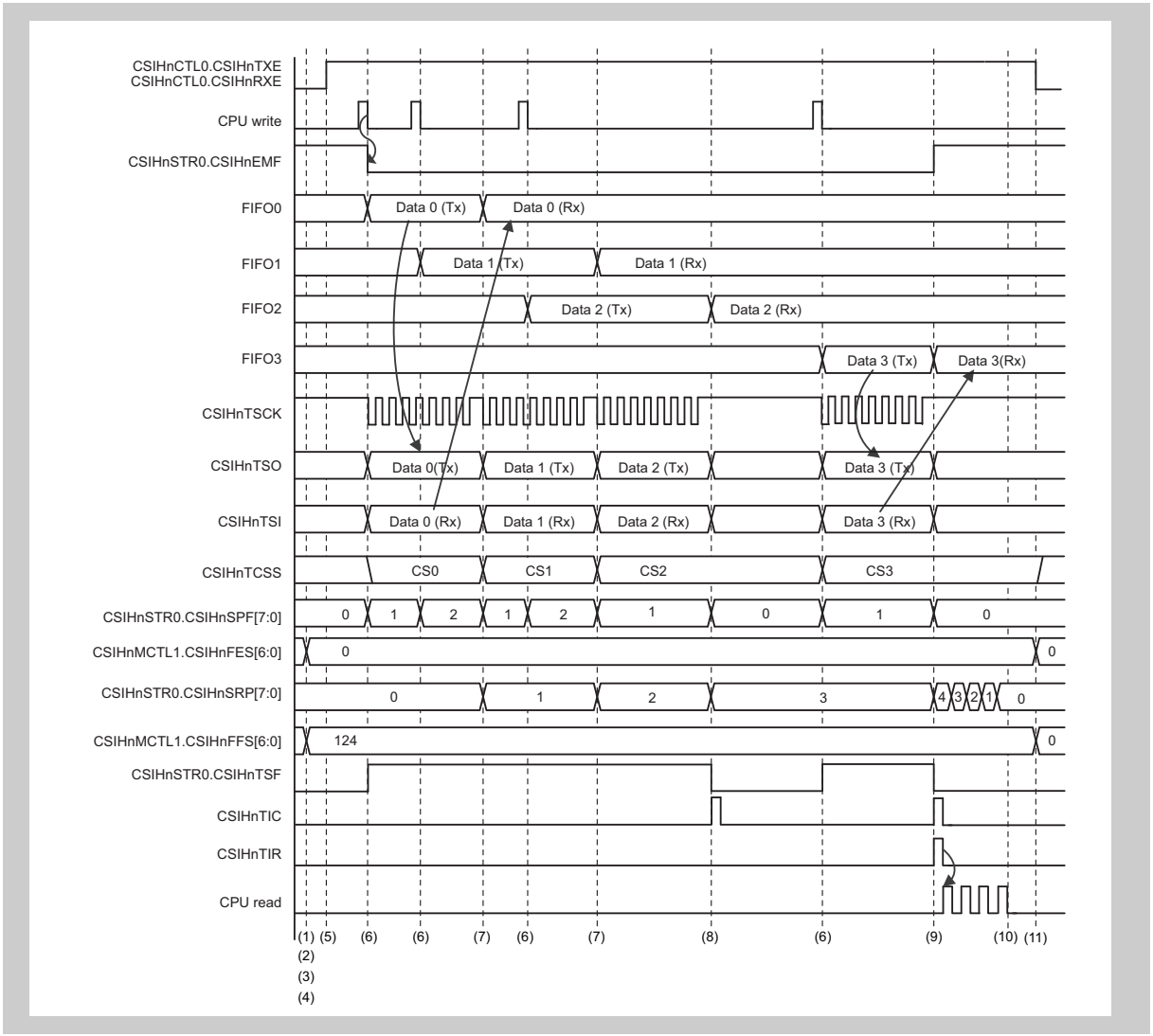


Figure 22-49 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Set up the communications protocol in the CSIHnCFGx register, set the corresponding bits in the CSIHnCTL1 and CSIHnCTL2 registers to disable job mode and select master mode, then select FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] = 00_B. In this example, the chip select signals CS0 to CS3 are used.
2. Set bit CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
3. Confirm the following values: CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. The CSIHnFES[6:0] bit field in register CSIHnMCTL1 specifies the conditions for output of the CSIHnTIC interrupt.

   The CSIHnFFS[6:0] field in the same register specifies the conditions for output of the CSIHnTIR interrupt.
5. Set CSIHnCTL0.CSIHnPWR = 1 (enabling the clock), CSIHnTXE = 1 (enabling transmission), and CSIHnRXE = 1 (enabling reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
6. Write the first packet of data for transmission to the transmission register (CSIHnTX0W). Transmission starts automatically once the first of the data are usable. Confirm that CSIHnSTR0.CSIHnEMF = 0.
7. The current round of transmission is complete. Since CSIHnFES[6:0] = CSIHnSPF[7:0] does not hold, a CSIHnTIC interrupt request is not generated.
8. CSIHnFES[6:0] = CSIHnSPF[7:0] now holds, so a CSIHnTIC interrupt request is generated.
9. A CSIHnTIR interrupt request is generated once CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0] is satisfied. After the interrupt is generated, the CPU starts reading the received data stored in the reception buffer.
10. Once the CPU has finished reading the received data from the reception buffer, the FIFO becomes empty and the CSIHnSTR0.CSIHnEMF bit is set to 1 accordingly.
11. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE. In order to minimize the power consumption of CSIHn while it is not used, set also CSIHnCTL0.CSIHnPWR = 0.

(2) Transmit/Receive in Master Mode, Job Mode Enabled

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- MSB is transmitted first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1)
- Job 1 = 4 data, Job 2 = 3 data, Job 3 = 5 data

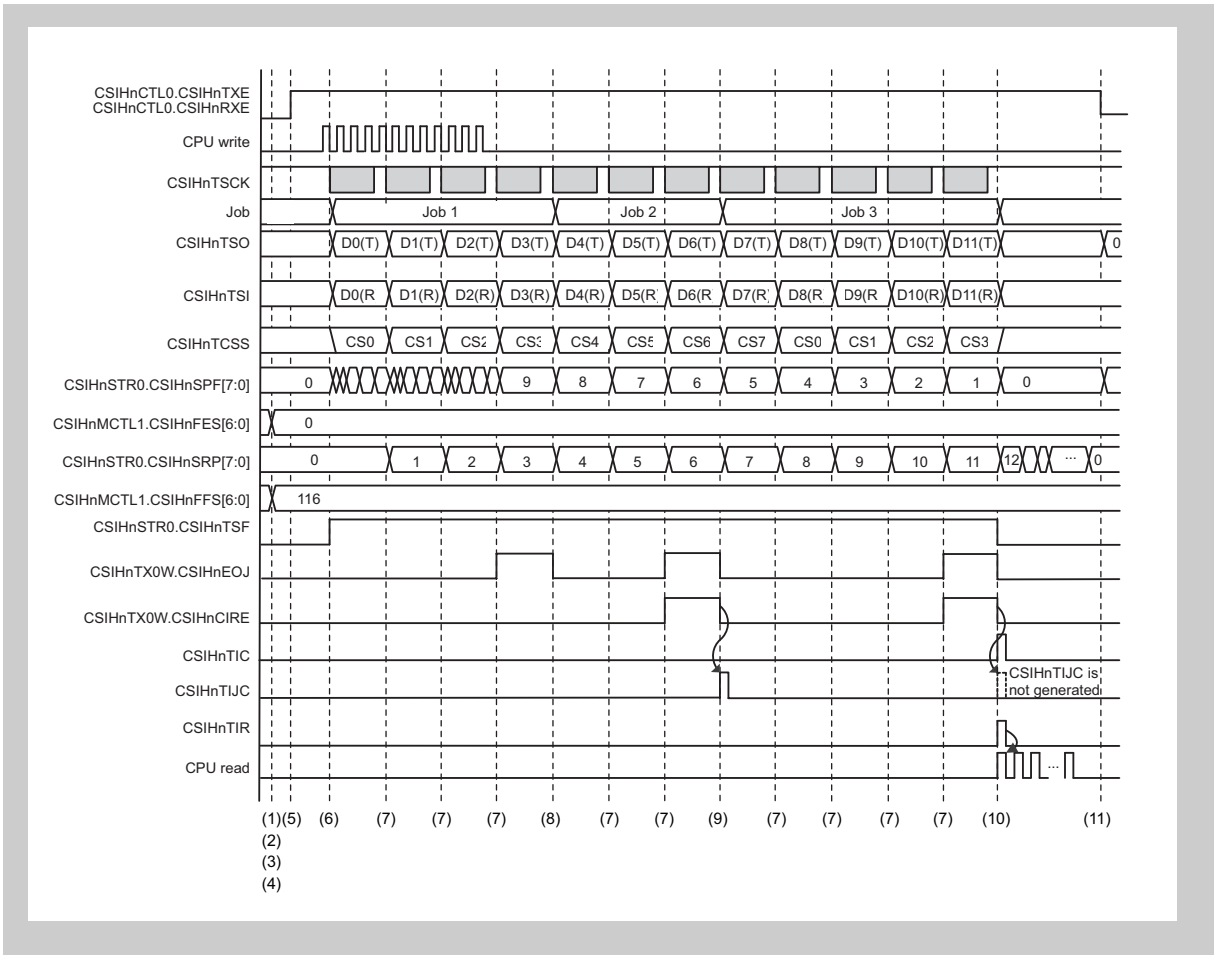


Figure 22-50 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Set up the communications protocol in the CSIHnCFGx register, select FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] = 00_B and set the corresponding bits in the CSIHnCTL1 and CSIHnCTL2 registers to enable job mode and select master mode. In this example, the chip select signals CS0 to CS7 are used.
2. Set bit CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
3. Confirm the following values: CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. The CSIHnFES[6:0] bit field in register CSIHnMCTL1 specifies the conditions for output of the CSIHnTIC interrupt. The CSIHnFFS[6:0] field in the same register specifies the conditions for output of the CSIHnTIR interrupt.
5. Set CSIHnCTL0.CSIHnPWR = 1 (enabling the clock), CSIHnTXE = 1 (enabling transmission), and CSIHnRXE = 1 (enabling reception). The CSIHnCTL0.CSIHnMBS bit must also be cleared.
6. Write the first packet of data for transmission to the transmission register (CSIHnTX0W). Transmission starts automatically once the first of the data are usable. Confirm that CSIHnSTR0.CSIHnEMF = 0.
7. The current round of transmission is complete. Since CSIHnFES[6:0] = CSIHnSPF[7:0] does not hold, a CSIHnTIC interrupt request is not generated.
8. Setting CSIHnTX0W.CSIHnEOJ = 1 specifies that the last data of the current job are sent.

   The interrupt request CSIHnTIC is not generated, because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. Since the last of the data have been transmitted (indicated by CSIHnTX0W.CSIHnEOJ = 1) with CSIHnTX0W.CSIHnCIRE = 1, a CSIHnTIJC interrupt request is generated.
10. A CSIHnTIC interrupt request is generated once CSIHnMCTL1.CSIHnFES[6:0] = CSIHnSTR0.CSIHnSPF[7:0]. Since this CSIHnTIC interrupt is generated, a CSIHnTIJC interrupt is not. A CSIHnTIR interrupt request is generated once CSIHnMCTL0.CSIHnFFS[6:0] = 128 - CSIHnSTR0.CSIHnSRP[7:0] is satisfied. After the interrupt is generated, the CPU starts reading the received data stored in the reception buffer.
11. Finally, disable transmission and reception by clearing CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE. In order to minimize the power consumption of CSIHn while it is not used, set also CSIHnCTL0.CSIHnPWR = 0.

Section 23 Synchronous/Asynchronous Serial Interface H (UARTH)

23.1 UARTHn Features

Instances This product has 3 instances of the synchronous/asynchronous serial interface H (UARTHn).

Table 23-1 Instances of UARTHn

Synchronous/Asynchronous Serial Interface H	
Instances	3
Name	UARTHn

Instances index n Throughout this section, the instance of a synchronous/asynchronous serial interface H is identified by the index "n" (n = 0 to 2), for example, URTHnCTL0 for the UARTHn control register 0.

Register addresses All UARTHn register addresses are given as offsets from the individual base addresses, <URTHn_base0> and <URTHn_base1>.

The <URTHn_base0> and <URTHn_base1> addresses of each UARTHn are listed in the following table.

Table 23-2 UARTHn Register Base Addresses <URTHn_base0> and <URTHn_base1>

UARTHn	<URTHn_base0> Address	<URTHn_base1> Address
UARTH0	$FF5C0000_H$	$FFFFEA00_H$
UARTH1	$FF5D0000_H$	$FFFFEB00_H$
UARTH2	$FF5E0000_H$	$FFFFEC00_H$

Clock supply A single clock signal is input to each UARTHn as follows.

Table 23-3 UARTHn Clock Supply

UARTHn	UARTHn Clock	Connected to
UARTH0	PCLK	Clock controller
UARTH1		
UARTH2		

I/O signals The I/O signals of the UARTHn are listed in the table below.

Table 23-4 UARTH I/O Signals

UARTHn Signals	Function	Connected to
URTHnTXD	Transmit data output	Port URTHnTXD
URTHnRXD	Receive data input	Port URTHnRXD
URTHnSC	Synchronizing clock input/output	Port URTHnSC
URTHnRTS	Handshake signal output	Port URTHnRTS
URTHnCTS	Handshake signal input	Port URTHnCTS

Interrupts Interrupts of UARTHn are listed in the table below.

Table 23-5 UARTHn Interrupts

UARTHn Signals	Function	Connected to
URTHnTIT	Transmission interrupt	Interrupt controller INTURTHnIT DMA, DTS
URTHnTIR	Reception interrupt	Interrupt controller INTURTHnIR DMA, DTS
URTHnTIS	Status interrupt	Interrupt controller INTURTHnIS DMA, DTS

Port Groups Use UARTHn pins from the groups listed below.

Table 23-6 Port Groups for UARTHn

UARTHn Channel	Pin Name	Group				
		1	2	3	4	5
UARTH0	URTH0SC	P0_2	P1_9	P5_2	P6_7	-
	URTH0TXD	P0_1	P1_8	P5_1	P6_8	-
	URTH0RXD	P0_0	P1_7	P5_0	P6_6	-
	URTH0CTS	P0_3	-	P5_3	-	-
	URTH0RTS	P0_4	-	P5_4	-	-
UARTH1	URTH1SC	P2_0	P6_4	-	-	-
	URTH1TXD	P3_1	P6_5	P4_1	P5_7	-
	URTH1RXD	P3_0	P6_3	P4_0	P5_6	P5_1*
	URTH1CTS	P4_3	P6_1	-	-	-
	URTH1RTS	P4_4	P6_2	-	-	-
UARTH2	URTH2SC	P4_0	-	-	-	-
	URTH2TXD	P3_6	P0_6	-	-	-
	URTH2RXD	P3_5	P0_5	P5_2*	-	-
	URTH2CTS	P4_1	-	-	-	-
	URTH2RTS	P4_2	-	-	-	-

Note * Reception only

23.2 Features

- Synchronous and asynchronous mode selectable
- Full-duplex communication:
  - Internal UARTHn receive data register n (URTHnRX)
  - Internal UARTHn transmit data register n (URTHnTX)
- 2-pin configuration:
  - URTHnTXD: Transmit data output pin
  - URTHnRXD: Receive data input pin
- Reception error and status output function
  - Parity error
  - Framing error
  - Overrun error
  - Data consistency error
- Interrupt sources: 3
  - Transmission interrupt URTHnTIT
  - Reception interrupt URTHnTIR
  - Status interrupt URTHnTIS
- Character length: 7, 8, or 9 bits (with the enhanced bit)
- Parity function: odd, even, 0, or none
- Transmission stop bit(s): 1 or 2
- MSB-/LSB-first transfer selectable
- Inversion of data for transmission and received data is available
- 13 to 20 bits selectable for the BF (break field) in the LIN (local interconnect network) communication format
  - Recognition of 11 bits or more possible for BF reception in LIN communication format
  - BF reception flag provided
- BF reception can be detected during data communication.
- Bus monitor function to keep data consistency of the transmit data
- Handshake mode (URTHnCTS pin, URTHnRTS pin: active level selectable)
- Transmission of 1 frame or 2 frames in succession
- Enhanced bit detection interrupt setting is available.
- Noise canceling function (for the URTHnRXD, URTHnCTS, and URTHnSC pins)
- Delayed sampling mode (in clock-synchronous mode)
- Baud-rate detection in operation as a slave device for LIN communications

23.3 Configuration

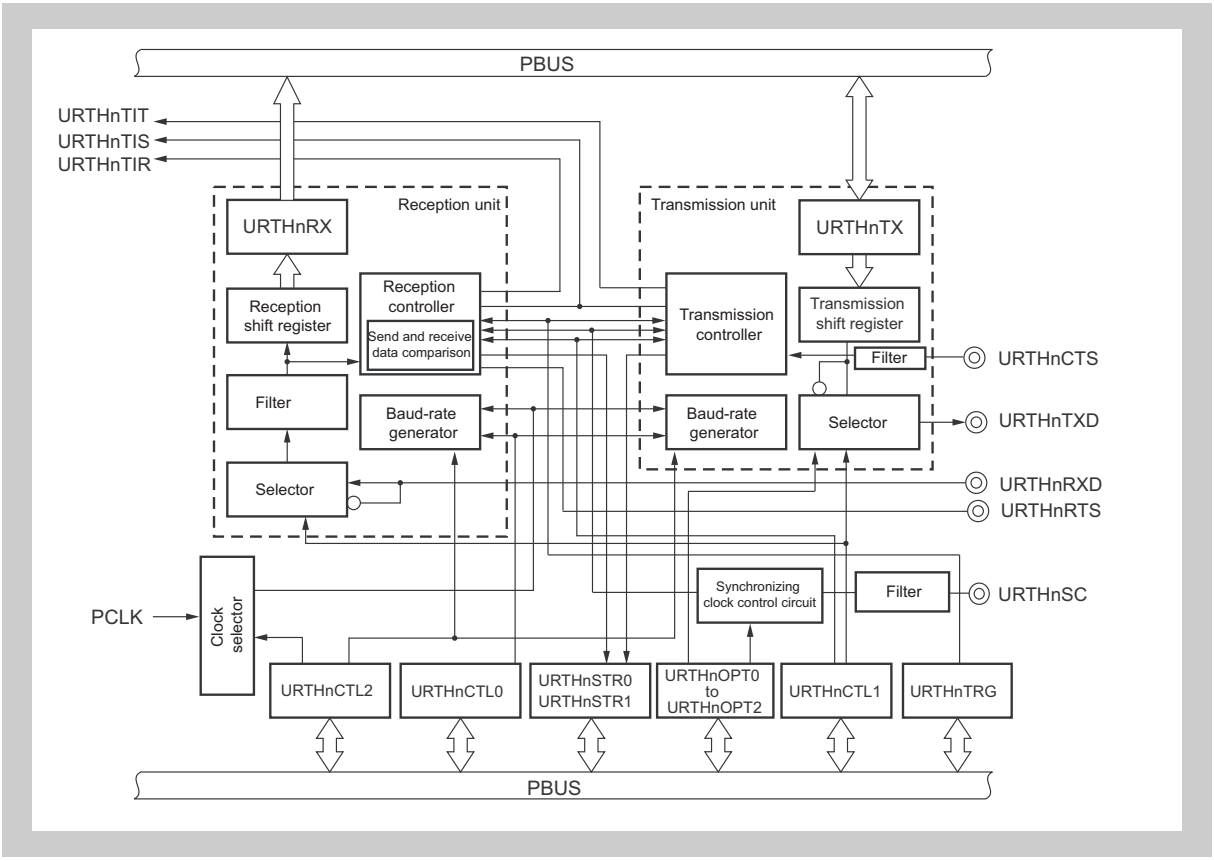


Figure 23-1 Block Diagram of Synchronous/Asynchronous Serial Interface (UARTHn)

23.4 UARTHn Registers

The UARTHn is controlled and operated by means of the following registers.

Table 23-7 UARTHn Registers

Register Function	Name	Address
Control register 0	URTHnCTL0	<URTHn_base1> + 00_H
Control register 1	URTHnCTL1	<URTHn_base0> + 40_H
Control register 2	URTHnCTL2	<URTHn_base0> + 44_H
Trigger register	URTHnTRG	<URTHn_base1> + $0C_H$
Status register 0	URTHnSTR0	<URTHn_base1> + 10_H
Status register 1	URTHnSTR1	<URTHn_base1> + 14_H
Status clear register	URTHnSTC	<URTHn_base1> + 18_H
Option register 0	URTHnOPT0	<URTHn_base0> + 48_H
Option register 1	URTHnOPT1	<URTHn_base1> + 04_H
Option register 2	URTHnOPT2	<URTHn_base1> + 08_H
Receive data register	URTHnRX	<URTHn_base1> + $1C_H$
Receive data register HL	URTHnRXHL	<URTHn_base1> + 20_H
Enhanced bit receive data register	URTHnERX	<URTHn_base1> + 24_H
Enhanced bit receive word data register	URTHnERXW	<URTHn_base1> + 28_H
Transmit data register	URTHnTX	<URTHn_base1> + $2C_H$
Transmit data register HL	URTHnTXHL	<URTHn_base1> + 30_H
Enhanced bit transmit data register	URTHnETX	<URTHn_base1> + 34_H
Enhanced bit transmit word data register	URTHnETXW	<URTHn_base1> + 38_H
LIN communications slave operation baud-rate detection register 0	IC0REG0	$FFFFFE00_H$
LIN communications slave operation baud-rate detection register 1	IC0REG1	$FFFFFE04_H$

<URTHn_base> For the base addresses of the UARTHn registers, i.e. <URTHn_base0> and <URTHn_base1>, refer to table 23-2, UARTHn Register Base Addresses <URTHn_base0> and <URTHn_base1>.

(1) URTHnCTL0 – UARTHn Control Register 0

This register controls the basic serial-transfer operations of the UARTHn.

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

Address <URTHn_base1> + 00_H

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
URTHn PW	URTHn TXE	URTHn RXE	0	0	0	0	URTHn SLDC
R/W	R/W	R/W	R	R	R	R	R/W

Table 23-8 URTHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	URTHnPW	UARTHn enable 0: Disable UARTHn operation 1: Enable UARTHn operation Changing this bit initializes all transmission and reception units.
6	URTHnTXE	Transmission operation enable 0: Disable transmission operation 1: Enable transmission operation • To start transmission, set URTHnPW to 1 and then set URTHnTXE to 1. To stop transmission, clear URTHnTXE to 0 and then URTHnPW to 0. Alternatively, clear URTHnPW and URTHnTXE simultaneously. • To initialize the transmission unit, clear URTHnTXE to 0, wait for 2 cycles of the PRSCLK clock, and then set URTHnTXE to 1 again.
5	URTHnRXE	Reception operation enable 0: Disable reception operation 1: Enable reception operation • To enable reception, set URTHnPW to 1 and then set URTHnRXE to 1. To disable reception, clear URTHnRXE to 0, or clear URTHnPW and URTHnRXE simultaneously. • To initialize the reception unit, clear URTHnRXE to 0, wait for 2 cycles of the PRSCLK clock, and then set URTHnRXE to 1 again. Setting URTHnRXE to 1 enables reception after 2 cycles of the PRSCLK clock have elapsed. 4 cycles of the base clock elapse from setting of URTHnRXE to 1 until the detection of edges on the URTHnRXD pin is enabled.
0	URTHnSLDC	Data consistency check enable/disable 0: Disable data consistency check 1: Enable data consistency check This bit selects the handling of checking for data consistency errors in data for transmission. When this bit is set to 1, the transmitted and received data are compared. If the two do not match, the URTHnSTR1.URTHnDCE flag is set to 1 and a status interrupt request (URTHnTIS) is issued. This bit is for access only when transmission is to be started. Consequently, if the value of this bit is changed during processing for transmission, the processing for transmission will continue with the same value as was set at the start of processing.

Caution 1. If UARTHn is in the following states:

- reception and transmission are enabled (URTHnPW = URTHnRXE = URTHnTXE = 1),
- data consistency checking is enabled (URTHnSLDC = 1), and
- transmission is ongoing or completed

Transmission shall be disabled while enabling of reception is maintained; in this case, follow the procedure below.

- Check that transmission has not been executed (URTHnSTR0.URTHnSSBT = URTHnSTR0.URTHnSST = 0).
- Disable transmission by setting URTHnCTL0.URTHnTXE = 0.

This procedure is required because the data consistency error flag URTHnSTR1.URTHnDCE is cleared to 0 in response to clearing of URTHnTXE to 0. Thus a data consistency error would not be reported if it occurred in a case where transmission is disabled during transfer or after its completion.

Caution 2. If UARTHn is in the following states:

- reception and transmission are enabled (URTHnPW = URTHnRXE = URTHnTXE = 1),
- data consistency checking is enabled (URTHnSLDC = 1), and
- transmission is ongoing or completed

Reception shall be disabled while enabling of transmission is maintained; in this case, follow the procedure below.

- Check that transmission has not been executed (URTHnSTR0.URTHnSSBT = URTHnSTR0.URTHnSST = 0).
- Disable reception by setting URTHnCTL0.URTHnRXE = 0.

The reason for this procedure is that the data consistency error flag URTHnSTR1.URTHnDCE is cleared to 0 and becomes invalid in response to the clearing of URTHnRXE to 0. Following the procedure above leads to non-reporting of data consistency errors in data that have already been transmitted.

Caution 3. Don't start any data transmission if:

- data consistency checking is enabled (URTHnSLDC = 1),
- BF reception is enabled (URTHnSTR0.URTHnSSBR = 1), and
- detection of a BF during reception is not enabled (URTHnCTL1.URTHnSLBM = 0)

Under the above conditions, a data-consistency error will occur on completion of BF reception. The status interrupt signal (URTHnTIS) will be asserted and the completion of BF reception will not be reported (the URTHnSTR1.URTHnBSF flag remains 0). Consequently, the completion of BF reception will not be recognized.

(2) URTHnCTL1 – UARTHn Control Register 1

This register defines the data frame properties of the UARTHn serial data transfers.

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

Address <URTHn_base0> + 40_H

Initial value 5002_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
URTHn SLBM	URTHnBLG[2:0]			0	0	0	URTHn CLG
R/W	R/W	R/W	R/W	R	R	R	R/W

7	6	5	4	3	2	1	0
URTHnSLP[1:0]		URTHn TDL	URTHn RDL	URTHn LPM	URTHn SLG	URTHn SLD	URTHn SLIT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-9 URTHnCTL1 Register Contents (1/3)

Bit Position	Bit Name	Function
15	URTHnSLBM	BF reception mode selection 0: BF reception during data reception disabled. 1: BF reception during data reception enabled. • Changing this bit is only allowed while reception is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnRXE = 0).
14 to 12	URTHnBLG [2:0]	BF bit length during transmission URTHnBLG2 / URTHnBLG1 / URTHnBLG0 / BF Length 1 / 0 / 1 / 13 bits 1 / 1 / 0 / 14 bits 1 / 1 / 1 / 15 bits 0 / 0 / 0 / 16 bits 0 / 0 / 1 / 17 bits 0 / 1 / 0 / 18 bits 0 / 1 / 1 / 19 bits 1 / 0 / 0 / 20 bits Changing these bits is only allowed while transmission is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnTXE = 0).
8	URTHnCLG	Bit length of transmit/receive data 0: 7 bits 1: 8 bits or 9 bits (enhanced bit) • When the transmission/reception is performed in the LIN format, set URTHnCTL1.URTHnCLG to 1. • Changing this bit is only allowed while reception and transmission are disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnRXE = URTHnCTL0.URTHnTXE = 0).

Table 23-9 URTHnCTL1 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>URTHnSLP [1:0]</td><td>Parity bit selection
<table>
<tr><th rowspan="2">URTHnSLP1</th><th rowspan="2">URTHnSLP0</th><th colspan="2">Operation</th></tr>
<tr><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Output without parity bit</td><td>Reception with no parity judgment</td></tr>
<tr><td>0</td><td>1</td><td>Parity bit output is for zero parity (the setting is fixed to 0).</td><td>No parity judgment</td></tr>
<tr><td>1</td><td>0</td><td>Output with odd parity</td><td>Judged as odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Output with even parity</td><td>Judged as even parity</td></tr>
</table>
• If "Reception with no parity judgment" is selected during reception, a parity check is not performed. Consequently, the URTHnSTR1.URTHnPE bit will not be set, so an error interrupt will not be output.
• When the transmission/reception is performed in the LIN format, set URTHnSLP[1:0] to 00_B.
• Changing these bits is only allowed while reception and transmission are disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnRXE = URTHnCTL0.URTHnTXE = 0).</td></tr>
<tr><td>5</td><td>URTHnTDL</td><td>Transmit data level control
0: No inverted output of transmit data
1: Inverted output of transmit data
• This bit can be used to invert the levels output on the URTHnTXD pin. Changing the bit inverts the URTHnTXD output level immediately, regardless of the values of URTHnCTL0.URTHnPW and URTHnCTL0.URTHnTXE. Therefore, if URTHnTDL is set to 1 while operations are disabled, the URTHnTXD outputs low level.
• Changing this bit is only allowed while transmission is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnTXE = 0).</td></tr>
<tr><td>4</td><td>URTHnRDL</td><td>Received data level control
0: No inverted input of received data
1: Inverted input of received data
• The input level of the URTHnRXD pin can be inverted using this bit. It inverts the URTHnRXD input level immediately, regardless of the values of URTHnCTL0.URTHnPW and URTHnCTL0.URTHnRXE. Therefore, if URTHnRDL is set to 1 while the operations are disabled, the URTHnRXD inputs low level.
• Changing this bit is only allowed while reception is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnRXE = 0).</td></tr>
<tr><td>3</td><td>URTHnLPM</td><td>Loop back mode control
0: Loop back of transmit data disabled
1: Loop back of transmit data enabled
• This bit can be used to connect the URTHnTXD output to the URTHnRXD input. When this is done, the URTHnTXD output is fixed to "H" if URTHnTDL = 0 and to "L" if URTHnTDL =1.
• If URTHnLPM = 1, disable handshake mode (URTHnOPT0.URTHnHS = 0).
• Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).</td></tr>
</table>

Table 23-9 URTHnCTL1 Register Contents (3/3)

Bit Position	Bit Name	Function
2	URTHnSLG	Stop bit number selection for transmit data 0: 1 bit 1: 2 bits • The stop bit length during data or BF reception is always handled as "1". • To use this register with data consistency checking enabled (URTHnCTL0.URTHnSLDC = 1) during slave operation, set URTHnSLG = 1. • Changing this bit is only allowed while transmission is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnTXE = 0).
1	URTHnSLD	Transfer direction selection 0: MSB-first transfer 1: LSB-first transfer • When the transmission/reception is performed in the LIN format, set URTHnSLD to 1. • Changing this bit is only allowed while transmission and reception are disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnRXE, URTHnCTL0.URTHnTXE = 0).
0	URTHnSLIT	Transmission interrupt request (URTHnTIT) timing selection 0: URTHnTIT is generated when transmission starts after the data for transmission have been stored in the transmission shift register. 1: URTHnTIT is generated at transmission completion. • Changing this bit is only allowed while transmission is disabled (URTHnCTL0.URTHnPW = 0 or URTHnCTL0.URTHnTXE = 0).

(3) URTHnCTL2 – UARTHn Control Register 2

This register defines the baud rate for serial data transfer by UARTHn.

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

Address <URTHn_base0> + 44_H

Initial value $EFFF_H$

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
URTHnPRS[2:0]			0	URTHnBRS[11:8]			
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTHnBRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-10 URTHnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	URTHnPRS [2:0]	Prescaler clock (PRSCLK) division value 0: PRSCLK = PCLK/2^0 1: PRSCLK = PCLK/2^1 2: PRSCLK = PCLK/2^2 3: PRSCLK = PCLK/2^3 4: PRSCLK = PCLK/2^4 5: PRSCLK = PCLK/2^5 6: PRSCLK = PCLK/2^6 7: PRSCLK = PCLK/2^7
11 to 0	URTHnBRS [11:0]	Baud-rate clock (BRCLK) division value

URTHn BRS[11:0]	Transmit/Receive BRCLK	BF Receive Clock
000_H	PRSCLK/(2 × 2)	PRSCLK/2
001_H		
002_H		
003_H	PRSCLK/(2 × 3)	PRSCLK/3
004_H	PRSCLK/(2 × 4)	PRSCLK/4
005_H	PRSCLK/(2 × 5)	PRSCLK/5
...	PRSCLK/ (2 × URTHnBRS[11:0])	PRSCLK/ URTHnBRS[11:0]
FFE_H	PRSCLK/(2 × 4094)	PRSCLK/4094
FFF_H	PRSCLK/(2 × 4095)	PRSCLK/4095

Caution Writing to this register is only allowed while the operation of UARTHn is disabled (URTHnCTL0.URTHnPW = 0).

Clock supply The value of the UARTHn input clock is defined in table 23-3, UARTHn Clock Supply.

(4) URTHnTRG – UARTHn Trigger Register

This register controls use of the BF as a trigger for transmission or reception by UARTHn.

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

Address <URTHn_base1> + $0C_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	URTHn BRT	URTHn BTT	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 23-11 URTHnTRG Register Contents (1/2)

Bit Position	Bit Name	Function
6	URTHnBRT	BF reception trigger 0: The value read is always 0. Writing 0 is ignored. 1: Enables BF reception. • When reception is enabled, writing 1 to this bit enables BF reception (URTHnSTR0.URTHnSSBR = 1) and BF reception processing begins when the falling edge of the receive serial signal is detected. • If 1 is written to this bit during reception processing, the current reception processing is terminated. Consequently, the received value is not stored, the framing, parity and overflow error bits are not updated on the basis of the received data, and no interrupts are generated. Meanwhile, the value of the BF counter stays in continuous use. • After BF reception, the reception status is set according to the URTHnCTL1.URTHnSLBM setting. • Setting this bit to 1 is only allowed while reception is enabled (URTHnCTL0.URTHnPW = URTHnCTL0.URTHnRXE = 1). After URTHnBRT is set to 1, completion of BF reception is reported by either of the following 2 methods, based on the URTHnCTL1.URTHnSLBM setting: After setting the URTHnBRT bit to 1, do not write 1 to the URTHnBRT bit again until a reception-complete interrupt is generated. • If URTHnCTL1.URTHnSLBM = 0, a reception interrupt request URTHnTIR is output on completion of BF reception. • If URTHnCTL1.URTHnSLBM = 1, when BF reception is complete, URTHnSTR1.URTHnBSF is set to 1 and a status interrupt request (URTHnTIS) is output on completion of BF reception.

Table 23-11 URTHnTRG Register Contents (2/2)

Bit Position	Bit Name	Function
5	URTHnBTT	BF transmission trigger 0: The value read is always 0. Writing 0 is ignored. 1: Triggers BF transmission. • Writing 1 to this bit while URTHnSTR0.URTHnSSBT = 0 and transmission is enabled (URTHnSTR1.URTHnDCE = 0) causes a BF transmission request to be issued and URTHnSTR0.URTHnSSBT to be set to 1. • Writing 1 to this bit during data transmission leads to the transmission of a BF after the transmission in progress is completed. Even if 1 is again written to this bit before BF transmission is completed, the BF will only be transmitted once. • When transmission is enabled (URTHnCTL0.URTHnPW = URTHnCTL0.URTHnTXE = 1), writing 1 to this bit clears all previously set data transmission requests (for which the data have not been transmitted), leaving only any BF transmission request. If a new value is written to the URTHnTX.URTHnTX[7:0] bits after 1 is written to this bit, the data are transmitted after the BF. • If both a BF transmit request and a data transmit request have been set when a transmission starts, the BF transmission takes priority. • When URTHnSTR1.URTHnDCE = 1, writing 1 to this bit is ignored. • Setting this bit is only allowed when transmission is enabled (URTHnCTL0.URTHnPW = 1, URTHnCTL0.URTHnTXE = 1).

(5) URTHnSTR0 – UARTHn Status Register 0

This register indicates the current status of serial data transmissions.

Access This register can be read in 8- or 1-bit units. This register is writable if UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). If this is enabled (URTHnCTL0.URTHnPW = 1), any written values are disregarded and the register takes on its initial value.

Address <URTHn_base1> + 10_H

Initial value 00_H

A reset from any source or, when the value of URTHnCTL0.URTHnPW is changed from 0 to 1 or from 1 to 0, will initialize the bits.

7	6	5	4	3	2	1	0
0	URTHn SSBR*1	URTHn SSBT*2	0	0	0	URTHn SSR*1	URTHn SST*2
R	R	R	R	R	R	R	R

Note 1. These bits are also initialized if reception is disabled by URTHnCTL0.URTHnRXE = 0.

Note 2. These bits are also initialized if transmission is disabled by URTHnCTL0.URTHnTXE = 0.

Table 23-12 URTHnSTR0 Register Contents

Bit Position	Bit Name	Function
6	URTHnSSBR	BF reception enable/disable status indication 0: BF reception disabled 1: BF reception has been enabled by setting URTHnTRG.URTHnBRT to 1 (BF reception standby mode or BF reception busy).
5	URTHnSSBT	BF transmission enable/disable status indication 0: BF transmission disabled 1: BF transmission has been enabled by setting URTHnTRG.URTHnBTT to 1 (BF transmission standby mode or BF transmission busy).
1	URTHnSSR	Data reception state 0: Data reception is not in progress. 1: Data reception is in progress (data reception busy)
0	URTHnSST	Data transmission state 0: No transmission is pending or ongoing 1: Transfer of data in URTHnTX.URTHnTX[7:0] is pending or transmission is in progress

(6) URTHnSTR1 – UARTHn Status Register 1

This register indicates results of serial data transmissions.

Access This register can be read in 8- or 1-bit units. This register is writable if UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). If this is enabled (URTHnCTL0.URTHnPW = 1), any written values are disregarded and the register takes on its initial value.

Address <URTHn_base1> + 14_H

Initial value 00_H

A reset from any source or, when the value of URTHnCTL0.URTHnPW is changed from 0 to 1 or from 1 to 0, will initialize the bits.

7	6	5	4	3	2	1	0
0	URTHn IDM*1	URTHn EBD*1	URTHn BSF*1	URTHn DCE*2	URTHn PE*1	URTHn FE*1	URTHn OVE*1
R	R	R	R	R	R	R	R

Note 1. These bits are also initialized if reception is disabled by URTHnCTL0.URTHnRXE = 0.

Note 2. This bit is also initialized if transmission is disabled by URTHnCTL0.URTHnTXE = 0.

Table 23-13 URTHnSTR1 Register Contents (1/2)

Bit Position	Bit Name	Function
6	URTHnIDM	ID value match detection flag 0: The ID value of the received data does not match. 1: ID value match with the received data detected • This bit is set to 1 if ID value matched when ID comparison is enabled (URTHnOPT1.URTHnIDCN = 1) by transfer of enhanced bit (9th bit), and the enhanced bit detection interrupt is enabled (URTHnOPT0.URTHnEGE = 1). • For details on ID match detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.
5	URTHnEBD	Enhanced bit detection flag 0: The received data do not have an enhanced bit. 1: Detects an enhanced bit of the receive data • When the transmission of the enhanced bit (9th bit) is enabled (URTHnOPT0.URTHnEBE = 1) and the enhanced bit detection interrupt is enabled (URTHnOPT0.URTHnEGE = 1), this bit is set to 1 if an enhanced bit is detected. • For details on enhanced bit detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.
4	URTHnBSF	BF reception successful flag 0: No BF was received while URTHnSLBM = 1. 1: BF reception completed successfully when URTHnSLBM = 1. When the BF reception mode selection bit (URTHnCTL1.URTHnSLBM) is set in LIN communication mode, it is necessary to read this bit by the status interrupt processing and to confirm the beginning of a new frame slot.
3	URTHnDCE	Data consistency error flag 0: No inconsistency between transmitted and received data (in terms of the transmitted and received BF). 1: Inconsistency between transmitted and received data (in terms of the transmitted and received BF) detected.

Table 23-13 URTHnSTR1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	URTHnPE	Parity error flag 0: No parity error in the received data 1: Parity error in the received data detected. The operation of URTHnSTR.URTHnPE is controlled by the settings of the URTHnCTL1.URTHnSLP[1:0]. Note: In consecutive two-frame reception mode, this reflects the state of reception of 2 frames being completed.
1	URTHnFE	Framing error flag 0: No framing error in the received data 1: Framing error in the received data detected. Note: In consecutive two-frame reception mode, this reflects the state of reception of the two frames being completed.
0	URTHnOVE	Overrun error flag 0: No overrun error in the received data 1: Overrun error in the received data detected. When an overrun error occurs, the data are discarded, and the next received data are not written to the receive data register (URTHnRX).

Note If the bits of this register are set (1) and cleared (0) at the same time, setting takes priority over clearing.

Caution For details on error detection, see section 23.6.9, Transmission Data Consistency Check, and section 23.6.13, Reception Errors. In case reception and transmission are enabled and data consistency checking reveals an error (URTHnSTR1.URTHnDCE = 1), follow the procedure below before transmitting further data:

- Disable transmission by setting URTHnCTL0.URTHnTXE = 0.
- Enable transmission by setting URTHnCTL0.URTHnTXE = 1.
- Initiate transmission by setting URTHnTRG.URTHnBTT = 1 (BF transmission trigger) or writing any value to URTHnTX.

Further transmission can now proceed.

(7) URTHnSTC – UARTHn Status Clear Register

This register is used to clear the status bits of the status register 1 (URTHnSTR1).

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

Reading this register always returns 00_H.

Address <URTHn_base1> $+18_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
0	URTHn CLIM	URTHn CLEB	URTHn CLBS	URTHn CLDC	URTHn CLP	URTHn CLF	URTHn CLOV
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-14 URTHnSTC Register Contents

Bit Position	Bit Name	Function
6	URTHnCLIM	Controls clearing of the ID value match detection flag (URTHnSTR1.URTHnIDM bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnIDM.
5	URTHnCLEB	Controls clearing of the enhanced bit detection flag (URTHnSTR1.URTHnEBD bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnEBD.
4	URTHnCLBS	Controls clearing of the BF reception successful flag (URTHnSTR1.URTHnBSF bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnBSF.
3	URTHnCLDC	Controls clearing of the data consistency error flag (URTHnSTR1.URTHnDCE bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnDCE. Clearing URTHnSTR1.URTHnDCE by setting this bit to 1 will initiate the next transmission. Accordingly, take care when using this in LIN communications.
2	URTHnCLP	Controls clearing of the parity error flag (URTHnSTR1.URTHnPE bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnPE.
1	URTHnCLF	Controls clearing of the framing error flag (URTHnSTR1.URTHnFE bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnFE.
0	URTHnCLOV	Controls clearing of the overrun error flag (URTHnSTR1.URTHnOVE bit). 0: Writing 0 is ignored. 1: Writing 1 clears URTHnSTR1.URTHnOVE.

(8) URTHnOPT0 – UARTHn Option Register 0

This register specifies the optional features for use in serial transfer by the UARTHn interface.

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

Address <URTHn_base0> + 48_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
URTHn EBE	URTHn EGE	URTHn EJL	URTHn SCFR	URTHn RSFR	URTHn RXFR	URTHn RTSL	URTHn CTSL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
0	URTHn HS	URTHn DM	URTHn MSS	URTHn SAS	URTHn F2R	URTHn F2T	0
R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 23-15 URTHnOPT0 Register Contents (1/3)

Bit Position	Bit Name	Function
15	URTHnEBE	Enables and disables operation with the enhanced bit 0: Disables enhanced bit operation The bit length set for transmission and reception (URTHnCTL1.URTHnCLG) is used in transmitting and receiving data. 1: Enables enhanced bit operation The character length used in transmitting and receiving data is 9 bits (an enhanced bit is appended to each 8-bit character for transmission and reception). • The setting of this bit is only valid when the setting of the bit length for transmission and reception is 8 bits (URTHnCTL1.URTHnCLG = 1). The setting becomes invalid when the length is 7 bits (URTHnCTL1.URTHnCLG = 0), in which case data are transmitted and received with a character length of 7 bits. • When the transmission/reception is performed in the LIN format, set URTHnEBE to 0. • Changing this bit is only allowed while the UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). • For details on enhanced bit detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.

Table 23-15 URTHnOPT0 Register Contents (2/3)

Bit Position	Bit Name	Function
14	URTHnEGE	Enables and disables the enhanced bit detection interrupts. 0: Disables enhanced bit detection interrupts Detection of an enhanced bit under the condition set by URTHnEJL does not change the enhanced bit detection flag (URTHnSTR1.URTHnEBD), and an enhanced bit detection interrupt will not be generated. 1: Enables enhanced bit detection interrupts Detection of an enhanced bit under the condition set by URTHnEJL leads to setting of the enhanced bit detection flag (URTHnSTR1.URTHnEBD) and generation of an enhanced bit detection interrupt. • This setting becomes valid when the setting of the bit length for transmission and reception is 8 bits (URTHnCTL1.URTHnCLG = 1) and operation with the enhanced bit is enabled (URTHnEBE = 1). • Changing this bit is only allowed while operation of UARTHn is disabled (URTHnCTL0.URTHnPW = 0). • For details on enhanced bit detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.
13	URTHnEJL	Selects the condition for detecting the enhanced bit. 0: The value zero is the condition for detecting the enhanced bit. 1: The value one is the condition for detecting the enhanced bit. • As condition for enhanced bit detection, this setting selects a level of the enhanced bit (0 or 1) after control of the reception data level (URTHnCTL1.URTHnRDL). • While ID comparison is enabled (URTHnOPT1.URTHnIDCN = 1), a match in ID comparison after detection of the enhanced bit produces an enhanced bit detection interrupt. • Changing this bit is only allowed while the UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). • For details on enhanced bit detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.
12	URTHnSCFR	Enables and disables the noise filter for the synchronizing clock input. 0: Enables noise filter 1: Disables noise filter • If slave mode and data consistency checking are enabled (i.e. if URTHnCTL0.URTHnSLDC = 1), set this register to the same value as URTHnRXFR. • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
11	URTHnRSFR	Enables and disables the noise filter for the handshake signal input. 0: Enables noise filter 1: Disables noise filter • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
10	URTHnRXFR	Enables and disables the noise filter for received data. 0: Enables noise filter 1: Disables noise filter • When operation is as a slave in synchronous mode and data consistency checking is enabled (URTHnCTL0.URTHnSLDC = 1), set this bit to the same value as URTHnSCFR. • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
9	URTHnRTSL	Selects the active level of the output signal for handshaking. 0: Low is the active level. 1: High is the active level. • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).

Table 23-15 URTHnOPT0 Register Contents (3/3)

Bit Position	Bit Name	Function
8	URTHnCTSL	Selects the active level of the input signal for handshaking. 0: Low is the active level. 1: High is the active level. • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
6	URTHnHS	Enables and disables handshaking. 0: Disables handshaking 1: Enables handshaking • Disable loop-back control (URTHnCTL1.URTHnLPM = 0) if the setting of the URTHnHS bit is 1. • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
5	URTHnDM	Selects the timing of sampling for data reception in clock synchronous mode. 0: Received data are sampled on rising edges of the synchronizing clock (normal sampling). 1: Received data are sampled on falling edges of the synchronizing clock (delayed sampling). • Changes to this bit only become effective when clock synchronous mode is selected (URTHnSAS = 1). • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
4	URTHnMSS	Selects master or slave operation in clock synchronous mode. 0: Master The master outputs the synchronizing clock for use in transmission and reception. 1: Slave The synchronizing clock supplied by the master is used in transmission and reception. • Changes to this bit only become effective when clock synchronous mode is selected (URTHnSAS = 1). • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
3	URTHnSAS	Selects the operating mode for UARTHn. 0: Asynchronous mode 1: Synchronous mode • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). • Only use the interface in asynchronous mode after disabling the clock pin (URTHnSC).
2	URTHnF2R	Selects the length of data for reception as 1 frame or 2 frames. 0: Selects reception of 1 frame 1: Selects reception of 2 frames • Set this bit (URTHnOPT0.URTHnF2R) to 0 when data consistency checking is enabled (URTHnCTL0.URTHnSLDC = 1) or ID value compare-match detection is enabled (URTHnOPT1.URTHnIDCN = 1). • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).
1	URTHnF2T	Selects the length of data for transmission as 1 frame or 2 frames. 0: Selects transmission of 1 frame 1: Selects transmission of 2 frames • Changing this bit is only allowed while UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0).

(9) URTHnOPT1 – UARTHn Option Register 1

This register specifies the optional features for use in serial transfer by the UARTHn interface.

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

Reading this register always returns 00_H.

Address <URTHn_base1> $+04_H$

Initial value 00_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
URTHn IDCN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Table 23-16 URTHnOPT1 Register Contents

Bit Position	Bit Name	Function
7	URTHnIDCN	Enables and disables ID match detection. 0: ID match detection at the time of transfer of an enhanced bit (9th bit) is disabled. 1: ID match detection at the time of transfer of an enhanced bit (9th bit) is enabled. • The setting of this bit is not effective when the unit of transfer is 7 or 8 bits. • Clear this bit (to 0) if the current setting is for two-frame data reception (URTHnOPT0.URTHnF2R = 0) or handshaking (URTHnOPT0.URTHnHS = 1). • For details on ID match detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.

(10) URTHnOPT2 – UARTHn Option Register 2

This register specifies the optional features for use in serial transfer by the UARTHn interface.

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

Address <URTHn_base1> + 08_H

Initial value 0000_H

This register is initialized by a reset from any source.

15	14	13	12	11	10	9	8
URTHnIDCD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTHnMID[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-17 URTHnOPT2 Register Contents

Bit Position	Bit Name	Function
15 to 8	URTHnIDCD [7:0]	Sets the ID value for comparison at the time the enhanced bit (9th bit) is transferred. • For details on enhanced bit detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.
7 to 0	URTHnMID [7:0]	Specifies which bits of the ID values are compared. 0: The corresponding bit is compared at the time the enhanced bit (9th bit) is transferred. 1: The corresponding bit is not compared at the time the enhanced bit (9th bit) is transferred. • For details on ID match detection, refer to section 23.6.5, Enhanced Bit Detection/ID Compare-Match Detection.

(11) URTHnRX – UARTHn Receive Data Register

This register stores received data.

The data stored in the reception shift register are transferred to the URTHnRX register upon completion of the reception of one frame containing 7 or 8 bits of data.

This register corresponds to the bits 7 to 0 in the URTHnERXW register.

7-bit transfers

If the data length has been specified as 7 bits (URTHnCTL1.URTHnCLG = 0), the received data stored in this register are transferred in accord with the specified direction for transfer (MSB-first/LSB-first) as described below.

- With LSB-first reception (URTHnCTL1.URTHnSLD = 1), the received data are transferred to URTHnRX[6:0] and the MSB (URTHnRX[7]) always becomes 0.
- With MSB-first reception (URTHnCTL1.URTHnSLD = 0), the received data are transferred to URTHnRX[7:1] and the LSB (URTHnRX[0]) bit always becomes 0.
- For details on data formats, refer to section 23.6.1, Data Formats.

Overrun error

When an overrun error (URTHnSTR1.URTHnOVE = 1) occurs, data received at the time are not transferred to URTHnRX but are discarded.

When processing for reception ends and data reception is confirmed without any overrun errors, the received data are stored in the URTHnRX register in the specified storage format.

This register is writable if UARTHn operation is disabled (URTHnCTL0.URTHnPW = 0). If this is enabled (URTHnCTL0.URTHnPW = 1), any written values are disregarded and the register takes on its initial value.

Access

This register can be read in 8-bit units.

Address

<URTHn_base1> + $1C_H$

Initial value

FF_H

A reset from any source or, when UARTHn operation is enabled, changing the value of URTHnCTL0.URTHnPW from 0 to 1, will initialize the bits.

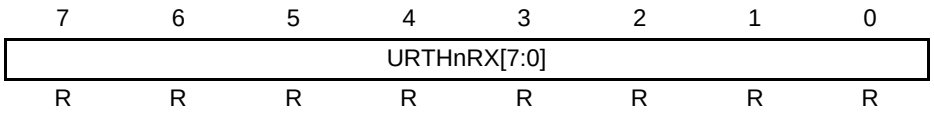


Table 23-18 URTHnRX Register Contents

Bit Position	Bit Name	Function
7 to 0	URTHnRX [7:0]	UARTHn receive data (see table 23-22, Access in the Various Modes of Reception.)

(12) URTHnRXHL – UARTHn Receive Data Register HL

This register stores received data.

The value stored in the reception shift register is transferred to the URTHnRXHL register upon completion of reception of data containing one frame with 9 bits of data.

This register corresponds to bits 15 to 0 in the URTHnERXW register.

Access This register can be read in 16-bit units.

Address <URTHn_base1> + 20_H

Initial value $FFFF_H$

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

15	14	13	12	11	10	9	8
URTHnRX[15:8]							
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
URTHnRX[7:0]							
R	R	R	R	R	R	R	R

Table 23-19 URTHnRXHL Register Contents

Bit Position	Bit Name	Function
15 to 0	URTHnRX [15:0]	UARTHn recieve data (see table 23-22, Access in the Various Modes of Reception.)

(13) URTHnERX – UARTHn Enhanced Bit Receive Data Register

This register stores received data.

The enhanced bit data stored in the reception shift register is transferred to the URTHnERX register upon completion of reception of data.

This register corresponds to the bits 23 to 16 in the URTHnERXW register.

Access This register can be read in 8-bit units.

Address <URTHn_base1> + 24_H

Initial value 11_H

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	URTHn ERXH8	0	0	0	URTHn ERXL8
R	R	R	R	R	R	R	R

Table 23-20 URTHnERX Register Contents

Bit Position	Bit Name	Function
4	URTHn ERXH8	Enhanced bit of UARTHn received data (see table 23-22, Access in the Various Modes of Reception.)
0	URTHn ERXL8	Enhanced bit of UARTHn received data (see table 23-22, Access in the Various Modes of Reception.)

(14) URTHnERXW – UARTHn Enhanced Bit Receive Word Data Register

This register stores received data.

The data stored in the reception shift register is transferred to the URTHnERXW register upon completion of reception of data.

Access This register can be read in 32-bit units.

Address <URTHn_base1> + 28_H

Initial value $0011FFFF_H$

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	URTHn ERXH8	0	0	0	URTHn ERXL8
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
URTHnRX[15:8]							
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
URTHnRX[7:0]							
R	R	R	R	R	R	R	R

Table 23-21 URTHnERXW Register Contents

Bit Position	Bit Name	Function
20	URTHn ERXH8	Enhanced bit of UARTHn received data
16	URTHn ERXL8	Enhanced bit of UARTHn received data
15 to 0	URTHn RX[15:0]	UARTHn received data

Table 23-22 Access in the Various Modes of Reception

Receive Mode	Data	Enhanced Bit
1-frame 8-bit data transfer	URTHnRX[7:0]	-
1-frame 7-bit data transfer (MSB first)	URTHnRX[7:1]	-
1-frame 7-bit data transfer (LSB first)	URTHnRX[6:0]	-
1-frame enhanced bit transfer	URTHnRX[7:0]	URTHnRX8 or URTHnERXL8
2-frame 8-bit data transfer	Frame 1 URTHnRX[15:8] Frame 2 URTHnRX[7:0]	-
2-frame 7-bit data transfer (MSB first)	Frame 1 URTHnRX[15:9] Frame 2 URTHnRX[7:1]	-
2-frame 7-bit data transfer (LSB first)	Frame 1 URTHnRX[14:8] Frame 2 URTHnRX[6:0]	-
2-frame enhanced bit transfer	Frame 1 URTHnRX[15:8] Frame 2 URTHnRX[7:0]	Frame 1 URTHnERXH8 Frame 2 URTHnERXL8

Caution When reading data and enhanced bits from separate registers, read the enhanced bit (from the URTHnERX register) first.

(15) URTHnTX – UARTHn Transmit Data Register

This register is used to store data to be transmitted.

Data for transmission in the URTHnTX register are stored in the transmission shift register in the data format specified for transmission.

This register corresponds to the bits 7 to 0 in the URTHnETXW register.

7-bit transfers If the data length has been specified as 7 bits (URTHnCTL1.URTHnCLG = 0), the transmitted data stored in this register are transferred in accord with the specified direction for transfer (MSB-first/LSB-first) as described below.

- If transmission is LSB-first (URTHnCTL1.URTHnSLD = 1), URTHnTX[6:0] is transferred to the shift register.
- If transmission is MSB-first (URTHnCTL1.URTHnSLD = 0), URTHnTX[7:1] is transferred to the shift register.

For details on data formats, refer to section 23.6.1, Data Formats.

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

Address <URTHn_base1> + $2C_H$

Initial value FF_H

This register is initialized by a reset from any source.

7	6	5	4	3	2	1	0
URTHnTX[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-23 URTHnTX Register Contents

Bit Position	Bit Name	Function
7 to 0	URTHnTX [7:0]	UARTHn transmit data (see table 23-27, Access in the Various Modes of Transmission.)

When transmission and reception are enabled (URTHnCTL0.URTHnPW = URTHnTXE = 1), writing to this register triggers the start of transmission.

Note If the next data are written to this register before ongoing transmission is completed, the interface will wait until the ongoing transmission is completed and then transmit the next data. Continuous transmission is thus possible.

(16) URTHnTXHL – UARTHn Transmit Data Register HL

This register is used to store data for transmission.

Data for transmission in the URTHnTXHL register are stored in the transmission shift register in accord with the data format specifications for transmission.

This register corresponds to bits 15 to 0 in the URTHnETXW register.

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

Address <URTHn_base1> + 30_H

Initial value $FFFF_H$

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

15	14	13	12	11	10	9	8
URTHnTX[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTHnTX[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-24 URTHnTXHL Register Contents

Bit Position	Bit Name	Function
15 to 0	URTHnTX [15:0]	UARTHn transmit data (see table 23-27, Access in the Various Modes of Transmission.)

(17) URTHnETX – UARTHn Enhanced Bit Transmit Data Register

This register is used to store data for transmission.

Data for transmission in the URTHnETX register are stored in the transmission shift register in accord with the data format specifications for transmission.

This register corresponds to the bits 23 to 16 in the URTHnETXW register.

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

Address <URTHn_base1> + 34_H

Initial value 11_H

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

7	6	5	4	3	2	1	0
0	0	0	URTHn ETXH8	0	0	0	URTHn ETXL8
R	R	R	R/W	R	R	R	R/W

Table 23-25 URTHnETX Register Contents

Bit Position	Bit Name	Function
4	URTHn ETXH8	Enhanced bit of UARTHn transmit data (see table 23-27, Access in the Various Modes of Transmission.)
0	URTHn ETXL8	Enhanced bit of UARTHn transmit data (see table 23-27, Access in the Various Modes of Transmission.)

(18) URTHnETXW – UARTHn Enhanced Bit Transmit Word Data Register

This register is used to store data for transmission.

Data for transmission in the URTHnETXW register are stored in the transmission shift register in accord with the data format specifications for transmission.

Access This register can be read/written in 32-bit units.

Address <URTHn_base1> + 38_H

Initial value $0011FFFF_H$

A reset from any source or, when UARTHn operation is enabled, setting URTHnCTL0.URTHnPW = 1, will initialize the bits.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	URTHn ETXH8	0	0	0	URTHn ETXL8
R	R	R	R/W	R	R	R	R/W

15	14	13	12	11	10	9	8
URTHnTX[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTHnTX[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-26 URTHnETXW Register Contents

Bit Position	Bit Name	Function
20	URTHn ETXH8	Enhanced bit of UARTHn transmit data
16	URTHn ETXL8	Enhanced bit of UARTHn transmit data
15 to 0	URTHnTX [15:0]	UARTHn transmit data

Table 23-27 Access in the Various Modes of Transmission

Receive Mode	Data	Enhanced Bit
1-frame 8-bit data transfer	URTHnTX[7:0]	-
1-frame 7-bit data transfer (MSB first)	URTHnTX[7:1]	-
1-frame 7-bit data transfer (LSB first)	URTHnTX[6:0]	-
1-frame enhanced bit transfer	URTHnTX[7:0]	URTHnTX8 or URTHnETXL8
2-frame 8-bit data transfer	Frame 1 URTHnTX[15:8] Frame 2 URTHnTX[7:0]	-
2-frame 7-bit data transfer (MSB first)	Frame 1 URTHnTX[15:9] Frame 2 URTHnTX[7:1]	-
2-frame 7-bit data transfer (LSB first)	Frame 1 URTHnTX[14:8] Frame 2 URTHnTX[6:0]	-
2-frame enhanced bit transfer	Frame 1 URTHnTX[15:8] Frame 2 URTHnTX[7:0]	Frame 1 URTHnETXH8 Frame 2 URTHnETXL8

Caution When reading data and enhanced bits from separate registers, read the enhanced bit (from the URTHnETX register) first.

(19) IC0REG0 – LIN Communications Slave Operation Baud-Rate Detection Register 0

LIN communications slave operation baud-rate detection register 0 is described below.

Access This register can be read/written in 32-bit units.

Address $FFFFFE00_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
IC0REG00151	IC0REG00150	IC0REG00141	IC0REG00140	IC0REG00131	IC0REG00130	IC0REG00121	IC0REG00120
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
IC0REG00111	IC0REG00110	IC0REG00101	IC0REG00100	IC0REG00091	IC0REG00090	IC0REG00081	IC0REG00080
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
IC0REG00071	IC0REG00070	IC0REG00061	IC0REG00060	IC0REG00051	IC0REG00050	IC0REG00041	IC0REG00040
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
IC0REG00031	IC0REG00030	IC0REG00021	IC0REG00020	IC0REG00011	IC0REG00010	IC0REG00001	IC0REG00000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-28 IC0REG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	IC0REG00m0, IC0REG00m1	Selects the signal to be input to TAUA0 ch (m). (m = 00 to 15)

IC0REG00m1	IC0REG00m0	Description
0	0	Input TAUA0Im to timer input of TAUA0 ch (m).
0	1	Input RXDH0 to timer input of TAUA0 ch (m).
1	0	Input RXDH1 to timer input of TAUA0 ch (m).
1	1	Input RXDH2 to timer input of TAUA0 ch (m).

Caution 1. Do not change the setting of this register while TAUA0 is operating.
Caution 2. Do not connect a serial data input pin to TAUA0 unless it is for use in baud-rate detection.

(20) IC0REG1 – LIN Communications Slave Operation Baud-Rate Detection Register 1

LIN communications slave operation baud-rate detection register 1 is described below.

Access This register can be read/written in 32-bit units.

Address $FFFFFE04_H$

Initial value 00000000_H

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24
IC0REG11151	IC0REG11150	IC0REG11141	IC0REG11140	IC0REG11131	IC0REG11130	IC0REG11121	IC0REG11120
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
IC0REG11111	IC0REG11110	IC0REG11101	IC0REG11100	IC0REG11091	IC0REG11090	IC0REG11081	IC0REG11080
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
IC0REG11071	IC0REG11070	IC0REG11061	IC0REG11060	IC0REG11051	IC0REG11050	IC0REG11041	IC0REG11040
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
IC0REG11031	IC0REG11030	IC0REG11021	IC0REG11020	IC0REG11011	IC0REG11010	IC0REG11001	IC0REG11000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-29 IC0REG1 Register Contents

Bit Position	Bit Name	Function
31 to 0	IC0REG11m0, IC0REG11m1	Selects the signal to be input to TAUA1 ch (m). (m = 00 to 15)

IC0REG11m1	IC0REG11m0	Description
0	0	Input TAUA1Im to timer input of TAUA1 ch (m).
0	1	Input RXDH0 to timer input of TAUA1 ch (m).
1	0	Input RXDH1 to timer input of TAUA1 ch (m).
1	1	Input RXDH2 to timer input of TAUA1 ch (m).

Caution 1. Do not change the setting of this register while TAUA1 is operating.
Caution 2. Do not connect a serial data input pin to TAUA1 unless it is for use in baud-rate detection.

23.5 Interrupt Request Signals

The following 3 interrupt request signals are generated by UARTHn.

- Transmission interrupt request URTHnTIT
- Reception interrupt request URTHnTIR
- Status interrupt request URTHnTIS

23.5.1 Transmission Interrupt Request URTHnTIT

When data for transmission are transferred from the URTHnTX register to the transmission shift register while transmission is enabled, a transmission interrupt request (URTHnTIT) will be generated.

The condition for generation of a transmission interrupt request depends on the setting of the URTHnCTL1.URTHnSLIT.

- At the start of transmission processing: URTHnCTL1.URTHnSLIT = 0.

  A transmission interrupt request is issued when transmission of the first bit starts (the start bit in data transmission or the first bit of the BF in BF transmission). During data transmission, a transmission interrupt request is issued when data for transmission in the URTHnTX register are transferred to the transmission shift register.

- At the end of transmission processing: URTHnCTL1.URTHnSLIT = 1.

  A transmission interrupt request is issued on completion of transmission of the last bit (the stop bit when the stop bit length is 1, or the 2nd of the stop bits when the stop bit length is 2).

Note Detection of a data consistency error leads to the generation of a URTHnTIS interrupt, so this interrupt will not be generated.

The following diagrams show when transmission interrupt requests are generated at the start and the end of transmission processing, respectively.

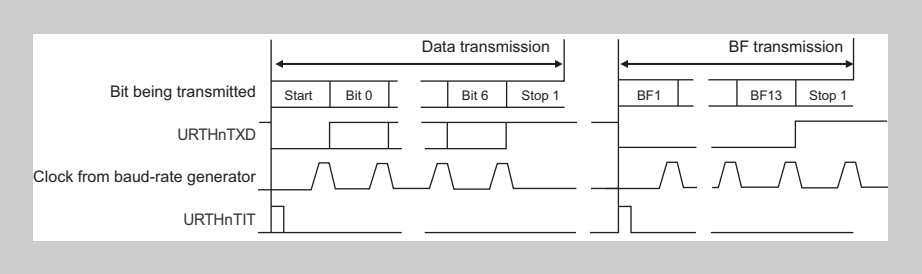


Figure 23-2 Timing of a Transmission Interrupt Request for URTHnCTL1.URTHnSLIT = 0

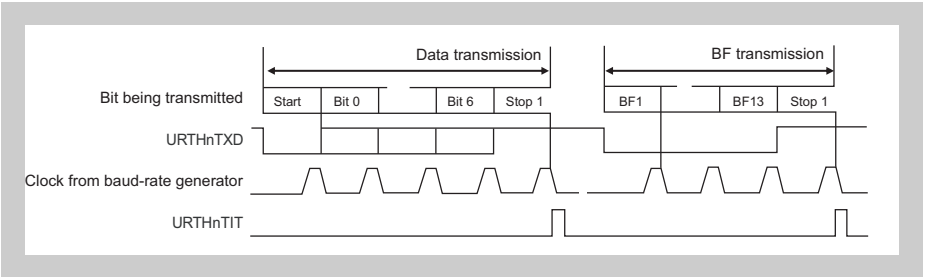


Figure 23-3 Timing of a Transmission Interrupt Request for URTHnCTL1.URTHnSLG = 1 and URTHnSLIT = 1.

23.5.2 Reception Interrupt Request URTHnTIR

A reception interrupt request is issued when the first bit of the stop bit is sampled.

In case of erroneous reception, the status interrupt (URTHnTIS) is generated instead of the URTHnTIR interrupt.

The reception interrupt request URTHnTIR is not generated while reception is disabled.

The following diagram shows the timing of the reception interrupt request during data/BF reception.

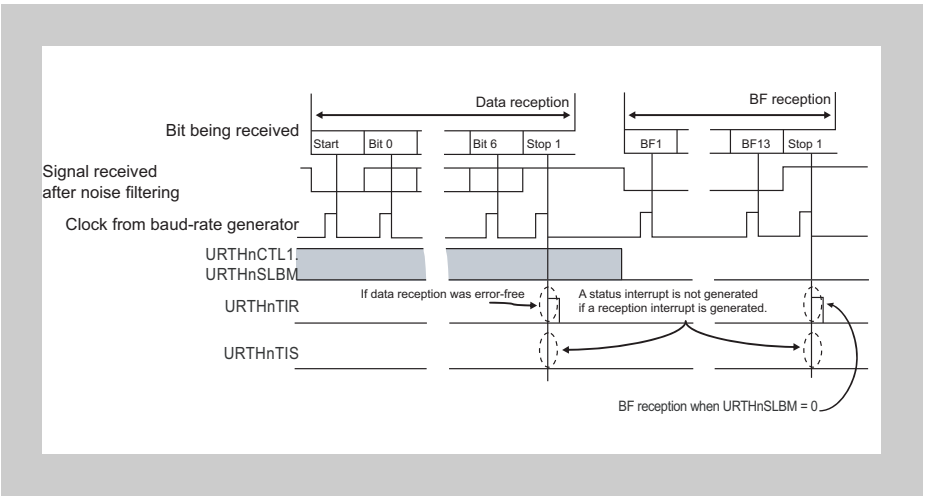


Figure 23-4 Timing of Reception Interrupt Request

23.5.3 Status Interrupt Request URTHnTIS

A status interrupt request (URTHnTIS) is generated in accord with the settings of status register 1 (URTHnSTR1) if an error condition is satisfied during reception or transmission.

When the BF reception mode selection bit is set in LIN communications mode (URTHnCTL1.URTHnSLBM = 1), the status interrupt request signal is generated when 11 or more consecutive bits at the low level (BF) are received.

23.6 Operation

23.6.1 Data Formats

The interface handles full-duplex serial data reception and transmission.

As shown in the figures below, one frame of data for transmission or received data consists of a start bit, the character bits, parity bit or bits, and stop bit or bits.

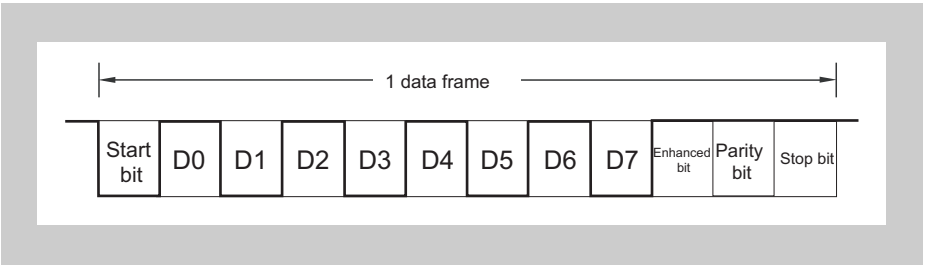
Several properties of transmitted and received data frames can be specified by control bits in the URTHnCTL1 register:

Table 23-30 Data Format Specification

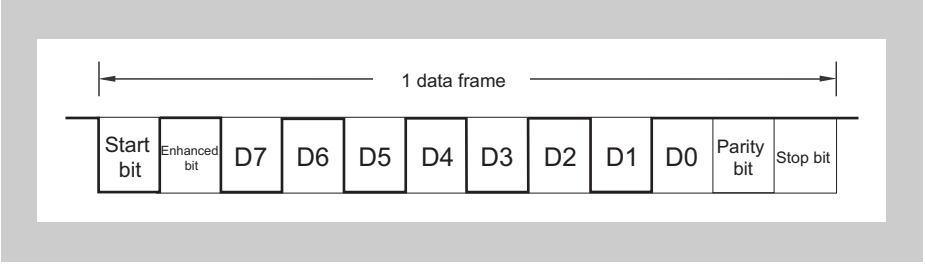
Item	Options	Control Bits
Start bit	1 bit	Fixed
Character bits	7 bits/8 bits/9 bits	URTHnCTL1.URTHnCLG
Parity	Even parity/odd parity/0 parity/ no parity	URTHnCTL1.URTHnSLP[1:0]
Number of stop bits	1 bit/2 bits	URTHnCTL1.URTHnSLG
Data order	MSB first/LSB first	URTHnCTL1.URTHnSLD
Levels of transmitted data	Inverted/not inverted	URTHnCTL1.URTHnTDL
Levels of received data	Inverted/not inverted	URTHnCTL1.URTHnRDL

(1) UARTHn Transmit/Receive Data Format

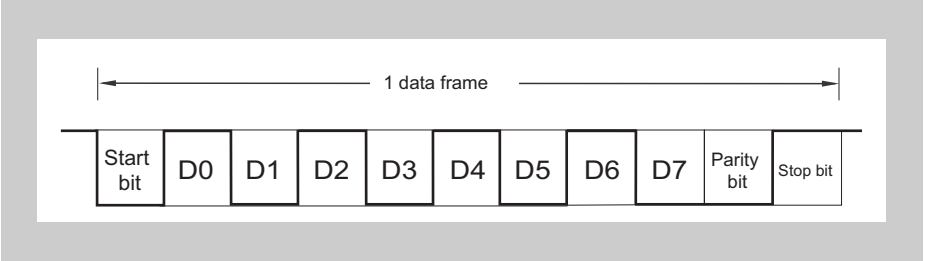
(a) 9-bit data length (enhanced bit), LSB first, even parity, 1 stop bit, data for transfer: 155_H



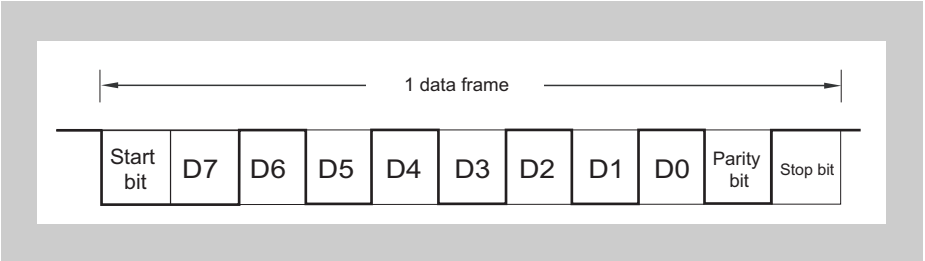
(b) 9-bit data length (enhanced bit), MSB first, even parity, 1 stop bit, data for transfer: 155_H



(c) 8-bit data length, LSB first, even parity, 1 stop bit, data for transfer: 55_H



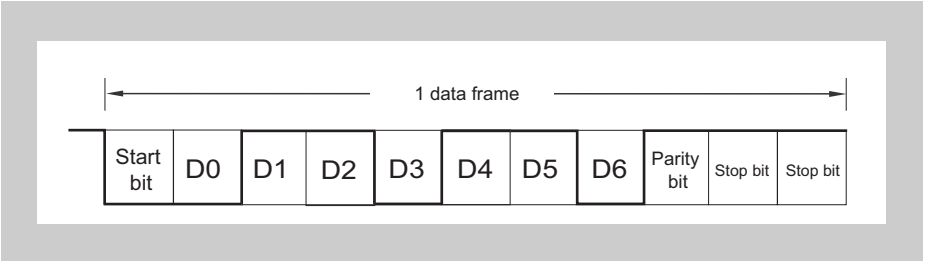
(d) 8-bit data length, MSB first, even parity, 1 stop bit, data for transfer: 55_H



(e) 8-bit data length, MSB first, even parity, 1 stop bit, data for transfer: 55_H, URTHnTTXD inverted



(f) 7-bit data length, LSB first, odd parity, 2 stop bits, data for transfer: 36_H



(g) 8-bit data length, LSB first, no parity, 1 stop bit, data for transfer: 87_H



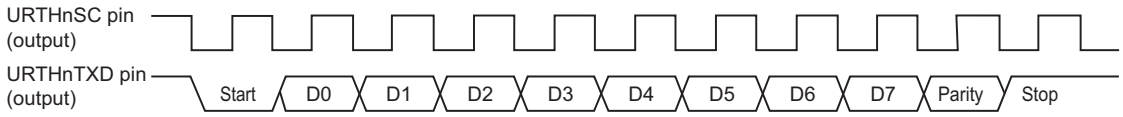
23.6.2 Clock Synchronous Mode

In clock synchronous mode, data are transmitted and received in synchronization with the synchronizing clock (provided by the master side).

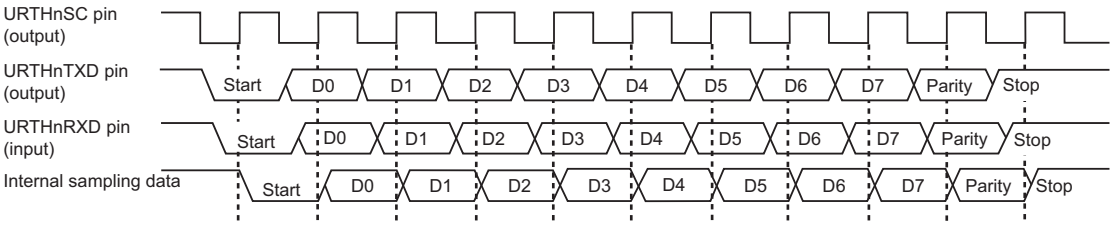
Normal sampling mode

Data are transmitted (output) on falling edges and received data are sampled on rising edges of the synchronizing clock.

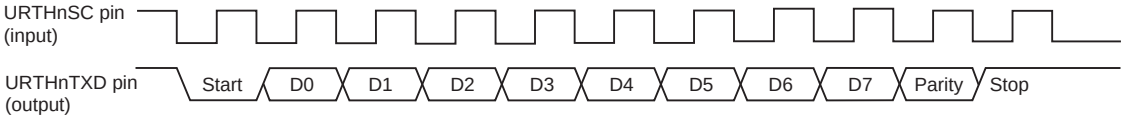
(a) Timing of transmission by the master



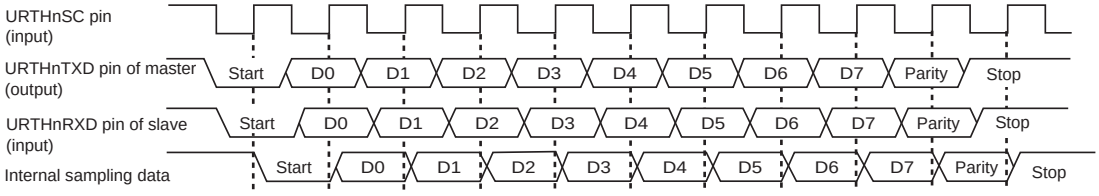
(b) Timing of reception by the master



(c) Timing of transmission by the slave



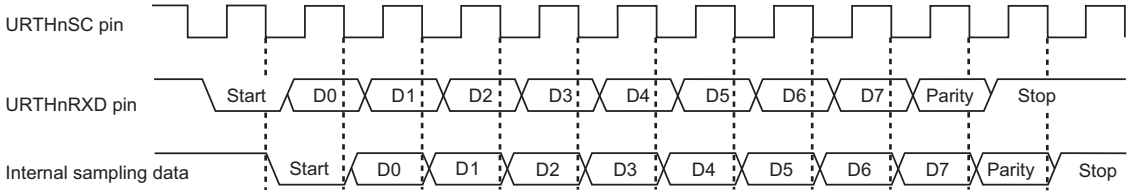
(d) Timing of reception by the slave



Delayed sampling mode

In clock synchronous mode, received data are sampled on falling edges of the synchronizing clock.

(a) Examples of operation in delayed sampling mode (master, reception)



23.6.3 Handshake Mode

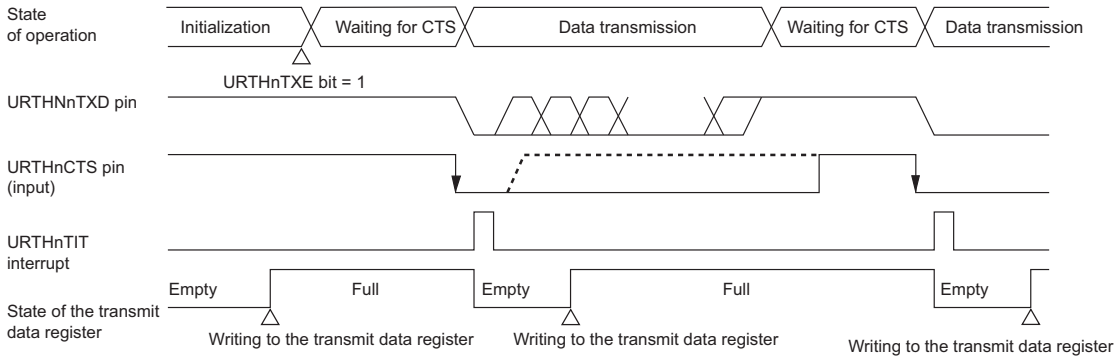
In this mode, the receive enable signal (URTHnRTS) and transmit enable signal (URTHnCTS) are used for handshaking in data transmission and reception.

Transmission

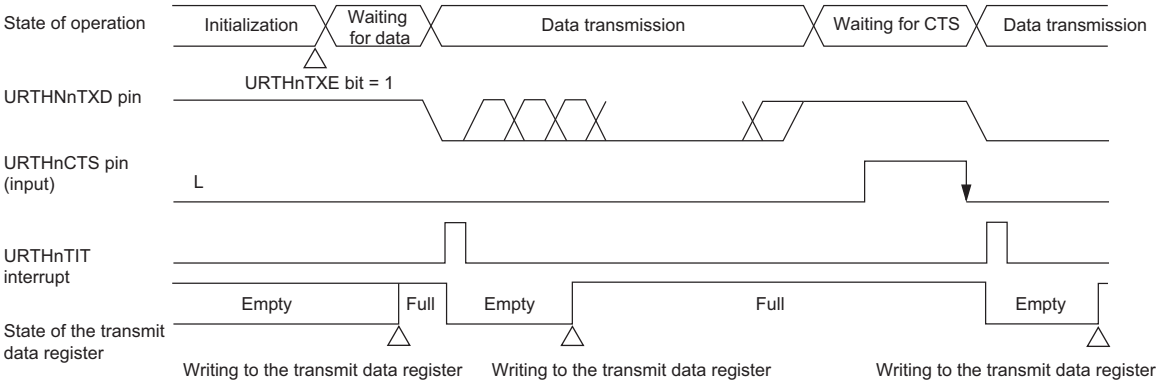
After data are written to the transmit data register, the data are transmitted if the URTHnCTS signal input is at the active level. If the URTHnCTS signal is at the inactive level, data transmission will be suspended.

The data are transmitted immediately when the URTHnCTS signal goes to the active level. If the URTHnCTS signal is then changed to the inactive level while transmission is in progress, 1 frame of data will still be transmitted.

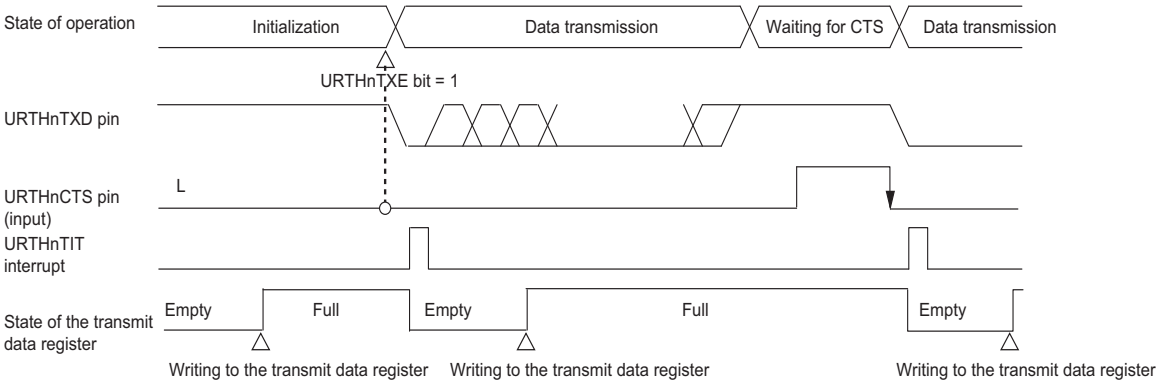
(a) Example of handshake signals (to wait for permission from URTHnCTS) in transmission



(b) Example of handshake signals (to wait for writing of data) in transmission

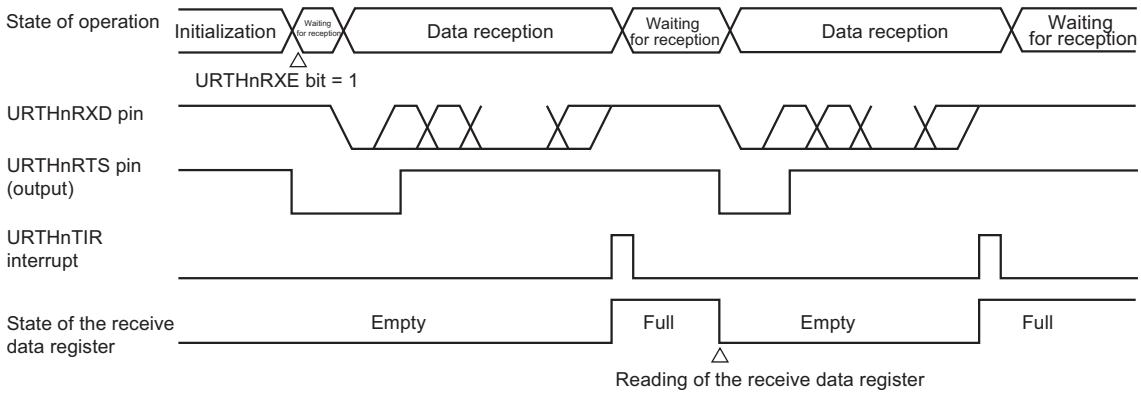


(c) Example of handshake signals (to wait for writing to URTHnTXE) in transmission



Reception When the receive data register is empty, the other party is notified of readiness to receive data by placing the reception enable signal (URTHnRTS) at the active level. When the receive data register is not empty (reading is incomplete), the URTHnRTS signal is placed at the inactive level because the register is not ready to receive data. The URTHnRTS signal must be placed at the inactive level at the time of sampling of the first bit of received data.

(a) Example of handshake signals in reception

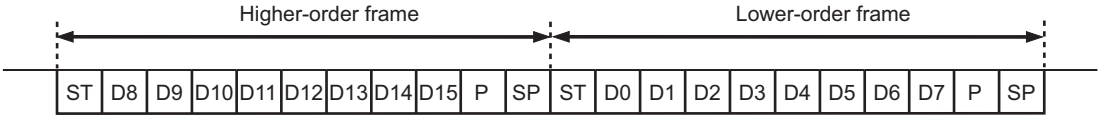


23.6.4 Consecutive Two-Frame Transfer Mode

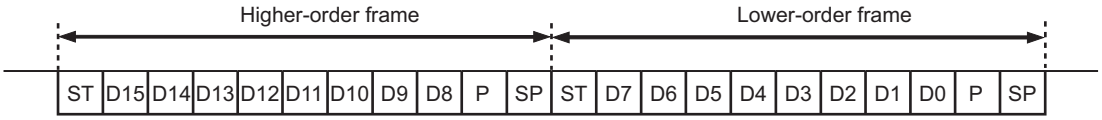
Consecutive two-frame transfer mode allows the reception or transmission of two consecutive frames. The format in this case is as follows.

(a) Format for consecutive two-frame transfer

Format of consecutive two-frame transfer, LSB first:



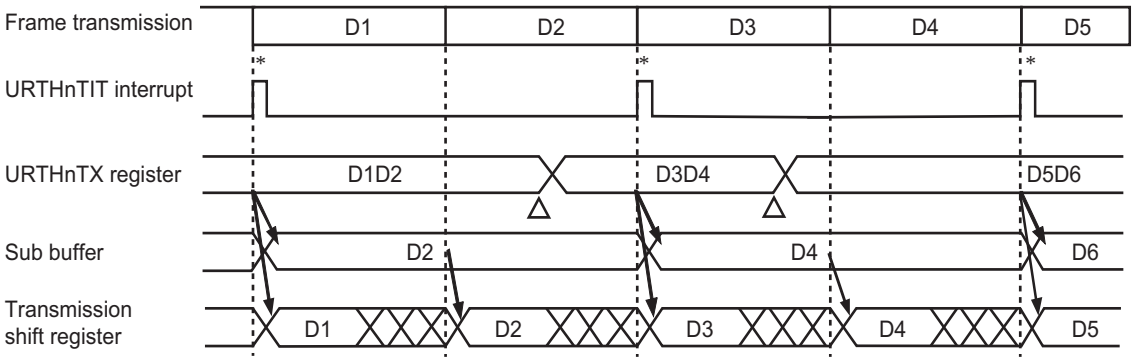
Format of consecutive two-frame transfer, MSB first:



ST:	Start bit
D0 to D15:	Character bits
P:	Parity bit
SP:	Stop bit

Caution In the consecutive two-frame transfer, the transmit data register is only accessible in 16-bit units. Do not write to this register in 8-bit units.

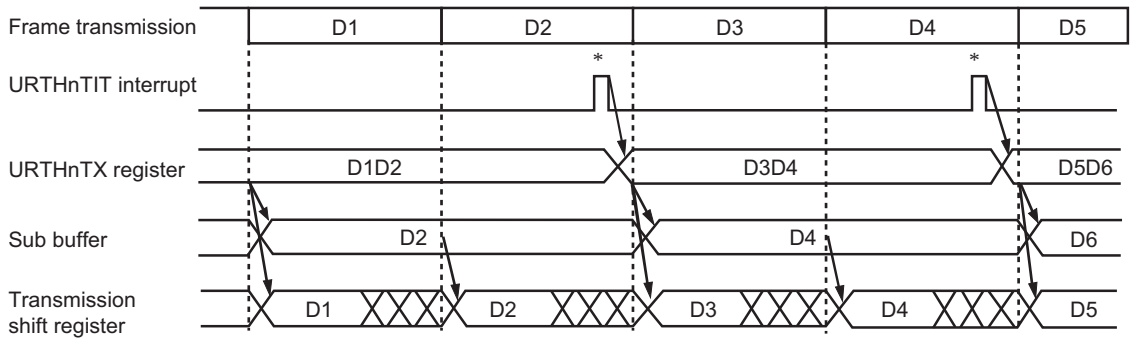
(b) Timing of consecutive two-frame transmission (when URTHnCTL1.URTHnSLIT = 0)



Note * A transmission enable interrupt is generated if the transmit data register is empty.

Caution In the consecutive two-frame transfer, the transmit data register is only accessible in 16-bit units. Do not write to this register in 8-bit units.

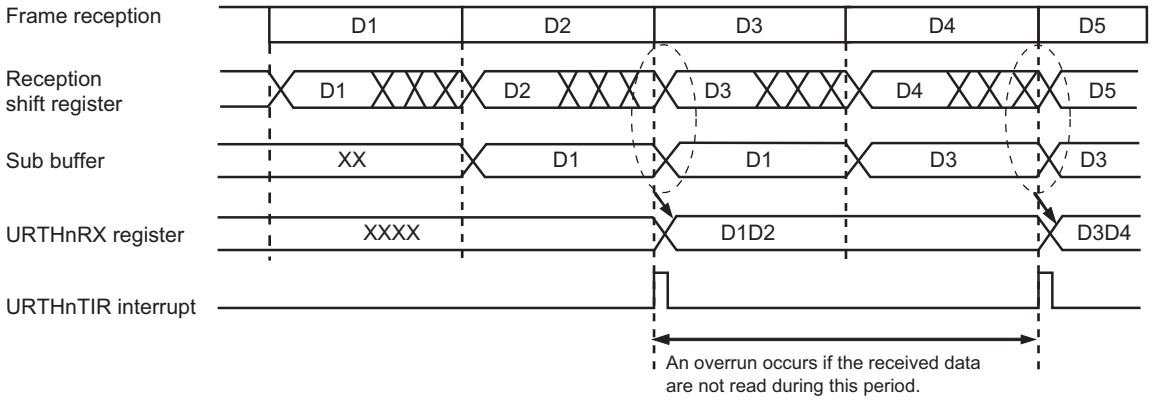
(c) Timing of consecutive two-frame transmission (when URTHnCTL1.URTHnSLIT = 1)



Note * A transmission enable interrupt is generated at the end of transmission.

Caution In the consecutive two-frame transfer, the transmit data register is only accessible in 16-bit units. Do not write to this register in 8-bit units.

(d) Timing of consecutive two-frame reception



Caution In the consecutive two-frame transfer, the transmit data register is only accessible in 16-bit units. Do not write to this register in 8-bit units.

23.6.5 Enhanced Bit Detection/ID Compare-Match Detection

For enhanced bit detection, the enhanced bit is enabled (URTHnOPT0.URTHnEBE = 1) and enhanced bit detection interrupts are enabled (URTHnOPT0.URTHnEGE = 1). On detection of an enhanced bit during reception, an enhanced bit detection interrupt is generated if the condition for enhanced bit detection set in URTHnOPT0.URTHnEJL is matched and the enhanced bit detection flag (URTHnSTR1.URTHnEBD) is set to 1.

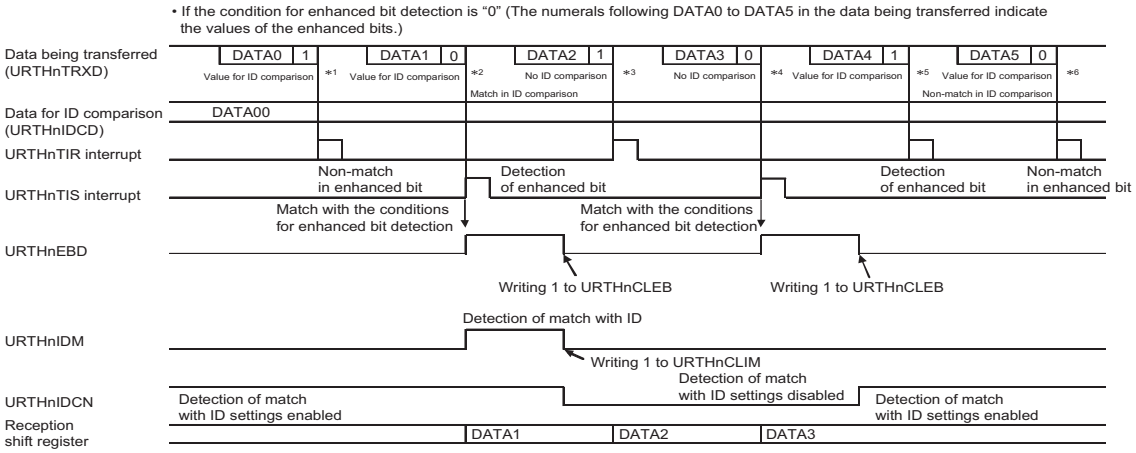
The enhanced bit detection flag (URTHnSTR1.URTHnEBD) is set to 1 when the enhanced bit is 0 if the setting of the condition for enhanced bit detection is URTHnOPT0.URTHnEJL=0 (URTHnOPT1.URTHnIDCN = 0) and when the enhanced bit is 1 if the setting is URTHnOPT0.URTHnEJL = 1 (URTHnOPT1.URTHnIDCN = 0) (however, if ID comparison is selected as a condition, i.e. if URTHnOPT1.URTHnIDCN = 1, the enhanced bit will only be detected if ID comparison also produces a match).

URTHnSTR1.URTHnEBD is cleared when 1 is written to URTHnSTC.URTHnCLEB.

When the ID and settings for ID comparison match while ID matching is selected as a condition (URTHnOPT1.URTHnIDCN = 1), URTHnSTR1.URTHnIDM is set to 1. When an enhanced bit is detected and the ID matches the settings for ID comparison in enhanced bit detection while URTHnOPT1.URTHnIDCN = 1, the ID comparison match flag is set to URTHnSTR1.URTHnIDM = 1, which enables data reception. Data are not received if the ID does not match the settings for ID comparison. Moreover, bits for which the corresponding URTHnOPT2.URTHnMID[7:0] bit is 1 are not included in ID comparison (i.e. these bits are masked).

URTHnSTR1.URTHnIDM is cleared when 1 is written to URTHnSTC.URTHnCLIM.

(a) Timing of enhanced bit detection and ID compare-match detection



Note 1. Enhanced bit detection and ID comparison are enabled (URTHnOPT1.URTHnIDCN = 1) during the transfer of DATA0. The level of the enhanced bit differs from the detection condition, so neither a reception interrupt (URTHnTIR) nor a status interrupt (URTHnTIS) is generated.

Note 2. Enhanced bit detection and ID comparison are enabled (URTHnOPT1.URTHnIDCN = 1) during the transfer of DATA1. The level of the enhanced bit and the ID match the respective detection conditions, so a status interrupt (URTHnTIS) is generated.

Note 3. In DATA2, data are being received because the ID was matched in DATA1. ID comparison is disabled (URTHnOPT1.URTHnIDCN = 0) and so will not proceed. A reception interrupt (URTHnTIR) is generated in response to the correct reception of DATA2.

Note 4. Enhanced bit detection and ID comparison are disabled (URTHnOPT1.URTHnIDCN = 0) during the transfer of DATA3, but the level of the enhanced bit matches the detection condition, so a status interrupt (URTHnTIS) is generated.

Note 5. Enhanced bit detection and ID comparison are enabled (URTHnOPT1.URTHnIDCN = 1) during the transfer of DATA4, and the level of the enhanced bit does not match the detection condition, so neither a reception interrupt (URTHnTIR) nor a status interrupt (URTHnTIS) is generated.

Note 6. Enhanced bit detection and ID comparison are enabled (URTHnOPT1.URTHnIDCN = 1) during the transfer of DATA5. The level of the enhanced bit matches the detection condition but the ID does not, so neither a reception interrupt (URTHnTIR) nor a status interrupt (URTHnTIS) is generated.

Caution 1. If a reception error (parity, framing, or overrun error) had occurred during the reception of DATA2, the error flag would have been set and a status interrupt request (URTHnTIS) generated instead of a reception completed interrupt request (URTHnTIR).

Caution 2. If a reception error (parity, framing or overrun error) had occurred during the reception of DATA1 and DATA3, the error flag would have been set (along with URTHnIDM or URTHnEBD). If a reception error occurred during DATA0, DATA4, or DATA5, the error flag would not have been set.

23.6.6 BF Transmission/Reception Format

The UARTHn has an BF (break field) transmission/reception control function to enable use of the LIN function.

About LIN

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud-rate information and the slave receives it and corrects the baud-rate error. Therefore, communications are possible when the baud-rate error of the slave is ± 14% or less.

For an outline of LIN transmission and reception, refer to figure 23-5, LIN Transmission in Outline, and figure 23-6, LIN Reception in Outline.

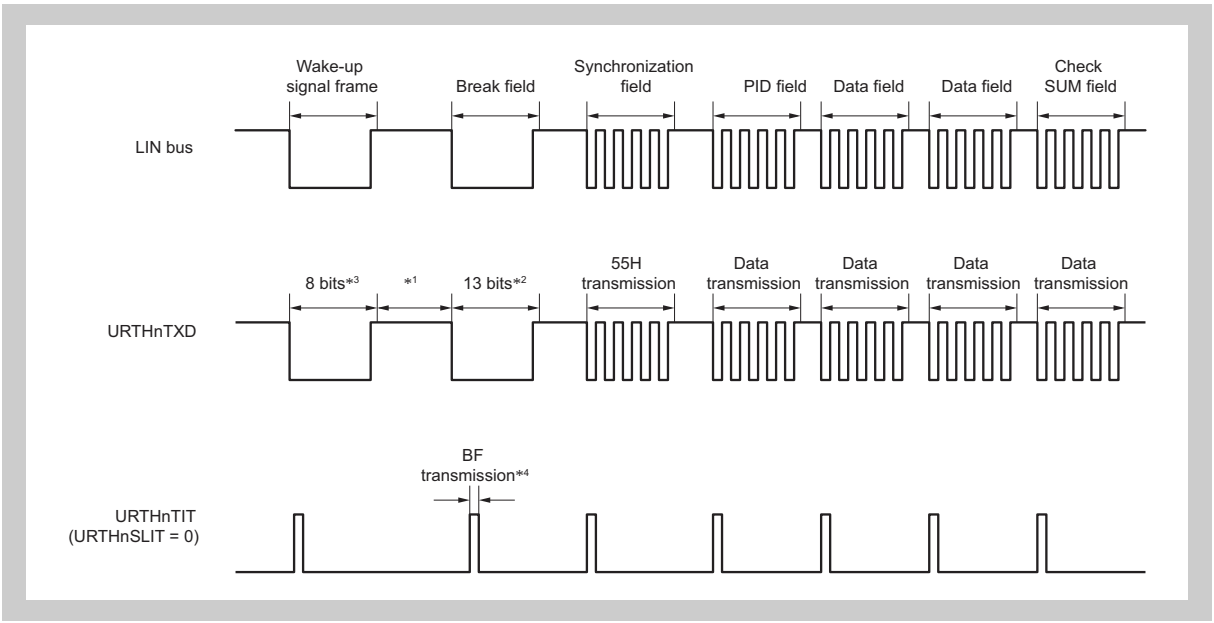


Figure 23-5 LIN Transmission in Outline

Note 1. The interval between each field is controlled by software.

Note 2. BF output is performed by hardware. The output width is the bit length set by URTHnCTL1.URTHnBLG[2:0]. If even finer output width adjustments are required, such adjustments can be performed using URTHnCTL2.URTHnBRS[11:0].

Note 3. 80_H transfer in the 8-bit mode is substituted for the wakeup signal frame.

Note 4. A transmission enable interrupt URTHnTIT is generated at the start of each transmission. URTHnTIT is also generated at the start of each BF transmission.

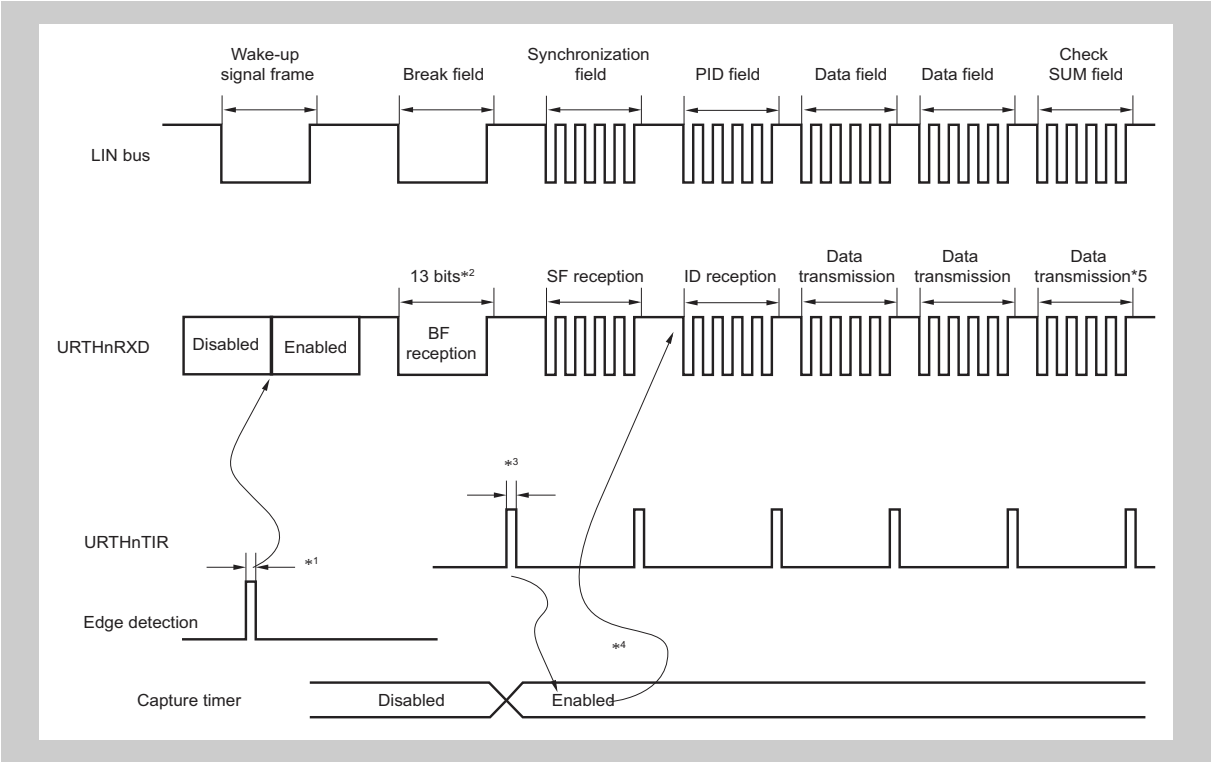


Figure 23-6 LIN Reception in Outline

Note 1. The wakeup signal is sent by the edge detector for the pin, the UARTHn is enabled, and the BF reception mode is set.

Note 2. BF reception is judged to have ended normally upon detection of 11 or more BF bits.
An interrupt is generated as follows in accord with the setting of the BF reception mode selection bit (URTHnCTL1.URTHnSLBM):

- a reception interrupt (URTHnTIR) is generated if URTHnCTL1.URTHnSLBM is set to "0", and
- a status interrupt (URTHnTIS) is generated if URTHnCTL1.URTHnSLBM is set to "1".

Upon detection of BF reception of less than 11 bits, an BF reception error is judged, no interrupt is generated, and the mode returns to the BF reception mode.

Note 3. An interrupt is generated as follows in accord with the setting of the BF reception mode selection bit (URTHnCTL1.URTHnSLBM) on the normal completion of BF reception:

- a reception interrupt (URTHnTIR) is generated if URTHnCTL1.URTHnSLBM is set to "0", and
- a status interrupt (URTHnTIS) is generated and the BF reception success flag (URTHnSTR1.URTHnBSF) is set if URTHnCTL1.URTHnSLBM is set to "1".

If the BF reception trigger bit URTHnTRG.URTHnBRT = 1, overrun, parity, and framing errors are not detecting during reception of the BF. Moreover, data are not transferred from the reception shift register to the receive data register URTHnRX. At that time, the URTHnRX retains its prior value.

Note 4. In order to adjust the baud-rate clock properly, the URTHnRXD signal needs to be connected to the capture input of a timer. (For details, refer to section 23.8, Detecting the Baud Rate in LIN Communications as a Slave.) The transfer rate and baud-rate error can be calculated from the measured time between two edges of the URTHnRXD signal, and the baud rate can be adjusted accordingly via the baud-rate setting bits (URTHnCTL2.URTHnBRS[11:0]).

Note 5. The check-sum field is distinguished by software. UARTHn is initialized following reception of the check-sum field, and processing to set the BF reception mode again is performed by software. When URTHnCTL1.URTHnSLBM = 1, BF can be received.

23.6.7 BF Transmission

Setting both the URTHnPW and URTHnTXE bits in URTHnCTL0 to 1 places the interface in the transmission enabled state, after which BF transmission is started by setting the BF transmission trigger (URTHnTRG.URTHnBTT) to 1.

Thereafter, URTHnSTR0.URTHnSSBT is set to "1" and the low level is output over a width of 13 to 20 bits, as specified by URTHnCTL1.URTHnBLG[2:0].

A transmission interrupt URTHnTIT is generated

- at the start of BF transmission if URTHnCTL1.URTHnSLIT = 0 and
- at the end of BF transmission if URTHnCTL1.URTHnSLIT = 1.

Following the end of BF transmission, URTHnSTR0.URTHnSSBT is automatically cleared. Thereafter, the UARTHn transmission mode is restored.

Transmission is suspended until the next data for transmission are written to the URTHnTX register, or until the BF transmission trigger (URTHnTRG.URTHnBTT) is set to 1 and URTHnSTR0.URTHnSSBT changes to 1.

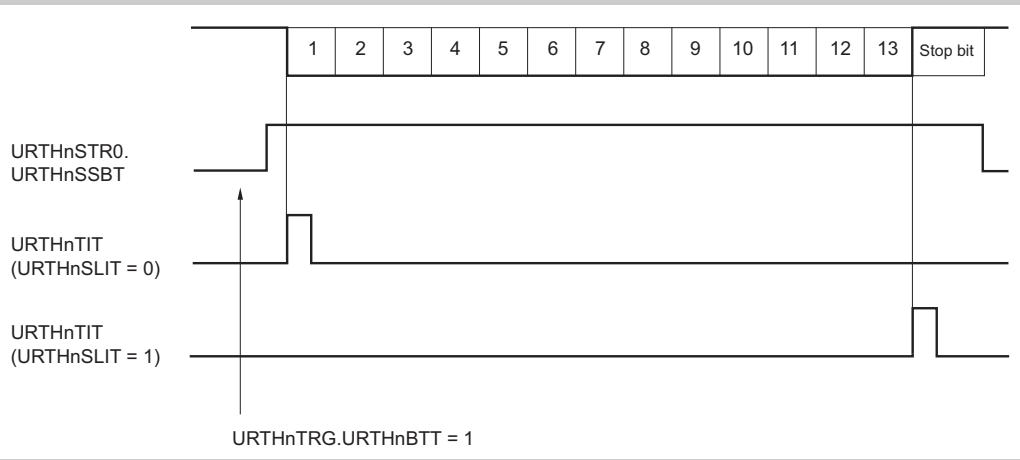


Figure 23-7 BF Transmission

23.6.8 BF Reception

The reception enabled status is achieved by setting the URTHnCTL0.URTHnPW bit to 1 and then setting the URTHnCTL0.URTHnRXE bit to 1.

The state of waiting for BF reception is set by setting the BF reception trigger (URTHnTRG.URTHnBRT) to 1.

In the BF reception wait status, similarly to the UARTHn reception wait status, the URTHnRXD pin is monitored and start bit detection is performed.

Following detection of the low level, reception is started and the internal counter counts up according to the set baud rate.

When a high level is received over at least 11 bit intervals (i.e. valid BF reception), the response depends on the value of the BF reception mode selection bit.

- A reception interrupt (URTHnTIR) is generated if URTHnCTL1.URTHnSLBM = 0.
- A status interrupt (URTHnTIS) is generated and the BF reception success flag (URTHnSTR1.URTHnBSF) is set if URTHnCTL1.URTHnSLBM = 1.

The URTHnSTR0.URTHnSSBR bit is automatically cleared and BF reception ends.

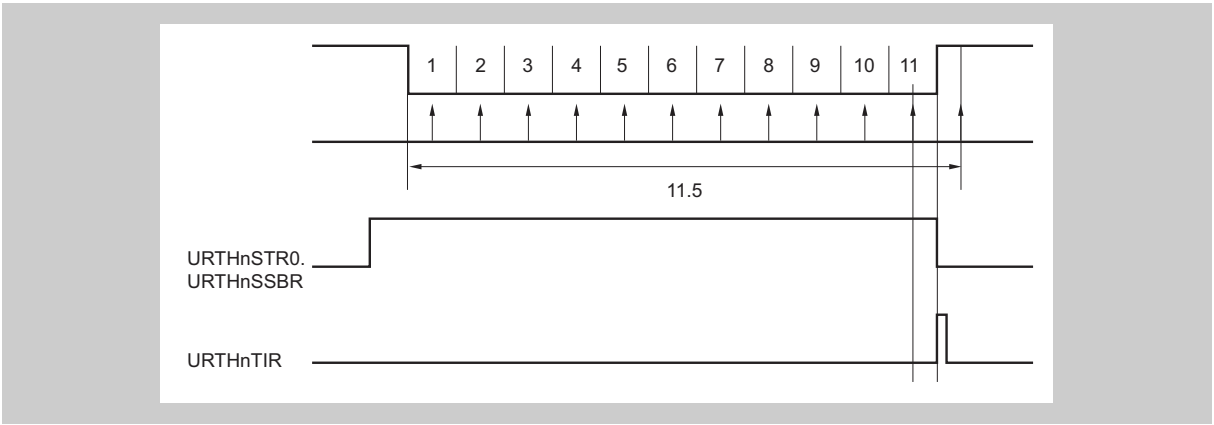


Figure 23-8 Normal BF Reception (Detection as Stop Bits After 10.5 Bits at the Low Level)

Error detection for the URTHnSTR1 error flags URTHnOVE, URTHnPE, and URTHnFE is suppressed and UARTHn communication error detection processing is not performed.

Moreover, the received data are not stored in URTHnRX.

If the BF width is 10 or fewer bits, reception is terminated as error processing without generating an interrupt, and the mode returns to the BF reception mode. URTHnSTR0.URTHnSSBR is not cleared at this time.

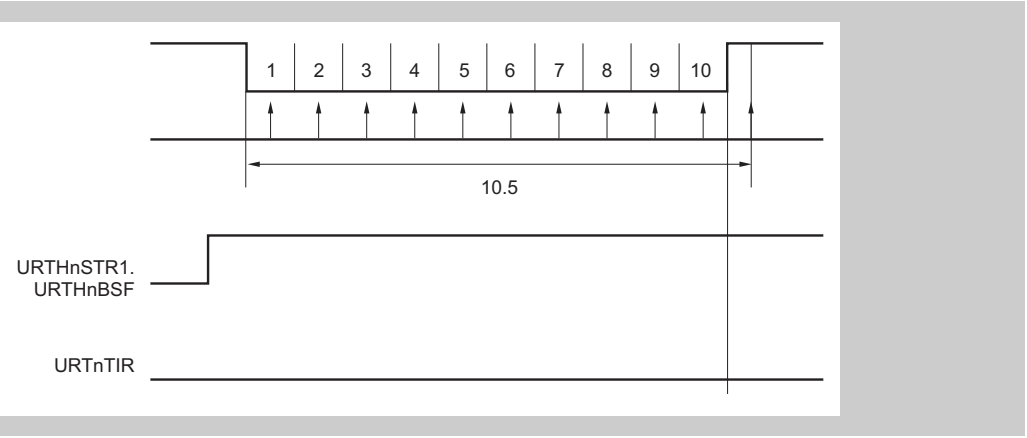


Figure 23-9 Error in BF Reception (Detection as Stop Bits within 10.5 Bits at the Low Level)

The URTHnSTR1.URTHnBSF bit indicates the state of successful reception of the BF.

Note URTHnSTR0.URTHnSSBR is set to "1"

- by setting URTHnTRG.URTHnBRT to "1" and
- is cleared by normal BF reception.

23.6.9 Transmission Data Consistency Check

The UARTHn incorporates a data consistency checking function to detect mismatches between data transmitted as the URTHnTXD signal and data received as the URTHnRXD signal while the device is operating with transmission enabled.

Data consistency check is enabled by URTHnCTL0.URTHnSLDC = 1.

The data consistency is checked by comparing the transmit data in the transmit register URTHnTX and the received data in the receive register URTHnRX. In case of a mismatch the data consistency error flag URTHnSTR1.URTHnDCE is set and a status interrupt request URTHnTIS is generated.

Data are not checked for consistency while transmission is suspended.

Consistency between the levels of the transmitted data and of the input data is checked even if reception is disabled during transmission. In this case, the receive completion interrupt request signal URTHnTIR does not occur, and status bits of URTHnSTR1 (URTHnBSF, URTHnFE and URTHnOVE) are not set either, but when a consistency error occurs, the status interrupt request signal URTHnTIS is generated and the status bit (URTHnDCE) of URTHnSTR1 is set. Received data does not need to be read.

For details, refer to section 23.4, (6) URTHnSTR1 – UARTHn Status Register 1.

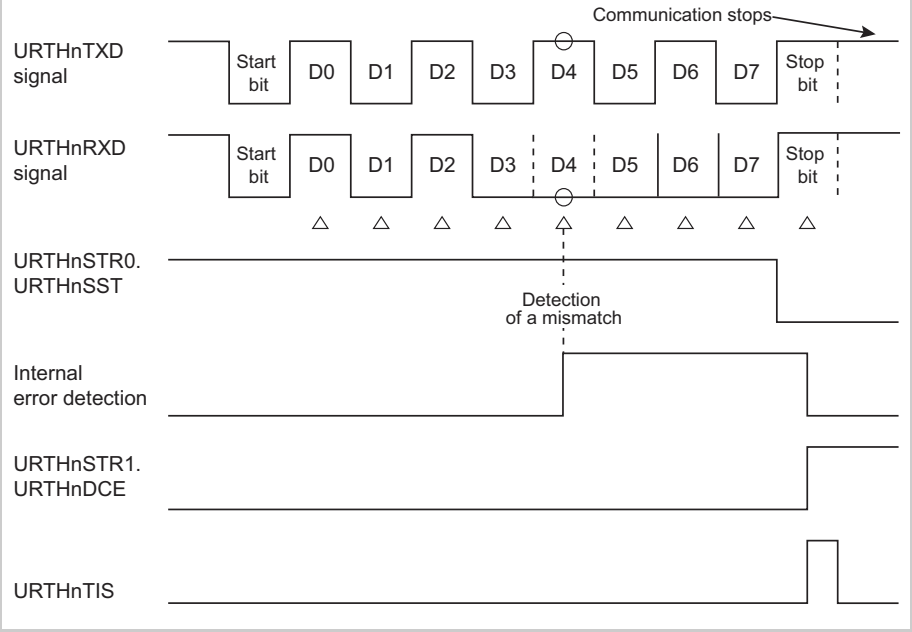


Figure 23-10 Timing Example of Data Consistency Error (No BF Reception Active, i.e. URTHnSTR0.URTHnSSBR = 0)

23.6.10 UARTHn Transmission

Transmission start

Set the transmission enabled status by performing the following procedures.

- Specify the baud rate by the UARTHn control register 2 (URTHnCTL2).
- Specify the transmit parity, data character length, stop bit length, transmit data order, transmission interrupt request timing and output logic level by the UARTHn control register 1 (URTHnCTL1).
- Enable UARTHn operation and transmission by setting URTHnCTL0.URTHnPW = URTHnCTL0.URTHnTXE = 1.

Writing of the data for transmission to the transmission buffer register URTHnTX causes transmission to start. Once saved in the URTHnTX register, the data are transferred to the transmission shift register. After that, the start, parity and stop bits are included in the data frame, which is then output serially via URTHnTXD.

Transmission stop

When URTHnCTL0.URTHnPW or URTHnCTL0.URTHnTXE is set to 0, transmission operations are stopped immediately, even during transmission processing.

Concurrent BF and data transmission

When a BF transmit request and a data transmit request have both been set, BF transmission takes priority.

Data consistency check

On detection of a data consistency error, further data are not transmitted until URTHnSTC.URTHnCLDC = 1, URTHnCTL0.URTHnPW = 0, or URTHnCTL0.URTHnTXE = 0 is written. (After an error is detected, follow the procedure described in section 23.4, (1) URTHnCTL0 – UARTHn Control Register 0, under Caution 1 and Caution 2.)

URTHnTIT timing

The time to generate the transmission interrupt URTHnTIT depends on the setting of the URTHnCTL1.URTHnSLIT bit:

- If URTHnCTL1.URTHnSLIT = 0, URTHnTIT is generated at the start of transmission, i.e. when the data from the data register (URTHnTX) are transferred to the transmission shift register.
- If URTHnCTL1.URTHnSLIT = 1, URTHnTIT is generated on completion of the entire data transmission process, i.e. when the last bit of the data frame has been transmitted.

Once URTHnTIT is generated, the next data can be written to URTHnTX.

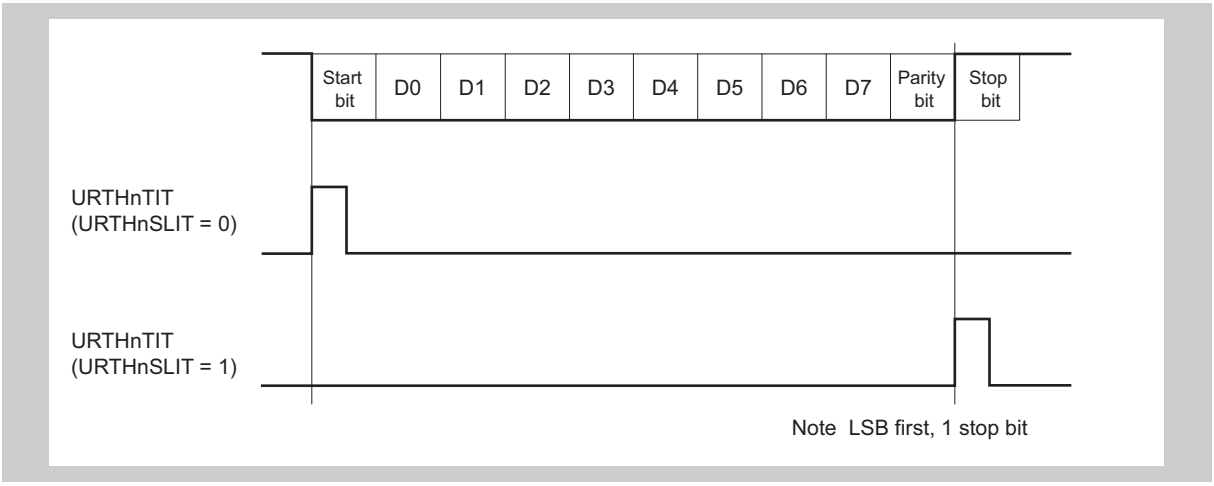


Figure 23-11 Transmission Interrupt Timing

23.6.11 Continuous Transmission Procedure

Continuous transmission is achieved by setting URTHnCTL1.URTHnSLIT = 0, using the URTHnTIT interrupt to confirm the timing with which data for transmission that have been written to the URTHnTX register are transferred to the transmission shift register, and then writing the next data for transmission to the data register while data transmission is in progress.

Caution If a value is written to the URTHnTX register before the URTHnTIT is generated, previously set data for transmission are overwritten by the new data.
To initialize the transmission unit (by setting URTHnTXE = 0 or URTHnPWR = 0), confirm that no transmission is ongoing (i.e. that URTHnSTR0.URTHnSSBT = URTHnSST = 0). Initialization while transmission is ongoing leads to transmission being aborted.

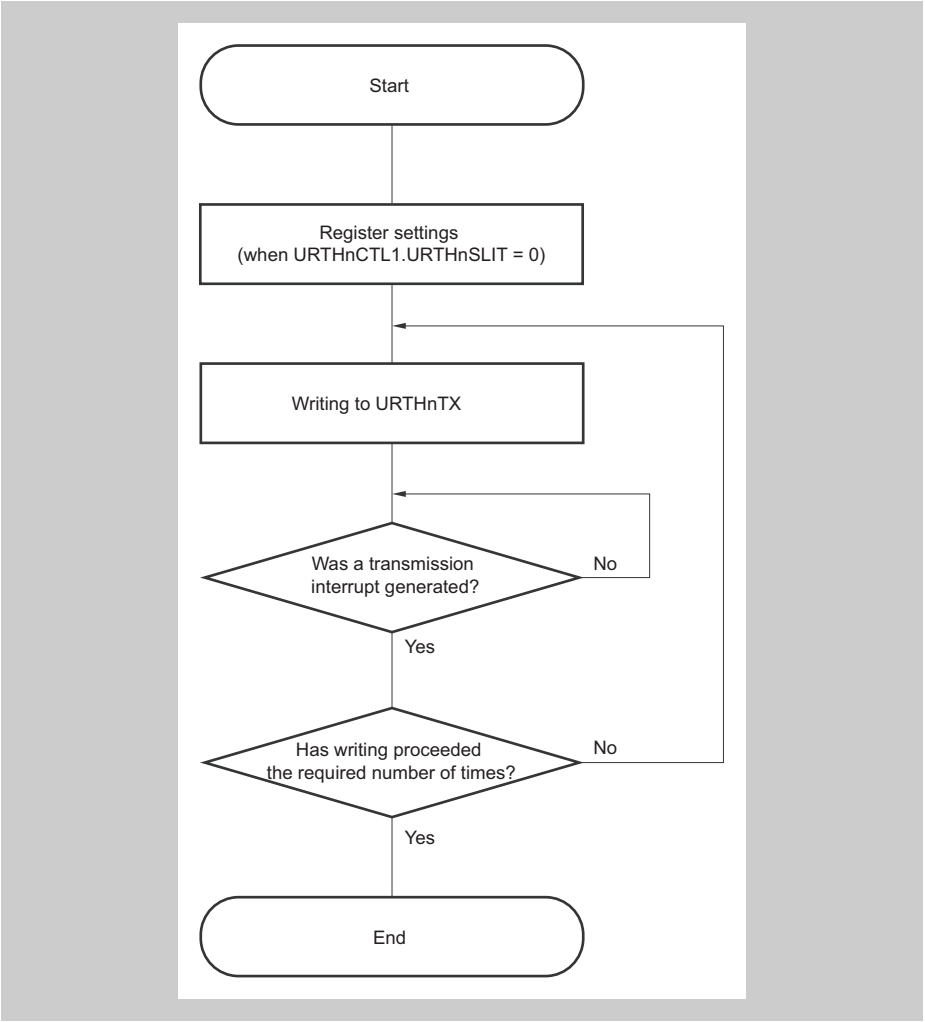


Figure 23-12 Continuous Transmission Processing Flow

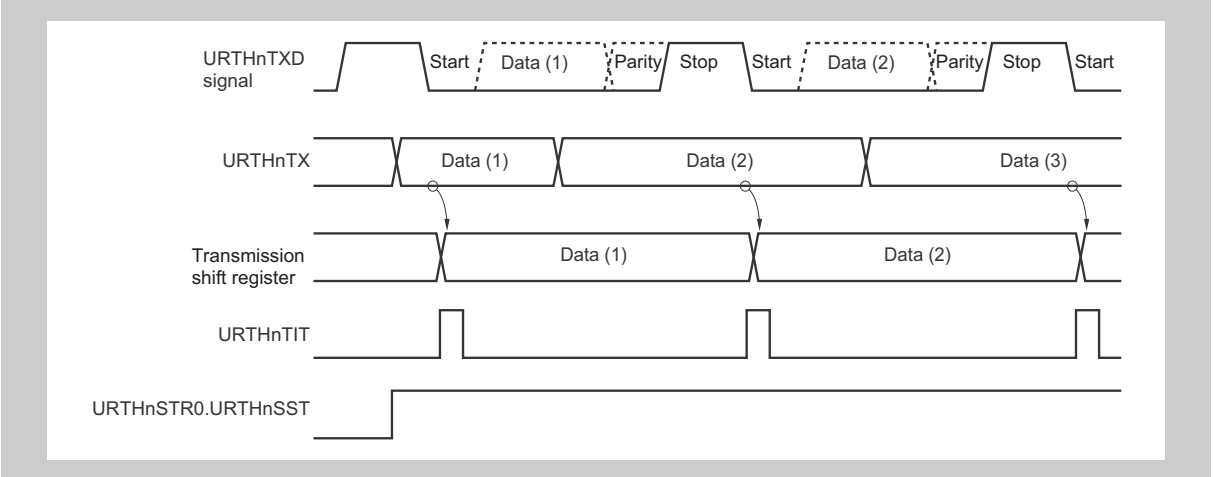


Figure 23-13 Continuous Transmission Operation Timing (Transmission Start)

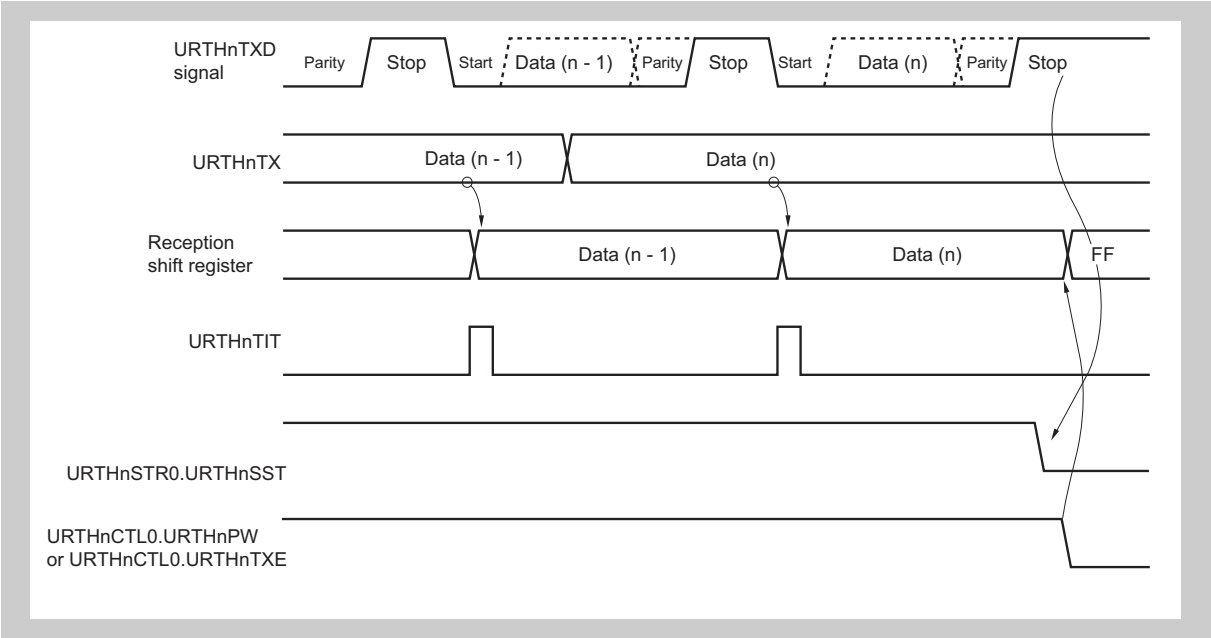


Figure 23-14 Continuous Transmission Operation Timing (Transmission End)

23.6.12 UARTHn reception

Reception start

Set the reception enabled status by the following procedure:

- Specify the baud rate by the UARTHn control register 2 (URTHnCTL2).
- Specify the receive parity, data character length, stop bit length, receive data order, and output logic level by the UARTHn control register 1 (URTHnCTL1).
- Enable UARTHn operation and reception by setting URTHnCTL0.URTHnPW = URTHnCTL0.URTHnRXE = 1.

When the sampling of the input level of the URTHnTRXD pin is performed and the falling edge is detected, the data sampling of the URTHnRXD input is started.
The start bit is recognized if the URTHnRXD pin is low level after the time of a half bit is passed after the detection of the falling edge (shown in the figure below).
After a start bit has been recognized, the receive operation starts, and serial data are stored in the reception shift register according to the set baud rate.
When the reception interrupt URTHnTIR is asserted upon reception of the stop bit, the data stored in the reception shift register are written to the receive data register (URTHnRX).

Reception stop

When URTHnCTL0.URTHnPW or URTHnCTL0.URTHnRXE is set to 0, reception operations are stopped immediately, even during reception processing.

Reception errors

If an overrun error occurs (URTHnSTR1.URTHnOVE = 1), the receive data at this time is not transferred to the URTHnRX register and is discarded.

Even if a parity error (URTHnSTR1.URTHnPE = 1) occurs during reception, reception continues until the stop bit, and the reception data are transferred to the RX register. And, a framing error (URTHnSTR1.URTHnFE = 1) is judged at the stop bit position, and even if an error is detected, the received date is transferred to the URTHnRX register.

An error in reception in this case leads to generation of the status interrupt (URTHnTIS) instead of the reception interrupt (URTHnTIR).

Clear the power bit to 0 (URTHnCTL0.URTHnPW = 0) before changing the order of data reception, parity, character length, or stop bit length.

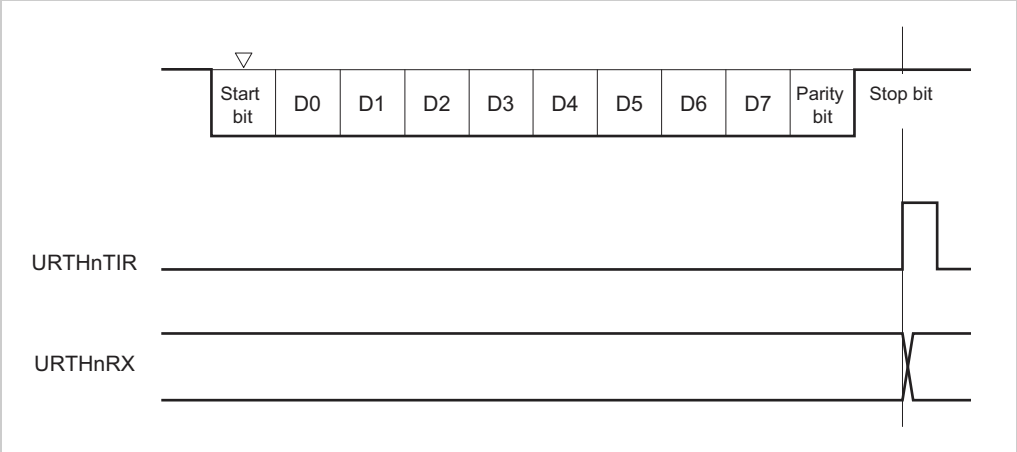


Figure 23-15 UARTHn Reception

Caution 1. Be sure to read the URTHnRX register even when a reception error occurs. If the URTHnRX register is not read, an overrun error occurs during reception of the next data.

Caution 2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

Caution 3. When trying to reduce current flow by stopping the unit on completion of reception, read the URTHnRX register after the reception interrupt URTHnTIR has been generated, and then clear the URTHnCTL0.URTHnPW or URTHnCTL0.URTHnRXE bit to 0. If the URTHnCTL0.URTHnPW or URTHnCTL0.URTHnRXE bit is cleared to 0 before the URTHnTIR is generated, the value read from the URTHnRX register cannot be guaranteed.

Caution 4. If processing to indicate completion of reception (generation of the URTHnTIR interrupt) and clearing of the URTHnCTL0.URTHnPW or URTHnCTL0.URTHnRXE bit to 0 occur simultaneously, the URTHnTIR may still be generated despite the URTHnRX register holding no data.

Note 1. When URTHnCTL0.URTHnPW/URTHnCTL0.URTHnRXE is changed from 0 to 1 while URTHnRXD pin is set to 0, it is not judged as the start bit, because falling edges are not detected.

Note 2. In continuous reception, the subsequent start bit may be detected immediately after the stop bit (in response to which the reception interrupt is generated).

23.6.13 Reception Errors

Errors during a receive operation are of 3 types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the URTHnSTR1 register and a status interrupt request signal URTHnTIS is generated when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the URTHnSTR1 register.

Clear a reception error flag by writing 1 to its associated bit in the status clear register URTHnSTC.

Table 23-31 Sources of Reception Errors

Error Flag in URTHnSTR1	Reception Error	Source
URTHnPE	Parity error	Received parity bit does not match the setting
URTHnFE	Framing error	Stop bit is not detected
URTHnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

Note Even in case of a parity or framing error, data are transferred from the reception shift register to the receive data register URTHnRX. Consequently the data from URTHnRX must be read. Otherwise an overrun error (URTHnSTR1.URTHnOVE = 1) will occur when reception of the next data is completed.

In case of an overrun error, data in the reception shift register are not transferred to the URTHnRX register, so the previous data are not overwritten.

23.6.14 Parity Types and Operations

Caution When using the LIN function, fix the URTHnCTL1.URTHnSLP[1:0] to 00_B.

The parity bit is used to detect bit errors in the communication data. In general, use the same parity on both sides (transmission and reception).

(1) Even Parity

- During transmission

  The parity bit is controlled so that the number of bits with the value "1" among the data is even. The values of the parity bit are as follows.

  – Odd number of bits having the value "1" among those for transmission: 1

  – Even number of bits having the value "1" among those for transmission: 0

- During reception

  The number of bits having the value "1" among the received data, including the parity bit, is counted. An odd result produces a parity error.

(2) Odd parity

- During transmission

  In the opposite way to even parity, the parity bit is controlled so that the number of bits having the value "1" among the data is odd. The values of the parity bit are as follows.

  – Odd number of bits having the value "1" among those for transmission: 0

  – Even number of bits having the value "1" among those for transmission: 1

- During reception

  The number of bits having the value "1" among the received data, including the parity bit, is counted. An even result produces a parity error.

(3) 0 Parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

23.6.15 Digital Receive Data Noise Filter

Each of the input pins for data reception signals (URTHnTRXD, URTHnSC, and URTHnCTS) is equipped with a digital noise filter for eliminating noise and spikes.

A filter samples the signal on the URTHnRXD, URTHnSC, or URTHnCTS pin with the prescaler output clock (PRSCLK) as the sampling clock.

When the same sampling value is read twice, the signal of each pin is validated as the input data.

Therefore, data not exceeding the width of 2 prescaler output clocks is judged to be noise and thus eliminated.

The noise filter leads to a delay of 4 cycles of the clock signal output by the prescaler (PRSCLK) from capture of each serial bit until it is forwarded as valid.

The noise filter circuit runs when reception is enabled (URTHnPW = URTHnRXE = 1) and in processing for transmission with transmission and data consistency error checking enabled (URTHnPW = URTHnTXE = URTHnSLDC = 1).

23.7 Baud-Rate Generator

The transmission and reception baud-rate clocks (BRCLK) are derived from the PCLK by using a prescaler and a baud-rate generator (refer to the figure below).

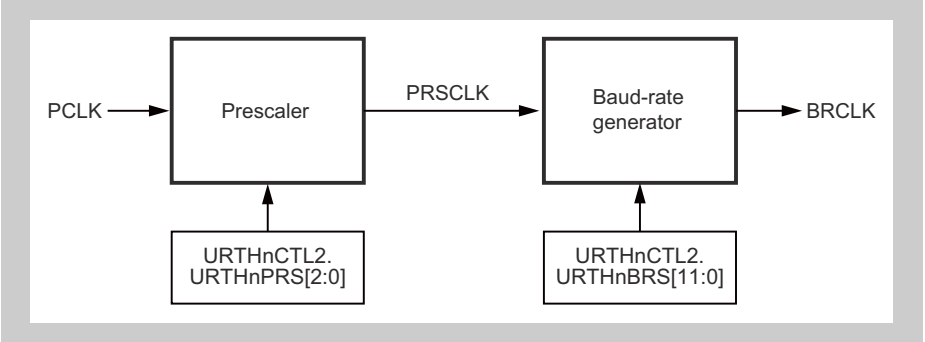


Figure 23-16 Configuration of a Baud-Rate Generator

The prescaler output clock PRSCLK is at a fraction of the frequency of PCLK. The setting of URTHnCTL2.URTHnPRS[2:0] determines the denominator of this fraction.

$$PRSCLK = PCLK/2^{URTHnPRS[2:0]}$$

PRSCLK is further divided by the baud-rate generator by a value, determined by URTHnCTL2.URTHnBRS[11:0].

The baud-rate generator distinguishes between the baud rate for data frames and BF receptions, as listed in the table below. The BF reception clock is the double of the baud-rate clock BRCLK.

Table 23-32 Baud-Rate Generator Clocks Output

URTHnCTL2. URTHnBRS[11:0]	Transmit/Receive BRCLK	BF Receive Clock
000_H	PRSCLK/(2 × 2)	PRSCLK/2
001_H		
002_H		
003_H	PRSCLK/(2 × 3)	PRSCLK/3
004_H	PRSCLK/(2 × 4)	PRSCLK × 4
005_H	PRSCLK/(2 × 5)	PRSCLK × 5
...	PRSCLK/(2 × URTHnBRS[11:0])	PRSCLK/URTHnBRS[11:0]
FFE_H	PRSCLK/(2 × 4094)	PRSCLK/4094
FFF_H	PRSCLK/(2 × 4095)	PRSCLK/4095

23.8 Detecting the Baud Rate in LIN Communications as a Slave

This product has a facility for input from the serial data input pins to TAUA0 and TAUA1. Using this function leads to activation of the detection of baud rates in LIN communications as a slave.

Note the following points when using this facility.

- Use TAUA0 and TAUA1 in an operating mode in which baud-rate detection is possible.
- Do not change the settings of the IC0REG[0:1] registers while the timer is in operation.
- Do not use this function to connect a serial data input pin to TAUA0 and TAUA1 for any purpose other than detecting the baud rate in LIN communications.

Section 24 FlexRay™ (FLX)

Note FlexRay™ is a trademark of Daimler AG.

This section contains a generic description of the FlexRay controller.

The first part describes all properties specific to the V850E2/Px4, such as instances, register base addresses, input/output signal names, etc.
The subsequent parts describe the features that apply to all implementations.

24.1 V850E2/Px4 FLXn Features

Instances This microcontroller has following number of instances of the FlexRay controller FLXn.

Table 24-1 Instances of FLXn

FlexRay Controller	
Instances	1
Name	FLX0

Instances index n Throughout this section, the instance of the FlexRay controller is identified by the index "n" (n = 0), for example, FLXnSUCC1 for the FLXn SUC configuration register 1.

Register addresses All FlexRay controller register addresses are given as address offsets to the individual base address <FLXn_base>.
The <FLXn_base> addresses of FLXn are listed in the following table.

Table 24-2 Register Base Addresses <FLXn_base>

FLXn Instance	<FLXn_base> Address
FLX0	FF58 0000_H

The FlexRay controller FLXn comprises the FlexRay core module E-Ray as the major function. An additional message buffer with the error correction module ECC completes the FlexRay controller.

Vendor information The value of the FLXnVI register contains product specific vendor information. The vendor information of the V850E2/Px4 is given in the following table.

Table 24-3 Vendor Information Register FLXnVI Value

FLXn Instance	FLXnVI
FLX0	$0105\ 0004_H$

Clock supply All FlexRay controllers provide two clock inputs.

Table 24-4 FLXn Clock Supply

FLXn Instance	FLXn Clock	Connected to
FLX0	eray_sclk	Clock generator FLXCLK = 80 MHz
	eray_bclk	Clock generator PCLK

Note 1. Set PCLK to 64 MHz or under (only for µPD70F3506 and µPD70F3507 products).

Note 2. There is the following limitation in the function when PCLK operates at 27 MHz or 45 MHz: The header section modification for the dynamic slot being set by the software during FlexRay communication may not be reflected to the message buffer at the time of frame transmission. (This also applies to the case where the modification has completed before the start of message buffer scan for the target slot.)
[Countermeasures]
The above limitation can be removed by satisfying any one of the following application usage condition and the setting conditions.

- Avoiding header section change of dynamic slots during FlexRay communication

  <The following is the case when PCLK = 32 MHz and MT = 1 µs>
- Setting gdMinislot (FLXnGTUC08.FLXnMSL[5:0]) ≥ 3 MT
- Setting the number of message buffers to 59 or less (FIFO not included)

In the description of the FlexRay controller, special signal names are used, which differ from the names used in other sections of this document. Provided in the following tables is the correspondence between the FlexRay signal names and those used in the other sections.

Interrupts The FlexRay controller can generate the following interrupts.

Table 24-5 FLXn Interrupts (1/3)

FLXn Signal	E-Ray Signal	Function	Connected to
FRINTP0	eray_int0	Interrupt 0	Interrupt controller INTFLX0I0 DTS
FRINTP1	eray_int1	Interrupt 1	Interrupt controller INTFLX0I1 DTS
FRTINTP0	eray_tint0	Timer interrupt 0	Interrupt controller INTFLX0I2
FRTINTP1	eray_tint1	Timer interrupt 1	Interrupt controller INTFLX0I3
FRIBUSY	eray_ibusy	IBF to MBF transmission busy	No connection destination
FROBUSY	eray_obusy	MBF to OBF transmission busy	No connection destination
-	-	ECC error detected	Interrupt controller INTECCFRED

Table 24-5 FLXn Interrupts (2/3)

FLXn Signal	E-Ray Signal	Function	Connected to
FREIFL0	-	POC error mode change detected	No connection destination
FREIFL1	-	Command not received	No connection destination
FREIFL2	-	Number of sync frames not enough	No connection destination
FREIFL3	-	Sync frame overflow	No connection destination
FREIFL4	-	Clock correction failed	No connection destination
FREIFL5	-	CHI command lock error	No connection destination
FREIFL6	-	Receive FIFO overrun error	No connection destination
FREIFL7	-	Empty FIFO access error	No connection destination
FREIFL8	-	Input buffer invalid access	No connection destination
FREIFL9	-	Output buffer invalid access	No connection destination
FREIFL10	-	Message handler restriction error	No connection destination
FREIFL11	-	Channel A error detected	No connection destination
FREIFL12	-	Channel A last transmission violation	No connection destination
FREIFL13	-	Channel A transmit slot boundary violation	No connection destination
FREIFL14	-	Channel B error detected	No connection destination
FREIFL15	-	Channel B last transmission violation	No connection destination
FREIFL16	-	Channel B transmit slot boundary violation	No connection destination
FRSIFL0	-	Wakeup status change flag	No connection destination
FRSIFL1	-	Collision avoidance symbol receive flag	No connection destination
FRSIFL2	-	Communication cycle start flag	Interrupt controller INTFLX0SIFL02
FRSIFL3	-	Transmission completion flag	No connection destination
FRSIFL4	-	Reception completion flag	No connection destination
FRSIFL5	-	Receive FIFO data existence flag	No connection destination
FRSIFL6	-	Receive FIFO limit reached flag	No connection destination
FRSIFL7	-	Network management vector change flag	Interrupt controller INTFLX0SIFL07
FRSIFL8	-	Input buffer transmission completion flag	Interrupt controller INTFLX0SIFL08 DTS
FRSIFL9	-	Output buffer transmission completion flag	Interrupt controller INTFLX0SIFL09 DTS
FRSIFL10	-	Stop watch event occurrence flag	No connection destination
FRSIFL11	-	Startup normal completion flag	No connection destination

Table 24-5 FLXn Interrupts (3/3)

FLXn Signal	E-Ray Signal	Function	Connected to
FRSIFL12	-	Message buffer status update flag	No connection destination
FRSIFL13	-	Dynamic segment start flag	No connection destination
FRSIFL14	-	Channel A wakeup pattern receive flag	No connection destination
FRSIFL15	-	Channel A MTS symbol receive flag	No connection destination
FRSIFL16	-	Wakeup pattern receive flag	No connection destination
FRSIFL17	-	Channel B MTS symbol receive flag	No connection destination

Reset signal The FlexRay controller is initialized by the following reset signal.

Table 24-6 FLXn Reset Signal

FLXn Signal	E-Ray Signal	Function	Name
FRSRESZ	eray_reset	Reset E-Ray module	FLXA0RES

I/O signals The I/O signals of the FlexRay controller are listed in the following table. Every I/O signal is connected to a port.

Table 24-7 FLXn I/O Signals

FLXn Signals	E-Ray Signal	Function	Connected to
FRTXDA	eray_txd1	Data transmitter output 1	Port FTXDA
FRRXDA	eray_rxd1	Data receiver input 1	Port FRXDA
FRTXENA	eray_txen1	Transmit enable signal 1	Port FTXENA
FRTXDB	eray_txd2	Data transmitter output 2	Port FTXDB
FRRXDB	eray_rxd2	Data receiver input 2	Port FRXDB
FRTXENB	eray_txen2	Transmit enable signal 2	Port FTXENB
FRSWT	eray_stpwt	Stop watch trigger	No connection destination

24.2 Error Correction Module (ECC)

The message RAM, the input buffer RAM, the output buffer RAM, and the transient buffer RAM are protected by the error correction (ECC) module. The ECC module automatically encodes or decodes ECC codes when E-Ray module or CPU accesses the message RAM.

When an ECC correction occurs in reading data from RAM, the ECC module corrects the error without notification of the error.

When an ECC error occurs, the ECC module notifies the E-Ray module of the error and generates an interrupt signal of INTECCFRED.

Note If an ECC error with additional 1-bit or longer error occurs, the ECC module may not be able to detect the error. Moreover, correction of error bits or generation of an interrupt or both of them may be done.

The error correction module constitutes the following register structure.

(1) E7AxCTL (x = 0 to 6) – FlexRay ECC0/1/2/3/4/5/6 Control Register)

This register sets the respective bit to 1 when a FlexRay RAM ECC error occurs, or clears the value that has been set. A bit is cleared to 0 by writing a 1 to the corresponding bit.

Access This register can be read/written in 32-bit units.

The result of reading E7AxCTL.E7AxCTLER2C (x = 0 to 6) is always 0.

Address FF46 4000$_H$ (E7A0CTL)

FF46 4100$_H$ (E7A1CTL)

FF46 4200$_H$ (E7A2CTL)

FF46 4300$_H$ (E7A3CTL)

FF46 4400$_H$ (E7A4CTL)

FF46 4500$_H$ (E7A5CTL)

FF46 4600$_H$ (E7A6CTL)

Initial value Undefined

This register is initialized by a reset from any source.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	E7AxCTLER2C	0	0	0	0	0	0	0	E7AxCTLECCER2	0/1	0/1
R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

Table 24-8 E7AxCTL Register (x = 0 to 6) Contents

Bit Position	Bit Name	Function
10	E7AxCTLER2C (x = 0 to 6)	0: E7AxCTL.E7AxCTLECCER2 (x = 0 to 6) not cleared 1: E7AxCTL.E7AxCTLECCER2 (x = 0 to 6) cleared
2	E7AxCTLECCER2 (x = 0 to 6)	0: FlexRay RAM ECC error does not exist. 1: FlexRay RAM ECC error exists. [Clear (0) condition] • Writing a 1 to E7AxCTL.E7A0CTLER2C (x = 0 to 6)

24.3 FlexRay Controller Overview

24.3.1 Conventions

The following conventions are used within this document

- CAPITALS POC states and CHI commands

24.3.2 Definition

- FlexRay frame: Header segment + Payload segment
- Message buffer: Header section + Data section
- Message RAM: Header partition + Data partition
- Data frame: FlexRay frame that is not a null frame

24.3.3 References

This section refers to the following documents.

Table 24-9 References

Reference No.	Author(s)	Title
1	FlexRay Group	FlexRay Communication System Protocol Specification v2.1 Revision A (05/12/22)
2	BOSCH AE/EIP	E-Ray FlexRay IP-Module Specification Revision 1.2.6

24.3.4 Terms and Abbreviations

This section uses the following terms and abbreviations.

Table 24-10 Terms and Abbreviations

Term	Meaning
AP	Action point
BD	Bus driver
BSS	Byte start sequence
CAS	Collision avoidance symbol
CC	Communication controller
CHI	Controller host interface
CRC	Cyclic redundancy check
FES	Frame end sequence
FIFO	First in first out (message buffer structure)
FSM	Finite state machine
FSP	Frame and symbol processing block
FSS	Frame start sequence
FTM	Fault tolerant midpoint
GTU	Global time unit block
HIF	Host CPU interface block
IBF	Input buffer
INT	Interrupt control block
MHD	Message handler
MT	Macrotick
MTS	Media access test symbol
NCT	Network communication time
NEM	Network management block
NIT	Network idle time
NM	Network management
OBF	Output buffer
POC	Protocol operation control
PRT	Protocol controller block
SDL	Specification and description language
SUC	System universal control block
TBF	Transient buffer
TDMA	Time division multiple access (media access method)
TSS	Transmission start sequence
TT-D	Time triggered distributed synchronization (protocol mode)
µT	Microtick
WUP	Wakeup Pattern
WUS	Wakeup Symbol

24.3.5 Functional Overview

The E-Ray IP module of the FlexRay controller performs communication according to the FlexRay protocol specification version 2.1. With maximum specified sample clock, the bit rate can be programmed to values up to 10 Mbit/s. Additional bus driver (BD) hardware is required for connection to the physical layer.

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message storage consists of a single-ported message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the message handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay channel protocol controllers and the message RAM, and maintaining the transmission schedule as well as providing message status information.

The register set of the E-Ray IP-module can be accessed directly by the V850E2/Px4. These registers are used to control/configure/monitor the FlexRay channel protocol controllers, message handler, global time unit, system universal control, frame and symbol processing, network management, interrupt control, and to access the message RAM via input/output buffers.

The E-Ray IP-module supports the following features.

- Conformance to FlexRay protocol specification v2.1
- Conformance to E-Ray specification v1.2.6
- Data rates of up to 10 Mbit/s on each channel
- Up to 128 message buffers configurable
- Message RAM of 8 Kbytes for data storage including:
  - 128 messages buffers each of which consisting of a data section of up to 48 bytes
  - 30 messages buffers each of which consisting of a data section of up to 254 bytes
- Configuration of message buffers with different payload lengths possible
- One configurable receive FIFO
- Each message buffer can be configured as a receive buffer, a transmit buffer, or a receive FIFO.
- Support the access to message buffers via input/output buffers
  Input buffer: holds messages to be transferred to the message RAM
  Output buffer: holds messages read from the message RAM
- Filtering based on the slot counter, the cycle counter, and channels
- Interrupt masks can be configured for each interrupt cause individually
- Network management supported

24.3.6 Block Diagram

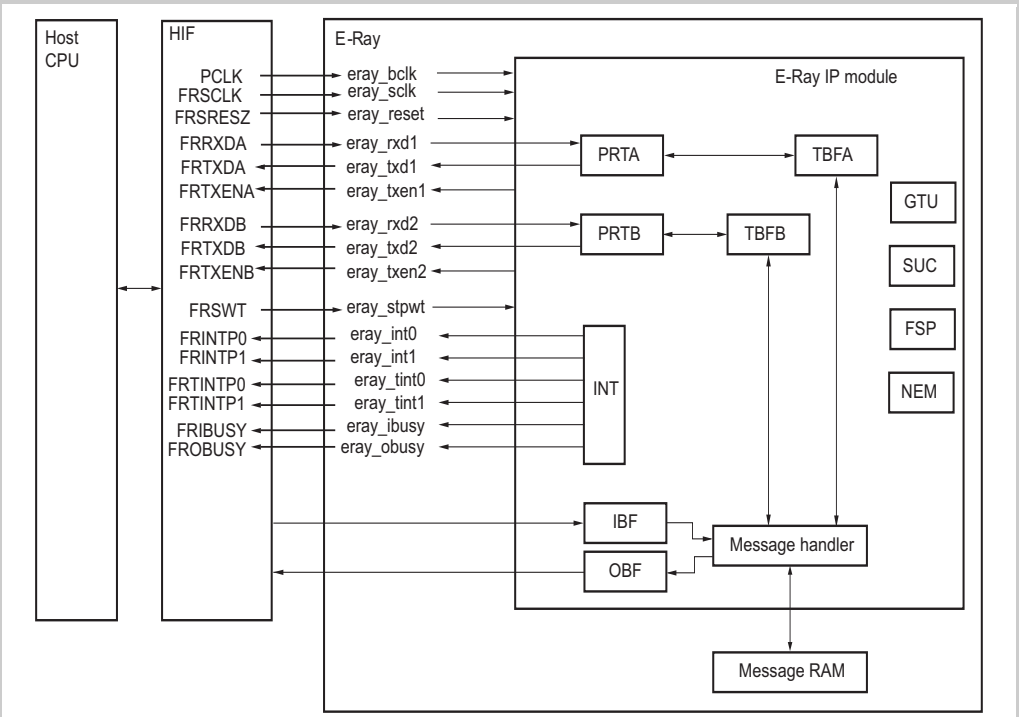


Figure 24-1 FlexRay Block Diagram

(1) Host Interface (HIF)

The host interface connects the CPU to the E-Ray IP-module. Configuration registers, status registers, and interrupt registers are attached to the respective blocks and can be accessed via the host interface.

(2) Input Buffer (IBF)

For write access to the message buffers configured in the message RAM, the host CPU can write the header and data section for a specific message buffer to the input buffer. The message handler then transfers the data from the input buffer to the selected message buffer in the message RAM.

(3) Output Buffer (OBF)

For read access to a message buffer configured in the message RAM, the message handler transfers the selected message buffer to the output buffer. After the transfer has completed, the host CPU can read the header and data section of the transferred message buffer from the output buffer.

(4) Message Handler (MHD)

The message handler controls data transfers between the following components:

- Input/output buffer and message RAM
- Transient buffer RAMs of the two FlexRay protocol controllers and message RAM

(5) TBF A/B (Transient Buffer RAM A/B)

Two channels of this type are provided for temporarily storing message data sections when sending or receiving messages.

(6) PRT A/B (FlexRay Channel Protocol Controllers)

The FlexRay channel protocol controllers consist of shift register and FlexRay protocol FSM. They are connected to the transient buffer RAMs for intermediate message storage and to the physical layer via bus driver BD.

They perform the following functionality:

- Control and check of bit timing
- Reception/transmission of FlexRay frames and symbols
- Check of header CRC
- Generation/check of frame CRC
- Interfacing to bus driver

The FlexRay channel protocol controllers have interfaces to:

- Physical layer (bus driver)
- Transient buffer RAM
- Message handler
- Global time unit
- System universal control
- Frame and symbol processing
- Network management
- Interrupt control

(7) Global Time Unit (GTU)

The global time unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
  - Rate correction
  - Offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

(8) System Universal Control (SUC)

The system universal control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal operation (NORMAL_ACTIVE state)
- Passive operation (NORMAL_PASSIVE state)
- Monitor mode (MONITOR_MODE state)

(9) Frame and Symbol Processing (FSP)

The frame and symbol processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

(10) Network Management (NEM)

The network management performs the following functions:

- Handling of the network management vector

(11) Interrupt Control (INT)

The interrupt controller performs the following functions:

- Provides error and status interrupt flags
- Controls enable/disable of interrupt sources
- Controls assignment of interrupt sources to the two module interrupt lines
- Enable/disable of the two module interrupt lines
- Manages the two interrupt timers
- Stop watch time capturing

24.3.7 Reset Timing

When the hardware is left in the reset state, an internal procedure is started that initializes the module-internal RAM blocks to zero. The module-internal RAMs can also be cleared by CHI command CLEAR_RAMS (FLXnSUCC1.FLXnCMD[3:0] = 1100) when the FlexRay controller is in the DEFAULT_CONFIG or CONFIG state. The initialization of the FlexRay internal RAM blocks requires 2048 eray_bclk cycles.

No host CPU access to IBF or OBF is possible during initialization of the internal RAM blocks after hardware reset or after assertion of CHI command CLEAR_RAMS.

After hardware reset, all registers hold the reset values as summarized in table 24-12, FlexRay Controller Register Map. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS.

During the initialization of the RAMs, FLXnSUCC1.FLXnPBSY will show POC busy.

24.4 Programmer's Model

24.4.1 Register Map

The FlexRay controller allocates an address space of 2 Kbytes (0000_H to $07FF_H$). The registers are organized as 32-bit registers. Host access to the message RAM is done via the input/output buffers. They buffer data to be transferred to and from the message RAM under control of the message handler, avoiding conflicts between host accesses and message reception/ transmission.

The assignment of the message buffers is done according to the scheme shown in table 24-11, Assignment of Message Buffers, below. The number N of available message buffers depends on the payload length of the configured message buffers. The maximum number of message buffers is 128. The maximum payload length supported is 254 bytes.

Message buffer size

The size of the implemented message buffer is defined in section 24.5.12, Message RAM.

The message buffers are separated into three consecutive groups.

Static buffers	Transmit/receive buffers assigned to static segment
Static buffers + Dynamic buffers	Transmit/receive buffers assigned to static segment or dynamic segment
FIFO	Receive FIFO

The message buffer separation configuration can be changed in the DEFAULT_CONFIG or CONFIG state only by programming the message RAM configuration register.

The first group starts with message buffer 0 and consists of static message buffers only. Message buffer 0 is dedicated for holding the startup/sync frame or the single slot frame, when the node transmits a frame of these types according to the setting of FLXnTXST, FLXnTXSY, and FLXnTSM in the SUC configuration register 1. In this case, message buffer 0 has to be configured with the key slot ID and can only be (re)configured in the DEFAULT_CONFIG or CONFIG state.

The second group consists of message buffers assigned to the static or dynamic segment.

The message buffers belonging to the third group are concatenated to a single receive FIFO.

Table 24-11 Assignment of Message Buffers

Message buffer 0	Static buffers
Message buffer 1	
…	Static + Dynamic buffers
	FIFO
Message buffer N-1	
Message buffer N	

Offset address The addresses in the table below are given as offsets to the base address <FLXn_base>. <FLXn_base> of the FLXn is defined in section 24.1 under the key word "Register addresses".

Table 24-12 FlexRay Controller Register Map (1/4)

Offset Address	Symbol	Name	Reset	Access	Block
HIF registers					
0000_H	FLXnCI	Controller information	6572 6179_H	R	HIF
0004_H	FLXnVI	Vendor information	*[1]	R	
0008_H	FLXnCS	Control setting	0000 0002_H	R/W	
$000C_H$		Reserved	0000 0000_H	R	
Special registers					
0010_H	FLXnTEST1	Test register 1	0000 0300_H	R/W	HIF
0014_H	FLXnTEST2	Test register 2	0000 0000_H	R/W	
0018_H		Reserved	0000 0000_H	R	
$001C_H$	FLXnLCK	Lock register	0000 0000_H	R/W	HIF
Interrupt registers					
0020_H	FLXnEIR	Error interrupt register	0000 0000_H	R/W	INT
0024_H	FLXnSIR	Status interrupt register	0000 0000_H	R/W	
0028_H	FLXnEILS	Error interrupt line select	0000 0000_H	R/W	
$002C_H$	FLXnSILS	Status interrupt line select	0303 $FFFF_H$	R/W	
0030_H	FLXnEIES	Error interrupt enable set	0000 0000_H	R/W	
0034_H	FLXnEIER	Error interrupt enable reset	0000 0000_H	R/W	
0038_H	FLXnSIES	Status interrupt enable set	0000 0000_H	R/W	
$003C_H$	FLXnSIER	Status interrupt enable reset	0000 0000_H	R/W	
0040_H	FLXnILE	Interrupt line enable	0000 0000_H	R/W	
0044_H	FLXnT0C	Timer 0 configuration	0000 0000_H	R/W	
0048_H	FLXnT1C	Timer 1 configuration	0002 0000_H	R/W	
$004C_H$	FLXnSTPW1	Stop watch register 1	0000 0000_H	R/W	
0050_H	FLXnSTPW2	Stop watch register 2	0000 0000_H	R	

Table 24-12 FlexRay Controller Register Map (2/4)

Offset Address	Symbol	Name	Reset	Access	Block
0054_H to $007C_H$		Reserved	$0000\ 0000_H$	R	
CC control registers					
0080_H	FLXnSUCC1	SUC configuration register 1	$0C40\ 1080_H$	R/W	SUC
0084_H	FLXnSUCC2	SUC configuration register 2	$0100\ 0504_H$	R/W	
0088_H	FLXnSUCC3	SUC configuration register 3	$0000\ 0011_H$	R/W	
$008C_H$	FLXnNEMC	NEM configuration register	$0000\ 0000_H$	R/W	NEM
0090_H	FLXnPRTC1	PRT configuration register 1	$084C\ 0633_H$	R/W	PRT
0094_H	FLXnPRTC2	PRT configuration register 2	$0F2D\ 0A0E_H$	R/W	
0098_H	FLXnMHDC	MHD configuration register	$0000\ 0000_H$	R/W	MHD
$009C_H$		Reserved	$0000\ 0000_H$	R	
$00A0_H$	FLXnGTUC01	GTU configuration register 1	$0000\ 0280_H$	R/W	GTU
$00A4_H$	FLXnGTUC02	GTU configuration register 2	$0002\ 000A_H$	R/W	
$00A8_H$	FLXnGTUC03	GTU configuration register 3	$0202\ 0000_H$	R/W	
$00AC_H$	FLXnGTUC04	GTU configuration register 4	$0008\ 0007_H$	R/W	
$00B0_H$	FLXnGTUC05	GTU configuration register 5	$0E00\ 0000_H$	R/W	
$00B4_H$	FLXnGTUC06	GTU configuration register 6	$0002\ 0000_H$	R/W	
$00B8_H$	FLXnGTUC07	GTU configuration register 7	$0002\ 0004_H$	R/W	
$00BC_H$	FLXnGTUC08	GTU configuration register 8	$0000\ 0002_H$	R/W	
$00C0_H$	FLXnGTUC09	GTU configuration register 9	$0000\ 0101_H$	R/W	
$00C4_H$	FLXnGTUC10	GTU configuration register 10	$0002\ 0005_H$	R/W	
$00C8_H$	FLXnGTUC11	GTU configuration register 11	$0000\ 0000_H$	R/W	
$00CC_H$ to $00FC_H$		Reserved	$0000\ 0000_H$	R	
CC status registers					
0100_H	FLXnCCSV	CC status vector	$0010\ 4000_H$	R	SUC
0104_H	FLXnCCEV	CC error vector	$0000\ 0000_H$	R	
0108_H to $010C_H$		Reserved	$0000\ 0000_H$	R	
0110_H	FLXnSCV	Slot counter value	$0000\ 0000_H$	R	GTU
0114_H	FLXnMTCCV	Macrotick and cycle counter value	$0000\ 0000_H$	R	
0118_H	FLXnRCV	Rate correction value	$0000\ 0000_H$	R	
$011C_H$	FLXnOCV	Offset correction value	$0000\ 0000_H$	R	
0120_H	FLXnSFS	Sync frame status	$0000\ 0000_H$	R	
0124_H	FLXnSWNIT	Symbol window and NIT status	$0000\ 0000_H$	R	
0128_H	FLXnACS	Aggregated channel status	$0000\ 0000_H$	R/W	
$012C_H$		Reserved	$0000\ 0000_H$	R	
0130_H to 0168_H	FLXnESIDm	Even sync ID [01…15]	$0000\ 0000_H$	R	GTU
$016C_H$		Reserved	$0000\ 0000_H$	R	
0170_H to $01A8_H$	FLXnOSIDm	Odd sync ID [01…15]	$0000\ 0000_H$	R	GTU

Table 24-12 FlexRay Controller Register Map (3/4)

Offset Address	Symbol	Name	Reset	Access	Block
$01AC_H$		Reserved	$0000\ 0000_H$	R	
$01B0_H$ to $01B8_H$	FLXnNMVm	Network management vector [1…3]	$0000\ 0000_H$	R	NEM
$01BC_H$ to $02FC_H$		Reserved	$0000\ 0000_H$	R	
Message buffer control registers					
0300_H	FLXnMRC	Message RAM configuration	$0180\ 0000_H$	R/W	MHD
0304_H	FLXnFRF	FIFO rejection filter	$0180\ 0000_H$	R/W	
0308_H	FLXnFRFM	FIFO rejection filter mask	$0000\ 0000_H$	R/W	
$030C_H$	FLXnFCL	FIFO critical level	$0000\ 0080_H$	R/W	
Message buffer status registers					
0310_H	FLXnMHDS	Message handler status	$0000\ 0000_H$	R/W	MHD
0314_H	FLXnLDTS	Last dynamic transmit slot	$0000\ 0000_H$	R	
0318_H	FLXnFSR	FIFO status register	$0000\ 0000_H$	R	
$031C_H$	FLXnMHDF	Message handler constraints flags	$0000\ 0000_H$	R/W	
0320_H	FLXnTXRQ1	Transmission request 1	$0000\ 0000_H$	R	
0324_H	FLXnTXRQ2	Transmission request 2	$0000\ 0000_H$	R	
0328_H	FLXnTXRQ3	Transmission request 3	$0000\ 0000_H$	R	
$032C_H$	FLXnTXRQ4	Transmission request 4	$0000\ 0000_H$	R	
0330_H	FLXnNDAT1	New data 1	$0000\ 0000_H$	R	
0334_H	FLXnNDAT2	New data 2	$0000\ 0000_H$	R	
0338_H	FLXnNDAT3	New data 3	$0000\ 0000_H$	R	
$033C_H$	FLXnNDAT4	New data 4	$0000\ 0000_H$	R	
0340_H	FLXnMBSC1	Message buffer status changed 1	$0000\ 0000_H$	R	
0344_H	FLXnMBSC2	Message buffer status changed 2	$0000\ 0000_H$	R	
0348_H	FLXnMBSC3	Message buffer status changed 3	$0000\ 0000_H$	R	
$034C_H$	FLXnMBSC4	Message buffer status changed 4	$0000\ 0000_H$	R	
0350_H to $03EC_H$		Reserved	$0000\ 0000_H$	R	
Identification registers					
$03F0_H$	FLXnCREL	Core release register	$1027\ 1031_H$	R	GIF
$03F4_H$	FLXnENDN	Endian register	$8765\ 4321_H$	R	
$03F8_H$ to $03FC_H$		Reserved	$0000\ 0000_H$	R	
Input buffer control registers					
0400_H to $04FC_H$	FLXnWRDSm	Write data section [01…64]	$0000\ 0000_H$	R/W	IBF
0500_H	FLXnWRHS1	Write header section 1	$0000\ 0000_H$	R/W	
0504_H	FLXnWRHS2	Write header section 2	$0000\ 0000_H$	R/W	
0508_H	FLXnWRHS3	Write header section 3	$0000\ 0000_H$	R/W	
$050C_H$		Reserved	$0000\ 0000_H$	R	

Table 24-12 FlexRay Controller Register Map (4/4)

Offset Address	Symbol	Name	Reset	Access	Block
0510_H	FLXnIBCM	Input buffer command mask	$0000\ 0000_H$	R/W	IBF
0514_H	FLXnIBCR	Input buffer command request	$0000\ 0000_H$	R/W	
0518_H to $05FC_H$		Reserved	$0000\ 0000_H$	R	
Output buffer control registers					
0600_H to $06FC_H$	FLXnRDDSm	Read data section [01…64]	$0000\ 0000_H$	R	OBF
0700_H	FLXnRDHS1	Read header section 1	$0000\ 0000_H$	R	
0704_H	FLXnRDHS2	Read header section 2	$0000\ 0000_H$	R	
0708_H	FLXnRDHS3	Read header section 3	$0000\ 0000_H$	R	
$070C_H$	FLXnMBS	Message buffer status	$0000\ 0000_H$	R	
0710_H	FLXnOBCM	Output buffer command mask	$0000\ 0000_H$	R/W	
0714_H	FLXnOBCR	Output buffer command request	$0000\ 0000_H$	R/W	
0718_H to $07FC_H$		Reserved	$0000\ 0000_H$	R	

Note 1. The vendor information of register FLXnVI is defined in section 24.1 under the key word "Vendor information".

24.4.2 HIF Registers

(1) FLXnCI – Controller Information

This register contains FlexRay controller information.

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0000_H

Initial value $6572\ 6179_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnMN[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMN[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-13 FLXnCI Register Contents

Bit Position	Bit Name	Function
31 to 0	FLXn MN[31:00]	Controller information This field contains a number that indicates the controller type. The E-Ray FlexRay controller stores the hex number $6572\ 6179_H$. This value corresponds to an ASCII code meaning "eray".

(2) FLXnVI – Vendor Information

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0004_H

Initial value Refer to the key word "Vendor information" in section 24.1. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnVI[7:0]								FLXnFMR[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FLXnPCN[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-14 FLXnVI Register Contents

Bit Position	Bit Name	Function
31 to 24	FLXnVI[7:0]	Vendor ID This field contains the identification number of the silicon vendor.
23 to 16	FLXnFMR[7:0]	FlexRay controller release number This field contains the release number of the controller.
7 to 0	FLXnPCN[7:0]	Product code number This field contains the product code number of the controller.

Vendor information The vendor information of register FLXnVI is defined in section 24.1 under the key word "Vendor information".

(3) FLXnCS – Control Setting

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0008_H

Initial value $0000\ 0002_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnCSLK[7:0]								0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXnMD	FLXnSR
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24-15 FLXnCS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	FLXnCSLK[7:0]	Control setting lock key To set and clear the FLXnSR bit and FLXnMD bit, the write operation has to be directly preceded by two consecutive write accesses to the FLXnCS register. If the host accesses to other FlexRay controller register in between this consecutive sequence, writing to the FLXnCS register is simply ignored. Reading these bits returns 0.
1	FLXnMD	Controller disable If this bit is set, internal clock distribution (eray_sclk, eray_bclk) is halted. Therefore, software cannot access registers/RAM of FlexRay controller when FLXnMD = 1. In case of that, write access will be simply ignored and read access will get incorrect value. Exceptionally, microcontroller specific HIF registers (i.e. FLXnCI/FLXnVI/FLXnCS) can be accessed even when FLXnMD = 1. Unlock sequence by the FLXnCSLK bit is required for setting and clearing the FLXnMD bit. The user must set FLXnMD in the CONFIG state and must wait a transfer between RAMs to complete. Otherwise, the last received message may be destroyed due to sudden clock stopping. Note that the read access to FLXnTEST1 during FLXnMD = 1 does not cause clearing FLXnTEST1.FLXnCERA and FLXnCERB. A module reset will be performed after setting FLXnMD and the RAMs cannot be accessed until FLXnSUCC1.FLXnPBSY is cleared. Note: After power up of the microcontroller, this bit is set and the FlexRay controller cannot be accessed until this bit is set to 0.

	Set FLXnSR Bit	Clear FLXnSR Bit	Set FLXnMD Bit	Clear FLXnMD Bit
First write to FLXnCS register	$67xx_xxxx_H$	$67xx_xxxx_H$	$67xx_xxxx_H$	$67xx_xxxx_H$
Second write to FLXnCS register	$98xx_xxx1_H$	$98xx_xxx0_H$	$98xx_xxx2_H$	$98xx_xxx0_H$

Table 24-15 FLXnCS Register Contents (2/2)

Bit Position	Bit Name	Function
0	FLXnSR	Software reset If this bit is set, a reset will be executed immediately. The user must clear the FLXnSR bit in order to release a reset. Unlock sequence by the FLXnCSLK bit is required for setting and clearing the FLXnSR bit. Note: After a software reset, the RAMs cannot be accessed until FLXnSUCC1.FLXnPBSY is cleared.

Accessing to FlexRay controller registers and the internal RAM is restricted by the combination of the FLXnSR and FLXnMD bits.

FLXn SR	FLXn MD	HIF Registers (FLXnCI, FLXnVI, FLXnCS)	FlexRay Controller Registers	FlexRay Controller RAM
0	0	R/W possible	R/W possible	R/W possible
0	1	R/W possible	Read only*1	Not possible*2
1	0	R/W possible	Not possible*2	Not possible*2
1	1	R/W possible	Not possible*2	Not possible*2

Note 1. Write access is simply ignored.
Note 2. In read access, invalid data is read, and write access is simply ignored.

The table below explains the possible bit transition of the FLXnSR and FLXnMD bits.

FLXnSR	FLXnMD	Bit Transition
0 → 1	0	Allowed
0	0 →1	Allowed
0 → 1	0 →1	Ignored
1	0 →1	Allowed
0 →1	1	Ignored
0 →1	1 →0	Allowed

There is no restriction in clearing the FLXnSR and FLXnMD bits.

24.4.3 Special Registers

(1) FLXnTEST1 – Test Register 1

The test register 1 holds the control bits to configure the test modes of the FlexRay controller. Write access to these bits is only possible if bit FLXnWRTEN is set to 1.

When the FlexRay controller is operated in one of its test modes that requires FLXnWRTEN to be set (RAM test mode, I/O test mode, asynchronous transmit mode, and loop back mode), only the selected test mode functionality is available.

The test functions cannot be used together with the normal operating mode functions because they change the functions of parts of the FlexRay controller. Therefore, normal operation as specified outside this section and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions.

The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0010_H

Initial value $0000\ 0300_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnCERB[3:0]				FLXnCERA[3:0]				0	0	FLXn TXENB	FLXn TXENA	FLXn TXB	FLXn TXA	FLXn RXB	FLXn RXA
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	FLXn AOB	FLXn AOA	0	0	FLXnTMC [1:0]		0	0	FLXn ELBE	FLXn WRTEN
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 24-16 FLXnTEST1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	FLXn CERB[3:0]	Coding error report channel B These bits are set when a coding error is detected on channel B. They are reset to zero when register FLXnTEST1 is read or written. Once the FLXnCERB[3:0] bits are set, they will remain unchanged until the FLXnTEST1 register is accessed. 0000 = No coding error detected 0001 = Header CRC error detected 0010 = Frame CRC error detected 0011 = Frame start sequence FSS too long 0100 = First bit of byte start sequence BSS seen low 0101 = Second bit of byte start sequence BSS seen high 0110 = First bit of frame end sequence FES seen high 0111 = Second bit of frame end sequence FES seen low 1000 = CAS/MTS symbol seen too short 1001 = CAS/MTS symbol seen too long 1010...1111 = Reserved
27 to 24	FLXn CERA[3:0]	Coding error report channel A These bits are set when a coding error is detected on channel A. They are reset to zero when register FLXnTEST1 is read or written. Once the FLXnCERA[3:0] bits are set, they will remain unchanged until the FLXnTEST1 register is accessed. 0000 = No coding error detected 0001 = Header CRC error detected 0010 = Frame CRC error detected 0011 = Frame start sequence FSS too long 0100 = First bit of byte start sequence BSS seen low 0101 = Second bit of byte start sequence BSS seen high 0110 = First bit of frame end sequence FES seen high 0111 = Second bit of frame end sequence FES seen low 1000 = CAS/MTS symbol seen too short 1001 = CAS/MTS symbol seen too long 1010...1111 = Reserved
21	FLXn TXENB	Control of channel B transmit enable signal 0 = eray_txen2_n signal drives a 0. 1 = eray_txen2_n signal drives a 1.
20	FLXn TXENA	Control of channel A transmit enable signal 0 = eray_txen1_n signal drives a 0 1 = eray_txen1_n signal drives a 1
19	FLXn TXB	Control of channel B transmit signal 0 = eray_txd2 signal drives a 0 1 = eray_txd2 signal drives a 1
18	FLXn TXA	Control of channel A transmit signal 0 = eray_txd1 signal drives a 0 1 = eray_txd1 signal drives a 1
17	FLXn RXB	Monitor channel B receive signal n 0 = eray_rxd2 = 0 1 = eray_rxd2 = 1
16	FLXn RXA	Monitor channel A receive signal 0 = eray_rxd1 = 0 1 = eray_rxd1 = 1

Table 24-16 FLXnTEST1 Register Contents (2/2)

Bit Position	Bit Name	Function
9	FLXn AOB	Activity detected on channel B FLXnAOB is set to 1 when there is activity on channel B. It is reset to 0 when 11 consecutive 1 bits are detected or if the POC state is DEFAULT_CONFIG or CONFIG. During STARTUP, NORM L_ACTIVE, or NORMAL_PASSIVE, the function of FLXnAOB is the inverse of zChannelIdle as specified in the FlexRay protocol spec v2.1, section 3, BITSTRB process. FLXnAOB should be ignored in all other POC states. 1 = Activity detected, channel B not idle 0 = No activity detected, channel B idle
8	FLXn AOA	Activity detected on channel A FLXnAOA is set to 1 when there is activity on channel A. It is reset to 0 when 11 consecutive 1 bits are detected or if the POC state is DEFAULT_CONFIG or CONFIG. During STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE, the function of FLXnAOA is the inverse of zChannelIdle as specified in the FlexRay protocol spec v2.1, section 3, BITSTRB process. FLXnAOA should be ignored in all other POC states. 1 = Activity detected, channel A not idle 0 = No activity detected, channel A idle
5, 4	FLXn TMC[1:0]	Test multiplexer control The FLXnTMC[1:0] bits can be written to in the DEFAULT_CONFIG state or CONFIG state. The value "00" will be written if these bits are written in other states. 00 = Normal signal path (default) 01 = RAM test mode - Internal busses are multiplexed to make all RAM blocks of the FlexRay controller directly accessible by the host. This mode is intended to enable testing of the embedded RAM blocks during production testing. 10 = I/O test mode - Output pins eray_txd1, eray_txd2, eray_txen1_n, and eray_txen2_n are driven to the values defined by bits FLXnTXA, FLXnTXB, FLXnTXENA, and FLXnTXENB. The values applied to the input pins eray_rxd1 and eray_rxd2 can be read from register bits FLXnRXA and FLXnRXB. 11 = Normal signal path
1	FLXn ELBE	External loop back enable There are two possibilities to perform a loop back test: external loop back via physical layer or internal loop back for in-system self-test (default). In case of an internal loop back, signals eray_txen1 and eray_txen2 are in their inactive state, signals eray_txd1 and eray_txd2 are set to high, and signals eray_rxd1 and eray_rxd2 are not evaluated. Bit FLXnELBE is evaluated only when POC is in loop back mode and test multiplexer control is in non-multiplexing mode FLXnTMC[1:0] = 00. 1 = External loop back 0 = Internal loop back (default)
0	FLXn WRTEN	Write test register enable Enables write access to the test registers. To set the bit from 0 to 1, the test mode key has to be written as defined in the description on (3), FLXnLCK – Lock Register. The unlock sequence is not required for changes in other bits of the register if FLXnWRTEN is kept 1. The bit can be reset to 0 at any time. 1 = Write access to the test register is enabled. 0 = Write access to the test register is disabled.

Note 1. Coding errors are signaled in all states where frame decoding is possible. FLXnCERA[3:0] and FLXnCERB[3:0] must be ignored in all other states. The error codes regarding CAS/MTS symbols concern only the monitored bit pattern, regardless of whether those bit patterns are seen in the symbol window or elsewhere.

Note 2. The following test register 1 bits are used to test the interface to the physical layer (connectivity test) by driving or reading the respective signals: FLXnTXENB, FLXnTXENA, FLXnTXB, FLXnTXA, FLXnRXB, FLXnRXA

Asynchronous transfer mode (ATM)

The asynchronous transfer mode is entered by writing FLXnSUCC1.FLXnCMD[3:0] = 1110 while the FlexRay controller is in the CONFIG state and bit FLXnTEST1.FLXnWRTEN is set to 1. This write operation has to be directly preceded by two consecutive write accesses to the configuration lock key (unlock sequence). When called in any other state or when bit FLXnTEST1.FLXnWRTEN is not set, FLXnSUCC1.FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted). Reading FLXnCCSV.FLXnPOCS[5:0] will return 00 1110 while the FlexRay controller is in ATM mode. Asynchronous transfer mode can be left by writing FLXnSUCC1.FLXnCMD[3:0] = 0001 (FLXnCHI command: CONFIG).

In ATM mode, transmission of a FlexRay frame is triggered by writing the number of the respective message buffer to FLXnIBCR.FLXnIBRH[6:0] while FLXnIBCM.STXR is set to 1. In this mode, wakeup, startup, and clock synchronization are bypassed. The CHI command SEND_MTS results in the immediate transmission of an MTS symbol.

The cycle counter value of frames sent in ATM mode can be programmed via FLXnMTCCV.FLXnCCV[5:0] (writeable in ATM and loop back modes only).

Loop back mode

The loop back mode is entered by writing FLXnSUCC1.FLXnCMD[3:0] to 1111 when CONFIG state is entered after transition to DEFAULT_CONFIG state is made by resetting the FlexRay controller (FLXnCS.FLXnSR) and bit FLXnTEST1.FLXnWRTEN is set to 1. This write operation has to be directly preceded by two consecutive write accesses to the configuration lock key (unlock sequence). When called in any other state or when FLXnTEST1.FLXnWRTEN is not set, FLXnSUCC1.FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted). Reading FLXnCCSV.FLXnPOCS[5:0] will return 00 1101 while the FlexRay controller is in loop back mode.

Loop back mode can be left by writing FLXnSUCC1.FLXnCMD[3:0] = 0001 (CHI command: CONFIG).

The loop back mode is intended to check the module's internal data paths. Normal time triggered operation is not possible in loop back mode.

There are two possibilities to perform a loop back test: external loop back via physical layer (FLXnTEST1.FLXnELBE = 1) or internal loop back for in-system self-test (FLXnTEST1.FLXnELBE = 0). In case of an internal loop back, signals eray_txen1 and eray_txen2 are in their inactive state, signals eray_txd1 and eray_txd2 are set to high, and signals eray_rxd1 and eray_rxd2 are not evaluated.

When the FlexRay controller is in loop back mode, a loop back test is started by the host writing a message to the input buffer and requesting the transmission by writing to register FLXnIBCR. The message handler will transfer the message into the message RAM and then into the transient buffer of the selected channel. The channel protocol controller (PRT) will read (in 32-bit words) the message from the transmit part of the transient buffer and load it into its Rx/Tx shift register. The serial transmission is looped back into the shift

register; its content is written into the receive part of the channel's transient buffer before the next word is loaded.

The PRT and the message handler will then treat this transmitted message like a received message, perform an acceptance filtering on frame ID and receive channel, and store the message into the message RAM if it passed acceptance filtering. The loop back test ends with the host requesting this received message from the message RAM and then checking the contents of the output buffer.

Each FlexRay channel is tested separately. The FlexRay controller cannot receive messages from the FlexRay bus while it is in the loop back mode.

The cycle counter value of frames used in loop back mode can be programmed via FLXnMTCCV.FLXnCCV[5:0] (writeable in ATM and loop back modes only).

Note that in case of an odd payload (PLC = 1, 3, 5, ...), the last two bytes of the looped-back payload will be shifted by 16 bits to the right inside the last 32-bit data word.

(2) FLXnTEST2 – Test Register 2

The test register 2 holds bits required for the RAM test of the seven embedded RAM blocks of the FlexRay controller. Write access to this register is only possible when FLXnTEST1.WRTEN is set to 1.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0014_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	FLXn SSEL[2:0]			0	FLXn RS[2:0]		
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 24-17 FLXnTEST2 Register Contents

Bit Position	Bit Name	Function
6 to 4	FLXnSSEL [2:0]	RAM segment setting The message RAM (8192 byte address) is divided into multiple segments so that it can be accessed at any part of it. 000 = Access to RAM bytes 0000_H to $03FF_H$ is allowed. 001 = Access to RAM bytes 0400_H to $07FF_H$ is allowed. 010 = Access to RAM bytes 0800_H to $0BFF_H$ is allowed. 011 = Access to RAM bytes $0C00_H$ to $0FFF_H$ is allowed. 100 = Access to RAM bytes 1000_H to $13FF_H$ is allowed. 101 = Access to RAM bytes 1400_H to $17FF_H$ is allowed. 110 = Access to RAM bytes 1800_H to $1BFF_H$ is allowed. 111 = Access to RAM bytes $1C00_H$ to $1FFF_H$ is allowed.
2 to 0	FLXnRS[2:0]	RAM selection setting In RAM test mode, the RAM blocks selected by FLXnRS[2:0] are mapped to module addresses 400_H to $7FF_H$ (1024 byte address). 000 = Input buffer RAM 1 (IBF1) 001 = Input buffer RAM 2 (IBF2) 010 = Output buffer RAM 1 (OBF1) 011 = Output buffer RAM 2 (OBF2) 100 = Transient buffer RAM A (TBF1) 101 = Transient buffer RAM B (TBF2) 110 = Message RAM (MBF) 111 = Unused

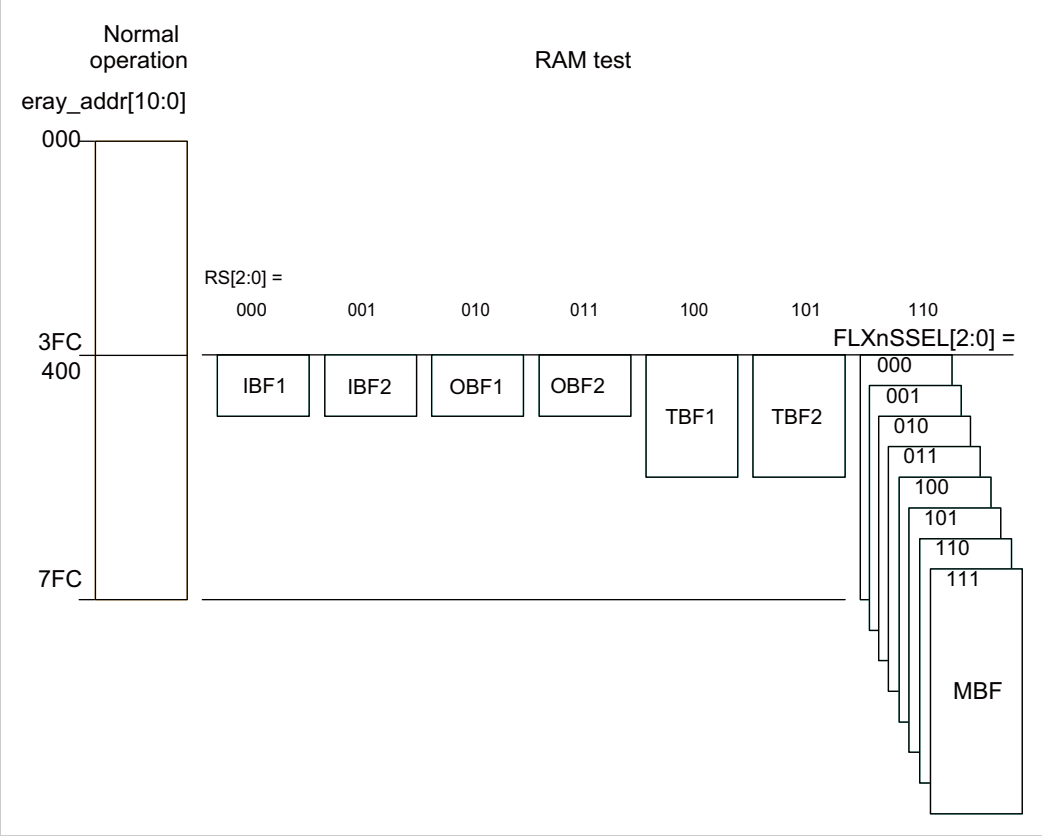
RAM test mode

In RAM test mode (FLXnTEST1.FLXnTMC[1:0] = 01), one of the seven RAM blocks can be selected for direct read/write access by setting FLXnTEST2.FLXnRS[2:0].

To allow external access, the selected RAM block is mapped to address space 400_H to $7FF_H$ (1024 byte address or 256 word address).

Because the length of the message RAM exceeds the available address space, the message RAM is segmented into segments of 1024 bytes. The segments can be selected by setting FLXnTEST2.FLXnSSEL[2:0].

Note When accessing MBF, do not access an area outside the connected MBF area.


Normal operation
RAM test
eray_addr[10:0]
000
3FC
400
7FC
RS[2:0] =
000
001
010
011
100
101
110
FLXnSSEL[2:0] =
IBF1
IBF2
OBF1
OBF2
TBF1
TBF2
000
001
010
011
100
101
110
111
MBF

(3) FLXnLCK – Lock Register

The lock register is write-only. Reading the register will return 0000 0000_H.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $001C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnTMK[7:0]								FLXnCLK[7:0]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 24-18 FLXnLCK Register Contents

Bit Position	Bit Name	Function
15 to 8	FLXnTMK [7:0]	Test mode key To write bit FLXnTEST1.FLXnWRTEN, the write operation has to be directly preceded by two consecutive write accesses to the test mode key (unlock sequence). If the write sequence specified below is interrupted by other write accesses between the second write to the test mode key and the write access to the FLXnTEST1 register, FLXnTEST1.FLXnWRTEN is not set to 1 and the sequence has to be repeated. First write: FLXnLCK.TMK[7:0] = 0111 0101 (75_H) Second write: FLXnLCK.TMK[7:0] = 1000 1010 ($8A_H$) Third write: FLXnTEST1.FLXnWRTEN = 1
7 to 0	FLXnCLK [7:0]	Configuration lock key To leave the CONFIG state by writing FLXnSUCC1.FLXnCMD[3:0] (commands READY, MONITOR_MODE, ATM, LOOP_BACK), the write operation has to be directly preceded by two write accesses to the configuration lock key (unlock sequence). If the write sequence specified below is interrupted by other write accesses between the second write to the configuration lock key and the write access to the FLXnSUCC1 register, the FlexRay controller remains in the CONFIG state and the sequence has to be repeated. First write: FLXnLCK.CLK[7:0] = 1100 1110 (CE_H) Second write: FLXnLCK.CLK[7:0] = 0011 0001 (31_H) Third write: FLXnSUCC1.FLXnCMD[3:0]

24.4.4 Interrupt Registers

(1) FLXnEIR – Error Interrupt Register

The flags are set when the FlexRay controller detects one of the listed error conditions. They remain set until the host clears them. A flag is cleared by writing a 1 to the corresponding bit position. After writing 1, read the register again to confirm that the corresponding bit has actually been cleared. If it has not been cleared, perform the operation to clear it again. Writing a 0 has no effect on the flag. A hard reset will also clear the register.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0020_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnEIR Read

31	30	29	28	27	26	25	24
0	0	0	0	0	FLXn TABBE	FLXn LTVBE	FLXn EDBE

23	22	21	20	19	18	17	16
0	0	0	0	0	FLXn TABAE	FLXn LTVAE	FLXn EDAE

15	14	13	12	11	10	9	8
0	0	0	0	FLXn MHFE	FLXn IOBAE	FLXn IIBAE	FLXn EFAE

7	6	5	4	3	2	1	0
FLXn RFOE	0	FLXn CCLE	FLXn CCFE	FLXn SFOE	FLXn SFBME	FLXn CNAE	FLXn PEMCE

Table 24-19 FLXnEIR Read Register Contents (1/3)

Bit Position	Bit Name	Function
Channel-specific error flags of channel B:		
26	FLXn TABBE	Transmission slot boundary violation on channel B The flag signals to the host that a transmission across a slot boundary occurred on channel B. 1 = Transmission across slot boundary detected on channel B 0 = No transmission across slot boundary detected on channel B
25	FLXn LTVBE	Latest transmit violation on channel B The flag signals a latest transmit violation on channel B to the host. 1 = Latest transmit violation detected on channel B 0 = No latest transmit violation detected on channel B
24	FLXn EDBE	Error detected on channel B This bit is set whenever one of the flags FLXnACS.FLXnSEDBE, FLXnACS.FLXnCEDBE, FLXnACS.FLXnCIBE, or FLXnACS.FLXnSBVBE changes from 0 to 1. 1 = Error detected on channel B 0 = No error detected on channel B

Table 24-19 FLXnEIR Read Register Contents (2/3)

Bit Position	Bit Name	Function
Channel-specific error flags of channel A:		
18	FLXn TABAE	Transmission slot boundary violation on channel A The flag signals to the host that a transmission across a slot boundary occurred on channel A. 1 = Transmission across slot boundary detected on channel A 0 = No transmission across slot boundary detected on channel A
17	FLXn LTVAE	Latest transmit violation on channel A The flag signals a latest transmit violation on channel A to the host. 1 = Latest transmit violation detected on channel A 0 = No latest transmit violation detected on channel A
16	FLXn EDAE	Error detected on channel A This bit is set whenever one of the flags FLXnACS.FLXnSEDAE, FLXnACS.FLXnCEDAE, FLXnACS.FLXnCIAE, or FLXnACS.FLXnSBVAE changes from 0 to 1. 1 = Error detected on channel A 0 = No error detected on channel A
Flags for all channels:		
11	FLXn MHFE	Message handler constraints flag The flag signals a message handler constraints violation condition. It is set whenever one of the flags FLXnMHDF.FLXnSNUAE, FLXnMHDF.FLXnSNUBE, FLXnMHDF.FLXnFNFAE, MHDF.FLXnFNFBE, FLXnMHDF.FLXnDTBFAE, FLXnMHDF.FLXnDTBFBE, or FLXnMHDF.FLXnWAHPE changes from 0 to 1. 1 = Message handler failure detected 0 = No message handler failure detected
10	FLXn IOBAE	Illegal output buffer access This flag is set by the FlexRay controller when the host requests the transfer of a message buffer from the message RAM to the output buffer while FLXnOBCR.FLXnOBSYS is set to 1. 1 = Illegal host access to output buffer occurred 0 = No illegal host access to output buffer occurred
9	FLXn IIBAE	Illegal input buffer access This flag is set by the FlexRay controller when the host attempts to modify a message buffer via input buffer in one of the following conditions. • The FlexRay controller is not in the CONFIG or DEFAULT_CONFIG state and the host writes to the input buffer command request register to modify: - The header section of message buffer 0, 1 if configured for transmission in key slot - The header section of static message buffers whose numbers are less than FLXnMRC.FLXnFDB[7:0] while FLXnMRC.FLXnSEC[1:0] = 01 - The header section of any static or dynamic message buffer while FLXnMRC.FLXnSEC[1:0] = 1x - The header and/or data section of any message buffer belonging to the receive FIFO • The host writes to any register of the input buffer while FLXnIBCR.FLXnIBSYH is set to 1. 1 = Illegal host access to input buffer occurred 0 = No illegal host access to input buffer occurred
8	FLXn EFAE	Empty FIFO access This flag is set by the FlexRay controller when the host requests the transfer of a message from the receive FIFO via output buffer while the receive FIFO is empty. 1 = Host access to empty FIFO occurred 0 = No host access to empty FIFO occurred

Table 24-19 FLXnEIR Read Register Contents (3/3)

Bit Position	Bit Name	Function
7	FLXn RFOE	Receive FIFO overrun The flag is set by the FlexRay controller when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FLXnFSR. 1 = A receive FIFO overrun has been detected 0 = No receive FIFO overrun detected
5	FLXn CCLE	CHI command locked The flag signals that the write access to the CHI command vector FLXnSUCC1.FLXnCMD[3:0] was not successful because the execution of the previous CHI command has not yet completed. In this case, issuing the CHI command sets this bit to 1 and bit FLXnCNAE is also set to 1. 1 = CHI command not accepted 0 = CHI command accepted
4	FLXn CCFE	Clock correction failure This flag is set at the end of the cycle whenever one of the following errors occurs: • Missing offset and/or rate correction • Clock correction limit reached The clock correction status is monitored in registers FLXnCCEV and FLXnSFS. A failure may occur during startup. Therefore, bit FLXnCCFE must be cleared by the host after the FlexRay controller entered the NORMAL_ACTIVE state. 1 = Clock correction failed 0 = No clock correction error
3	FLXn SFOE	Sync frame overflow This bit is set when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by FLXnGTUC02.FLXnSNM[3:0]. 1 = More sync frames received than configured by FLXnGTUC02.FLXnSNM[3:0] 0 = Number of received sync frames is less than configured by FLXnGTUC02.FLXnSNM[3:0]
2	FLXn SFBME	Sync frames below minimum This flag signals that the number of sync frames received during the last communication cycle was below the limit required by the FlexRay protocol. This error may occur during startup and therefore must be cleared by the host after the FlexRay controller entered the NORMAL_ACTIVE state. 1 = Less than the required minimum of sync frames received 0 = Sync node: 1 or more sync frames received Non-sync node: 2 or more sync frames received
1	FLXn CNAE	Command not accepted This flag signals that the write access to the CHI command vector FLXnSUCC1.FLXnCMD[3:0] was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (FLXnCCLE = 1). 1 = CHI command not accepted 0 = CHI command accepted
0	FLXn PEMCE	POC error mode changed This flag is set whenever the error mode signaled by FLXnCCEV.FLXnERRM[1:0] has changed. 1 = Error mode has changed 0 = Error mode has not changed

(b) FLXnEIR Write

31	30	29	28	27	26	25	24
					FLXnCL TABB	FLXnCL LTVB	FLXnCL EDB

23	22	21	20	19	18	17	16
					FLXnCL TABA	FLXnCL LTVA	FLXnCL EDA

15	14	13	12	11	10	9	8
				FLXnCL MHF	FLXnCL IOBA	FLXnCL IIBA	FLXnCL EFA

7	6	5	4	3	2	1	0
FLXnCL RFO		FLXnCL CCL	FLXnCL CCF	FLXnCL SFO	FLXnCL SFBM	FLXnCL CNA	FLXnCL PEMC

Table 24-20 FLXnEIR Write Register Contents

Bit Position	Bit Name	Function
31 to 0	See above	Clears the corresponding flags described in table 24-19, FLXnEIR Read Register Contents. 1 = Clears the corresponding flag. 0 = No function

Note Writing to bits 31 to 27, 23 to 19, 15 to 12, and 6 is ignored.

(2) FLXnSIR – Status Interrupt Register

The flags are set when the FlexRay controller detects one of the listed events. The flags remain set until the host clears them. A flag is cleared by writing a 1 to the corresponding bit position. After writing 1, read the register again to confirm that the corresponding bit has actually been cleared. If it has not been cleared, perform the operation to clear it again. Writing a 0 has no effect on the flag. A hard reset will also clear the register.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0024_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnSIR Read

31	30	29	28	27	26	25	24
0	0	0	0	0	0	FLXn MTSBF	FLXn WUPBF

23	22	21	20	19	18	17	16
0	0	0	0	0	0	FLXn MTSAF	FLXn WUPAF

15	14	13	12	11	10	9	8
FLXn SDSF	FLXn MBSIF	FLXn SUCSF	FLXn SWEF	FLXn TOBCF	FLXn TIBCF	FLXn TI1F	FLXn TI0F

7	6	5	4	3	2	1	0
FLXn NMVCF	FLXn RFCLF	FLXn RFNEF	FLXn RXIF	FLXn TXIF	FLXn CYCSF	FLXn CASF	FLXn WSTF

Table 24-21 FLXnSIR Read Register Contents (1/3)

Bit Position	Bit Name	Function
Channel-specific status flags:		
25	FLXn MTSBF	MTS received on channel B (vSS!ValidMTSB) The media access test symbol (MTS) received on channel B during the preceding symbol window. This bit is updated at the end of the symbol window. 1 = MTS symbol received on channel B 0 = No MTS symbol received on channel B
24	FLXn WUPBF	Wakeup pattern received on channel B This flag is set by the FlexRay controller when a wakeup pattern was received on channel B. This bit is set only when the FlexRay controller is in the WAKEUP, READY, or STARTUP state, or when in MONITOR_MODE state. 1 = Wakeup pattern received on channel B 0 = No wakeup pattern received on channel B
17	FLXn MTSAF	MTS received on channel A (vSS!ValidFLXnMTSA) The media access test symbol (MTS) received on channel A during the preceding symbol window. This bit is updated at the end of the symbol window. 1 = MTS symbol received on channel A 0 = No MTS symbol received on channel A

Table 24-21 FLXnSIR Read Register Contents (2/3)

Bit Position	Bit Name	Function
16	FLXn WUPAF	Wakeup pattern received on channel A This flag is set by the FlexRay controller when a wakeup pattern was received on channel A. This bit is set only when the FlexRay controller is in the WAKEUP, READY, or STARTUP state, or when in MONITOR_MODE state. 1 = Wakeup pattern received on channel A 0 = No wakeup pattern received on channel A
Flags for all channels:		
15	FLXn SDSF	Start of dynamic segment This flag is set by the FlexRay controller when the dynamic segment starts. 1 = Dynamic segment started 0 = Dynamic segment not yet started
14	FLXn MBSIF	Message buffer status interrupt This flag is set by the FlexRay controller when the message buffer status MBS has changed and if bit FLXnRHMBI of that message buffer is set. 1 = Message buffer status of at least one message buffer with FLXnRHMBI = 1 has changed 0 = No message buffer status change of message buffer with FLXnRHMBI = 1
13	FLXn SUCSF	Startup completed successfully This flag is set whenever a startup completes successfully and the FlexRay controller enters the NORMAL_ACTIVE state. 1 = Startup completed successfully 0 = No startup completed successfully
12	FLXn SWEF	Stop watch event This flag is set when a stop watch is enabled and the actual cycle counter and macrotick value are stored in the stop watch register (FLXnSTPW1). 1 = Stop watch event occurred 0 = No stop watch Event
11	FLXn TOBCF	Transfer output buffer completed This flag is set whenever a transfer from the message RAM to the output buffer has completed and bit FLXnOBCR.FLXnOBSYS in the output buffer command request register has been reset by the message handler. 1 = Transfer between message RAM and output buffer complete 0 = No transfer completed since bit was reset
10	FLXn TIBCF	Transfer input buffer completed This flag is set whenever a transfer from input buffer to the message RAM has completed and bit FLXnIBCR.FLXnIBSYS in the input buffer command request register has been reset by the message handler. 1 = Transfer between input buffer and message RAM completed 0 = No transfer completed since bit was reset
9	FLXn TI1F	Timer interrupt 1 This flag is set whenever timer 1 matches the conditions configured in register FLXnT1C. A timer interrupt 1 is signaled on signal eray_tint1. 1 = Timer interrupt 1 occurred 0 = No timer interrupt 1
8	FLXn TI0F	Timer interrupt 0 This flag is set whenever timer 0 matches the conditions configured in register FLXnT0C. A timer interrupt 0 is signaled on signal eray_tint0. 1 = Timer interrupt 0 occurred 0 = No timer interrupt 0
7	FLXn NMVCF	Network management vector changed This interrupt flag signals a change in the network management vector. 1 = Network management vector changed 0 = No change in the network management vector

Table 24-21 FLXnSIR Read Register Contents (3/3)

Bit Position	Bit Name	Function
6	FLXn RFCLF	Receive FIFO critical level This flag is set when the receive FIFO fill level FLXnFSR.FLXnRFFL[7:0] is equal or greater than the critical level as configured by FLXnFCL.FLXnCL[7:0]. 1 = Receive FIFO critical level reached 0 = Receive FIFO below critical level
5	FLXn RFNEF	Receive FIFO storage This flag is set by the FlexRay controller when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register FLXnFSR. 1 = Receive FIFO is not empty 0 = Receive FIFO is empty
4	FLXn RXIF	Receive interrupt This flag is set by the FlexRay controller whenever the set condition of a message buffers ND flag is fulfilled (see New Data 1/2/3/4 (FLXnNDAT1/2/3/4)), and if bit FLXnRHMBI of that message buffer is set to 1. 1 = At least one data section has been updated 0 = No data section has been updated
3	FLXn TXIF	Transmit interrupt This flag is set by the FlexRay controller after successful frame transmission if bit FLXnWHMBI in the respective message buffer is set to 1. 1 = At least one frame was transmitted successfully 0 = No frame transmitted
2	FLXn CYCSF	Communication cycle start interrupt This flag is set by the FlexRay controller when a communication cycle starts. 1 = Communication cycle started 0 = No communication cycle started
1	FLXn CASF	Collision avoidance symbol This flag is set by the FlexRay controller during the STARTUP state when a CAS or a potential CAS was received. 1 = Bit pattern matching the CAS symbol received 0 = No bit pattern matching the CAS symbol received
0	FLXn WSTF	Wakeup status This flag is set when the wakeup status vector FLXnCCSV.FLXnWSV[2:0] is changed to a value other than UNDEFINED. 1 = Wakeup status changed 0 = Wakeup status unchanged

(b) FLXnSIR Write

31	30	29	28	27	26	25	24
						FLXn CLMTSB	FLXn CLWUPB

23	22	21	20	19	18	17	16
						FLXn CLMTSA	FLXn CLWUPA

15	14	13	12	11	10	9	8
FLXn CLSDS	FLXn CLMBSI	FLXn CLSUCS	FLXn CLSWE	FLXn CLTOBC	FLXn CLTIBC	FLXn CLTI1	FLXn CLTI0

7	6	5	4	3	2	1	0
FLXn CLNMVC	FLXn CLRFCL	FLXn CLRFNE	FLXn CLRXI	FLXn CLTXI	FLXn CLCYCS	FLXn CLCAS	FLXn CLWST

Table 24-22 FLXnSIR Write Register Contents

Bit Position	Bit Name	Function
31 to 0	See above	Clears the corresponding flags described in table 24-21, FLXnSIR Read Register Contents. 1 = Clears the corresponding flag. 0 = No function

Note Writing to bits 31 to 26 and 23 to 18 is ignored.

(3) FLXnEILS – Error interrupt Line Select Register

The error interrupt line select register assigns an interrupt generated by a specific error interrupt flag from register FLXnEIR to one of the two interrupt lines:

1 = Interrupt assigned to interrupt line eray_int1

0 = Interrupt assigned to interrupt line eray_int0

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0028_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXn TABBL	FLXn LTVBL	FLXn EDBL	0	0	0	0	0	FLXn TABAL	FLXn LTVAL	FLXn EDAL
R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FLXn MHFL	FLXn IOBAL	FLXn IIBAL	FLXn EFAL	FLXn RFOL	0	FLXn CCLL	FLXn CCFL	FLXn SFOL	FLXn SFBML	FLXn CNAL	FLXn PEMCL
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-23 FLXnEILS Register Contents (1/2)

Bit Position	Bit Name	Function
26	FLXn TABBL	Transmission across boundary channel B interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
25	FLXn LTVBL	Latest transmit violation channel B interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
24	FLXn EDBL	Error detected on channel B interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
18	FLXn TABAL	Transmission across boundary channel A interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
17	FLXn LTVAL	Latest transmit violation channel A interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
16	FLXn EDAL	Error detected on channel A interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
11	FLXn MHFL	Message handler constraints flag interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
10	FLXn IOBAL	Illegal output buffer access interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
9	FLXn IIBAL	Illegal input buffer access interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0

Table 24-23 FLXnEILS Register Contents (2/2)

Bit Position	Bit Name	Function
8	FLXn EFAL	Empty FIFO access interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
7	FLXn RFOL	Receive FIFO overrun interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
5	FLXn CCLL	CHI command locked interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
4	FLXn CCFL	Clock correction failure interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
3	FLXn SFOL	Sync frame overflow interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
2	FLXn SFBML	Sync frames below minimum interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
1	FLXn CNAL	Command not accepted interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
0	FLXn PEMCL	POC error mode changed interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0

(4) FLXnSILS – Status Interrupt Line Select Register

The settings in the status interrupt line select register assign an interrupt generated by a specific status interrupt flag to one of the two interrupt lines (eray_int0 or eray_int1).

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $002C_H$

Initial value $0303\ FFFF_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	FLXn MTSBL	FLXn WUPBL	0	0	0	0	0	0	FLXn MTSAL	FLXn WUPAL
R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXn SDSL	FLXn MBSIL	FLXn SUCSL	FLXn SWEL	FLXn TOBCL	FLXn TIBCL	FLXn TI1L	FLXn TI0L	FLXn NMVCL	FLXn RFCLL	FLXn RFNEL	FLXn RXIL	FLXn TXIL	FLXn CYCSL	FLXn CASL	FLXn WSTL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-24 FLXnSILS Register Contents (1/2)

Bit Position	Bit Name	Function
25	FLXnMTSBL	Media access test symbol channel B interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
24	FLXnWUPBL	Wakeup pattern channel B interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
17	FLXnMTSAL	Media access test symbol channel A interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
16	FLXnWUPAL	Wakeup pattern channel A interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
15	FLXnSDSL	Start of dynamic segment interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
14	FLXnMBSIL	Message buffer status interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
13	FLXnSUCSL	Startup completed successfully interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
12	FLXnSWEL	Stop watch event interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
11	FLXnTOBCL	Transfer output buffer completed interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
10	FLXnTIBCL	Transfer input buffer completed interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0

Table 24-24 FLXnSILS Register Contents (2/2)

Bit Position	Bit Name	Function
9	FLXnTI1L	Timer interrupt 1 line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
8	FLXnTI0L	Timer interrupt 0 line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
7	FLXnNMVCL	Network management vector changed interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
6	FLXnRFCLL	Receive FIFO critical level interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
5	FLXnRFNEL	Receive FIFO not empty interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
4	FLXnRXIL	Receive interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
3	FLXnTXIL	Transmit interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
2	FLXnCYCSL	Cycle start interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
1	FLXnCASL	Collision avoidance symbol interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0
0	FLXnWSTL	Wakeup status interrupt line 1 = Interrupt assigned to interrupt line eray_int1 0 = Interrupt assigned to interrupt line eray_int0

(5) FLXnEIES/FLXnEIER – Error Interrupt Enable Set/Reset Register

The settings in the error interrupt enable register determine which status changes in the error interrupt register will result in an interrupt.

The enable bits are set by writing to FLXnEIES and reset by writing to FLXnEIER. Writing a 1 sets/resets the specific enable bit, writing a 0 has no effect. Reading from both addresses will result in the same value.

Access This register can be read/written in 32-bit units.

Address FLXnEIES: <FLXn_base> + 0030_H
FLXnEIER: <FLXn_base> + 0034_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnEIES

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnSETABBE	FLXnSELTVBE	FLXnSEEDBE	0	0	0	0	0	FLXnSETABAE	FLXnSELTVAE	FLXnSEEDAE
R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FLXnSEMHFE	FLXnSEIOBAE	FLXnSEIIBAE	FLXnSEEFAE	FLXnSERFOE	0	FLXnSECCLE	FLXnSECCFE	FLXnSESFOE	FLXnSESFBME	FLXnSECNAE	FLXnSEPEMCE
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

(b) FLXnEIER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnCLTABBE	FLXnCLLTVBE	FLXnCLEDBE	0	0	0	0	0	FLXnCLTABAE	FLXnCLLTVAE	FLXnCLEDAE
R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FLXnCLMHFE	FLXnCLIOBAE	FLXnCLIIBAE	FLXnCLEFAE	FLXnCLRFOE	0	FLXnCLCCLE	FLXnCLCCFE	FLXnCLSFOE	FLXnCLSFBME	FLXnCLCNAE	FLXnCLPEMCE
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

All bits of these registers behave as follows:

- Write FLXnEIES.FLXnSE... = 1: Enables interrupts
- Write FLXnEIER.FLXnCL... = 1: Disables interrupts
- Read FLXnEIES.FLXnSE... or FLXnEIER.FLXnCL... = 1: Indicates that interrupts enabled
- Read FLXnEIES.FLXnSE... or FLXnEIER.FLXnCL... = 0: Indicates that interrupts disabled

Table 24-25 FLXnEIES/FLXnEIER Register Contents

Bit Position	Bit Name	Function
26	FLXnSETABBE FLXnCLTABBE	Transmission across boundary channel B interrupt
25	FLXnSELTVBE FLXnCLLTVBE	Latest transmit violation channel B interrupt
24	FLXnSEEDBE FLXnCLEDBE	Error detected on channel B interrupt
18	FLXnSETABAE FLXnCLTABAE	Transmission across boundary channel A interrupt
17	FLXnSELTVAE FLXnCLLTVAE	Latest transmit violation channel A interrupt
16	FLXnSEEDAE FLXnCLEDAE	Error detected on channel A interrupt
11	FLXnSEMHFE FLXnCLMHFE	Message handler constraints flag interrupt
10	FLXnSEIOBAE FLXnCLIOBAE	Illegal output buffer access interrupt
9	FLXnSEIIBAE FLXnCLIIBAE	Illegal input buffer access interrupt
8	FLXnSEEFAE FLXnCLEFAE	Empty FIFO access interrupt
7	FLXnSERFOE FLXnCLRFOE	Receive FIFO overrun interrupt
5	FLXnSECCLE FLXnCLCCLE	CHI command locked interrupt
4	FLXnSECCFE FLXnCLCCFE	Clock correction failure interrupt
3	FLXnSESFOE FLXnCLSFOE	Sync frame overflow interrupt
2	FLXnSESFBME FLXnCLSFBME	Sync frames below minimum interrupt
1	FLXnSECNAE FLXnCLCNAE	Command not accepted interrupt
0	FLXnSEPEMCE FLXnCLPEMCE	POC error mode changed interrupt

Note Among the functions in table 24-25, FLXnEIES/FLXnEIER Register Contents, those for which interrupt can be enabled are the ones for which an interrupt controller is specified in the "Connected to" column of table 24-5, FLXn Interrupts. Enabling interrupt for other functions does not take effect.

(6) FLXnSIES/FLXnSIER – Status Interrupt Enable Set/Reset Register

The settings in the status interrupt enable register determine which status changes in the status interrupt register will result in an interrupt.

The enable bits are set by writing to FLXnSIES and reset by writing to FLXnSIER. Writing a 1 sets/resets the specific enable bit, writing a 0 has no effect. Reading from both addresses will result in the same value.

Access This register can be read/written in 32-bit units.

Address FLXnSIES: <FLXn_base> + 0038_H
FLXnSIER: <FLXn_base> + $003C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnSIES

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	FLXnSE MTSBE	FLXnSE WUPBE	0	0	0	0	0	0	FLXnSE MTSAE	FLXnSE WUPAE
R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnSE SDSE	FLXnSE MBSIE	FLXnSE SUCSE	FLXnSE SWEE	FLXnSE TOBCE	FLXnSE TIBCE	FLXnSE TI1E	FLXnSE TI0E	FLXnSE NMVCE	FLXnSE RFCLE	FLXnSE RFNEE	FLXnSE RXIE	FLXnSE TXIE	FLXnSE CYCSE	FLXnSE CASE	FLXnSE WSTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(b) FLXnSIER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	FLXnCL MTSBE	FLXnCL WUPBE	0	0	0	0	0	0	FLXnCL MTSAE	FLXnCL WUPAE
R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnCL SDSE	FLXnCL MBSIE	FLXnCL SUCSE	FLXnCL SWEE	FLXnCL TOBCE	FLXnCL TIBCE	FLXnCL TI1E	FLXnCL TI0E	FLXnCL NMVCE	FLXnCL RFCLE	FLXnCL RFNEE	FLXnCL RXIE	FLXnCL TXIE	FLXnCL CYCSE	FLXnCL CASE	FLXnCL WSTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

All bits of these registers behave as follows:

- Write FLXnSIES.FLXnSE... = 1: Enables interrupts
- Write FLXnSIER.FLXnCL... = 1: Disables interrupts
- Read FLXnSIES.FLXnSE... or FLXnSIER.FLXnCL... = 1: Indicates that interrupts enabled
- Read FLXnSIES.FLXnSE... or FLXnSIER.FLXnCL... = 0: Indicates that interrupts disabled

Table 24-26 FLXnSIES/FLXnSIER Register Contents

Bit Position	Bit Name	Function
25	FLXnSEMTSBE FLXnCLMTSBE	MTS received on channel B interrupt
24	FLXnSEWUPBE FLXnCLWUPBE	Wakeup pattern channel B interrupt
17	FLXnSEMTSAE FLXnCLMTSAE	MTS received on channel A interrupt
16	FLXnSEWUPAE FLXnCLWUPAE	Wakeup pattern channel A interrupt
15	FLXnSESDSE FLXnCLSDSE	Start of dynamic segment interrupt
14	FLXnSEMBSIE FLXnCLMBSIE	Message buffer status interrupt
13	FLXnSESUCSE FLXnCLSUCSE	Startup completed successfully interrupt
12	FLXnSESWEE FLXnCLSWEE	Stop watch event interrupt
11	FLXnSETOBCE FLXnCLTOBCE	Transfer output buffer completed interrupt
10	FLXnSETIBCE FLXnCLTIBCE	Transfer input buffer completed interrupt
9	FLXnSETI1E FLXnCLTI1E	Timer interrupt 1
8	FLXnSETI0E FLXnCLTI0E	Timer interrupt 0
7	FLXnSENMVCE FLXnCLNMVCE	Network management vector changed interrupt
6	FLXnSERFCLE FLXnCLRFCLE	Receive FIFO critical level interrupt
5	FLXnSERFNEE FLXnCLRFNEE	Receive FIFO not empty interrupt
4	FLXnSERXIE FLXnCLRXIE	Receive interrupt
3	FLXnSETXIE FLXnCLTXIE	Transmit interrupt
2	FLXnSECYCSE FLXnCLCYCSE	Cycle start interrupt
1	FLXnSECASE FLXnCLCASE	Collision avoidance symbol interrupt
0	FLXnSEWSTE FLXnCLWSTE	Wakeup status interrupt

Note Among the functions in table 24-26, FLXnSIES/FLXnSIER Register Contents, those for which interrupt can be enabled are the ones for which an interrupt controller is specified in the "Connected to" column of table 24-5, FLXn Interrupts and DMA Requests. Enabling interrupt for other functions does not take effect.

(7) FLXnILE – Interrupt Line Enable

Each of the two interrupt lines to the host CPU (eray_int0, eray_int1) can be enabled/disabled separately by setting bit FLXnEINTL0 and FLXnEINTL1.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0040_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn EINTL1	FLXn EINTL0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24-27 FLXnILE Register Contents

Bit Position	Bit Name	Function
1	FLXnEINTL1	Enable interrupt line 1 1 = Interrupt line eray_int1 enabled 0 = Interrupt line eray_int1 disabled
0	FLXnEINTL0	Enable interrupt line 0 1 = Interrupt line eray_int0 enabled 0 = Interrupt line eray_int0 disabled

(8) FLXnT0C – Timer 0 Configuration Register

This register sets the absolute timer. It specifies the point in time when the timer 0 interrupt occurs in units of cycle counts and macroticks. When the timer 0 interrupt is asserted, output signal eray_tint0 is set to 1 for the duration of one macrotick and FLXnSIR.FLXnTI0F is set to 1.

Timer 0 can be activated as long as the POC is either in the NORMAL_ACTIVE state or in the NORMAL_PASSIVE state. Timer 0 is deactivated when leaving the NORMAL_ACTIVE state or the NORMAL_PASSIVE state. Timer 0 continues to be active during transition between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXnT0RC to 0.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0044_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnT0MO[13:00]													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FLXnT0CC[6:0]							0	0	0	0	0	0	FLXnT0MS	FLXnT0RC
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24-28 FLXnT0C Register Contents

Bit Position	Bit Name	Function
29 to 16	FLXnT0MO [13:00]	Timer 0 macrotick offset Configures the macrotick offset from the beginning of the cycle where the interrupt is to occur. The timer 0 interrupt occurs at this offset for each cycle in the cycle set.
14 to 8	FLXnT0CC [6:0]	Timer 0 cycle code The 7-bit timer 0 cycle code determines the cycle set used for generation of the timer 0 interrupt. For details about the configuration of the cycle code see section 24.5.7 (2), Cycle Counter Filtering.
1	FLXnT0MS	Timer 0 mode select 1 = Continuous mode 0 = Single-shot mode
0	FLXnT0RC	Timer 0 run control 1 = Timer 0 running 0 = Timer 0 halted

Note The configuration of timer 0 is compared against the macrotick counter value. There is no separate counter for timer 0. When the FlexRay controller leaves the NORMAL_ACTIVE or NORMAL_PASSIVE state, or timer 0 is halted, output signal eray_tint0 is reset to 0 immediately.

(9) FLXnT1C – Timer 1 Configuration Register

This register sets the relative timer. After the specified number of macroticks has expired, the timer 1 interrupt is asserted, output signal eray_tint1 is set to 1 for the duration of one macrotick, and FLXnSIR.FLXnTI1F is set to 1.

Timer 1 can be activated as long as the POC is either in the NORMAL_ACTIVE state or in the NORMAL_PASSIVE state. Timer 1 is deactivated when leaving the NORMAL_ACTIVE state or the NORMAL_PASSIVE state. Timer 1 continues to be active during transition between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXnT1RC to 0.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0048_H

Initial value $0002\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnT1MC[13:00]													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXnT1MS	FLXnT1RC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24-29 FLXnT1C Register Contents

Bit Position	Bit Name	Function
29 to 16	FLXnT1MC [13:00]	Timer 1 macrotick count When the configured macrotick count is reached the timer 1 interrupt is generated. Valid values are: 2 to 16383 MT in continuous mode 1 to 16383 MT in single-shot mode
1	FLXnT1MS	Timer 1 mode select 1 = Continuous mode 0 = Single-shot mode
0	FLXnT1RC	Timer 1 run control 1 = Timer 1 running 0 = Timer 1 halted

Note When the FlexRay controller leaves the NORMAL_ACTIVE or NORMAL_PASSIVE state, or timer 1 is halted, output signal eray_tint1 is reset to 0 immediately.

(10) FLXnSTPW1 – Stop Watch Register 1

The stop watch is activated by an interrupt 0, 1 event (detection of rising edge on signal eray_int0 or eray_int1) or by the host by writing bit FLXnSSWT to 1. When the macrotick counter is incremented following activation of the stop watch, the actual cycle counter and macrotick values are captured in register FLXnSTPW1 while the slot counter values for channels A and B are captured in register FLXnSTPW2.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $004C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnSMTV[13:00]													
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnSCCV[5:0]						0	FLXn EINT1	FLXn EINT0	0	FLXn SSWT	0	FLXn SWMS	FLXn ESWT
R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W

Table 24-30 FLXnSTPW1 Register Contents (1/2)

Bit Position	Bit Name	Function
29 to 16	FLXnSMTV [13:00]	Stopped macrotick value State of the macrotick counter when the stop watch event occurred. Valid values are 0 to 15,999.
13 to 8	FLXnSCCV [5:0]	Stopped cycle counter value State of the cycle counter when the stop watch event occurred. Valid values are 0 to 63.
6	FLXnEINT1	Enable interrupt 1 (eray_int1) trigger Enables stop watch trigger by interrupt 1 event if FLXnESWT = 1. 1 = Interrupt 1 event triggers stop watch 0 = Stop watch trigger by interrupt 1 disable
5	FLXnEINT0	Enable interrupt 0 (eray_int0) trigger Enables stop watch trigger by interrupt 0 event if FLXnESWT = 1. 1 = Interrupt 0 event triggers stop watch 0 = Stop watch trigger by interrupt 0 disable
3	FLXn SSWT	Software stop watch trigger When the host writes this bit to 1, the stop watch is activated. After the actual cycle counter and macrotick value are stored in the stop watch register, this bit is reset to 0. The bit is writeable only when FLXnESWT = 0. 1 = Stop watch activated by software trigger 0 = Software trigger reset
1	FLXn SWMS	Stop watch mode select 1 = Continuous mode 0 = Single-shot mode

Table 24-30 FLXnSTPW1 Register Contents (2/2)

Bit Position	Bit Name	Function
0	FLXn ESWT	Stop watch trigger If this bit is enabled, an interrupt 0, 1 event (detection of rising edge on signal eray_int0 or eray_int1) activates the stop watch. In single-shot mode, this bit is reset to 0 after the actual cycle counter and macrotick value are stored in the stop watch register. 1 = Stop watch trigger enabled 0 = Stop watch trigger disabled

Note Bits FLXnESWT and FLXnSSWT cannot be set to 1 simultaneously. In this case, the write is ignored, and both bits keep their previous values.

(11) FLXnSTPW2 – Stop Watch Register 2

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0050_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnSSCVB[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnSSCVA[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-31 FLXnSTPW2 Register Contents

Bit Position	Bit Name	Function
26 to 16	FLXnSSCVB [10:00]	Stop watch captured slot counter value channel B State of the slot counter for channel B when the stop watch event occurred. Valid values are 0 to 2047.
10 to 0	FLXnSSCVA [10:00]	Stop watch captured slot counter value channel A State of the slot counter for channel A when the stop watch event occurred. Valid values are 0 to 2047.

24.4.5 CC Control Registers

The CC control registers are the registers for controlling the communication operation. The FlexRay protocol specification permits the host to write application configuration data only in the CONFIG state. However, note that the configuration registers are not locked for writing in the DEFAULT_CONFIG state.

The configuration data is reset when the DEFAULT_CONFIG state is entered by hard reset. To change the POC state from the DEFAULT_CONFIG state to the CONFIG state, the host has to apply CHI command CONFIG. To leave the CONFIG state, the host has to perform the steps described in the section for the lock register (LCK).

All bits marked with an asterisk * can be updated in the DEFAULT_CONFIG or CONFIG state only.

(1) FLXnSUCC1 – SUC Configuration Register 1

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0080_H

Initial value $0C40\ 1080_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	FLXn CCHB*	FLXn CCHA*	FLXn MTSB*	FLXn MTSA*	FLXn HCSE*	FLXn TSM*	FLXn WUCS*	FLXnPTA[4:0]*				
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnCSA[4:0]*					0	FLXn TXSY*	FLXn TXST*	FLXn PBSY	0	0	0	FLXnCMD[3:0]			
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24-32 FLXnSUCC1 Register Contents (1/3)

Bit Position	Bit Name	Function
27	FLXnCCHB	Connected to channel B (pChannels) Configures whether the node is connected to channel B. 1 = Node connected to channel B (default by hardware reset) 0 = Not connected to channel B
26	FLXnCCHA	Connected to channel A (pChannels) Configures whether the node is connected to channel A. 1 = Node connected to channel A (default by hardware reset) 0 = Not connected to channel A

Table 24-32 FLXnSUCC1 Register Contents (2/3)

Bit Position	Bit Name	Function
25	FLXnMTSB	Select channel B for MTS transmission This bit selects channel B for MTS symbol transmission. The flag is reset by default and can be modified only in the DEFAULT_CONFIG or CONFIG state. 1 = Channel B selected for MTS transmission 0 = Channel B disabled for MTS transmission Note: FLXnMTSA and FLXnMTSB may also be changed outside the DEFAULT_CONFIG or CONFIG state when the write to the FLXnSUCC1 register is directly preceded by the unlock sequence for the configuration lock key as described in section 24.4.3 (3), FLXnLCK – Lock Register. If both bits FLXnMTSA and FLXnMTSB are set to 1, an MTS symbol will be transmitted on both channels when requested by writing FLXnCMD[3:0] = 1000.
24	FLXnMTSA	Select channel A for MTS transmission The bit selects channel A for MTS symbol transmission. The flag is reset by default and can be modified only in the DEFAULT_CONFIG or CONFIG state. 1 = Channel A selected for MTS transmission 0 = Channel A disabled for MTS transmission
23	FLXnHCSE	Halt due to clock sync error (pAllowHaltDueToClock) This bit enables transition to the HALT state when a clock synchronization error occurs. The bit can be modified in the DEFAULT_CONFIG or CONFIG only. 1 = CC will enter HALT state 0 = CC will enter/remain in NORMAL_PASSIVE
22	FLXnTSM	Transmission slot mode (pSingleSlotEnabled) Selects the initial transmission slot mode. In SINGLE slot mode, the CC can transmit frames only in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0, or depending on bit FLXnMRC.FLXnSPLM, in the header sections of both message buffers 0 and 1. In the case when FLXnTSM = 1, message buffer 0 or message buffers 0, 1 can be (re)configured in the DEFAULT_CONFIG or CONFIG state only. In ALL slot mode, the CC can transmit frames in all slots. FLXnTSM is a configuration bit which can be set/reset only by the host. The bit can be written in the DEFAULT_CONFIG or CONFIG state only. The FlexRay controller changes to ALL slot mode when the host successfully applied the ALL_SLOTS command by writing FLXnCMD[3:0] = 0101 in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by FLXnCCSV.FLXnSLM[1:0]. 1 = SINGLE slot mode (default by hardware reset) 0 = ALL slot mode
21	FLXnWUCS	Wakeup channel select (pWakeupChannel) This bit selects the channel on which the CC sends the wakeup pattern. The CC ignores any attempt to change the state of this bit when it is not in the DEFAULT_CONFIG or CONFIG state. 1 = Send wakeup pattern on channel B 0 = Send wakeup pattern on channel A
20 to 16	FLXnPTA[4:0]	Passive to active (pAllowPassiveToActive) Defines the number of consecutive even/odd cycle pairs that allows the FlexRay controller to transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. If it is set to 00000, the FlexRay controller is not allowed to transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. It can be modified in the DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 31 even/odd cycle pairs.

Table 24-32 FLXnSUCC1 Register Contents (3/3)

Bit Position	Bit Name	Function
15 to 11	FLXnCSA[4:0]	Cold start attempts (gColdStartAttempts) Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node. It can be modified in the DEFAULT_CONFIG or CONFIG state only. This value must be identical in all nodes of a cluster. Valid values are 2 to 31.
9	FLXnTXSY	Transmit sync frame in key slot (pKeySlotUsedForSync) Defines whether the key slot is used to transmit sync frames. The bit can be modified in the DEFAULT_CONFIG or CONFIG state only. 1 = Key slot used to transmit sync frame, node is sync node 0 = No sync frame transmission in key slot, node is neither sync nor coldstart node
8	FLXnTXST	Transmit startup frame in key slot (pKeySlotUsedForStartup) Defines whether the key slot is used to transmit startup frames. The bit can be modified in the DEFAULT_CONFIG or CONFIG state only. 1 = Key slot used to transmit startup frame, node is leading or following coldstarter node 0 = No startup frame transmission in key slot, node is non-coldstarter
7	FLXnPBSY	POC busy Signals that the POC is busy and cannot accept a command from the host. FLXnCMD[3:0] is locked against write accesses. This bit is set to 1 after hard reset or during initialization of internal RAM blocks by the execution of CLEAR_RAMS command. This bit is cleared to 0 at the end of initialization in 2048 eray_bclk cycles. 1 = POC is busy, FLXnCMD[3:0] locked 0 = POC not busy, FLXnCMD[3:0] writeable
3 to 0	FLXnCMD[3:0]	CHI command vector The host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted), and flag FLXnEIR.FLXnCNAE will be set to 1. In the case when the previous CHI command has not yet completed, FLXnEIR.FLXnCCLE is set to 1 together with FLXnEIR.FLXnCNAE; the CHI command needs to be repeated. Except for the HALT state, a POC state change command executed when the FlexRay controller is already in the requested POC state does not cause the state to be changed or FLXnEIR.FLXnCNAE to be set. For details, refer to table 24-33, FLXnSUCC1.FLXnCMD[3:0] Function.

Reading FLXnCMD[3:0] shows whether the last CHI command was accepted. The actual POC state is monitored by FLXnCCSV.FLXnPOCS[5:0]. In general, the host must check FLXnSUCC1.FLXnPBSY before writing a new CHI command.

Table 24-33 FLXnSUCC1.FLXnCMD[3:0] Function (1/2)

FLXn CMD[3:0]	Function
0000	command_not_accepted FLXnCMD[3:0] is reset to 0000 due to one of the following conditions: • Illegal command applied by the host • Command applied to leave the CONFIG state without performing an unlock sequence • Host applied new command while execution of the previous host command has not completed. • Host writes command_not_accepted (0000) When FLXnCMD[3:0] is reset to 0000, FLXnEIR.FLXnCNAE is set to 1, and - if enabled - an interrupt is generated. Commands which are not accepted are not executed.
0001	CONFIG The state becomes CONFIG when this command is applied in POC states DEFAULT_CONFIG, READY, or in MONITOR_MODE. When called in HALT state, the CC transits to POC state DEFAULT_CONFIG. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0010	READY The state becomes READY when this command is applied in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0011	WAKEUP The state becomes WAKEUP when this command is applied in POC state READY. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0100	RUN The state becomes STARTUP when this command is applied in POC state READY. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0101	ALL_SLOTS If this command is applied in POC states NORMAL_ACTIVE or NORMAL_PASSIVE, transition from single slot mode to all slot mode occurs after successful startup/integration at the end of the next cycle. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0110	HALT If this command is called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE, HALT request FLXnCCSV.FLXnHRQ is set to 1 and transition to POC state HALT occurs at the end of the next cycle. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
0111	FREEZE If this command is called, the freeze status indicator CCSV.FLXnFSI is set to 1 and transition to POC state HALT occurs immediately. This command can be called in any state.
1000	SEND_MTS An MTS symbol is sent in the next symbol window on the channel configured by FLXnMTSA and FLXnMTSB if this command is called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (FLXnCCSV.FLXnSLM[1:0] = 11). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).
1001	ALLOW_COLDSTART The command resets FLXnCCSV.FLXnCSI to enable the node to become leading coldstarter. When called in states DEFAULT_CONFIG, CONFIG, HALT, or MONITOR_MODE, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted). To become leading coldstarter, it is also required that both FLXnTXST and FLXnTXSY are set.
1010	RESET_STATUS_INDICATORS Resets status flags FLXnCCSV.FLXnCSNI, FLXnCCSV.FLXnCSAI, and FLXnCCSV.FLXnWSV[2:0] to their default values. This command may be called in POC states READY and STARTUP. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted).

Table 24-33 FLXnSUCC1.FLXnCMD[3:0] Function (2/2)

FLXn CMD[3:0]	Function
1011	MONITOR_MODE Transition to MONITOR_MODE occurs when this command is called in POC state CONFIG. In this mode, the CC is able to receive FlexRay frames and wakeup pattern. It is also able to detect coding errors. The temporal integrity of received frames is not checked. This mode can be used for debugging purposes, in such cases that the startup of a FlexRay network fails. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted). For details, see section 24.5.5 (4), MONITOR_MODE State.
1100	CLEAR_RAMS Sets FLXnMHDS.FLXnCRAM to 1 when called in the DEFAULT_CONFIG or CONFIG state. When called in any other state, FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted). FLXnMHDS.FLXnCRAM is also set to 1 when the FlexRay controller is reset. By setting FLXnMHDS.FLXnCRAM, all internal RAM blocks are initialized to zero. During the initialization of the RAMs, FLXnPBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS. The initialization of internal RAM blocks of the FlexRay controller requires 2048 eray_bclk cycles. There should be no host access to IBF or OBF during initialization of the internal RAM blocks after hard reset or after assertion of CHI command CLEAR_RAMS. Before asserting CHI command CLEAR_RAMS, the host should make sure that no transfer between message RAM and IBF/OBF or the transient buffer RAMs is ongoing. This command also resets the message buffer status registers FLXnMHDS, FLXnLDTS, FLXnFSR, FLXnMHDF, FLXnTXRQ1/2/3/4, FLXnNDAT1/2/3/4, and FLXnMBSC1/2/3/4.
1101 1110 1111	Reserved CHI commands Used in the test modes described in section 24.4.3, Special Registers.

Note All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the eray_sclk domain after at most 8 cycles of the slower of the two clocks eray_bclk and eray_sclk, provided that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXnCCSV will show data that is additionally delayed by synchronization from eray_sclk to eray_bclk domain and by the host-specific CPU interface. The maximum additional delay is 12 cycles of the slower of the two clocks eray_bclk and eray_sclk.

table 24-34, Reference to CHI Host Command Summary from FlexRay Protocol Specification, below references the CHI commands from the FlexRay Protocol Specification V2.1 (section 2.2.1.1, Table 2-2) to the E-Ray CHI command vector FLXnCMD[3:0].

Table 24-34 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI Command	Where Processed (POC States)	CHI Command Vector FLXnCMD[3:0]
ALL_SLOTS	POC: normal active POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config POC: config POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active POC: normal passive	HALT
READY	All except POC: default config, POC: config	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

(2) FLXnSUCC2 – SUC Configuration Register 2

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0084_H

Initial value $0100\ 0504_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	FLXnLTN[3:0]*				0	0	0	FLXnLT[20:16]*				
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnLT[15:00]*															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-35 FLXnSUCC2 Register Contents

Bit Position	Bit Name	Function
27 to 24	FLXnLTN [3:0]	Listen timeout noise (gListenNoise - 1) Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout. The range for gListenNoise is 2 to 16. FLXnLTN[3:0] must be configured identical in all nodes of a cluster.
20 to 0	FLXnLT [20:00]	Listen timeout (pdListenTimeout) Configures wakeup/startup listen timeout in μT. The range for pdListenTimeout is 1284 to 1283846 μT.

Note The wakeup/startup noise timeout is calculated as follows:
pdListenTimeout • gListenNoise = FLXnLT[20:00] • (FLXnLTN[3:0] + 1)

(3) FLXnSUCC3 – SUC Configuration Register 3

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0088_H

Initial value $0000\ 0011_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FLXnWCF[3:0]*				FLXnWCP[3:0]*			
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-36 FLXnSUCC3 Register Contents

Bit Position	Bit Name	Function
7 to 4	FLXnWCF [3:0]	Maximum without clock correction fatal (gMaxWithoutClockCorrectionFatal) These bits define how many consecutive even/odd cycle pairs the clock correction is to be suspended before causing a transition from the NORMAL_ACTIVE or NORMAL_PASSIVE state to the HALT state. These bits must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.
3 to 0	FLXnWCP [3:0]	Maximum without clock correction passive (gMaxWithoutClockCorrectionPassive) These bits define how many consecutive even/odd cycle pairs the clock correction is to be suspended before causing a transition from the NORMAL_ACTIVE state to the NORMAL_PASSIVE state. These bits must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.

Note The transition to the HALT state is prevented if FLXnSUCC1.FLXnHCSE is not set to 1.

(4) FLXnNEMC – NEM Configuration Register

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $008C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	FLXnNML[3:0]*			
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 24-37 FLXnNEMC Register Contents

Bit Position	Bit Name	Function
3 to 0	FLXnNML [3:0]	Network management vector length (gNetworkManagementVectorLength) These bits configure the length of the NM vector. The configured length must be identical in all nodes of a cluster. Valid values are 0 to 12 bytes.

(5) FLXnPRTC1 – PRT Configuration Register 1

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0090_H

Initial value $084C\ 0633_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnRWP[5:0]*						0	FLXnRXW[8:0]*								
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnBRP [1:0]*		FLXnSPP [1:0]*		0	FLXnCASM[6:0]*							FLXnTSST[3:0]*			
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-38 FLXnPRTC1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	FLXnRWP [5:0]	Repetitions of Tx wakeup pattern (pWakeupPattern) Configures the number of repetitions (sequences) of the Tx wakeup symbol. Valid values are 2 to 63.
24 to 16	FLXnRXW [8:0]	Wakeup symbol receive window length (gdWakeupSymbolRxWindow) Configures the number of bit times used by the node to test the duration of the received wakeup pattern. Must be identical in all nodes of a cluster. Valid values are 76 to 301 bit times.
15, 14	FLXnBRP [1:0]	Baud rate prescaler (gdSampleClockPeriod, pSamplePerMicrotick) The baud rate prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock eray_sclk = 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate. 00 = 10 Mbit/s (default) gdSampleClockPeriod = 12.5 ns = 1 eray_sclk pSamplesPerMicrotick = 2 (1 µT = 25 ns) 01 = 5 Mbit/s gdSampleClockPeriod = 25 ns = 2 eray_sclk pSamplesPerMicrotick = 1 (1 µT = 25 ns) 10, 11 = 2.5 Mbit/s gdSampleClockPeriod = 50 ns = 4 eray_sclk pSamplesPerMicrotick = 1 (1 µT = 50 ns)
13, 12	FLXnSPP [1:0]	Strobe point position Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by FLXnSPP[1:0]. 00, 11= Sample 5 (default) 01 = Sample 4 10 = Sample 6 Note: The current revision 2.1 of the FlexRay protocol requires that FLXnSPP[1:0] = 00. The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.
10 to 4	FLXnCASM [6:0]	Collision avoidance symbol max (gdCASRxLowMax) Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS). FLXnCASM06 is fixed to 1. Valid values are 67 to 99 bit times.

Table 24-38 FLXnPRTC1 Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 0	FLXnTSST [3:0]	Transmission start sequence transmitter (gdTSSTransmitter) Configures the duration of the transmission start sequence (TSS) in terms of bit times (1 bit time = 4 µT = 100ns@10Mbps). Must be identical in all nodes of a cluster. Valid values are 3 to 15 bit times.

(6) FLXnPRTC2 – PRT Configuration Register 2

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0094_H

Initial value $0F2D\ 0A0E_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnTXL[5:0]*						FLXnTXI[7:0]*							
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnRXL[5:0]*						0	0	FLXnRXI[5:0]*					
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-39 FLXnPRTC2 Register Contents

Bit Position	Bit Name	Function
29 to 24	FLXnTXL[5:0]	Wakeup symbol transmit low (gdWakeupSymbolTxLow) Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 15 to 60 bit times.
23 to 16	FLXnTXI[7:0]	Wakeup symbol transmit idle (gdWakeupSymbolTxIdle) Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 45 to 180 bit times.
13 to 8	FLXnRXL [5:0]	Wakeup symbol receive low (gdWakeupSymbolRxLow) Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 10 to 55 bit times.
5 to 0	FLXnRXI[5:0]	Wakeup symbol receive idle (gdWakeupSymbolRxIdle) Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 14 to 59 bit times.

(7) FLXnMHDC – MHD Configuration Register

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0098_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	FLXnSLT[12:00]*												
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	FLXnSFDL[6:0]*						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-40 FLXnMHDC Register Contents

Bit Position	Bit Name	Function
28 to 16	FLXnSLT [12:00]	Start of latest transmit (pLatestTx) Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXnSLT[12:00] is set to zero. Valid values are 0 to 7981 minislots.
6 to 0	FLXnSFDL [6:0]	Static frame data length (gPayloadLengthStatic) Configures the cluster-wide frame length for all frames sent in the static segment in double bytes. The frame length must be identical in all nodes of a cluster. Valid values are 0 to 127.

(8) FLXnGTUC01 – GTU Configuration Register 1

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00A0_H$

Initial value $0000\ 0280_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	FLXnUT[19:16]*			
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnUT[15:00]*															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-41 FLXnGTUC01 Register Contents

Bit Position	Bit Name	Function
19 to 0	FLXnUT [19:00]	Microtick per cycle (pMicroPerCycle) Configures the duration of the communication cycle in microticks. Valid values are 640 to 640000 µT.

(9) FLXnGTUC02 – GTU Configuration Register 2

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00A4_H$

Initial value $0002\ 000A_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	FLXnSNM[3:0]*			
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnMPC[13:00]*													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-42 FLXnGTUC02 Register Contents

Bit Position	Bit Name	Function
19 to 16	FLXnSNM [3:0]	Sync node max (gSyncNodeMax) Maximum number of frames within a cluster with sync frame indicator bit FLXnRHSYN set to 1. Must be identical in all nodes of a cluster. Valid values are 2 to 15.
13 to 0	FLXnMPC [13:00]	Macrotick per cycle (gMacroPerCycle) Configures the duration of one communication cycle in macroticks. The cycle length must be identical in all nodes of a cluster. Valid values are 10 to 16000 MT.

(10) FLXnGTUC03 – GTU Configuration Register 3

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00A8_H$

Initial value $0202\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	FLXnMIOB[6:0]*							0	FLXnMIOA[6:0]*						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnUIOB[7:0]*								FLXnUIOA[7:0]*							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-43 FLXnGTUC03 Register Contents

Bit Position	Bit Name	Function
30 to 24	FLXnMIOB [6:0]	Macrotick initial offset channel B (pMacroInitialOffset[B]) Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.
22 to 16	FLXnMIOA [6:0]	Macrotick initial offset channel A (pMacroInitialOffset[A]) Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.
15 to 8	FLXnUIOB [7:0]	Microtick initial offset channel B (pMicroInitialOffset[B]) Configures the number of microticks which describe the distance between the macrotick boundary described by gMacroInitialOffset and the exact secondary time reference point. The parameter depends on pDelayCompensation[B] and therefore has to be set for each channel independently. Valid values are 0 to 240 µT.
7 to 0	FLXnUIOA [7:0]	Microtick initial offset channel A (pMicroInitialOffset[A]) Configures the number of microticks which describe the distance between the macrotick boundary described by gMacroInitialOffset and the exact secondary time reference point. The parameter depends on pDelayCompensation[A] and therefore has to be set for each channel independently. Valid values are 0 to 240 µT.

(11) FLXnGTUC04 – GTU Configuration Register 4

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only. For details about configuration of FLXnNIT[13:00] and FLXnOCS[13:00], see section 24.5.1 (5), Configuration of NIT Start and Offset Correction Start.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00AC_H$

Initial value $0008\ 0007_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnOCS[13:00]*													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnNIT[13:00]*													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-44 FLXnGTUC04 Register Contents

Bit Position	Bit Name	Function
29 to 16	FLXnOCS [13:00]	Offset correction start (gOffsetCorrectionStart - 1) Determines the start of the offset correction within the NIT phase, in terms of macroticks (MT) from the beginning of the cycle. Must be identical in all nodes of a cluster. For cluster consisting of E-Ray implementations only, it is sufficient to program FLXnOCS = FLXnNIT + 1. Valid values are 8 to 15998 MT.
13 to 0	FLXnNIT [13:00]	Network idle time start (gMacroPerCycle - gdNIT - 1) Configures the starting point of the network idle time NIT in terms of macroticks (MT) from the beginning of the cycle. The start of NIT is recognized if macrotick = gMacroPerCycle - gdNIT -1 and the increment pulse of macrotick is set. Must be identical in all nodes of a cluster. Valid values are 7 to 15997 MT.

(12) FLXnGTUC05 – GTU Configuration Register 5

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00B0_H$

Initial value $00E00\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnDEC[7:0]*								0	0	0	FLXnCDD[4:0]*				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnDCB[7:0]*								FLXnDCA[7:0]*							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-45 FLXnGTUC05 Register Contents

Bit Position	Bit Name	Function
31 to 24	FLXnDEC [7:0]	Decoding correction (pDecodingCorrection) Configures the decoding correction value used to determine the primary time reference point. Valid values are 14 to 143 µT.
20 to 16	FLXnCDD [4:0]	Cluster drift damping (pClusterDriftDamping) Configures the cluster drift damping value used in clock synchronization in microtick units to minimize accumulation of rounding errors. Valid values are 0 to 20 µT.
15 to 8	FLXnDCB [7:0]	Delay compensation channel B (pDelayCompensation[B]) Used to compensate for reception delays on the indicated channel. This compensates for propagation delays up to cPropagationDelayMax when microticks are within the range of 0.0125 to 0.05 µs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 µT.
7 to 0	FLXnDCA [7:0]	Delay compensation channel A (pDelayCompensation[A]) Used to compensate for reception delays on the indicated channel. This compensates for propagation delays up to cPropagationDelayMax when microticks are within the range of 0.0125 to 0.05 µs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 µT.

(13) FLXnGTUC06 – GTU Configuration Register 6

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00B4_H$

Initial value $0002\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnMOD[10:00]*										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnASR[10:00]*										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-46 FLXnGTUC06 Register Contents

Bit Position	Bit Name	Function
26 to 16	FLXnMOD [10:00]	Maximum oscillator drift (pdMaxDrift) Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT. Valid values are 2 to 1923 µT.
10 to 0	FLXnASR [10:00]	Accepted startup range (pdAcceptedStartupRange) Number of microticks that defines the maximum measurement deviation allowed for startup frames during integration. Valid values are 0 to 1875 µT.

(14) FLXnGTUC07 – GTU Configuration Register 7

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00B8_H$

Initial value $0002\ 0004_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	FLXnNSS[9:0]*									
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	FLXnSSL[9:0]*									
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-47 FLXnGTUC07 Register Contents

Bit Position	Bit Name	Function
25 to 16	FLXnNSS [9:0]	Number of static slots (gNumberOfStaticSlots) Configures the number of static slots in a cycle. At least 2 coldstart nodes must be configured to startup a FlexRay network. The number of static slots must be identical in all nodes of a cluster. Valid values are 2 to 1023.
9 to 0	FLXnSSL [9:0]	Static slot length (gdStaticSlot) Configures the duration of a static slot in macroticks. The static slot length must be identical in all nodes of a cluster. Valid values are 4 to 659 MT.

(15) FLXnGTUC08 – GTU Configuration Register 8

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00BC_H$

Initial value $0000\ 0002_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	FLXnNMS[12:00]*												
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	FLXnMSL[5:0]*					
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-48 FLXnGTUC08 Register Contents

Bit Position	Bit Name	Function
28 to 16	FLXnNMS [12:00]	Number of minislots (gNumberOfMinislots) Configures the number of minislots within the dynamic segment of a cycle. The number of minislots must be identical in all nodes of a cluster. Valid values are 0 to 7986.
5 to 0	FLXnMSL [5:0]	Minislot length (gdMinislot) Configures the duration of a minislot in macroticks. The minislot length must be identical in all nodes of a cluster. Valid values are 2 to 63 MT.

(16) FLXnGTUC09 – GTU Configuration Register 9

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00C0_H$

Initial value $0000\ 0101_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn DSI[1:0]*	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	FLXnMAPO[4:0]*					0	0	FLXnAPO[5:0]*					
R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-49 FLXnGTUC09 Register Contents

Bit Position	Bit Name	Function
17, 16	FLXnDSI[1:0]	Dynamic slot idle phase (gdDynamicSlotIdlePhase) The duration of the dynamic slot idle phase has to be greater than or equal to the idle detection time. Must be identical in all nodes of a cluster. Valid values are 0 to 2 Minislot.
12 to 8	FLXnMAPO [4:0]	Minislot action point offset (gdMinislotActionPointOffset) Configures the action point offset in macroticks. Must be identical in all nodes of a cluster. Valid values are 1 to 31 MT.
5 to 0	FLXnAPO [5:0]	Action point offset (gdActionPointOffset) Configures the action point offset in macroticks within static slots and symbol window. Must be identical in all nodes of a cluster. Valid values are 1 to 63 MT.

(17) FLXnGTUC10 – GTU Configuration Register 10

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00C4_H$

Initial value $0002\ 0005_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnMRC[10:00]*										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnMOC[13:00]*													
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-50 FLXnGTUC10 Register Content

Bit Position	Bit Name	Function
26 to 16	FLXnMRC [10:00]	Maximum rate correction (pRateCorrectionOut) Holds the maximum permitted rate correction value in microticks to be applied by the internal clock synchronization algorithm. The FlexRay controller checks only the internal rate correction value against the maximum rate correction value (absolute value). Valid values are 2 to 1923 µT.
13 to 0	FLXnMOC [13:00]	Maximum offset correction (pOffsetCorrectionOut) Holds the maximum permitted offset correction value in microticks to be applied by the internal clock synchronization algorithm (absolute value). The FlexRay controller checks only the internal offset correction value against the maximum offset correction value. Valid values are 5 to 15266 µT.

(18) FLXnGTUC11 – GTU Configuration Register 11

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $00C8_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnERC[2:0]*			0	0	0	0	0	FLXnEOC[2:0]*		
R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	FLXn ERCC[1:0]		0	0	0	0	0	0	FLXn EOCC[1:0]	
R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24-51 FLXnGTUC11 Register Contents

Bit Position	Bit Name	Function
26 to 24	FLXnERC [2:0]	External rate correction (pExternRateCorrection) Holds the external clock rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated rate correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 µT.
18 to 16	FLXnEOC [2:0]	External offset correction (pExternOffsetCorrection) Holds the external clock offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated offset correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 µT.
9, 8	FLXnERCC [1:0]	External rate correction control (vExternRateControl) By writing to FLXnERCC[1:0] the external rate correction is enabled as specified below. Should be modified only outside NIT. 00, 01 = No external rate correction 10 = External rate correction value subtracted from calculated rate correction value 11 = External rate correction value added to calculated rate correction value
1, 0	FLXnEOCC [1:0]	External offset correction control (vExternOffsetControl) By writing to FLXnEOCC[1:0] the external offset correction is enabled as specified below. Should be modified only outside NIT. 00, 01 = No external offset correction 10 = External offset correction value subtracted from calculated offset correction value 11 = External offset correction value added to calculated offset correction value

24.4.6 CC Status Registers

The status vector may change faster than the host can poll the status vector, depending on eray_bclk frequency.

(1) FLXnCCSV – CC Status Vector Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0100_H

Initial value $0010\ 4000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnPSL[5:0]						FLXnRCA[4:0]					FLXnWSV[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FLXn CSI	FLXn CSAI	FLXn CSNI	0	0	FLXn SLM[1:0]		FLXn HRQ	FLXn FSI	FLXnPOCS[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-52 FLXnCCSV Register Contents (1/3)

Bit Position	Bit Name	Function
29 to 24	FLXnPSL [5:0]	POC status log Indicates the status of FLXnPOCS[5:0] immediately before entering the HALT state. These bits are set to HALT (00 0100) when the FREEZE command is applied during the HALT state and FLXnFSI is not yet set, in other words, when it is not the FREEZE command that has caused the HALT state. These bits are reset to 00 0000 when leaving the HALT state.
23 to 19	FLXnRCA [4:0]	Number of remaining coldstart attempts (vRemainingColdstartAttempts) Indicates the number of remaining coldstart attempts. The RUN command resets this counter to the maximum number of coldstart attempts as configured by FLXnSUCC1.FLXnCSA[4:0]. The initial value of FLXnRCA[4:0] during the CONFIG and DEFAULT_CONFIG state is also FLXnSUCC1.FLXnCSA[4:0].
18 to 16	FLXnWSV [2:0]	Wakeup status (vPOC!WakeupStatus) Indicates the status of the current wakeup attempt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from the HALT to DEFAULT_CONFIG state or from the READY to STARTUP state. 000 = UNDEFINED. Wakeup is not yet executed. 001 = RECEIVED_HEADER. This value indicates the completion of wakeup due to the reception of a frame header without coding violation on either channel in the WAKEUP_LISTEN state. 010 = RECEIVED_WUP. This indicates the completion of wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in the WAKEUP_LISTEN state. 011 = COLLISION_HEADER. This value indicates that wakeup is stopped due to detection of a collision during wakeup pattern transmission by receiving a valid header on either channel. 100 = COLLISION_WUP. This value indicates that wakeup is stopped due to detection of a collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel. 101 = COLLISION_UNKNOWN. This value indicates that wakeup is stopped by leaving the WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header. 110 = TRANSMITTED. This value indicates that the transmission of the wakeup pattern has been successfully completed.

Table 24-52 FLXnCCSV Register Contents (2/3)

Bit Position	Bit Name	Function
14	FLXnCSI	Cold start inhibit (vColdStartInhibit) Indicates that the node is disabled from cold starting. The flag is set whenever the POC enters the READY state due to CHI command READY. The flag has to be reset under control of the host by CHI command ALLOW_COLDSTART (FLXnCMD[3:0] = 1001). 1 = Cold starting of node disabled 0 = Cold starting of node enabled
13	FLXnCSAI	Coldstart abort indicator Coldstart aborted. Reset by CHI command RESET_STATUS_INDICATORS or by transition from the HALT to DEFAULT_CONFIG state or from the READY state to the STARTUP state.
12	FLXnCSNI	Coldstart noise indicator (vColdStartNoise) Indicates that the cold start procedure occurred under noisy conditions. Reset by CHI command RESET_STATUS_INDICATORS or by transition from the HALT state to the DEFAULT_CONFIG state or from the READY state to the STARTUP state.
9, 8	FLXnSLM [1:0]	Slot mode (vPOC!SlotMode) Indicates the actual slot mode of the POC in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE. Default is SINGLE. Changes to ALL, depending on FLXnSUCC1.FLXnTSM. In the NORMAL_ACTIVE or NORMAL_PASSIVE state, the CHI command ALL_SLOTS will change the slot mode from SINGLE to ALL_PENDING, and then to ALL. These bits are set to SINGLE in all other states. 00 = SINGLE 01 = Reserved 10 = ALL_PENDING 11 = ALL
7	FLXnHRQ	Halt request (vPOC!CHIHaltRequest) Indicates that a request from the host has been received to halt the POC at the end of the communication cycle. Reset by CHI command RESET_STATUS_INDICATORS or by transition from the HALT state to the DEFAULT_CONFIG state or when entering the READY state.
6	FLXnFSI	Freeze status indicator (vPOC!Freeze) Indicates that the POC has entered the HALT state due to CHI command FREEZE or due to an error condition requiring an immediate POC halt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from the HALT state to the DEFAULT_CONFIG state.

Table 24-52 FLXnCCSV Register Contents (3/3)

Bit Position	Bit Name	Function
5 to 0	FLXnPOCS [5:0]	Protocol operation control status Indicates the current operational state of protocol operation control (POC) of the FlexRay controller. 00 0000 = DEFAULT_CONFIG state 00 0001 = READY state 00 0010 = NORMAL_ACTIVE state 00 0011 = NORMAL_PASSIVE state 00 0100 = HALT state 00 0101 = MONITOR_MODE state 00 0110…00 1110 = Reserved 00 1111 =CONFIG state Indicates the actual state of operation of the POC in the wakeup path. 01 0000 = WAKEUP_STANDBY state 01 0001 = WAKEUP_LISTEN state 01 0010 = WAKEUP_SEND state 01 0011 = WAKEUP_DETECT state 01 0100…01 1111 = Reserved Indicates the actual state of operation of the POC in the startup path. 10 0000 = STARTUP_PREPARE state 10 0001 = COLDSTART_LISTEN state 10 0010 = COLDSTART_COLLISION_RESOLUTION state 10 0011 = COLDSTART_CONSISTENCY_CHECK state 10 0100 = COLDSTART_GAP state 10 0101 = COLDSTART_JOIN State 10 0110 = INTEGRATION_COLDSTART_CHECK state 10 0111 = INTEGRATION_LISTEN state 10 1000 = INTEGRATION_CONSISTENCY_CHECK state 10 1001 = INITIALIZE_SCHEDULE state 10 1010 = ABORT_STARTUP state 10 1011 = STARTUP_SUCCESS state 10 1100…11 1111 = Reserved

(2) FLXnCCEV – CC Error Vector Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0104_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	FLXnPTAC[4:0]					FLXn ERRM[1:0]		0	0	FLXnCCFC[3:0]			
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset by transition from the HALT state to the DEFAULT_CONFIG state or when entering the READY state.

Table 24-53 FLXnCCEV Register Contents

Bit Position	Bit Name	Function
12 to 8	FLXnPTAC [4:0]	Passive to active count (vAllowPassiveToActive) This field indicates, by the number of consecutive even/odd cycle pairs, how much valid rate correction and offset correction were preformed during the period when the node is waiting to transit from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. The transition takes place when FLXnPTAC[4:0] equals FLXnSUCC1.FLXnPTA[4:0] -1.
7, 6	FLXnERRM [1:0]	Error mode (vPOC!ErrorMode) Indicates the actual error mode of the POC. 00 = ACTIVE 01 = PASSIVE 10 = COMM_HALT 11 = Reserved
3 to 0	FLXnCCFC [3:0]	Clock correction failed counter (vClockCorrectionFailed) The clock correction failed counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error is active. The clock correction failed counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors is active. The clock correction failed counter stops at 15.

(3) FLXnSCV – Slot Counter Value Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0110_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnSCCB[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnSCCA[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-54 FLXnSCV Register Contents

Bit Position	Bit Name	Function
26 to 16	FLXnSCCB [10:00]	Slot counter channel B (vSlotCounter[B]) Current slot counter value on channel B. The value is incremented by the FlexRay controller and reset at the start of a communication cycle. Valid values are 0 to 2047.
10 to 0	FLXnSCCA [10:00]	Slot counter channel A (vSlotCounter[A]) Current slot counter value on channel A. The value is incremented by the FlexRay controller and reset at the start of a communication cycle. Valid values are 0 to 2047.

(4) FLXnMTCCV – Macrotick and Cycle Counter Value Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0114_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	FLXnCCV[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	FLXnMTV[13:00]													
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-55 FLXnMTCCV Register Contents

Bit Position	Bit Name	Function
21 to 16	FLXnCCV [5:0]	Cycle counter value (vCycleCounter) Current cycle counter value. The value is incremented by the FlexRay controller at the start of a communication cycle. Valid values are 0 to 63.
13 to 0	FLXnMTV [13:0]	Macrotick value (vMacrotick) Current Macrotick value. The value is incremented by the FlexRay controller and reset at the start of a communication cycle. Valid values are 0 to 15999.

(5) FLXnRCV – Rate Correction Value Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0118_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FLXnRCV[11:00]											
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-56 FLXnRCV Register Contents

Bit Position	Bit Name	Function
11 to 0	FLXnRCV [11:00]	Rate correction value (vRateCorrection) Rate correction value (two's complement). This value is the internal rate correction value during the period before performing the threshold exceeding processing. If the FLXnRCV value exceeds the limits defined by FLXnGTUC010.FLXnMRC[10:00], flag FLXnSFS.FLXnRCLR is set to 1.

(6) FLXnOCV – Offset Correction Value Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + $011C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	FLXnOCV[18:16]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnOCV[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-57 FLXnOCV Register Contents

Bit Position	Bit Name	Function
18 to 0	FLXnOCV [18:00]	Offset correction value (vOffsetCorrection) Offset correction value (two's complement). This value is the internal offset correction value during the period before performing the threshold exceeding processing. If the FLXnOCV value exceeds the limits defined by FLXnGTUC010.FLXnMOC[13:00], flag FLXnSFS.FLXnOCLR is set to 1.

Note The external rate/offset correction value is added to/subtracted from the limited rate/offset correction value.

(7) FLXnSFS – Sync Frame Status Register

The maximum number of valid sync frames in a communication cycle is 15.

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0120_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	FLXn RCLR	FLXn MRCS	FLXn OCLR	FLXn MOCS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnVSBO[3:0]				FLXnVSBE[3:0]				FLXnVSAO[3:0]				FLXnVSAE[3:0]			
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-58 FLXnSFS Register Contents (1/2)

Bit Position	Bit Name	Function
19	FLXnRCLR	Rate correction limit reached The rate correction limit reached flag signals to the host, that the rate correction value has exceeded its limit as defined by FLXnGTUC10.FLXnMRC[10:00]. The flag is updated by the FlexRay controller at start of offset correction phase. 1 = Rate correction limit reached 0 = Rate correction below limit
18	FLXnMRCS	Missing rate correction signal The missing rate correction flag signals to the host, that no rate correction calculation can be performed because no pairs of even/odd sync frames were received. The flag is updated by the FlexRay controller at start of offset correction phase. 1 = Missing rate correction signal 0 = Rate correction signal valid
17	FLXnOCLR	Offset correction limit reached The offset correction limit reached flag signals to the host, that the offset correction value has exceeded its limit as defined by FLXnGTUC10.FLXnMOC[13:00]. The flag is updated by the FlexRay controller at start of offset correction phase. 1 = Offset correction limit reached 0 = Offset correction below limit
16	FLXnMOCS	Missing offset correction signal The missing offset correction flag signals to the host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the FlexRay controller at start of offset correction phase. 1 = Missing offset correction signal 0 = Offset correction signal valid
15 to 12	FLXnVSBO [3:0]	Valid sync frames channel B, odd communication cycle Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by FLXnSUCC1.FLXnTXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Table 24-58 FLXnSFS Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	FLXnVSBE [3:0]	Valid sync frames channel B, even communication cycle Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by FLXnSUCC1.FLXnTXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.
7 to 4	FLXnVSAO [3:0]	Valid sync frames channel A, odd communication cycle Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by FLXnSUCC1.FLXnTXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.
3 to 0	FLXnVSAE [3:0]	Valid sync frames channel A, even communication cycle Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by FLXnSUCC1.FLXnTXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Note Bits FLXnVSAE[3:0], FLXnVSAO[3:0], FLXnVSBE[3:0], and FLXnVSBO[3:0] are valid only if the respective channel is enabled by FLXnSUCC1.FLXnCCHA or FLXnSUCC1.FLXnCCHB.

(8) FLXnSWNIT – Symbol Window and NIT Status Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0124_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FLXn SWNIT SBNB	FLXn SWNIT SENB	FLXn SWNIT SBNA	FLXn SWNIT SENA	FLXn SWNIT MTSB	FLXn SWNIT MTSA	FLXn SWNIT TCSB	FLXn SWNIT SBSB	FLXn SWNIT SESB	FLXn SWNIT TCSA	FLXn SWNIT SBSA	FLXn SWNIT SESA
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

This register indicates the symbol window related status information, and it is updated by the FlexRay controller at the end of the symbol window for each channel. During startup, the status data is not updated. This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Table 24-59 FLXnSWNIT Register Contents (1/2)

Bit Position	Bit Name	Function
11	FLXn SWNIT SBNB	Slot boundary violation during NIT channel B (vSS!BViolationB) 1 = Slot boundary violation during NIT detected on channel B 0 = No slot boundary violation detected
10	FLXn SWNIT SENB	Syntax error during NIT channel B (vSS!SyntaxErrorB) 1 = Syntax error during NIT detected on channel B 0 = No syntax error detected
9	FLXn SWNIT SBNA	Slot boundary violation during NIT channel A (vSS!BViolationA) 1 = Slot boundary violation during NIT detected on channel A 0 = No slot boundary violation detected
8	FLXn SWNIT SENA	Syntax error during NIT channel A (vSS!SyntaxErrorA) 1 = Syntax error during NIT detected on channel A 0 = No syntax error detected
7	FLXn SWNIT MTSB	MTS received on channel B (vSS!ValidMTSB) Media access test symbol (MTS) received on channel B during the preceding symbol window. Updated by the FlexRay controller for each channel at the end of the symbol window. When this bit is set to 1, interrupt flag FLXnSIR.FLXnMTSBF is also set to 1. 1 = MTS symbol received on channel B 0 = No MTS symbol received on channel B
6	FLXn SWNIT MTSA	MTS received on channel A (vSS!ValidMTSA) Media access test symbol (MTS) received on channel A during the preceding symbol window. Updated by the FlexRay controller for each channel at the end of the symbol window. When this bit is set to 1, interrupt flag FLXnSIR.FLXnMTSAF is also set to 1. 1 = MTS symbol received on channel A 0 = No MTS symbol received on channel A
5	FLXn SWNIT TCSB	Transmission conflict in symbol window channel B (vSS!TxConflictB) 1 = Transmission conflict in symbol window detected on channel B 0 = No transmission conflict detected

Table 24-59 FLXnSWNIT Register Contents (2/2)

Bit Position	Bit Name	Function
4	FLXn SWNIT SBSB	Slot boundary violation in symbol window channel B (vSS!BViolationB) 1 = Slot boundary violation during symbol window detected on channel B 0 = No slot boundary violation detected
3	FLXn SWNIT SESB	Syntax error in symbol window channel B (vSS!SyntaxErrorB) 1 = Syntax error during symbol window detected on channel B 0 = No syntax error detected
2	FLXn SWNIT TCSA	Transmission conflict in symbol window channel A (vSS!TxConflictA) 1 = Transmission conflict in symbol window detected on channel A 0 = No transmission conflict detected
1	FLXn SWNIT SBSA	Slot boundary violation in symbol window channel A (vSS!BViolationA) 1 = Slot boundary violation during symbol window detected on channel A 0 = No slot boundary violation detected
0	FLXn SWNIT SESA	Syntax error in symbol window channel A (vSS!SyntaxErrorA) 1 = Syntax error during symbol window detected on channel A 0 = No syntax error detected

Note Bits FLXnSWNITSENA, FLXnSWNITSBNA, FLXnSWNITSENB, and FLXnSWNITSBNB are NIT related status information. They are updated by the FlexRay controller at the end of the NIT for each channel.

(9) FLXnACS – Aggregated Channel Status Register

The aggregated channel status provides the host with the channel status for all communication slots regardless of whether they are assigned for transmission or reception. The aggregated channel status also includes status data from the symbol window and the network idle time (NIT). The status data is updated (set) after each slot and retained until it is reset by the host. During startup the status data is not updated. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0128_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnACS Read

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	FLXn SBVBE	FLXn CIBE	FLXn CEDBE	FLXn SEDBE	FLXn VFRBF

7	6	5	4	3	2	1	0
0	0	0	FLXn SBVAE	FLXn CIAE	FLXn CEDAE	FLXn SEDAE	FLXn VFRAF

Table 24-60 FLXnACS Read Register Contents (1/2)

Bit Position	Bit Name	Function
12	FLXnSBVBE	Slot boundary violation on channel B (vSS!BViolationB) One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots including symbol window and NIT). 1 = Slot boundary violation(s) observed on channel B 0 = No slot boundary violation observed
11	FLXnCIBE	Communication indicator channel B One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 1 = Valid frame(s) received on channel B in slots containing any additional communication 0 = No valid frame(s) received in slots containing any additional communication
10	FLXnCEDBE	Content error detected on channel B (vSS!ContentErrorB) One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period. 1 = Frame(s) with content error received on channel B 0 = No frame with content error received

Table 24-60 FLXnACS Read Register Contents (2/2)

Bit Position	Bit Name	Function
9	FLXnSEDBE	Syntax error detected on channel B (vSS!SyntaxErrorB) One or more syntax errors in static or dynamic slots including symbol window and NIT were observed on channel B. 1 = Syntax error(s) observed on channel B 0 = No syntax error observed
8	FLXnVFRBF	Valid frame received on channel B (vSS!ValidFrameB) One or more valid frames were received on channel B in any static or dynamic slot during the observation period. Reset under control of the Host. 1 = Valid frame(s) received on channel B 0 = No valid frame received
4	FLXnSBVAE	Slot boundary violation on channel A (vSS!BViolationA) One or more slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots including symbol window and NIT). 1 = Slot boundary violation(s) observed on channel A 0 = No slot boundary violation observed
3	FLXnCIAE	Communication indicator channel A One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 1 = Valid frame(s) received on channel A in slots containing any additional communication 0 = No valid frame(s) received in slots containing any additional communication
2	FLXnCEDAE	Content error detected on channel A (vSS!ContentErrorA) One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period. 1 = Frame(s) with content error received on channel A 0 = No frame with content error received
1	FLXnSEDAE	Syntax error detected on channel A (vSS!SyntaxErrorA) One or more syntax errors in static or dynamic slots including symbol window and NIT were observed on channel A. 1 = Syntax error(s) observed on channel A 0 = No syntax error observed
0	FLXnVFRAF	Valid frame received on channel A (vSS!ValidFrameA) One or more valid frames were received on channel A in any static or dynamic slot during the observation period. 1 = Valid frame(s) received on channel A 0 = No valid frame received

Note The set condition of flags FLXnCIAE and FLXnCIBE is also fulfilled, if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

When one of the flags FLXnSEDBE, FLXnCEDBE, FLXnCIBE, and FLXnSBVBE changes from 0 to 1, interrupt flag FLXnEIR.FLXnEDBE is set to 1. When one of the flags FLXnSEDAE, FLXnCEDAE, FLXnCIAE, and FLXnSBVAE changes from 0 to 1, interrupt flag FLXnEIR.FLXnEDAE is set to 1.

(b) FLXnACS Write

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8
			FLXn CLSBVB	FLXn CLCIB	FLXn CLCEDB	FLXn CLSEDB	FLXn CLVFRB

7	6	5	4	3	2	1	0
			FLXn CLSBVA	FLXn CLCIA	FLXn CLCEDA	FLXn CLSEDA	FLXn CLVFRA

Table 24-61 FLXnACS Write Register Contents

Bit Position	Bit Name	Function
31 to 0	See above	Clears the corresponding flags described in Table 24-60, FLXnACS Read Register Contents. 1 = Clears the corresponding flag. 0 = No function

Note Writing to bits 31 to 13 and 7 to 5 is ignored.

(10) FLXnESIDm – Even Sync ID Register (m = 01 to 15)

Registers FLXnESID01 to FLXnESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXnESID01 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXnESID01 holds the respective sync frame ID as configured in message buffer 0 and flags FLXnRXEA and FLXnRXEB are set. The value is updated during the NIT of each even communication cycle. This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0130_H to <FLXn_base> + 0168_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXn RXEB	FLXn RXEA	0	0	0	0	FLXn EIDm[9:0]									
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-62 FLXnESIDm Register Contents

Bit Position	Bit Name	Function
15	FLXnRXEB	Received/configured even sync ID on channel B Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXnEIDm[9:0] (FLXnESID01 only). 1 = Sync frame received on channel B, or node configured to transmit sync frames 0 = No sync frame received on channel B, or node not configured to transmit sync frames
14	FLXnRXEA	Received/configured even sync ID on channel A Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXnEIDm[9:0] (FLXnESID01 only). 1 = Sync frame received on channel A, or node configured to transmit sync frames 0 = No sync frame received on channel A, or node not configured to transmit sync frames
9 to 0	FLXnEIDm [9:0]	Even sync ID (vsSyncIDListA,B even) Sync frame ID even communication cycle.

(11) FLXnOSIDm – Odd Sync ID Register (m = 01 to 15)

Registers FLXnOSID01 to FLXnOSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXnOSID01 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXnOSID01 holds the respective sync frame ID as configured in message buffer 0 and flags FLXnRXOA are FLXnRXOB are set. The value is updated during the NIT of each odd communication cycle. This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0170_H to <FLXn_base> + $01A8_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXn RXOB	FLXn RXOA	0	0	0	0	FLXnOIDm[9:0]									
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-63 FLXnOSIDm Register Contents

Bit Position	Bit Name	Function
15	FLXn RXOB	Received/configured odd sync ID on channel B Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXnOIDm[9:0] (FLXnOSID01 only). 1 = Sync frame received on channel B, or node configured to transmit sync frames 0 = No sync frame received on channel B, or node not configured to transmit sync frames
14	FLXn RXOA	Received/configured odd sync ID on channel A Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXnOIDm[9:0] (FLXnOSID01 only). 1 = Sync frame received on channel A, or node configured to transmit sync frames 0 = No sync frame received on channel A, or node not configured to transmit sync frames
9 to 0	FLXnOIDm [9:0]	Odd sync ID (vsSyncIDListA,B odd) Sync frame ID odd communication cycle.

(12) FLXnNMVm – Network Management Vector Register (m = 1 to 3)

The three network management registers hold the accrued NM vector (0 to 12 bytes). The accrued NM vector is generated by the FlexRay controller by bit-wise ORing each NM vector received (valid static frames with FLXnRHPPI = 1) on each channel (see section 24.5.6, Network Management).

The FlexRay controller updates the NM vector at the end of each communication cycle as long as the FlexRay controller is either the NORMAL_ACTIVE or NORMAL_PASSIVE state.

FLXnNMVm-bytes exceeding the configured NM vector length are not valid.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $01B0_H$ to <FLXn_base> + $01B8_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnNMm[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnNMm[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The table below shows the assignment of the data bytes to the network management vector.

Table 24-64 Assignment of Data Bytes to Network Management Vector

Bit / Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnNMV1	Data 3								Data 2							
FLXnNMV2	Data 7								Data 6							
FLXnNMV3	Data 11								Data 10							

Bit / Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnNMV1	Data 1								Data 0							
FLXnNMV2	Data 5								Data 4							
FLXnNMV3	Data 9								Data 8							

24.4.7 Message Buffer Control Registers

(1) FLXnMRC – Message RAM Configuration Register

The message RAM configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO. This register can be written during the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0300_H

Initial value $0180\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXn SPLM*	FLXn SEC[1:0]*		FLXnLCB[7:0]*							
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnFFB[7:0]*								FLXnFDB[7:0]*							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

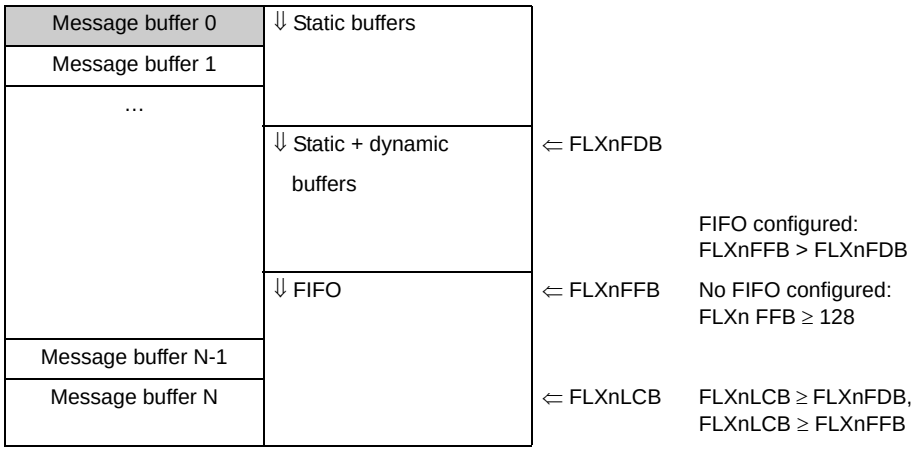
Table 24-65 FLXnMRC Register Contents (1/2)

Bit Position	Bit Name	Function
26	FLXnSPLM	sync frame payload multiplex This bit is evaluated only if the node is configured as a sync node (FLXnSUCC1.FLXnTXSY = 1) or for single slot mode operation (FLXnSUCC1.FLXnTSM = 1). When this bit is set to 1, message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B. When this bit is set to 0, sync frames are transmitted from message buffer 0 with the same payload data on both channels. Note that channels have to be selected by the channel filter configuration for message buffer 0 and message buffer 1 according to the setting of this bit. 1 = Both message buffers 0 and 1 are locked against reconfiguration 0 = Only message buffer 0 locked against reconfiguration
25, 24	FLXnSEC [1:0]	Secure buffers Not evaluated when the FlexRay controller is in the DEFAULT_CONFIG or CONFIG state. 00 = Reconfiguration is enabled for message buffers whose numbers are less than FLXnFFB. As an exception, in nodes configured for sync frame transmission or for single slot mode operation, message buffer 0 (and if FLXnSPLM = 1, also message buffer 1) is always locked. 01 = Reconfiguration is locked for message buffers whose numbers are less than FLXnFDB, or equal to or larger than FLXnFFB. Transmission is disabled for static-segment message buffers whose assigned numbers are equal to or larger than FLXnFDB. 10 = Reconfiguration of all message buffers locked 11 = Reconfiguration of all message buffers are locked, and transmission is disabled for static-segment message buffers whose numbers are equal to or larger than FLXnFDB.

Table 24-65 FLXnMRC Register Contents (2/2)

Bit Position	Bit Name	Function
23 to 16	FLXnLCB [7:0]	Last configured buffer 0 to 127 = Number of message buffers is FLXnLCB + 1. 128 = No message buffer configured
15 to 8	FLXnFFB[7:0]	First buffer of FIFO 0 = All message buffers is assigned to the FIFO. 1 to 127 = Message buffers from FLXnFFB to FLXnLCB are assigned to the FIFO. 128 = No message buffer is assigned to the FIFO.
7 to 0	FLXnFDB [7:0]	First dynamic buffer 0 = No message buffer group is configured exclusively for static segments. 1 to 127 = Message buffers 0 to FLXnFDB - 1 are reserved for static segments. 128 = No dynamic message buffers configured

Note When the node is configured as a sync node (FLXnSUCC1.FLXnTXSY = 1) or for single slot mode operation (FLXnSUCC1.FLXnTSM = 1), message buffers 0 and 1 are reserved for sync frames or single slot frames. They must be configured with the node-specific key slot ID. When the node is not configured as a sync node or for single slot operation, message buffers 0 and 1 are treated in the same way as any other message buffers.



The user has to ensure that the configuration defined by FLXnFDB[7:0], FLXnFFB[7:0], and FLXnLCB[7:0] is valid. The FlexRay controller does not check for erroneous configurations.

Note The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer.

When two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "Static Buffers" or at the beginning of the "Static + Dynamic Buffers" section.

The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore, at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement, a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non-protocol-conformant configuration in which transmission slots in the static segment are not used would still be operational.

The payload length and the length of the data section need to be configured identically for all message buffers belonging to the FIFO using FLXnWRHS2.FLXnWHPLC[6:0] and FLXnWRHS3.FLXnWHDP[10:00]. When the FlexRay controller is not in the DEFAULT_CONFIG or CONFIG state, reconfiguration of message buffers belonging to the FIFO is locked.

(2) FLXnFRF – FIFO Rejection Filter Register

The FIFO rejection filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO rejection filter mask register, this register determines whether a message is rejected by the FIFO. The FLXnFRF register can be written during the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0304_H

Initial value $0180\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	FLXn RNF*	FLXn RSS*	FLXnCYF[6:0]*						
R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	FLXnFID[10:00]*											FLXnCH [1:0]*	
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-66 FLXnFRF Register Contents

Bit Position	Bit Name	Function
24	FLXnRNF	Reject Null Frames If this bit is set, received null frames are not stored in the FIFO. 1 = Reject all null frames 0 = Null frames are stored in the FIFO
23	FLXnRSS	Reject in Static Segment If this bit is set, the FIFO is used only for the dynamic segment. 1 = Reject messages in static segment 0 = FIFO also used in static segment
22 to 16	FLXnCYF [6:0]	Cycle counter filter The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXnCYF[6:0], all frames are rejected. For details on the configuration of the cycle counter filter, see section 24.5.7, Filtering and Masking.
12 to 2	FLXnFID [10:00]	Frame ID Filter Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXnFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FLXnFRFM.FLXnMFID[10:00] is zero, a frame ID filter value of zero means that **no** frame ID is rejected. 0...2047 = Frame ID filter values
1, 0	FLXnCH[1:0]	Channel filter 11 = No reception 10 = Receive only on channel A 01 = Receive only on channel B 00 = Receive on both channels **Note:** If the configuration specifies that reception is performed on both channels, static segments and identical frames (identical between channel A and B) of both channels are stored in the FIFO.

(3) FLXnFRFM – FIFO Rejection Filter Mask Register

The FIFO rejection filter mask specifies which of the corresponding FLXnFRF bits are relevant for rejection filtering. If a bit is set, it indicates that the state of the corresponding bit in the FLXnFRF register will not affect whether or not the message is rejected by the FIFO. The FLXnFRFM register can be written during the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0308_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	FLXnMFID[10:00]*											0	0
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 24-67 FLXnFRFM Register Contents

Bit Position	Bit Name	Function
12 to 2	FLXnMFID [10:00]	FIFO rejection filter mask 1 = Ignore corresponding FIFO rejection filter bit. 0 = Corresponding FIFO rejection filter bit not ignored.

(4) FLXnFCL – FIFO Critical Level Register

The FlexRay controller accepts modifications of the register in the DEFAULT_CONFIG or CONFIG state only.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $030C_H$

Initial value $0000\ 0080_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FLXnCL[7:0]*							
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-68 FLXnFCL Register Contents

Bit Position	Bit Name	Function
7 to 0	FLXnCL[7:0]	Critical level When the receive FIFO fill level FLXnFSR.FLXnRFFL[7:0] is equal or greater than the critical level configured by FLXnCL[7:0], the receive FIFO critical level flag FLXnFSR.FLXnRFCL is set. If FLXnCL[7:0] is programmed to values > 128, bit FLXnFSR.FLXnRFCL is never set. When FLXnFSR.FLXnRFCL changes from 0 to 1, bit FLXnSIR.FLXnRFCLF is set to 1, and if enabled, an interrupt is generated.

24.4.8 Message Buffer Status Registers

(1) FLXnMHDS – Message Handler Status Register

A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. This register will also be cleared by hard reset or by CHI command CLEAR_RAMS.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0310_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnMHDS Read

31	30	29	28	27	26	25	24
0	FLXnMBU[6:0]						
23	22	21	20	19	18	17	16
0	FLXnMBT[6:0]						
15	14	13	12	11	10	9	8
0	FLXnFMB[6:0]						
7	6	5	4	3	2	1	0
FLXn CRAM	FLXn MFMBE	FLXn FMBDE	FLXn DTBFBE	FLXn DTBFAE	FLXn DMRE	FLXn DOBFE	FLXn DIBFE

Table 24-69 FLXnMHDS Read Register Contents (1/2)

Bit Position	Bit Name	Function
30 to 24	FLXnMBU [6:0]	Message buffer updated Number of message buffer that was updated last by the FlexRay controller. For this message buffer, the respective ND and/or MBC flag in the FLXnNDAT1/2/3/4 registers and the FLXnMBSC1/2/3/4 registers are also set.
22 to 16	FLXnMBT [6:0]	Message buffer transmitted Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective FLXnTXR flag in the transmission request register 1, 2, 3, 4 was reset.
14 to 8	FLXnFMB [6:0]	Faulty message buffer ECC error occurred when reading from the message buffer referenced by FLXnFMB[6:0] or during data transfer from the input buffer/transient buffer A, B to the message buffer. The value is valid only when one of the flags FLXnDIBFE, FLXnDMRE, FLXnDTBFAE, and FLXnDTBFBE, and flag FLXnFMBDE is set. Is not updated while flag FLXnFMBDE is set.
7	FLXnCRAM	Clear all internal RAM's Signals that execution of the CHI command CLEAR_RAMS is ongoing (all bits of all internal RAM blocks are written to 0). The bit is set by hardware reset or by CHI command CLEAR_RAMS. 1 = Execution of the CHI command CLEAR_RAMS ongoing 0 = No execution of the CHI command CLEAR_RAMS
6	FLXnMFMBE	Multiple faulty message buffers detected 1 = Another faulty message buffer was detected while flag FLXnFMBDE is set. 0 = No additional faulty message buffer

Table 24-69 FLXnMHDS Read Register Contents (2/2)

Bit Position	Bit Name	Function
5	FLXnFMBDE	Faulty message buffer detected 1 = Message buffer referenced by FLXnFMB[6:0] holds faulty data due to a ECC error 0 = No faulty message buffer
4	FLXn DTBFBE	ECC error transient buffer RAM B 1 = ECC error occurred when reading transient buffer RAM B 0 = No ECC error
3	FLXn DTBFAE	ECC error transient buffer RAM A 1 = ECC error occurred when reading transient buffer RAM A 0 = No ECC error
2	FLXnDMRE	ECC error message RAM 1 = ECC error occurred when reading the message RAM 0 = No ECC error
1	FLXnDOBFE	ECC error output buffer RAM 1, 2 1 = ECC error occurred when reading output buffer RAM 1, 2 0 = No ECC error
0	FLXnDIBFE	ECC error input buffer RAM 1, 2 1 = ECC error occurred when reading input buffer RAM 1, 2 0 = No ECC error

Note 1. When one of the flags FLXnDIBFE, FLXnDOBFE, FLXnDMRE, FLXnDTBFAE, and FLXnDTBFBE changes from 0 to 1, the ECC error detection interrupt INTECCFRED is asserted.

Note 2. FLXnMBT[6:0] and FLXnMBU[6:0] are reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

(b) FLXnMHDS Write

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
	FLXnCL MFMB	FLXnCL FMBD	FLXnCL DTBFB	FLXnCL DTBFA	FLXnCL DMR	FLXnCL DOBF	FLXnCL DIBF

Table 24-70 FLXnMHDS Write Register Contents

Bit Position	Bit Name	Function
31 to 0	See above	Clears the corresponding flags described in table 24-69, FLXnMHDS Read Register Contents. 1 = Clears the corresponding flag. 0 = No function

Note Writing to bits 31 to 7 is ignored.

(2) FLXnLDTS – Last Dynamic Transmit Slot Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0314_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	FLXnLDTB[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnLDTA[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-71 FLXnLDTS Register Contents

Bit Position	Bit Name	Function
26 to 16	FLXnLDTB [10:00]	Last dynamic transmission channel B Value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to 0 if no frame was transmitted during the dynamic segment.
10 to 0	FLXnLDTA [10:00]	Last dynamic transmission channel A Value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to 0 if no frame was transmitted during the dynamic segment.

(3) FLXnFSR – FIFO Status Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0318_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnRFFL[7:0]								0	0	0	0	0	FLXn RFO	FLXn RFCL	FLXn RFNE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-72 FLXnFSR Register Contents

Bit Position	Bit Name	Function
15 to 8	FLXnRFFL [7:0]	Receive FIFO fill level Number of FIFO buffers filled with new data not yet read by the host. Maximum value is 128.
2	FLXnRFO	Receive FIFO overrun The flag is set by the FlexRay controller when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag FLXnEIR.FLXnRFOE is set. The flag is cleared by the next FIFO read access issued by the host. 1 = A receive FIFO overrun has been detected 0 = No receive FIFO overrun detected
1	FLXnRFCL	Receive FIFO critical level This flag is set when the receive FIFO fill level FLXnRFFL[7:0] is equal or greater than the critical level as configured by FLXnFCL.FLXnCL[7:0]. The flag is cleared by the FlexRay controller as soon as FLXnRFFL[7:0] drops below FLXnFCL.FLXnCL[7:0]. When FLXnRFCL changes from 0 to 1, bit FLXnSIR.FLXnRFCLF is set to 1, and if enabled, an interrupt is generated. 1 = Receive FIFO critical level reached 0 = Receive FIFO below critical level
0	FLXnRFNE	Receive FIFO not empty This flag is set by the FlexRay controller when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag FLXnSIR.FLXnRFNEF is set. This bit is reset after the host has read all message from the FIFO. 1 = Receive FIFO is not empty 0 = Receive FIFO is empty

(4) FLXnMHDF – Message Handler Constraints Flags Register

Some constraints exist for the message handler regarding eray_bclk frequency, message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FLXnMHDF.

A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. A hard reset will also clear the register. This register is reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + $031C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

(a) FLXnMHDF Read

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	FLXn WAHPE
7	6	5	4	3	2	1	0
FLXn TNSBE	FLXn TNSAE	FLXn TBFBE	FLXn TBFAE	FLXn FNFBE	FLXn FNFAE	FLXn SNUBE	FLXn SNUAE

Table 24-73 FLXnMHDF Read Register Contents (1/2)

Bit Position	Bit Name	Function
8	FLXnWAHPE	Write attempt to header partition Outside DEFAULT_CONFIG and CONFIG state, this flag is set by the FlexRay controller when the message handler tries to write message data into the header partition of the message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses. 1 = Write attempt to header partition 0 = No write attempt to header partition
7	FLXnTNSBE	Transmission not started channel B This flag is set by the FlexRay controller when the message handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot. 1 = Transmission not started on channel B 0 = No transmission not started on channel B
6	FLXnTNSAE	Transmission not started channel A This flag is set by the FlexRay controller when the message handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot. 1 = Transmission not started on channel A 0 = No transmission not started on channel A

Table 24-73 FLXnMHDF Read Register Contents (2/2)

Bit Position	Bit Name	Function
5	FLXnTBFBE	Transient buffer access failure B This flag is set by the FlexRay controller when a read or write access to TBF B requested by PRT B could not complete within the available time. 1 = TBF B access failure 0 = No TBF B access failure
4	FLXnTBFAE	Transient buffer access failure A This flag is set by the FlexRay controller when a read or write access to TBF A requested by PRT A could not complete within the available time. 1 = TBF A access failure 0 = No TBF A access failure
3	FLXnFNFBE	Find sequence not finished channel B This flag is set by the FlexRay controller when the message handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel B. 1 = Find sequence not finished for channel B 0 = No find sequence not finished for channel B
2	FLXnFNFAE	Find sequence not finished channel A This flag is set by the FlexRay controller when the message handler, due to overload condition, was not able to finish a find sequence (scan of message RAM for matching message buffer) with respect to channel A. 1 = Find sequence not finished for channel A 0 = No find sequence not finished for channel A
1	FLXnSNUBE	Status not updated channel B This flag is set by the FlexRay controller when the message handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel B. 1 = MBS for channel B not updated 0 = No overload condition occurred when updating MBS for channel B
0	FLXnSNUAE	Status not updated channel A This flag is set by the FlexRay controller when the message handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel A. 1 = MBS for channel A not updated 0 = No overload condition occurred when updating MBS for channel A

Note When one of the flags FLXnSNUAE, FLXnSNUBE, FLXnFNFAE, FLXnFNFBE, FLXnDTBFAE, FLXnDTBFBE, FLXnTNSAE, FLXnTNSBE, and FLXnWAHPE changes from 0 to 1, interrupt flag FLXnEIR.FLXnMHFE is set to 1.

(b) FLXnMHDF Write

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
							FLXnCL WAHP
7	6	5	4	3	2	1	0
FLXnCL TNSB	FLXnCL TNSA	FLXnCL TBFB	FLXnCL TBFA	FLXnCL FNFB	FLXnCL FNFA	FLXnCL SNUB	FLXnCL SNUA

Table 24-74 FLXnMHDF Write Register Contents

Bit Position	Bit Name	Function
31 to 0	See above	Clears the corresponding flags described in table 24-73, FLXnMHDF Read Register Contents. 1 = Clears the corresponding flag. 0 = No function

Note Writing to bits 31 to 9 is ignored.

(5) FLXnTXRQ1/2/3/4 – Transmission Request 1/2/3/4 Registers

The four registers reflect the state of the FLXnTXR flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining FLXnTXR flags have no meaning.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $032C_H$
<FLXn_base> + 0328_H
<FLXn_base> + 0324_H
<FLXn_base> + 0320_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

FLXnTXRQ4:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnTXR[127:112]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnTXR[111:096]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnTXRQ3:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnTXR[095:080]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnTXR[079:064]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnTXRQ2:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnTXR[063:048]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnTXR[047:032]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnTXRQ1:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnTXR[031:016]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnTXR[015:000]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-75 FLXnTXRQ1/2/3/4 Registers Contents

Bit Position	Bit Name	Function
31 to 0 in all registers	FLXnTXR [127:000]	Transmission request If the flag is set, the corresponding message buffer is ready for transmission or is actually transmitting data. In single-shot mode, the flags are reset after transmission has completed.

(6) FLXnNDAT1/2/3/4 – New Data 1/2/3/4 Registers

The four registers reflect the state of the FLXnND flags of all configured message buffers. FLXnND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining FLXnND flags have no meaning. The registers are reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $033C_H$
<FLXn_base> + 0338_H
<FLXn_base> + 0334_H
<FLXn_base> + 0330_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

FLXnNDAT4:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnND[127:112]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnND[111:096]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnNDAT3:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnND[095:080]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnND[079:064]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnNDAT2:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnND[063:048]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnND[047:032]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnNDAT1:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnND[031:016]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnND[015:000]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-76 FLXnNDAT1/2/3/4 Registers Contents

Bit Position	Bit Name	Function
31 to 0 in all registers	FLXnND [127:000]	New data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An FLXnND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the output buffer.

(7) FLXnMBSC1/2/3/4 – Message Buffer Status Changed 1/2/3/4 Registers

The four registers reflect the state of the FLXnMBC flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining FLXnMBC flags have no meaning. The registers are reset when the FlexRay controller leaves the CONFIG state or enters the STARTUP state.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $034C_H$
<FLXn_base> + 0348_H
<FLXn_base> + 0344_H
<FLXn_base> + 0340_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

FLXnMBSC4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnMBC[127:112]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMBC[111:096]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnMBSC3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnMBC[095:080]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMBC[079:064]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnMBSC2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnMBC[063:048]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMBC[047:032]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

FLXnMBSC1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnMBC[031:016]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMBC[015:000]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-77 FLXnMBSC1/2/3/4 Registers Contents

Bit Position	Bit Name	Function
31 to 0 in all registers	FLXnMBC [127:000]	Message buffer status changed An MBC flag is set whenever the message handler changes one of the status flags FLXnMBSVFRA, FLXnMBSVFRB, FLXnMBSSEOA, FLXnMBSSEOB, FLXnMBSCEOA, FLXnMBSCEOB, FLXnMBSSVOA, FLXnMBSSVOB, FLXnMBSTCIA, FLXnMBSTCIB, FLXnMBSESA, FLXnMBSESB, FLXnMBSMLST, FLXnMBSFTA, or FLXnMBSFTB in the header section (See section 24.4.11 (5), FLXnMBS – Message Buffer Status Register, and section 24.5.12 (1), Header Partition, Header 4.) of the respective message buffer. An MBC flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the output buffer.

24.4.9 Identification Registers

(1) FLXnCREL – Core Release Register

Access This register can be read in 32-bit units.

Address <FLXn_base> + $03F0_H$

Initial value $1027\ 1031_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnREL[3:0]				FLXnSTEP[7:0]								FLXnYEAR[3:0]			
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMON[7:0]								FLXnDAY[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-78 FLXnCREL Register Contents

Bit Position	Bit Name	Function
31 to 28	FLXnREL [3:0]	Core release One digit, BCD-coded. 1_H: The IP release is the Revision 1.0.2
27 to 20	FLXnSTEP [7:0]	Step of core release Two digits, BCD-coded. 02_H: The IP release is the Revision 1.0.2
19 to 16	FLXnYEAR [3:0]	Design time stamp, year One digit, BCD-coded. 7_H: The IP was designed in the year 2007.
15 to 8	FLXnMON [7:0]	Design time stamp, month Two digits, BCD-coded. 10_H: The IP was designed in October.
7 to 0	FLXnDAY [7:0]	Design time stamp, day Two digits, BCD-coded. 31_H: The IP was designed on the 31th of October.

Table below shows how releases are coded in register FLXnCREL.

Table 24-79 Coding for Releases

Release	Step	Sub-Step	Name
0	7	0	Beta2
0	7	1	Beta2ct
0	7	2	R1.0 RC1
1	0	0	Revision 1.0.0
1	0	1	Revision 1.0.1
1	0	2	Revision 1.0.2

(2) FLXnENDN – Endian Register

This register indicates the endianness test value.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $03F4_H$

Initial value $8765\ 4321_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnETV[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnETV[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-80 FLXnENDN Register Contents

Bit Position	Bit Name	Function
31 to 0	FLXnETV [31:00]	Endianness test value The endianness test value is $8765\ 4321_H$.

24.4.10 Input Buffer

Double buffer structure consisting of input buffer host and input buffer shadow. While the host can write to input buffer host, the transfer to the message RAM is done from input buffer shadow. The input buffer holds the header and data sections to be transferred to the selected message buffer in the message RAM. It is used to configure the message buffers in the message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the message RAM from the input buffer, the message buffer status as described in section 24.4.11 (5), FLXnMBS – Message Buffer Status Register, is automatically reset to 0.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the FlexRay controller is in the DEFAULT_CONFIG or CONFIG state. For those message buffers, only the payload length configured and the data pointer need to be configured via FLXnWRHS2.FLXnWHPLC[6:0] and FLXnWRHS3.FLXnWHDP[10:00]. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between input buffer (IBF) and message RAM is described in detail in section 24.5.11, Message Handling, "Data transfer from input buffer to message RAM".

(1) FLXnWRDSm – Write Data Section Register (m = 01 to 64)

This register holds the data words to be transferred to the data section of the addressed message buffer. The data words (DWn) are written to the message RAM in transmission order from DW1 (byte 0, byte 1) to DWPL (PL = number of data words as defined by the payload length configured FLXnWRHS2.FLXnWHPLC[6:0]).

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0400_H to <FLXn_base> + $04FC_H$

Initial Value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnWDMDm[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnWDMDm[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-81 FLXnWRDSm Register Contents

Bit Position	Bit Name	Function
31 to 0	FLXn WDMDm [31:00]	Message data FLXnWDMDm[07:00] = DW_{2m-1}, $byte_{4m-4}$ FLXnWDMDm[15:08] = DW_{2m-1}, $byte_{4m-3}$ FLXnWDMDm[23:16] = DW_{2m}, $byte_{4m-2}$ FLXnWDMDm[31:24] = DW_{2m}, $byte_{4m-1}$

Note DW_{127} is located on FLXnWRDS64.FLXnWDMD64[15:00]. In this case, FLXnWRDS64.FLXnWDMD64[31:16] is unused (no valid data). The input buffer RAMs are initialized to 0 when leaving hard reset or by CHI command CLEAR_RAMS.
Transmission order on the FlexRay bus is FLXnWRDSm[7:0], FLXnWRDSm[15:8], FLXnWRDSm[23:16], and then FLXnWRDSm[31:24] with the most significant bit transmitted first. To check how the endianness of the FlexRay controller matches with the host CPU's endianness, read register FLXnENDN.

(2) FLXnWRHS1 – Write Header Section 1 Register

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0500_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnWHMBI	FLXnWHTXM	FLXnWHPPIT	FLXnWHCFG	FLXnWHCHB	FLXnWHCHA	0	FLXnWHCYC[6:0]						
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnWHFID[10:00]										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-82 FLXnWRHS1 Register Contents (1/2)

Bit Position	Bit Name	Function
29	FLXnWHMBI	Message buffer interrupt This bit enables the receive/transmit interrupt for the corresponding message buffer. After a dedicated receive buffer has been updated by the message handler, flag FLXnRXIF and/or FLXnMBSIF in the status interrupt register are set. After successful transmission, the FLXnTXIF flag in the status interrupt register is set. 1 = The corresponding message buffer interrupt is enabled 0 = The corresponding message buffer interrupt is disabled
28	FLXnWHTXM	Transmission mode This bit is used to select the transmission mode. For details on the transmission mode, see section 24.5.8 (3), Transmit Buffers. 1 = Single-shot mode 0 = Continuous mode
27	FLXnWHPPIT	Payload preamble indicator transmit This bit is used to control the state of the payload preamble indicator in transmit frames. If the bit is set in a static message buffer, the respective message buffer holds network management information. If the bit is set in a dynamic message buffer, the first two bytes of the payload segment may be used for message ID filtering by the receiver. Since the message ID filtering of received FlexRay frames is not supported by the FlexRay controller, it should be done by the user program. 1 = Payload preamble indicator set 0 = Payload preamble indicator not set
26	FLXnWHCFG	Message buffer configuration bit This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO, this bit is not evaluated. 1 = The corresponding buffer is configured as transmit buffer 0 = The corresponding buffer is configured as receive buffer

Table 24-82 FLXnWRHS1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>25, 24</td><td>FLXnWHCHB,
FLXnWHCHA</td><td>Channel filter control
The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

<table>
<tr><th>FLXnWHCHA</th><th>FLXnWHCHB</th><th>Transmit Buffer transmit frame on</th><th>Receive Buffer store frame received from</th></tr>
<tr><td>1</td><td>1</td><td>Both channels
(static segment only)</td><td>Channel A or B
(store first semantically valid frame, static segment only)</td></tr>
<tr><td>1</td><td>0</td><td>Channel A</td><td>Channel A</td></tr>
<tr><td>0</td><td>1</td><td>Channel B</td><td>Channel B</td></tr>
<tr><td>0</td><td>0</td><td>No transmission</td><td>Ignore frame</td></tr>
</table>

Note: If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as FLXnWHCHA = FLXnWHCHB = 0).</td></tr>
<tr><td>22 to 16</td><td>FLXnWHCYC
[6:0]</td><td>Cycle code
The 7-bit cycle code determines the cycle set used for cycle counter filtering. For details about the configuration of the cycle code, see section 24.5.7 (2), Cycle Counter Filtering.</td></tr>
<tr><td>10 to 0</td><td>FLXnWHFID
[10:00]</td><td>Frame ID
Frame ID of the selected message buffer. The frame ID defines the slot number for transmission/reception of the respective message. The message buffers with frame ID = 0 are considered as not valid.</td></tr>
</table>

(3) FLXnWRHS2 – Write Header Section Register 2

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0504_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	FLXnWHPLC[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnWHCRC[10:00]										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-83 FLXnWRHS2 Register Contents

Bit Position	Bit Name	Function
22 to 16	FLXnWHPLC [6:0]	Payload length configured Length of data section (number of 2-byte words) as configured by the host. During static segment, the static frame payload length as configured by FLXnMHDC.FLXnSFDL[6:0] defines the payload length for all static frames. If the payload length configured by FLXnWHPLC[6:0] is shorter than this value, padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is logical zero. For details, see section 24.5.8 (3), Transmit Buffers.
10 to 0	FLXnWHCRC [10:00]	Header CRC setting (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required. Transmit Buffer: The header CRC calculated and configured by the host. For calculation of the header CRC, the payload length of the frame send on the bus has to be considered. In static segment, the payload length of all frames is configured by FLXnMHDC.FLXnSFDL[6:0].

(4) FLXnWRHS3 – Write Header Section Register 3

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0508_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnWHDP[10:00]										
R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-84 FLXnWRHS3 Register Contents

Bit Position	Bit Name	Function
10 to 0	FLXnWHDP [10:00]	Data pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the message RAM.

(5) FLXnIBCM – Input Buffer Command Mask Register

This register configures how the message buffer in the message RAM selected by the input buffer command request register is updated. When IBF host and IBF shadow are swapped, mask bits FLXnLHSH, FLXnLDSH, and FLXnSTXRH are also swapped with bits FLXnLHSS, FLXnLDSS, and FLXnSTXRS to keep them attached to the respective input buffer transfer.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0510_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn STXRS	FLXn LDSS	FLXn LHSS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn STXRH	FLXn LDSH	FLXn LHSH
R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24-85 FLXnIBCM Register Contents

Bit Position	Bit Name	Function
18	FLXnSTXRS	Set transmission request shadow 1 = Set FLXnTXR flag, transmit buffer released for transmission (operation ongoing or finished) 0 = Reset FLXnTXR flag
17	FLXnLDSS	Load data section shadow 1 = Data section selected for transfer from input buffer to the message RAM (transfer ongoing or finished) 0 = Data section is not updated
16	FLXnLHSS	Load header section shadow 1 = Header section selected for transfer from input buffer to the message RAM (transfer ongoing or finished) 0 = Header section is not updated
2	FLXnSTXRH	Set transmission request host If this bit is set to 1, the FLXnTXR flag for the selected message buffer is set in the FLXnTXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode, the flag is cleared by the FlexRay controller after transmission has completed. FLXnTXR is evaluated for transmit buffers only. 1 = Set FLXnTXR flag, transmit buffer released for transmission 0 = Reset FLXnTXR flag
1	FLXnLDSH	Load data section host 1 = Data section selected for transfer from input buffer to the message RAM 0 = Data section is not updated
0	FLXnLHSH	Load header section host 1 = Header section selected for transfer from input buffer to the message RAM 0 = Header section is not updated

(6) FLXnIBCR – Input Buffer Command Request Register

When the host writes the number of a target message buffer in the message RAM to FLXnIBRH[6:0] in the input buffer command request register, IBF host and IBF shadow are swapped. In addition, the message buffer numbers stored under FLXnIBRH[6:0] and FLXnIBRS[6:0] are also swapped. (See also section 24.5.11 (2), Host Access to Message RAM, "Data transfer from input buffer to message RAM".)

With this write operation, the FLXnIBSYS bit in the input buffer command request register is set to 1. The message handler then starts to transfer the contents of IBF shadow to the message buffer in the message RAM selected by FLXnIBRS[6:0].

While the message handler transfers the data from IBF shadow to the target message buffer in the message RAM, the host is allowed to configure the next message in the IBF host. After the transfer between IBF shadow and the message RAM has completed, the FLXnIBSYS bit is set back to 0 and the next transfer to the message RAM can be started by the host by writing the respective target message buffer number to FLXnIBRH[6:0].

If a write access to FLXnIBRH[6:0] occurs while FLXnIBSYS is 1, FLXnIBSYH is set to 1. After completion of the ongoing data transfer from IBF shadow to the message RAM, IBF host and IBF shadow are swapped, and FLXnIBSYH is reset to 0. FLXnIBSYS remains to be 1, and the next transfer to the message RAM is started. In addition, the message buffer numbers stored under FLXnIBRH[6:0] and FLXnIBRS[6:0] are also swapped.

Any write access to an input buffer register while both FLXnIBSYS and FLXnIBSYH are set will cause the error flag FLXnEIR.FLXnIIBAE to be set. In this case, the input buffer will not be changed.

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0514_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnIBSYS	0	0	0	0	0	0	0	0	FLXnIBRS[6:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnIBSYH	0	0	0	0	0	0	0	0	FLXnIBRH[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-86 FLXnIBCR Register Contents

Bit Position	Bit Name	Function
31	FLXnIBSYS	Input buffer busy shadow Set to 1 after writing FLXnIBRH[6:0]. When the transfer between IBF shadow and the message RAM has completed, FLXnIBSYS is set back to 0. 1 = Transfer between IBF shadow and message RAM in progress 0 = Transfer between IBF shadow and message RAM completed
22 to 16	FLXnIBRS [6:0]	Input buffer request shadow Number of the target message buffer actually updated/lately updated. Valid values are 00_H to $7F_H$ (0 to 127).
15	FLXnIBSYH	Input buffer busy host Set to 1 by writing FLXnIBRH[6:0] while FLXnIBSYS is still 1. After the ongoing transfer between IBF shadow and the message RAM has completed, the FLXnIBSYH is set back to 0. 1 = Request while transfer between IBF shadow and message RAM in progress 0 = No request pending
6 to 0	FLXnIBRH [6:0]	Input buffer request host Selects the target message buffer in the message RAM for data transfer from input buffer. Valid values are 00_H to $7F_H$ (0 to 127).

24.4.11 Output Buffer

The output buffer has a double buffer structure consisting of output buffer host and output buffer shadow. It is used to read out message buffers from the message RAM. While the host can read from output buffer host, the message handler transfers the selected message buffer from message RAM to output buffer shadow. The data transfer between message RAM and output buffer (OBF) is described in section 24.5.11, Message Handling, "Data transfer from message RAM to output buffer".

(1) FLXnRDDSm – Read Data Section Register (m = 01 to 64)

This register holds the data words read from the data section of the addressed message buffer. The data words (DWn) are read from the message RAM in reception order from DW_1 (byte 0, byte 1) to DW_{PL} (PL = number of data words as defined by the payload length configured FLXnRDHS2.FLXnRHPLC[6:0]).

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0600_H to <FLXn_base> + $06FC_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLXnRDMDm[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnRDMDm[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-87 FLXnRDDSm Register Contents

Bit Position	Bit Name	Function
31 to 0	FLXnRD MDm[31:00]	Message data FLXnRDMDm[07:00] = DW_{2m-1}, $byte_{4m-4}$ FLXnRDMDm[15:08] = DW_{2m-1}, $byte_{4m-3}$ FLXnRDMDm[23:16] = DW_{2m}, $byte_{4m-2}$ FLXnRDMDm[31:24] = DW_{2m}, $byte_{4m-1}$

Note DW_{127} is located on FLXnRDDS64.FLXnRDMD64[15:0]. In this case, FLXnRDDS64.FLXnRDMD64[31:16] is not used (no valid data). The output buffer RAMs are initialized to 0 when leaving hard reset or by CHI command CLEAR_RAMS.

(2) FLXnRDHS1 – Read Header Section Register 1

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0700_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnRHMBI	FLXnRHTXM	FLXnRHPPIT	FLXnRHCFG	FLXnRHCHB	FLXnRHCHA	0	FLXnRHCYC[6:0]						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnRHFID[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Values as configured by the host via FLXnWRHS1:

Table 24-88 FLXnRDHS1 Register Contents

Bit Position	Bit Name	Function
29	FLXnRHMBI	Message buffer interrupt
28	FLXnRHTXM	Transmission mode
27	FLXnRHPPIT	Payload preamble indicator transmit
26	FLXnRHCFG	Message buffer direction configuration bit
25	FLXnRHCHB	Channel filter control
24	FLXnRHCHA	Channel filter control
22 to 16	FLXnRHCYC[6:0]	Cycle code
10 to 0	FLXnRHFID[10:00]	Frame ID

In the case when the message buffer read from the message RAM belongs to the receive FIFO, FLXnRHFID[10:00] holds the received frame ID, while FLXnRHCYC[6:0], FLXnRHCHA, FLXnRHCHB, FLXnRHCFG, FLXnRHPPIT, FLXnRHTXM, and FLXnRHMBI are reset to 0.

(3) FLXnRDHS2 – Read Header Section Register 2

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0704_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	FLXnRHPLR[6:0]							0	FLXnRHPLC[6:0]						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnRHCRC[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-89 FLXnRDHS2 Register Contents

Bit Position	Bit Name	Function
30 to 24	FLXnRHPLR [6:0]	Payload length received (vRF!Header!Length) Payload length value updated from received data frames in 2-byte units. (Exception: If message buffer belongs to the receive FIFO, FLXnRHPLR[6:0] is also updated from received null frames.)
22 to 16	FLXnRHPLC [6:0]	Payload length configured Length of data section (number of 2-byte units) as configured by the host.
10 to 0	FLXnRHCRC [10:00]	Header CRC (vRF!Header!HeaderCRC) Receive buffer: Header CRC updated from received frame Transmit buffer: Header CRC calculated and configured by the host

When a message is stored into a message buffer, the following behavior with respect to payload length received and payload length configured is implemented:

FLXnRHPLR[6:0] > FLXnRHPLC[6:0]:
The payload data stored in the message buffer is truncated to the configured payload length if the value in FLXnRHPLC[6:0] is even. If it is odd, the message buffer is truncated to FLXnRHPLC[6:0] + 1.

FLXnRHPLR[6:0] ≤ FLXnRHPLC[6:0]:
The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by FLXnRHPLC[6:0] are filled with undefined data.

FLXnRHPLR[6:0] = 0:
The message buffers data section is filled with undefined data.

FLXnRHPLC[6:0] = 0:
Message buffer has no data section configured. No data is stored into the message buffers data section.

Note The message RAM is organized in 4-byte words. When received data is stored into a message buffers data section, the number of 2-byte data words written into the message buffer is FLXnRHPLC[6:0] rounded to the next even value. FLXnRHPLC[6:0] should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.

(4) FLXnRDHS3 – Read Header Section Register 3

Access This register can be read in 32-bit units.

Address <FLXn_base> + 0708_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXn RHRES	FLXn RHPPI	FLXn RHNFI	FLXn RHSYN	FLXn RHSFI	FLXn RHRCI	0	0	FLXnRHRCC[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FLXnRHDP[10:00]										
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24-90 FLXnRDHS3 Register Contents

Bit Position	Bit Name	Function
29	FLXnRHRES	Reserved bit (vRF!Header!Reserved) Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.
28	FLXnRHPPI	Payload preamble indicator (vRF!Header!PPIndicator) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 1 = Static segment: Network management vector at the beginning of the payload dynamic segment: Message ID at the beginning of the payload 0 = The payload segment of the received frame does not contain a network management vector or a message ID
27	FLXnRHNFI	Null frame indicator (vRF!Header!NFIndicator) Is set to 1 after storage of the first received data frame. 1 = Received frame is not a null frame 0 = Received frame is a null frame
26	FLXnRHSYN	Sync frame indicator (vRF!Header!SyFIndicator) A sync frame is marked by the sync frame indicator. 1 = The received frame is a sync frame 0 = No sync frame received
25	FLXnRHSFI	Startup frame indicator (vRF!Header!SuFIndicator) A startup frame is marked by the startup frame indicator. 1 = The received frame is a startup frame 0 = No startup frame received
24	FLXnRHRCI	Received on channel indicator (vSS!Channel) Indicates the channel from which the received data frame was taken to update the respective receive buffer. 1 = Frame received on channel A 0 = Frame received on channel B
21 to 16	FLXnRHRCC [5:0]	Receive cycle count (vRF!Header!CycleCount) Cycle counter value updated from received frame.
10 to 0	FLXnRHDP [10:00]	Data pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the message RAM.

Note Header 3 is updated by data frames only.

(5) FLXnMBS – Message Buffer Status Register

The FlexRay controller updates the message buffer status of the channel assigned to the messaged buffer by the end of the slot that follows the slot assigned to the message buffer. The flags are updated only when the FlexRay controller is in the NORMAL_ACTIVE or NORMAL_PASSIVE state. If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to 0. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated. The message buffer status is updated only when the slot counter reaches the configured frame ID and when the cycle counter filter matches. When the host updates a message buffer via input buffer, all MBS flags are reset to 0 independent of which FLXnIBCM bits are set or not. For details about receive/transmit filtering, see section 24.5.7, Filtering and Masking; section 24.5.8, Transmit Process; and section 24.5.9, Receive Process. Whenever the message handler changes one of the flags FLXnMBSVFRA, FLXnMBSVFRB, FLXnMBSSEOA, FLXnMBSSEOB, FLXnMBSCEOA, FLXnMBSCEOB, FLXnMBSSVOA, FLXnMBSSVOB, FLXnMBSTCIA, FLXnMBSTCIB, FLXnMBSESA, FLXnMBSESB, FLXnMBSMLST, FLXnMBSFTA, and FLXnMBSFTB, the respective message buffer's MBC flag in registers FLXnMBSC1/2/3/4 is set.

Access This register can be read in 32-bit units.

Address <FLXn_base> + $070C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	FLXnMBSRESS	FLXnMBSPPIS	FLXnMBSNFIS	FLXnMBSSYNS	FLXnMBSSFIS	FLXnMBSRCIS	0	0	FLXnMBSCCS[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXnMBSFTB	FLXnMBSFTA	0	FLXnMBSMLST	FLXnMBSESB	FLXnMBSESA	FLXnMBSTCIB	FLXnMBSTCIA	FLXnMBSSVOB	FLXnMBSSVOA	FLXnMBSCEOB	FLXnMBSCEOA	FLXnMBSSEOB	FLXnMBSSEOA	FLXnMBSVFRB	FLXnMBSVFRA
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note For receive buffers (FLXnRHCFG = 0), the following status bits are updated by both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flags have no meaning and should be ignored.

Table 24-91 FLXnMBS Register Contents – Receive Buffers

Bit Position	Bit Name	Function
29	FLXnMBS RESS	Reserved bit status (vRF!Header!Reserved) Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.
28	FLXnMBS PPIS	Payload preamble indicator status (vRF!Header!PPIndicator) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 1 = Static segment: Network management vector at the beginning of the payload dynamic segment: Message ID at the beginning of the payload 0 = The payload segment of the received frame does not contain a network management vector or a message ID
27	FLXnMBS NFIS	Null frame indicator status (vRF!Header!NFIndicator) If set to 0, the payload segment of the received frame contains no usable data. 1 = Received frame is not a null frame 0 = Received frame is a null frame
26	FLXnMBS SYNS	Sync frame indicator status (vRF!Header!SyFIndicator) A sync frame is marked by the sync frame indicator. 1 = The received frame is a sync frame 0 = No sync frame received
25	FLXnMBS SFIS	Startup frame indicator status (vRF!Header!SuFIndicator) A startup frame is marked by the startup frame indicator. 1 = The received frame is a startup frame 0 = No startup frame received
24	FLXnMBS RCIS	Received on channel indicator status (vSS!Channel) Indicates the channel on which the frame was received. 1 = Frame received on channel A 0 = Frame received on channel B

Note The FlexRay protocol specification requires that FLXnMBSFTA and FLXnMBSFTB can only be reset by the host. Therefore, the cycle count status FLXnMBSCCS[5:0] for these bits is only valid for the cycle where the bits are set to 1.

Table 24-92 FLXnMBS Register Contents – Cycle Count Status

Bit Position	Bit Name	Function
21 to 16	FLXnMBS CCS[5:0]	Cycle count status Actual cycle count when status was updated.

Table 24-93 FLXnMBS Register Contents – Transmit & Receive Buffers (1/2)

Bit Position	Bit Name	Function
15	FLXnMBS FTB	Frame transmitted on channel B Indicates that this node has transmitted a data frame in the configured slot on channel B. 1 = Data frame transmitted on channel B 0 = No data frame transmitted on channel B
14	FLXnMBS FTA	Frame transmitted on channel A Indicates that this node has transmitted a data frame in the configured slot on channel A. 1 = Data frame transmitted on channel A 0 = No data frame transmitted on channel A
12	FLXnMBS MLST	Message lost This flag is set in the case when the host did not read the message before the message buffer was updated from a received data frame. Not affected by reception of null frames except for message buffers belonging to the receive FIFO. This flag is reset when a new message is stored in the message buffer after the ND flag of the message buffer is reset by the host operation of writing data to the message buffer via IBF or of reading the message buffer via OBF. 1 = Unprocessed message was overwritten 0 = No message lost
11	FLXnMBS ESB	Empty slot channel B In an empty slot, there is no activity on the bus i.e. no frame transmission detected. This status is checked for message buffers assigned as static or dynamic buffers. 1 = No bus activity detected in the configured slot on channel B 0 = Bus activity detected in the configured slot on channel B
10	FLXnMBS ESA	Empty slot channel A In an empty slot, there is no activity on the bus i.e. no frame transmission detected. This status is checked for message buffers assigned as static or dynamic buffers. 1 = No bus activity detected in the configured slot on channel A 0 = Bus activity detected in the configured slot on channel A
9	FLXnMBS TCIB	Transmission conflict indication channel B (vSS!TxConflictB) A transmission conflict indication is set if a transmission conflict has occurred on channel B. 1 = Transmission conflict occurred on channel B 0 = No transmission conflict occurred on channel B
8	FLXnMBS TCIA	Transmission conflict indication channel A (vSS!TxConflictA) A transmission conflict indication is set if a transmission conflict has occurred on channel A. 1 = Transmission conflict occurred on channel A 0 = No transmission conflict occurred on channel A
7	FLXnMBS SVOB	Slot boundary violation observed on channel B (vSS!BViolationB) A slot boundary violation was observed on channel B, i.e. the channel has become active at the start or at the end of the configured slot. 1 = Slot boundary violation observed on channel B 0 = No slot boundary violation observed on channel B
6	FLXnMBS SVOA	Slot boundary violation observed on channel A (vSS!BViolationA) A slot boundary violation was observed on channel A, i.e. the channel has become active at the start or at the end of the configured slot. 1 = Slot boundary violation observed on channel A 0 = No slot boundary violation observed on channel A

Table 24-93 FLXnMBS Register Contents – Transmit & Receive Buffers (2/2)

Bit Position	Bit Name	Function
5	FLXnMBS CEOB	Content error observed on channel B (vSS!ContentErrorB) A content error was observed in the configured slot on channel B. 1 = Content error observed on channel B 0 = No content error observed on channel B
4	FLXnMBS CEOA	Content error observed on channel A (vSS!ContentErrorA) A content error was observed in the configured slot on channel A. 1 = Content error observed on channel A 0 = No content error observed on channel A
3	FLXnMBS SEOB	Syntax error observed on channel B (vSS!SyntaxErrorB) A syntax error was observed in the configured slot on channel B. 1 = Syntax error observed on channel B 0 = No syntax error observed on channel B
2	FLXnMBS SEOA	Syntax error observed on channel A (vSS!SyntaxErrorA) A syntax error was observed in the configured slot on channel A. 1 = Syntax error observed on channel A 0 = No syntax error observed on channel A
1	FLXnMBS VFRB	Valid frame received on channel B A valid frame indication is set if a valid frame was received on channel B. 1 = Valid frame received on channel B 0 = No valid frame received on channel B
0	FLXnMBS VFRA	Valid frame received on channel A A valid frame indication is set if a valid frame was received on channel A. 1 = Valid frame received on channel A 0 = No valid frame received on channel A

(6) FLXnOBCM – Output Buffer Command Mask Register

This register configures how the output buffer is updated from the message buffer in the message RAM selected by FLXnOBCR.FLXnOBRS[6:0]. Mask bits FLXnRDSS and FLXnRHSS are copied to the register internal storage when a message RAM transfer is requested by FLXnOBCR.FLXnREG. When OBF host and OBF shadow are swapped, mask bits FLXnRDSH and FLXnRHSH are swapped with the register internal storage to keep them attached to the respective output buffer transfer. The data transfer between output buffer and message RAM is described in detail in section 24.5.11, Message Handling, "Data transfer from message RAM to output buffer".

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0710_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn RDSH	FLXn RHSH
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLXn RDSS	FLXn RHSS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24-94 FLXnOBCM Register Contents

Bit Position	Bit Name	Function
17	FLXnRDSH	Read data section host 1 = Data section selected for transfer from message RAM to output buffer 0 = Data section is not read
16	FLXnRHSH	Read header section host 1 = Header section selected for transfer from message RAM to output buffer 0 = Header section is not read
1	FLXnRDSS	Read data section shadow 1 = Data section selected for transfer from message RAM to output buffer 0 = Data section is not read
0	FLXnRHSS	Read header section shadow 1 = Header section selected for transfer from message RAM to output buffer 0 = Header section is not read

Note After the transfer of the header section from the message RAM to OBF shadow has completed, the message buffer status changed flag FLXnMBC of the selected message buffer in the FLXnMBSC1/2/3/4 registers is cleared. After the transfer of the data section from the message RAM to OBF shadow has completed, the new data flag ND of the selected message buffer in the FLXnNDAT1/2/3/4 registers is cleared.

(7) FLXnOBCR – Output Buffer Command Request Register

After setting bit FLXnREQ to 1 while FLXnOBSYS is 0, FLXnOBSYS is automatically set to 1, FLXnOBRS[6:0] is copied to the register internal storage, mask bits FLXnOBCM.FLXnRDSS and FLXnOBCM.FLXnRHSS are copied to register FLXnOBCM internal storage, and the transfer of the message buffer selected by FLXnOBRS[6:0] from the message RAM to OBF shadow is started. When the transfer between the message RAM and OBF shadow has completed, this is signaled by setting FLXnOBSYS back to 0.

By setting bit FLXnVIEW to 1 while FLXnOBSYS is 0, OBF host and OBF shadow are swapped. Additionally, mask bits FLXnOBCM.FLXnRDSH and FLXnOBCM.FLXnRHSH are swapped with the register FLXnOBCM internal storage to keep them attached to the respective output buffer transfer. FLXnOBRH[6:0] signals the number of the message buffer currently accessible by the host.

If bits FLXnREQ and FLXnVIEW are set to 1 with the same write access while FLXnOBSYS is 0, FLXnOBSYS is automatically set to 1 and OBF shadow and OBF host are swapped. Additionally, mask bits FLXnOBCM.FLXnRDSH and FLXnOBCM.FLXnRHSH are swapped with the registers internal storage to keep them attached to the respective output buffer transfer. Afterwards, FLXnOBRS[6:0] is copied to the register internal storage, and the transfer of the selected message buffer from the message RAM to OBF shadow is started. While the transfer is ongoing, the host can read the message buffer transferred by the previous transfer from OBF host. When the current transfer between message RAM and OBF shadow has completed, this is signaled by setting FLXnOBSYS back to 0.

Any write access to FLXnOBCR[15:8] while FLXnOBSYS is set will cause the error flag FLXnEIR.FLXnIOBAE to be set. In this case, the output buffer will not be changed.

The data transfer between output buffer and message RAM is described in section 24.5.11, Message Handling, "Data transfer from message RAM to output buffer".

Access This register can be read/written in 32-bit units.

Address <FLXn_base> + 0714_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	FLXnOBRH[6:0]						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXn OBSYS	0	0	0	0	0	FLXn REQ	FLXn VIEW	0	FLXnOBRS[6:0]						
R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24-95 FLXnOBCR Register Contents

Bit Position	Bit Name	Function
22 to 16	FLXnOBRH [6:0]	Output buffer request host Number of message buffer currently accessible by the host via FLXnRDHS[1...3], FLXnMBS, and FLXnRDDS[1...64]. By writing FLXnVIEW to 1, OBF shadow and OBF host are swapped and the transferred message buffer is accessible by the host. Valid values are 00_H to $7F_H$ (0 to 127).
15	FLXnOBSYS	Output buffer busy shadow This bit is set to 1 after bit FLXnVIEW is set. When the transfer between the message RAM and OBF shadow has completed, FLXnOBSYS is set back to 0. 1 = Transfer between message RAM and OBF shadow in progress 0 = No transfer in progress
9	FLXnREQ	Request message RAM transfer This bit requests transfer of message buffer addressed by FLXnOBRS[6:0] from message RAM to OBF shadow. Only writeable while FLXnOBSYS = 0. 1 = Transfer to OBF shadow requested 0 = No request
8	FLXnVIEW	View shadow buffer Toggles between OBF shadow and OBF host. Only writeable while FLXnOBSYS = 0. 1 = Swap OBF shadow and OBF host 0 = No action
6 to 0	FLXnOBRS [6:0]	Output buffer request shadow Number of source message buffer to be transferred from the message RAM to OBF Shadow. Valid values are 00_H to $7F_H$ (0 to 127). If the number of the first message buffer of the receive FIFO is written to this register, the message handler transfers the message buffer addressed by the GET Index Register (GIDX: See section 24.5.10, FIFO Function.) to OBF shadow.

24.5 Functional Description

This section describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay Protocol Specification v2.1.

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

24.5.1 Communication Cycle

A communication cycle in FlexRay consists of the following elements:

- Static segment
- Dynamic segment (optional)
- Symbol window (optional)
- Network idle time (NIT)

Static segment, dynamic segment, and symbol window form the network communication time (NCT). For each communication channel, the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid, which means that they use the same synchronized macrotick.

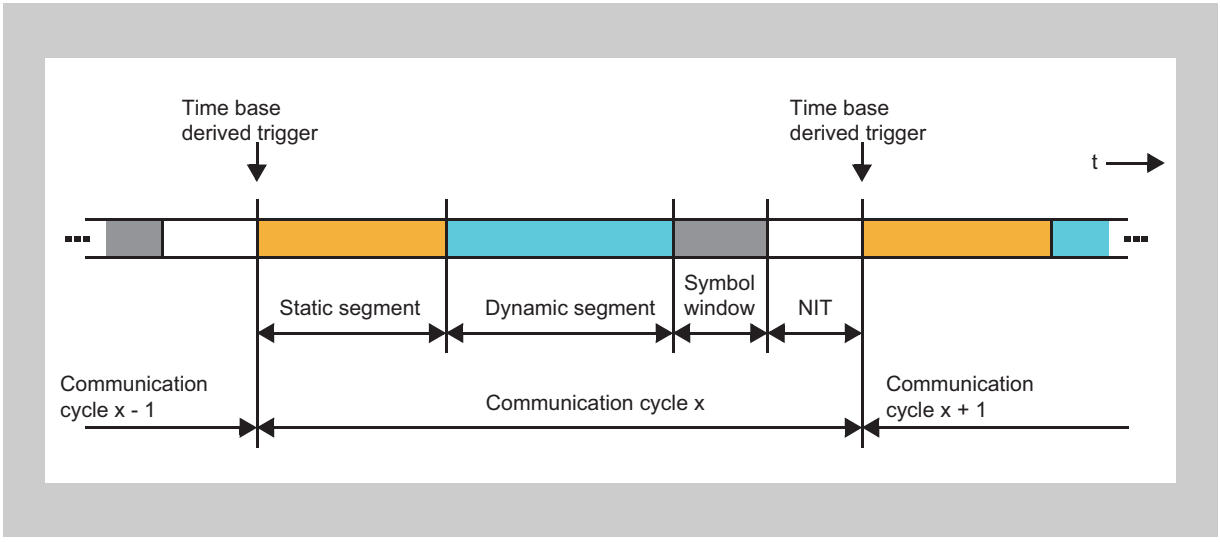


Figure 24-2 Structure of Communication Cycle

(1) Static Segment

The static segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters:
Number of static slots FLXnGTUC07.FLXnNSS[9:0],
static slot length FLXnGTUC07.FLXnSSL[9:0],
payload length static FLXnMHDC.FLXnSFDL[6:0], and
action point offset FLXnGTUC09.FLXnAPO[5:0].

(2) Dynamic Segment

The dynamic segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters:
Number of minislots FLXnGTUC08.FLXnNMS[12:00],
minislot length FLXnGTUC08.FLXnMSL[5:0],
minislot action point offset FLXnGTUC09.FLXnMAPO[4:0], and
start of latest transmit (last minislot) FLXnMHDC.FLXnSLT[12:00].

(3) Symbol Window

During the symbol window, only one media access test symbol (MTS) can be transmitted per channel. MTS symbols are sent in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Sending single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters:
Symbol window action point offset FLXnGTUC09.FLXnAPO[4:0] (same as for static slots) and
network idle time start FLXnGTUC04.FLXnNIT[13:00].

(4) Network Idle Time (NIT)

During network idle time, the FlexRay controller has to perform the following tasks:

- Calculating clock correction terms (offset and rate)
- Distributing offset correction over multiple macroticks after offset correction start
- Performing cluster cycle related tasks

Parameters:
Network idle time start FLXnGTUC04.FLXnNIT[13:00] and
offset correction start FLXnGTUC04.FLXnOCS[13:00].

(5) Configuration of NIT Start and Offset Correction Start

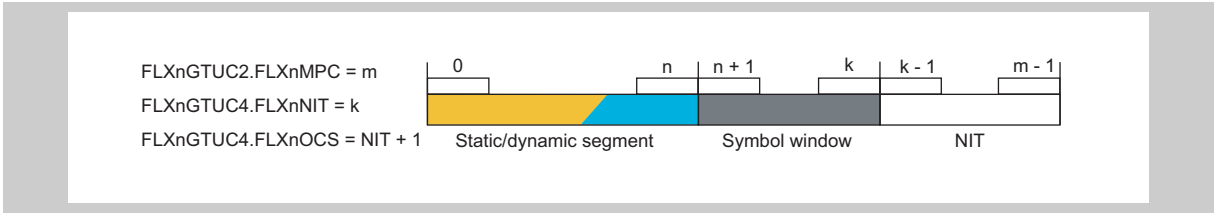


Figure 24-3 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming FLXnGTUC02.FLXnMPC = m.

The static/dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset + dynamic segment length - 1 MT
n = gNumberOfStaticSlots • gdStaticSlot + dynamic segment offset + gNumberOfMinislots • gdMinislot - 1 MT

The static segment length is configured by FLXnGTUC07.FLXnSSL and FLXnGTUC07.FLXnNSS.
The dynamic segment length is configured by FLXnGTUC08.FLXnMSL and FLXnGTUC08.FLXnNMS.

The dynamic segment offset is:
If gdActionPointOffset ≤ gdMinislotActionPointOffset:
dynamic segment offset = 0 MT

If gdActionPointOffset > gdMinislotActionPointOffset:
dynamic segment offset = gdActionPointOffset - gdMinislotActionPointOffset

The NIT starts with macrotick k + 1 and ends with the last macrotick of cycle m - 1. It has to be configured by setting FLXnGTUC04.FLXnNIT = k.

For the FlexRay controller, the offset correction start is required to be FLXnGTUC04.FLXnOCS ≥ FLXnGTUC04.FLXnNIT + 1 = k + 1.

The length of symbol window results from the number of macroticks between the end of the static/dynamic segment and the beginning of the NIT. It can be calculated by k - n.

24.5.2 Communication Modes

The FlexRay Protocol Specification v2.1 defines the time-triggered distributed (TT-D) mode.

(1) Time-Triggered Distributed (TT-D)

In TT-D mode, the following configurations are possible:

- Pure static: minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes need to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

24.5.3 Clock Synchronization

In TT-D mode, a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

(1) Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time, even though each node maintains its own view of it. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other nodes operating with independent clock mechanisms. The global time is a vector of two values: the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick (MT) = Basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (µT)
- Cycle length = Duration of a communication cycle in units of macroticks (MT)

(2) Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore, microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (µT).

Node specific:

- Oscillator clock -> prescaler -> microtick (µT)
- µT = Basic unit of time measurement in a FlexRay controller, clock correction is done in units of µTs
- Cycle counter + macrotick counter = nodes local view of the global time

(3) Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two channel cluster, a sync node has to send its sync frame on both channels.

For synchronization in FlexRay, the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXnGTUC02.FLXnSNM[3:0]) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization, the time difference between expected and observed arrival time of sync frames received during the static segment, valid on both channels (two-channel cluster), is measured. The calculation of correction terms is done during NIT (offset: every cycle, rate: odd cycle) by using a FTA/FTM algorithm. For details, see the FlexRay Protocol Specification v2.1, section 8.

Offset (phase) correction

- Only deviation values measured and stored in the current cycle is used.
- For a two channel node, the smaller value will be taken.
- Calculation is done during NIT of every communication cycle.
- Offset correction value is calculated in even cycles used for error checking only.
- Checked against limit values
- Correction value is a signed integer number of µT.
- Correction in odd numbered cycles is distributed over the macroticks beginning at offset correction start and ending at the cycle end (end of NIT), according to which the starting point of the next cycle of the node is shifted (MTs lengthened/shortened).

Rate (frequency) correction

- Pairs of deviation values is measured and stored in even/odd cycle pair is used.
- For a two channel node, the average of the differences from the two channels is used.
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value.
- Checked against limit values
- Correction value is a signed integer number of µT.
- The correction value is applied to multiple macroticks comprising the next even/odd cycle pair (MTs lengthened/shortened).

Sync frame transmission

Sync frame transmission is possible only from buffers 0 and 1. Message buffer 1 can be used for sync frame transmission when sync frames have different payloads on the two channels. In this case, bit FLXnMRC.FLXnSPLM has to be set to 1.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in the DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames, FLXnSUCC1.FLXnTXSY must be set to 1.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset/rate correction value is a signed integer.
- External offset/rate correction value is added to calculated offset/rate correction value.
- Aggregated offset/rate correction term (external + internal) is not checked against configured limits.

24.5.4 Error Handling

The implemented error handling concept is intended to ensure that, even if a lower layer protocol error occurs in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the FlexRay controller to resume normal operation. A change of the error handling state will set FLXnEIR.FLXnPEMCE and may trigger an interrupt to the host if enabled. The current error mode is indicated by FLXnCCEV.FLXnERRM[1:0].

Table 24-96 Error Modes of the POC (Degradation Model)

Error Mode	Activity
ACTIVE	Full operation, State: NORMAL_ACTIVE The FlexRay controller is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXnEIR and FLXnSIR.
PASSIVE	Reduced operation, State: NORMAL_PASSIVE, FlexRay controller self rescue allowed The FlexRay controller stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. There is no effect on the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXnEIR and FLXnSIR.
COMM_HALT	Operation halted, State: HALT, self rescue of RlexRay controller not allowed The FlexRay controller stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host still has access to error and status information by reading the error and status interrupt flags from registers FLXnEIR and FLXnSIR. The bus drivers are disabled.

(1) Clock Correction Failed Counter

When the clock correction failed counter reaches the "maximum without clock correction passive" limit defined by FLXnSUCC3.FLXnWCP[3:0], the POC transits from the NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the "maximum without clock correction fatal" limit defined by FLXnSUCC3.FLXnWCF[3:0], it transits from the NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The clock correction failed counter FLXnCCEV.FLXnCCFC[3:0] allows the host to monitor the duration of the inability of a node to compute clock correction terms after the FlexRay controller passes protocol startup phase. It will be incremented by 1 at the end of any odd communication cycle during which either the missing offset correction FLXnSFS.FLXnMOCS or the missing rate correction FLXnSFS.FLXnMRCS flag is set.

The clock correction failed counter is reset to 0 at the end of an odd communication cycle if neither the missing offset correction FLXnSFS.FLXnMOCS nor the missing rate correction FLXnSFS.FLXnMRCS flag is set.

The clock correction failed counter stops incrementing when the "maximum without clock correction fatal" value FLXnSUCC3.FLXnWCF[3:0] is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to 0). The clock correction failed counter is initialized to 0 when the FlexRay controller enters the READY state or when the NORMAL_ACTIVE state is entered.

Note The transition to the HALT state is prevented if FLXnSUCC1.FLXnHCSE is not set.

(2) Passive to Active Counter

The passive to active counter controls the transition of the POC from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. FLXnSUCC1.FLXnPTA[4:0] defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the FlexRay controller is allowed to transit from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. If FLXnSUCC1.FLXnPTA[4:0] is set to 0, the FlexRay controller is not allowed to transit from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state.

(3) HALT Command

When the host wants to stop FlexRay communication of the local node, it can bring the FlexRay controller into the HALT state by asserting the HALT command. This can be done by writing FLXnSUCC1.FLXnCMD[3:0] = 0110. In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to the HALT state took place can be read from FLXnCCSV.FLXnPSL[5:0].

When called in the NORMAL_ACTIVE or NORMAL_PASSIVE state, the POC transits to the HALT state at the end of the current cycle. When called in any other state, FLXnSUCC1.FLXnCMD[3:0] will be reset to 0000 (= command_not_accepted) and bit FLXnEIR.FLXnCNAE is set to 1. If enabled, an interrupt to the host is generated.

(4) FREEZE Command

When the host detects a severe error condition, it can bring the FlexRay controller into the HALT state by asserting the FREEZE command. This can be done by writing FLXnSUCC1.FLXnCMD[3:0] = 0111. The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to the HALT state took place can be read from FLXnCCSV.FLXnPSL[5:0].

24.5.5 Communication Controller States

(1) Communication Controller State Diagram

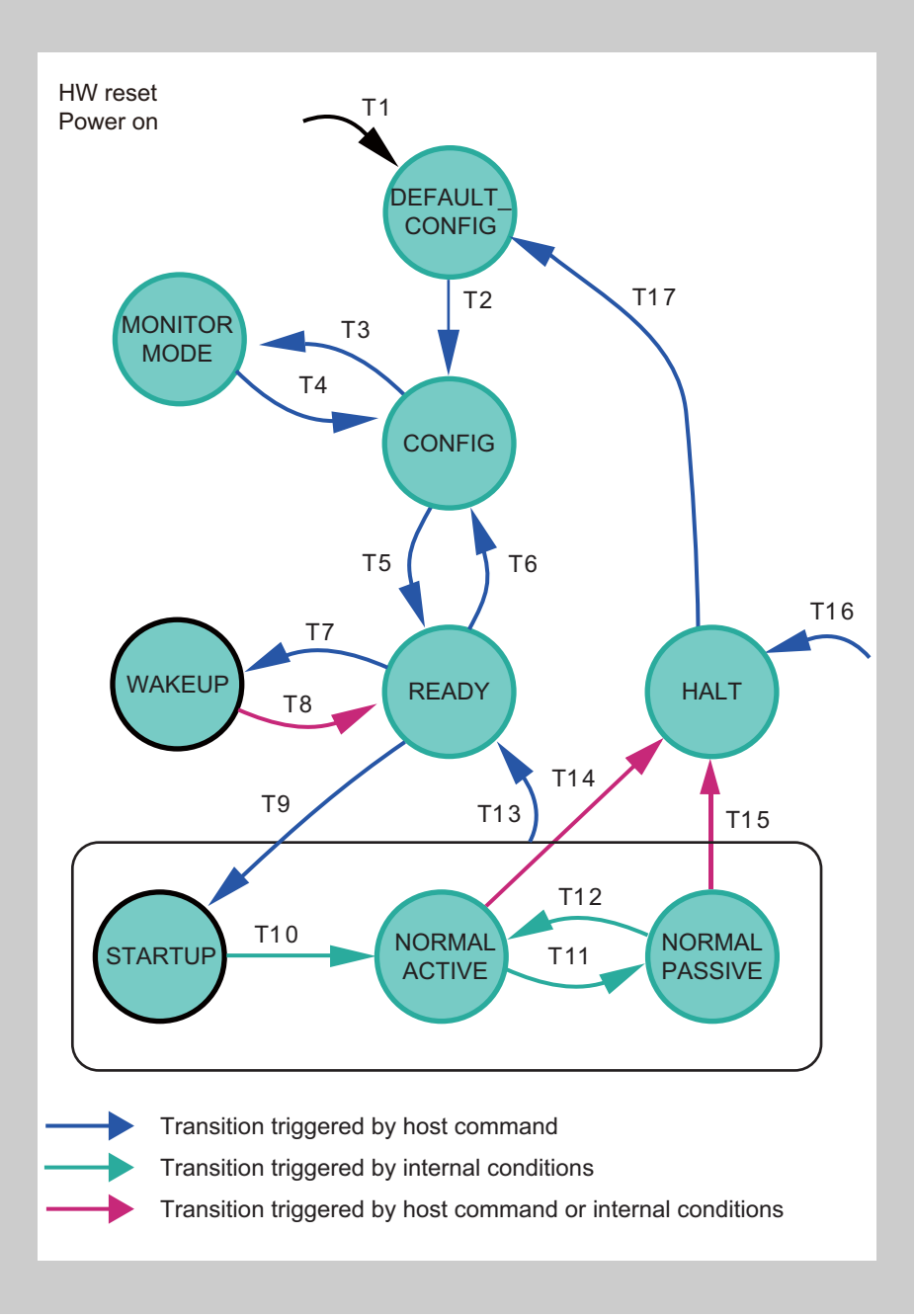


Figure 24-4 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by eray_reset, eray_rxd1, and eray_rxd2, by the POC state machine, and by the CHI command vector FLXnSUCC1.FLXnCMD[3:0].

The FlexRay controller exits from all states to the HALT state after application of the FREEZE command (FLXnSUCC1.FLXnCMD[3:0] = 0111).

Table 24-97 State Transitions of FlexRay Controller Overall State Machine

T#	Condition	From	To
1	Power on, reset	All states	DEFAULT_CONFIG
2	Command CONFIG, FLXnSUCC1.FLXnCMD[3:0] = 0001	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command MONITOR_MODE, FLXnSUCC1.FLXnCMD[3:0] = 1011	CONFIG	MONITOR_MODE
4	Command CONFIG, FLXnSUCC1.FLXnCMD[3:0] = 0001	MONITOR_MODE	CONFIG
5	Unlock sequence followed by command READY, FLXnSUCC1.FLXnCMD[3:0] = 0010	CONFIG	READY
6	Command CONFIG, FLXnSUCC1.FLXnCMD[3:0] = 0001	READY	CONFIG
7	Command WAKEUP, FLXnSUCC1.FLXnCMD[3:0] = 0011	READY	WAKEUP
8	Complete, non-aborted transmission of wakeup pattern or received WUP or received frame header or command READY, FLXnSUCC1.FLXnCMD[3:0] = 0010	WAKEUP	READY
9	Command RUN, FLXnSUCC1.FLXnCMD[3:0] = 0100	READY	STARTUP
10	Successful startup	STARTUP	NORMAL_ACTIVE
11	Clock correction failed counter reached “maximum without clock correction passive” limit configured by FLXnSUCC3.FLXnWCP[3:0]	NORMAL_ACTIVE	NORMAL_PASSIVE
12	Number of valid correction terms reached the “passive to active” limit configured by FLXnSUCC1.FLXnPTA[4:0]	NORMAL_PASSIVE	NORMAL_ACTIVE
13	Command READY, FLXnSUCC1.FLXnCMD[3:0] = 0010	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
14	Clock correction failed counter reached “maximum without clock correction fatal” limit configured by FLXnSUCC3.FLXnWCF[3:0] and bit FLXnSUCC1.FLXnHCSE set to 1 or command HALT, FLXnSUCC1.FLXnCMD[3:0] = 0110	NORMAL_ACTIVE	HALT
15	Clock correction failed counter reached “maximum without clock correction fatal” limit configured by FLXnSUCC3.FLXnWCF[3:0] and bit FLXnSUCC1.FLXnHCSE set to 1 or command HALT, FLXnSUCC1.FLXnCMD[3:0] = 0110	NORMAL_PASSIVE	HALT
16	Command FREEZE, FLXnSUCC1.FLXnCMD[3:0] = 0111	All states	HALT
17	Command CONFIG, FLXnSUCC1.FLXnCMD[3:0] = 0001	HALT	DEFAULT_CONFIG

(2) DEFAULT_CONFIG State

In the DEFAULT_CONFIG state, the FlexRay controller is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The FlexRay controller enters this state

- when leaving hardware reset, and
- when exiting from the HALT state.

To leave the DEFAULT_CONFIG state, the host has to write FLXnCMD[3:0] = 0001 in the SUC configuration register 1. The FlexRay controller then transits to the CONFIG state.

(3) CONFIG State

In the CONFIG state, the FlexRay controller is stopped. All configuration registers are accessible and the signals to the physical layer are in their inactive state. This state is used to initialize the configuration of the FlexRay controller.

The FlexRay controller enters this state

- when exiting from the DEFAULT_CONFIG state, and
- when exiting from the MONITOR_MODE or READY state.

When this state has been entered via the HALT and DEFAULT_CONFIG state, the host can analyze status information and configuration. Before leaving the CONFIG state, the host has to assure that the configuration is fault-free.

To leave the CONFIG state, the host has to perform the unlock sequence as described in section 24.4.3 (3), FLXnLCK – Lock Register. Directly after unlocking the CONFIG state, the host has to write FLXnSUCC1.FLXnCMD[3:0] to enter the next state.

Note Status bits FLXnMHDS[14:00], registers FLXnTXRQ1/2/3/4, and status data stored in the message RAM are not affected by the transition of the POC from the CONFIG state to the READY state.

When the FlexRay controller is in the CONFIG state, it is also possible to bring the FlexRay controller into a power saving mode by halting the module clocks (eray_sclk, eray_bclk). To do this, the host has to assure that all message RAM transfers have finished before turning off the clocks.

(4) MONITOR_MODE State

After unlocking the CONFIG state and writing FLXnSUCC1.FLXnCMD[3:0] = 1011, the FlexRay controller enters MONITOR_MODE. In this mode, the FlexRay controller is able to receive FlexRay frames and to detect wakeup pattern. The temporal integrity of received frames is not checked, and therefore cycle counter filtering is not supported. This mode can be used for debugging purposes in cases such that startup of a FlexRay network fails. After writing FLXnSUCC1.FLXnCMD[3:0] = 0001, the FlexRay controller transits back to the CONFIG state.

In the MONITOR_MODE state, the mechanism to pick the first valid frame is disabled. This means that a receive message buffer may only be configured to receive on one channel. Received frames are stored into message buffers according to frame ID and receive channel. Null frames are handled like data frames. After frame reception, only status bits FLXnMBS.FLXnMBSVFRA, FLXnMBS.FLXnMBSVFRB, FLXnMBS.FLXnMBSMLST, FLXnMBS.FLXnMBSRCIS, FLXnMBS.FLXnMBSSFIS,

FLXnMBS.FLXnMBSSYNS, FLXnMBS.FLXnMBSNFIS, FLXnMBS.FLXnMBSPPIS, and FLXnMBS.FLXnMBSRESS have valid values.

In the MONITOR_MODE state, the FlexRay controller cannot discriminate between CAS and MTS. When one of these symbols is received on one or both channels, FLXnSIR.FLXnMTSAF and FLXnSIR.FLXnMTSBF are set to 1. FLXnSIR.FLXnCASF does not function in the MONITOR_MODE state.

(5) READY State

After unlocking the CONFIG state and writing FLXnSUCC1.FLXnCMD[3:0] = 0010, the FlexRay controller enters the READY state. From this state, the FlexRay controller can transit to the WAKEUP state and perform a cluster wakeup or to the STARTUP state to perform a coldstart or to integrate into a running cluster.

The FlexRay controller enters this state

- when exiting from the CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing FLXnSUCC1.FLXnCMD[3:0] = 0010 (READY command).

The FlexRay controller exits from this state

- to the CONFIG state by writing FLXnSUCC1.FLXnCMD[3:0] = 0001 (CONFIG command),
- to the WAKEUP state by writing FLXnSUCC1.FLXnCMD[3:0] = 0011 (WAKEUP command), or
- to the STARTUP state by writing FLXnSUCC1.FLXnCMD[3:0] = 0100 (RUN command).

Internal counters and the CC status flags are reset when the FlexRay controller enters the STARTUP state.

Note Status bits FLXnMHDS[14:00], registers FLXnTXRQ1/2/3/4, and status data stored in the message RAM are not affected by the transition of the POC from the READY state to the STARTUP state.

(6) WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay controller. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay Protocol Specification v2.1, section 7.1.

The FlexRay controller enters this state

- when exiting from the READY state by writing FLXnSUCC1.FLXnCMD[3:0] = 0011 (WAKEUP command).

The FlexRay controller exits from this state to the READY state

- after complete non-aborted transmission of wakeup pattern,
- after WUP reception,
- after detecting a WUP collision,
- after reception of a frame header, and
- by writing FLXnSUCC1.FLXnCMD[3:0] = 0010 (READY command).

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a

wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the FlexRay controller and configures bus guardian (if available) and the FlexRay controller to perform the cluster wakeup. The FlexRay controller provides to the host the ability to transmit a special wakeup pattern on each of its available channels separately. The FlexRay controller needs to recognize the wakeup pattern only during the WAKEUP state.

Wakeup can be performed on only one channel at a time. The host has to configure the wakeup channel while the FlexRay controller is in the CONFIG state by writing FLXnSUCC1.FLXnWUCS. The FlexRay controller ensures that ongoing communication on this channel is not disturbed. The FlexRay controller cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation by allowing only one node to transmit the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup, the FlexRay controller returns to the READY state and signals the change of the wakeup status to the host by setting flag FLXnSIR.FLXnWSTF. The wakeup status vector can be read from FLXnCCSV.FLXnWSV[2:0]. If a valid wakeup pattern was received, either flag FLXnSIR.FLXnWUPAF or flag FLXnSIR.FLXnWUPBF is set.

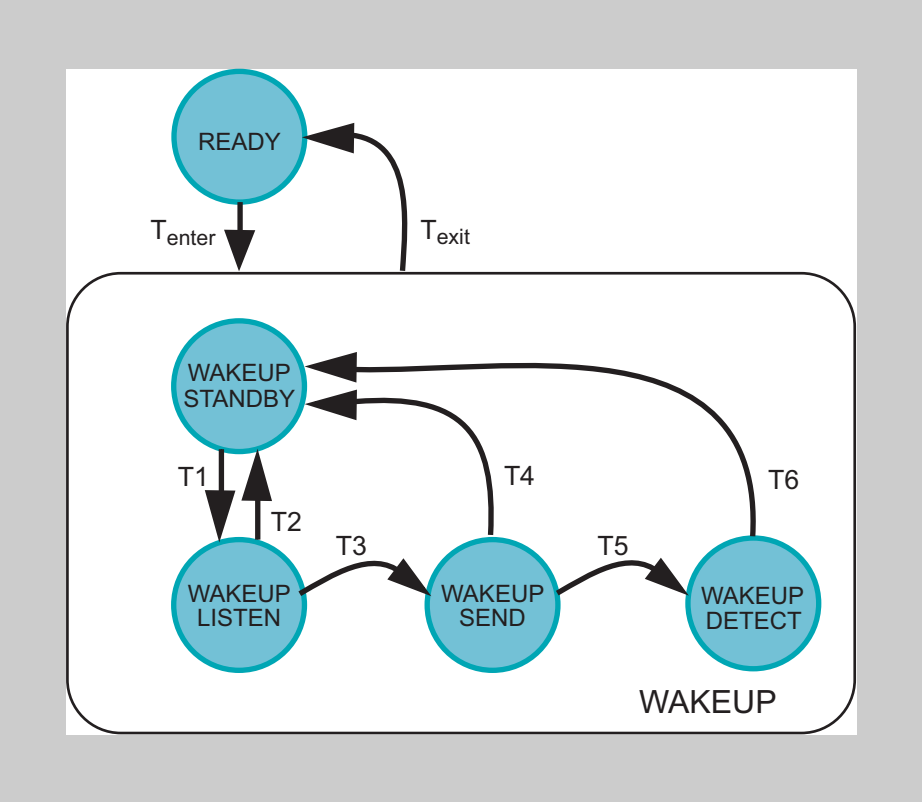


Figure 24-5 Structure of POC State WAKEUP

Table 24-98 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to the POC WAKEUP state by writing FLXnSUCC1.FLXnCMD[3:0] = 0011 (WAKEUP command).	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to the WAKEUP_LISTEN state.	WAKEUP_STANDBY	WAKEUP_LISTEN
2	WUP on wakeup channel selected by bit FLXnSUCC1.FLXnWUCS is received, or a frame header is received on either available channel.	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	The wakeup pattern has been transmitted normally.	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired or WUP detected on the wakeup channel selected by bit FLXnSUCC1.FLXnWUCS or frame header received on either available channel	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup is completed (after T2 or T4 or T6), or the host has issued command FLXnSUCC1.FLXnCMD[3:0] = 0010 (READY command). This command also resets the wakeup FSM to the WAKEUP_STANDBY state.	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters listen timeout FLXnSUCC2.FLXnLT[20:0] and listen timeout noise. Listen timeout enables a fast cluster wakeup in a noise free environment, while listen timeout

noise enables wakeup under more difficult conditions regarding noise interference.

In the WAKEUP_SEND state, the FlexRay controller transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup, the host has to bring the FlexRay controller into the STARTUP state by CHI command RUN.

In the WAKEUP_DETECT state, the FlexRay controller attempts to identify the reason for the wakeup collision detected in the WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by FLXnSUCC2.FLXnLT[20:0]. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to the READY state. Otherwise, WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any startup attempt by a wakened node for the minimum time required for other coldstart nodes to become awake and operable.

The FlexRay Protocol Specification v2.1 recommends that two different FlexRay controllers shall awake the two channels.

Host operation

The host must coordinate the wakeup of the two channels and must decide whether to wake a specific channel. The sending of the wakeup pattern is initiated by the host. The wakeup pattern is detected by the remote BDs and signaled to their local host.

Wakeup procedure controlled by host (single-channel wakeup):

- Configuring the FlexRay controller to the CONFIG state
  - Selecting wakeup channel by setting bit FLXnSUCC1.FLXnWUCS
- Checking whether a local BD has received a WUP
- Activating BD of selected wakeup channel
- Command the FlexRay controller to enter the READY state
- Commanding the FlexRay controller to start wakeup on the configured channel by writing FLXnSUCC1.FLXnCMD[3:0] = 0011
  - The FlexRay controller enters WAKEUP.
  - The FlexRay controller returns to the READY state and signals status of wakeup attempt to the host.
- Waiting predefined time to allow the other nodes to wakeup and become operable
- Coldstart node:
  - Waiting for WUP on the other channel in the case of a dual channel cluster
  - Resetting coldstart inhibit flag FLXnCCSV.FLXnCSI by writing FLXnSUCC1.FLXnCMD[3:0] = 1001 (ALLOW_COLDSTART command)
- Commanding the FlexRay controller to enter startup by writing FLXnSUCC1.FLXnCMD[3:0] = 0100 (RUN command)

Wakeup procedure triggered by BD:

- BD recognizes a wakeup.
- BD triggers power-up of host (if required).

- BD signals wakeup event to host.
- The host configures its local FlexRay controller.
- If necessary, host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and become operable.
- The host commands the FlexRay controller to enter the STARTUP state by writing FLXnSUCC1.FLXnCMD[3:0] = 0100 (RUN command).

Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbols and wakeup patterns are configured by registers FLXnPRTC1 and FLXnPRTC2.

- Single channel wakeup: Wakeup symbols cannot be sent on both channels at the same time.
- A collision of wakeup symbols sent by two nodes can be recovered. (Two overlapping wakeup symbols are always recognizable.)
- Wakeup symbols must be configured identical in all nodes of a cluster.
- The wakeup symbol transmit low level time is configured by FLXnPRTC2.FLXnTXL[5:0].
- Wakeup symbol idle time is used to listen for activity on the bus, and is configured by FLXnPRTC2.FLXnTXI[7:0].
- A wakeup pattern is composed of at least two Tx-wakeup symbols needed for wakeup.
- Number of repetitions is configurable by FLXnPRTC1.FLXnRWP[5:0] (2 to 63 repetitions).
- Wakeup symbol receive window length is configured by FLXnPRTC1.FLXnRXW[8:0].
- Wakeup symbol receive low time is configured by FLXnPRTC2.FLXnRXL[5:0].
- Wakeup symbol receive idle time is configured by FLXnPRTC2.FLXnRXI[5:0].

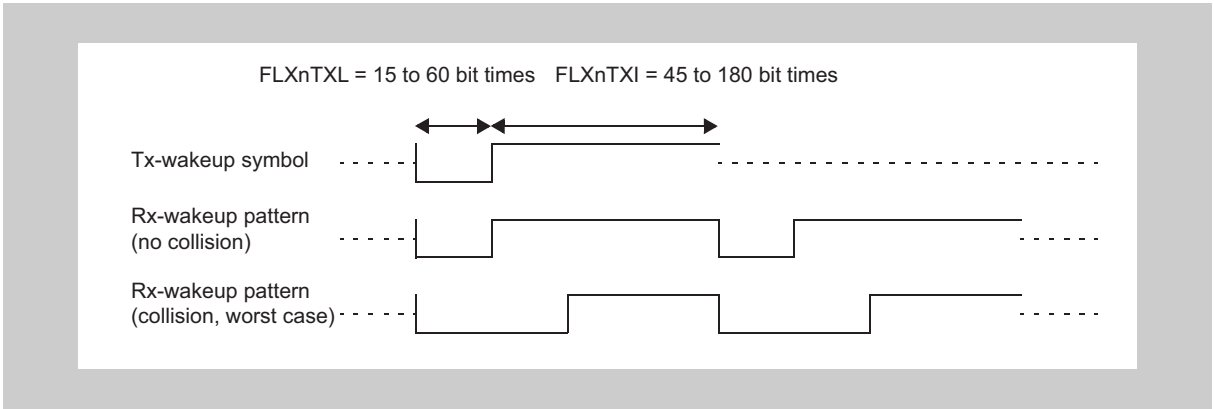


Figure 24-6 Timing of Wakeup Pattern

(7) STARTUP State

The description below is intended to help to configure startup for the FlexRay controller. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay Protocol Specification v2.1, section 7.2.

Any node entering the STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and operable. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any startup attempt by the wakened node at least for the time required for another coldstart node to become awake, to be operable, and to enter the startup state. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and operable.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter the NORMAL_ACTIVE state via the state shown in *figure24-7 State Diagram Time-Triggered Startup*:

- Coldstart initiating synchronization (leading coldstart node)
- Coldstart joining other coldstart nodes (following coldstart node)
- Coldstart integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXnSUCC1.FLXnTXST and FLXnSUCC1.FLXnTXSY set to 1. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is sent. In the frame header of the startup frame, the startup frame indicator bit is set.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by FLXnSUCC1.FLXnCSA[4:0].

At least two startup frames from distinct nodes are required for a non-coldstart node to be integrated. A non-coldstart node can start integration before the coldstart nodes have finished their startup. A non-coldstart node will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes.

The node can only leave the integration phase and actively participate in communication when these checks are passed.

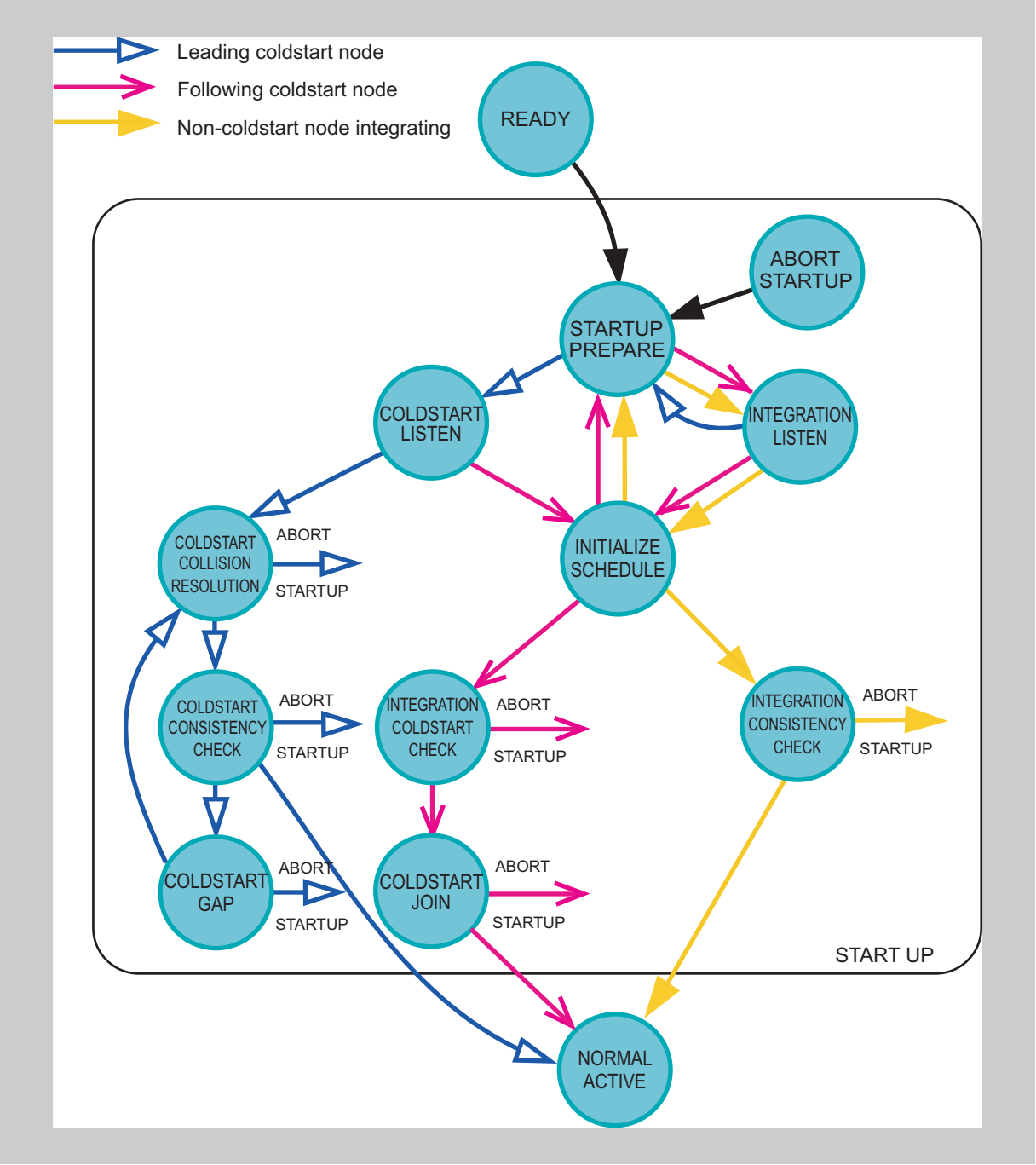


Figure 24-7 State Diagram Time-Triggered Startup

Coldstart inhibit mode

In coldstart inhibit mode, the node is prevented from initializing the TDMA communication schedule. If bit FLXnCCSV.FLXnCSI is set, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart process is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXnCCSV.FLXnCSI is set whenever the POC enters the READY state. The bit has to be cleared under control of the host by CHI command ALLOW_COLDSTART (FLXnSUCC1.FLXnCMD[3:0] = 1001).

Startup timeouts

The FlexRay controller supplies two different µT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are reset when the FlexRay controller enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (the COLDSTART_LISTEN state) with the intention of starting up communication.

Note The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXnSUCC2.FLXnLT[20:0] and FLXnSUCC2.FLXnLTN[3:0].

Startup timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by setting FLXnSUCC2.FLXnLT[20:0] (see section 24.4.5 (2), FLXnSUCC2 – SUC Configuration Register 2).

The startup timeout is:
pdListenTimeout = FLXnSUCC2.FLXnLT[20:0].

The startup timer is restarted upon:

- entering the COLDSTART_LISTEN state, or
- both channels reaching the idle state while in the COLDSTART_LISTEN state.

The startup timer is stopped:

- if communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state, or
- when the COLDSTART_LISTEN state is left.

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

Startup noise timer

At the same time when the startup timer is started for the first time (transition from the STARTUP_PREPARE state to the COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by setting FLXnSUCC2.FLXnLTN[3:0] (see section 24.4.5 (2), FLXnSUCC2 – SUC Configuration Register 2).

The startup noise timeout is:
pdListenTimeout • gListenNoise = FLXnSUCC2.FLXnLT[20:00] • (FLXnSUCC2.FLXnLTN[3:0] + 1)

The startup noise timer is restarted upon:

- entering the COLDSTART_LISTEN state, or
- reception of correctly decoded headers or CAS symbols while the node is in the COLDSTART_LISTEN state.

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer is not restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

Path of leading coldstart node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart process. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in the COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters the NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FLXnSUCC1.FLXnCSA[4:0]. The number of remaining coldstarts attempts can be read from FLXnCCSV.FLXnRCA[4:0]. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node is allowed to enter the COLDSTART_LISTEN state only if this value is larger than one, and it is allowed to enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

Path of following coldstart node (responding to leading coldstart node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization allows the node to successfully receive a second valid startup frame and to derive a

schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In the INTEGRATION_COLDSTART_CHECK state, it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In the COLDSTART_JOIN state, integrating coldstart nodes begin to transmit their own startup frames. Thereby the node that initiated the coldstart and the nodes joining it can check if their schedules agree to each other. If, for the following three cycles, the clock correction does not signal errors and at least one other coldstart node is visible, the node leaves the COLDSTART_JOIN state and enters the NORMAL_ACTIVE state. Thereby, it leaves STARTUP at least one cycle later than the node that initiated the coldstart.

Path of non-coldstart node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization allows the node to successfully receive a matching second valid startup frame and to derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In the INTEGRATION_CONSISTENCY_CHECK state, the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. If any errors are detected by clock correction, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

For a node of this state to leave the STARTUP state and to enter the NORMAL_OPERATION state, it needs to receive two valid startup frame pairs in each of two consecutive double-cycles. Consequently, they leave startup at least one double-cycle later than the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

(8) NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster finishes. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even/odd cycle pairs required).

In the NORMAL_ACTIVE state, the FlexRay controller supports regular communication functions:

- The FlexRay controller performs transmission and reception on the FlexRay bus as configured.
- Clock synchronization is executed.
- The host interface is operational.

The FlexRay controller exits from that state to enter:

- the HALT state by writing FLXnSUCC1.FLXnCMD[3:0] = 0110 (HALT command, at the end of the current cycle)
- the HALT state by writing FLXnSUCC1.FLXnCMD[3:0] = 0111 (FREEZE command, immediately)
- the HALT state due to change of the error state from ACTIVE to COMM_HALT
- the NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- the READY state by writing FLXnSUCC1.FLXnCMD[3:0] = 0010 (READY command)

(9) NORMAL_PASSIVE State

The node state changes from the NORMAL_ACTIVE state to the NORMAL_PASSIVE state when the error state changes from ACTIVE to PASSIVE.

In the NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In the NORMAL_PASSIVE state,

- The FlexRay controller performs reception on the FlexRay bus.
- The FlexRay controller does not transmit any frames or symbols on the FlexRay bus.
- Clock synchronization is executed.
- The host interface is operational.

The FlexRay controller exits from this state to

- the HALT state by writing FLXnSUCC1.FLXnCMD[3:0] = 0110 (HALT command, at the end of the current cycle)
- the HALT state by writing FLXnSUCC1.FLXnCMD[3:0] = 0111 (FREEZE command, immediately)
- the HALT state due to change of the error state from PASSIVE to COMM_HALT

- the NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when FLXnCCEV.FLXnPTAC[4:0] equals FLXnSUCC1.FLXnPTA[4:0] - 1.
- the READY state by writing FLXnSUCC1.FLXnCMD[3:0] = 0010 (READY command)

(10) HALT State

In this state, all communication (reception and transmission) is stopped.

The FlexRay controller enters this state:

- by writing FLXnSUCC1.FLXnCMD[3:0] = 0110 (HALT command) while the FlexRay controller is in the NORMAL_ACTIVE or NORMAL_PASSIVE state
- by writing FLXnSUCC1.FLXnCMD[3:0] = 0111 (FREEZE command) from all states
- when exiting from the NORMAL_ACTIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and FLXnSUCC1.FLXnHCSE is set
- when exiting from the NORMAL_PASSIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and FLXnSUCC1.FLXnHCSE is set

The FlexRay controller exits from this state and enters the DEFAULT_CONFIG state:

- by writing FLXnSUCC1.FLXnCMD[3:0] = 0001 (CONFIG command)

When the FlexRay controller enters the HALT state, all configuration and status data is maintained for analyzing purposes.

When the host writes FLXnSUCC1.FLXnCMD[3:0] = 0110 (HALT command), the FlexRay controller sets bit FLXnCCSV.FLXnHRQ and enters the HALT state at the next end of cycle.

When the host writes FLXnSUCC1.FLXnCMD[3:0] = 0111 (FREEZE command), the FlexRay controller enters the HALT state immediately and sets bit FLXnCCSV.FLXnFSI.

The POC state from which the transition to the HALT state took place can be read from FLXnCCSV.FLXnPSL[5:0].

24.5.6 Network Management

The accrued network management (NM) vector can be read from registers FLXnNMV1 to FLXnNMV3. The FlexRay controller performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the payload preamble indicator (PPI) bit set. Only static frames can be configured to hold NM information. The FlexRay controller updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by FLXnNEMC.FLXnNML[3:0]. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set, bit PPIT in the header section of the respective transmit buffer has to be set via FLXnWRHS1.FLXnWHPPIT. In addition, the host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the host.

Note When a message buffer is configured for transmission/reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by FLXnNEMC.FLXnNML[3:0].
When the FlexRay controller transits to the HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case, FLXnNMV1 to FLXnNMV3 hold the value from the cycle before.

24.5.7 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channels A and B). A message buffer is updated/transmitted only if the required matches occur.

Filtering is done on the following fields.

- Channel ID
- Frame ID
- Cycle counter

The following filter combinations for acceptance/transmit filtering are allowed:

- Frame ID + channel ID
- Frame ID + channel ID + cycle counter

All configured filters must match in order to store a received message in a message buffer.

Note For the FIFO, the acceptance filter is configured by the FIFO rejection filter register and the FIFO rejection filter mask register.

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled, the configured cycle filter value must also match.

(1) Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

(2) Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 or 1 is configured to hold the startup/sync frame or the single slot frame by bits FLXnSUCC1.FLXnTXST, FLXnSUCC1.FLXnTXSY, and FLXnSUCC1.FLXnTSM, cycle counter filtering for message buffer 0 or 1 must be disabled.

Note Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in table 24-99, Definition of Cycle Set.

Table 24-99 Definition of Cycle Set

Cycle Code	Matching Cycle Counter Values		
0b000000x	All cycles		
0b000001c	Every second cycle	at (Cycle Count)mod2	= c
0b00001cc	Every fourth cycle	at (Cycle Count)mod4	= cc
0b0001ccc	Every eighth cycle	at (Cycle Count)mod8	= ccc
0b001cccc	Every sixteenth cycle	at (Cycle Count)mod16	= cccc
0b01ccccc	Every thirty-second cycle	at (Cycle Count)mod32	= ccccc
0b1cccccc	Every sixty-fourth cycle	at (Cycle Count)mod64	= cccccc

table 24-100, Examples for Valid Cycle Sets, below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 24-100 Examples for Valid Cycle Sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7- ... -63
0b0000100	0-4-8-12- … -60
0b0001110	6-14-22-30- ... -62
0b0011000	8-24-40-56
0b0100011	3-35
0b1001001	9

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Other filter criteria must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Other filter criteria must also be met.

(3) Channel ID Filtering

There is a 2-bit channel filtering field (CHA, CHB) located in the header section of each message buffer in the message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see table 24-101, Channel Filtering Configuration).

Table 24-101 Channel Filtering Configuration

CHA	CHB	Transmit Buffer transmit frame	Receive Buffer store valid receive frame
1	1	On both channels (static segment only)	Received on channel A or B (store first semantically valid frame, static segment only)
1	0	On channel A	Received on channel A
0	1	On channel B	Received on channel B
0	0	No transmission	Ignore frame

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment, a transmit buffer can be set up for transmission on both channels (FLXnWHCHA and FLXnWHCHB set).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment, a receive buffer can be setup for reception on both channels (FLXnWHCHA and FLXnRHCHB set).

Note If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as FLXnWHCHA = FLXnWHCHB = 0 and FLXnRHCHA = FLXnRHCHB = 0).

(4) FIFO Filtering

For FIFO filtering there is one rejection filter and one rejection filter mask available. The FIFO filter consists of channel filter FLXnFRF.FLXnCH[1:0], frame ID filter FLXnFRF.FLXnFID[10:00], and cycle counter filter FLXnFRF.FLXnCYF[6:0]. Registers FLXnFRF and FLXnFRFM can be configured in the DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXnFRF.FLXnCYF[6:0], all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

24.5.8 Transmit Process

(1) Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the input buffer has to be started by writing to the input buffer command request register latest at this time.

(2) Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment, different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the input buffer has to be started by writing to the input buffer command request register at this time at the latest.

The start of latest transmit configured by FLXnMHDC.FLXnSLT[12:00] defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

(3) Transmit Buffers

Message buffers can be configured as transmit buffers by setting bit FLXnWHCFG in the header section of the respective message buffer to 1 via FLXnWRHS1.

There exist the following possibilities to assign a transmit buffer to the FlexRay controller channels:

- Static segment: channel A or channel B, channel A and channel B
- Dynamic segment: channel A or channel B

Message buffer 0 or 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by FLXnSUCC1.FLXnTXST, FLXnSUCC1.FLXnTXSY, and FLXnSUCC1.FLXnTSM. In this case, it can be reconfigured in the DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup/sync frame per communication cycle. Transmission of startup/sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXnMRC.FLXnSEC[1:0]. Due to the organization of the data partition in the message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The FlexRay controller does not have the capability to calculate the header CRC. The host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the host has to set the PPIT bit in

the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer.

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by FLXnMHDC.FLXnSFDL[6:0], the FlexRay controller generates padding bytes to ensure that frames have a proper physical length. The padding pattern is logical zero.

Note In the case of an odd payload length (PLC = 1, 3, 5, ...), the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is all zero.

Each transmit buffer provides a transmission mode flag TXM, which allows the host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode, the FlexRay controller resets the respective FLXnTXR flag after transmission has completed. Now the host may update the transmit buffer.

In continuous mode, the FlexRay controller does not reset the respective transmission request flag FLXnTXR after successful transmission. In this case, a frame is sent out each time the filter criteria match. The FLXnTXR flag can be reset by the host by writing the respective message buffer number to the FLXnIBCR register while bit FLXnIBCM.FLXnSTXRH is set to 0.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

(4) Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configuring the transmit buffer in the message RAM via FLXnWRHS1, FLXnWRHS2, and FLXnWRHS3
- Writing the data section of the transmit buffer via FLXnWRDSm
- Transferring the configuration and message data from input buffer to the message RAM by writing the number of the target message buffer to register FLXnIBCR
- If configured in register FLXnIBCM, the transmission request flag FLXnTXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer becomes ready for transmission.
- Checking whether the message buffer has been transmitted by checking the respective FLXnTXR bit (FLXnTXR = 0) in the FLXnTRXQ1/2/3/4 registers (single-shot mode only)

After transmission has completed, the respective FLXnTXR flag in the FLXnTXRQ1/2/3/4 registers is reset (single-shot mode), and, if bit MBI in the header section of the message buffer is set, flag FLXnSIR.FLXnTXIF is set to 1. If enabled, an interrupt is generated.

(5) Null Frame Transmission

In the case when the host does not set, in static segment, the transmission request flag before transmit time and there is no other transmit buffer with matching filter criteria, the FlexRay controller transmits a null frame with the null frame indication bit set to 0 and the payload data set to 0.

In the following cases, the FlexRay controller transmits a null frame:

- The message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (FLXnTXR = 0).
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status MBS is updated.

Null frames are not transmitted in the dynamic segment.

24.5.9 Receive Process

(1) Receive Buffers

Portion of message buffers can be configured as receive buffers by setting bit FLXnRHCFG in the header section of the respective message buffer to 0 using FLXnWRHS1.

The following possibilities exist for assigning a receive buffer to the FlexRay controller channels:

- Static segment: channel A or channel B, channel A and channel B (the FlexRay controller stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The FlexRay controller transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer that matches the filter condition. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXnMRC.FLXnSEC[1:0]. If a message buffer is reconfigured (if header section is updated) during runtime, the received message may be lost in the respective communication cycle.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

(2) Frame Reception

The following steps are required to prepare a message buffer for reception:

- Configuring the receive buffer in the message RAM via FLXnWRHS1, FLXnWRHS2, and FLXnWRHS3
- Transferring the configuration from input buffer to the message RAM by writing the number of the target message buffer to register FLXnIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the FlexRay controller receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective FLXnND flag in the FLXnNDAT1/2/3/4 registers is set, and, if bit MBI in the header section of that message buffer is set, flag FLXnSIR.FLXnRXIF is set to 1. An interrupt is generated if it is enabled.

In the case when bit FLXnND was already set when the message handler updates the message buffer, bit FLXnMBS.FLXnMBSMLST of the respective message buffer is set and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status MBS is updated.

When the message handler changes the message buffer status MBS in the header section of a message buffer, the respective FLXnMBC flag in the FLXnMBSC1/2/3/4 registers is set, and if bit MBI in the header section of that message buffer is set, flag FLXnSIR.FLXnMBSIF is set to 1. An interrupt is generated if it is enabled.

If the payload length indicated by PLR[6:0] of the received frame is longer than the value specified by PLC[6:0] in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the message RAM via the output buffer, proceed as described in section 24.5.11, Message Handling, "Data transfer from message RAM to output buffer".

Note The FLXnND and FLXnMBC flags are automatically cleared by the message handler when the payload data and the header of a received message have been transferred to the output buffer, respectively.

(3) Null Frame Reception

The payload segment of a received null frame is not copied into the matching receive buffer. If a null frame has been received, only the message buffer status MBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the message handler changes the message buffer status MBS in the header section of a message buffer, the respective FLXnMBC flag in the FLXnMBSC1/2/3/4 registers is set, and if bit MBI in the header section of that message buffer is set, flag FLXnSIR.FLXnMBSIF is set to 1. An interrupt is generated if it is enabled.

24.5.10 FIFO Function

(1) Description

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXnMRC.FLXnFFB[7:0] and ending with the message buffer referenced by FLXnMRC.FLXnLCB[7:0]. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any receive buffer but matching the configured condition of the FIFO filter is stored into the FIFO. In this case, frame ID, payload length, receive cycle count, and the message buffer status MBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit FLXnSIR.FLXnRFNEF shows that the FIFO is not empty, bit FLXnSIR.FLXnRFCLF is set when the receive FIFO fill level FLXnFSR.FLXnRFFL[7:0] is equal or greater than the critical level as configured by FLXnFCL.FLXnCL[7:0], and bit FLXnEIR.FLXnRFOE shows that a FIFO overrun has been detected. Interrupts are generated if they are enabled.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT index register (PIDX) is an index to the next available location in the FIFO. When a new message is received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET index register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the output buffer. The PUT index register and the GET index register are not accessible by the host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message is read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXnEIR.FLXnRFOE.

FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case, flag FLXnSIR.FLXnRFNEF is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in *figure24-8 FIFO Status: Empty, Not Empty, Overrun*, for a three message buffer FIFO.

The programmable FIFO rejection filter (FRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXnFRF.FLXnRSS is set to 1 (default), all messages received in the static segment are rejected by the FIFO. If bit FLXnFRF.FLXnRSS is set to 1 (default), received null frames are not stored in the FIFO.

The FIFO rejection filter mask (FRFM) specifies which bits of the frame ID filter in the FIFO rejection filter register are ignored by rejection filtering.

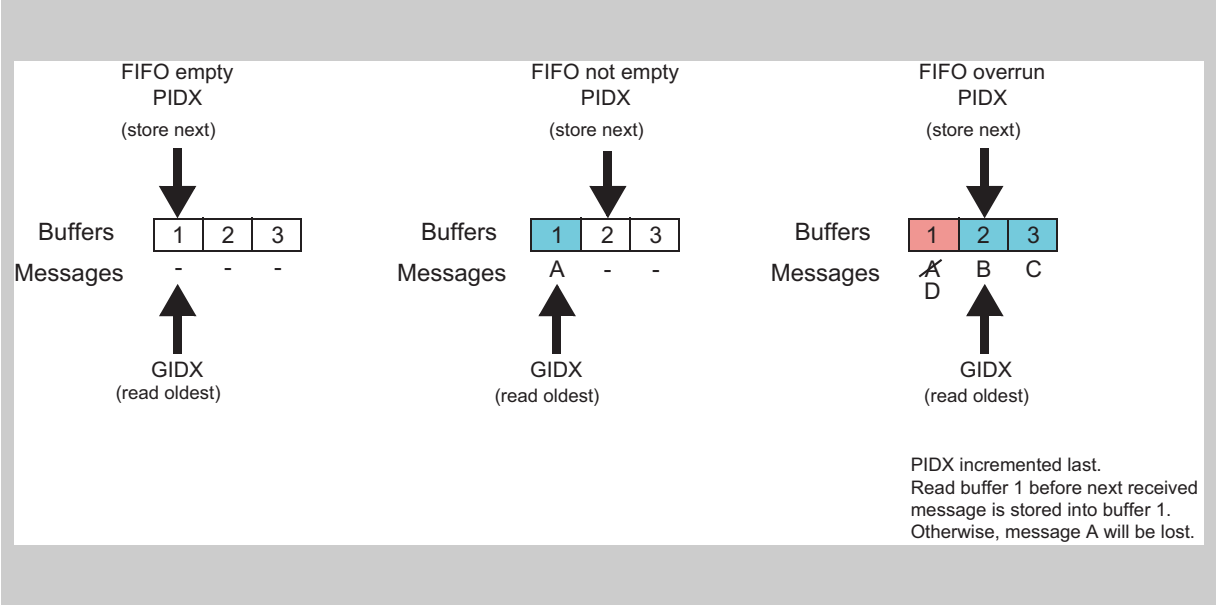


Figure 24-8 FIFO Status: Empty, Not Empty, Overrun

(2) Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is possible only when the FlexRay controller is in the DEFAULT_CONFIG or CONFIG state. While the FlexRay controller is in the DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO, the payload length configured should be set to the same value via FLXnWRHS2.FLXnWHPLC[6:0]. The data pointer to the first 32-bit word of the data section of the respective message buffer in the message RAM has to be configured via FLXnWRHS3.FLXnWHDP[10:00].

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of FLXnWHDP and FLXnWHPLC, irrelevant.

Note It is recommended that the MBI bits of the message buffers belonging to the FIFO is set to 0 via FLXnWRHS1.FLXnWHMBI to avoid generation of RX interrupts.
If the payload length of a received frame is longer than the value set in FLXnWRHS2.FLXnWHPLC[6:0] in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

(3) Access to the FIFO

For FIFO access outside the DEFAULT_CONFIG and CONFIG state, the host has to trigger a transfer from the message RAM to the output buffer by writing the number of the first message buffer of the FIFO (referenced by FLXnMRC.FLXnFFB[7:0]) to the register FLXnOBCR. The message handler then transfers the message buffer addressed by the GET index register (GIDX) to the output buffer. After this transfer, the GET index register (GIDX) is incremented.

24.5.11 Message Handling

The message handler controls data transfers between the input/output buffer and the message RAM and between the message RAM and the two transient buffer RAMs. All accesses to the internal RAMs are 32+7 bit accesses. The additional 7 bits are for the ECC code.

Access to the message buffers stored in the message RAM is done under control of the message handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the host to the message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to the value in FLXnGTUC07.FLXnNSS[9:0]. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FLXnGTUC07.FLXnNSS[9:0] + 1 to 2047.

Received messages with no matching receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

(1) Reconfiguration of Message Buffers

In the case when an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via input buffer registers FLXnWRHS1 to FLXnWRHS3.

Reconfiguration has to be enabled via control bits FLXnMRC.FLXnSEC[1:0] in the message RAM configuration register.

If a message buffer has not been transmitted/updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission/reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore, it may happen that a reconfigured message buffer is not transmitted/updated from a received frame in the cycle where it was reconfigured. The message RAM is scanned according to Table 24-102 below.

Table 24-102 Scan of Message RAM

Start of Scan in Slot	Scan for Slots
1	2 to 15, 1 (next cycle)
8	6 to 23, 1 (next cycle)
16	24 to 31, 1 (next cycle)
24	32 to 39, 1 (next cycle)
...	...

A message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the message RAM for slot 1 is done in the preceding cycle by checking, in parallel with the scan of the message RAM of the preceding cycle, whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by FLXnMRC.FLXnFDB[7:0]. In the case when message RAM scan starts during a dynamic segment, the scan starts with the message buffer number configured by FLXnMRC.FLXnFDB[7:0].

When a message buffer needs to be reconfigured to be used in slot 1 of the next cycle, the following points have to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the "static buffers", it will only be found if it is reconfigured before the last message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the "static + dynamic buffers", it will be used for communication if the host reconfigures it before the last message RAM scan in the current cycle checks this message buffer.
- The start of NIT terminates the message RAM scan. In the case when the message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

Note Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In the worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted/ updated from a received frame.

(2) Host Access to Message RAM

The message transfer between input buffer and message RAM as well as between message RAM and output buffer is triggered by the host by writing the number of the target/source message buffer to be accessed to the FLXnIBCR or FLXnOBCR register.

The FLXnIBCM and FLXnOBCM registers can be used to write/read header and data section of the selected message buffer separately.

If bit FLXnIBCM.FLXnSTXRH is set to = 1, the transmission request flag FLXnTXR of the selected message buffer is automatically set after the message buffer has been updated.
If bit FLXnIBCM.FLXnSTXRH is reset to 0, the transmission request flag FLXnTXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

The input buffer (IBF) and output buffer (OBF) are built up as a double buffer structure. One half of this double buffer structure is accessible by the host (IBF host/OBF host), while the other half (IBF shadow/OBF shadow) is accessed by the message handler for data transfers between IBF/OBF and message RAM.

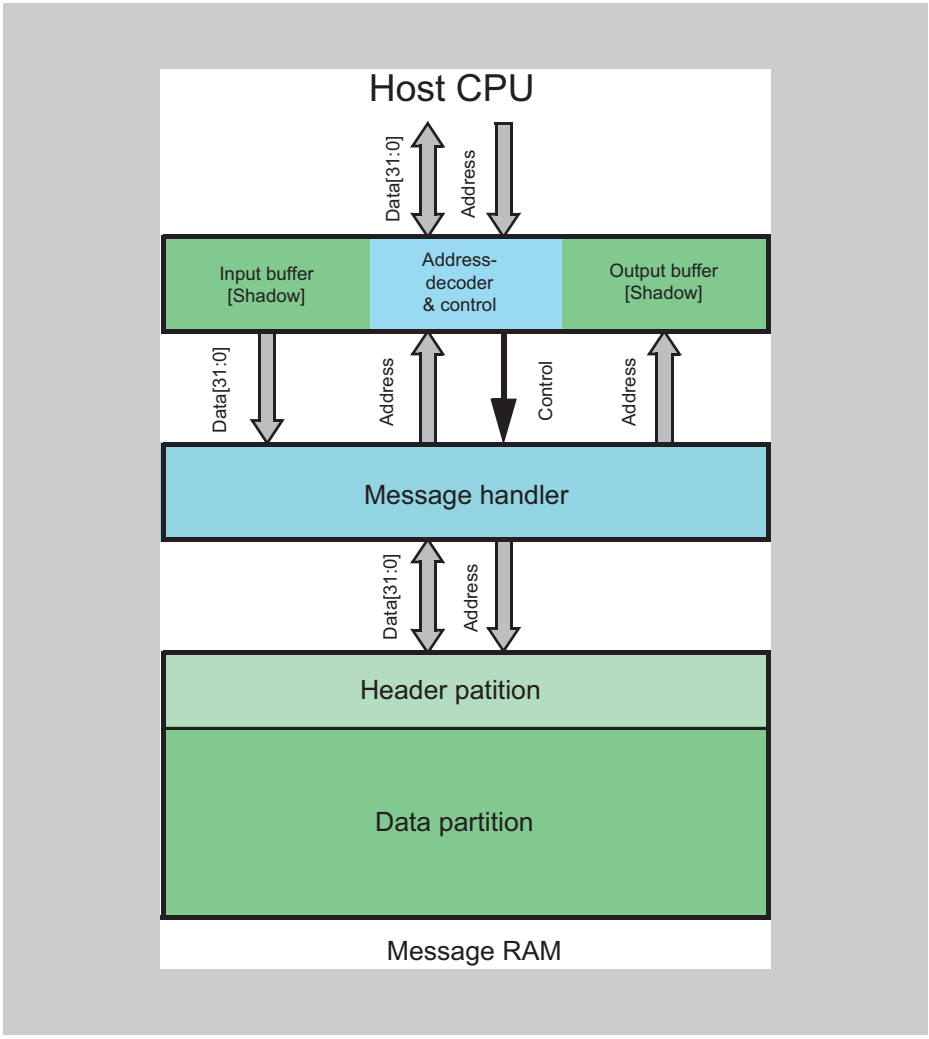


Figure 24-9 Host Access to Message RAM

Data transfer from input buffer to message RAM

To configure/update a message buffer in the message RAM, the host has to write the data to FLXnWRDSm and the header to FLXnWRHS1 to FLXnWRHS3. The specific action is selected by configuring the input buffer command mask FLXnIBCM.

When the host writes the number of the target message buffer in the message RAM to FLXnIBCR.FLXnIBRH[6:0], IBF host and IBF shadow are swapped (see *figure24-10 Double Buffer Structure Input Buffer*).

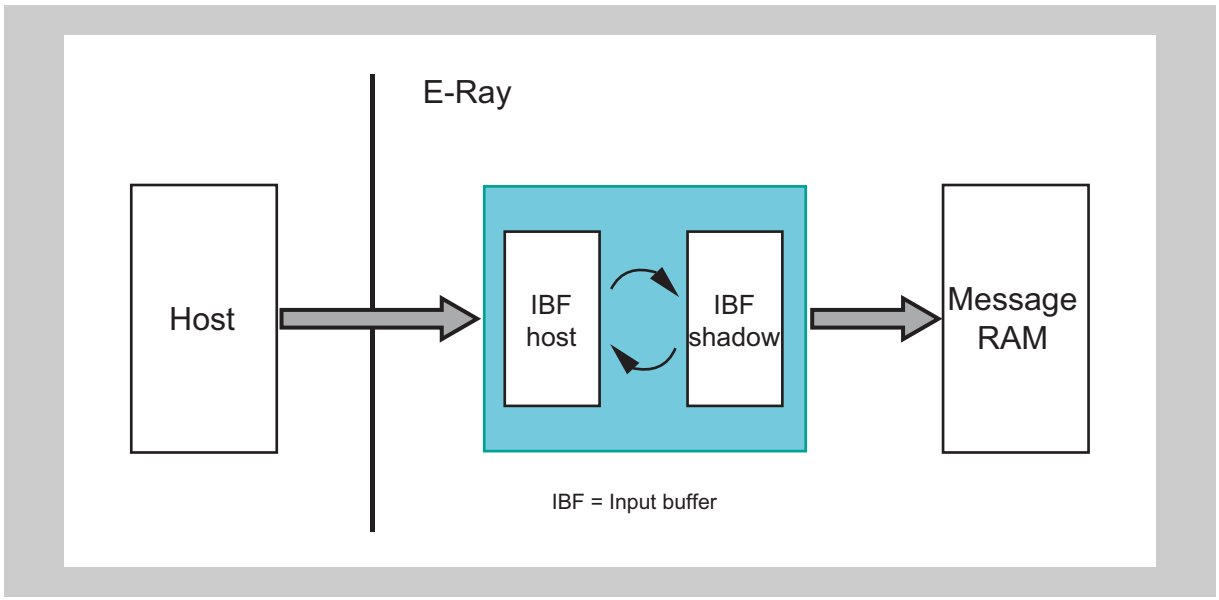


Figure 24-10 Double Buffer Structure Input Buffer

In addition, the bits in the FLXnIBCM and FLXnIBCR registers are also swapped to keep them attached to the respective IBF section.

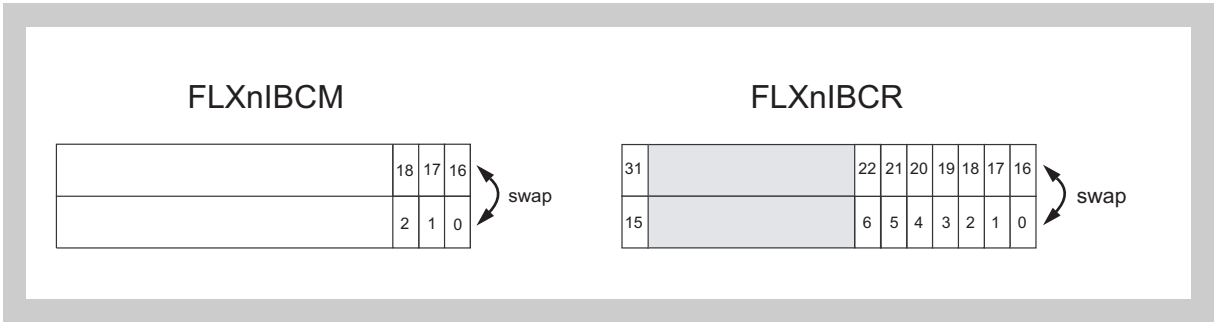


Figure 24-11 Swapping of FLXnIBCM and FLXnIBCR Bits

With this write operation, bit FLXnIBCR.FLXnIBSYS is set to 1. The message handler then starts to transfer the content of IBF shadow to the message buffer in the message RAM selected by FLXnIBCR.FLXnIBRS[6:0].

While the message handler transfers the data from IBF shadow to the target message buffer in the message RAM, the host may write the next message to IBF host. After the transfer between IBF shadow and the message RAM has completed, bit FLXnIBCR.FLXnIBSYS is set back to 0 and the next transfer to the message RAM can be started by the host by writing the respective target message buffer number to FLXnIBCR.FLXnIBRH[6:0].

If a write access to FLXnIBCR.FLXnIBRH[6:0] occurs while FLXnIBCR.FLXnIBSYS is 1, FLXnIBCR.FLXnIBSYH is set to 1. After completion of the ongoing data transfer from IBF shadow to the message RAM, IBF host and IBF shadow are swapped, FLXnIBCR.FLXnIBSYH is reset to 0, FLXnIBCR.FLXnIBSYS remains set to 1, and the next transfer to the message RAM is started. In addition, the message buffer numbers stored under FLXnIBCR.FLXnIBRH[6:0] and FLXnIBCR.FLXnIBRS[6:0], and the command mask flags are also swapped.

Example of a 32-bit host access sequence:

Configuring/updating n-th message buffer via IBF.

- Waiting until FLXnIBCR.FLXnIBSYH is reset
- Writing data section to FLXnWRDSm
- Writing header section to FLXnWRHS1 to FLXnWRHS3
- Writing command mask: write FLXnIBCM.FLXnSTXRH, FLXnIBCM.FLXnLDSH, and FLXnIBCM.FLXnLHSH
- Requesting data transfer to target message buffer: writing in FLXnIBCR.FLXnIBRH[6:0]

Configuring/updating (n+1)th message buffer via IBF.

- Waiting until FLXnIBCR.FLXnIBSYH is reset
- Writing data section to FLXnWRDSm
- Writing header section to FLXnWRHS1 to FLXnWRHS3
- Writing command mask: writing FLXnIBCM.FLXnSTXRH, FLXnIBCM.FLXnIBCM, and FLXnIBCM.FLXnLHSH
- Requesting data transfer to target message buffer: writing in FLXnIBCR.FLXnIBRH[6:0]

Note Any write access to IBF while FLXnIBCR.FLXnIBSYH is 1 will set error flag FLXnEIR.FLXnIIBAE to 1. In this case, the write access has no effect.

Table 24-103 Assignment of FLXnIBCM Bits

Position	Access	Bit	Function
18	r	FLXn STXRS	Set transmission request (shadow)
17	r	FLXn LDSS	Load data section (shadow)
16	r	FLXn LHSS	Load header section (shadow)
2	r/w	FLXn STXRH	Set transmission request (host)
1	r/w	FLXn LDSH	Load data section (host)
0	r/w	FLXn LHSH	Load header section (host)

Table 24-104 Assignment of FLXnIBCR Bits

Position	Access	Bit	Function
31	r	FLXn IBSYS	IBF busy (shadow) Signals ongoing transfer from IBF shadow to message RAM
22 to 16	r	FLXn IBRS[6:0]	IBF request (shadow) Number of message buffer currently/last updated
15	r	FLXn IBSYH	IBF busy (host) Transfer request pending for message buffer referenced by IBRH[6:0]
6 to 0	r/w	FLXn IBRH[6:0]	IBF request (host) Number of message buffer to be updated next

Data transfer from message RAM to output buffer

To read a message buffer from the message RAM, the host has to write to register FLXnOBCR to trigger the data transfer as configured in FLXnOBCM. After the transfer has completed, the host can read the transferred data from FLXnRDDSm, FLXnRDHS1 to FLXnRDHS3, and FLXnMBS.

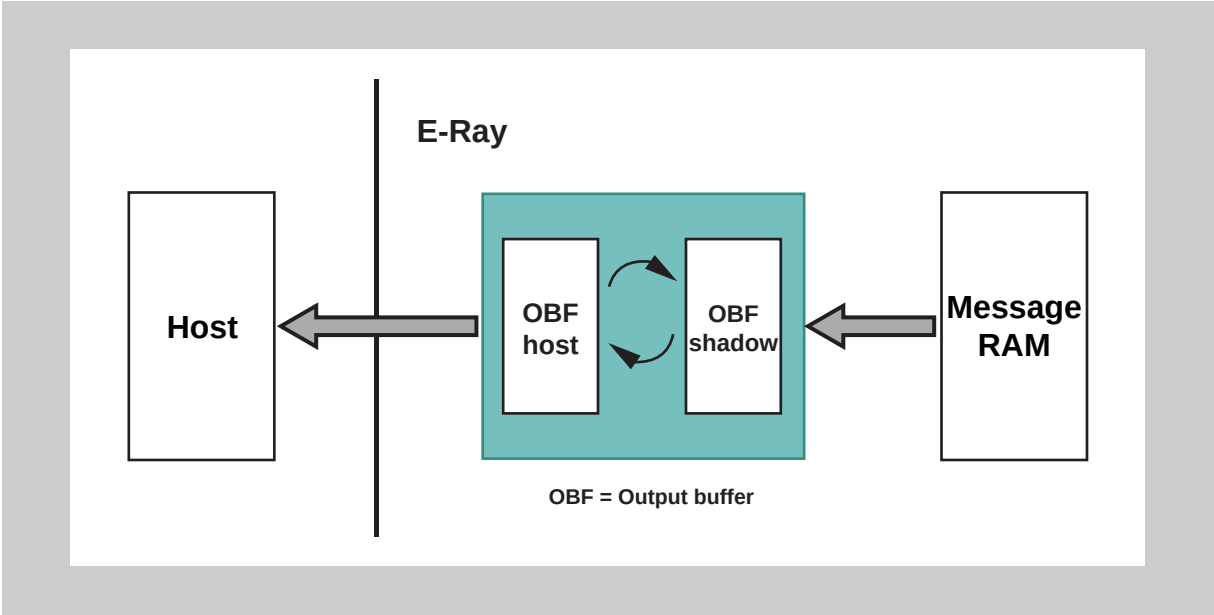


Figure 24-12 Double Buffer Structure Output Buffer

The OBF host and OBF shadow as well as bits FLXnOBCM.FLXnRHSS, FLXnOBCM.FLXnRDSS, FLXnOBCM.FLXnRHSH, FLXnOBCM.FLXnRDSH and bits FLXnOBCR.FLXnOBRS[6:0], FLXnOBCR.FLXnOBRH[6:0] are swapped under control of bits FLXnOBCR.FLXnVIEW and FLXnOBCR.FLXnREQ.

Writing bit FLXnOBCR.FLXnREQ to 1 copies bits FLXnOBCM.FLXnRHSS, FLXnOBCM.FLXnRDSS and bits FLXnOBCR.FLXnOBRS[6:0] to an internal storage.

After setting FLXnOBCR.FLXnREQ to 1, FLXnOBCR.FLXnOBSYS is set to 1, and the transfer of the message buffer selected by FLXnOBCR.FLXnOBRS[6:0] from the message RAM to OBF shadow is started. After the transfer between the message RAM and OBF shadow has completed, the FLXnOBCR.FLXnOBSYS bit is set back to 0. Bits FLXnOBCR.FLXnREQ and FLXnOBCR.FLXnVIEW can only be set to 1 while FLXnOBCR.FLXnOBSYS is 0.

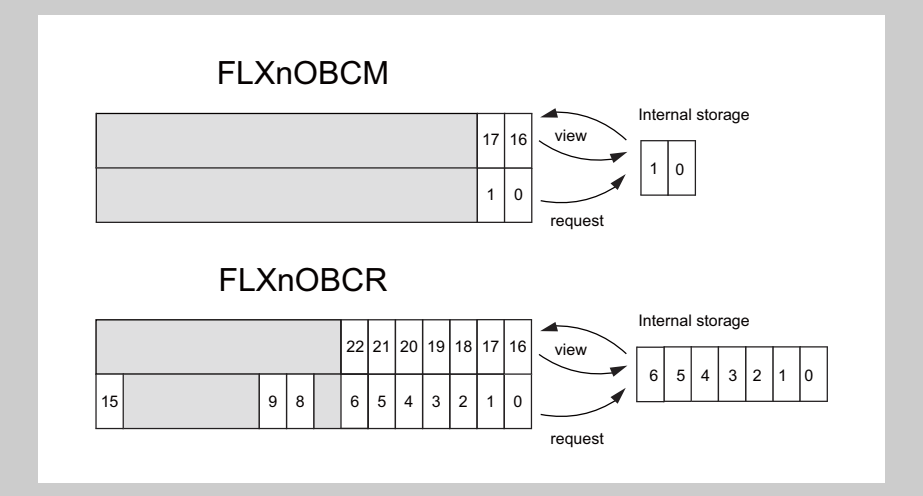


Figure 24-13 Swapping of FLXnOBCM and FLXnOBCR Bits

The OBF host and OBF shadow are swapped by setting bit FLXnOBCR.FLXnVIEW to 1 while bit FLXnOBCR.FLXnOBSYS is 0.

In addition, bits FLXnOBCR.FLXnOBRH[6:0] and bits FLXnOBCM.FLXnRHSH, FLXnOBCM.FLXnRDSH are swapped with the internal storage of the register in order to assure that the message buffer number stored in FLXnOBCR.FLXnOBRH[6:0] and the mask configuration stored in FLXnOBCM.FLXnRHSH and FLXnOBCM.FLXnRDSH matches the transferred data stored in OBF host.

Now the host can read the transferred message buffer from OBF host, and the message handler can transfer the next message from the message RAM to OBF shadow.

If bits FLXnREQ and FLXnVIEW are set to 1 with the same write access while FLXnOBSYS is 0, FLXnOBSYS is automatically set to 1 and OBF shadow and OBF host are swapped. Additionally, mask bits FLXnOBCM.FLXnRDSH and FLXnOBCM.FLXnRHSH are swapped with the internal storage of the register to keep them attached to the respective output buffer transfer. Afterwards, FLXnOBRS[6:0] is copied to the register internal storage, mask bits FLXnOBCM.FLXnRDSS and FLXnOBCM.FLXnRHSS are copied to register FLXnOBCM internal storage, and the transfer of the selected message buffer from the message RAM to OBF shadow is started. While the transfer is ongoing, the host can read the message buffer transferred by the previous transfer from OBF host. When the current transfer between message RAM and OBF shadow has completed, this is signaled by setting FLXnOBSYS back to 0.

Example of a 32-bit host access (single access):

If a single message buffer has to be read out, two separate write accesses to FLXnOBCR.FLXnREQ and FLXnOBCR.FLXnVIEW are necessary.

- Waiting until FLXnOBCR.FLXnOBSYS is reset
- Writing output buffer command mask FLXnOBCM.FLXnRHSS, FLXnOBCM.FLXnRDSS
- Requesting transfer of message buffer to OBF shadow by writing FLXnOBCR.FLXnOBRS[6:0] and FLXnOBCR.FLXnREQ
- Waiting until FLXnOBCR.FLXnOBSYS is reset
- Toggling OBF shadow and OBF host by writing FLXnOBCR.FLXnVIEW = 1
- Reading out transferred message buffer by reading FLXnRDDSm, FLXnRDHS1 to FLXnRDHS3, and FLXnMBS

Example of a 32-bit host access sequence (continuous access):

Transfer of the first message buffer to OBF shadow is requested.

- Waiting until FLXnOBCR.FLXnOBSYS is reset
- Writing output buffer command mask FLXnOBCM.FLXnRHSS, FLXnOBCM.FLXnRDSS for the 1st message buffer
- Requesting transfer of the 1st message buffer to OBF shadow by writing FLXnOBCR.FLXnOBRS[6:0] and FLXnOBCR.FLXnREQ

The OBF shadow and OBF host is swapped to read out the 1st transferred message buffer and to request transfer of the 2nd message buffer:

- Waiting until FLXnOBCR.FLXnOBSYS is reset
- Writing output buffer Command Mask FLXnOBCM.FLXnRHSS, FLXnOBCM.FLXnRDSS for the 2nd message buffer
- Toggling OBF shadow and OBF host and starting transfer of the 2nd message buffer to OBF shadow simultaneously by writing FLXnOBCR.FLXnOBRS[6:0] of the 2nd message buffer, FLXnOBCR.FLXnREQ, and FLXnOBCR.FLXnVIEW.
- Reading out the 1st transferred message buffer by reading FLXnRDDSm, FLXnRDHS1 to FLXnRDHS3, and FLXnMBS

. . .

Access to the last requested message buffer is requested (there will be no succeeding requests):

- Waiting until FLXnOBCR.FLXnOBSYS is reset
- Requesting access to the last transferred message buffer by writing FLXnOBCR.FLXnVIEW
- Reading out the last transferred message buffer by reading FLXnRDDSm, FLXnRDHS1 to FLXnRDHS3, and FLXnMBS

Table 24-105 Assignment of FLXnOBCM Bits

Position	Access	Bit	Function
17	r	FLXn RDSH	Data section available for host access
16	r	FLXn RHSH	Header section available for host access
1	r/w	FLXn RDSS	Read data section (shadow)
0	r/w	FLXn RHSS	Read header section (shadow)

Table 24-106 Assignment of FLXnOBCR Bits

Position	Access	Bit	Function
22 to 16	r	FLXn OBRH[6:0]	OBF request (host) Number of message buffer available for host access
15	r	FLXn OBSYS	OBF busy (shadow) Signals ongoing transfer from message RAM to OBF shadow
9	r/w	FLXn REQ	Request transfer from message RAM to OBF shadow
8	r/w	FLXn VIEW	View (shadow) Swap OBF shadow and OBF host
6 to 0	r/w	FLXn OBRS[6:0]	OBF request (shadow) Number of message buffer for next request

(3) FlexRay Protocol Controller Access to Message RAM

The two transient buffer RAMs (TBF A, TBF B) are used to buffer the data for transfer between the two FlexRay protocol controllers and the message RAM.

Each transient buffer RAM is built up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding protocol controller while the other one is accessible by the message handler.

While the message handler is writing the next message to be sent to transient buffer Tx, the FlexRay channel protocol controller can access transient buffer Rx to store the message it is actually receiving. During transmission of the message stored in transient buffer Tx, the message handler transfers the last received message stored in transient buffer Rx to the message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the transient buffer RAMs and the shift registers of the FlexRay channel protocol controllers are done in words of 32-bit. This enables the use of a 32-bit shift register independent of the length of the FlexRay messages.

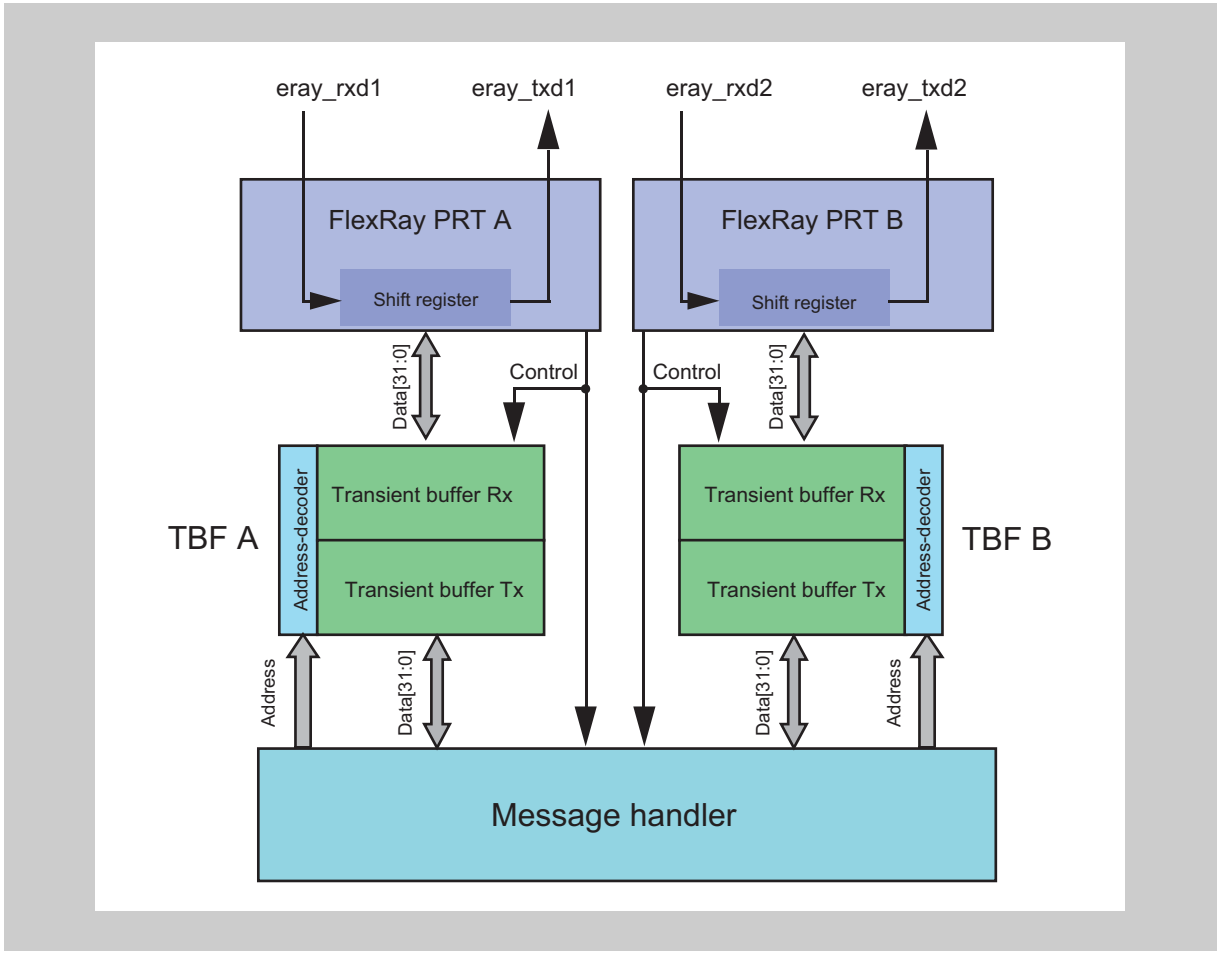


Figure 24-14 Access to Transient Buffer RAMs

24.5.12 Message RAM

To avoid conflicts between host access to the message RAM and FlexRay message reception/transmission, the host cannot directly access the message buffers in the message RAM. These accesses are handled via the input and output buffers. The message RAM is able to store up to 128 message buffers depending on the configured payload length.

The message RAM consists of 2048 × 39 = 79872 bits. Each 32-bit word is protected by a 7-bit ECC-code. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the message RAM has a structure as shown in *figure24-15 Structure of Message RAM*.

The data partition is allowed to start at message RAM word number: (FLXnMRC.FLXnLCB + 1) 4.

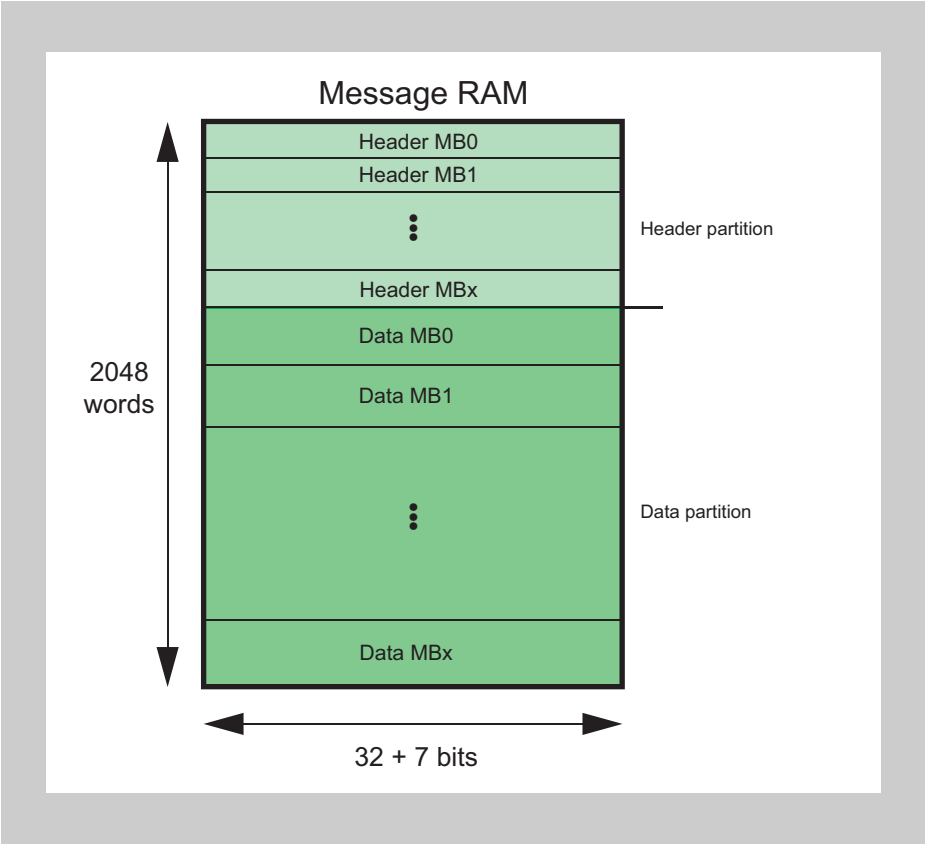


Figure 24-15 Structure of Message RAM

Header partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32+7 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each, or
- 56 message buffers with 128 byte data section each, or
- 128 message buffers with 48 byte data section each.

Restriction: Header partition + data partition may not occupy more than 2048 39-bit words.

(1) Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the message RAM as listed in the table below. Configuration of the header sections of the message buffers is done via IBF (FLXnWRHS1 to FLXnWRHS3). Read access to the header sections is done via OBF (FLXnRDHS1 to FLXnRDHS3 + FLXnMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the respective message buffer in the data partition of the message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO, (re)configuration is possible in the DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 39-bit words in the header partition of the message RAM. The header of message buffer 0 starts with the first word in the message RAM.

For transmit buffers, the header CRC has to be calculated by the host CPU.

The payload length received PLR[6:0], receive cycle count RCC[5:0], received on channel indicator RCI, startup frame indicator SFI, sync frame indicator SYN, null frame indicator NFI, payload preamble indicator PPI, and reserved bit RES are updated from received valid data frames only.

The header word 4 of each configured message buffer holds the respective message buffer status information.

Table 24-107 Header Section of a Message Buffer in the Message RAM

Bit Word	38 to 32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	E			MBI	TXM	PPIT	CFG	CHB	CHA		Cycle code						
2	E		Payload length received								Payload length configured						
3	E			RES	PPI	NFI	SYN	SFI	RCI			Receive cycle count					
4	E			RESS	PPIS	NFIS	SYNS	SFIS	RCIS			Cycle count status					
…	E																
…	E																

Bit Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1						Frame ID										
2						Tx buffer: Header CRC configured Rx buffer: Header CRC received										
3						Data pointer										
4	FTB	FTA		MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
…																
…																

- Frame configuration
- Filter configuration
- Message buffer control
- Message RAM configuration
- Updated from received data frame
- Message buffer status MBS
- ECC code
- Unused

Header 1

Write access via FLXnWRHS1, read access via FLXnRDHS1:

- Frame ID - Slot counter filtering configuration
- Cycle code - Cycle counter filtering configuration
- CHA, CHB - Channel filtering configuration
- CFG - Message buffer direction configuration: receive/transmit
- PPIT - Payload preamble indicator transmit
- TXM - Transmit mode configuration: single-shot/continuous
- MBI - Message buffer receive/transmit interrupt enable

Header 2

Write access via FLXnWRHS2, read access via FLXnRDHS2:

- Header CRC
  - Transmit buffer: Configured by the host (calculated from frame header)
  - Receive buffer: Updated from received frame
- Payload length configured
  - Length of data section (2-byte words) as configured by the host
- Payload length received
  - Length of payload segment (2-byte words) stored from received frame

Header 3

Write access via FLXnWRHS3, read access via FLXnRDHS3:

- Data pointer - Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXnRDHS3, valid for receive buffers only, updated from received frames:

- Receive cycle count - Cycle count from received frame
- RCI - Received on channel indicator
- SFI - Startup frame indicator
- SYN - Sync frame indicator
- NFI - Null frame indicator
- PPI - Payload preamble indicator
- RES - Reserved bit

Header 4

Read access via FLXnMBS, updated by the FlexRay controller at the end of the configured slot:

- VFRA - Valid frame received on channel A
- VFRB - Valid frame received on channel B
- SEOA - Syntax error observed on channel A
- SEOB - Syntax error observed on channel B

- CEOA - Content error observed on channel A
- CEOB - Content error observed on channel B
- SVOA - Slot boundary violation observed on channel A
- SVOB - Slot boundary violation observed on channel B
- TCIA - Transmission conflict indication channel A
- TCIB - Transmission conflict indication channel B
- ESA - Empty slot channel A
- ESB - Empty slot channel B
- MLST - Message lost
- FTA - Frame transmitted on channel A
- FTB - Frame transmitted on channel B
- Cycle count status - Actual cycle count when status was updated
- RCIS - Received on channel indicator status
- SFIS - Startup frame indicator status
- SYNS - Sync frame indicator status
- NFIS - Null frame indicator status
- PPIS - Payload preamble indicator status
- RESS - Reserved bit status

(2) Data Partition

The data partition of the message RAM stores the data sections of the message buffers configured for reception/transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay protocol controllers and the message RAM as well as between the host interface and the message RAM, the physical width of the message RAM is set to 4 bytes + 7-bits ECC code.

The data partition starts after the last word of the header partition. When configuring the message buffers in the message RAM the user has to ensure that the data pointers point to addresses within the data partition. table 24-108, Example for Structure of the Data Partition in the Message RAM, shows an example of how the data sections of the configured message buffers can be stored in the data partition of the message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused.

Table 24-108 Example for Structure of the Data Partition in the Message RAM

Bit Word	38 to 32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
…	E	MB0 data 3								MB0 data 2							
…	E	Unused								Unused							
…	E	MB1 data 3								MB1 data 2							
…	E	…								…							
…	E	MB1 data (k)								MB1 data (k - 1)							
…	E	…								…							
…	E	…								…							
…	E	MBx data 3								MBx data2							
…	E	…								…							
…	E	…								…							
2047	E	…								…							
2048	E	MBx data (m)								MBx data (m - 1)							

Bit Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
…	MB0 data 1								MB0 data 0							
…	MB0 data 5								MB0 data 4							
…	MB1 data 1								MB1 data 0							
…	…								…							
…	MB1 data (k - 2)								MB1 data (k - 3)							
…	…								…							
…	…								…							
…	MBx data 1								MBx data 0							
…	…								…							
…	…								…							
2047	…								…							
2048	MBx data (m - 2)								MBx data (m - 3)							

(3) ECC

There is ECC implemented in the FlexRay controller to assure the integrity of the data stored in all RAM blocks as shown in *figure24-16 ECC Generation and Check*.

The FlexRay RAMs have the ECC decoding/encoding module. When data is written to the RAMs, the local ECC encoder generates the ECC code. The ECC code is stored together with the respective data word. The ECC code is decoded each time a data word is read from the RAMs. ECC correction and error detection are performed for each word.

If an ECC error is detected in the RAMs, the respective error flag is set. The ECC error flags FLXnMHDS.FLXnDIBFE, FLXnMHDS.FLXnDOBFE, FLXnMHDS.FLXnDMRE, FLXnMHDS.FLXnDTBFAE, and FLXnMHDS.FLXnDTBFBE, and the faulty message buffer indicators FLXnMHDS.FLXnFMBDE, FLXnMHDS.FLXnMFMBE, and FLXnMHDS.FLXnFMB[6:0] are located in the message handler status register. These ECC error flags control the error interrupt flag.

The FlexRay controller performs ECC correction without notifying the host. The ECC correction does not cause the error information to be displayed in an interrupt register or a status register.

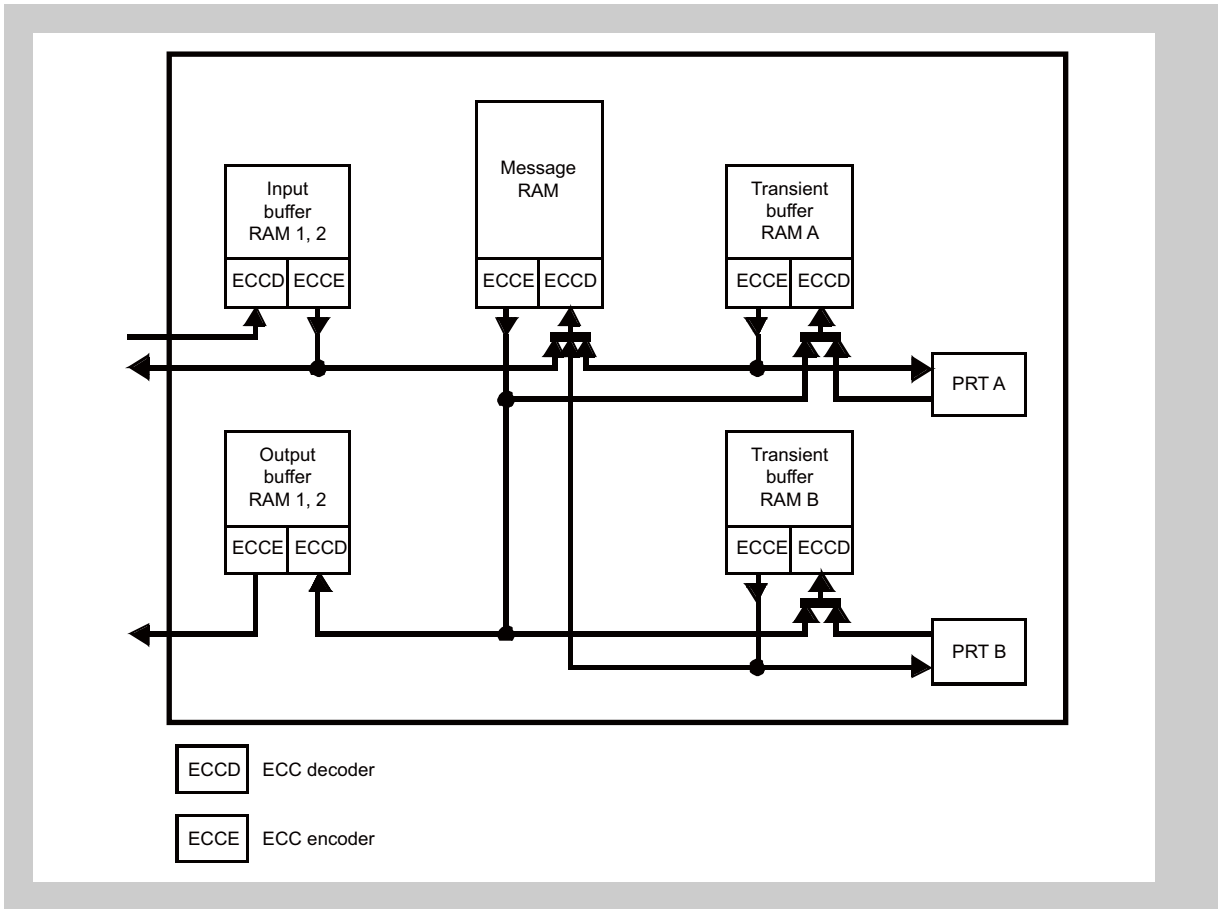


Figure 24-16 ECC Generation and Check

In the case when an ECC error has been detected, the following actions will be performed.

In all cases:

- The respective flag in the message handler status register FLXnMHDS is set.
- The ECC error interrupt is asserted.

Additionally in specific cases

1. ECC error during data transfer from input buffer RAM 1, 2 to message RAM
   a) Transfer of header and/or data section:
   - FLXnMHDS.FLXnDIBFE bit is set.
   - FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.
   - Transmit buffer: Transmission request for the respective message buffer is not set.

   b) Transfer of data section only:
   ECC error when reading header section of respective message buffer from message RAM
   - FLXnMHDS.FLXnDMRE bit is set.
   - FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.
   - The data section of the respective message buffer is not updated.
   - Transmit buffer: Transmission request for the respective message buffer is not set.

2. ECC error during host reading input buffer RAM 1, 2
   - FLXnMHDS.FLXnDIBFE bit is set.

3. ECC error during scan of header sections in message RAM
   - FLXnMHDS.FLXnDMRE bit is set.
   - FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.
   - The message buffer is ignored (skipped).

4. ECC error during data transfer from message RAM to transient buffer RAM A, B
   - FLXnMHDS.FLXnDMRE bit is set.
   - FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.
   - Frames not transmitted and frames already in transmission are invalidated by setting the frame CRC to 0.

5. ECC error during data transfer from transient buffer RAM A, B to protocol controller A, B

   – Bit FLXnMHDS.FLXnDTBFAE or bit FLXnMHDS.FLXnDTBFBE is set.

   – Frames already in transmission are invalidated by setting the frame CRC to 0.

6. ECC error during data transfer from transient buffer RAM A, B to message RAM

   a) ECC error when reading header section of respective message buffer from message RAM:

   – FLXnMHDS.FLXnDMRE bit is set.

   – FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.

   – The data section of the respective message buffer is not updated.

   b) ECC error when reading transient buffer RAM A, B:

   – Bit FLXnMHDS.FLXnDTBFAE or bit FLXnMHDS.FLXnDTBFBE is set.

   – FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.

7. ECC error during data transfer from message RAM to output buffer RAM

   – FLXnMHDS.FLXnDMRE bit is set.

   – FLXnMHDS.FLXnFMBDE bit is set to indicate that FLXnMHDS.FLXnFMB[6:0] points to a faulty message buffer.

8. ECC error during host reading output buffer RAM 1, 2

   – FLXnMHDS.FLXnDOBFE bit is set.

9. ECC error during data read of transient buffer RAM A, B

   When an ECC error occurs while the message handler is reading a frame with network management information (FLXnRHPPI = 1) from the transient buffer RAM A, B, the corresponding network management vectors NMV1 to NMV3 are not updated from that frame.

24.5.13 Module Interrupt

In general, the interrupt timing can be linked to the protocol timing as they are triggered almost immediately when an error or status change is detected by the controller, when a frame is received or transmitted, when a configured timer interrupt is activated, or when a stop watch event occurs. On the other hand, too many interrupts can cause the host to miss deadlines required for the application. Therefore, the FlexRay controller supports disable/enable controls for each individual interrupt source.

An interrupt can be triggered when:

- an error is detected
- a status flag is set
- a timer reaches a preconfigured value
- a message transfer from input buffer to message RAM or from message RAM to output buffer has completed
- a stop watch event occurs

Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the FlexRay controller. The host has access to the actual status and error information by reading registers FLXnEIR and FLXnSIR.

Table 24-109 Module Interrupt Flags and Interrupt Line Enable (1/2)

Register	Bit	Function
FLXnEIR	FLXnPEMCE	Protocol error mode changed
	FLXnCNAE	Command not valid
	FLXnSFBME	Sync frames below minimum
	FLXnSFOE	Sync frame overflow
	FLXnCCFE	Clock correction failure
	FLXnCCLE	CHI command locked
	FLXnRFOE	Receive FIFO overrun
	FLXnEFAE	Empty FIFO access
	FLXnIIBAE	Illegal input buffer access
	FLXnIOBAE	Illegal output buffer access
	FLXnMHFE	Message handler constraints flag
	FLXnEDAE	Error detected on channel A
	FLXnLTVAE	Latest transmit violation channel A
	FLXnTABAE	Transmission across boundary channel A
	FLXnEDBE	Error detected on channel B
	FLXnLTVBE	Latest transmit violation channel B
	FLXnTABBE	Transmission across boundary channel B

Table 24-109 Module Interrupt Flags and Interrupt Line Enable (2/2)

Register	Bit	Function
FLXnSIR	FLXnWSTF	Wakeup status
	FLXnCASF	Collision avoidance symbol
	FLXnCYCSF	Communication cycle start interrupt
	FLXnTXIF	Transmit interrupt
	FLXnRXIF	Receive interrupt
	FLXnRFNEF	Receive FIFO not empty
	FLXnRFCLF	Receive FIFO critical level
	FLXnNMVCF	Network management vector changed
	FLXnTI0F	Timer interrupt 0
	FLXnTI1F	Timer interrupt 1
	FLXnTIBCF	Transfer input buffer completed
	FLXnTOBCF	Transfer output buffer completed
	FLXnSWEF	Stop watch event
	FLXnSUCSF	Startup completed successfully
	FLXnMBSIF	Message buffer status interrupt
	FLXnSDSF	Start of dynamic segment
	FLXnWUPAF	Wakeup pattern channel A
	FLXnMTSAF	MTS received on channel A
	FLXnWUPBF	Wakeup pattern channel B
	FLXnMTSBF	MTS received on channel B
FLXnILE	FLXnEINTL0	Enable interrupt line 0
	FLXnEINTL1	Enable interrupt line 1

Note Among the functions in table 24-109, Module Interrupt Flags and Interrupt Line Enable, only those that are connected to the interrupt controller as per table 24-5, FLXn Interrupts, can have interrupts enabled. Enabling interrupts for other functions does not take effect.

The interrupt lines to the host, eray_int0 and eray_int1, are controlled by the enabled interrupts. In addition, each of the two interrupt lines can be enabled/ disabled separately by setting bit FLXnILE.FLXnEINTL0 and FLXnILE.FLXnEINTL1.

The two timer interrupts generated by interrupt timer 0 and 1 are available on pins eray_tint0 and eray_tint1. They can be configured via registers FLXnT0C and FLXnT1C.

The status of the data transfer between IBF/OBF and the message RAM is signaled on pins eray_ibusy and eray_obusy. When a transfer has completed, bit FLXnSIR.FLXnTIBCF or FLXnSIR.FLXnTOBCF is set.

24.6 Appendix

24.6.1 Assignment of FlexRay Configuration Parameters

Table 24-110 FlexRay Configuration Parameters (1/2)

Parameter	Bit (Field)
pKeySlotusedForStartup	FLXnSUCC1.FLXnTXST
pKeySlotUsedForSync	FLXnSUCC1.FLXnTXSY
gColdStartAttempts	FLXnSUCC1.FLXnCSA[4:0]
pAllowPassiveToActive	FLXnSUCC1.FLXnPTA[4:0]
pWakeupChannel	FLXnSUCC1.FLXnWUCS
pSingleSlotEnabled	FLXnSUCC1.FLXnTSM
pAllowHaltDueToClock	FLXnSUCC1.FLXnHCSE
pChannels	FLXnSUCC1.FLXnCCHA FLXnSUCC1.FLXnCCHB
pdListenTimeOut	FLXnSUCC2.FLXnLT[20:00]
gListenNoise	FLXnSUCC2.FLXnLTN[3:0]
gMaxWithoutClockCorrectionPassive	FLXnSUCC3.FLXnWCP[3:0]
gMaxWithoutClockCorrectionFatal	FLXnSUCC3.FLXnWCF[3:0]
gNetworkManagementVectorLength	FLXnNEMC.FLXnNML[3:0]
gdFLXnTSSTransmitter	FLXnPRTC1.FLXnTSST[3:0]
gdCASRxLowMax	FLXnPRTC1.FLXnCASM[6:0]
gdSampleClockPeriod	FLXnPRTC1.FLXnBRP[1:0]
pSamplesPerMicrotick	FLXnPRTC1.FLXnBRP[1:0]
gdWakeupSymbolRxWindow	FLXnPRTC1.FLXnRXW[8:0]
pWakeupPattern	FLXnPRTC1.FLXnRWP[5:0]
gdWakeupSymbolRxIdle	FLXnPRTC2.FLXnRXI[5:0]
gdWakeupSymbolRxLow	FLXnPRTC2.FLXnRXL[5:0]
gdWakeupSymbolTxIdle	FLXnPRTC2.FLXnTXI[7:0]
gdWakeupSymbolTxLow	FLXnPRTC2.FLXnTXL[5:0]
gPayloadLengthStatic	FLXnMHDC.FLXnSFDL[6:0]
pLatestTx	FLXnMHDC.FLXnSLT[12:00]
pMicroPerCycle	FLXnGTUC01.FLXnUT[19:00]
gMacroPerCycle	FLXnGTUC02.FLXnMPC[13:00]
gSyncNodeMax	FLXnGTUC02.FLXnSNM[3:0]
pMicroInitialOffset[A]	FLXnGTUC03.FLXnUIOA[7:0]
pMicroInitialOffset[B]	FLXnGTUC03.FLXnUIOB[7:0]
pMacroInitialOffset[A]	FLXnGTUC03.FLXnMIOA[6:0]
pMacroInitialOffset[B]	FLXnGTUC03.FLXnMIOB[6:0]
gdNIT	FLXnGTUC04.FLXnNIT[13:00]
gOffsetCorrectionStart	FLXnGTUC04.FLXnOCS[13:00]
pDelayCompensation[A]	FLXnGTUC05.FLXnDCA[7:0]
pDelayCompensation[B]	FLXnGTUC05.FLXnDCB[7:0]

Table 24-110 FlexRay Configuration Parameters (2/2)

Parameter	Bit (Field)
pClusterDriftDamping	FLXnGTUC05.FLXnCDD[4:0]
pDecodingCorrection	FLXnGTUC05.FLXnDEC[7:0]
pdAcceptedStartupRange	FLXnGTUC06.FLXnASR[10:00]
pdMaxDrift	FLXnGTUC06.FLXnMOD[10:00]
gdStaticSlot	FLXnGTUC07.FLXnSSL[9:0]
gNumberOfStaticSlots	FLXnGTUC07.FLXnNSS[9:0]
gdMinislot	FLXnGTUC08.FLXnMSL[5:0]
gNumberOfMinislots	FLXnGTUC08.FLXnNMS[12:00]
gdActionPointOffset	FLXnGTUC09.FLXnAPO[5:0]
gdMinislotActionPointOffset	FLXnGTUC09.FLXnMAPO[4:0]
gdDynamicSlotIdlePhase	FLXnGTUC09.FLXnDSI[1:0]
pOffsetCorrectionOut	FLXnGTUC10.FLXnMOC[13:00]
pRateCorrectionOut	FLXnGTUC10.FLXnMRC[10:00]
pExternOffsetCorrection	FLXnGTUC11.FLXnEOC[2:0]
pExternRateCorrection	FLXnGTUC11.FLXnERC[2:0]

24.7 Cautions

24.7.1 Loop Back Mode Operates Only at 10 MBit/s

Description The looped back data is falsified at the two lower baud rates of 5 Mbit/s and 2.5 Mbit/s.

Scope This phenomenon is limited to test cases where loop back is used with the baud rate prescaler (FLXnPRTC1.FLXnBRP[1:0]) configured to 5 or 2.5 Mbit/s.

Effects The loop back test is only possible at the highest baud rate.

Countermeasures Run loop back tests with 10 Mbit/s (FLXnPRTC1.FLXnBRP[1:0] = 00_B).

24.7.2 Noise Following a Dynamic Frame that Delays Idle Detection may Fail to Stop Slot Counting for the Remainder of the Dynamic Segment

Description If (in case of noise) the time between "potential idle start on X" and "CHIRP on X" (see Protocol Spec. v2.1, Figure 5-21) is greater than gdDynamicSlotIdlePhase, the FlexRay controller will not remain for the remainder of the current dynamic segment in the state "wait for the end of dynamic slot rx". Instead, the FlexRay controller continues slot counting. This may let the node continue to transmit data in the current dynamic segment.

Scope This phenomenon occurs only when noise is detected only locally and in the interval between the end of a dynamic frame's DTS and idle detection (CHIRP on X).

Effects In the described case, the faulty node might not stop slot counting and might continue to transmit dynamic frames. This may lead to a frame collision in the current dynamic segment.

Countermeasures None.

24.7.3 Register FLXnRCV Displays Wrong Value

Description If the calculated rate correction value is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping], vRateCorrection of the CSP process is set to zero. In this case, register FLXnRCV must be updated with this value. FLXnRCV.FLXnRCV[11:00] erroneously holds the calculated value in the range [-pClusterDriftDamping .. +pClusterDriftDamping] instead of zero.

Scope The phenomenon is limited to the case where the calculated rate correction value is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping].

Effects The displayed rate correction value FLXnRCV.FLXnRCV[11:00] is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping] instead of zero. The error of the displayed value is limited to the range of [-pClusterDriftDamping .. +pClusterDriftDamping]. This is a problem of only register display. For rate correction in the next double cycle, the correct value of zero is always used.

Countermeasures A value of FLXnRCV.FLXnRCV[11:00] in the range of [-pClusterDriftDamping .. +pClusterDriftDamping] has to be interpreted as zero.

24.7.4 After Reception of a Valid Sync Frame Followed by a Valid Non-Sync Frame in the Same Static Slot, the Received Sync Frame may be Ignored

Description If a valid sync frame followed by a valid non-sync frame is received in a static slot of an even cycle, and the frame valid detection (prt_frame_decoded_on_X) of the DEC process occurs one sclk after valid frame detection of FSP process (fsp_val_syncfr_chx), the sync frame is not taken into account by the CSP process (devte_xxs_reg).

Scope The phenomenon is limited to the case where more than one valid frame is received in a static slot of an even cycle.

Effects In the described case, the sync frame is not considered by the CSP process. This may lead to a SyncCalcResult of MISSIMG_TERM (error flag FLXnSFS.FLXnMRCS set). As a result, the POC state may switch to NORMAL_PASSIVE or HALT, or the startup procedure is aborted.

Countermeasures Avoid static slot configurations long enough to receive two valid frames.

24.7.5 Sync Frame Overflow Flag FLXnEIR.FLXnSFO may be Set If Slot Counter is Greater Than 1024

Description If the number of transmitted and received sync frames reaches gSyncNodeMax in the static segment and the slot counter in the dynamic segment reaches the value cStaticSlotIDMax + gSyncNodeMax = 1023 + gSyncNodeMax, the sync frame overflow flag FLXnEIR.FLXnSFO is set erroneously.

Scope The phenomenon is limited to configurations where the number of transmitted and received sync frames equals to gSyncNodeMax and the number of static slots plus the number of dynamic slots is greater than or equal to 1023 + gSyncNodeMax.

Effects In the described case, the sync frame overflow flag FLXnEIR.FLXnSFO is set erroneously. This has no effect to the POC state.

Countermeasures Configure gSyncNodeMax to the number of transmitted and received sync frames plus one or avoid configurations where the total of static and dynamic slots is greater than cStaticSlotIDMax.

24.7.6 Acceptance of Startup Frames Received After Reception of More Than gSyncNodeMax sync Frames

Description If a node receives in an even cycle a startup frame after it has received more than gSyncNodeMax sync frames, this startup frame is added erroneously by process CSP to the number of valid startup frames (zStartupNodes). The faulty number of startup frames is delivered to the process POC. As a consequence, this node may integrate erroneously to the running cluster because the node considers that it has received the required number of startup frames.

Scope The phenomenon is limited to the case of more than gSyncNodeMax sync frames.

Effects In the described case, a node may erroneously succeed in integrating into a running cluster.

Countermeasures Use frame schedules where all startup frames are placed in the first static slots. gSyncNodeMax must be configured to be greater than or equal to the number of sync frames in the cluster.

24.7.7 Initial Rate Correction Value of an Integrating Node is Zero If pMicroInitialOffsetA, B = 00_H

Description The initial rate correction value as calculated in Figure 8-8 of protocol spec v2.1 is zero if parameter pMicroInitialOffsetA, B is configured to be zero.

Scope The phenomenon is limited to the case where pMicroInitialOffsetA, B is configured to zero.

Effects Starting with an initial rate correction value of zero delays the adjustment of the rate correction at least 3 cycles (see figure 7-10 of protocol spec v2.1). In a worst case scenario, the cluster rate becomes too fast, making the integrating node unable to follow and abort integration.

Countermeasures Avoid configurations with pMicroInitialOffsetA, B equal to zero. If the related configuration constraint of the protocol specification results in pMicroInitialOffsetA, B equal to zero, configure it to one instead. This will lead to a correct initial rate correction value, and it will delay the startup of the node by only one microtick.

24.7.8 Incorrect Rate and/or Offset Correction Value If Second Secondary Time Reference Point (STRP) Coincides with the Action Point After Detection of a Valid Frame

Description If an STRP coincides with the action point because of a prior-action-point reception of a valid sync frame followed by a noise occurrence or the reception of a second frame, an incorrect deviation value of zero is used for further calculations of rate and/or offset correction values.

Scope The phenomenon is limited to configurations with an action point offset greater than the static frame length.

Effects In the described case, a deviation value of zero is used for further calculations of rate and/or offset correction values. This may lead to an incorrect rate and/or offset correction of the node.

Countermeasures Configure action point offset smaller than static frame length.

24.7.9 Flag FLXnSFS.FLXnMRCS is Set Erroneously Although at Least One Valid Sync Frame Pair is Received

Description In the case when the total number of different sync frames per double cycle in an odd cycle 2c+1 has exceeded gSyncNodeMax after reception of a sync frame in slot n, and the node receives, in slot n+1, a sync frame that matches with a sync frame received in the even cycle 2c, the sync frame pair is not taken into account by the CSP process. This may cause the flags FLXnSFS.FLXnMRCS and FLXnEIR.FLXnCCF to be set erroneously.

Scope The phenomenon is limited to the case of a faulty cluster configuration where different sets of sync frames are transmitted in even and odd cycles and the total number of different sync frames is greater than gSyncNodeMax.

Effects In the described case, the error interrupt flag FLXnEIR.FLXnCCF is set and the node may enter either the POC state NORMAL_PASSIVE or HALT.

Countermeasures Correct configuration of gSyncNodeMax.

24.7.10 Rate Correction Set to Zero in Case of SyncCalcResult = MISSING_TERM

Description In the case when a node receives too few sync frames for rate correction calculation and signals a SyncCalcResult of MISSING_TERM, the rate correction value is set to the last calculated value instead of to zero.

Scope The phenomenon is limited to the case of receiving too few sync frames for rate correction calculation (SyncCalcResult = MISSING_TERM in an odd cycle).

Effects In the described case, a rate correction value of zero is applied in the NORMAL_ACTIVE/NORMAL_PASSIVE state instead of the last rate correction value calculated in the NORMAL_ACTIVE state. This may lead to a resynchronization of the node although it may stay in the NORMAL_ACTIVE state (depending on gMaxWithoutClockCorrectionPassive) and decreases the probability to re-enter the NORMAL_ACTIVE state if it has switched to NORMAL_PASSIVE (pAllowHaltDueToclock = false).

Countermeasures It is recommended that gMaxWithoutClockCorrectionPassive is set to 1. If the node enters the NORMAL_PASSIVE state due to not enough number of sync frames, use a piece of upper layer application software to have the node exit from the state and start integration to the cluster. Follow the same procedure when pAllowHaltDueToclock is set to "true" and the node enters the HALT state instead of the NORMAL_PASSIVE state.

24.7.11 A Sequence of Received WUS May Generate Redundant FLXnSIR.FLXnWUPA/B Events

Description If a sequence of wakeup symbols (WUS) is received, all separated by appropriate idle phases, a valid wakeup pattern (WUP) should be detected after every second WUS. The FlexRay controller currently detects a valid wakeup pattern after the second WUS and then after each following WUS.

Scope The phenomenon is limited to the case where the application program frequently resets the appropriate FLXnSIR.FLXnWUPA/B bits.

Effects In the described case, there are more FLXnSIR.FLXnWUPA/B events seen than expected.

Countermeasures Ignore redundant FLXnSIR.FLXnWUPA/B events.

24.7.12 The Interval of Cycles after Startup Commences is Shortened by the READY Command During Startup and Normal Operation

Description An abort of startup or normal operation by a READY command near the macrotick border may lead to the effect that the state INITIALIZE_SCHEDULE is one macrotick too short during the first following integration attempt.

This accounts for shortening of the interval for the starting of cycles by INTEGRATION_COLDSTART_CHECK or INTEGRATION_CONSISTENCY_CHECK.

Scope Generation of this operation is restricted to the use of the Ready command to break out of the STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE states.

Effects Within the above scope, the integration node attempts to correct the incorrect offset in cycles of the macroticks within startup.

Countermeasures Set pOffsetCorrectionOut to a value sufficiently larger than gdMacrotick x (1+cClockDeviationMax).

24.7.13 Prohibit Issuing of the READY Command while the POC is in the Ready State

Description Issuing the READY command while the POC is in the ready state leads to setting of the FLXnCCSV.FLXnCSI (vColdStartInhibit) bit.

Scope Generation of this behavior is restricted to the ready state.

Effects The FLXnCCSV.FLXnCSI bit is set. A cold start may occur, depending on the ALLOW_COLDSTART(FLXnSUCC1.FLXnCMD[3:0]=1001) command for the POC.

Countermeasure Do not allow issuing of the READY command while the POC is in the ready state.

24.7.14 There is a Possibility of the First Wakeup Symbol (WUS) after the Reception of a Valid Wakeup Pattern (WUP) being Ignored

Description A transition to the ready state and setting of the FLXnWUPBF/FLXnWUPAF bits in the FLXnSIR register follows the reception of a normal WUP during WAKEUP_LISTEN.

If further WUSs continue to be received after that, setting of the FLXnWUPAF/FLXnWUPBF bits in the FLXnSIR is on reception of the third WUS rather than the second WUS following the transition to the READY state.

Scope Generation of this behavior is restricted to cases where redundant wakeup patterns are received.

Effects Setting of the FLXnSIR.FLXnWUPAF/FLXnWUPBF bits is delayed.

Countermeasures None

24.7.15 Initialization of the Slot Mode by the HALT or FREEZE Command is Immediately Executed

Description Issuing of a HALT or FREEZE command in the NORMAL_ACTIVE or NORMAL_PASSIVE state leads to clearing of the FLXnSLM[1:0] bits in the FLXnCCSV register instead of a transition from the HALT state to the DEFAULT_CONFIG state.

Scope Generation of this behavior is restricted to the "HALT" state.

Effects Set the slot mode to "SINGLE" on entry to the "HALT" state.

Countermeasures None

Section 25 A/D Converter

25.1 ADCA Features

This product incorporates the A/D converter (ADC).

Specifications such as the number of channels differ among products. Specifications for each product are given in the following table.

ADC \ Product		μPD70F3505A	μPD70F3506, μPD70F3507	μPD70F3508, μPD70F3509
Instance (represented by n)		1 (n = 0)		
Number of analog input pins (represented by m)		18 (m = 1 to 18)	22 (m = 1 to 22)	22 (m = 1 to 22)
Number of channel groups (represented by i)		3 (i = 0 to 2)		
Register base address	<ADCAn_base0>*1	FF81 D000$_H$		
	<ADCAn_base1>*1	FFFF DC00$_H$		
Resolution		10 or 12 bits*2		
Conversion result check		Available		
Discharge		Available		
Hardware trigger expansion		Available		
Channel sample and hold (represented by x)		6 (x = 1 to 6) Corresponds to m = 1 to 6		
Supplied clock		PCLK		

Note 1. All the register addresses related to the A/D converter that are described in this section are defined as address offsets from the base addresses given in the above table.

Note 2. The μPD70F3506 and μPD70F3508 products only support 10-bit resolution.

Caution Undefined bits should be used with the initial value settings.

The following table shows the unit to which each signal is connected.

Table 25-1 ADCA0 Interrupt Requests

Signal Name	Function	Connected to	
		Unit	Signal
INTADCA0Ti	A/D conversion end interrupt for CGi	Interrupt controller, DMA, DTS	INTADCA0Ii
INTADCA0TLLT	Latest conversion latch and timing signal	Interrupt controller	INTADCA0LLT
INTADCA0TERR	Error interrupt	Interrupt controller, DMA, DTS	INTADCA0ERR

This product has the hardware trigger function. The following table shows the units to which the hardware trigger signals ADCA0TRGi for CGi are connected.

Table 25-2 Units to which Hardware Trigger Signals are Connected (1/2)

ADCAn Channel Group	Trigger Input Signal			ADCAn Trigger Signal
	Name	Connected to		
		Unit	Signal	
CG0	ADCA0TTIN000	PIC	ADOPA0ADCATTIN00	ADCA0TTRG0
	ADCA0TTIN001	PIC	ADOPA0ADCATTIN01	
	ADCA0TTIN002	ENCA0	INTENCA0I1	
	ADCA0TTIN003	ENCA1	INTENCA1I1	
	ADCA0TTIN004	TSG20	TS0ADTRG0	
	ADCA0TTIN005	TSG20	TS0ADTRG1	
	ADCA0TTIN006	TSG21	TS1ADTRG0	
	ADCA0TTIN007	TSG21	TS1ADTRG1	
	ADCA0TTIN008	Port	ADCA0TRG0	
	ADCA0TTIN009	Not connected	–	
	ADCA0TTIN010	Not connected	–	
	ADCA0TTIN011	Not connected	–	
	ADCA0TTIN012	Not connected	–	
	ADCA0TTIN013	Not connected	–	
	ADCA0TTIN014	Not connected	–	
	ADCA0TTIN015	Not connected	–	
CG1	ADCA0TTIN100	PIC	ADOPA1ADCATTIN00	ADCA0TTRG1
	ADCA0TTIN101	PIC	ADOPA1ADCATTIN01	
	ADCA0TTIN102	ENCA0	INTENCA0I1	
	ADCA0TTIN103	ENCA1	INTENCA1I1	
	ADCA0TTIN104	TSG20	TS0ADTRG0	
	ADCA0TTIN105	TSG20	TS0ADTRG1	
	ADCA0TTIN106	TSG21	TS1ADTRG0	
	ADCA0TTIN107	TSG21	TS1ADTRG1	
	ADCA0TTIN108	Port	ADCA0TRG1	
	ADCA0TTIN109	TAPA0	TAPA0TADOUT0	
	ADCA0TTIN110	TAPA1	TAPA1TADOUT0	
	ADCA0TTIN111	Not connected	–	
	ADCA0TTIN112	Not connected	–	
	ADCA0TTIN113	Not connected	–	
	ADCA0TTIN114	Not connected	–	
	ADCA0TTIN115	Not connected	–	

Table 25-2 Units to which Hardware Trigger Signals are Connected (2/2)

ADCAn Channel Group	Trigger Input Signal			ADCAn Trigger Signal
	Name	Connected to		
		Unit	Signal	
CG2	ADCA0TTIN200	PIC	ADOPA2ADCATTIN00	ADCA0TTRG2
	ADCA0TTIN201	PIC	ADOPA2ADCATTIN01	
	ADCA0TTIN202	ENCA0	INTENCA0I1	
	ADCA0TTIN203	ENCA1	INTENCA1I1	
	ADCA0TTIN204	TSG20	TS0ADTRG0	
	ADCA0TTIN205	TSG20	TS0ADTRG1	
	ADCA0TTIN206	TSG21	TS1ADTRG0	
	ADCA0TTIN207	TSG21	TS1ADTRG1	
	ADCA0TTIN208	Port	ADCA0TRG2	
	ADCA0TTIN209	TAPA0	TAPA0TADOUT1	
	ADCA0TTIN210	TAPA1	TAPA1TADOUT1	
	ADCA0TTIN211	Not connected	–	
	ADCA0TTIN212	Not connected	–	
	ADCA0TTIN213	Not connected	–	
	ADCA0TTIN214	Not connected	–	
	ADCA0TTIN215	Not connected	–	

Note For the trigger input signals ADCA0TTINi00 and ADCA0TTINi01, the following signals are connected.

- The logical sum (OR) of INTTAUA0I0 (TAUA0INT0) to INTTAUA0I15 (TAUA0INT15) is used as the trigger input signal ADOPAiADCATTIN00.
- The logical sum (OR) of INTTAUA1I0 (TAUA1INT0) to INTTAUA1I15 (TAUA1INT15) is used as the trigger input signal ADOPAiADCATTIN01.

25.2 Functional Overview

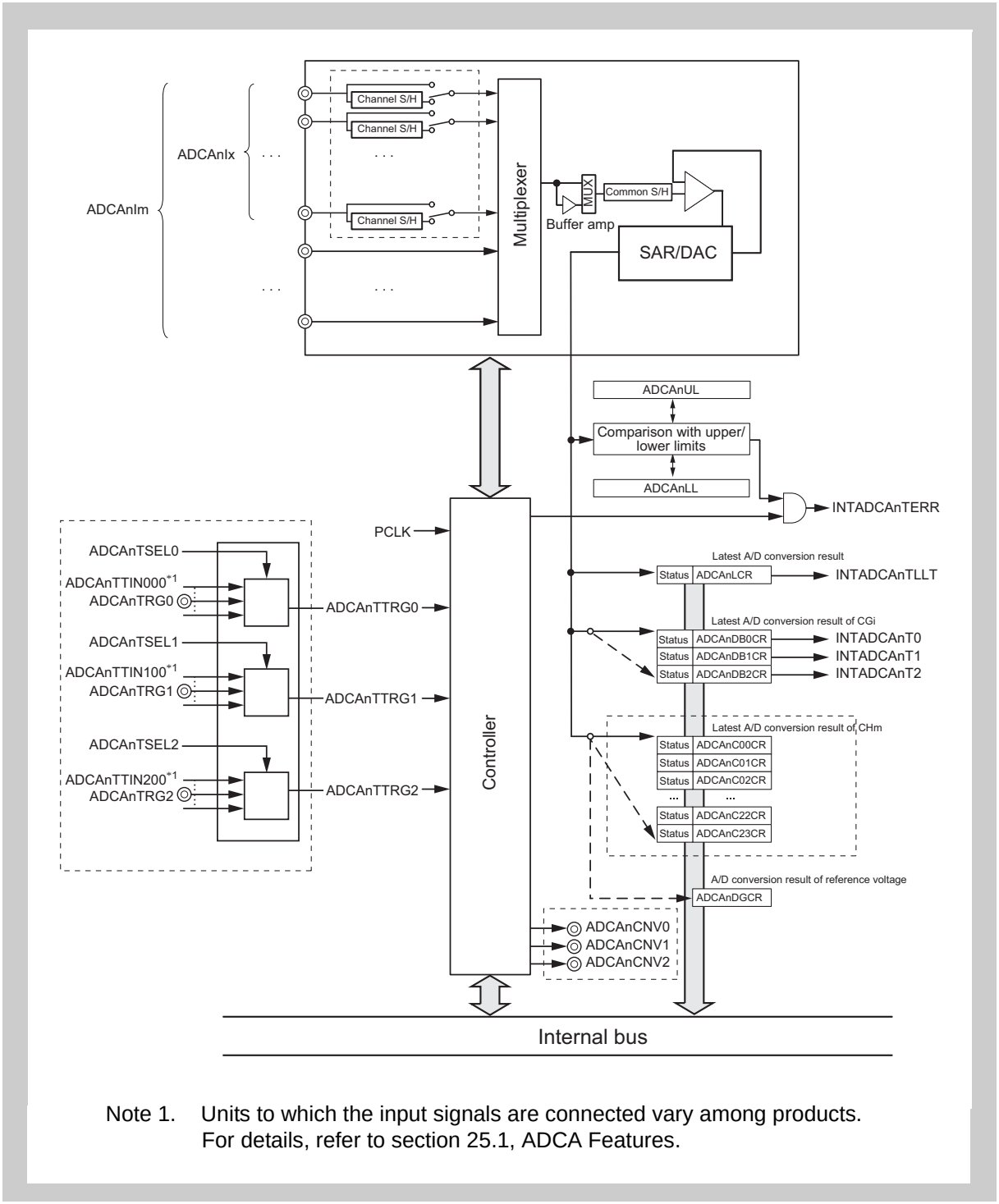
The A/D converter A (ADCAn) converts analog input signals into digital values.

Features summary

The ADCAn has the following features:

- Support of 10-bit and 12-bit resolution
- A/D conversion based on successive approximation method
- A/D conversions of up to 22 analog input signals
- A/D conversions of up to 3 differently-prioritized groups of channels
- One-shot and continuous A/D conversion modes (channel group 0 only)
- Autorepeat function (conversions repeated 1 to 4 times in one-shot mode)
- Software and hardware start trigger modes
- Selectable hardware trigger source from multiple input signals
- Configurable channels on which A/D conversion end interrupt is generated at conversion completion
- Three types of conversion result check
- Discharge function to empty the capacitor before a new sample value is stored
- On-off switch function for buffer amplifier
- Self-diagnosis functions to check the ADCA components

The following figure shows the main components of the ADCAn.



Note 1. Units to which the input signals are connected vary among products. For details, refer to section 25.1, ADCA Features.

Figure 25-1 ADCAn Block Diagram

25.3 Functional Description

The A/D converter A (ADCAn) converts up to 22 analog input signals into digital values and supports 10-bit and 12-bit resolution.

Note The setting of 10-bit or 12-bit resolution is applied to all the channels. The resolution cannot be set independently for each channel.

Channels and channel groups

Input channels can be grouped into 3 channel groups (CG). A list of input channels assigned to each CG is called the scan list (also applied to the diagnostic A/D conversion of CG0). The scan list can be easily set in 1 register and re-configured during A/D conversion. The A/D conversion of all the channels included in the scan list is called the scan list conversion.

The ADCAn supports up to 3 differently-prioritized channel groups and 2 conversion modes:

- One-shot conversion mode: The scan list conversion is performed only once. This mode allows the repeated scan list conversions for the specified number of times (1 to 4 times).
- Continuous conversion mode: The scan list conversion is performed repeatedly.

A/D conversion

The A/D conversion can be triggered by software or hardware.

A multiplexer selects the channel to be converted and the common sample and hold circuit holds the input voltage.

The successive approximation register (SAR) holds the output voltage values of the digital-to-analog converter (DAC) to be compared with the analog input voltage values, as the 10-bit or 12-bit digital values.

After each successful conversion, the INTADCAnTLLT interrupt is generated.

A/D conversion result registers

When the A/D conversion is complete, the contents of the SAR register are stored in 3 registers, allowing the latest conversion results, the latest conversion results for CGi, and the latest conversion results for channel m to be read, respectively.

Depending on the setting, the ADCAn generates a conversion end interrupt after the A/D conversion of certain channels and/or at the end of the A/D conversion of all the channels of a channel group.

Conversion result check

The ADCAn allows the A/D conversion results to be checked using the following functions.

- Overwrite check for A/D conversion result
- Read flag for A/D conversion result
- Upper/lower limit comparison for A/D conversion result

Discharge

If required, the internal capacitor of the sample and hold circuit can be discharged prior to every conversion.

On-off switch function for buffer amplifier

In order to reduce the load of the external analog signal source, the signal can be connected to an internal buffer amplifier.

The buffer amplifier accelerates charging of the internal sampling capacitor during A/D sampling.

Self-diagnosis functions

The following 4 self-diagnosis functions are provided to verify that the ADCAn works properly and to detect open analog input pins:

- Diagnosis of the A/D conversion circuit
- Diagnosis of the channel multiplexer
- Diagnosis of the analog input pins
- Diagnosis of the channel sample and hold circuit

Available stabilization time

The optimal stabilization time can be secured after power on by setting an arbitrary value to the stabilization counter.

25.3.1 Basic Operation

This section describes the basic procedure for an A/D conversion. More detailed descriptions are given in the following sections.

1. To optimize the start-up time after power on of the A/D converter (by setting ADCA0CTL1.ADCA0GPS = 1), adjust the stabilization time by modifying the setting of stabilization counter ADCAnCNT before turning the A/D converter on (see section 25.3.15, Stabilization Control).
2. Before you enable the A/D converter (ADCAnCTL0.ADCAnCE = 1), set power on, resolution, ADCAn clock, trigger mode, conversion mode, interrupt generation, channel groups, a channel's sample-and-hold function and so on in the following registers. Be sure to set ADCAnIOCi before ADCAnCGi:
   - ADCAnCTL0 register
   - ADCAnCTL1 register
   - ADCAnIOCi register
   - ADCAnCGi register
   - ADCAnTSELi register
   - ADCAnSHCTL register
3. If you want to check that the A/D conversion results are within a specified range, enable the upper/lower limit comparison function for conversion results of the desired channels (ADCAnCTL2.ADCAnRCKm) and specify the lower and upper limits in ADCAnLL and ADCAnUL.
4. If you want the capacitor of the sample and hold circuit to be discharged before sampling a new value, enable the discharge function by setting ADCAnCTL1.ADCAnDISC to 1.
5. Enable or disable the buffer amplifier by setting ADCAnCTL1.ADCAnBPC. To use a channel's sample-and-hold function, enable the buffer amplifier.
6. Enable the A/D converter by setting ADCAnCTL0.ADCAnCE to 1.

   Conversion by the A/D converter starts once the stabilization time measured by the stabilization counter (register ADCAnCNT) has elapsed after power is supplied.

If the stabilization counter is not in use, enable the A/D converter after the power-down return time (refer to section 28, Electrical Characteristics) has elapsed following the supply of power to the A/D converter (by setting ADCA0CTL1.ADCA0GPS to 1).

7. Depending on the trigger mode configured, A/D conversion is started by either of the following channel group-related start triggers.
   – Software trigger (ADCAnTRGi.ADCAnSTTi = 1)
   – Hardware trigger (input signal ADCAnTTRGi)

   If the A/D conversions of multiple CGs are triggered, the order of A/D conversions depends on the priority of the CGs.
8. The A/D conversion end interrupt INTADCAnTi is generated when the A/D conversion of channels set in ADCAnIOCi is completed.
9. Read the results from the A/D conversion result registers ADCAnLCR, ADCAnDBiCR, and ADCAnCmCR.
10. Monitor the following registers:
    – ADCAnSTR1: Check whether A/D conversion results have been overwritten before reading them out according to their use.
    – ADCAnSTR0: Check whether A/D conversion results are within the specified value range (only if the upper/lower limit comparison function for A/D conversion results has been enabled).
11. Before you reconfigure the A/D converter, disable it by clearing ADCAnCTL0.ADCAnCE to 0.

Note The self-diagnosis functions are described in section 25.3.11, Self-Diagnosis Functions.

25.3.2 Clock Usage

The ADCAn clock ADCAnTCLK is derived from PCLK. The division ratio is specified in ADCAnCTL1.ADCAnFR[3:0].

Caution The maximum and minimum frequencies of ADCAnTCLK should be 16 MHz and 2 MHz, respectively.

For frequency setting, see table 25-4, Total Conversion Time.

25.3.3 Channel and Channel Group

Input channels are grouped into channel groups (CG). The scan list for each CG can be set in a register and reconfigured during A/D conversion. The conversion settings for a CG are applied to all the channels of the CG.

The ADCAn supports up to 3 channel groups CGi (i = 0 to 2). The channels of CGi are specified in ADCAnCGi.

Note The ADCAn processes A/D conversion requests for CGs only. To convert a single channel, only 1 channel must be assigned to a CG.

(1) Order of A/D Conversion

If a trigger for a CG occurs, the channels of the CG are converted one by one in ascending order (from CH01 to CH22).

If A/D conversion requests for multiple CGs are pending, the CGs are converted in the following hierarchical order:

CG2 (highest priority) > CG1 > CG0 (lowest priority)

The current A/D conversion is interrupted when a start trigger for a higher-priority CG is set. Depending on the setting in ADCAnCTL1.ADCAnTRMi, there are 2 options:

- The A/D conversion of CG is interrupted immediately (ADCAnCTL1.ADCAnTRMi = 0).

  After the A/D conversions of all the higher-priority CGs are finished, the A/D conversion of the interrupted channel is resumed.

- The A/D conversion of the current channel is completed before the higher-priority CG is converted (ADCAnCTL1.ADCAnTRMi = 1).

  After the A/D conversions of all the higher-priority CGs are finished, the interrupted A/D conversion is resumed from the next channel.

ADCAnSTR2.ADCAnST[2:0] indicates the current conversion status of all the CGs.

Examples The following figures illustrate the different types of A/D conversion interruption;
CH3, CH9, and CH20 are assigned to CG0, CH5 and CH9 are assigned to CG2.

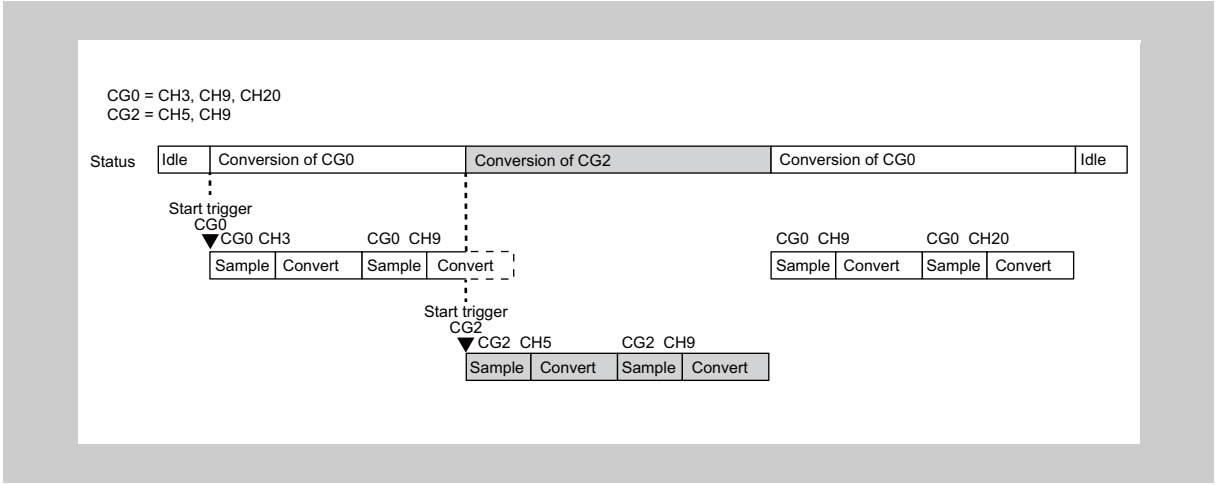


Figure 25-2 When A/D Conversion of CG0 is Immediately Interrupted (ADCAnCTL1.ADCAnTRM0 = 0)

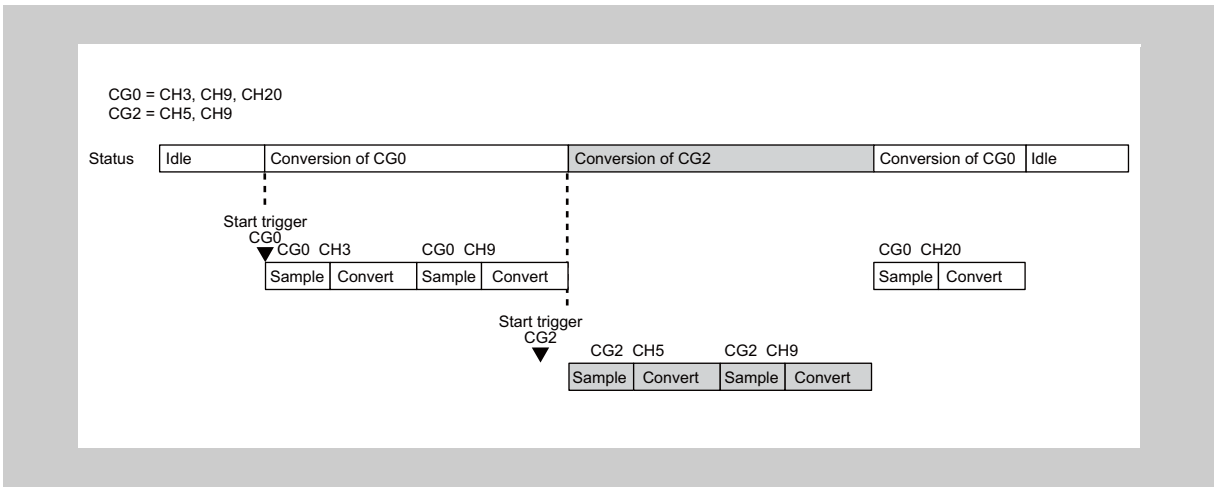


Figure 25-3 When Waiting until A/D Conversion of Current Channel is Completed (ADCAnCTL1.ADCAnTRM0 = 1)

25.3.4 A/D Conversion Modes

The A/D converter provides 2 conversion modes:

Mode	Operation	Channel Group
One-shot conversion mode	The scan list conversion is performed only once. This mode allows the repeated scan list conversions for the specified number of times (1 to 4 times).	CG0, CG1, CG2
Continuous conversion mode	The scan list conversion is performed repeatedly.	CG0

Note 1. When a running A/D conversion is interrupted by an A/D conversion request for a higher priority CG, the interrupted conversion is automatically continued when all the conversions of higher priority CGs are finished (see section 25.3.3 (1) Order of A/D Conversion).

Note 2. CG1 and CG2 are processed in one-shot conversion mode regardless of the conversion mode settings. For CG0, the A/D conversion mode can be set in ADCAnCTL1.ADCAnMD0.

(1) One-Shot Conversion Mode

In one-shot conversion mode, a start trigger causes the scan list conversion of CGi. The number of repetitions of the conversion can be set to 1 to 4 in ADCAnCTL0.ADCAnSCTi[1:0] for each CG.

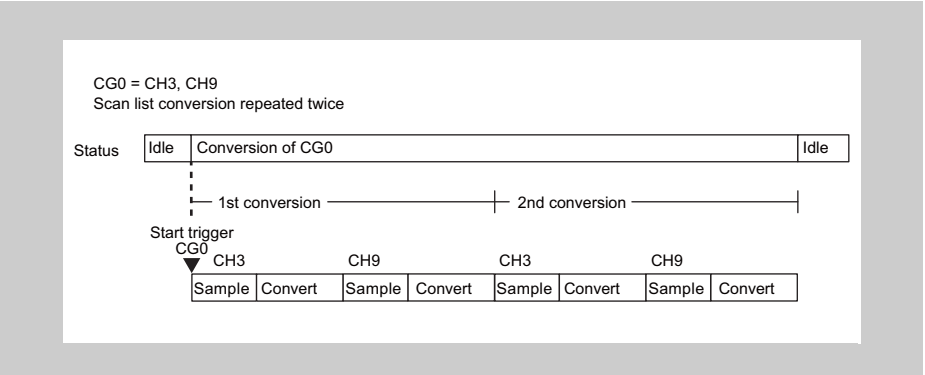


Figure 25-4 Scan List Conversion Repeated Twice in One-Shot Conversion Mode

Operation when start triggers are input before conversion completion

The A/D converter can hold one additional start trigger before conversion completion of the same CG. That means, if start triggers are set for CGi before the previously triggered A/D conversion of the CGi is completed, the first additional start trigger causes another A/D conversion to be executed immediately after the current CGi A/D conversion, but all the other additional start triggers of the same CGi are ignored.

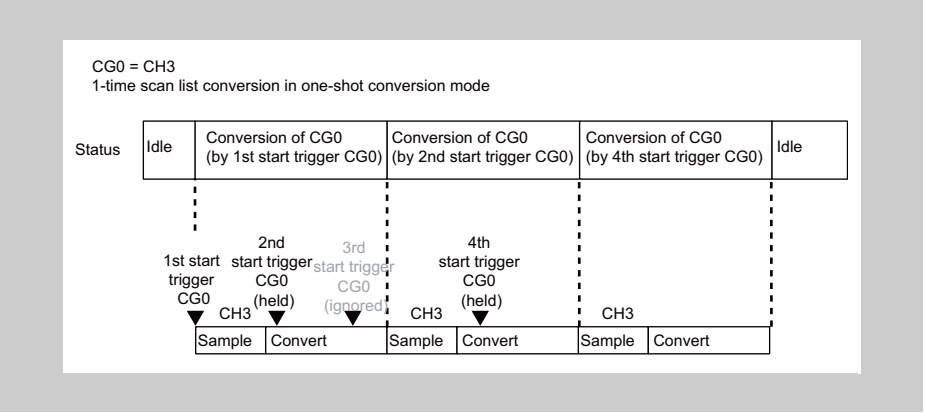


Figure 25-5 Operation when Start Triggers are Input before Conversion Completion

Caution During A/D conversions of higher-priority CGs, the start triggers before conversion completion for lower-priority CGs are ignored. In contrast, when the start triggers for lower-priority CGs are input before the start of A/D conversions of higher-priority CGs, the start triggers are held. If no start triggers are generated, acceptance is enabled even during A/D conversions of higher-priority CGs.

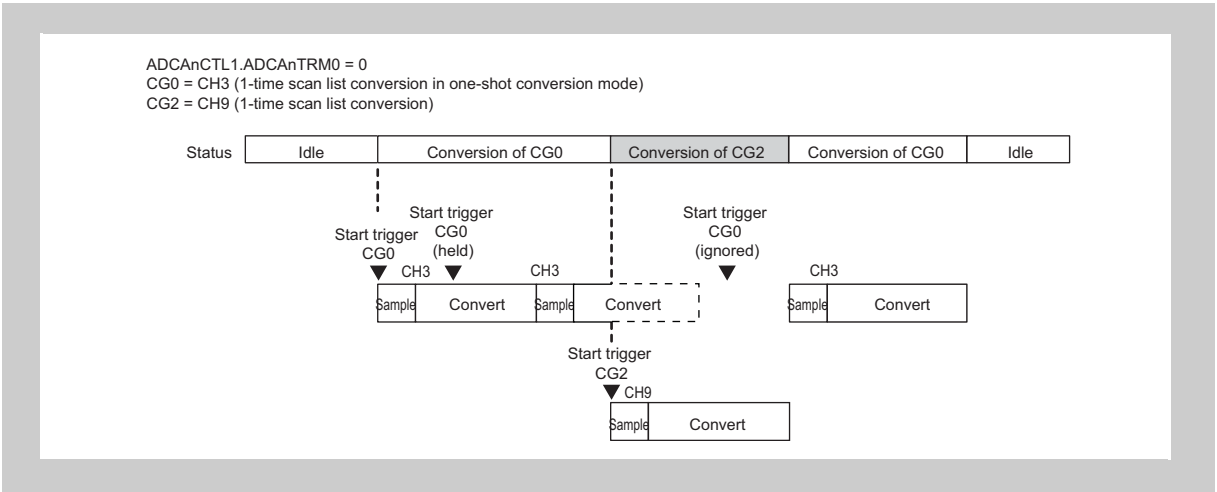


Figure 25-6 Operation when Start Triggers are Input before Conversion Completion (when Start Triggers for Higher-Priority CGs Occur)

(2) Continuous Conversion Mode

Continuous conversion mode is available for CG0 only (ADCAnCTL1.ADCAnMD0 = 1).

In continuous conversion mode, a start trigger causes the channels of CG0 to be sampled and converted continuously until a stop trigger is generated or another stop condition occurs (see section 25.3.6, Stopping A/D Conversion (Stop Trigger)).

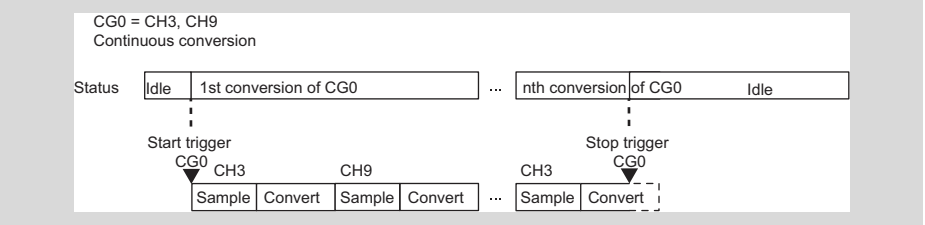


Figure 25-7 Continuous Conversion Mode

Caution The idle state is entered after a stop trigger is generated and the sampling and conversion are not performed.

Note Additional start triggers for CG0 are ignored in continuous conversion mode.

25.3.5 Starting A/D Conversion (Start Triggers)

A/D conversion can be started by a software or hardware trigger which is specified in ADCAnCTL1.ADCAnMD1.

When an A/D conversion is triggered for multiple CGs, the conversion order depends on the priorities of the CGs (see section 25.3.3 (1), Order of A/D Conversion).

Note 1. The ADCAn ignores triggers for CGi if there is no channel assigned to the CGi (ADCAnCGi = 0000 0000_H).

Note 2. In one-shot conversion mode, the A/D converter stores one start trigger.

If start triggers are set for CGi before the previously triggered A/D conversion of the CGi is completed, the first additional start trigger causes another A/D conversion to be performed immediately after the current CGi A/D conversion, but all the other additional start triggers of the same CGi are ignored (see figure 25-5, Operation when Start Triggers are Input before Conversion Completion).

Note 3. In continuous conversion mode, additional start triggers that are generated before a stop trigger is generated are ignored.

(1) Software Start Trigger

The A/D conversion of CGi is triggered by setting ADCAnTRGi.ADCAnSTTi to 1, if the A/D converter is enabled (ADCAnCTL0.ADCAnCE = 1).

Example of software start trigger timing

The following figure shows the timing of a software start trigger under the following conditions:

- ADCAnTCLK clock = PCLK/2 (ADCAnCTL1.ADCAnFR[3:0] = 0000_B)

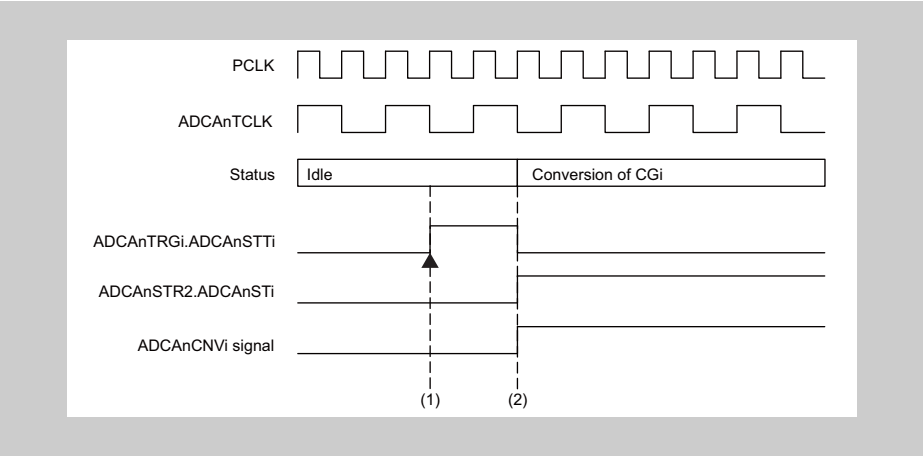


Figure 25-8 Example of Software Start Trigger Timing

(1) Software trigger for CGi is written.

(2) A/D conversion starts at the next falling edge of ADCAnTCLK.

The status signal ADCAnCNVi becomes active and the status bit ADCAnSTR2.ADCAnSTi is set to indicate that A/D conversion of CGi is running.

(2) Hardware Start Trigger

The A/D conversion of CGi is triggered by valid edge detection of signal ADCAnTTRGi, if the A/D converter is enabled (ADCAnCTL0.ADCAnCE = 1) and the hardware trigger mode is set (ADCAnCTL1.ADCAnMD1 = 1).

The valid edge is specified in ADCAnCTL1.ADCAnTiETS[1:0] for each CG.

Hardware trigger expansion

This product supports a hardware trigger expansion. Multiple hardware trigger sources can be specified for each ADCAnTTRGi input: 9 for CG0 and 11 each for CG1 and CG2. ADCAnTSELi specifies the input signals to be used as ADCATTRGi.

Note For units to which the hardware trigger signals are connected, see table 25-2, Units to which Hardware Trigger Signals are Connected.

Hardware start trigger timing

The A/D converter starts A/D conversion when a valid edge of ADCAnTTRGi is detected.

The following figure shows the timing of a hardware start trigger under the following conditions:

- ADCAnTCLK clock = PCLK/2 (ADCAnCTL1.ADCAnFR[3:0] = 0000_B)
- Valid edge of ADCAnTTRGi is rising edge (ADCAnCTL1.ADCAnTiETS[1:0] = 01_B)

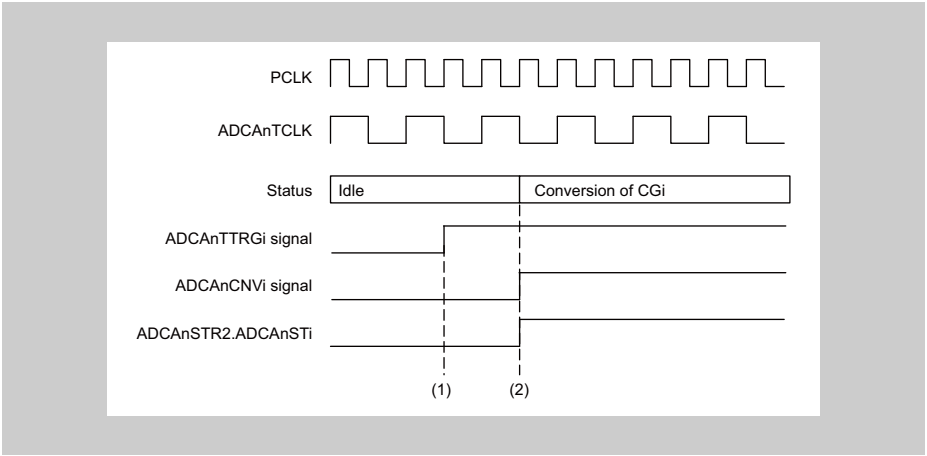


Figure 25-9 Hardware Start Trigger Timing

(1) Input signal ADCAnTTRGi rises.

(2) A/D conversion starts at the next falling edge of ADCAnTCLK.

The status signal ADCAnCNVi becomes active and the status bit ADCAnSTR2.ADCAnSTi is set to indicate that A/D conversion of CGi is running.

25.3.6 Stopping A/D Conversion (Stop Trigger)

(1) Stop Trigger

Setting the stop trigger bit for CGi (ADCAnTRG4 + i.ADCAnSPi = 1) stops the A/D conversion of CGi. The A/D conversion end interrupt signal INTADCAnTi is not generated and the A/D conversion result registers are not updated if the stop trigger is generated before the A/D conversion is completed. If a start trigger signal is input again after the stop of A/D conversion due to a stop trigger, the scan list conversion is executed from the beginning.

If the hardware trigger is being used, follow the procedure below.

1. Stop the generation of hardware triggers.
2. Set the stop trigger bit (ADCAnTRG4 + i.ADCAnSPi) to 1.
3. Check the status of ADCAnSTR2.ADCAnSTi.

If the above procedure is not followed, the A/D conversion may not be stopped due to conflict between the hardware trigger and the stop trigger.

Stop trigger timing

1. Stop trigger for CGi is written.
2. A/D conversion of CGi stops at the next falling edge of ADCAnTCLK.

   The status bit ADCAnSTR2.ADCAnSTi is cleared to indicate that A/D conversion of CGi is stopped.
   If the digital value of ADCAnIm is already available, the ADCA operates as follows:
   - All the A/D conversion result registers are updated.
   - A conversion end interrupt INTADCAnTi is generated as set in ADCAnIOCi (see section 25.3.8, Interrupt Generation).
   - According to the setting in ADCAnCTL2, the A/D conversion results are checked to see whether they are in the specified value range (see section 25.3.10, Result Check Functions).

   The A/D converter proceeds with pending A/D conversion requests for other CGs, if there are any.

The following figures show the timing of a stop trigger under the following conditions:

- ADCAnTCLK clock = PCLK/2 (ADCAnCTL1.ADCAnFR[3:0] = 0000_B)
- A conversion end interrupt INTADCAnTi is generated at the end of the A/D conversion of CGi (ADCAnIOCi = 0000 0000_H)

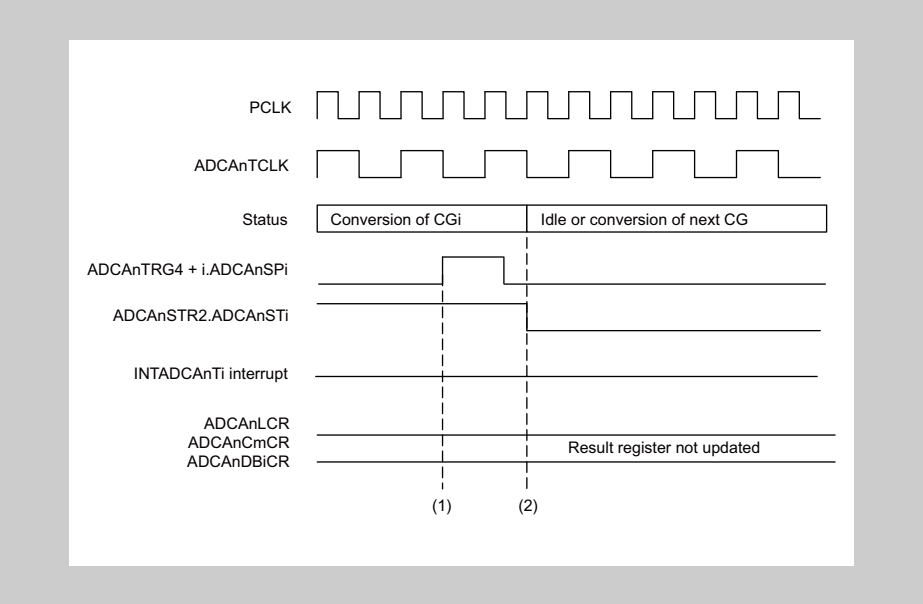


Figure 25-10 Stop Trigger Timing (before A/D Conversion Completion)

(1) The stop trigger bit (ADCAnTRG4 + i.ADCAnSPi) is set to 1.

(2) The status bit ADCAnSTR2.ADCAnSTi is cleared.

The A/D conversion end interrupt INTADCAnTi is not generated and the A/D result register is not updated.

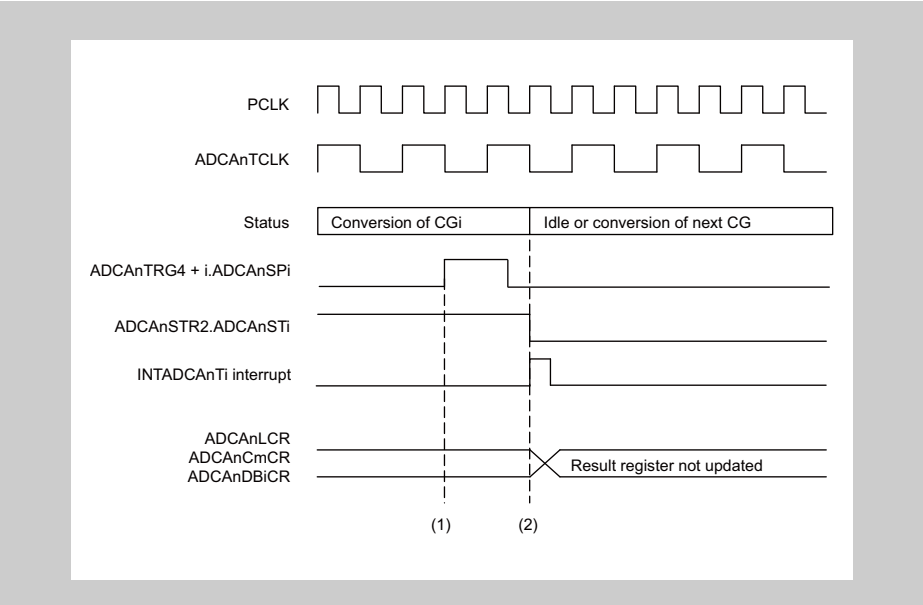


Figure 25-11 Stop Trigger Timing (after A/D Conversion Completion)

(1) The stop trigger bit (ADCAnTRG4 + i.ADCAnSPi) is set to 1.

(2) The status bit ADCAnSTR2.ADCAnSTi is cleared.

The A/D conversion end interrupt INTADCAnTi is generated and the A/D result register is updated.

(2) Other Stop Conditions

Other than a software stop trigger, the following conditions stop the A/D conversion:

- The A/D converter is powered off (ADCAnCTL1.ADCAnGPS = 0)
- The A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0)

25.3.7 Resolution, Sampling Time and Conversion Time

The total conversion time comprises sampling time and A/D conversion time.

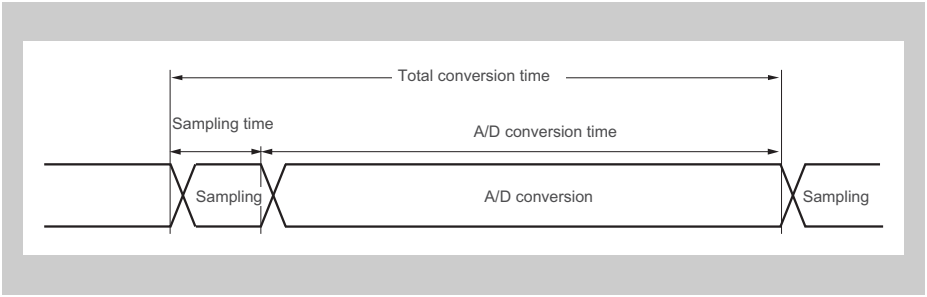


Figure 25-12 Total Conversion Time

- The sampling time is the time of connecting an analog input voltage to the sample and hold circuit. The A/D conversion time is the time required to obtain 1 digital value out of an analog input voltage.
  The A/D conversion time and the total conversion time depend on the conversion resolution, as shown below.

Table 25-3 Sampling Time and Conversion Time

Conversion Resolution	Sampling Time	Total Conversion Time
10 bits (ADCAnCTL1.ADCAnCTYP = 1)	6.5 clocks (ADCAnTCLK)	18 clocks (ADCAnTCLK)
12 bits (ADCAnCTL1.ADCAnCTYP = 0)	6.5 clocks (ADCAnTCLK)	20 clocks (ADCAnTCLK)

Note 1. When the discharge function is enabled (ADCAnCTL1.ADCAnDISC = 1), the total conversion time is increased by 1 clock cycle (ADCAnTCLK).
For details, refer to section 25.3.13, Discharge Function.

Note 2. The buffer amplifier function is enabled (ADCAnCTL1.ADCAnBPC = 1), the total conversion time is increased by 4 clock cycles (ADCAnTCLK).
For details, refer to section 25.3.14, Buffer Amplifier Function.

Table 25-4 Total Conversion Time

When resolution is 12 bits, discharge is enabled, and buffer amplifier is powered on (total conversion time for 25 clock cycles (ADCAnTCLK) (µs))

12-bit Resolution		PCLK<MHz>							
ADCAnFR3 to ADCAnFR0 bits	Clock Division	16	20	24	32	40	48	64	80
0000B	PCLK/2	3.125	2.500	2.083	1.563	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0001B	PCLK/3	4.688	3.750	3.125	2.344	1.875	1.563	Setting prohibited	Setting prohibited
0010B	PCLK/4	6.250	5.000	4.167	3.125	2.500	2.083	1.563	Setting prohibited
0011B	PCLK/5	7.813	6.250	5.208	3.906	3.125	2.604	1.953	1.563
0100B	PCLK/6	9.375	7.500	6.250	4.688	3.750	3.125	2.344	1.875
0110B	PCLK/8	Setting prohibited	10.000	8.333	6.250	5.000	4.167	3.125	2.500
1000B	PCLK/10	Setting prohibited	Setting prohibited	Setting prohibited	7.813	6.250	5.208	3.906	3.125
1010B	PCLK/12	Setting prohibited	Setting prohibited	Setting prohibited	9.375	7.500	6.250	4.688	3.750
1100B	PCLK/14	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	8.750	7.292	5.469	4.375
1110B	PCLK/16	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	10.000	8.333	6.250	5.000

When resolution is 10 bits, discharge is enabled, and buffer amplifier is powered on (total conversion time for 23 clock cycles (ADCAnTCLK) (µs))

10-bit Resolution		PCLK<MHz>							
ADCAnFR3 to ADCAnFR0 bits	Clock Division	16	20	24	32	40	48	64	80
0000B	PCLK/2	2.875	2.300	1.917	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0001B	PCLK/3	4.313	3.450	2.875	2.156	1.725	Setting prohibited	Setting prohibited	Setting prohibited
0010B	PCLK/4	5.750	4.600	3.833	2.875	2.300	1.917	Setting prohibited	Setting prohibited
0011B	PCLK/5	7.188	5.750	4.792	3.594	2.875	2.396	1.797	Setting prohibited
0100B	PCLK/6	8.625	6.900	5.750	4.313	3.450	2.875	2.156	1.725
0110B	PCLK/8	Setting prohibited	9.200	7.667	5.750	4.600	3.833	2.875	2.300
1000B	PCLK/10	Setting prohibited	Setting prohibited	9.583	7.188	5.750	4.792	3.594	2.875
1010B	PCLK/12	Setting prohibited	Setting prohibited	Setting prohibited	8.625	6.900	5.750	4.313	3.450
1100B	PCLK/14	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	8.050	6.708	5.031	4.025
1110B	PCLK/16	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	9.200	7.667	5.750	4.600

Note The conversion time in the above tables does not include the overhead time for the A/D controller processing. The following overhead time is necessary.

Acceptance of conversion start request: 1 clock cycle (ADCAnTCLK)
Initialization before the start of conversion: 1 clock cycle (ADCAnTCLK)
Conversion result storing process: 1 clock cycle (ADCAnTCLK)

25.3.8 Interrupt Generation

(1) A/D Conversion End Interrupt: INTADCAnTi

The interrupt INTADCAnTi indicates that the new A/D conversion result has been stored in the A/D conversion result register.

The A/D conversion end interrupt is generated at the A/D conversion completion of any channel of CGi which is specified in ADCAnIOCi.

If no channel is specified (ADCAnIOCi = 0000 0000_H), the interrupt INTADCAnTi is generated at the end of the A/D conversion of CGi.

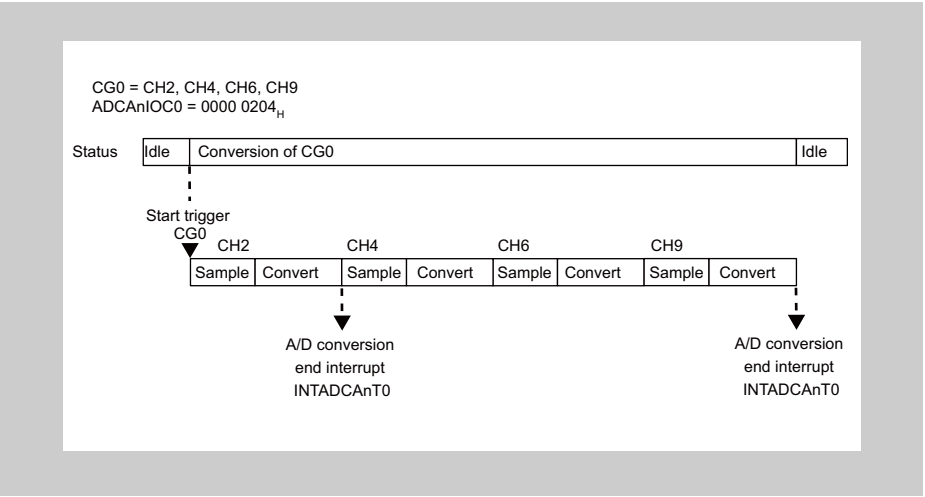


Figure 25-13 Generation of A/D Conversion End Interrupt INTADCAnTi

Note 1. ADCAnIOCi can be written at any time even when the A/D converter is enabled (ADCAnCTL0.ADCAnCE = 1). The new value takes effect after the current A/D conversion of CGi has been completed.

Note 2. ADCAnIOCi is associated with ADCAnCGi and their buffer registers must be updated simultaneously. As the update time depends on writing to ADCAnCGi, always write to ADCAnIOCi before ADCAnCGi if you want to change the interrupt generation for a CG.

(2) Error Interrupt INTADCAnTERR

The error interrupt INTADCAnTERR is generated in the following cases:

- If the A/D conversion result of a specified channel is outside the specified range with the upper/lower limit comparison for A/D conversion results enabled, refer to section 25.3.10 (3), Upper/Lower Limit Comparison for A/D Conversion Results.
- If the A/D conversion result in ADCAnLCR, ADCAnDBiCR, or ADCAnCmCR has been overwritten before it was read

  The generation of the error interrupt INTADCAnTERR when an A/D conversion result is overwritten can be controlled separately for each register by the setting of bits ADCAnCTL0.ADCAnOEM[4:0]. For details, refer to section 25.3.10 (1), Overwrite Check for A/D Conversion Results.

25.3.9 Storage of A/D Conversion Results

(1) A/D Conversion Result Registers

The A/D conversion results are stored in the following registers:

- ADCAnLCR register

  This register stores the latest A/D conversion result.

- ADCAnDBiCR register

  This register stores the latest A/D conversion result of CGi.

- ADCAnCmCR register

  This register stores the latest A/D conversion result of channel m.

These registers contain the digital values for the sampled analog input voltages in bits 15 to 00. They also provide additional information, for example, status flags allowing to check the A/D conversion result (see section 25.3.10, Result Check Functions).

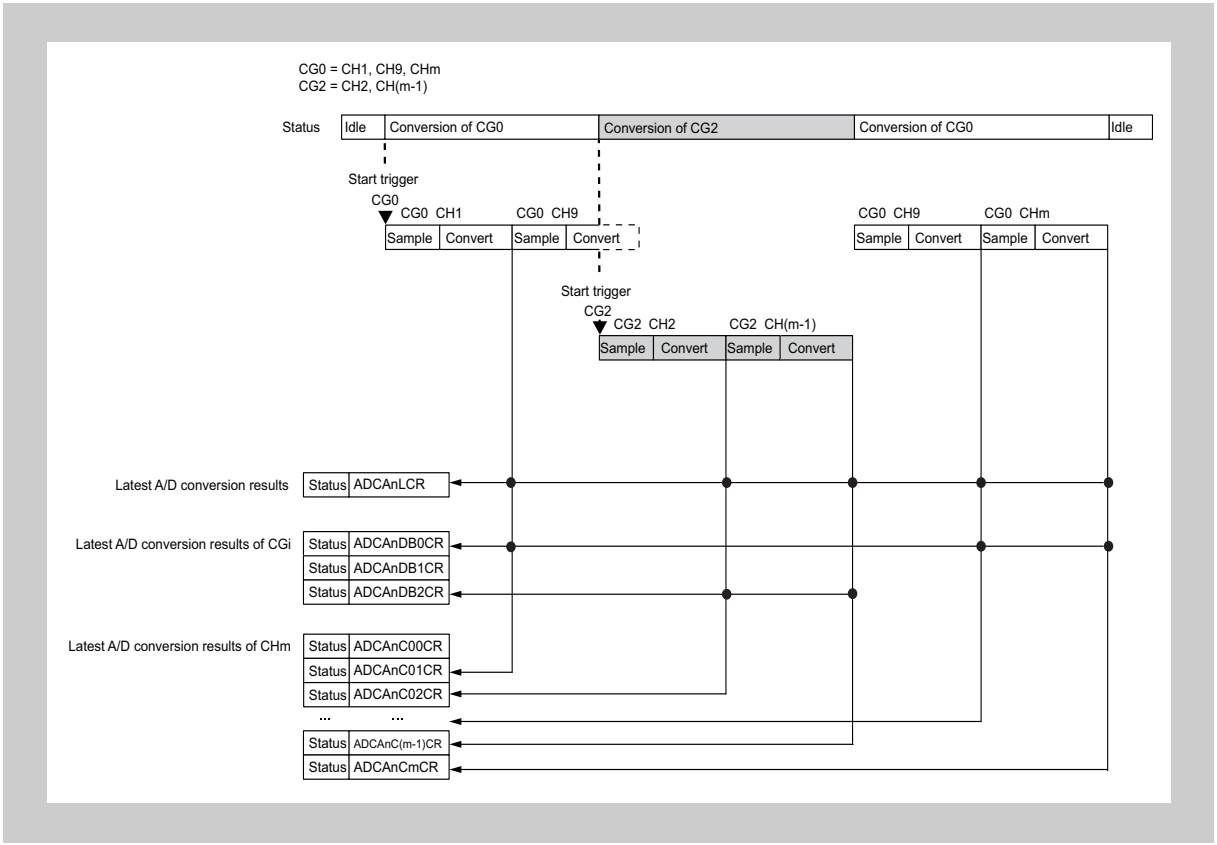


Figure 25-14 Storage of AD Conversion Results

(2) Setting about Storage of Conversion Result

(a) Alignment Specification of Conversion Result

The ADCAnCTL1.ADCAnCRAC bit can specify whether the 12-bit or 10-bit A/D conversion result is right-aligned (ADCAnCRAC = 0) or left-aligned (ADCAnCRAC = 1).

(b) Reading and Clearing Conversion Result

The ADCAnCTL1.ADCAnRCL bit can specify whether the A/D conversion result ADCAnCmCR is retained or cleared after being read out.

(3) Relationship between Analog Input Voltage and A/D Conversion Result

There is a relationship between the analog input voltages that are input to the analog input pin (ADCAnIm) and the A/D conversion result values (values of the ADCAnLCR[15:00] bits, ADCAnCmCR[15:00] bits and ADCAnDBiCR[15:00] bits), as shown in the following equations.

$$\text{A/D conversion result value} = \text{INT}\left(\frac{V_{IAN} - AV_{REFnM}}{AV_{REFnP} - AV_{REFnM}} \times 2^k + 0.5\right)$$

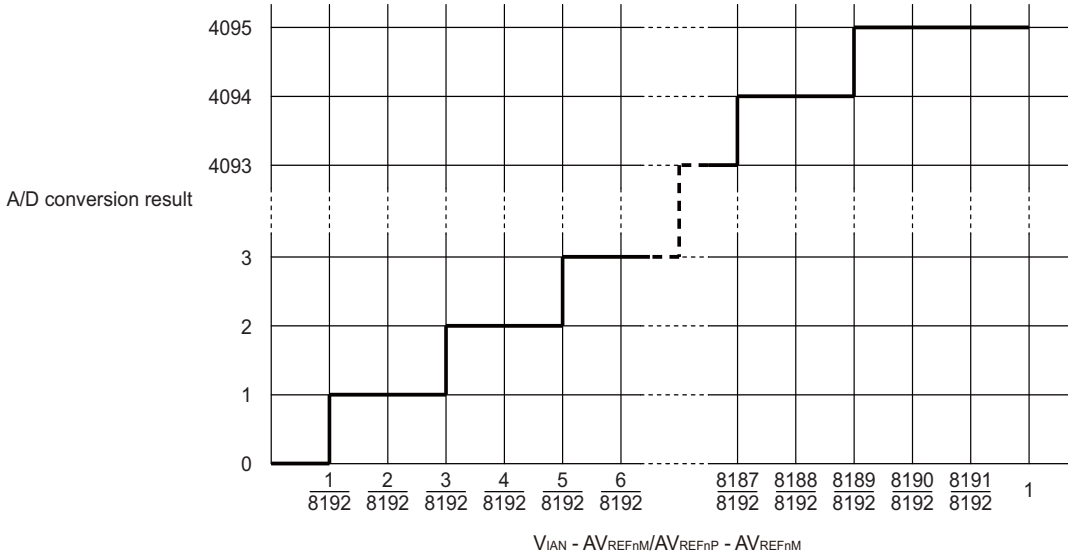
or

$$(\text{A/D conversion result value} - 0.5) \times \frac{AV_{REFnP} - AV_{REFnM}}{2^k} \leq V_{IAN} - AV_{REFnM} < (\text{A/D conversion result value} + 0.5) \times \frac{AV_{REFnP} - AV_{REFnM}}{2^k}$$

INT( ):	A function which returns the integer part of the value in parentheses
V_{IAN}:	Analog input voltage
AV_{REFnP}:	AV_{REFnP} pin voltage
AV_{REFnM}:	AV_{REFnM} pin voltage
A/D conversion results:	Values of the ADCAnLCR[15:00] bits, ADCAnCmCR[15:00] bits and ADCAnDBiCR[15:00] bits
k:	Resolution

Figure 25-15 shows the relationship between the analog input voltages and the A/D conversion result values.

(i) When the conversion characteristics of 12-bit A/D converter are applied (ADCAnCTL1.ADCAnCTYP = 0)



(ii) When the conversion characteristics of 10-bit A/D converter are applied (ADCAnCTL1.ADCAnCTYP = 1)

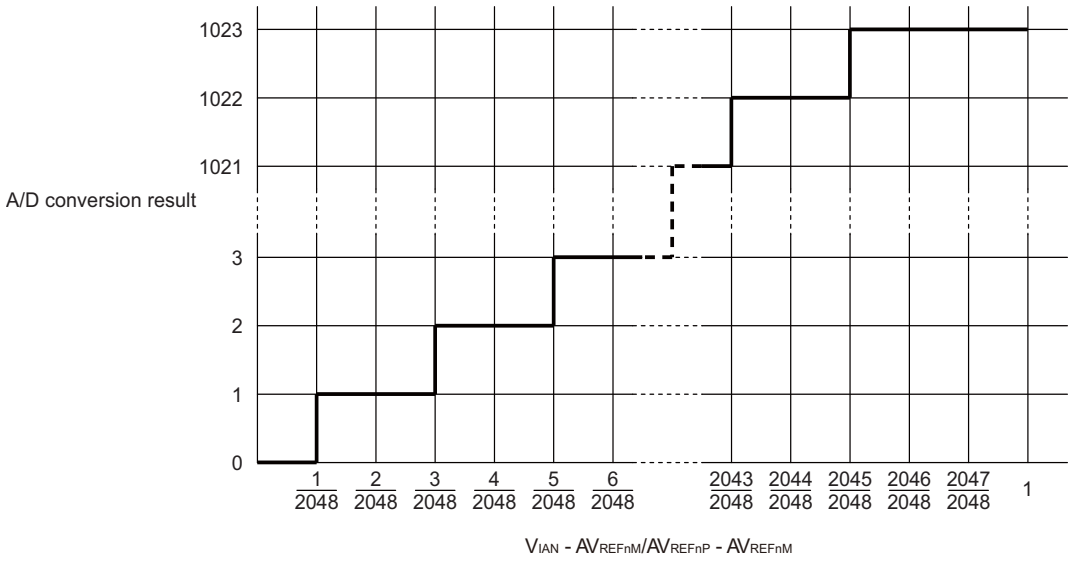


Figure 25-15 Relationship between Analog Input Voltages and A/D Conversion Results

25.3.10 Result Check Functions

The ADCAn provides the following functions to check the A/D conversion results:

- Overwrite check for A/D conversion results
- Read flag for A/D conversion results
- Upper/lower limit comparison for A/D conversion results

(1) Overwrite Check for A/D Conversion Results

The ADCAn can check if an A/D conversion result has been overwritten before it is read.

Error flags

Each of the A/D conversion result registers provides the overwrite error flag as follows:

ADCAnLCR.ADCAnLER1
ADCAnDBiCR.ADCAnDBiER1
ADCAnCmCR.ADCAnCmER1

For example, if the A/D conversion result stored in ADCAnCmCR has been overwritten before it is read, ADCAnCmCR.ADCAnCmER1 is set to 1.

The same applies to ADCAnLCR and ADCAnDBiCR.

The value of ADCAnCmCR.ADCAnCmER1 is also reflected in ADCAnSTR1.ADCAnOWEm.

Error interrupt

The error interrupt INTADCAnTERR is generated if an A/D conversion result in ADCAnLCR, ADCAnDBiCR, or ADCAnCmCR is overwritten before it is read.

For the conversion result register that the stored conversion result is not read out, an error interrupt should be masked by appropriately setting ADCAnCTL0.ADCAnOEM[4:0]. (Do not perform the overwrite check for A/D conversion result.)

Unless an error interrupt is masked, the ADCAn does not detect reading of the conversion result. In this case, the ADCAn detects that the result is overwritten at storage of the next A/D conversion result, and the error interrupt INTADCAnTERR occurs.

(2) Read Flag for A/D Conversion Results

The ADCAn can check whether an A/D conversion result has already been read, or not yet been read.

Update status flag

Each of the A/D conversion result registers provides the update status flag as follows.

ADCAnLCR.ADCAnLUR
ADCAnDBiCR.ADCAnDBiUR
ADCAnCmCR.ADCAnCmUR

If these flags are set to 1, the updated A/D conversion result has not been read yet. The update status flag is cleared to 0 after the A/D conversion result is read.

(3) Upper/Lower Limit Comparison for A/D Conversion Results

The ADCAn can check if A/D conversion results lie within a configurable value range.

This function can be enabled or disabled for each channel in ADCAnCTL2.

For the enabled channel, the A/D conversion result is compared with the specified lower limit (in ADCAnLL) and upper limit (in ADCAnUL).

Error flag

If the A/D conversion result of a specified channel is below the lower limit or above the upper limit, the corresponding error flag ADCAnSTR0.ADCAnRCE is set to 1.

ADCAnSTR0 indicates the error status of upper/lower limit comparison for the latest A/D conversion result for every channel. This register can be used to check which A/D conversion results are outside the specified range.

The value of the result check error flag ADCAnSTR0.ADCAnRCE is also reflected in ADCAnLCR.ADCAnLER0, ADCAnDBiCR.ADCAnDBiER0 and ADCAnCmCR.ADCAnCmER0.

Error interrupt

The error interrupt INTADCAnTERR is generated if an A/D conversion result of a specified channel is outside the specified range.

25.3.11 Self-Diagnosis Functions

The following self-diagnosis functions can be used to verify that the ADCAn works properly:

(1) Diagnosis of A/D conversion circuit

(2) Diagnosis of channel multiplexer

(3) Diagnosis of analog input pins

(4) Diagnosis of channel sample and hold circuit

The figure below outlines the self-diagnosis functions, which are explained in detail in the following sections.

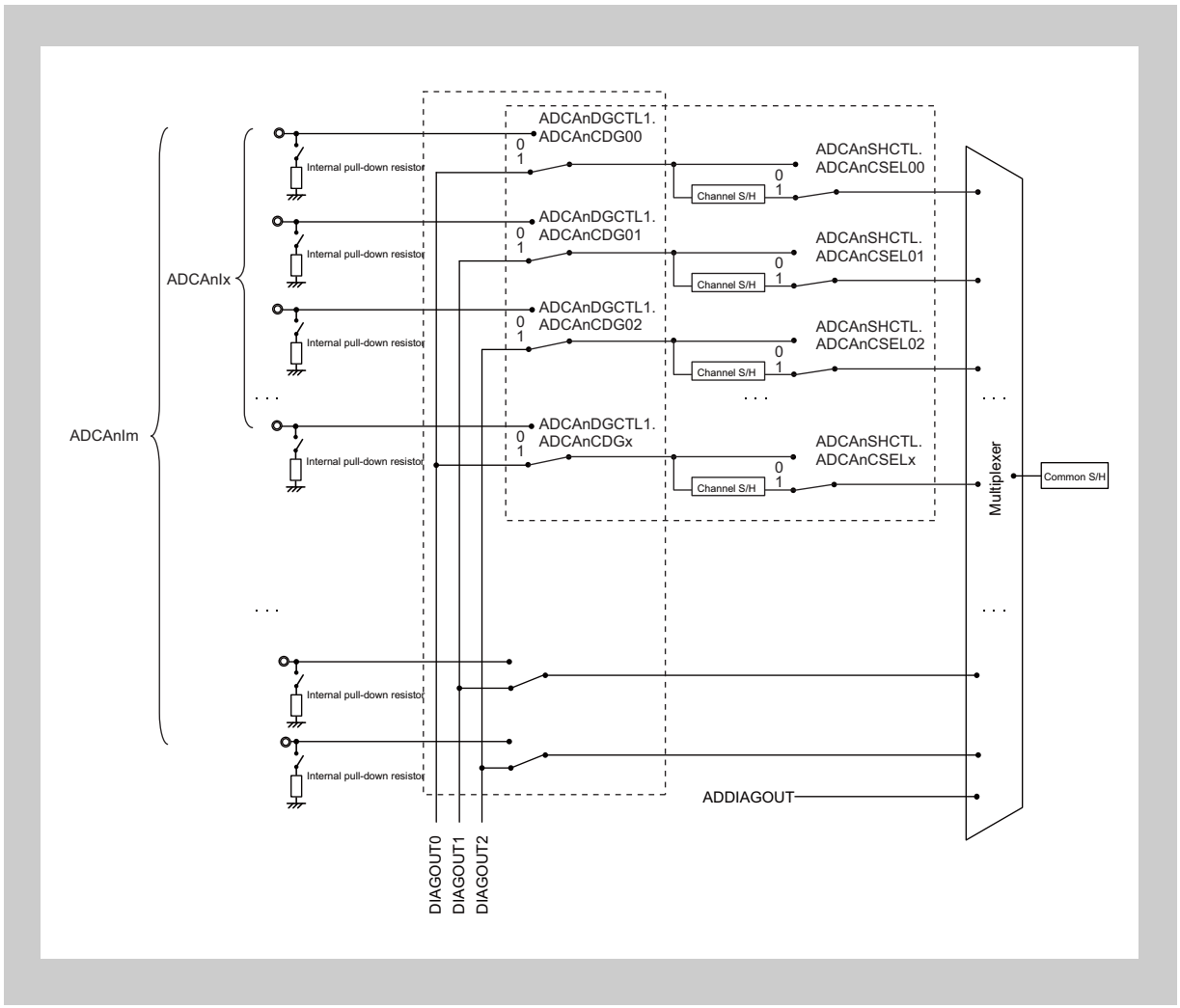


Figure 25-16 Outline of Self-Diagnosis Functions

The following table shows the register settings with which each diagnosis function is available.

Table 25-5 Self-Diagnosis Function Settings

Self-Diagnosis Function	ADCAnCE Bit = 0	ADCAnCE Bit = 1
Diagnosis of A/D conversion circuit	Available	Available
Diagnosis of channel multiplexer	Available	Unavailable
Diagnosis of analog input pins	Available	Unavailable
Diagnosis of channel sample and hold circuit	Available	Available

(1) Diagnosis of A/D Conversion Circuit

The ADCAn can diagnose operation of the A/D conversion circuit.

The A/D conversion circuit can be diagnosed during normal A/D conversion operation. The ADCAn additionally converts the reference voltage ADDIAGOUT after the A/D conversion of CG0. If the result of this diagnostic A/D conversion differs greatly from the expected value, the hardware may have had an error or malfunction.

The diagnostic A/D conversion is enabled by setting ADCAnCG0.ADCAnDIAG to 1.

Note The diagnosis of A/D conversion circuit is available for CG0 only.

The diagnostic A/D conversion is started after the A/D conversion of the last channel of CG0 is completed.

- The A/D conversion results of CG0 are stored in the normal A/D conversion result register (see section 25.3.9 (1), A/D Conversion Result Registers).
- The diagnostic A/D conversion result is stored in ADCAnDGCR.

Diagnosis procedure

1. Power on the ADCAn by setting ADCAnCTL1.ADCAnGPS to 1.
2. Set CG0 and the A/D conversion mode as follows:
   – Set ADCAnCG0.ADCAnDIAG to 1 to enable the diagnostic A/D conversion of the reference voltage.
   For example, write 8000 000E$_H$ to first convert the analog input voltages of CH1, CH2, and CH3, and then convert the reference voltage signal ADDIAGOUT for diagnosis.
   – Set ADCAnIOC0.ADCAnCG0IDG to 1 to generate the A/D conversion end interrupt INTADCAnT0 at the end of the diagnostic A/D conversion.
3. Specify the reference voltage signal ADDIAGOUT in ADCAnDGCTL0.ADCAnPSEL[2:0].
   For example, set ADCAnDGCTL0.ADCAnPSEL[2:0] to 010$_B$ to use the reference voltage 1/2 AV_{DD}.
4. Enable the ADCAn by setting ADCAnCTL0.ADCAnCE to 1.
5. Generate software or hardware start trigger to start the A/D conversion.
6. When the A/D conversion end interrupt INTADCAnT0 occurs, read the A/D conversion results of the diagnostic A/D conversion in ADCAnDGCR.

Bits ADCAnDGCTL0.ADCAnPSEL[2:0] are writable even during A/D conversion. Operation when writing is performed during A/D conversion is shown in the figure below.

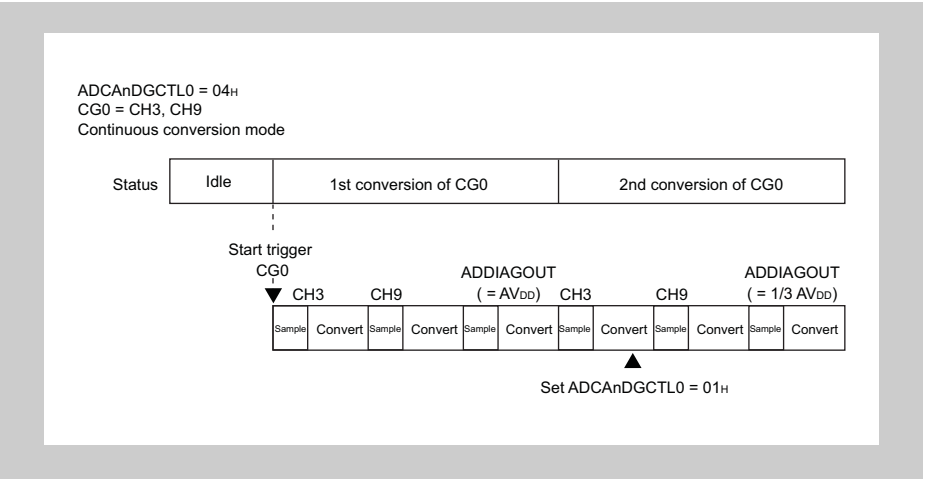


Figure 25-17 Writing during A/D Conversion

Note The value set in ADCAnDGCTL0.ADCAnPSEL[2:0] becomes valid after completion of current channel conversion. Therefore, set the reference voltage of the next diagnostic A/D conversion by the start of the conversion.

(2) Diagnosis of Channel Multiplexer

The ADCAn can diagnose if the switching and selection of channels work properly.

Each channel is assigned 1 of 3 reference voltages. The reference voltage values can be altered using ADCAnDGCTL0.ADCAnPSEL[2:0].

Table 25-6 Assignment of Reference Voltages to Channels

Reference Voltage	Channel
DIAGOUT0	21, 18, 15, 12, 9, 6, 3
DIAGOUT1	22, 19, 16, 13, 10, 7, 4, 1
DIAGOUT2	20, 17, 14, 11, 8, 5, 2

To diagnose the channel multiplexer, different reference voltages can be input to the channels.

Diagnosis procedure

1. Power on the ADCAn by setting ADCAnCTL1.ADCAnGPS to 1.
2. Specify the channels of CG0 in ADCAnCG0.
   For example, write 0000 000E$_H$ to use CH1, CH2, and CH3 for diagnosis.
3. Set ADCAnIOC0 to 0000 0000$_H$ to generate the A/D conversion end interrupt INTADCAnTi at the end of all the A/D conversions of CG0.
4. Set other A/D conversions as necessary.
5. Specify the reference voltages in ADCAnDGCTL0.ADCAnPSEL[2:0].
   For example, set ADCAnDGCTL0.ADCAnPSEL[2:0] to 010$_B$ to use the following reference voltages.
   – DIAGOUT0 = 1/2 AV$_{DD}$
   – DIAGOUT1 = 2/3 AV$_{DD}$
   – DIAGOUT2 = 1/3 AV$_{DD}$
6. Set which channel of CG0 a reference voltage (instead of the analog input voltage ADCAnIm) is applied to in ADCAnDGCTL1. For example, when ADCAnDGCTL1 is set to 0000 0006$_H$, the reference voltages are applied to the channels as follows:
   – DIAGOUT1 (2/3 AV$_{DD}$) is applied to CH1.
   – DIAGOUT2 (1/3 AV$_{DD}$) is applied to CH2.
   – Analog input voltage is applied to CH3.
7. Enable the ADCAn by setting ADCAnCTL0.ADCAnCE to 1.
8. Generate software or hardware start trigger to start the A/D conversion.
9. When the A/D conversion end interrupt INTADCAnT0 occurs, read the A/D conversion results of CG0.

Caution When ADCAnCTL0.ADCAnCE is set to 1, changing the ADCAnDGCTL1 register setting is prohibited.

(3) Diagnosis of Analog Input Pins

Open input pins result in erroneous A/D conversion results.

The internal pull-down resistors can be connected to diagnose the analog inputs ADCAnIm.

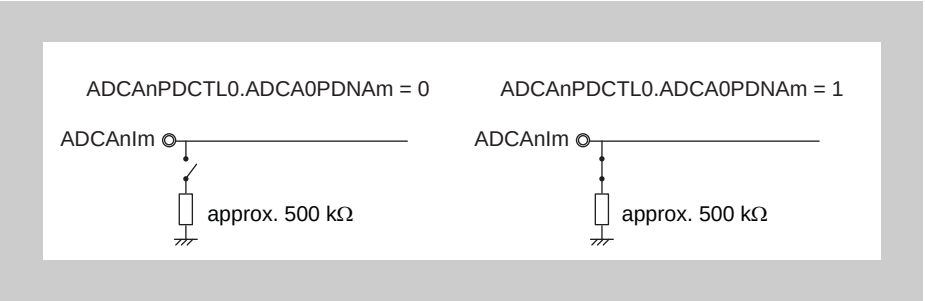


Figure 25-18 Internal Pull-down Resistors

When an internal pull-down resistor is connected to the analog input ADCAnIm (ADCAnPDCTL0.ADCAnPDNAm = 1) and the ADCAnIm connection is open, the A/D conversion result approaches to 0 V.

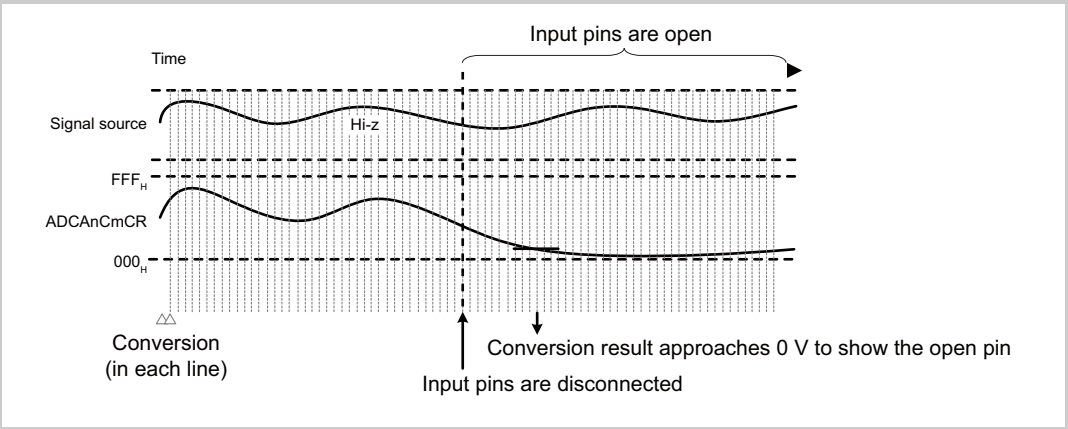


Figure 25-19 Detection of Open Input Pins

Note 1. If the analog input ADCAnIm is below 0.2 V, whether or not an input pin is open cannot be distinguished (refer to Section 28 Electrical Characteristics).

Note 2. The pull-down resistors must not be connected during normal A/D conversion operation. Connected pull-down resistors may lead to a drop in the input voltage and result in erroneous A/D conversion results.

Diagnosis procedure

The procedure for diagnosing open input pins is given below.

1. Set the CG and A/D conversion as usual.
2. Connect an internal pull-down resistor after setting ADCAnPDCTL0.
3. Perform A/D conversion multiple times.
4. Monitor the channel's A/D conversion results and check if any result declines to 0 V.

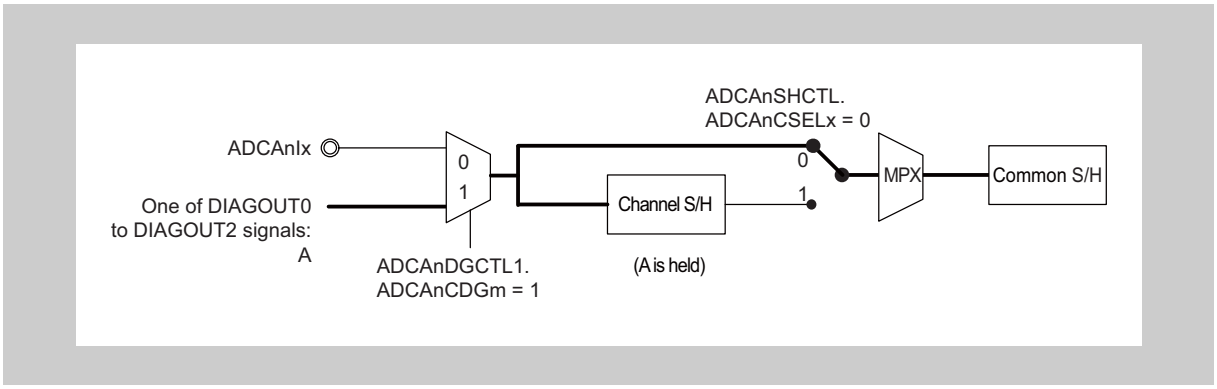
(4) Diagnosis of Channel Sample and Hold Circuit

For the channel sample and hold function, refer to section 25.3.12, Channel Sample and Hold Function.

The ADCAn can diagnose the channel sample and hold circuit. The outline is shown below.

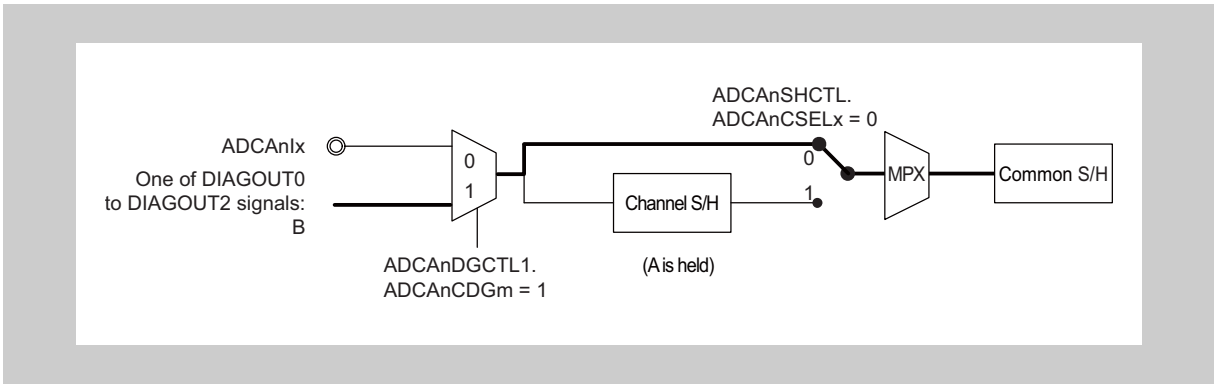
(1) One of the reference voltage signals DIAGOUT0 to DIAGOUT2 is used.

(2) A reference voltage A is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The channel sample and hold circuit holds the voltage A and an A/D conversion is performed without using the channel sample and hold circuit.



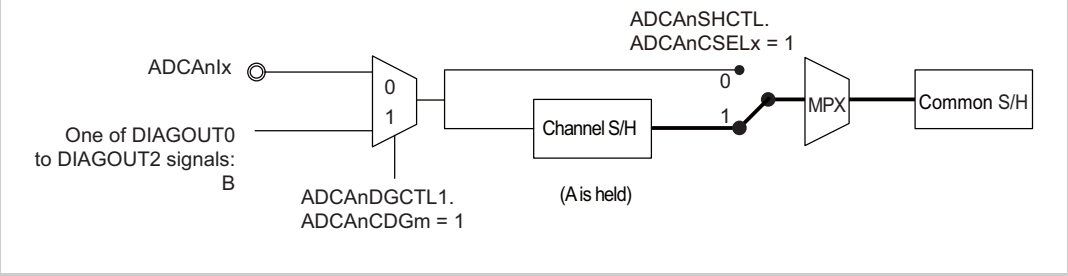
(3) A reference voltage B is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2.

An A/D conversion is performed without using the channel sample and hold circuit. The conversion result is B.
The channel sample and hold circuit continues to hold the voltage A.



(4) An A/D conversion is performed using the channel sample and hold circuit.

The channel sample and hold circuit continues to hold the voltage A. The conversion result is A.



(5) The conversion results are as follows:

- The first result value (step 3) is B.
- The second result value (step 4) is A.

The following shows the diagnosis flow.

Initial setting

When the ADCAnI1 pin is used:

- Set ADCAnCTL1. ADCAnGPS to 1.
- Specify a software trigger as the start trigger and one-shot conversion mode (repetition times: 1) as the A/D conversion mode.
- Set ADCAnCG0 to 0000 0002_H (to select the ADCAnI1 pin connected to the channel sample and hold circuit).
- Set ADCAnIOC0 to 0000 0002_H to generate the A/D conversion end interrupt INTADCAnT0 (this interrupt must be generated at the end of A/D conversion).
- Set ADCAnDGCTL0. ADCAnPSEL[2:0] to 001_B (to select $1/3AV_{DD}$ for the reference voltage signal DIAGOUT1).
- Set ADCAnDGCTL1. ADCAnCDG01 to 1.
- Set ADCAnCTL0. ADCAnCE to 1.

Operation flow

[Step 1]

Input the first software start trigger for CG0 (the channel sample and hold circuit holds 1/3AV_{DD}).

Then change the ADCAnPSEL[2:0] bit setting to 010_B (after completion of the first A/D conversion, 2/3AV_{DD} is selected for the reference voltage signal DIAGOUT1).

Input the second software start trigger for CG0 (this start trigger is held).

The entire first step must be performed by the end of the first A/D conversion.

[Step 2]

When the first A/D conversion is completed, the A/D conversion end interrupt INTADCAnT0 is generated.

Change the ADCAnCSEL01 bit setting to 1 (the reference voltage held by the channel sample and hold circuit is converted at the next A/D conversion).

Input the third software start trigger for CG0 (this start trigger is held).

The entire second step must be performed by the end of the second A/D conversion.

[Step 3]

When the second A/D conversion is completed, the A/D conversion end interrupt INTADCAnT0 is generated.

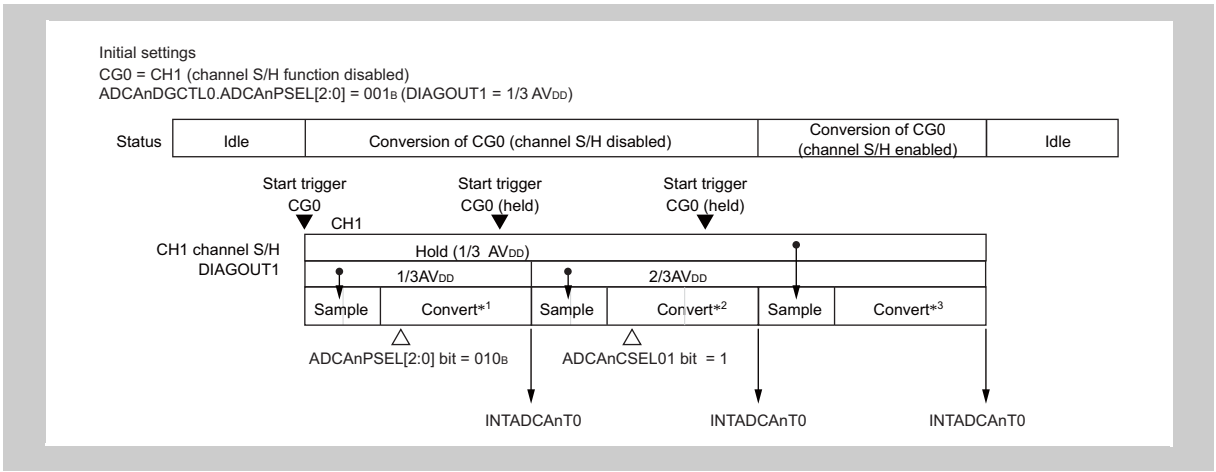
Read ADCAnC01CR and check if the expected value (2/3AV_{DD}) has been stored.

[Step 4]

When the third A/D conversion is completed, the A/D conversion end interrupt INTADCAnT0 is generated.

Read ADCAnC01CR and check if the expected value (1/3AV_{DD}) has been stored.

A timing example of diagnosis of the channel sample and hold circuit is shown below.



Note 1. Convert DIAGOUT1. The conversion result is 1/3 AV_{DD}.

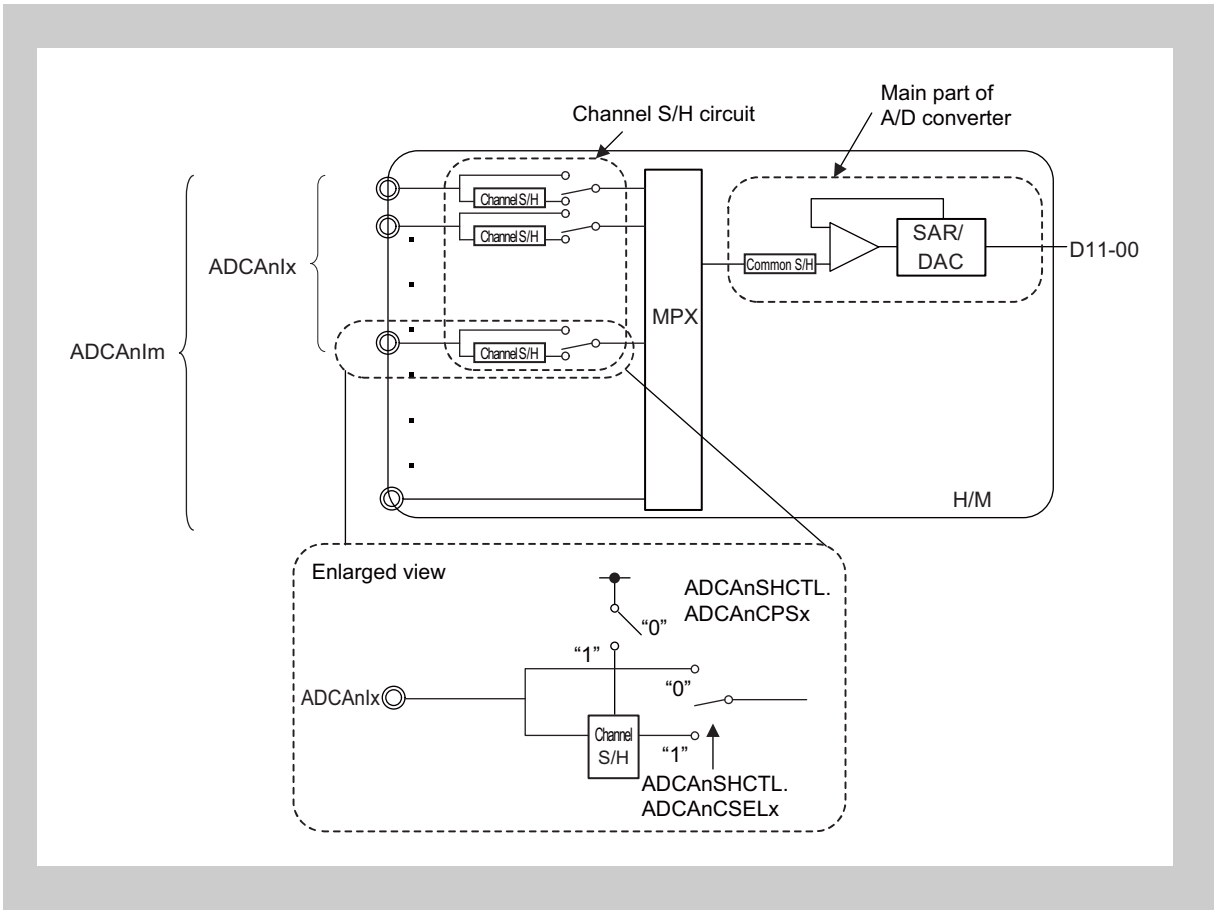
Note 2. Convert DIAGOUT1. The conversion result is 2/3 AV_{DD}.

Note 3. Convert the input voltage held by channel S/H. The conversion result is 1/3 AV_{DD}.

25.3.12 Channel Sample and Hold Function

(1) Channel Sample and Hold Function

The channel sample and hold circuit is shown below.

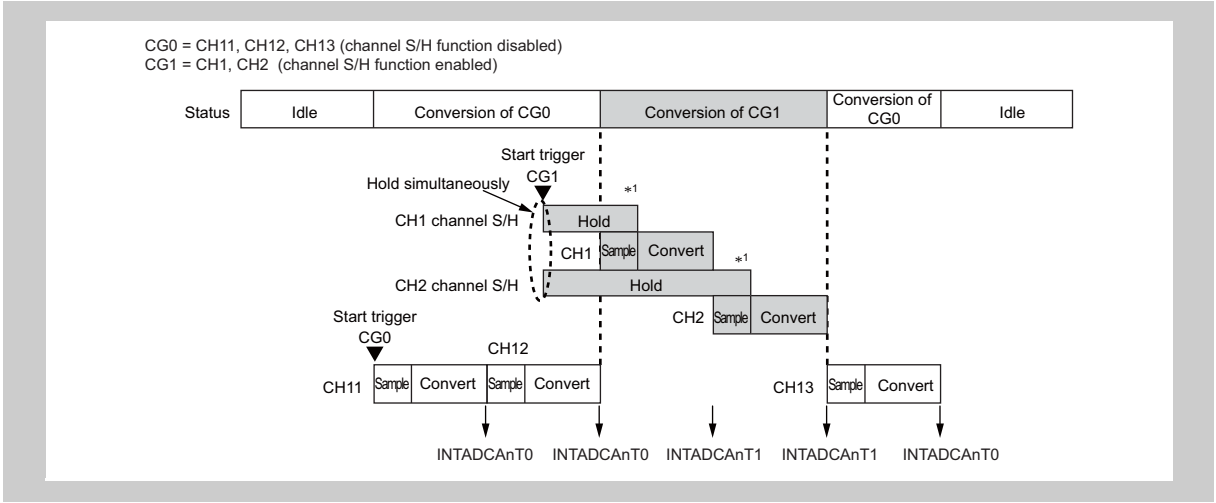


The channel sample and hold circuit can be used in one-shot conversion mode for CG0, CG1 and CG2 (no repetition).

The ADCAnSHCTL.ADCAnCPSx bit is used to control power supply to the channel sample and hold circuit and the ADCAnSHCTL.ADCAnCSELx bit is used to enable or disable the channel sample and hold function.

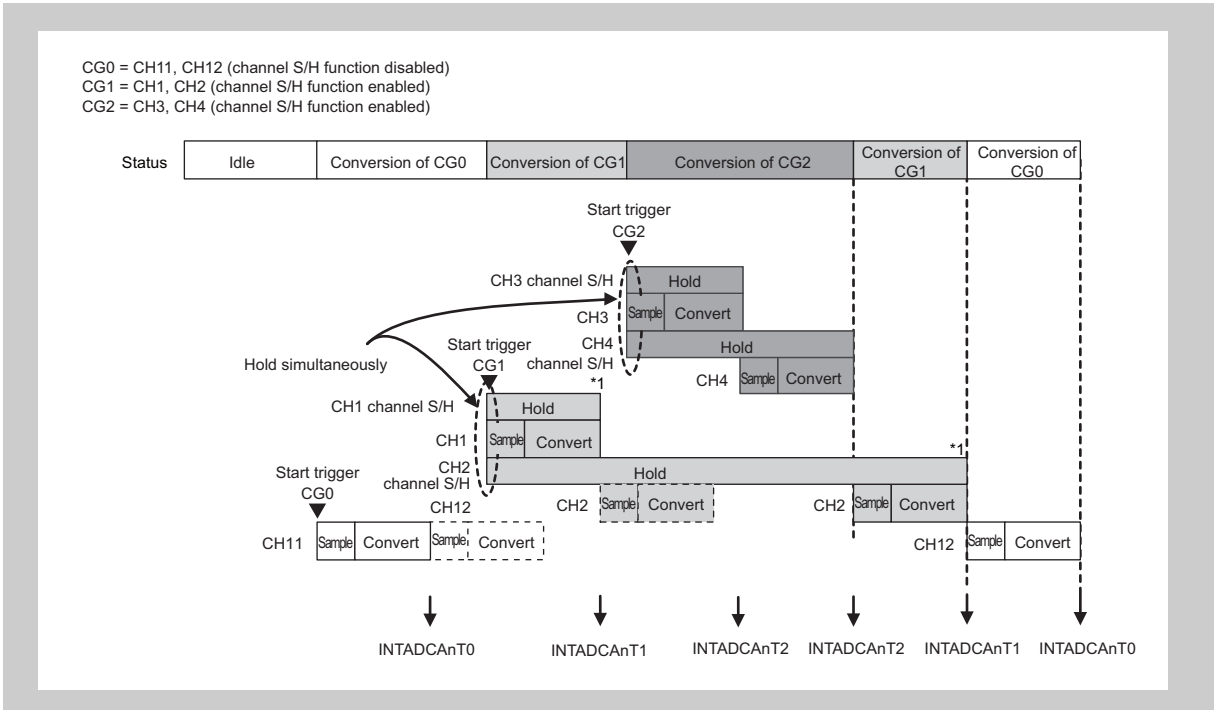
When a hardware or software start trigger occurs, an analog input signal for a channel for which the sample and hold function is enabled using the ADCAnSHCTL.ADCAnCSELx bit is held in the channel sample and hold circuit. Then the scan list conversion is started according to the setting of the ADCAnCTL1.ADCAnTRMi bit.

[Example 1: When a higher-priority start trigger occurs (ADCAnTRM0 bit = 1, ADCAnTRM1 bit = 1)]



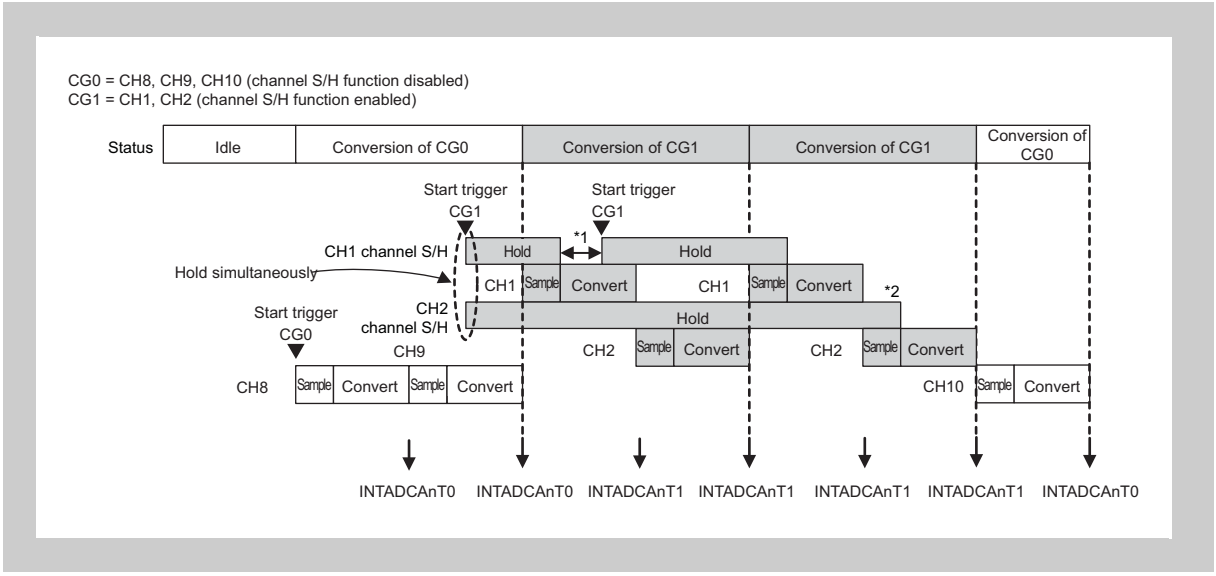
Note 1. When the ADCAnTRM1 bit = 1, operation of the channel sample and hold circuit switches to sampling at this point.

[Example 2: When a higher-priority start trigger occurs (ADCAnTRM0 bit = 0, ADCAnTRM1 bit = 0)]



Note 1. When the ADCAnTRM1 bit = 0, operation of the channel sample and hold circuit switches to sampling at this point.

[Example 3: When a subsequent start trigger occurs (ADCAnTRM0 bit = 1, ADCAnTRM1 bit = 1)]

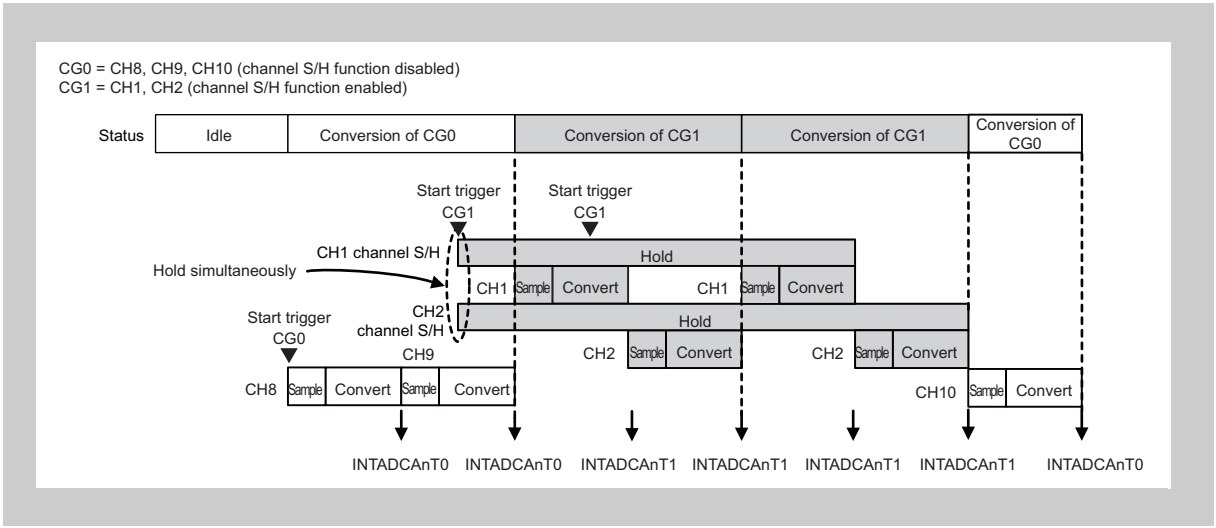


Note 1. The sampling time for the channel sample and hold circuit needs to be considered; 1.5 µs or more should be secured.

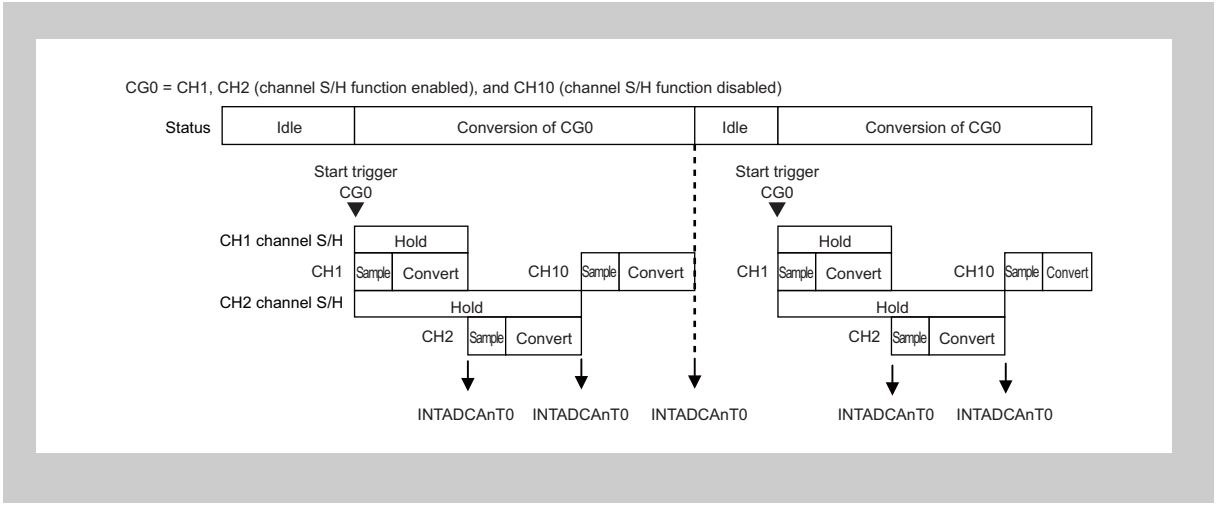
Note 2. If a start trigger occurs, the sample and hold circuit continues to hold the current analog input voltage until the next conversion.

[Example 4: When a subsequent start trigger occurs (ADCAnTRM0 bit = 1, ADCAnTRM1 bit = 0)]

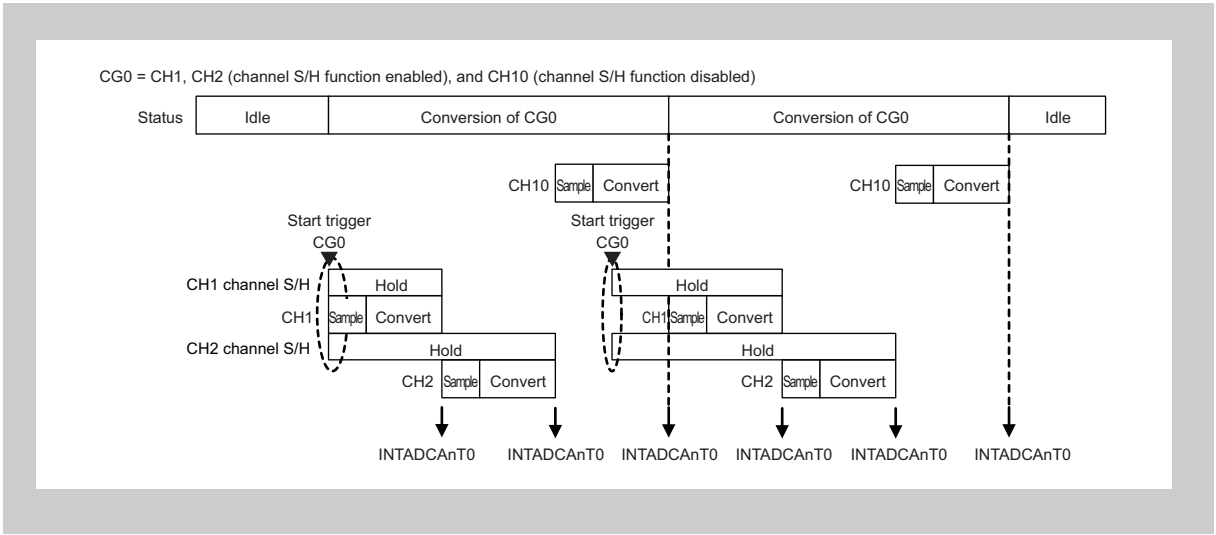
If a subsequent start trigger occurs during A/D conversion, the sample and hold circuit does not perform the further sampling.



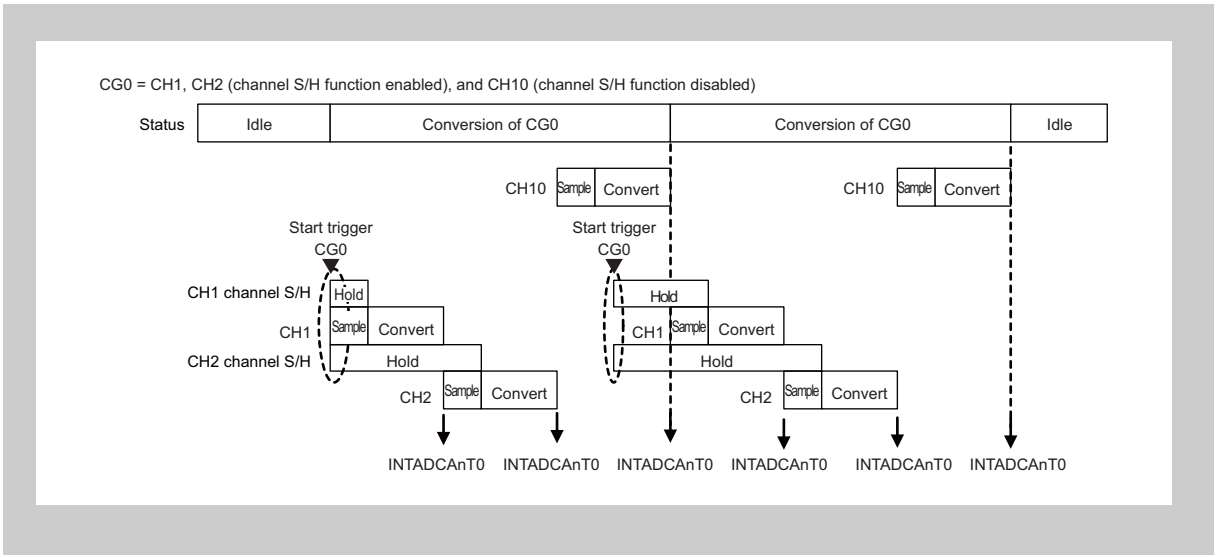
[Example 5: When CG0 is in one-shot conversion mode (no repetition) (ADCAnTRM0 bit = 0)]



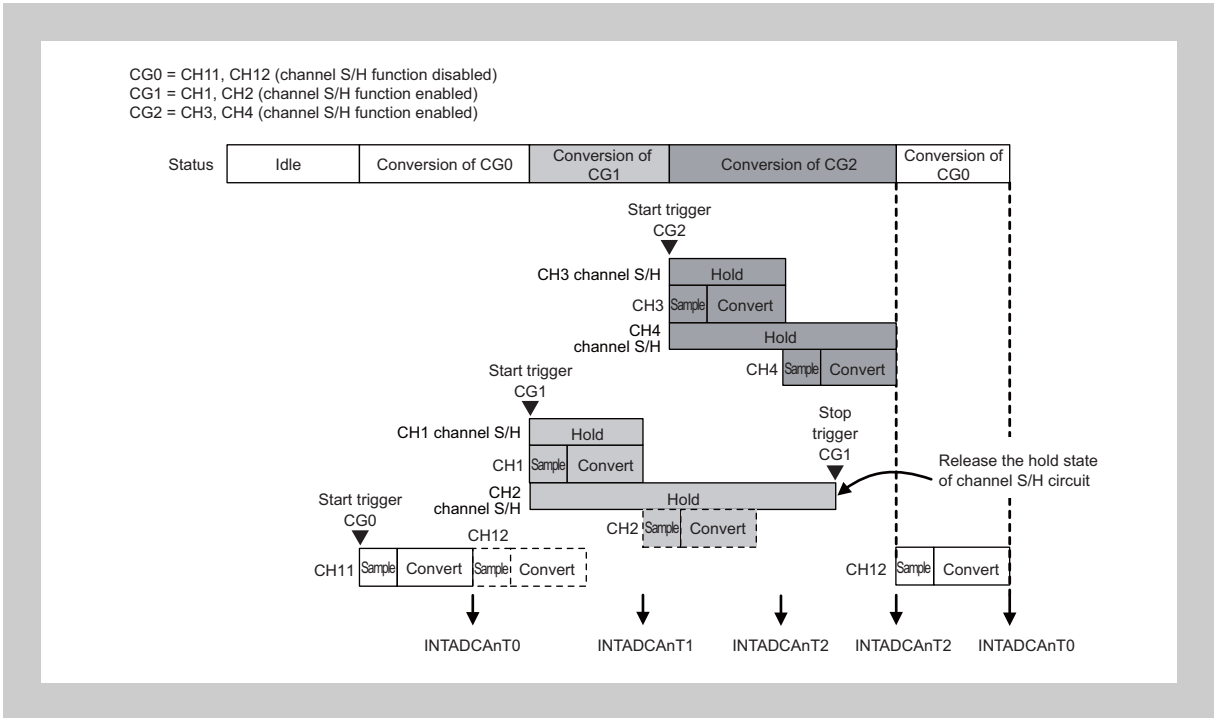
[Example 6: When CG0 is in one-shot conversion mode (no repetition) and a subsequent start trigger occurs (ADCAnTRM0 bit = 0)]



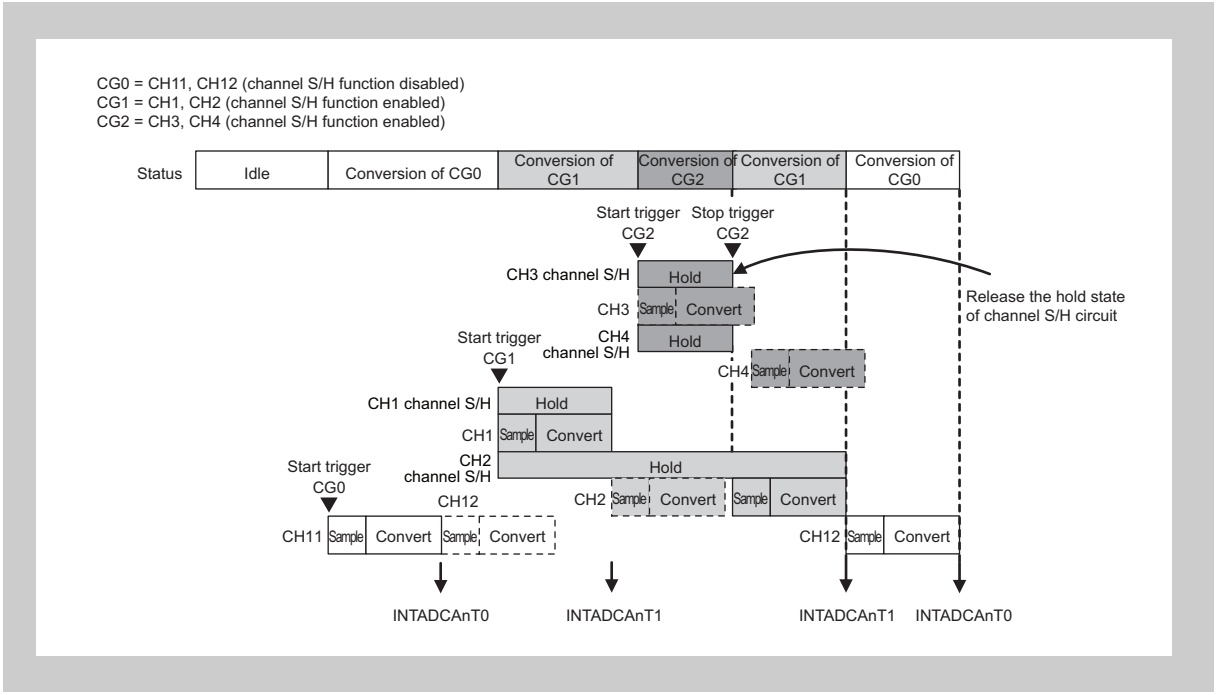
[Example 7: When CG0 is in one-shot conversion mode (no repetition) and a subsequent start trigger occurs (ADCAnTRM0 bit = 1)]



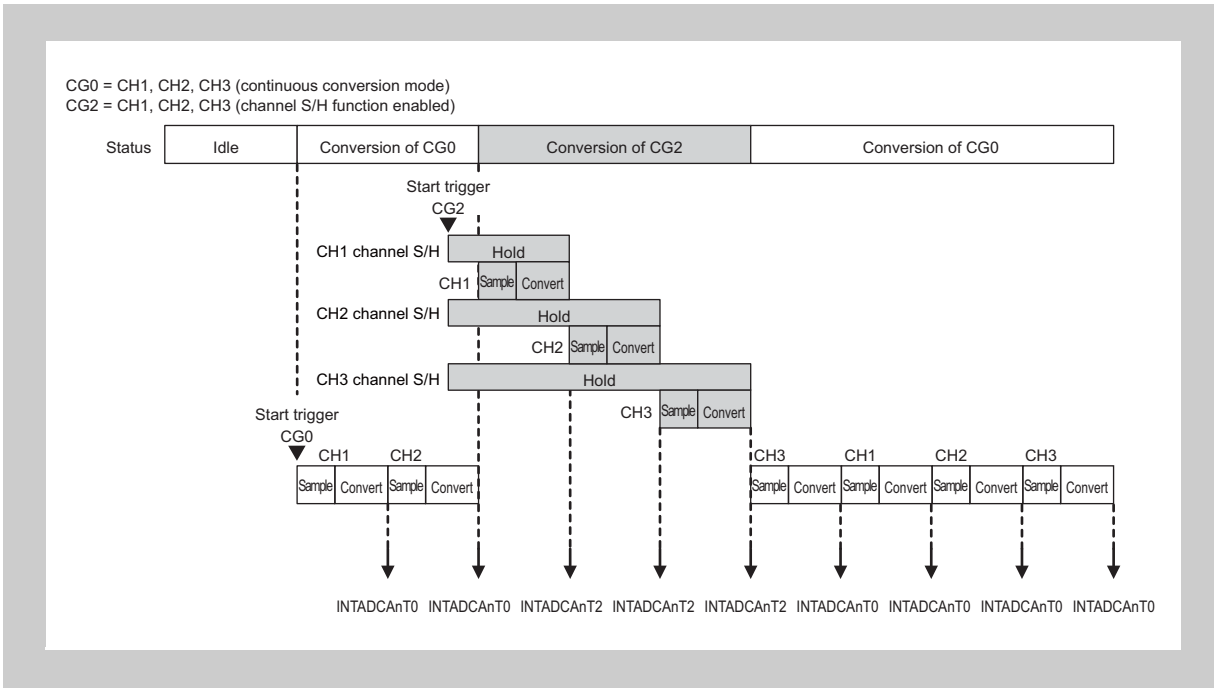
[Example 8: When mutually different channels are set for CG1 and CG2 and a stop trigger for CG1 occurs (ADCAnTRM0 bit = 0, ADCAnTRM1 bit =0)]



[Example 9: When mutually different channels are set for CG1 and CG2 and a stop trigger for CG2 occurs (ADCAnTRM0 bit = 0, ADCAnTRM1 bit =0)]



[Example 10: When CG0 is in continuous conversion mode and the same channels are set for CG0 and CG2 (ADCAnTRM0 bit = 1, ADCAnTRM2 bit = 0)]



(2) Restrictions when Using Sample and Hold Function

Restrictions when using sample and hold function are as follows.

(1) The channels to which the channel sample and hold function is applied must not be assigned to multiple CGs at the same time.

Example 1: When the channel sample and hold function is applied to CG0, CG1, and CG2 (CG0 in one-shot conversion mode)

The following combination is possible.

– CG0 (CH1 selected), CG1 (CH2 selected), and CG2 (CH3 selected)

The following combinations are prohibited.

– CG0 (CH1 selected), CG1 (CH1 selected), and CG2 (CH2 and CH3 selected)

Example 2: When the channel sample and hold function is applied to CG1 and CG2 (CG0 in continuous conversion mode)

The following combinations are possible.

– CG0 (CH1 selected), CG1 (CH1 and CH2 selected), and CG2 (CH3 selected)

– CG0 (CH1, CH2, and CH3 selected), CG1 (CH1 selected), and CG2 (CH2 and CH3 selected)

The following combination is prohibited.

– CG0 (CH1, CH2, and CH3 selected), CG1 (CH1 and CH2 selected), and CG2 (CH2 and CH3 selected)

(2) When changing the scan list of CGi to which the channel sample and hold function is applied during A/D conversion, set the scan list so that the channels for applying the function are not changed.

Example: When the channel sample and hold function is applied to CH1, CH2, and CH3.

The following changes are possible.

– Change from CG0 (CH1, CH2, and CH3) to CG0 (CH1, CH2, CH3, CH10, and CH11)

– Change from CG0 (CH1, CH2, CH3, CH10, and CH11) to CG0 (CH1, CH2, and CH3)

– Change from CG0 (CH7, CH8, and CH9) to CG0 (CH10, CH11, and CH12)

The following changes are prohibited.

– Change from CG0 (CH1, CH2, and CH3) to CG0 (CH1 and CH2)

– Change from CG0 (CH1) to CG0 (CH1, CH2, and CH3)

– Change from CG0 (CH7, CH8, and CH9) to CG0 (CH1, CH7, CH8, and CH9)

– Change from CG0 (CH1, CH2, and CH9) to CG0 (CH9, CH10, and CH11)

(3) Repetition of A/D conversion is not available for CG to which the channel sample and hold function is applied. When the function is applied to CGi, the repetition times setting bit (ADCAnCTL0.ADCAnSCTi[1:0]) for the appropriate CG should be set to 00_B.

(4) When using the channel sample and hold function, use the buffer amplifier function as well. For details on the buffer amplifier function, refer to section 25.3.14, Buffer Amplifier Function.

25.3.13 Discharge Function

If required, the internal capacitor in the common sample and hold circuit can be discharged prior to every A/D conversion.

Note When the discharge function is enabled, the total conversion time is increased by 1 ADCAnTCLK clock cycle (refer to section 25.3.7, Resolution, Sampling Time and Conversion Time).

Setting The discharge function is enabled by setting ADCAnCTL1.DCAnDISC to 1.

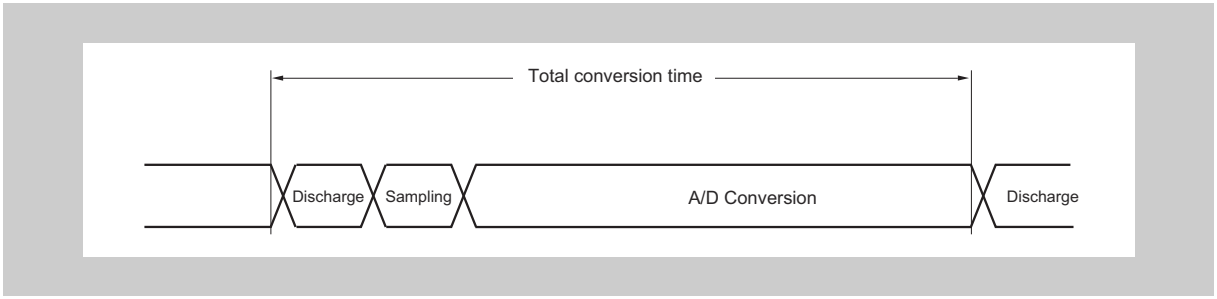


Figure 25-20 A/D Conversion Timing when Discharge Function is Enabled

25.3.14 Buffer Amplifier Function

If required, the analogue input signal can be connected to an internal buffer amplifier. Charging the internal sampling capacitor is accelerated by the buffer amplifier during sampling in A/D conversion.

The buffer amplifier function is enabled by setting ADCAnCTL1.ADCAnBPC to 1.

Note When the buffer amplifier function is enabled, the total conversion time is increased by 4 clock cycles of ADCAnTCLK (refer to section 25.3.7, Resolution, Sampling Time and Conversion Time).

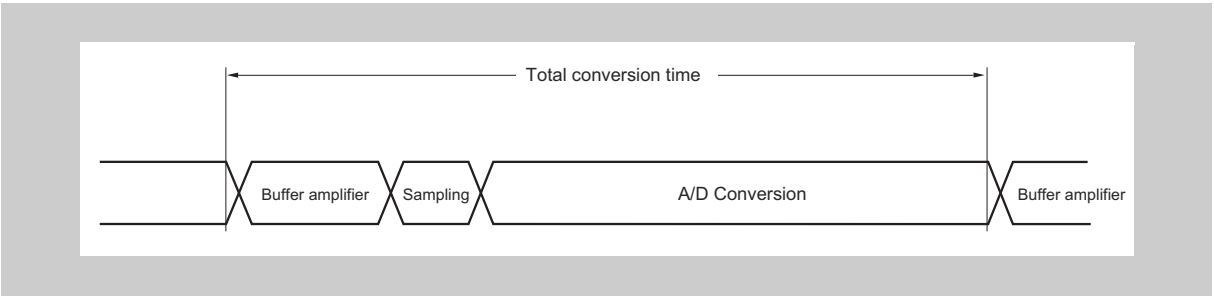


Figure 25-21 A/D Conversion Timing when Buffer Amplifier Function is Enabled

25.3.15 Stabilization Control

The A/D converter needs time to stabilize operation at power on of the A/D converter (ADCAnCTL1.ADCAnGPS = 1).

Although a start trigger is acceptable even during the stabilization time, A/D conversion is not started before the stabilization time has elapsed.

In order to secure the minimum stabilization time, the setting of the stabilization time counter ADCAnCNT must be adapted (the counter should be set so that the stabilization time is not less than the Power-down return time (refer to Section 28 Electrical Characteristics)).

25.4 Registers

This section contains a description of all the registers of the ADCAn.

25.4.1 ADCAn Registers Overview

The ADCAn is controlled and operated based on the settings of the registers in the following table.

- Where there is one register per channel, the channel number is represented by affix m.
- Where there is one register per CG, the CG number is represented by affix i (i = 0 to 2).

Table 25-7 List of ADCAn Registers (1/2)

Register Name	Symbol	Address
Control registers		
A/D converter mode control register 0	ADCAnCTL0	<ADCAn_base0> + 100_H
A/D converter mode control register 1	ADCAnCTL1	<ADCAn_base0> + 104_H
A/D converter CG register i	ADCAnCGi	<ADCAn_base1> + i × 4_H
A/D converter interrupt control register i	ADCAnIOCi	<ADCAn_base1> + C_H + i × 4_H
A/D converter trigger select control register i	ADCAnTSELi	<ADCAn_base0> + 108_H + i × 4_H
A/D converter stabilization counter	ADCAnCNT	<ADCAn_base0> + 114_H
Conversion status registers		
A/D converter overwrite error flag register	ADCAnSTR1	<ADCAn_base1> + 28_H
ADCAnSTR1 flag clear register	ADCAnSTC1	<ADCAn_base1> + 34_H
A/D converter status flag register 2	ADCAnSTR2	<ADCAn_base1> + $2C_H$
ADCAnSTR2 flag clear register	ADCAnSTC2	<ADCAn_base1> + 38_H
Software trigger registers		
A/D converter software trigger register i	ADCAnTRGi	<ADCAn_base1> + $A4_H$ + i × 4_H
A/D converter software trigger register 4 + i	ADCAnTRG4+i	<ADCAn_base1> + $B4_H$ + i × 4_H
A/D conversion result registers		
A/D converter latest conversion result register	ADCAnLCR	<ADCAn_base1> + $A0_H$
A/D converter conversion result register m	ADCAnCmCR	<ADCAn_base1> + $3C_H$ + m × 4_H
A/D converter CGi DMA buffer register i	ADCAnDBiCR	<ADCAn_base1> + $C4_H$ + i × 4_H
A/D converter CGi DMA buffer register i (conversion result)	ADCAnDBiCRL	<ADCAn_base1> + $D0_H$ + i × 4_H
A/D converter diagnosis conversion result register	ADCAnDGCR	<ADCAn_base1> + $9C_H$
A/D conversion result upper/lower limit comparison registers		
A/D conversion result check register	ADCAnCTL2	<ADCAn_base1> + 18_H
A/D conversion result check (upper limit)	ADCAnUL	<ADCAn_base1> + $1C_H$
A/D conversion result check (lower limit)	ADCAnLL	<ADCAn_base1> + 20_H
A/D conversion result check error flag	ADCAnSTR0	<ADCAn_base1> + 24_H
ADCAnSTR0 flag clear register	ADCAnSTC0	<ADCAn_base1> + 30_H

Table 25-7 List of ADCAn Registers (2/2)

Register Name	Symbol	Address
Diagnosis function control registers		
A/D converter self-diagnosis function control register 0	ADCAnDGCTL0	<ADCAn_base1> + DC_H
A/D converter self-diagnosis function control register 1	ADCAnDGCTL1	<ADCAn_base0> + $11C_H$
A/D converter internal pull-down resistance control register 0	ADCAnPDCTL0	<ADCAn_base0> + 120_H
Channel S/H function setting register		
A/D converter channel S/H control register	ADCAnSHCTL	<ADCAn_base0> + 118_H

25.4.2 Control Registers

(1) ADCAnCTL0 – A/D Converter Mode Control Register 0

This register enables/disables the A/D converter. Additionally it specifies the number of repetition times for A/D conversion in one-shot conversion mode and the generation of error interrupt requests when an A/D conversion result is overwritten before it was read.

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

Address <ADCAn_base0> + 100_H

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ADCAn OEM4	ADCAnOEM[3:1]			ADCAn OEM0	ADCAn CE	0	ADCAn SCT2[1:0]		ADCAn SCT1[1:0]		ADCAn SCT0[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-8 ADCAnCTL0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>12</td><td>ADCAn
OEM4</td><td>Specifies whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in ADCAnLCR is overwritten before it is read:
0: Generates the error interrupt INTADCAnTERR when an A/D conversion result is overwritten
1: Does not generate the error interrupt INTADCAnTERR
For details, refer to section 25.3.10 (1), Overwrite Check for A/D Conversion Results</td></tr>
<tr><td>11 to 9</td><td>ADCAn
OEM[3:1]</td><td>Specifies whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in any ADCAnDBiCR is overwritten before it is read:
0: Generates the error interrupt INTADCAnTERR when an A/D conversion result is overwritten
1: Does not generate the error interrupt INTADCAnTERR
CGi is controlled by the ADCAnOEM (i + 1) bit
For details, refer to section 25.3.10 (1), Overwrite Check for A/D Conversion Results.</td></tr>
<tr><td>8</td><td>ADCAn
OEM0</td><td>Specifies whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in any ADCAnCmCR is overwritten before it is read:
0: Generates the error interrupt INTADCAnTERR when an A/D conversion result is overwritten
1: Does not generate the error interrupt INTADCAnTERR
For details, refer to section 25.3.10 (1), Overwrite Check for A/D Conversion Results.</td></tr>
<tr><td>7</td><td>ADCAn
CE</td><td>Enables or disables the A/D converter:
0: Disables the A/D converter
1: Enables the A/D converter
The A/D conversion is started by generating a hardware or software start trigger (using bit ADCAnTRGi.ADCAnSTTi) after setting the ADCAnCTL0.ADCAnCE bit to 1.
Although a hardware or software start trigger is acceptable immediately after power on using bit ADCAnCTL1.ADCAnGPS (during the stabilization time), A/D conversion is not started before the stabilization time has elapsed. After the stabilization time , A/D conversion is started.</td></tr>
<tr><td>5 to 0</td><td>ADCAn
SCTi[1:0]</td><td>Number of times the scan list conversions of CG0, CG1, and CG2 are repeated in one-shot conversion mode
<table>
<tr><th>ADCAn SCTi1</th><th>ADCAn SCTi0</th><th>Number of Repetitions of Scan List Conversions of CGi</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table>
</td></tr>
</table>

Caution When a number of times for repeated conversion has been set for the CGi scan list (by the ADCAnCTL0.ADCAnSCTi bits), and the next hardware trigger is generated within one cycle of the ADCAnTCLK clock following the final conversion, the number of repetitions becomes one.
Accordingly, the A/D conversion time must be taken into account in the settings for hardware triggers.

(2) ADCAnCTL1 – A/D Converter Mode Control Register 1

This register specifies the A/D conversion mode and controls the conversion operations.

Access This register can be read/written in 32-bit units.

Address <ADCAn_base0> + 104_H

Initial value $0100\ 0008_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAn T2ETS[1:0]		ADCAn T1ETS[1:0]		ADCAn T0ETS[1:0]		0	ADCAn CRAC	0	0	ADCAn MD1	ADCAn MD0	0	0	ADCAn DISC	ADCAn RCL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAn CTYP	0	0	ADCAn STL	ADCAnFR[3:0]				0	0	ADCAn TRM[1:0]		ADCAn BPC	0	0	ADCAn GPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-9 ADCAnCTL1 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 26	ADCAn TiETS[1:0]	Specifies the valid edge of the hardware trigger signal ADCAnTTRGi. **ADCAn TiETS1** / **ADCAn TiETS0** / **Valid Edge** 0 / 0 / No edge detection (trigger is not accepted) 0 / 1 / Rising edge 1 / 0 / Falling edge 1 / 1 / Rising and falling edges
24	ADCAn CRAC	Specifies the alignment of the results of A/D conversion and diagnosis conversion: 0: Right-aligned 1: Left-aligned
21	ADCAn MD1	Specifies the A/D conversion start trigger for CG0 0: Software trigger 1: Hardware and software triggers This setting is applied to the A/D conversion of CG0 only. CG1 and CG2 always accept both hardware and software triggers. The A/D converter detects triggers only when the A/D converter is enabled. For details, refer to section 25.3.5, Starting A/D Conversion (Start Triggers).
20	ADCAn MD0	Specifies the A/D conversion mode for CG0: 0: One-shot conversion mode The number of repetitions is set in ADCAnCTL0.ADCAnSCTi[1:0] for each CG. 1: Continuous conversion mode This setting applies to the A/D conversion of CG0 only. CG1 and CG2 always operate in one-shot conversion mode. For details, refer to section 25.3.4, A/D Conversion Modes.

Table 25-9 ADCAnCTL1 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17</td><td>ADCAn
DISC</td><td>Enables/disables the discharge function
0: Disable
1: Enable
For details, refer to section 25.3.13, Discharge Function.</td></tr>
<tr><td>16</td><td>ADCAn
RCL</td><td>Specifies whether the A/D conversion result in ADCAnCmCR, ADCAnDBiCR, or ADCAnDBiCRL is retained after being read:
0: The A/D conversion result is retained until it is overwritten by the next A/D conversion result
1: The A/D conversion result is cleared after being read</td></tr>
<tr><td>15</td><td>ADCAn
CTYP</td><td>Specifies the resolution mode
0: 12-bit resolution
1: 10-bit resolution</td></tr>
<tr><td>12</td><td>ADCAn
STL</td><td>Specifies the level of the ADCAnCNVi signal
0: When ADCAnCNVi = low, CGi is not currently performing conversion
When ADCAnCNVi = high, CGi is currently performing conversion
1: When ADCAnCNVi = high, CGi is not currently performing conversion
When ADCAnCNVi = low, CGi is currently performing conversion</td></tr>
<tr><td>11 to 8</td><td>ADCAn
FR[3:0]</td><td>Specifies the ADCAn system clock ADCAnTCLK
For setting values, refer to section 25.3.7, Resolution, Sampling Time and Conversion Time.
<table>
<tr><th>ADCAnFR[3:0]</th><th>ADCAn Clock</th></tr>
<tr><td>0000</td><td>PCLK/2</td></tr>
<tr><td>0001</td><td>PCLK/3</td></tr>
<tr><td>0010</td><td>PCLK/4</td></tr>
<tr><td>0011</td><td>PCLK/5</td></tr>
<tr><td>0100</td><td>PCLK/6</td></tr>
<tr><td>0110</td><td>PCLK/8</td></tr>
<tr><td>1000</td><td>PCLK/10</td></tr>
<tr><td>1010</td><td>PCLK/12</td></tr>
<tr><td>1100</td><td>PCLK/14</td></tr>
<tr><td>1110</td><td>PCLK/16</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>5 to 4</td><td>ADCAn
TRM[1:0]</td><td>Specifies the interrupt behavior when the A/D conversion of a higher-priority CG is triggered. The settings in ADCAnTRM1 and ADCAnTRM0 apply to CG1 and CG0, respectively. The setting of CG2 is always 0.
0: Halts the current A/D conversion of CGi immediately, and starts the A/D conversion of the higher-priority CG (immediately halts the A/D conversion due to a higher-priority trigger, holds the current conversion status so that the conversion can be resumed later, and releases the hold state upon completion of the higher-priority A/D conversion).
1: Halts the A/D conversion of CGi after the conversion of the current channel is completed, and starts the A/D conversion of the higher-priority CG (even if a higher-priority trigger is generated, the A/D conversion is suspended until conversion completion of the current channel; therefore the hold state of the trigger signal is released upon completion of sampling).
After the A/D conversions of all the higher-priority CGs are finished, the A/D conversion of CGi is resumed.
The priority order is: CG2 > CG1 > CG0.
For details, see section 25.3.3 (1), Order of A/D Conversion.</td></tr>
</table>

Table 25-9 ADCAnCTL1 Register Contents (3/3)

Bit Position	Bit Name	Function
3	ADCAn BPC	Enables/disables the buffer amplifier function: 0: Disable 1: Enable For details, see section 25.3.14, Buffer Amplifier Function.
0	ADCAn GPS	Powers on or off the ADCAn 0: Power off 1: Power on The AD converter needs time to stabilize its operation after being powered on (refer to section 25.3.15, Stabilization Control).

Caution If you are using the μPD70F3506 or μPD70F3508 product, be sure to set ADCAnCTYP to 1.

(3) ADCAnCGi – A/D Converter Channel Group Register i

This register makes a scan list for each CG. Channels included in the scan list are converted in ascending order. For details, refer to section 25.3.3, Channel and Channel Group.

This register can also enable or disable the diagnostic A/D conversion which uses the reference voltage signal ADDIAGOUT, by using ADCAnCG0.ADCAnDIAG. For details, refer to section 25.3.11 (1), Diagnosis of A/D Conversion Circuit.

Access This register can be read/written in 32-bit units.
Since this register uses master/slave configuration, the next A/D conversion channel can be set in the master register during current A/D conversion. The value in the master register is transferred to the slave register at the following timing:

- When CGi is not currently performing A/D conversion, the value is transferred 1 clock cycle (PCLK) after it is written in the master register.
- When CGi is currently performing A/D conversion, the value is transferred upon completion of the scan list conversion of the current CGi.
- If a stop trigger bit of CGi (ADCAnSPi bit) is set after the value is written to this register, the value is transferred when the A/D conversion is stopped.

Address <ADCAn_base1> + i × 4_H

Initial value 0000 0000_H

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAn DIAG	0	0	0	0	0	0	0	ADCAnCGiS[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCGiS[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-10 ADCAn Register Contents

Bit Position	Bit Name	Function
31	ADCAn DIAG	Enables or disables the diagnostic A/D conversion of the reference voltage signal ADDIAGOUT at the end of the A/D conversion of CG0: 0: Disables A/D conversion of ADDIAGOUT 1: Converts ADDIAGOUT This bit only applies to ADCAnCG0. This bit should be set to 0 in ADCAnCG1 and ADCAnCG2.
23 to 00	ADCAn CGiS[23:00]	Specifies the analog inputs to be converted for CGi: 0: Does not convert analog input ADCAnIm 1: Converts analog input ADCAnIm Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

(4) ADCAnIOCi – A/D Converter Interrupt Control Register i

The A/D conversion end interrupt INTADCAnTi can be generated when the A/D conversion of a certain channel has been completed.

This register specifies the channels for which INTADCAnTi is generated on A/D conversion completion.

If ADCAnIOCi = 0000 0000_H, INTADCAnTi is automatically generated at the completion of A/D conversion of CGi.

Access This register can be read/written in 32-bit units.
It can be written at any time even when the A/D converter is enabled (ADCAnCTL0.ADCAnCE = 1). The new value takes effect after the current A/D conversion of CGi has been completed.

Address <ADCAn_base1> + $0C_H$ + i × 4_H

Initial value 0000 0000_H

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAnCG0IDG	0	0	0	0	0	0	0	ADCAnCGil[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCGil[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-11 ADCAnIOCi Register Contents

Bit Position	Bit Name	Function
31	ADCAn CG0IDG	Specifies whether the interrupt INTADCAnTi is generated on completion of the A/D conversion of the reference voltage when the diagnostic mode is enabled for CG0 (ADCAnCG0.ADCAnDIAG = 1): 0: Does not generate the A/D conversion end interrupt INTADCAnTi 1: Generates the A/D conversion end interrupt INTADCAnTi This bit only applies to ADCAnIOC0. This bit should be cleared to 0 in ADCAnIOC1 and ADCAnIOC2. For details, refer to section 25.3.11 (1), Diagnosis of A/D Conversion Circuit.
23 to 00	ADCAn CGil[23:00]	Specifies whether or not the interrupt INTADCAnTi is generated on A/D conversion completion of channel m: 0: Does not generate the A/D conversion end interrupt INTADCAnTi 1: Generates the A/D conversion end interrupt INTADCAnTi Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

Note ADCAnIOCi is associated with ADCAnCGi and their buffer registers should be updated simultaneously. As the update time depends on writing to ADCAnCGi, always write to ADCAnIOCi before ADCAnCGi if you want to change the interrupt generation for a CG.

(5) ADCAnCNT – A/D Converter Stabilization Counter

This register specifies the stabilization time.

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

Address <ADCAn_base0> + 114_H

Initial value 00_H

This register is initialized by any reset.

7	6	5	4	3	2	1	0
ADCAnCNT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-12 ADCAnCNT Register Contents

Bit Position	Bit Name	Function
7 to 0	ADCAnCNT [7:0]	Specifies the stabilization counter: Stabilization time = ADCAnCNT[7:0] × PCLK cycle

Note The value of stabilization counter should be set so that the stabilization time is not less than the Power-down return time (refer to Section 28 Electrical Characteristics).

(6) ADCAnTSELi – A/D Converter Trigger Select Control Register 0

This register specifies the input signals to be used in combination with hardware start triggers ADCAnTTRGi. Multiple trigger sources can be used simultaneously.

Access This register can be read/written in 16-bit units.
It can only be written when the A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0).

Address <ADCAn_base0> + 108_H + i × 4_H

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnTiSEL[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-13 ADCAnTSELi Register Contents

Bit Position	Bit Name	Function
15 to 0	ADCAnTiSEL [15:00]	Specifies whether the corresponding input signal is to be used as hardware start trigger. 0: Not used as hardware start trigger 1: Used as hardware start trigger Note: The bits corresponding to the triggers that are not implemented in this product should be cleared to 0.

Note For units to which the hardware trigger signals are connected, refer to table 25-2, Units to which Hardware Trigger Signals are Connected.

25.4.3 Conversion Status Registers

(1) ADCAnSTR1 – A/D Converter Overwrite Error Flag

This register indicates whether the latest A/D conversion result in ADCAnCmCR has been overwritten before it is read.

Access This register can be read in 32-bit units.

Address <ADCAn_base1> + 28_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnOWE[23:16]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnOWE[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-14 ADCAnSTR1 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAn OWE[23:00]	Indicates whether the A/D conversion result of channel m has been overwritten before it is read. 0: Not overwritten 1: Overwritten This error flag is cleared by setting ADCAnSTC1.ADCAnOWECm to 1. Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

Note The value of ADCAnSTR1.ADCAnOWEm is reflected in the following overwrite error flag:

- The error flag in the A/D converter conversion result register for channel m (ADCAnCmCR.ADCAnCmER1)

(2) ADCAnSTC1 – ADCAnSTR1 Flag Clear Register

This register is the clear control register for ADCAnSTR1.

Access This register can be written in 32-bit units.

It is always read as 0000 0000_H.

Address <ADCAn_base1> + 34_H

Initial value 0000 0000_H

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnOWEC[23:16]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnOWEC[15:00]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25-15 ADCAnSTC1 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAnOWEC [23:00]	Clears the A/D converter overwrite error flag (ADCAnSTR1.ADCAnOWEm bit). 0: Invalid (the setting 0 has no effect on handling of the A/D converter overwrite error flag, i.e. the ADCAnSTR1.ADCAnOWEm bit) 1: Clears ADCAnSTR1.ADCAnOWEm Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

(3) ADCAnSTR2 – A/D Converter Status Flag 2

This register indicates the current conversion status.

Access This register can be read in 16-bit units.

Address <ADCAn_base1> + $2C_H$

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	ADCAnRQ[2:0]			0	0	0	0	0	ADCAnST[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-16 ADCAnSTR2 Register Contents

Bit Position	Bit Name	Function
10 to 8	ADCAnRQ [2:0]	Indicates whether the A/D conversion request for CGi is pending: 0: A/D conversion request for CGi is not pending 1: A/D conversion request for CGi is pending
2 to 0	ADCAnST [2:0]	Indicates whether A/D conversion of CGi is currently performed: 0: A/D conversion is not currently performed (including a halt due to an A/D conversion of a higher-priority CG) 1: A/D conversion is currently performed This bit is cleared when the A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0).

(4) ADCAnSTC2 – A/D Converter Status Flag Clear Register 2

This register is used to clear the overwrite and result check status flags of ADCAnLCR and ADCAnDBiCR.

Access This register can be written in 8-bit units.

Address <ADCAn_base1> + 38_H

Initial value 00_H

This register is initialized by any reset.

7	6	5	4	3	2	1	0
ADCAn LERC1	ADCAn LERC0	ADCAn DB2ERC1	ADCAn DB2ERC0	ADCAn DB1ERC1	ADCAn DB1ERC0	ADCAn DB0ERC1	ADCAn DB0ERC0
W	W	W	W	W	W	W	W

Table 25-17 ADCAnSTC2 Register Contents

Bit Position	Bit Name	Function
7	ADCAn LERC1	Clears the overwrite flag (ADCAnLCR.ADCAnLER1 bit): 0: Invalid (the setting 0 has no effect on handling of the flag) 1: Clears ADCAnLCR.ADCAnLER1
6	ADCAn LERC0	Clears the result check error flag (ADCAnLCR.ADCAnLER0 bit): 0: Invalid (the setting 0 has no effect on handling of the flag) 1: Clears ADCAnLCR.ADCAnLER0
5, 3, 1	ADCAn DBiERC1	Clears the overwrite flag (ADCAnDBiCR.ADCAnDBiER1 bit): 0: Invalid (the setting 0 has no effect on handling of the flag) 1: Clears ADCAnDBiCR.ADCAnDBiER1
4, 2, 0	ADCAn DBiERC0	Clears the result check error flag (ADCAnDBiCR.ADCAnDBiER0 bit): 0: Invalid (the setting 0 has no effect on handling of the flag) 1: Clears ADCAnDBiCR.ADCAnDBiER0

25.4.4 Software Trigger Registers

(1) ADCAnTRGi – A/D Converter Software Trigger Register i

This register is the trigger register to start the A/D conversion of CGi.

Access This register can be written in 8-bit units.
It is always read as 00_H.

Address <ADCAn_base1> + $A4_H$ + i × 4_H

Initial value 00_H

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAn STTi
W	W	W	W	W	W	W	W

Table 25-18 ADCAnTRGi Register Contents

Bit Position	Bit Name	Function
0	ADCAnSTTi	Starts the A/D conversion of CGi: 0: Invalid (the setting 0 does not start the A/D conversion of CGi) 1: Starts A/D conversion of CGi

For details, refer to section 25.3.5, Starting A/D Conversion (Start Triggers).

(2) ADCAnTRG4+i – A/D Converter Software Trigger Register 4 + i

This register is the software trigger register to stop the A/D conversion of CGi.

Access This register can be written in 8-bit units.
It is always read as 00_H.

Address <ADCAn_base1> + $B4_H + i \times 4_H$

Initial value 00_H

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAnSPi
W	W	W	W	W	W	W	W

Table 25-19 ADCAnTRG4+i Register Contents

Bit Position	Bit Name	Function
0	ADCAnSPi	Stops the A/D conversion of CGi: 0: Invalid (the setting 0 does not stop the A/D conversion of CGi) 1: Stops the A/D conversion of CGi

For details, refer to section 25.3.6, Stopping A/D Conversion (Stop Trigger).

25.4.5 A/D Conversion Result Registers

(1) ADCAnLCR – A/D Converter Latest Conversion Result Register

This register stores the result and the status of the latest A/D conversion.

It allows the latest A/D conversion results to be read.

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base1> + $A0_H$

Initial value $0300\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn LCG[1:0]		ADCAn LER1	ADCAn LER0	ADCAn LUR	ADCAnLCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnLCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-20 ADCAnLCR Register Contents (1/2)

Bit Position	Bit Name	Function
25, 24	ADCAn LCG[1:0]	Indicates the CG to which the conversion result in ADCAnLCR[15:00] belongs. **ADCAn LCG1** / **ADCAn LCG0** / **Channel Group** 0 / 0 / CG0 0 / 1 / CG1 1 / 0 / CG2 1 / 1 / None
23	ADCAn LER1	Indicates the overwrite error status: 0: The conversion result has not been overwritten 1: The conversion result has been overwritten This error flag is cleared when ADCAnSTC2.ADCAnLERC1 is set to 1.
22	ADCAn LER0	Indicates the upper/lower limit comparison result of the A/D conversion result: 0: The conversion result is within the specified range 1: The conversion result is out of the specified range This error flag is cleared when ADCAnSTC2.ADCAnLERC0 is set to 1.
21	ADCAn LUR	Indicates the update status of the A/D conversion result: 0: The A/D conversion result has been read from ADCAnLCR (the result has not been updated) 1: The A/D conversion result has not been read from ADCAnLCR (the result has been updated) This bit is cleared after the conversion result is read.

Table 25-20 ADCAnLCR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>20 to16</td><td>ADCAn LCN[4:0]</td><td>Indicates the number of the channel to which the result in ADCAnLCR[15:00] belongs:
00001 × m = CHm</td></tr>
<tr><td>15 to 0</td><td>ADCAn LCR[15:00]</td><td>Indicates the result of the A/D conversion.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>Bit Position of Conversion Result Value</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>ADCAnLCR[11:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>ADCAnLCR[15:04]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>ADCAnLCR[09:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>ADCAnLCR[15:06]</td></tr>
</table>
</td></tr>
</table>

Note The result of the diagnostic A/D conversion of an internal reference voltage is stored in ADCAnDGCR, not in ADCAnLCR, ADCAnCmCR, or ADCAnDBiCR (refer to section 25.4.5 (5), ADCAnDGCR – A/D Converter Diagnosis Conversion Result Register).

(2) ADCAnCmCR – A/D Converter Conversion Result Register for Channel m

This register stores the result and the status of the latest A/D conversion of channel m.

It allows you to read the A/D conversion result of the specified channel (m).

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base1> + $3C_H$ + m × 4_H

Initial value $0300\ 0000_H$ + m × $0001\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn CmCG[1:0]		ADCAn CmER1	ADCAn CmER0	ADCAn CmUR	ADCAnCmCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCmCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The functions of the individual bits are identical to those of the corresponding bits in ADCAnLCR, except that this register indicates the latest A/D conversion result of a specified channel rather than the latest results of all the channels (refer to table 25-20, ADCAnLCR Register Contents).

Note 2. After reset, ADCAnCmCG[1:0] are set to 11_B.

Note 3. When ADCAnCTL1.ADCAnRCL = 0, the A/D conversion result in ADCAnCmCR[15:00] is kept until it is overwritten by the next A/D conversion result.

When ADCAnCTL1.ADCAnRCL = 1, the A/D conversion result in ADCAnCmCR[15:00] is cleared after being read.

Table 25-21 ADCAnCmCR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>25, 24</td><td>ADCAn CmCG[1:0]</td><td>Indicates the CG to which the conversion result in ADCAnCmCR [15:00] belongs.
<table>
<tr><th>ADCAn CmCG1</th><th>ADCAn CmCG0</th><th>Channel Group</th></tr>
<tr><td>0</td><td>0</td><td>CG0</td></tr>
<tr><td>0</td><td>1</td><td>CG1</td></tr>
<tr><td>1</td><td>0</td><td>CG2</td></tr>
<tr><td>1</td><td>1</td><td>None</td></tr>
</table></td></tr>
<tr><td>23</td><td>ADCAn CmER1</td><td>Indicates the overwrite error status:
0: The conversion result has not been overwritten
1: The conversion result has been overwritten
This error flag reflects the value of ADCAnSTR1.ADCAnOWEm and is cleared when ADCAnSTC1.ADCAnQWECm is set to 1.</td></tr>
<tr><td>22</td><td>ADCAn CmER0</td><td>Indicates the upper/lower limit comparison result of the A/D conversion result:
0: The conversion result is within the specified range
1: The conversion result is out of the specified range
This error flag reflects the value of ADCAnSTR0.ADCAnRCEm and is cleared when ADCAnSTC0.ADCAnRCECm is set to 1.</td></tr>
<tr><td>21</td><td>ADCAn CmUR</td><td>Indicates the update status of the A/D conversion result:
0: The A/D conversion result has been read from ADCAnCmCR (the result has not been updated)
1: The A/D conversion result has not been read from ADCAnCmCR (the result has been updated)
This bit is cleared after the conversion result is read.</td></tr>
<tr><td>20 to 16</td><td>ADCAn CmCN[4:0]</td><td>Indicates the number of the channel to which the conversion result in ADCAnCmCR [15:00] belongs:
00001 × m = CHm</td></tr>
<tr><td>15 to 0</td><td>ADCAn CmCR [15:00]</td><td>Indicates the result of the A/D conversion.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>Bit Position of A/D Conversion Result Value</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>ADCAnCmCR[11:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>ADCAnCmCR[15:04]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>ADCAnCmCR[09:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>ADCAnCmCR[15:06]</td></tr>
</table></td></tr>
</table>

Note The result of the diagnostic A/D conversion of an internal reference voltage is stored in ADCAnDGCR, not in ADCAnLCR, ADCAnCmCR, or ADCAnDBiCR (refer to section 25.4.5 (5), ADCAnDGCR – A/D Converter Diagnosis Conversion Result Register).

(3) ADCAnDBiCR – CGi DMA buffer register

This register holds the result and the status of the latest A/D conversion of CGi. It allows the A/D conversion results of all the channels of CGi to be read.

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base1> + $C4_H$ + i × 4_H

Initial value $0000\ 0000_H$ + i × $0100\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn DBiCG[1:0]		ADCAn DBiER1	ADCAn DBiER0	ADCAn DBiUR	ADCAnDBiCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDBiCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note The functions of the individual bits are identical to those of the corresponding bits in ADCAnLCR, except that this register indicates the latest A/D conversion result of CGi rather than the latest results of all the CGs (refer to table 25-20, ADCAnLCR Register Contents).

Table 25-22 ADCAnDBiCR Register Contents (1/2)

Bit Position	Bit Name	Function
25, 24	ADCAn DBiCG[1:0]	Indicates the CG to which the conversion result in ADCAnDBiCR [15:00] belongs. **ADCAn DBiCG1** / **ADCAn DBiCG0** / **Channel Group** 0 / 0 / CG0 0 / 1 / CG1 1 / 0 / CG2 1 / 1 / None These bit settings are fixed since the conversion results and status of the same CG are always stored.
23	ADCAn DBiER1	Indicates the overwrite error status: 0: The conversion result has not been overwritten 1: The conversion result has been overwritten This error flag is cleared when ADCAnSTC2.ADCAnDBiERC1 is set to 1.
22	ADCAn DBiER0	Indicates the upper/lower limit comparison result of the A/D conversion result: 0: The conversion result is within the specified range 1: The conversion result is out of the specified range This error flag is cleared when ADCAnSTC2.ADCAnDBiERC0 is set to 1.

Table 25-22 ADCAnDBiCR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>21</td><td>ADCAn DBiUR</td><td>Indicates the update status of the A/D conversion result:
0: The A/D conversion result has been read from ADCAnDBiCR (the result has not been updated)
1: The A/D conversion result has not been read from ADCAnDBiCR (the result has been updated)
This bit is cleared after the conversion result is read.</td></tr>
<tr><td>20 to 16</td><td>ADCAn DBiCN[4:0]</td><td>Indicates the number of the channel to which the conversion result in ADCAnDBiCR [15:00] belongs:
00001 × m = CHm</td></tr>
<tr><td>15 to 0</td><td>ADCAn DBiCR [15:00]</td><td>Indicates the result of the A/D conversion.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>Bit Position of A/D Conversion Result Value</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>ADCAnDBiCR[11:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>ADCAnDBiCR[15:04]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>ADCAnDBiCR[09:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>ADCAnDBiCR[15:06]</td></tr>
</table>
</td></tr>
</table>

Note 1.
- When the ADCAnCTL1.ADCAnRCL bit is 0, the value of the ADCAnDBiCR[15:00] bits is retained until it is overwritten by the next result of A/D conversion.
- When the ADCAnCTL1.ADCAnRCL bit is 1, the value of the ADCAnDBiCR[15:00] bits is cleared after the ADCAnDBiCR register is read.

Note 2. The result of the diagnostic A/D conversion of an internal reference voltage is stored in ADCAnDGCR, not in ADCAnLCR, ADCAnCmCR, or ADCAnDBiCR (refer to section 25.4.5 (5), ADCAnDGCR – A/D Converter Diagnosis Conversion Result Register).

(4) ADCAnDBiCRL – A/D Converter CGi DMA Buffer Register (Conversion Result)

This register holds the result of the latest A/D conversion of CGi. It allows the A/D conversion results of all the channels of CGi to be read. This register reflects the settings of ADCAnDBiCR[15:00].

Access This register is read-only and can be read in 16-bit units.

Address <ADCAn_base1> + $D0_H$ + i × 4_H

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDBiCRL[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-23 ADCAnDBiCRL Register Contents

Bit Position	Bit Name	Function
15 to 0	ADCAn DBiCRL [15:00]	Indicates the result of the A/D conversion. The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:

ADCAn CTL1. ADCAn CTYP	ADCAn CTL1. ADCAn CRAC	Resolusion and Alignment	Bit Position of Conversion Result Value
0	0	12-bit resolution, right-aligned	ADCAnDBiCRL[11:00]
0	1	12-bit resolution, left-aligned	ADCAnDBiCRL[15:04]
1	0	10-bit resolution, right-aligned	ADCAnDBiCRL[09:00]
1	1	10-bit resolution, left-aligned	ADCAnDBiCRL[15:06]

Note 1. The value of this register is:

- Retained until the A/D conversion result is overwritten by the next A/D conversion result when ADCAnCTL1.ADCAnRCL = 0.
- Cleared after the A/D conversion results in ADCAnDBiCR and ADCAnDBiCRL are read when ADCAnCTL1.ADCAnRCL = 1.

Note 2. The result of the diagnostic A/D conversion of an internal reference voltage is stored in ADCAnDGCR, not in ADCAnLCR, ADCAnCmCR, or ADCAnDBiCR (refer to section 25.4.5 (5), ADCAnDGCR – A/D Converter Diagnosis Conversion Result Register).

(5) ADCAnDGCR – A/D Converter Diagnosis Conversion Result Register

This register stores the A/D conversion result of the reference voltage signal ADDIAGOUT (when ADCAnCG0.ADCAnDIAG = 1).

The diagnosis A/D conversion is started after the A/D conversion of the last channel of CG0 is completed.

Access This register can be read in 16-bit units.

Address <ADCAn_base1> + $9C_H$

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDGCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-24 ADCAnDGCR Register Contents

Bit Position	Bit Name	Function
15 to 0	ADCAn DGCR[15:00]	Indicates the result of the diagnostic A/D conversion. The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC, as is the case with the normal A/D conversion result registers.

25.4.6 A/D Conversion Result Upper/Lower Limit Comparison Registers

(1) ADCAnCTL2 – A/D Conversion Result Check Register

This register enables or disables the upper/lower limit comparison function for the conversion result for every channel.

For details, refer to section 25.3.10, Result Check Functions.

Access This register can be read/written in 32-bit units.
It can only be written when the A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0).

Address <ADCAn_base1> + 18_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCK[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCK[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-25 ADCAnCTL2 Register Contents

Bit Position	Bit Name	Function
23 to 00	ADCAn RCK[23:00]	Enables or disables the upper/lower limit comparison for the result of CHm 0: Does not perform the upper/lower limit comparison for the A/D conversion result of CHm 1: Performs the upper/lower limit comparison for the A/D conversion result of CHm Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

Note These settings are valid for the A/D conversions of every CG.

(2) ADCAnUL – A/D Converter Result Upper/Lower Limit Comparison (Upper Limit)

This register specifies the upper limit of the A/D conversion result.

For details, refer to section 25.3.10, Result Check Functions.

Access This register can be read/written in 16-bit units.
It can only be written when the A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0).

Address <ADCAn_base1> + $1C_H$

Initial value 0000_H

This register is initialized by any reset

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnUL[11:00]												0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-26 ADCAnUL Register Contents

Bit Position	Bit Name	Function
15 to 4	ADCAnUL [11:00]	Specifies the upper limit of the A/D conversion result. In 10-bit mode, set the limit in ADCAnUL[11:02].

(3) ADCAnLL – A/D Converter Result Upper/Lower Limit Comparison (Lower Limit)

This register specifies the lower limit of the A/D conversion result.

For details, refer to section 25.3.10, Result Check Functions.

Access This register can be read/written in 16-bit units.
It can only be written when the A/D converter is disabled (ADCAnCTL0.ADCAnCE = 0).

Address <ADCAn_base1> + 20_H

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnLL[11:00]												0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-27 ADCAnLL Register Contents

Bit Position	Bit Name	Function
15 to 4	ADCAnLL [11:00]	Specifies the lower limit of the A/D conversion result. In 10-bit mode, set the limit in ADCAnLL[11:02].

(4) ADCAnSTR0 – A/D Converter Result Upper/Lower Limit Comparison Error Flag

This register indicates the error status of the latest A/D conversion result check for the channels set in ADCAnCTL2. This allows to check which A/D conversion results are outside the specified range.

For details, refer to section 25.3.10, Result Check Functions.

Access This register can be read in 32-bit units.

Address <ADCAn_base1> + 24_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCE[23:16]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCE[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-28 ADCAnSTR0 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAn RCE[23:00]	Indicates whether the A/D conversion result is within the specified range: 0: The conversion result is within the specified range 1: One or more conversion results are out of the specified range This error flag is cleared when ADCAnSTC0.ADCAnRCECm is set to 1. Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

Note The value of ADCAnSTR0.ADCAnRCEm is reflected in the following A/D conversion result error flag.

- The error flag in the A/D converter conversion result register for channel m (ADCAnCmCR.ADCAnCmER0)

(5) ADCAnSTC0 – ADCAnSTR0 Flag Clear Register

This register is the clear control register of ADCAnSTR0.

Access This register can be written in 32-bit units.
It is always read as 0000 0000_H.

Address <ADCAn_base1> + 30_H

Initial value 0000 0000_H

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCEC[23:16]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCEC[15:00]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25-29 ADCAnSTC0 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAnRCEC [23:00]	Clears the A/D converter result upper/lower limit comparison error flag (ADCAnSTR0.ADCAnRCEm bit). 0: No function 1: Clears ADCAnSTR0.ADCAnRCEm Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

25.4.7 Diagnosis Function Control Registers

(1) ADCAnDGCTL0 – Self-Diagnosis Function Control Register 0

This register specifies the reference voltages applied for diagnosing the operation of the A/D conversion circuit.

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

Address <ADCAn_base1> + DC_H

Initial value 0000_H

This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	ADCAnPSEL[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-30 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Function
2 to 0	ADCAn PSEL[2:0]	Specifies the reference voltages.

ADCAn PSEL2	ADCAn PSEL1	ADCAn PSEL0	ADDIAGOUT Signal	DIAGOUT2 Signal	DIAGOUT1 Signal	DIAGOUT0 Signal
0	0	0	AV_{SS}	2/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}
0	0	1	1/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}
0	1	0	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}
0	1	1	2/3 AV_{DD}	Hi-Z	Hi-Z	Hi-Z
1	0	0	AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}
1	0	1	AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}
1	1	0	AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}
1	1	1	AV_{DD}	Hi-Z	Hi-Z	Hi-Z

The A/D conversion result is undefined when the conversion is performed with Hi-Z selected.

For details, refer to section 25.3.11 (1), Diagnosis of A/D Conversion Circuit.

(2) ADCAnDGCTL1 – Self-Diagnosis Function Control Register 1

This register specifies to which channel an internal reference voltage (instead of the analog input voltage ADCAnIm) is applied.

This register is writable only when ADCAnCTL0.ADCAnCE = 0.

Access This register can be read/written in 32-bit units.

Address <ADCAn_base0> + $11C_H$

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnCDG[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCDG[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-31 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAn CDG[23:00]	Specifies the input voltage 0: Uses the analog input voltage ADCAnIm 1: Uses the following reference voltages: DIAGOUT0 for m = 21, 18, 15, 12, 9, 6, and 3 DIAGOUT1 for m = 22, 19, 16, 13, 10, 7, 4, and 1 DIAGOUT2 for m = 20, 17, 14, 11, 8, 5, and 2 Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

(3) ADCAnPDCTL0 – Internal Pull-Down Resistance Control Register 0

This register specifies the channels to which the internal pull-down resistors of the ADCAnIm pins are connected. For details, refer to section 25.3.11 (3), Diagnosis of Analog Input Pins.

This register is writable only when ADCAnCTL0.ADCAnCE = 0.

Access This register can be read/written in 32-bit units.

Address <ADCAn_base0> + 120_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnPDNA[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnPDNA[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-32 ADCAnPDCTL0 Register Contents

Bit Position	Bit Name	Function
23 to 0	ADCAn PDNA[23:00]	Specifies whether an internal pull-down resistor is connected to CHm: 0: Does not connect an internal pull-down resistor 1: Connects an internal pull-down resistor Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Number of analog input pins fields in the table in section 25.1, ADCA Features).

25.4.8 Channel Sample and Hold Function Setting Register

(1) ADCAnSHCTL – A/D Converter Channel Sample and Hold Control Register

This register turns on or off the power supply to the channel sample and hold circuit and enables or disables the channel sample and hold function.

This register is writable when ADCAnCTL0.ADCAnCE = 0.

However, it is writable for diagnosis of the channel sample and hold circuit even if ADCAnCTL0.ADCAnCE = 1.

Access This register can be read/written in 32-bit units.

Address <ADCAn_base0> + 118_H

Initial value $0000\ 0000_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	ADCAnCPS[12:00]												
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ADCAnCSEL[12:00]												
R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-33 ADCAnSHCTL Register Contents

Bit Position	Bit Name	Function
28 to 16	ADCAn CPS[12:00]	Controls the power supply to the channel sample and hold circuit. 0: Turns off the power supply to the channel sample and hold circuit 1: Turns on the power supply to the channel sample and hold circuit In order to reduce the power consumption, turn off the power supply to the channel sample and hold circuit when the circuit is not used. Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Channel sample and hold fields in the table in section 25.1, ADCA Features).
12 to 00	ADCAn CSEL[12:00]	Enables or disables the channel sample and hold function. 0: Disables the channel sample and hold function 1: Enables the channel sample and hold function When the channel sample and hold circuit is diagnosed, the setting of ADCAnSHCTL.ADCAnCSELx can be changed even if ADCAnCE = 1. Note: The bits corresponding to the channels that are not implemented in this product should be cleared to 0 (for the applicable bits, refer to the Channel sample and hold fields in the table in section 25.1, ADCA Features).

25.5 Usage Notes

25.5.1 Range of Channel Input Voltage

Caution The input voltage to the ADCAnIm pin should be within the specified voltage range. If the channel input voltage is above AV_{DD} or below AV_{SS}, the conversion result value of the channel is saturated and it may affect the electric characteristics of the other channels.

25.5.2 Stopping Conversion Operation

Writing 0 to ADCAnCTL0.ADCAnCE during conversion operation stops the operation and the conversion result is not stored in ADCAnCmCR.

25.5.3 Restrictions on Using Channel Sample and Hold Function

Refer to section 25.3.12 (2), Restrictions when Using Sample and Hold Function.

25.5.4 Application Design Notes

(1) Analog Input Pin (ADCAnIm)

(a) The input voltage to the ADCAnIm pin should be within the specified voltage range. In order to avoid input of the voltage equal to or above AV_{REFnP} or equal to or below AV_{REFnM}, the input voltage is recommended to be clamped with a diode with a 0.3-V or less V_F. If the voltage equal to or above AV_{REFnP} or equal to or below AV_{REFnM} is input, the conversion result value of the pertinent channel is undefined and cannot be guaranteed. In addition, it may affect the conversion result values of the other channels.

(b) Noise on the analog input pin (ADCAnIm) should be removed by connecting a resistor Re between the pertinent pin and the external analog signal input source and a capacitor Ce between the pertinent pin and the AV_{SSn} pin.

(c) Crossing or close positioning of the analog and digital signal lines should be avoided as much as possible. The A/D conversion characteristics can be degraded due to noise induction or other possible causes.

(d) For both the input and output ports near the ADCAnIm pin, it is recommended to avoid high current drive and to use toggle switching as infrequently as possible.

(e) During self-diagnostic BIST, the voltage for the self-diagnostic circuit of ADCAn is generated from the voltage on the analog pin, so take care on this point when setting up the circuit (as a reference value, an impedance equivalent to about 56 kΩ is suitable when HEAPCLK is running at 80 MHz).

(2) Wiring of Power Supply

To minimize the influence on the A/D converter accuracy due to the factors such as switching noise on the digital circuit, the following measures are recommended.

(a) Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

(b) Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (EV_{DD}, $OSCV_{DD}$, FV_{DD}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$ (provided in some products only), V_{DDM}, V_{DDC}, AV_{DDn}) and ground pins (EV_{SS}, $OSCV_{SS}$, FV_{SS}, $CPUPLLV_{SS}$, $FRPLLV_{SS}$ (provided in some products only), V_{SSM}, V_{SSC}, AV_{SSn}). It is recommended to use approximately 0.1-µF (reference value) multilayer ceramic capacitors or 4.7-µF (reference value) or greater tantalum electrolytic capacitors.

(c) We recommend isolating the analog power supply (AV_{DDn}) from the digital power supply (EV_{DD}, $OSCV_{DD}$, FV_{DD}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$ (provided in some products only), V_{DDM}, and V_{DDC}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board.

Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

(3) Analog Reference Voltage Input Pins (AV_{REFnP}, AV_{REFnM})

Insert a bypass capacitor between the leads of the AV_{REFnP} pin and the AV_{REFnM} pin as close as possible to the leads. It is recommended to use approximately 0.1-µF (reference value) multilayer ceramic capacitors or 4.7-µF (reference value) or greater tantalum electrolytic capacitors.

(4) Variation in A/D Conversion Results

A/D conversion results may vary according to the factors such as fluctuation of the power supply voltage and noise. Erroneous conversion results may also be produced if noise is generated on the analog input pin (ADCAnIm) and reference voltage input pins (AV_{REFnP}, AV_{REFnM}).

To prevent the system from being affected by such variation and erroneous conversion results, reduce them through software.

The following gives examples of using software to prevent bad influence.

- Perform A/D conversion multiple times and use the average value of the results.
- If A/D conversion is continuously performed multiple times and any singular conversion result is obtained, exclude it and use other results.
- If any obtained A/D conversion result is judged to be caused by a system error, do not handle the error immediately; first check to see that the erroneous result is caused again and then handle the error.

(5) Hysteretic Characteristics of A/D Conversion

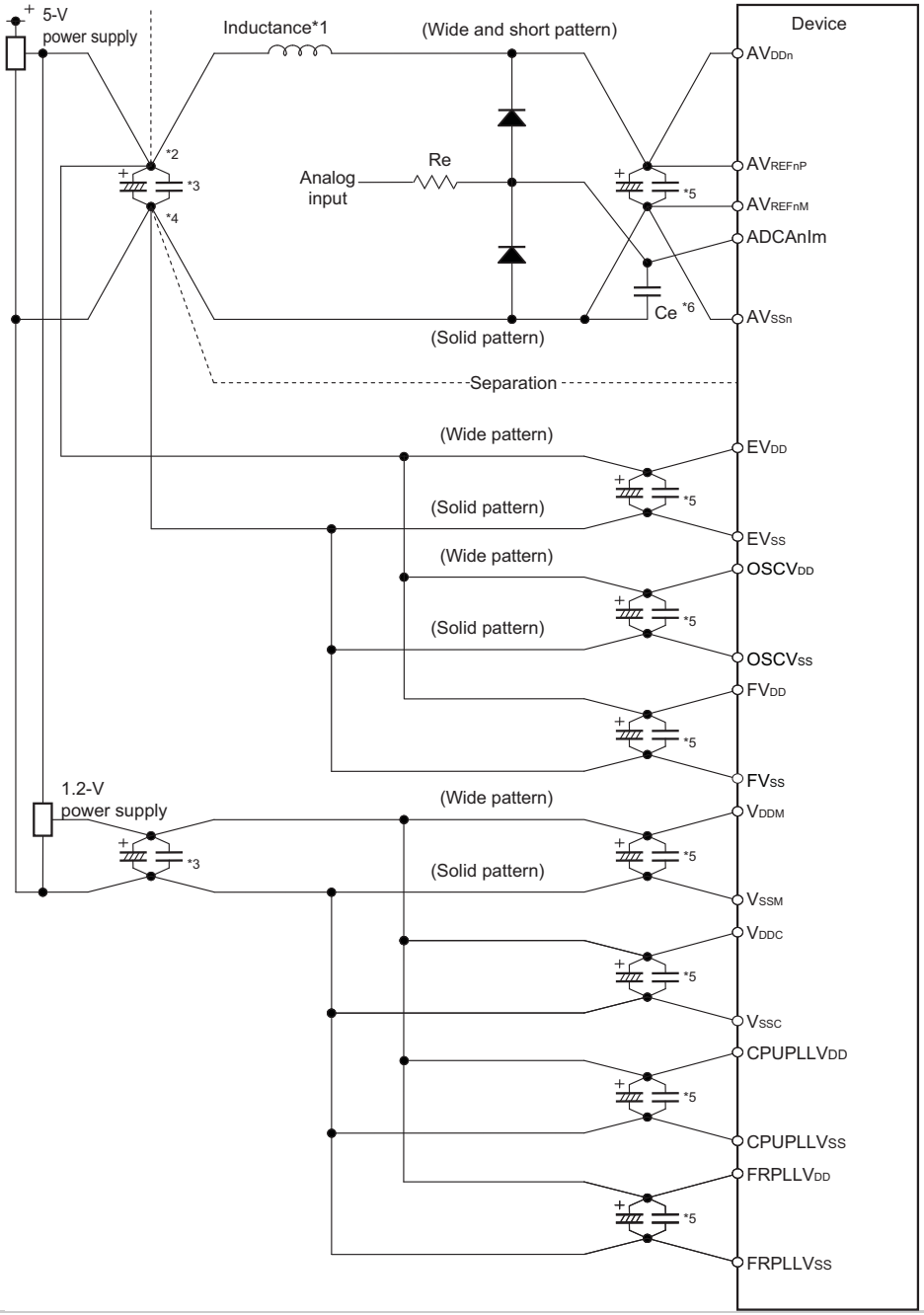
A successive-approximation A/D converter holds the analog input voltage in the internal common S/H capacitor and then performs A/D conversion. After A/D conversion is completed, the analog input voltage remains in the internal common S/H capacitor. This may cause the following phenomena:

(a) If the same channel is used for A/D conversion consecutively and the higher or lower voltage is used than the voltage having been used immediately before, the hysteretic characteristics may be presented, in which the conversion result is influenced by the value of the immediately preceding conversion, and the different conversion results may be obtained even for the same potentials. For reference, the hysteretic characteristics have a tendency to be great when the signal source impedance of the analog input pin in the external circuit and the resistor Re value are large, or the capacitor Ce value is small.

(b) If the analog input channels are switched to be used for A/D conversion, since a single A/D converter is used, the hysteretic characteristics may be presented, in which the conversion result is influenced by the value of the channel having been used immediately before and the different conversion results may be obtained even for the same potentials.

To obtain the more accurate conversion result, perform A/D conversion two consecutive times using the same channel and ignore the first conversion result.

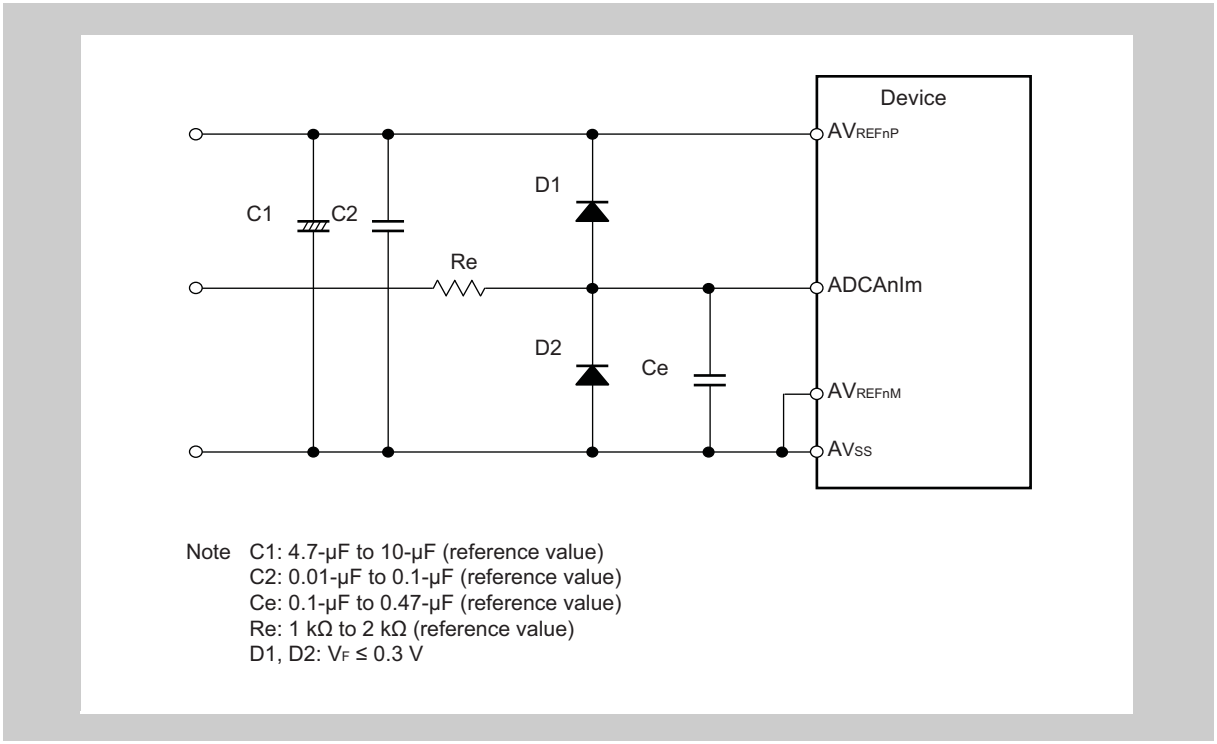
Example: V850E2/Px4



Note 1. Insert an inductor at the entrance of the analog power supply.
Note 2. Short-circuit the analog power supply and digital power supply at a single point near the power supply source, and separate the analog and digital wiring patterns from each other on the circuit board.
Note 3. Place bypass capacitors (electrolytic and multilayer ceramic capacitors) near the power supply.
Note 4. Short-circuit the analog power supply ground and digital power supply ground at a single point near the power supply source, and separate the analog and digital wiring patterns from each other on the circuit board.
Note 5. Place bypass capacitors (electrolytic and multilayer ceramic capacitors) close to the power supply pins of the device.
Note 6. Place the capacitor Ce close to the analog input pin of the device.

Note: This wiring shows the example when the same power supply source is shared by the analog and digital uses and $AV_{REFnP} = AV_{DDn}$.

Figure 25-22 Example of Wiring



Note C1: 4.7-μF to 10-μF (reference value)
C2: 0.01-μF to 0.1-μF (reference value)
Ce: 0.1-μF to 0.47-μF (reference value)
Re: 1 kΩ to 2 kΩ (reference value)
D1, D2: $V_F \leq 0.3$ V

Figure 25-23 Example of Measures against Noise on Analog Input Circuit

The capacitor C1 is effective against low-frequency noise, while capacitors C2 and Ce are effective against high-frequency noise.

Immediately after operation is resumed from the stop state of A/D conversion, the voltages applied to the AV_{DDn} and AV_{REFnP} pins are unstable and it may degrade the A/D conversion accuracy. In such a case, connect capacitors C1 and C2 to the AV_{DDn} and AV_{REFnP} pins.

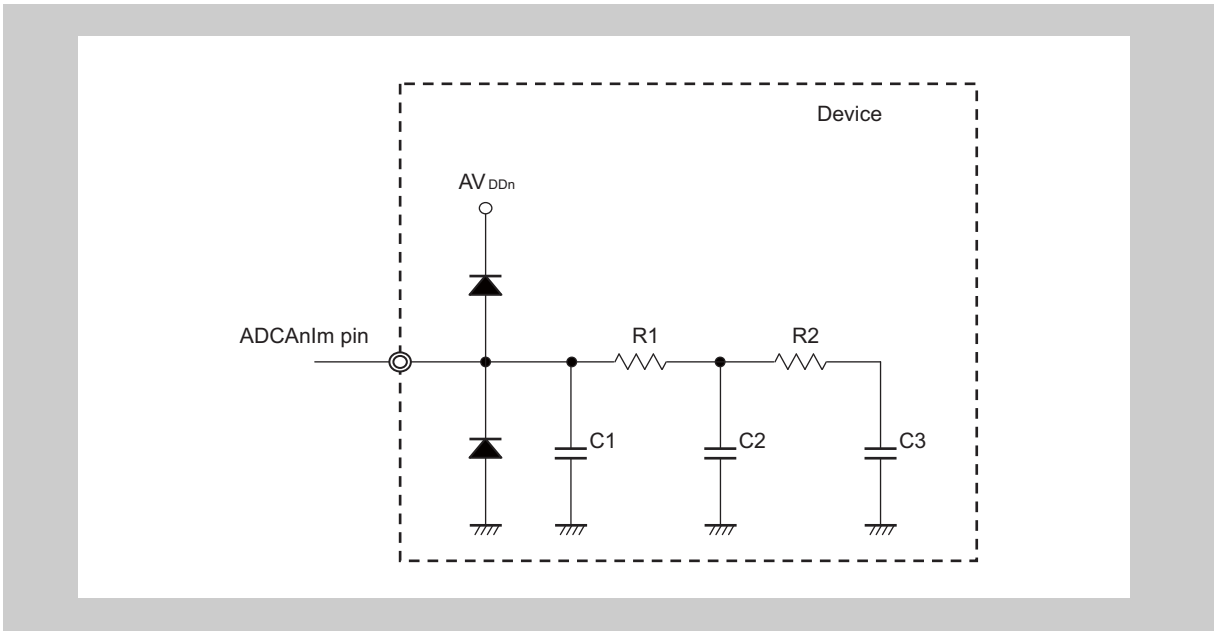


Figure 25-24 Internal Equivalent Circuit of ADCAnIm Pin

Note The following shows the reference values of the circuit constants.

			C1[pF]	C2[pF]		C3[pF]		R1[kΩ]		R2[kΩ]	
			Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max
For pins of channels that have an S/H circuit	Ch S/H used		15	0.62	1.08	1.57	2.43	0.09	0.22	0.18	0.44
	Ch S/H not used	Buf. not used		4.77	5.98	5.26	6.53	0.53	1.31	0.15	0.28
		Buf. used		5.09	6.34	0.54	0.67	0.53	1.31	0.09	0.17
For pins of channels that do not have an S/H circuit	Buf. not used			4.25	5.29	5.26	6.53	0.35	0.87	0.15	0.28
	Buf. used			4.7	6.34	0.54	0.67	0.35	0.87	0.09	0.17

25.6 How to Read A/D Converter Characteristics Table

This section describes the terms specific to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be detected, or the ratio of analog input voltage per one digital output, is called 1 LSB (least significant bit). The ratio of 1 LSB to full scale can be represented in units of %FSR (full scale range). %FSR is the range of the convertible analog input voltage represented in percentage. It is given by the following equation regardless of the resolution.

1 %FSR = (maximum convertible analog input voltage - minimum convertible analog input voltage)/100

$= (AV_{REFP} - AV_{REFM})/100$

When the resolution is 10 bits, 1 LSB is expressed as follows:

$1\ LSB = 1/2^{10}$

$= 1/1024$

$= 0.098\ \%FSR$

When the resolution is 12 bits, 1 LSB is expressed as follows:

$1\ LSB = 1/2^{12}$

$= 1/4096$

$= 0.024\ \%FSR$

The conversion accuracy is determined by the overall error, independent of the resolution.

(2) Overall Error

The overall error is the maximum difference between a measured value and a theoretical value.

It shows the sum of the zero-scale error, full-scale error, linearity error, and the error derived from the combination of these errors.

The quantization error is not included in the overall error in the characteristics table.

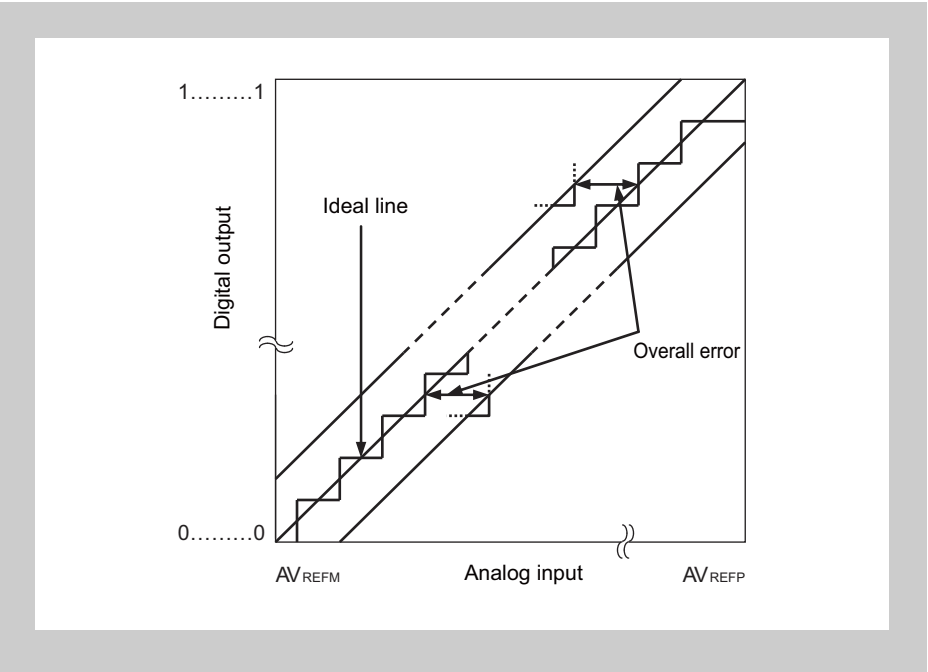


Figure 25-25 Overall Error

(3) Quantization Error

The quantization error is the error of ± 1/2 LSB that is inevitably generated when an analog value is converted to a digital value. The quantization error is unavoidable because analog input voltages within ± 1/2 LSB are converted to the same digital code.

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.

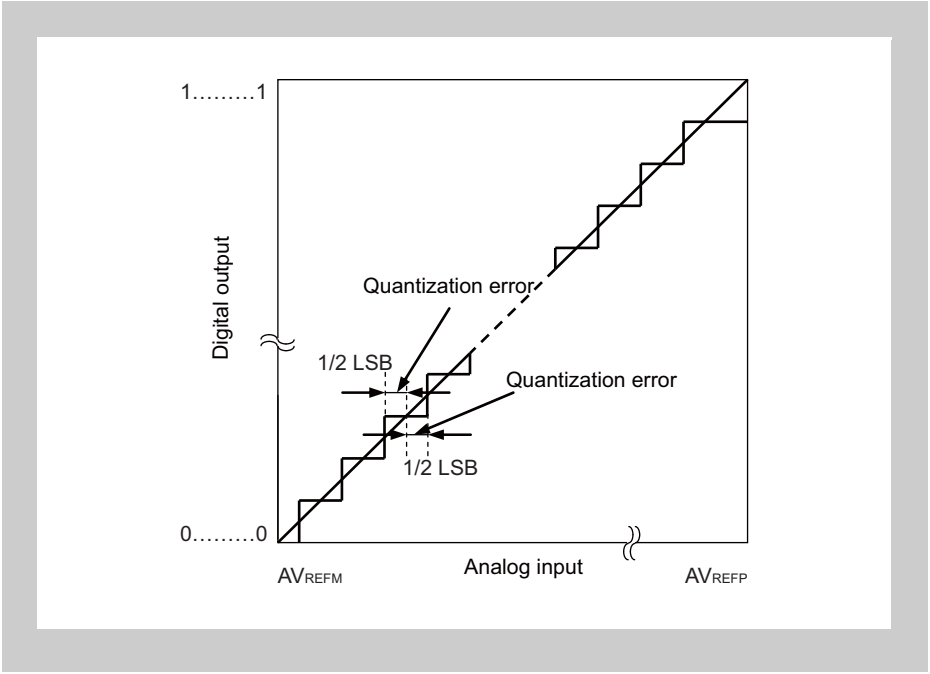


Figure 25-26 Quantization Error

(4) Zero-Scale Error

The zero-scale error is the difference between a measured value and a theoretical value (1/2 LSB) of the analog input voltage when the digital output changes from 0…000 to 0…001.

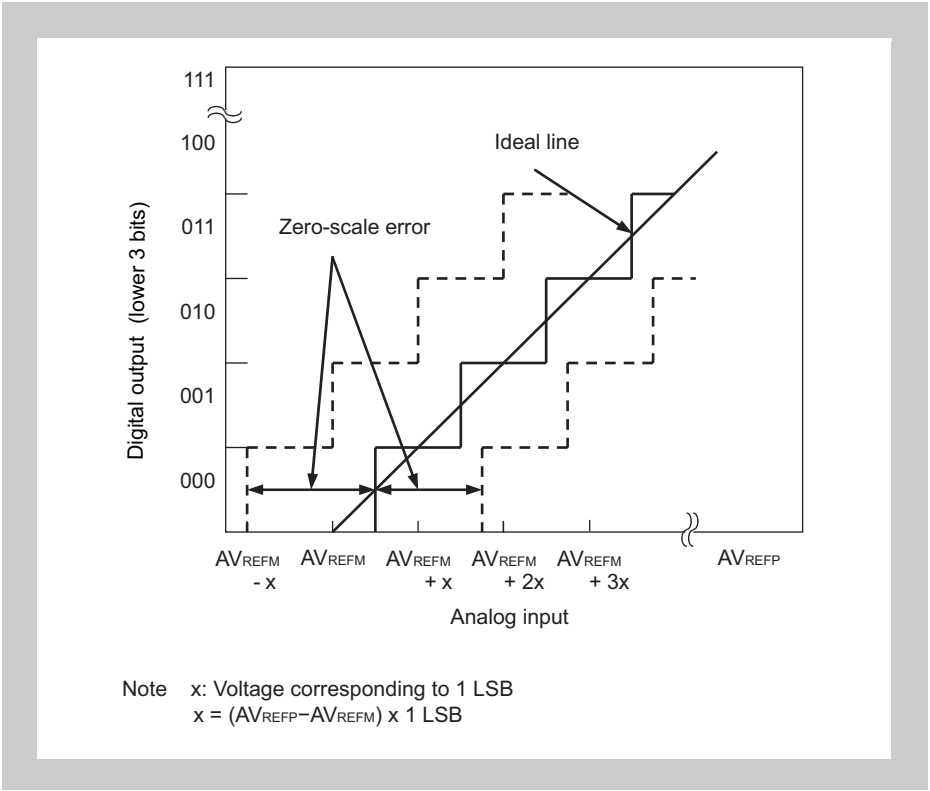


Figure 25-27 Zero-Scale Error

(5) Full-Scale Error

The full-scale error is the difference between a measured value and a theoretical value (full scale - 3/2 LSB) of the analog input voltage when the digital output changes from 1…110 to 1…111.

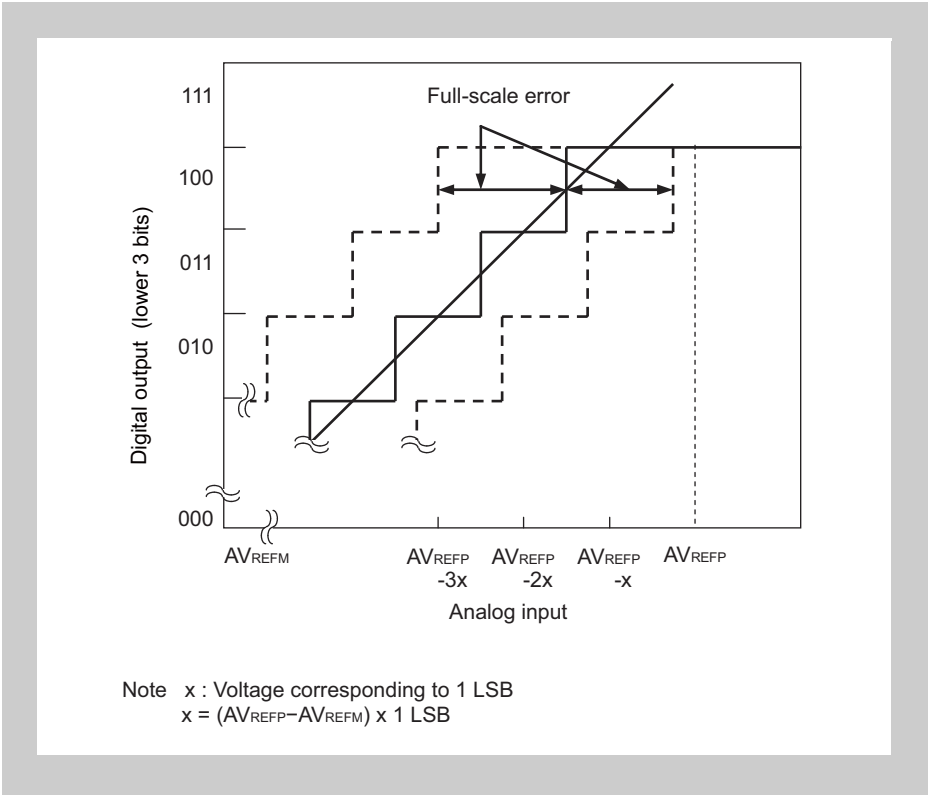


Figure 25-28 Full-Scale Error

(6) Differential Linearity Error

Ideally, the analog input voltage width for outputting a certain code is 1 LSB. The differential linearity error is the difference between a measured value and a theoretical value of the voltage width for outputting a certain code.

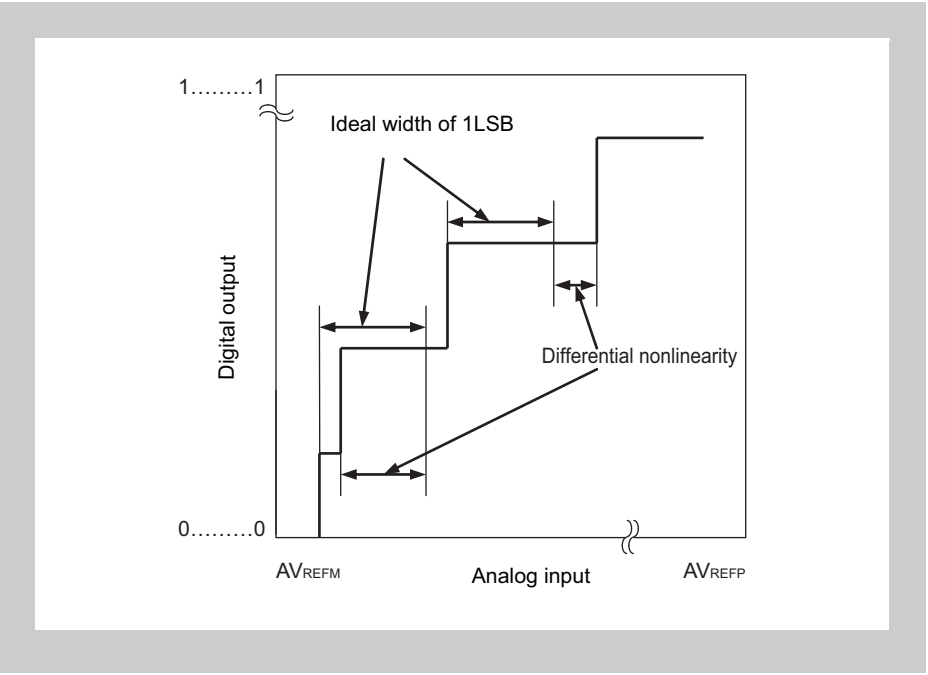


Figure 25-29 Differential Linearity Error

(7) Integral Linearity Error

The integral linearity error is the amount of deviation of the conversion characteristics from the ideal linearity. The integral linearity error is the maximum difference between a measured value and a theoretical value when both the zero-scale error and full-scale error are assumed to be 0.

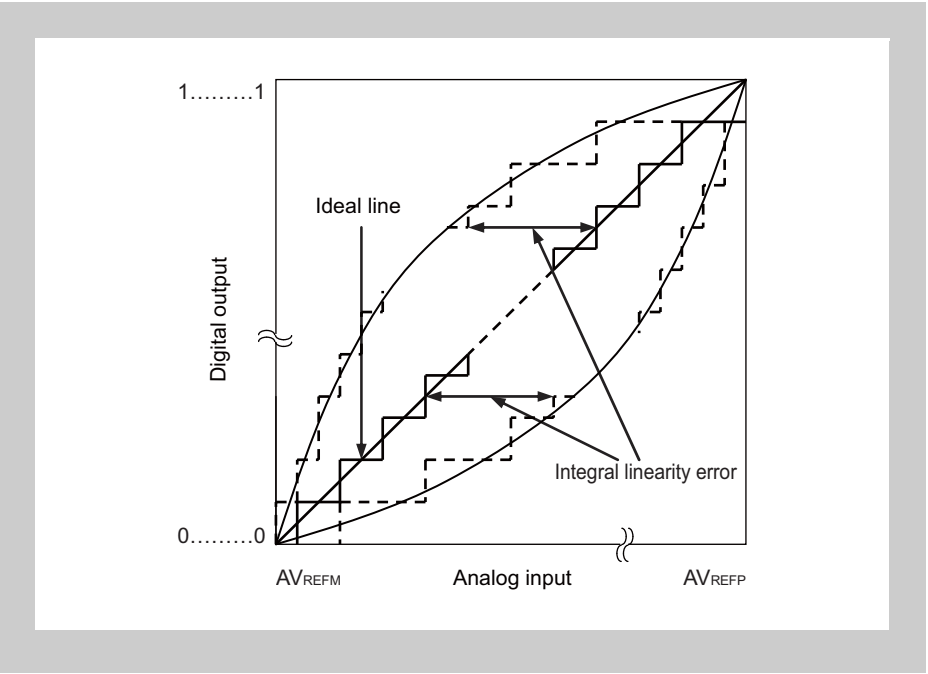


Figure 25-30 Integral Linearity Error

(8) Conversion Time

The conversion time is the time required to obtain the digital output after the analog input voltage is applied.

The conversion time in the characteristics table includes the sampling time.

(9) Sampling Time

The sampling time is the duration while the analog switch is on to allow the analog voltage to be sampled by the common sample and hold circuit.

(10) A/D Conversion Start Time

The A/D conversion start time is the time from occurrence of an A/D conversion trigger to the start of A/D conversion.

Section 26 Peripheral Interconnection (PIC)

26.1 Features of Peripheral Interconnection

The peripheral interconnection (PIC) realizes various functions by synchronous operation using multiple timers and by connecting the timer internal signals between the timers.

Meaning of n The unit number of each timer is represented by affix n (n = 0, 1).

For example, TAUAn represents TAUA0 or TAUA1.

Meaning of m The channel number of the timer TAUA is represented by affix m (m = 00 to 15)

Meaning of x Arbitrary value set for registers to be used

Meaning of y The registers to be used are identified by affix y (y = 200, 201, 202, 203, 210, 211, 212, 213, 30, 31, 50, 51).

Register addresses Refer to section 26.3, Peripheral Interconnection Registers.

Clock supply The PIC provides the following clock input. The PIC is connected to the PCLK.

Table 26-1 PIC Clock Supply

PIC	Clock Supply
PIC	PCLK

Input/output signals Refer to section 26.4, Connection Functions.

26.2 Functional Overview

The PIC has the following functions:

- Simultaneous start trigger function
- A/D conversion trigger selection function
- Three-phase PWM output function 1 with dead time
- High accuracy triangle wave PWM output function with dead time
- Delay pulse output function with dead time
- Trigger and pulse width measurement function
- A/D trigger encoder capture function
- Two-phase encoder function (control method 1)
- Two-phase encoder function (control method 2)
- Three-phase pulse input control function
- Three-phase encoder function
- CAN time stamp function

26.3 Peripheral Interconnection Registers

The functions of the peripheral interconnection are controlled and operated based on the settings of the following registers.

Table 26-2 PIC Registers

Register Name	Symbol	Address
Control register EN	PIC0EN	FFFF DB00$_H$
Simultaneous start trigger control register	PIC0SST	FFFF DB04$_H$
Simultaneous start control register 0	PIC0SSER0	FF81 C080$_H$
Simultaneous start control register 1	PIC0SSER1	FF81 C084$_H$
Simultaneous start control register 2	PIC0SSER2	FF81 C088$_H$
Hi-Z output control register 0	PIC0HIZCEN0	FF81 C0B4$_H$
Hi-Z output control register 1	PIC0HIZCEN1	FF81 C0B8$_H$
Hi-Z output control register 2	PIC0HIZCEN2	FF81 C0DC$_H$
Hi-Z output control register 3	PIC0HIZCEN3	FF81 C0E4$_H$
A/D converter trigger output control register 400	PIC0ADTEN400	FF81 C0C4$_H$
A/D converter trigger output control register 401	PIC0ADTEN401	FF81 C0C8$_H$
A/D converter trigger output control register 402	PIC0ADTEN402	FF81 C0CC$_H$
A/D converter trigger output control register 410	PIC0ADTEN410	FF81 C0D0$_H$
A/D converter trigger output control register 411	PIC0ADTEN411	FF81 C0D4$_H$
A/D converter trigger output control register 412	PIC0ADTEN412	FF81 C0D8$_H$
Timer I/O control register 200	PIC0REG200	FF81 C08C$_H$
Timer I/O control register 201	PIC0REG201	FF81 C090$_H$
Timer I/O control register 202	PIC0REG202	FF81 C094$_H$
Timer I/O control register 203	PIC0REG203	FF81 C098$_H$
Timer I/O control register 210	PIC0REG210	FF81 C09C$_H$
Timer I/O control register 211	PIC0REG211	FF81 C0A0$_H$
Timer I/O control register 212	PIC0REG212	FF81 C0A4$_H$
Timer I/O control register 213	PIC0REG213	FF81 C0A8$_H$
Timer I/O control register 30	PIC0REG30	FF81 C0BC$_H$
Timer I/O control register 31	PIC0REG31	FF81 C0C0$_H$
Timer I/O control register 50	PIC0REG50	FF81 C0E0$_H$
Timer I/O control register 51	PIC0REG51	FF81 C0E8$_H$

Caution For bit definition for the registers above, refer to the description of the pertinent register.

For signal connection between the timers, various registers are used depending on the functions. The combinations of the control registers to be used for various functions are shown in the following tables.

Table 26-3 Registers for Various Functions (1/2)

Section No.	Function Name	PIC0SSER			PIC0HIZCEN				PIC0ADTEN					
		0	1	2	0	1	2	3	400	401	402	410	411	412
26.4.1	Simultaneous Start Trigger Function	√	√	√	—	—	—	—	—	—	—	—	—	—
26.4.2	ADC Trigger Selection Function	—	—	—	—	—	—	—	√	√	√	√	√	√
26.4.3	Three-Phase PWM Output Function with Dead Time	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.4	High Accuracy Triangle Wave PWM Output Function with Dead Time	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.5	Delay Pulse Output Function with Dead Time	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.6	Trigger and Pulse Width Measurement Function	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.7	A/D Trigger Encoder Capture Function	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.8	Two-Phase Encoder Control Function (Control Method 1)	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.9	Two-Phase Encoder Control Function (Control Method 2)	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.10	Three-Phase Pulse Input Control Function	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.11	Three-Phase Encoder Control Function	—	—	—	—	—	—	—	—	—	—	—	—	—
26.4.12	CAN Time Stamp Function	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 26-4 Registers for Various Functions (2/2)

Section No.	Function Name	Register Name (PIC0REGy)											
		200	201	202	203	210	211	212	213	30	31	50	51
26.4.1	Simultaneous Start Trigger Function	—	—	—	—	—	—	—	—	—	—	—	—
26.4.2	ADC Trigger Selection Function	—	—	—	—	—	—	—	—	—	—	—	—
26.4.3	Three-Phase PWM Output Function with Dead Time	—	—	√	√	—	—	√	√	—	—	—	—
26.4.4	High Accuracy Triangle Wave PWM Output Function with Dead Time	√	√	√	√	√	√	√	√	—	—	—	—
26.4.5	Delay Pulse Output Function with Dead Time	—	—	√	√	—	—	√	√	—	—	—	—
26.4.6	Trigger and Pulse Width Measurement Function	—	—	—	—	—	—	—	—	—	√	—	—
26.4.7	A/D Trigger Encoder Capture Function	—	—	—	—	—	—	—	—	√	—	—	—
26.4.8	Two-Phase Encoder Control Function (Control Method 1)	—	—	—	—	—	—	—	—	√	—	√	√
26.4.9	Two-Phase Encoder Control Function (Control Method 2)	—	—	—	—	—	—	—	—	√	—	√	√
26.4.10	Three-Phase Pulse Input Control Function	√	—	—	—	√	—	—	—	—	—	√	√
26.4.11	Three-Phase Encoder Control Function	√	—	—	—	√	—	—	—	—	—	√	√
26.4.12	CAN Time Stamp Function	—	—	—	—	—	—	—	—	—	√	—	—

26.3.1 Start of PIC Function

Set the following register to use the PIC functions.

(1) Control Register EN (PIC0EN)

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

Address FF81 DB00$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PIC0EN0
R	R	R	R	R	R	R	R/W

Table 26-5 PIC0EN Contents

Bit Position	Bit Name	Function
0	PIC0EN0	Enables or disables the PIC functions. 0: Disables the PIC functions. 1: Enables the PIC functions.

26.4 Connection Functions

26.4.1 Simultaneous Start Trigger Function

26.4.1.1 Functional Overview

Allows any combination of the timers (TAUAn, TAUJn, TSG2n, TPBAn, OSTMn, and ENCAn) to be started simultaneously.

26.4.1.2 Configuration

Configuration/Timer Function	Timer
Timer configuration	TAUAn, TAUJn, TSG2n, TPBAn, OSTMn, ENCAn

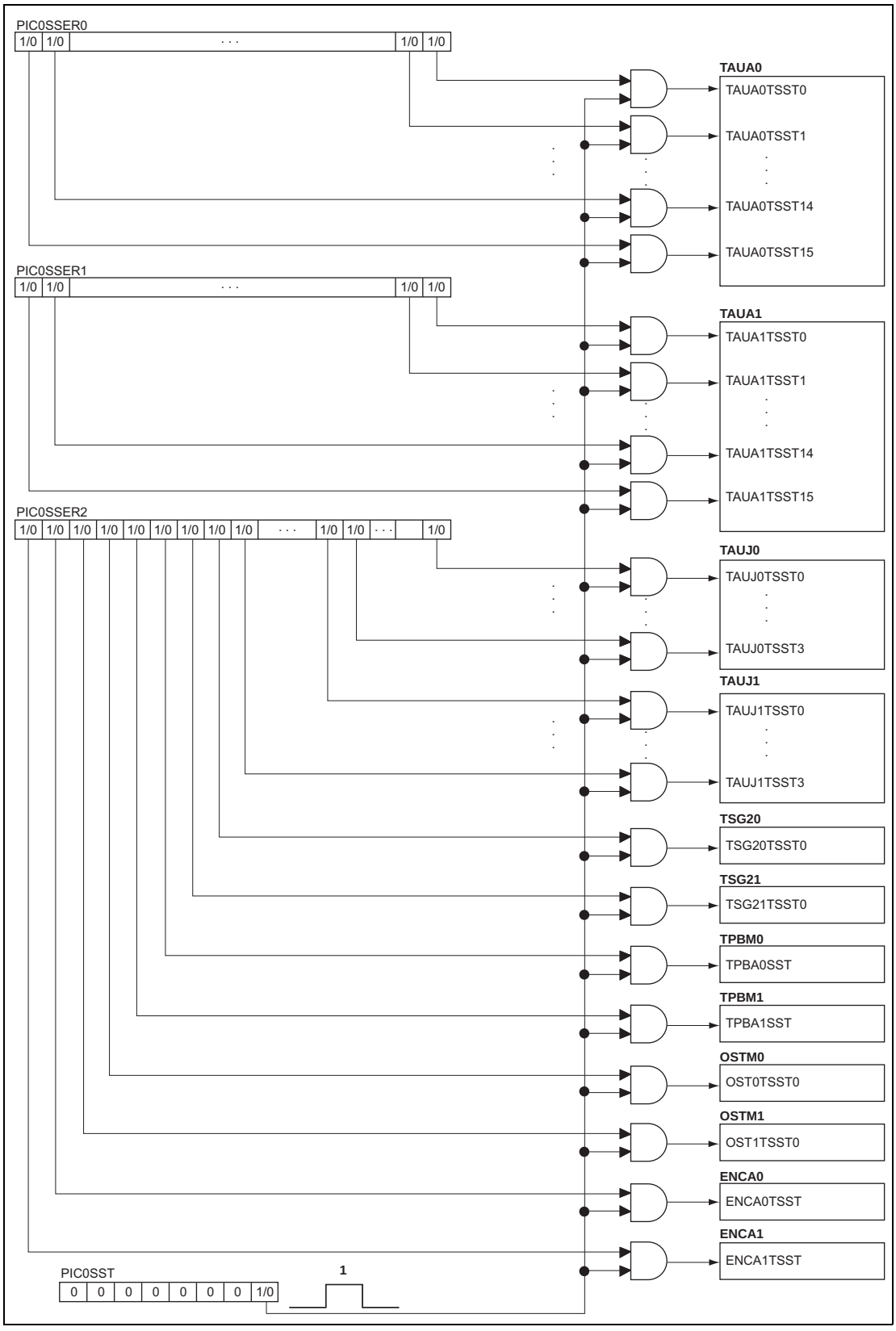


Figure 26-1 Block Diagram

26.4.1.3 Registers

(1) Simultaneous Start Control Register 0 (PIC0SSER0)

The PIC0SSER0 register enables the simultaneous start trigger for the TAUA0 channels.

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

Address FF81 C080$_H$

Initial value 0000$_H$

This register is initialized by any reset.

15	14	13	12	11	10	9	8
PIC0SSER015	PIC0SSER014	PIC0SSER013	PIC0SSER012	PIC0SSER011	PIC0SSER010	PIC0SSER009	PIC0SSER008
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0SSER007	PIC0SSER006	PIC0SSER005	PIC0SSER004	PIC0SSER003	PIC0SSER002	PIC0SSER001	PIC0SSER000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26-6 PIC0SSER0 Contents

Bit Position	Bit Name	Function
m	PIC0SSER0m	Enables the simultaneous start trigger for CHm of TAUA0. 0: Disables the simultaneous start trigger for CHm of TAUA0. 1: Enables the simultaneous start trigger for CHm of TAUA0.

(2) Simultaneous Start Control Register 1 (PIC0SSER1)

The PIC0SSER1 register enables the simultaneous start trigger for the TAUA1 channels.

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

Address FF81 C084$_H$

Initial value 0000$_H$

This register is initialized by any reset.

15	14	13	12	11	10	9	8
PIC0SSER115	PIC0SSER114	PIC0SSER113	PIC0SSER112	PIC0SSER111	PIC0SSER110	PIC0SSER109	PIC0SSER108
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0SSER107	PIC0SSER106	PIC0SSER105	PIC0SSER104	PIC0SSER103	PIC0SSER102	PIC0SSER101	PIC0SSER100
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26-7 PIC0SSER1 Contents

Bit Position	Bit Name	Function
m	PIC0SSER1m	Enables the simultaneous start trigger for CHm of TAUA1. 0: Disables the simultaneous start trigger for CHm of TAUA1. 1: Enables the simultaneous start trigger for CHm of TAUA1.

(3) Simultaneous Start Control Register 2 (PIC0SSER2)

The PIC0SSER2 register enables the simultaneous start trigger for TAUJ, TSG2, TPBA, OSTM, and ENCA.

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

Address FF81 C088$_H$

Initial value 0000$_H$

This register is initialized by any reset.

15	14	13	12	11	10	9	8
PIC0SSER215	PIC0SSER214	PIC0SSER213	PIC0SSER212	PIC0SSER211	PIC0SSER210	PIC0SSER209	PIC0SSER208
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0SSER207	PIC0SSER206	PIC0SSER205	PIC0SSER204	PIC0SSER203	PIC0SSER202	PIC0SSER201	PIC0SSER200
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26-8 PIC0SSER2 Contents (1/2)

Bit Position	Bit Name	Function
15	PIC0SSER215	Enables the simultaneous start trigger for ENCA1. 0: Disables the simultaneous start trigger for ENCA1. 1: Enables the simultaneous start trigger for ENCA1.
14	PIC0SSER214	Enables the simultaneous start trigger for ENCA0. 0: Disables the simultaneous start trigger for ENCA0. 1: Enables the simultaneous start trigger for ENCA0.
13	PIC0SSER213	Enables the simultaneous start trigger for OSTM1. 0: Disables the simultaneous start trigger for OSTM1. 1: Enables the simultaneous start trigger for OSTM1.
12	PIC0SSER212	Enables the simultaneous start trigger for OSTM0. 0: Disables the simultaneous start trigger for OSTM0. 1: Enables the simultaneous start trigger for OSTM0.
11	PIC0SSER211	Enables the simultaneous start trigger for TPBA1. 0: Disables the simultaneous start trigger for TPBA1. 1: Enables the simultaneous start trigger for TPBA1.
10	PIC0SSER210	Enables the simultaneous start trigger for TPBA0. 0: Disables the simultaneous start trigger for TPBA0. 1: Enables the simultaneous start trigger for TPBA0.
9	PIC0SSER209	Enables the simultaneous start trigger for TSG21. 0: Disables the simultaneous start trigger for TSG21. 1: Enables the simultaneous start trigger for TSG21.
8	PIC0SSER208	Enables the simultaneous start trigger for TSG20. 0: Disables the simultaneous start trigger for TSG20. 1: Enables the simultaneous start trigger for TSG20.
7	PIC0SSER207	Enables the simultaneous start trigger for CH03 of TAUJ1. 0: Disables the simultaneous start trigger for CH03 of TAUJ1. 1: Enables the simultaneous start trigger for CH03 of TAUJ1.
6	PIC0SSER206	Enables the simultaneous start trigger for CH02 of TAUJ1. 0: Disables the simultaneous start trigger for CH02 of TAUJ1. 1: Enables the simultaneous start trigger for CH02 of TAUJ1.

Table 26-8 PIC0SSER2 Contents (2/2)

Bit Position	Bit Name	Function
5	PIC0SSER205	Enables the simultaneous start trigger for CH01 of TAUJ1. 0: Disables the simultaneous start trigger for CH01 of TAUJ1. 1: Enables the simultaneous start trigger for CH01 of TAUJ1.
4	PIC0SSER204	Enables the simultaneous start trigger for CH00 of TAUJ1. 0: Disables the simultaneous start trigger for CH00 of TAUJ1. 1: Enables the simultaneous start trigger for CH00 of TAUJ1.
3	PIC0SSER203	Enables the simultaneous start trigger for CH03 of TAUJ0. 0: Disables the simultaneous start trigger for CH03 of TAUJ0. 1: Enables the simultaneous start trigger for CH03 of TAUJ0.
2	PIC0SSER202	Enables the simultaneous start trigger for CH02 of TAUJ0. 0: Disables the simultaneous start trigger for CH02 of TAUJ0. 1: Enables the simultaneous start trigger for CH02 of TAUJ0.
1	PIC0SSER201	Enables the simultaneous start trigger for CH01 of TAUJ0. 0: Disables the simultaneous start trigger for CH01 of TAUJ0. 1: Enables the simultaneous start trigger for CH01 of TAUJ0.
0	PIC0SSER200	Enables the simultaneous start trigger for CH00 of TAUJ0. 0: Disables the simultaneous start trigger for CH00 of TAUJ0. 1: Enables the simultaneous start trigger for CH00 of TAUJ0.

(4) Simultaneous Start Trigger Control Register (PIC0SST)

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

Address FFFF DB04$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SYNCTRG
W	W	W	W	W	W	W	W

Table 26-9 PIC0SST Contents

Bit Position	Bit Name	Function
0	SYNCTRG	Generates a start trigger for the timers for which simultaneous start is enabled. 0: Invalid 1: Generates a simultaneous start trigger. (Outputs a pulse of 1-PCLK wide.)

SYNCTRG is always read as 0.

26.4.1.4 Example of Operation

(1) Operation Example of Timer Configuration

In this example, an arbitrary combination of multiple timers (from among TAUAn, TAUJn, TSG2n, TPBAn, OSTMn, and ENCAn) that operate in arbitrary modes is started simultaneously.

(2) Setting Procedure

1. Initial Setting
   Set the operating mode for each of the timers TAUAn, TAUJn, TSG2n, TPBAn, OSTMn, and ENCAn (register setting is included). (For details on initial setting of TAUAn, TAUJn, TSG2n, TPBAn, OSTMn, and ENCAn, see the section of each timer.)
2. Enabling Simultaneous Start
   Enable simultaneous start of the target timers by setting the corresponding bits in PIC0SSER0, PIC0SSER1, and PIC0SSER2 to 1.
3. Start Trigger Output
   Set the SYNCTRG bit in PIC0SST0 to 1 to start the timers selected in step 2 simultaneously.
4. Repeat steps 2 and 3 for the channels that are not yet started. This allows each of the different target timer groups to be started simultaneously in separate timing.

26.4.1.5 Operation Flow

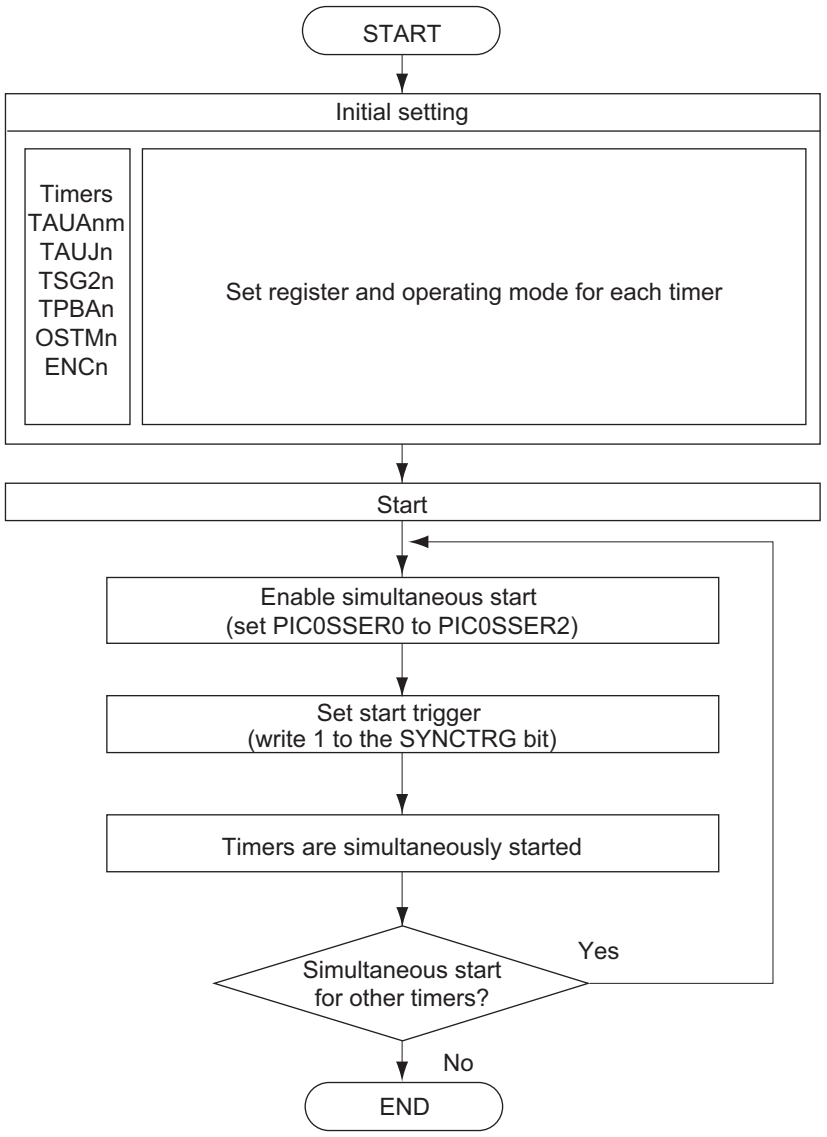


Figure 26-2 Setting Flow

Note For operation of the timers in simultaneous start function, refer to the section of each timer.

26.4.1.6 Register Setting for Various Functions

Refer to section 26.4.1.3, Registers.

26.4.2 ADC Trigger Selection Function

26.4.2.1 Functional Overview

The ADC consists of three channel groups and provides the ADC hardware triggers corresponding to each of the channel groups.

The internal trigger signals selected by ADCA0TSEL0 from TAUA0, TAUA1, ENCA0, ENCA1, TSG20, and TSG21 are ORed with the external trigger signal from the pin and the result can be input as the ADC hardware trigger signal for the pertinent channel group. Similarly, the internal trigger signals selected by ADCA0TSEL1 and ADCA0TSEL2 from TAUA0, TAUA1, ENCA0, ENCA1, TSG20, TSG21, TAPA0, and TAPA1 are ORed with the external trigger signal from the pin to generate the ADC hardware trigger signal for the pertinent channel groups.

The PIC provides the function to allow TAUAnINTm (internal trigger signal from each of TAUA0 and TAUA1 described above) to be selected as the ADC hardware trigger signal.

Caution The trigger signals from the TAUA0 and TAUA1 channels are not controlled by the TAUAnMD0 bit in TAUAnCMORm. Accordingly, they are used as the ADC hardware trigger signals for all the interrupts.

26.4.2.2 Configuration

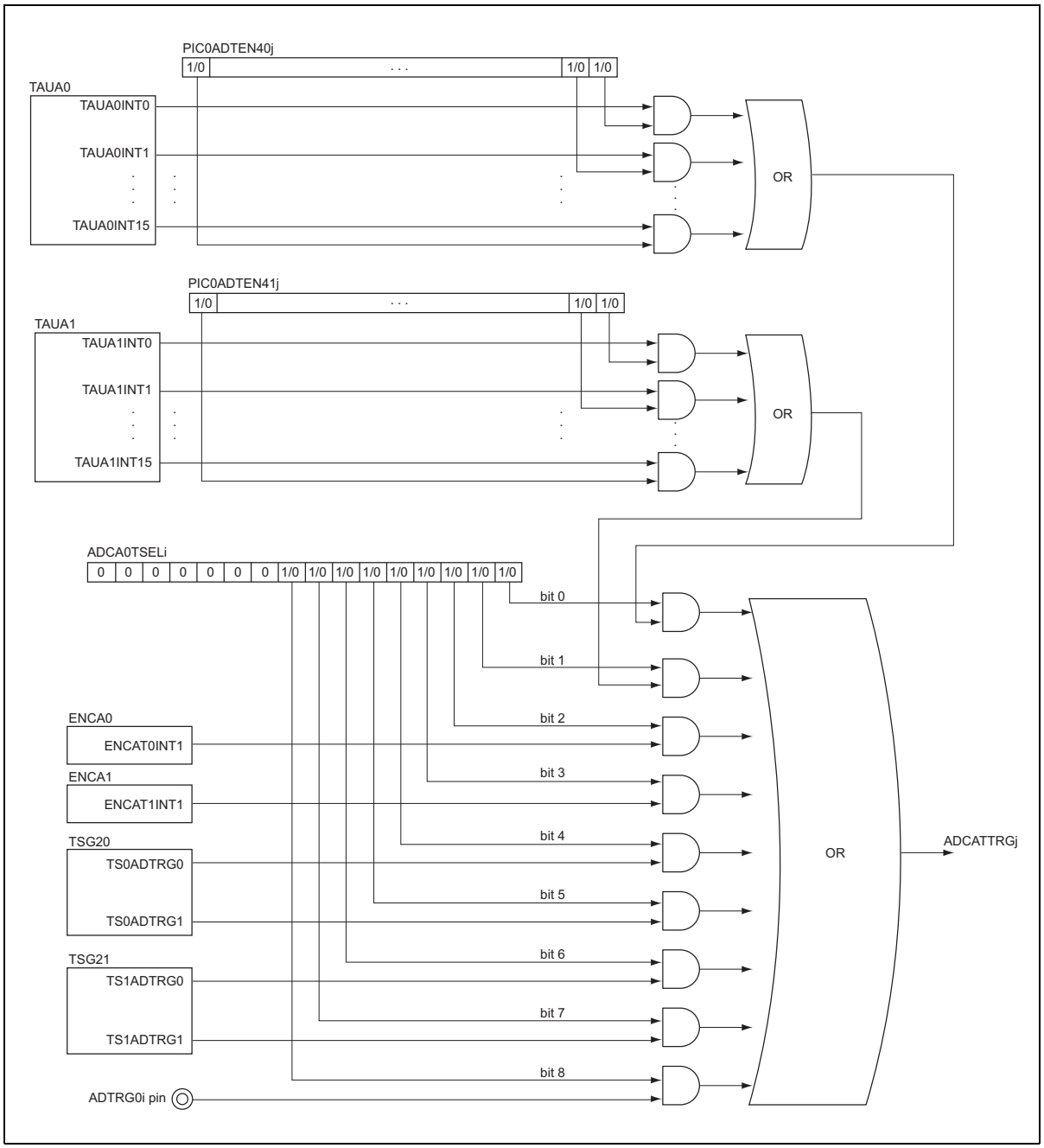


Figure 26-3 Block Diagram (i = 0, j = 0 to 2)

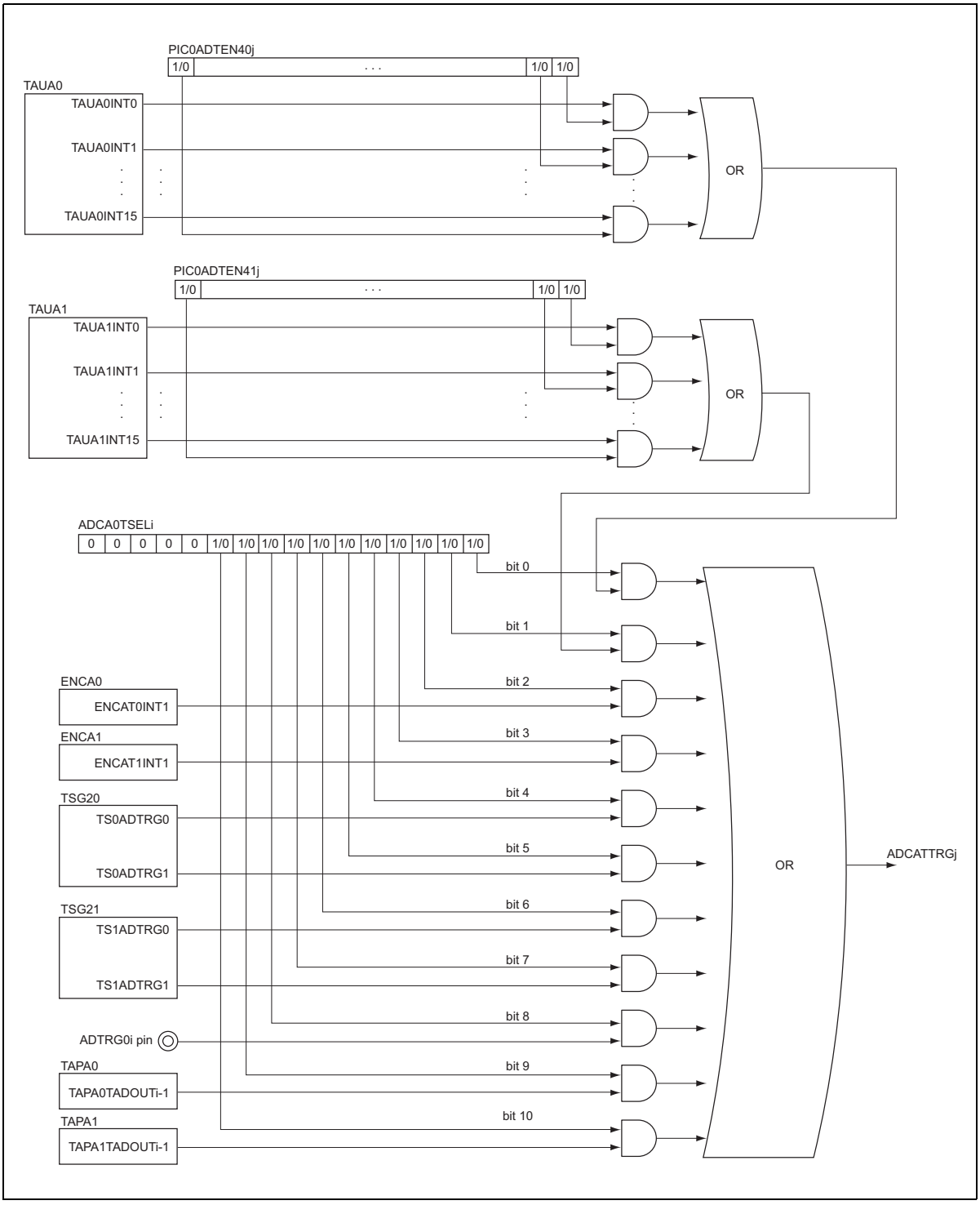


Figure 26-4 Block Diagram (i = 1 or 2, j = 0 to 2)

26.4.2.3 Registers

(1) A/D Converter Trigger Output Control Register 4nj (PIC0ADTEN4nj)

The PIC0ADTENnj register (j = 0 to 2) enables a trigger source from TAUAn CHm to be selected as an ADC trigger.

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

Address FF81 C0C4$_H$ (n = 0, j = 0), FF81 C0C8$_H$ (n = 0, j = 1), FF81 C0CC$_H$ (n = 0, j = 2)

FF81 C0D0$_H$ (n = 1, j = 0), FF81 C0D4$_H$ (n = 1, j = 1), FF81 C0D8$_H$ (n = 1, j = 2)

Initial value 0000$_H$

This register is initialized by any reset.

15	14	13	12	11	10	9	8
PIC0 ADTEN 4nj15	PIC0 ADTEN 4nj14	PIC0 ADTEN 4nj13	PIC0 ADTEN 4nj12	PIC0 ADTEN 4nj11	PIC0 ADTEN 4nj10	PIC0 ADTEN 4nj09	PIC0 ADTEN 4nj08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0 ADTEN 4nj07	PIC0 ADTEN 4nj06	PIC0 ADTEN 4nj05	PIC0 ADTEN 4nj04	PIC0 ADTEN 4nj03	PIC0 ADTEN 4nj02	PIC0 ADTEN 4nj01	PIC0 ADTEN 4nj00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26-10 PIC0ADTEN4nm Contents

Bit Position	Bit Name	Function
m	PIC0 ADTEN4njm	Sets a trigger source from TAUAn CHm. 0: Disables a trigger source from TAUAn CHm to be selected as an ADC trigger. 1: Enables a trigger source from TAUAn CHm to be selected as an ADC trigger.

(2) A/D Converter Trigger Selection Control Register i (ADCA0TSELi)

The ADCA0TSELi register selects a trigger for the ADC channel group i. (i = 0 to 2)

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

Address FF81 D108$_H$ (i = 0), FF81 D10C$_H$ (i = 1), FF81 D110$_H$ (i = 2)

Initial value 0000$_H$

This register is initialized by any reset.

15	14	13	12	11	10*1	9*1	8
0	0	0	0	0	ADCA0 TiSEL10	ADCA0 TiSEL09	ADCA0 TiSEL08
R	R	R	R	R	R/W	R/W	R/W

7	6	5	4	3	2	1	0
ADCA0 TiSEL07	ADCA0 TiSEL06	ADCA0 TiSEL05	ADCA0 TiSEL04	ADCA0 TiSEL03	ADCA0 TiSEL02	ADCA0 TiSEL01	ADCA0 TiSEL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 7. i = 1 and 2 for bits 9 and 10.

Table 26-11 ADCA0TSELi Contents (i = 0, j = 0 to 2)

Bit Position	Bit Name	Function
8	ADCA0TiSEL08	Selects the ADTRG0i pin as a trigger source of ADC channel group i. 0: Does not select the ADTRG0i pin as a trigger source. 1: Selects the ADTRG0i pin as a trigger source.
7	ADCA0TiSEL07	Selects TS1ADTRG1 as a trigger source of ADC channel group i. 0: Does not select TS1ADTRG1 as a trigger source. 1: Selects TS1ADTRG1 as a trigger source.
6	ADCA0TiSEL06	Selects TS1ADTRG0 as a trigger source of ADC channel group i. 0: Does not select TS1ADTRG0 as a trigger source. 1: Selects TS1ADTRG0 as a trigger source.
5	ADCA0TiSEL05	Selects TS0ADTRG1 as a trigger source of ADC channel group i. 0: Does not select TS0ADTRG1 as a trigger source. 1: Selects TS0ADTRG1 as a trigger source.
4	ADCA0TiSEL04	Selects TS0ADTRG0 as a trigger source of ADC channel group i. 0: Does not select TS0ADTRG0 as a trigger source. 1: Selects TS0ADTRG0 as a trigger source.
3	ADCA0TiSEL03	Selects ENCAT1INT1 as a trigger source of ADC channel group i. 0: Does not select ENCAT1INT1 as a trigger source. 1: Selects ENCAT1INT1 as a trigger source.
2	ADCA0TiSEL02	Selects ENCAT0INT1as a trigger source of ADC channel group i. 0: Does not select ENCAT0INT1 as a trigger source. 1: Selects ENCAT0INT1 as a trigger source.
1	ADCA0TiSEL01	Selects the trigger selected by the PIC0ADTEN41j register as a trigger source of ADC channel group i. 0: Does not select the trigger selected by the PIC0ADTEN41j register as a trigger source. 1: Selects the trigger selected by the PIC0ADTEN41j register as a trigger source.
0	ADCA0TiSEL00	Selects the trigger selected by the PIC0ADTEN40j register as a trigger source of ADC channel group i. 0: Does not select the trigger selected by the PIC0ADTEN40j register as a trigger source. 1: Selects the trigger selected by the PIC0ADTEN40j register as a trigger source.

Table 26-12 ADCA0TSELi Contents (i = 1 or 2, j = 0 to 2)

Bit Position	Bit Name	Function
10	ADCA0TiSEL10	Selects the TAPA1TADOUTi-1 pin as a trigger source of ADC channel group i. 0: Does not select the TAPA1TADOUTi-1 pin as a trigger source. 1: Selects the TAPA1TADOUTi-1 pin as a trigger source.
9	ADCA0TiSEL09	Selects the TAPA0TADOUTi-1 pin as a trigger source of ADC channel group i. 0: Does not select the TAPA0TADOUTi-1 pin as a trigger source. 1: Selects the TAPA0TADOUTi-1 pin as a trigger source.
8	ADCA0TiSEL08	Selects the ADTRG0i pin as a trigger source of ADC channel group i. 0: Does not select the ADTRG0i pin as a trigger source. 1: Selects the ADTRG0i pin as a trigger source.
7	ADCA0TiSEL07	Selects TS1ADTRG1 as a trigger source of ADC channel group i. 0: Does not select TS1ADTRG1 as a trigger source. 1: Selects TS1ADTRG1 as a trigger source.
6	ADCA0TiSEL06	Selects TS1ADTRG0 as a trigger source of ADC channel group i. 0: Does not select TS1ADTRG0 as a trigger source. 1: Selects TS1ADTRG0 as a trigger source.
5	ADCA0TiSEL05	Selects TS0ADTRG1 as a trigger source of ADC channel group i. 0: Does not select TS0ADTRG1 as a trigger source. 1: Selects TS0ADTRG1 as a trigger source.
4	ADCA0TiSEL04	Selects TS0ADTRG0 as a trigger source of ADC channel group i. 0: Does not select TS0ADTRG0 as a trigger source. 1: Selects TS0ADTRG0 as a trigger source.
3	ADCA0TiSEL03	Selects ENCAT1INT1 as a trigger source of ADC channel group i. 0: Does not select ENCAT1INT1 as a trigger source. 1: Selects ENCAT1INT1 as a trigger source.
2	ADCA0TiSEL02	Selects ENCAT0INT1 as a trigger source of ADC channel group i. 0: Does not select ENCAT0INT1 as a trigger source. 1: Selects ENCAT0INT1 as a trigger source.
1	ADCA0TiSEL01	Selects the trigger selected by the PIC0ADTEN41j register as a trigger source of ADC channel group i. 0: Does not select the trigger selected by the PIC0ADTEN41j register as a trigger source. 1: Selects the trigger selected by the PIC0ADTEN41j register as a trigger source.
0	ADCA0TiSEL00	Selects the trigger selected by the PIC0ADTEN40j register as a trigger source of ADC channel group i. 0: Does not select the trigger selected by the PIC0ADTEN40j register as a trigger source. 1: Selects the trigger selected by the PIC0ADTEN40j register as a trigger source.

26.4.2.4 Example of Operation

1. Initial Setting
   Set the timers to use from among TAUA0, TAUA1, ENCA0, ENCA1, TSG20, TSG21, TAPA0, and TAPA1.
   (Refer to the specification of each timer for detailed initial setting procedure.)

2. Setting A/D Converter Trigger Output Control Register 4nj (PIC0ADTEN4nj)
   Set bit m in the A/D converter trigger output control register 4nj (PIC0ADTEN4nj; j = 0 to 2) to 1 to allow the interrupt trigger signal from each channel of TAUA0 and TAUA1 to be selected as a trigger of ADC channel group i.

   Note Set the registers while the AD converter is stopped (ADCA0CE = 0).

3. Setting A/D Converter Trigger Selection Control Register i (ADCA0TSELi)
   By setting the bit corresponding to each trigger to 1, the trigger signals are ORed and the result can be used as a trigger for ADC channel group i.

   Note Set the registers while the AD converter is stopped (ADCA0CE = 0).

4. Enabling TAUA0, TAUA1, ENCA0, ENCA1, TSG20, TSG21, TAPA0, and TAPA1 Operation
   Timers selected in step 1 from among TAUA0, TAUA1, ENCA0, ENCA1, TSG20, TSG21, TAPA0, and TAPA1 are started.

26.4.2.5 Operation Flow

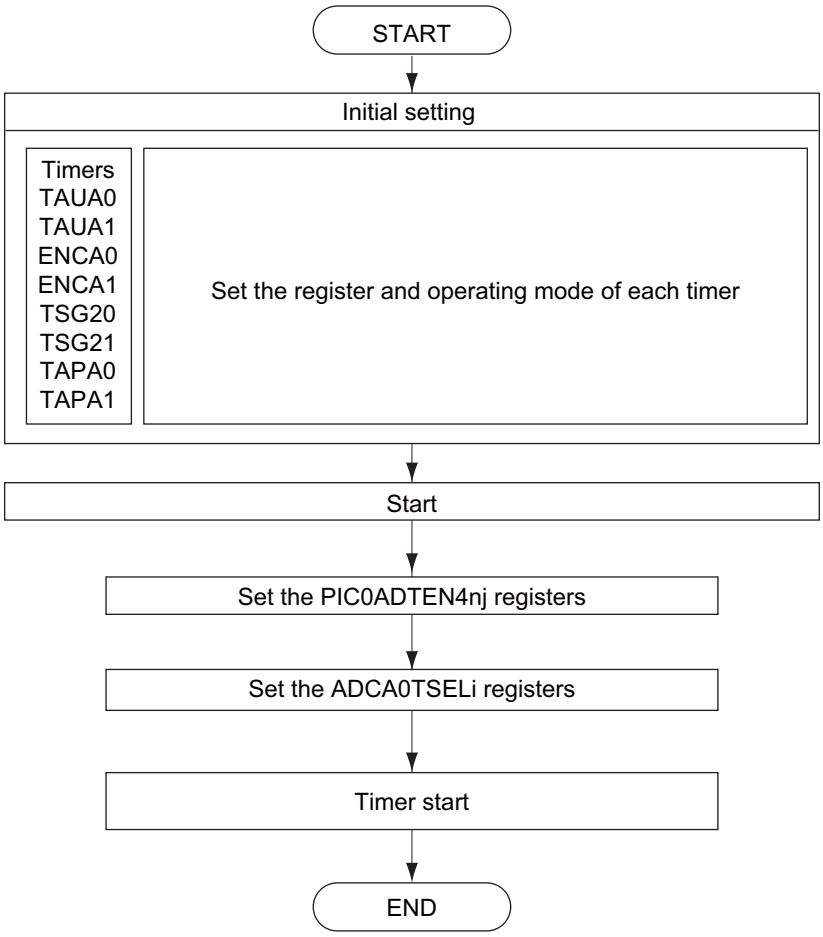


Figure 26-5 Operation Flow (i = 0 to 2, j = 0 to 2)

26.4.2.6 Register Setting Examples for Various Functions

Refer to section 26.4.2.3, Registers.

26.4.3 Three-Phase PWM Output Function with Dead Time

26.4.3.1 Functional Overview

This function generates and outputs three-phase PWM waveforms with the dead time by using the set signal (timing signal for changing to active level) and the clear signal (timing signal for changing to inactive level) generated once or less per one period.

The PWM output function of the TAUA sets only the clear timing in a period under the duty ratio specification. This function can set the set timing in addition to the clear timing to output more flexible PWM waveforms with the dead time.

26.4.3.2 Configuration

Note In this section, signal names are abbreviated as shown below.

INTm → TAUAnINTm

TINm → TAUAnTTINm

TOUTm → TAUAnTTOUTm

CDRm → TAUAnCDRm

CNTm → TAUAnCNTm

Configuration/ Timer Function	TAUA	TAPA
Timer configuration 1	TAUA0	TAPA0
Timer configuration 2	TAUA1	TAPA1

- Setting functions of TAUAn channels
  Since CH10, CH12, and CH14 only use TOUTm, they can be used for any function not using TOUTm (m = 10, 12 and 14).

CH	Function Name	M/S*	CDR Set Value	Description
02	PWM output function (CH02 is master for CH04 to CH09.)	M	Period	
04		S	Duty (U phase setting)	
05		S	Duty (U phase clearing)	
06		S	Duty (V phase setting)	
07		S	Duty (V phase clearing)	
08		S	Duty (W phase setting)	
09		S	Duty (W phase clearing)	
10	Any function not using TOUT10	S		TOUT10: U phase output
11	One-phase PWM output function	S	Dead time (U phase)	TOUT11: UB phase output
12	Any function not using TOUT12	S		TOUT12: V phase output
13	One-phase PWM output function	S	Dead time (V phase)	TOUT13: VB phase output
14	Any function not using TOUT14	S		TOUT14: W phase output
15	One-phase PWM output function	S	Dead time (W phase)	TOUT15: WB phase output

Note * M = master channel, S = slave channel

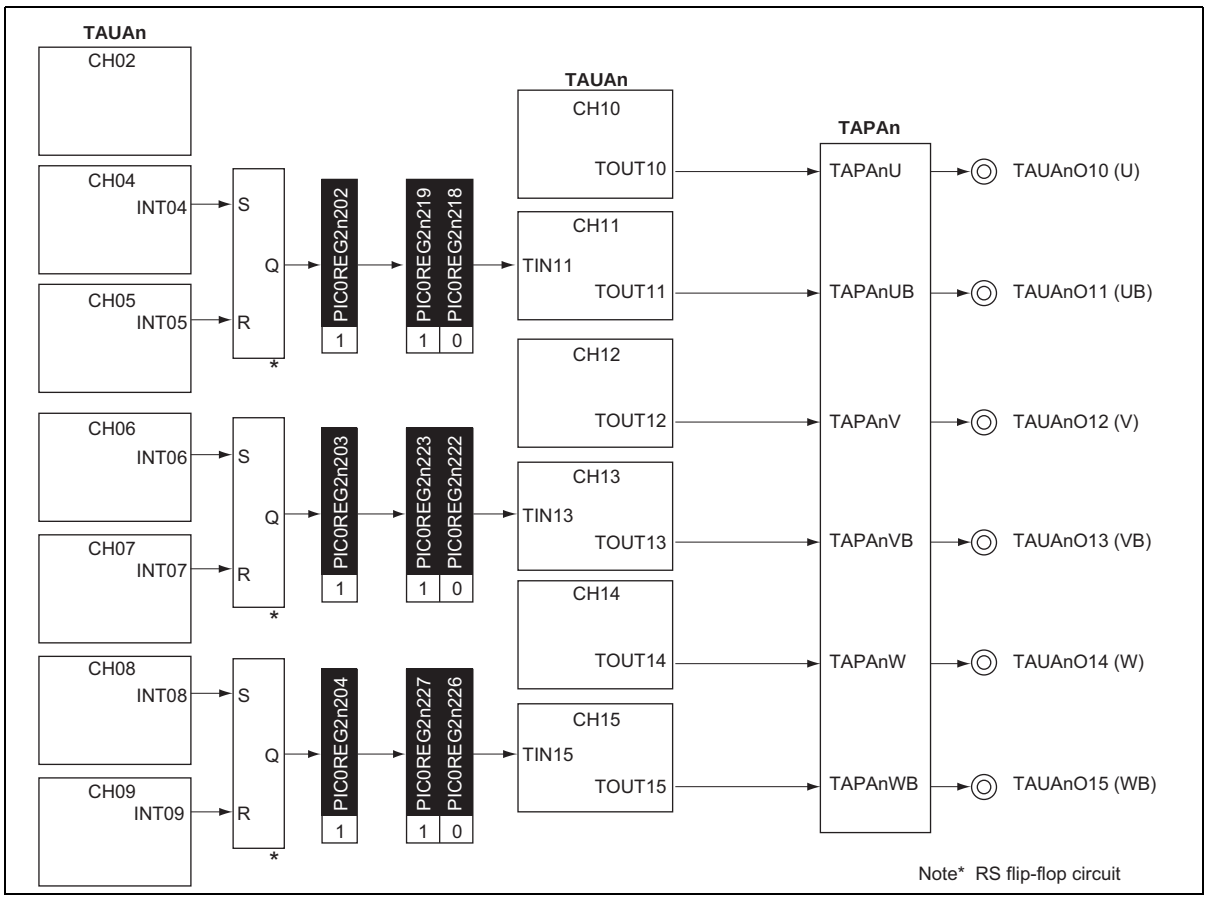


Figure 26-6 Block Diagram

26.4.3.3 Registers

(1) Timer I/O Control Register 2n2 (PIC0REG2n2)

Access This register can be read/written in 32-bit units.

Address FF81 C094$_H$ (n = 0), FF81 C0A4$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	PIC0REG2n227	PIC0REG2n226	0	0
R	R	R	R	R/W	R/W	R	R

23	22	21	20	19	18	17	16
PIC0REG2n223	PIC0REG2n222	0	0	PIC0REG2n219	PIC0REG2n218	0	0
R/W	R/W	R	R	R/W	R/W	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	0	0
R	R	R	R/W	R/W	R/W	R	R

Table 26-13 PIC0REG2n2 Contents (1/2)

Bit Position	Bit Name	Function
27 26	PIC0REG2n227 PIC0REG2n226	Selects the TIN input signal of TAUAn CH15. **PIC0REG2n227** / **PIC0REG2n226** / **Input Signal** 1 / 0 / Signal selected with PIC0REG2n204 bit. Other than above / Setting prohibited
23 22	PIC0REG2n223 PIC0REG2n222	Selects the TIN input signal of TAUAn CH13. **PIC0REG2n223** / **PIC0REG2n222** / **Input Signal** 1 / 0 / Signal selected with PIC0REG2n203 bit. Other than above / Setting prohibited
19 18	PIC0REG2n219 PIC0REG2n218	Selects the TIN input signal of TAUAn CH11 **PIC0REG2n219** / **PIC0REG2n218** / **Input Signal** 1 / 0 / Signal selected with PIC0REG2n202 bit. Other than above / Setting prohibited
4	PIC0REG2n204	Selects the TIN input signal of TAUAn CH15. 0: Setting prohibited. 1: Selects the set and clear signal output according to TAUAnINT08 and TAUAnINT09.

Table 26-13 PIC0REG2n2 Contents (2/2)

Bit Position	Bit Name	Function
3	PIC0REG2n203	Selects the TIN input signal of TAUAn CH13. 0: Setting prohibited. 1: Selects the set and clear signal output according to TAUAnINT06 and TAUAnINT07.
2	PIC0REG2n202	Selects the TIN input signal of TAUAn CH11. 0: Setting prohibited. 1: Selects the set and clear signal output according to TAUAnINT04 and TAUAnINT05.

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

(2) Timer I/O Control Register 2n3 (PIC0REG2n3)

Access This register can be read/written in 32-bit units.

Address FF81 C098$_H$ (n = 0), FF81 C0A8$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

26.4.3.4 Example of Operation

This function generates and outputs three-phase PWM waveforms with the dead time by using the set signal and the clear signal generated once or less per one period.

- PWM output function
- One-phase PWM output function

The following function of the PIC is used to generate PWM waveforms supplied to the input TINm (m = 11, 13, and 15) of the one-phase PWM output function using the set signal and the clear signal generated by the PWM output function.

- RS flip-flop circuit

The three-phase PWM output function is actually implemented by assigning the one-phase PWM output with the dead time obtained by the above functions to the U, V, and W phases, which allows independent set/clear signal setting for the PWM output of each phase. Since the operation of the three phases is identical except for the channel assigned, only the operation of the U phase is explained below.

(1) PWM Output Function

CH02, CH04, and CH05 are used in combination.

Setting the period, U phase set value, and U phase clear value to CDR02, CDR04, and CDR05, respectively generates the set and clear signals of the RS flip-flop circuit for generating the input TIN11 of the one-phase PWM output function using INT04 and INT05.

Similarly, for V phase and W phase, the set and clear signals can be generated using CH06 and CH07 (V phase) and CH08 and CH09 (W phase) instead of the above mentioned CH04 and CH05 for U phase.

(2) One-Phase PWM Output Function

CH10 and CH11 are combined to allow the one-phase PWM to be output from TOUT10 and TOUT11.

By setting the dead time to CDR11, the one-phase PWM with the dead time inserted to the TIN11 input is output.

Similarly, the one-phase PWM with the dead time can be output for the V and W phases using CH12 and CH13 (V phase), and CH14 and CH15 (W phase).

Caution Set the same operating clock for all the TAUAn channels that are used for the PWM output function and one-phase PWM output function.

For details of TAUA functions, refer to section 13, Timer Array Unit A (TAUA).

(3) RS Flip-Flop Circuit

The RS flip-flop circuit receives the U phase set signal generated in CH04 and U phase clear signal generated in CH05 to generate PWM waveforms to be supplied to the input TIN11 of the one-phase PWM output function.

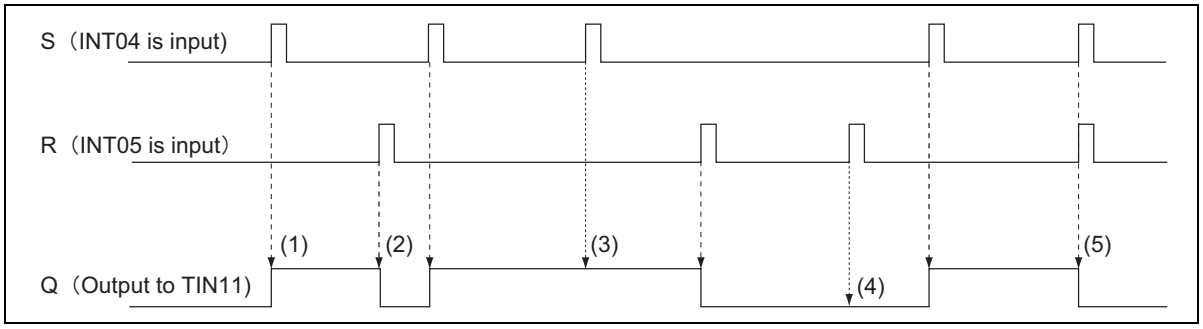


Figure 26-7 Operation Timing of RS Flip-Flop Circuit (Example of U Phase)

(1) When a signal is input to the input S, the output Q goes high at the rising edge of S.

(2) When a signal is input to the input R, the output Q goes low at the rising edge of R.

(3) The output Q is not affected by a signal input to the input S while the output Q is high.

(4) The output Q is not affected by a signal input to the input R while the output Q is low.

(5) When signals are input to the inputs S and R simultaneously, the input R takes precedence over the input S and the output Q goes low at the rising edge of R.

For V phase and W phase, INT06 and INT07 (V phase) and INT08 and INT09 (W phase) are used as input to generate PWM waveforms supplied to TIN13 (V phase) and TIN15 (W phase).

The PWM output function generates the timing to change the PWM output level generated by the one-phase PWM output function.

The PWM output function generates a set signal and a clear signal as the PWM active and inactive level output timing signals, respectively. The set and clear signals are input to the RS flip-flop circuit to generate the PWM signal that changes its level at an arbitrary timing.

The one-phase PWM output function generates positive or negative phase PWM and inserts the dead time according to the changes of the generated PWM signal, and then outputs the one-phase PWM signal.

The PIC provides a connection for using the set and clear signals generated by the PWM output function as the TIN input of the one-phase PWM output function through use of the RS flip-flop circuit.

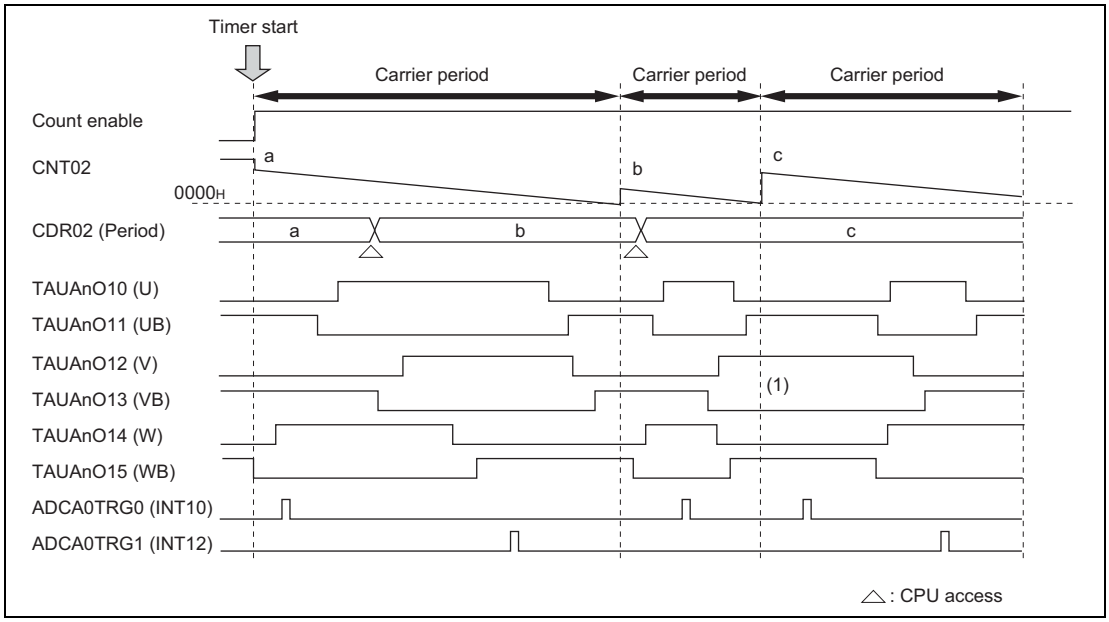


Figure 26-8 Three-Phase PWM (U/UB, V/VB, W/WB) Output Example with Dead Time

Figure 26-8 shows a typical example of three-phase PWM output with the dead time.

The PWM output extending over two carrier periods ((1) in the figure) is possible, depending on the setting of the set/clear signal output timing.

The ADCA0TRG0 and ADCA0TRG1 outputs (the lowest two signals in the figure) are the examples in which the CNT and INT signals of CH10 and CH12 (not used for the one-phase PWM output function) are used to output the AD conversion trigger by the AD trigger output function Type-1.

As described above, in the one-phase PWM output function, the channels for positive phase output only use TOUTm, which outputs signals; therefore, any function is available that only uses CNTm, CDRm, and/or INTm. For details, refer to section 13, Timer Array Unit A (TAUA).

(m = 10, 12 and 14)

The timing charts of the one-phase PWM output function with the dead time are shown in the following pages.

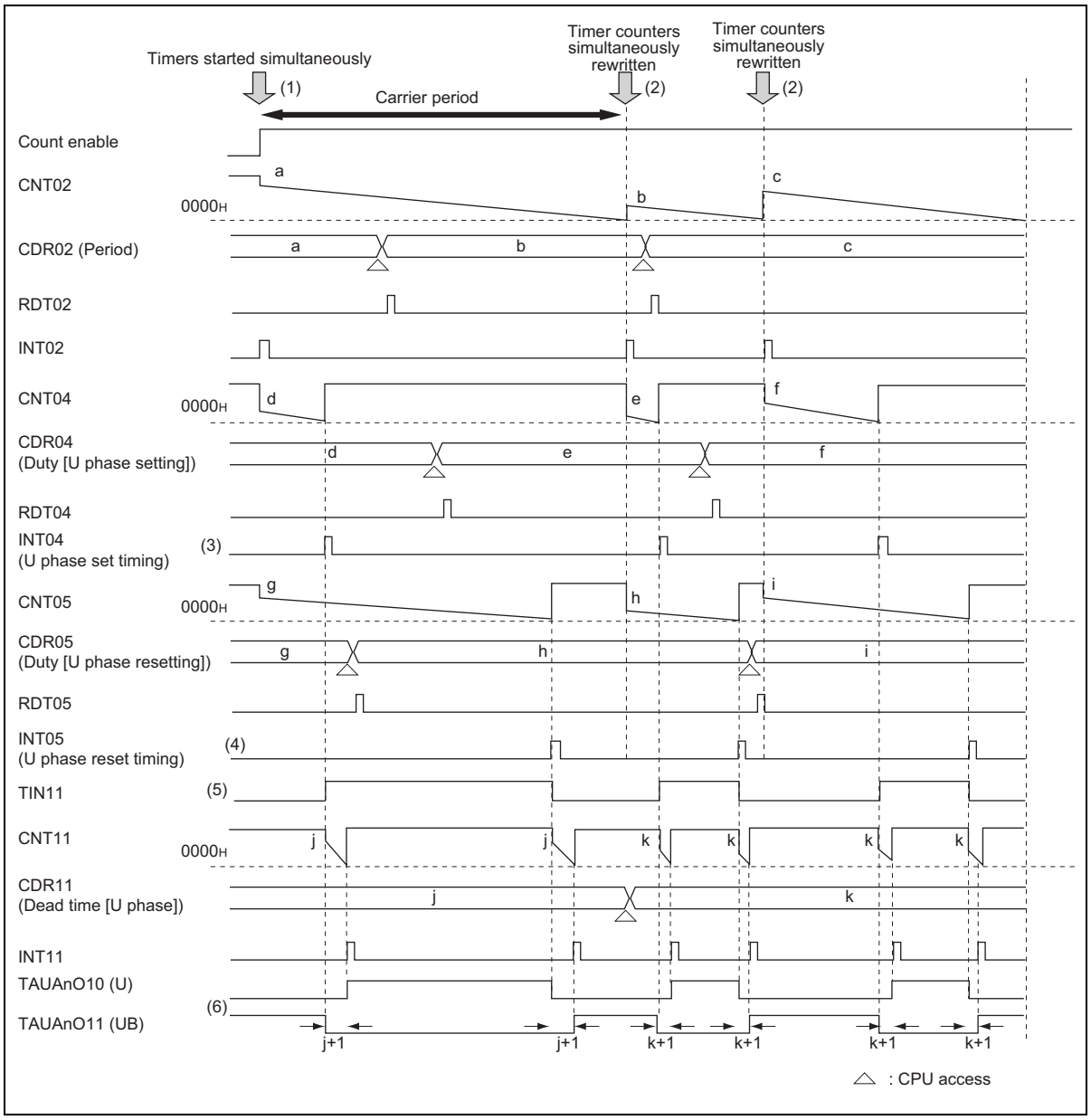


Figure 26-9 One-Phase PWM (U Phase and UB Phase) Output Example with Dead Time

The following describes the operation example of the timer configuration for U phase PWM output shown in Figure 26-9.

(1) Using the simultaneous timer start function, CH02 (carrier period timer), CH04 (U phase set signal output timing timer), and CH05 (U phase clear signal output timing timer) are started.
Although the timer CH11 is started as well, it will not start counting until the TIN11 edge (the count start timing) is detected.

(2) For CH04 and CH05, CH02 underflow allows the set values in CDR04 and CDR05 to be reloaded to CNT04 and CNT05, respectively.

(3) CH04 underflow allows the U phase set timing signal (INT04) to be generated.

(4) CH05 underflow allows the U phase clear timing signal (INT05) to be generated.

(5) The PIC function supplies the output from the RS flip-flop circuit generated from the INT04 (set timing signal) and INT05 (clear timing signal) inputs to the input TIN11 of the one-phase PWM output function.

(6) With the one-phase PWM output function, PWM waveform with the dead time is generated and output, according to the TIN11 edges detected.

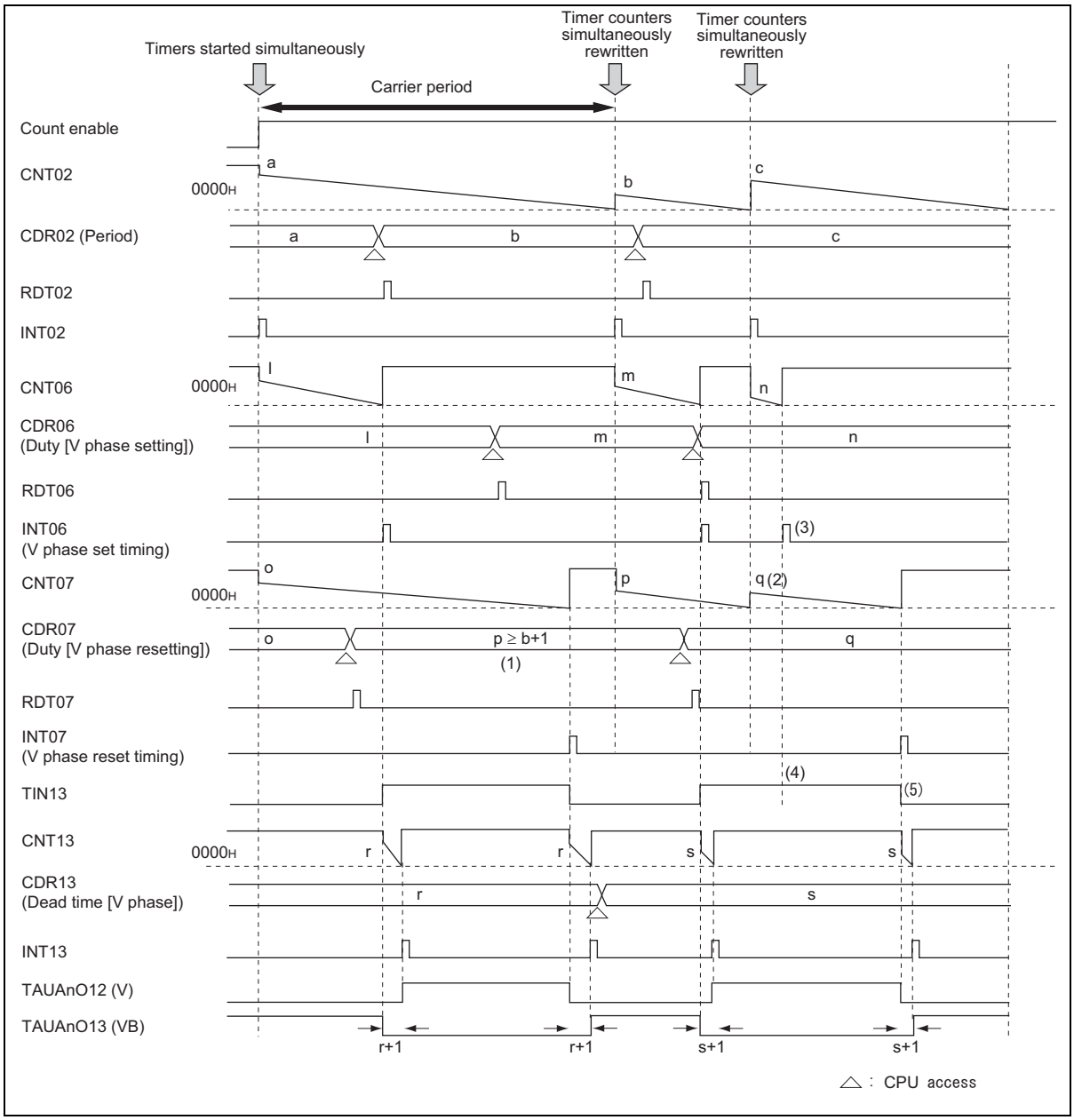


Figure 26-10 One-Phase PWM (V Phase and VB Phase) Output Example with Dead Time

The following describes the operation example of the timer configuration for V phase PWM output shown in Figure 26-10.

For the operation from the simultaneous timer start to PWM output by the one-phase PWM output function, refer to the description of the operation example of the U phase.

(1) When CH07 (V phase clear signal output timing timer) for generating the V phase clear timing signal (INT07) is set to the value greater than the set value for CH02 (carrier period timer),

(2) CH02 underflow occurs before generation of the V phase clear timing signal (INT07) caused by CH07 underflow, allowing the set value for CH07 to be reloaded.

(3) The above operation prevents generation of the V phase clear timing signal (INT07), which should be generated normally, and allows the V phase set timing signals (INT06) to be generated successively.

(4) In this case, PWM output waveform is not affected by the operation above because the V phase set timing signal (INT06) is ignored by the RS flip-flop circuit. Accordingly, a waveform extending over two carrier periods is output as the PWM output.

(5) The PWM output changes at the next output timing of the V phase clear timing signal (INT07).

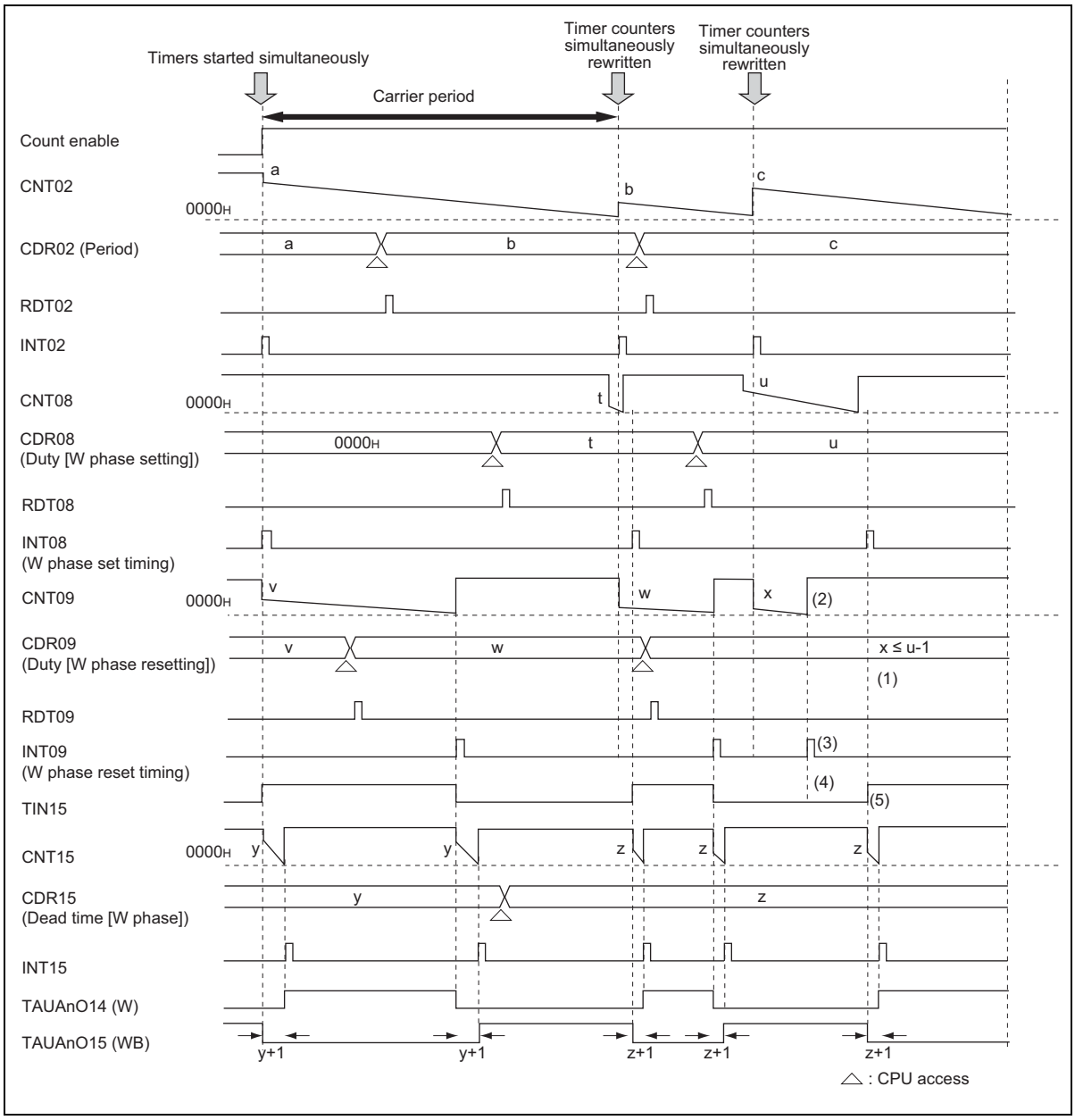


Figure 26-11 One-Phase PWM (W Phase and WB Phase) Output Example with Dead Time

The following describes the operation example of the timer configuration for W phase PWM output shown in Figure 26-11.

For the operation from the simultaneous timer start to PWM output by the one-phase PWM output function, refer to the description of the operation example of the U phase.

(1) When CH09 (W phase clear signal output timing timer) for generating the W phase clear timing signal (INT09) is set to the value smaller than the set value for CH08 (W phase set signal output timing timer),

(2) CH09 underflow occurs before generation of the W phase set timing signal (INT08) caused by CH08 underflow, allowing the W phase clear timing signal (INT09) to be generated.

(3) The above operation allows the W phase clear timing signals (INT09) to be generated successively.

(4) In this case, PWM output waveform is not affected by the operation above because the successively generated W phase clear timing signals (INT09) are ignored by the RS flip-flop circuit.

(5) The PWM output changes at the next output timing of the W phase set timing signal (INT08).

26.4.3.5 Operation Flow

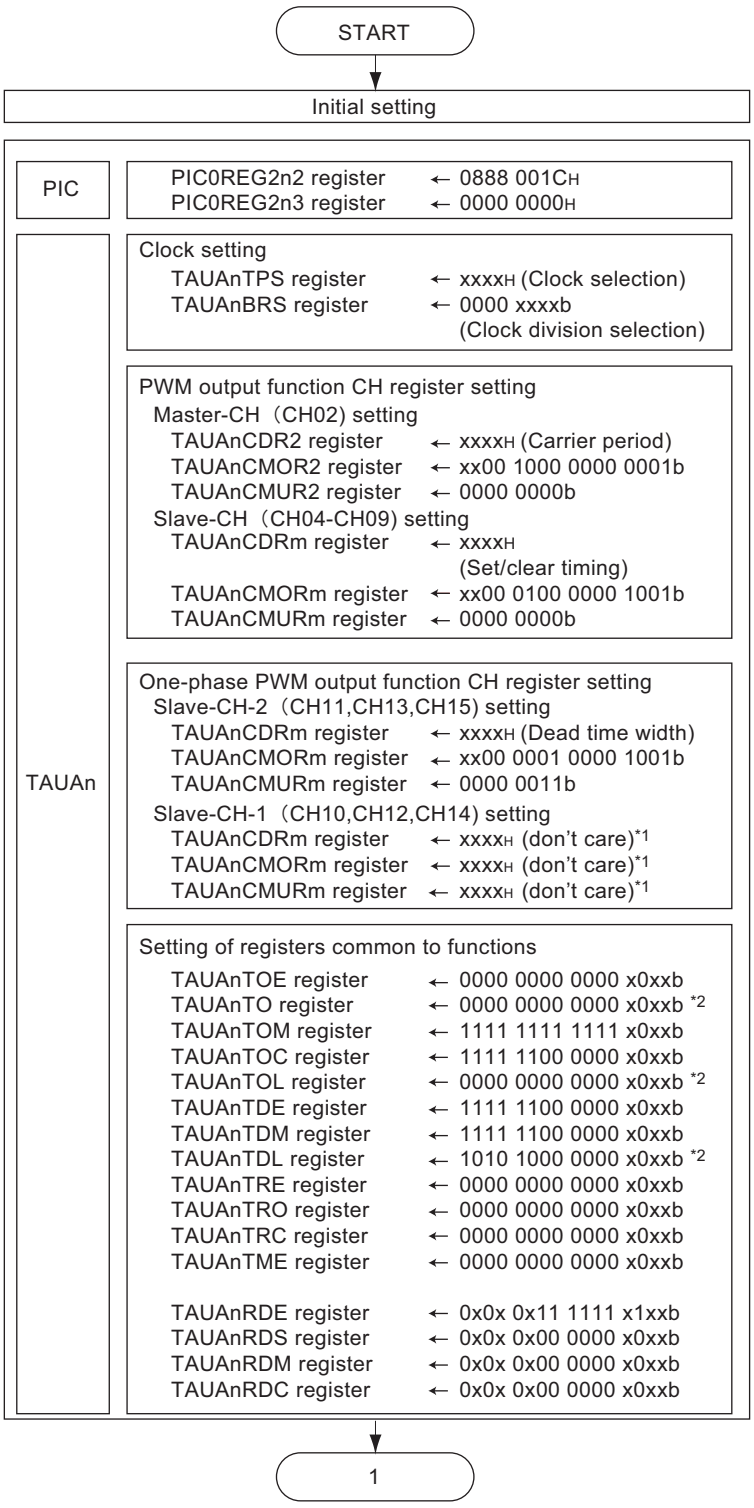


Figure 26-12 Setting Flow (Active High) (1/2)

Note 1. Set the function not using TOUTm.
Note 2. Change the set value depending on the active level of the PWM output.

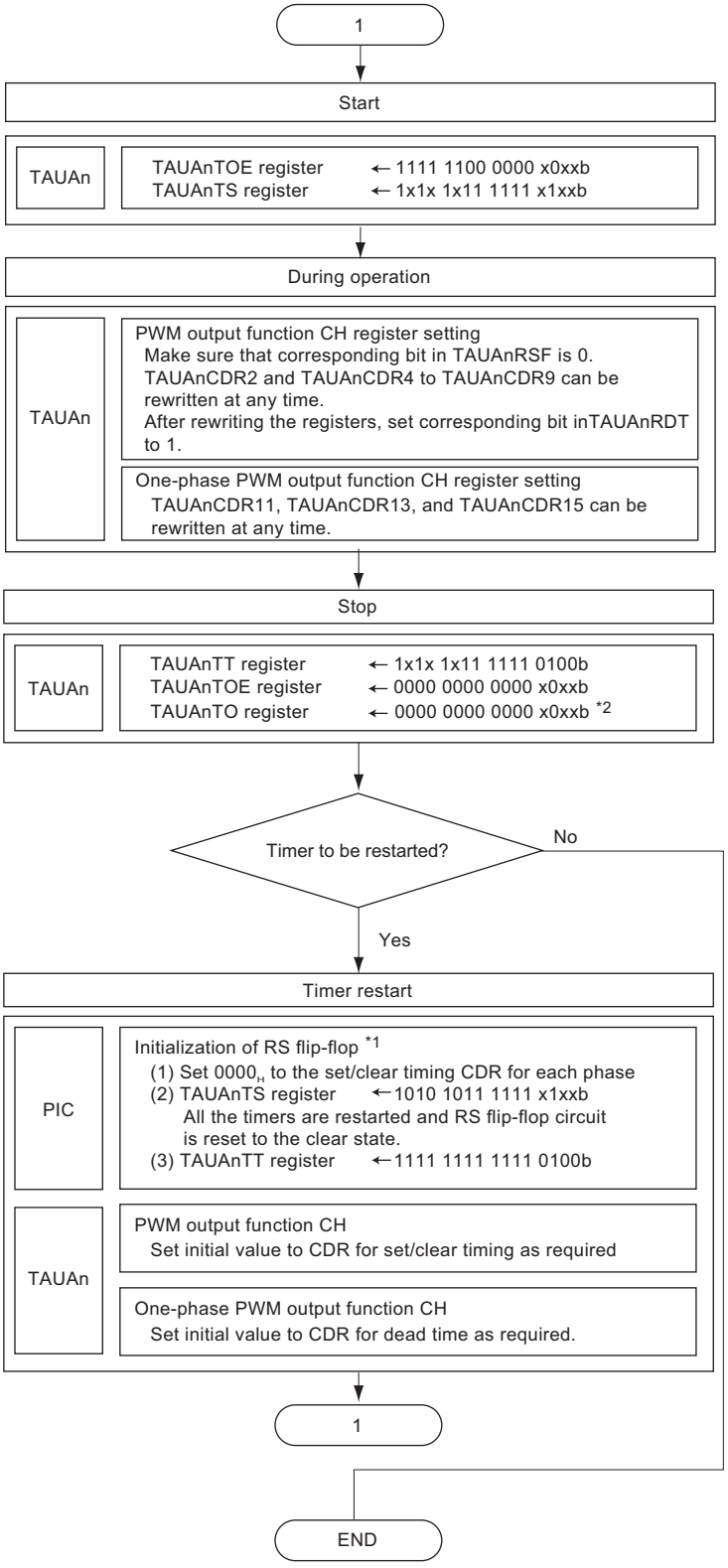


Figure 26-12 Setting Flow (Active High) (2/2)

Note 1. When initialization is not performed, the RS flip-flop circuit is placed in the state in which the timer is stopped, and in some cases, an unintended level of the timer restart pulse may be output.

Note 2. Change the set value depending on the active level of the PWM output.

26.4.3.6 Register Setting Examples for Various Functions

Table 26-14 TAUAn Setting (Active High)

Registers related to TAUAn CH02 (PWM output function Master-CH*1)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMOR2	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 0	
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	0	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	
	TAUAn CMUR2	1, 0	TIS1 TIS0	0 0	

Registers related to TAUAn CH04 to CH09 (PWM output function Save-CH*1) (m = 4 to 9)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 0 0	
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Note 1. Master-CH and Slave-CH are the names defined with the TAUA PWM output function. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH.

Note In TAUAnCMORm for the PWM output function, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn CH11, CH13, and CH15 (one-phase PWM output function) (m = 11, 13, and 15)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*1	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10, 9, 8	STS2 STS1 STS0	0 0 1	
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	
	TAUAn CMURm	1, 0	TIS1 TIS0	1 1	Detects both edges of TINm as effective edge (High level width)

Note 1. Set the same operating clock as the clock for Master-CH (CH02) for the PWM output function.

Note In TAUAnCMORm for the one-phase PWM output function, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. CH10, CH12, and CH14 can be used for any function that does not use TOUTm output (such as the AD trigger output function).
For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn channels in common

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTOE	15 to 10	TOE15 to TOE10	0 1	Timer operation is stopped. Timer operation is started.
		9 to 4	TOE09 to TOE04	0	Fixed to 0 since TOUT09 to TOUT4 are not used.
		3	TOE03	Don't care	
		2	TOE02	0	Fixed to 0 since TOUT02 is not used.
		1, 0	TOE01 TOE00	Don't care	
	TAUAnTO	15 to 10	TO15 to TO10	0*1	Low level is output to TOUT15 to TOUT10 pins.
		9 to 4	TO09 to TO04	0	Low level is output to TOUT09 to TOUT04 pins.
		3	TO03	Don't care	
		2	TO02	0	Low level is output to TOUT02 pin.
		1, 0	TO01 TO00	Don't care	
	TAUAnTOM	15 to 4	TOM15 to TOM04	1	Synchronous operation mode
		3	TOM03	Don't care	
		2	TOM02	0	Independent operation mode
		1, 0	TOM01 TOM00	Don't care	
	TAUAnTOC	15 to 10	TOC15 to TOC10	1	Synchronous operation mode 2
		9 to 4	TOC09 to TOC04	0	Synchronous operation mode 1
		3	TOC03	Don't care	
		2	TOC02	0	Toggle mode
		1, 0	TOC01 TOC00	Don't care	
	TAUAnTOL	15 to 4	TOL15 to TOL04	0*1	Positive logic output (active high)
		3	TOL03	Don't care	
		2	TOL02	0	Positive logic output (active high)
		1, 0	TOL01 TOL00	Don't care	
	TAUAnTDE	15 to 10	TDE15 to TDE10	1	Dead time control is enabled.*2
		9 to 4	TDE09 to TDE04	0	Dead time control is disabled.
		3	TDE03	Don't care	
		2	TDE02	0	Dead time control is disabled.
		1, 0	TDE01 TDE00	Don't care	
	TAUAnTDM	15 to 10	TDM15 to TDM10	1	Dead time is generated upon detection of TINm input edge of lower odd numbered CH.
		9 to 4	TDM09 to TDM04	0	Invalid since dead time control is disabled.
		3	TDM03	Don't care	
		2	TDM02	0	Invalid since dead time control is disabled.
		1, 0	TDM01 TDM00	Don't care	

Note 1. Change the set value depending on the system used.
Note 2. For dead time control, an even numbered CH and an odd numbered CH are used as a pair to output the positive and negative phase waveforms. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn channels in common (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTDL	15	TDL15	1*[1]	Operates for negative phase of W phase.
		14	TDL14	0*[1]	Operates for positive phase of W phase.
		13	TDL13	1*[1]	Operates for negative phase of V phase.
		12	TDL12	0*[1]	Operates for positive phase of V phase.
		11	TDL11	1*[1]	Operates for negative phase of U phase.
		10	TDL10	0*[1]	Operates for positive phase of U phase.
		9 to 4	TDL09 to TDL04	0	Invalid since dead time control is disabled.
		3	TDL03	Don't care	
		2	TDL02	0	Invalid since dead time control is disabled.
		1, 0	TDL01 TDL00	Don't care	
	TAUAnTRE	15 to 4	TRE15 to TRE04	0	Realtime output is disabled.
		3	TRE03	Don't care	
		2	TER02	0	Realtime output is disabled.
		1, 0	TRE01 TRE00	Don't care	
	TAUAnTRO	15 to 4	TRO15 to TRO04	0	Invalid since realtime output is disabled.
		3	TRO03	Don't care	
		2	TRO02	0	Invalid since realtime output is disabled.
		1, 0	TRO01 TRO00	Don't care	
	TAUAnTRC	15 to 4	TRC15 to TRC04	0	Does not operate as realtime output trigger generating CH.
		3	TRC03	Don't care	
		2	TRC02	0	Does not operate as realtime output trigger generating CH.
		1, 0	TRC01 TRC00	Don't care	
	TAUAnTME	15 to 4	TME15 to TME04	0	Modulation and output of timer output and realtime output is disabled.
		3	TME03	Don't care	
		2	TME02	0	Modulation and output of timer output and realtime output is disabled.
		1, 0	TME01 TME00	Don't care	

Note 1. Change the set value depending on the system used.

Registers related to TAUAn channels in common (n = 0, 1) (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnRDE	15	RDE15	0	Simultaneous rewrite control is disabled.
		14	RDE14	Don't care	
		13	RDE13	0	Simultaneous rewrite control is disabled.
		12	RDE12	Don't care	
		11	RDE11	0	Simultaneous rewrite control is disabled.
		10	RDE10	Don't care	
		9 to 4	RDE09 to RDE04	1	Simultaneous rewrite control is enabled.
		3	RDE03	Don't care	
		2	RDE02	1	Simultaneous rewrite control is enabled.
		1, 0	RDE01 RDE00	Don't care	
	TAUAnRDS	15	RDS15	0	Simultaneous rewrite control through another upper CH is disabled.
		14	RDS14	Don't care	
		13	RDS13	0	Simultaneous rewrite control through another upper CH is disabled.
		12	RDS12	Don't care	
		11	RDS11	0	Simultaneous rewrite control through another upper CH is disabled.
		10	RDS10	Don't care	
		9 to 4	RDS09 to RDS04	0	Simultaneous rewrite control through Master-CH is enabled.
		3	RDS03	Don't care	
		2	RDS02	0	Simultaneous rewrite control through Master-CH is enabled.
		1, 0	RDS01 RDS00	Don't care	
	TAUAnRDM	15	RDM15	0	Invalid since simultaneous rewrite control is disabled.
		14	RDM14	Don't care	
		13	RDM13	0	Invalid since simultaneous rewrite control is disabled.
		12	RDM12	Don't care	
		11	RDM11	0	Invalid since simultaneous rewrite control is disabled.
		10	RDM10	Don't care	
		9 to 4	RDM09 to RDM04	0	Simultaneous load is executed at start of Master-CH.
		3	RDM03	Don't care	
		2	RDM02	0	Simultaneous load is executed at start of Master-CH.
		1, 0	RDM01 RDM00	Don't care	
TAUAn	TAUAnRDC	15	RDC15	0	Invalid since simultaneous rewrite control is disabled.
		14	RDC14	Don't care	
		13	RDC13	0	Invalid since simultaneous rewrite control is disabled.
		12	RDC12	Don't care	
		11	RDC11	0	Invalid since simultaneous rewrite control is disabled.
		10	RDC10	Don't care	
		9 to 4	RDC09 to RDC04	0	Does not operate as simultaneous rewrite trigger generating CH.
		3	RDC03	Don't care	
		2	RDC02	1	Does not operate as simultaneous rewrite trigger generating CH.
		1, 0	RDC01 RDC00	Don't care	

Table 26-15 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG2n2	27, 26	PIC0REG2n227 PIC0REG2n226	1 0	The input selected with PIC0REG2n204 bit is selected.
		23, 22	PIC0REG2n223 PIC0REG2n222	1 0	The input selected with PIC0REG2n203 bit is selected.
		19, 18	PIC0REG2n219 PIC0REG2n218	1 0	The input selected with PIC0REG2n202 bit is selected.
		4	PIC0REG2n204	1	The set and clear signal output according to TAUAnINT08 and TAUAnINT09 is selected.
		3	PIC0REG2n203	1	The set and clear signal output according to TAUAnINT06 and TAUAnINT07 is selected.
		2	PIC0REG2n202	1	The set and clear signal output according to TAUAnINT04 and TAUAnINT05 is selected.

26.4.4 High Accuracy Triangle Wave PWM Output Function with Dead Time

26.4.4.1 Functional Overview

This function enables a control of the variable dead time range (where duty cycle is close to 100% and 0%), thus providing the more accurate triangle wave PWM output than that provided by the triangle wave PWM output function with the dead time.

26.4.4.2 Configuration

Note In this section, signal names are abbreviated as shown below.

INTm → TAUAnINTm

TINm → TAUAnTINm

TOUTm → TAUAnTOUTm

CDRm → TAUAnCDRm

CNTm → TAUAnCNTm

Configuration/ Timer Function	TAUA	TAPA
Timer configuration 1	TAUA0	TAPA0
Timer configuration 2	TAUA1	TAPA1

- Setting functions of TAUAn channels

CH	Function Name	M/S*	CDR Set Value	Description
02	Triangle wave PWM output function with dead time (CH02 is master for CH03 to CH09.)	M	Period	
04		S	Duty (U phase)	
05		S	Dead time (U phase)	
06		S	Duty (V phase)	
07		S	Dead time (V phase)	
08		S	Duty (W phase)	
09		S	Dead time (W phase)	
10	One-shot pulse output function	M	Delay	Generates pulses to be inserted into U phase PWM in variable dead time range.
11		S	Pulse width	
12	One-shot pulse output function	M	Delay	Generates pulses to be inserted into V phase PWM in variable dead time range.
13		S	Pulse width	
14	One-shot pulse output function	M	Delay	Generates pulses to be inserted into W phase PWM in variable dead time range.
15		S	Pulse width	

Note * M = master channel, S = slave channel

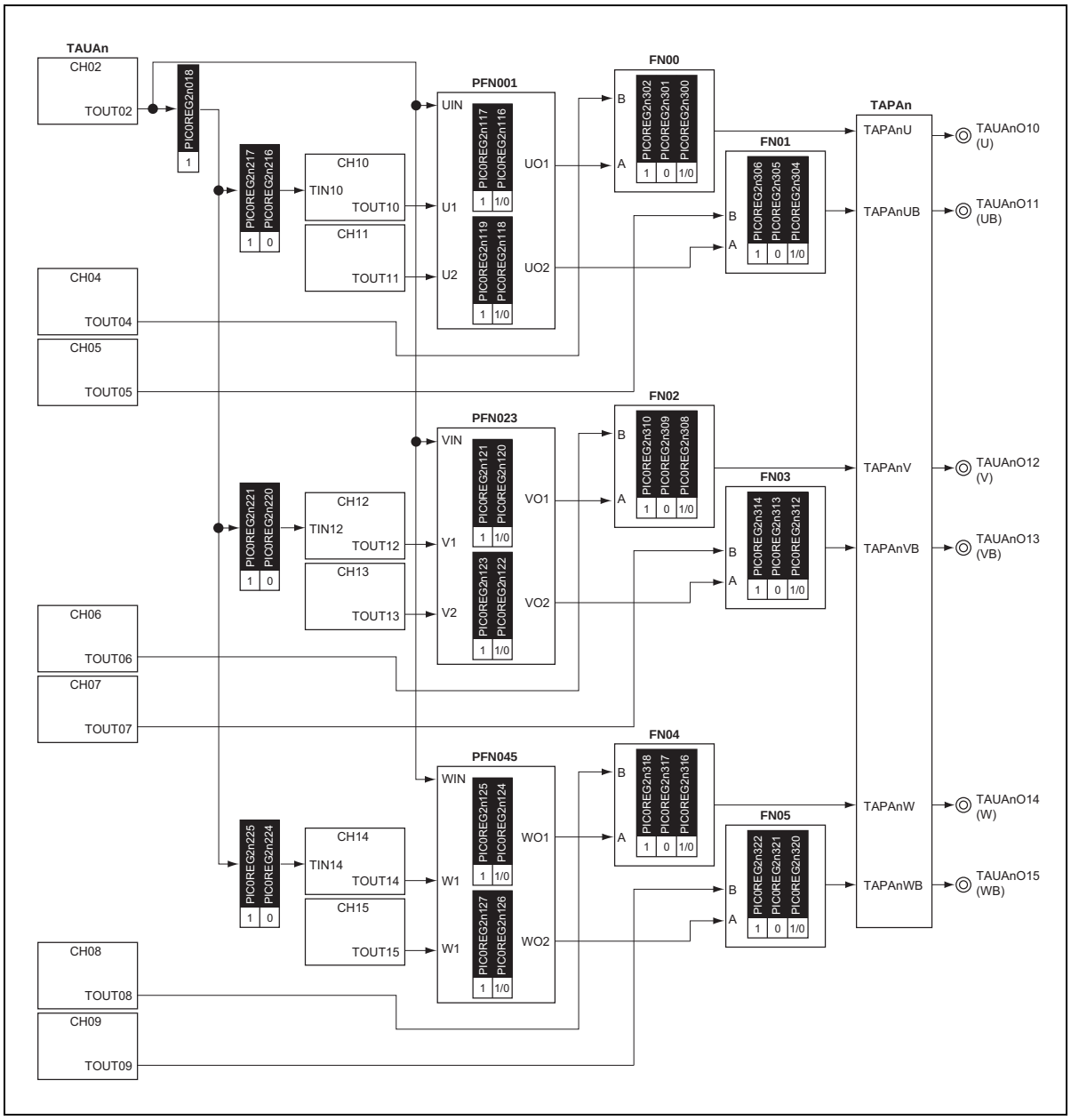


Figure 26-13 Block Diagram

26.4.4.3 Registers

(1) Timer I/O Control Register 2n0 (PIC0REG2n0)

Access This register can be read/written in 32-bit units.

Address FF81 C08C$_H$ (n = 0), FF81 C09C$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	PIC0REG2n018	0	0
R	R	R	R	R	R/W	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 26-16 PIC0REG2n0 Contents

Bit Position	Bit Name	Function
18	PIC0REG2n018	Selects the TIN input signal of TAUAn CH10, TAUAn CH12, and TAUAn CH14. 0: Setting prohibited 1: Selects TOUT of TAUAn CH02.

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

(2) Timer I/O Control Register 2n1 (PIC0REG2n1)

Access This register can be read/written in 32-bit units.

Address FF81 C090$_H$ (n = 0), FF81 C0A0$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	PIC0REG2n127	PIC0REG2n126	PIC0REG2n125	PIC0REG2n124
R	R	R	R	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
PIC0REG2n123	PIC0REG2n122	PIC0REG2n121	PIC0REG2n120	PIC0REG2n119	PIC0REG2n118	PIC0REG2n117	PIC0REG2n116
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 26-17 PIC0REG2n1 Contents (1/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>27
26</td><td>PIC0REG2n127
PIC0REG2n126</td><td>Selects the A input signal of FN05 according to the setting of the TOL bit of TAUAn CH09.
<table>
<tr><th>PIC0REG2n127</th><th>PIC0REG2n126</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH09 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH09 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>25
24</td><td>PIC0REG2n125
PIC0REG2n124</td><td>Selects the A input signal of FN04 according to the setting of the TOL bit for TAUAn CH08.
<table>
<tr><th>PIC0REG2n125</th><th>PIC0REG2n124</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH08 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH08 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Table 26-17 PIC0REG2n1 Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>23
22</td><td>PIC0REG2n123
PIC0REG2n122</td><td>Selects the A input signal of FN03 according to the setting of the TOL bit for TAUAn CH07.
<table>
<tr><th>PIC0REG2n123</th><th>PIC0REG2n122</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH07 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH07 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>21
20</td><td>PIC0REG2n121
PIC0REG2n120</td><td>Selects the A input signal of FN02 according to the setting of the TOL bit for TAUAn CH06.
<table>
<tr><th>PIC0REG2n121</th><th>PIC0REG2n120</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH06 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH06 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>19
18</td><td>PIC0REG2n119
PIC0REG2n118</td><td>Selects the A input signal of FN01 according to the setting of the TOL bit for TAUAn CH05.
<table>
<tr><th>PIC0REG2n119</th><th>PIC0REG2n118</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH05 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH05 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>17
16</td><td>PIC0REG2n117
PIC0REG2n116</td><td>Selects the A input signal of FN00 according to the setting of the TOL bit for TAUAn CH04.
<table>
<tr><th>PIC0REG2n117</th><th>PIC0REG2n116</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Output from the combinational circuit (When TOL of TAUAn CH04 is 0)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted output from the combinational circuit (When TOL of TAUAn CH04 is 1)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

(3) Timer I/O Control Register 2n2 (PIC0REG2n2)

Access This register can be read/written in 32-bit units.

Address FF81 C094$_H$ (n = 0), FF81 C0A4$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	PIC0REG2n225	PIC0REG2n224
R	R	R	R	R	R	R/W	R/W

23	22	21	20	19	18	17	16
0	0	PIC0REG2n221	PIC0REG2n220	0	0	PIC0REG2n217	PIC0REG2n216
R	R	R/W	R/W	R	R	R/W	R/W

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 26-18 PIC0REG2n2 Contents

Bit Position	Bit Name	Function
25 24	PIC0REG2n225 PIC0REG2n224	Selects the TIN input signal of TAUAn CH14.
21 20	PIC0REG2n221 PIC0REG2n220	Selects the TIN input signal of TAUAn CH12.
17 16	PIC0REG2n217 PIC0REG2n216	Selects the TIN input signal of TAUAn CH10.

PIC0REG2n225	PIC0REG2n224	Input Signal
1	0	Signal selected with PIC0REG2n018 bit. (TOUT of TAUAn CH02)
Other than above		Setting prohibited

PIC0REG2n221	PIC0REG2n220	Input Signal
1	0	Signal selected with PIC0REG2n018 bit. (TOUT of TAUAn CH02)
Other than above		Setting prohibited

PIC0REG2n217	PIC0REG2n216	Input Signal
1	0	Signal selected with PIC0REG2n018 bit. (TOUT of TAUAn CH02)
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

(4) Timer I/O Control Register 2n3 (PIC0REG2n3)

The PIC0REG2n3 register selects the logical operation.

Access This register can be read/written in 32-bit units.

Address FF81 C098$_H$ (n = 0), FF81 C0A8$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	PIC0REG2n322	PIC0REG2n321	PIC0REG2n320	0	PIC0REG2n318	PIC0REG2n317	PIC0REG2n316
R	R/W	R/W	R/W	R	R/W	R/W	R/W

15	14	13	12	11	10	9	8
0	PIC0REG2n314	PIC0REG2n313	PIC0REG2n312	0	PIC0REG2n310	PIC0REG2n309	PIC0REG2n308
R	R/W	R/W	R/W	R	R/W	R/W	R/W

7	6	5	4	3	2	1	0
0	PIC0REG2n306	PIC0REG2n305	PIC0REG2n304	0	PIC0REG2n302	PIC0REG2n301	PIC0REG2n300
R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 26-19 PIC0REG2n3 Contents (1/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>22
21
20</td><td>PIC0REG2n322
PIC0REG2n321
PIC0REG2n320</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH09.
<table>
<tr><th>PIC0REG2n322</th><th>PIC0REG2n321</th><th>PIC0REG2n320</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH09 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH09 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>18
17
16</td><td>PIC0REG2n318
PIC0REG2n317
PIC0REG2n316</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH08.
<table>
<tr><th>PIC0REG2n318</th><th>PIC0REG2n317</th><th>PIC0REG2n316</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH08 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH08 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Table 26-19 PIC0REG2n3 Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>14
13
12</td><td>PIC0REG2n314
PIC0REG2n313
PIC0REG2n312</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH07.
<table>
<tr><th>PIC0REG2n314</th><th>PIC0REG2n313</th><th>PIC0REG2n312</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH07 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH07 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>10
9
8</td><td>PIC0REG2n310
PIC0REG2n309
PIC0REG2n308</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH06.
<table>
<tr><th>PIC0REG2n310</th><th>PIC0REG2n309</th><th>PIC0REG2n308</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH06 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH06 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>6
5
4</td><td>PIC0REG2n306
PIC0REG2n305
PIC0REG2n304</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH05.
<table>
<tr><th>PIC0REG2n306</th><th>PIC0REG2n305</th><th>PIC0REG2n304</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH05 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH05 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>2
1
0</td><td>PIC0REG2n302
PIC0REG2n301
PIC0REG2n300</td><td>Selects a logical operation to be performed on input signals A and B according to the setting of the TOL bit of TAUAn CH04.
<table>
<tr><th>PIC0REG2n302</th><th>PIC0REG2n301</th><th>PIC0REG2n300</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(When TOL of TAUAn CH04 is 0)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(When TOL of TAUAn CH04 is 1)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

26.4.4.4 Example of Operation

This function is achieved by combining the following TAUA functions.

- Triangle wave PWM output function with dead time
- One-shot pulse output function

The following functions of the PIC are used to generate the variable dead time range pulses for positive and negative phases to be inserted in the variable dead time range.

- Combinational circuit (PFN001, PNF023, and PNF045)

In addition, the following functions of the PIC are used to synthesize the triangle wave PWM output waveform and the variable dead time range pulses to be inserted in the variable dead time range.

- Logical operation circuit (FN0i) (i = 0 to 5)

The high accuracy triangle wave PWM output function with the dead time is actually implemented by assigning the PWM output obtained by the above functions to the U, V, and W phases, which allows independent dead time setting to the PWM output of each phase. Since the operation of the three phases is identical except for the channel assigned, only the operation of the U phase is explained below.

(1) Triangle Wave PWM Output Function with Dead Time

CH02, CH04, and CH05 are combined to allow the triangle wave PWM with the dead time to be output from TOUT04 and TOUT05.

(2) One-Shot Pulse Output Function

CH10 and CH11 are combined to allow the pulse of the width set in CDR11 to be output as TOUT11 with a delay amount set in CDR10 behind the effective edge of TIN10 (TOUT02) of CH10.

This pulse (TOUT11) is used as the variable dead time range pulse that is used in the range where duty cycle is close to 100% or 0%.

Caution Set each CDR for the one-shot pulse output function so that the following condition is satisfied.

$$CDR05 \geq (CDR10 + CDR11)$$

If the condition above is not satisfied, the output waveform may be influenced. To minimize the influence, satisfy the condition above, and also fix the value of CDR11 to 0000H until the variable dead time range pulse is required.

Set the both edges to be detected of TIN10 (TOUT02) as effective, and set TAUAnTOL11 to 1 (active low).

Set the same operating clock for all the TAUAn channels that are used for the triangle wave PWM output function with the dead time and one-shot pulse output function.

For details on TAUA functions, refer to section 13, Timer Array Unit A (TAUA).

(3) U Phase Combinational Circuit (PFN001)

The U phase combination circuit generates variable dead time range pulses (FN00 A and FN01 A) that are used to insert the dead time pulse generated by the one-shot pulse output function into the triangle wave PWM generated by the triangle wave PWM output function with the dead time.

Table 26-31 shows the relation between the inputs (UIN and U2) and the outputs (UO0 and UO1) of the combinational circuit.

Table 26-20 Inputs and Outputs of Combinational Circuit PFN001 (U Phase and UB Phase)

UO0 output (U phase variable dead time range pulse*)

UIN (TOUT02)	U2 (TOUT11)	UO0	
		PIC0REG2n117 and PIC0REG2n116 = 10B U Phase Output: Active High (TAUAnTOL04 = 0)	PIC0REG2n117 and PIC0REG2n116 = 11B U Phase Output: Active Low (TAUAnTOL04 = 1)
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	0

UO1 output (UB phase variable dead time range pulse*)

UIN (TOUT02)	U2 (TOUT11)	UO1	
		PIC0REG2n119 and PIC0REG2n118 = 10B UB Phase Output: Active High (TAUAnTOL05 = 0)	PIC0REG2n119 and PIC0REG2n118 = 11B UB Phase Output: Active Low (TAUAnTOL05 = 1)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Note * Description of variable dead time range pulse

Variable dead time range pulses are the pseudo pulses that are inserted into the PWM output provided by the triangle wave PWM output function with the dead time of TAUA and that are modeled on the dead time pulses that are generated in the range where duty cycle is close to 100% or 0% in PWM output in HT-PWM mode of TSG2.

Note: Setting of PIC0REG2n1k (k = 16 to 19) depends on the active level of the U or UB phase output of the triangle wave PWM generated by the triangle wave PWM output function with the dead time.

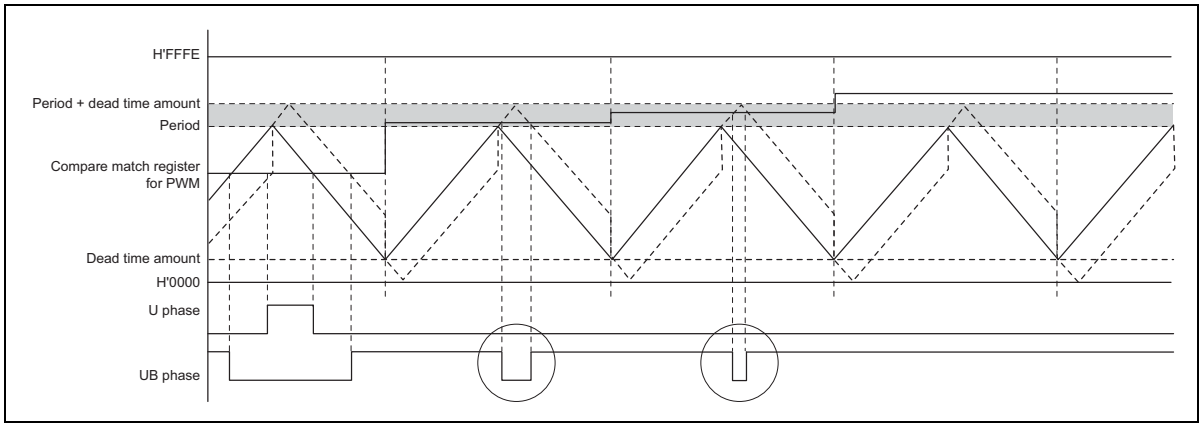


Figure 26-14 Dead Time Output Timing in HT-PTM Mode of TSG2

When the duty cycle of U phase PWM output changes to 0% in HT-PWM mode of TSG2, a compare match that changes the U phase pulse does not occur if a compare match value is set within the shaded range (variable dead time range). However, in the same condition, a compare match that changes the UB phase pulse occurs. Accordingly, only the UB phase pulses are generated.

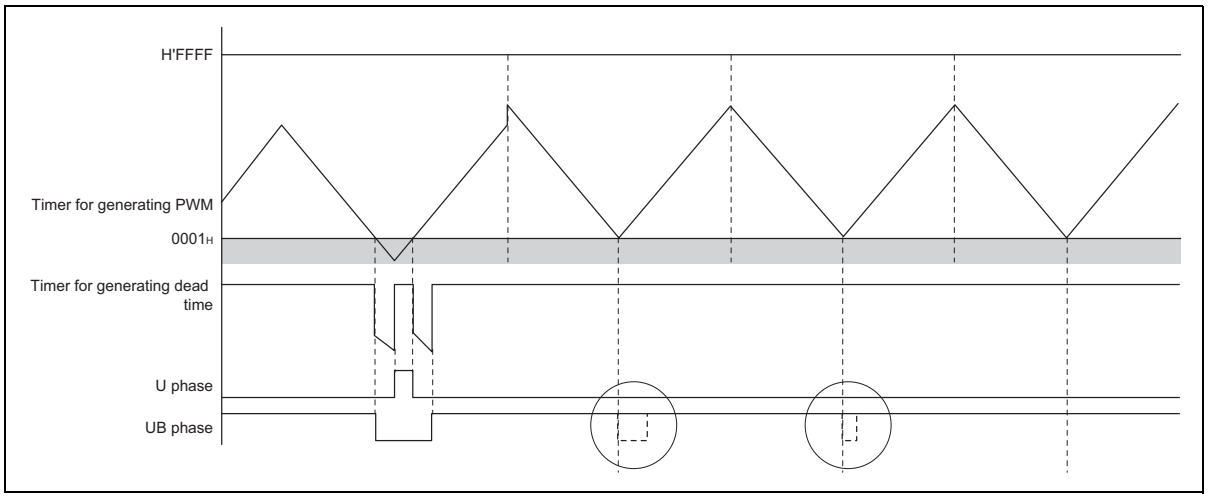


Figure 26-15 Dead Time Output Timing Generated by Triangle Wave PWM Output Function with Dead Time of TAUA

On the other hand, when the duty cycle of U phase is changed to 0% in PWM output by the triangle wave PWM output function with the dead time of TAUA, the UB phase pulses, which can be output by TSG2, cannot be output.

This function uses the timer to generate the pulses to be used in the variable dead time range, and uses the combinational circuit to generate the pulses to be inserted into U or UB phase appropriately according to the conditions in Table 26-20.

(4) Logical Operation Circuit (FN0i) (i = 0, 1)

The logical operation circuit synthesizes the triangle wave PWM output (TOUT04 and TOUT05) from the triangle wave PWM output function with the dead time and the outputs UO0 and UO1 from the combinational circuit PFN001 to generate PWM with variable dead time range pulses inserted.

The synthesizing logic of the logical operation circuit is selected with PIC0REG2n3k (k = 00 to 06).

Set the synthesizing logic for this function as shown in Table 26-21. The pins TAUAnO10 and TAUAnO11 output the signals synthesized according to the specified logic.

Table 26-21 Setting of Logical Operation Circuit (FN0i, i = 0, 1) and TAUAnO10 and TAUAnO11 Outputs

U phase output (TOUT04)

Active Level	PIC0REG2n302 to PIC0REG2n300	TAUAnO10 Pin Output Waveform
Active high (TAUAnTOL04 = 0)	100B	AND operation between FN00 B (TOUT04) and FN00 A (UO0)
Active low (TAUAnTOL04 = 1)	101B	OR operation between FN00 B (TOUT04) and FN00 A (UO0)

UB phase output (TOUT05)

Active Level	PIC0REG2n306 to PIC0REG2n304	TAUAnO11 Pin Output Waveform
Active high (TAUAnTOL05 = 0)	100B	AND operation between FN01 B (TOUT05) and FN01 A (UO1)
Active low (TAUAnTOL05 = 1)	101B	OR operation between FN01 B (TOUT05) and FN01 A (UO1)

With the logical operations above, TAUA can perform the variable dead time control for securing accurate output when the duty cycle is close to 0% or 100%, thus enabling output of triangle wave PWM more accurate than that provided by the triangle wave PWM output function with the dead time of the TAUA function.

The values to be set for V, VB, W, and WB phases are the same as those for U phase, although the channels and the register bits are different as shown in *figure26-13 Block Diagram*.

The PIC provides the connection for inserting the pulses generated by the one-shot pulse output function into the PWM signal generated by the triangle wave PWM output function with the dead time through use of the combinational circuit and the logical operation circuit, which are provided as the PIC functions.

Figures in the following pages show the timing charts of the high accuracy triangle wave PWM output function with the dead time.

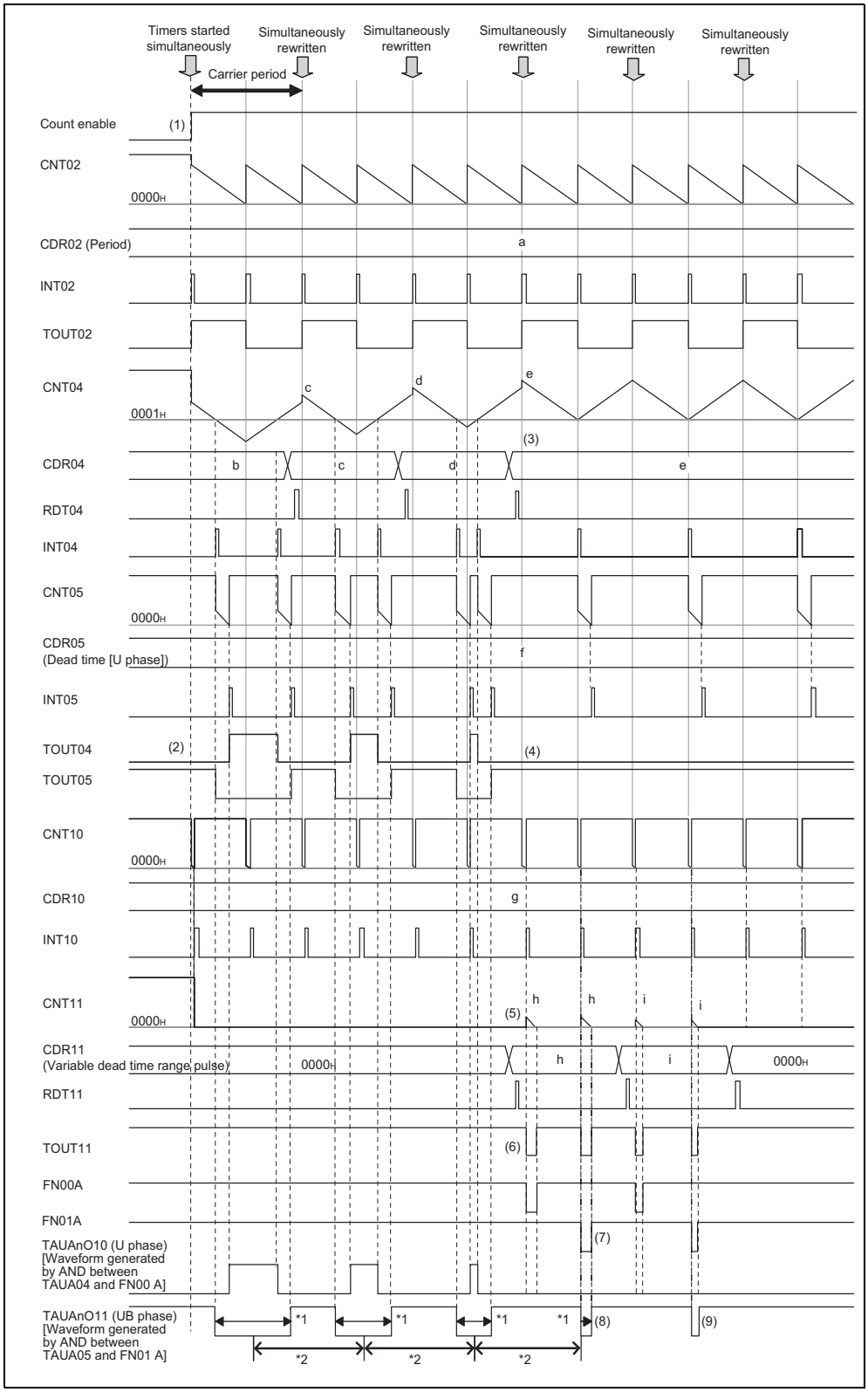


Figure 26-16 High Accuracy Triangle Wave PWM (U Phase Duty Cycle = 0%; UB Phase Duty Cycle = 100%) Output Example with Dead Time (When TOL4 = 0 (Active High) and TOL5 = 0 (Active High))

The following describes the operation example shown in Figure 26-16, where duty cycles of U-phase and UB-phase outputs change to 0% and 100%, respectively, using the timer configuration for U phase PWM output. The output of the triangle wave PWM output function with the dead time is active high.

(1) The triangle wave PWM output function with the dead time using CH02, CH04, and CH05 of TAUAn starts triggered by the start of timer operation.

(2) PWM waveforms with the dead time are output from TOUT04 and TOUT05 by the triangle wave PWM output function with the dead time.

(3) A 0% duty cycle value is set in CDR04 for U phase output.

(4) The setting described in (3) sets the TOUT04 output to the inactive level and the TOUT05 output to the active level. Note that, however, the variable dead time range pulse is not output by this operation.

(5) To generate the variable dead time range pulse, a value for the variable dead time range pulse width is set in CDR11 when a 0% duty cycle value for U phase output is set in step (3).
In this example, the value of CDR11 is fixed to 0000_H until the entry into the variable dead time range to minimize the influence on the output PWM.

(6) The variable dead time range pulse of the width set in CDR11 is output at the edge of TOUT02 after the elapse of the delay time set in CDR10.

(7) The pulse that is output in step (6) is converted to the variable dead time range pulse for U phase (FN00A) or UB phase (FN01A) by the combinational circuit PFN001.

(8) The pulses generated in step (7) are synthesized with the TOUT04 and TOUT05 output pulses in the logical operation circuits (FN00 and FN01) and are output from TAUAnO10 (U phase output) and TAUAnO11 (UB phase output).

(9) After step (8), variable dead time range pulse of a desired width can be inserted by changing the set values of CDR11, which specifies the variable dead time range pulse width.

Note 1. Since the variable dead time range pulses are sawtooth waves, they expand and contract on one side, unlike the triangle wave pulses, which expand and contract on both sides.

Note 2. Since one-side expansion and contraction applies to the variable dead time range pulses, a one-phase PWM output period in the variable dead time range is longer by half the width of an inserted variable dead time range pulse.

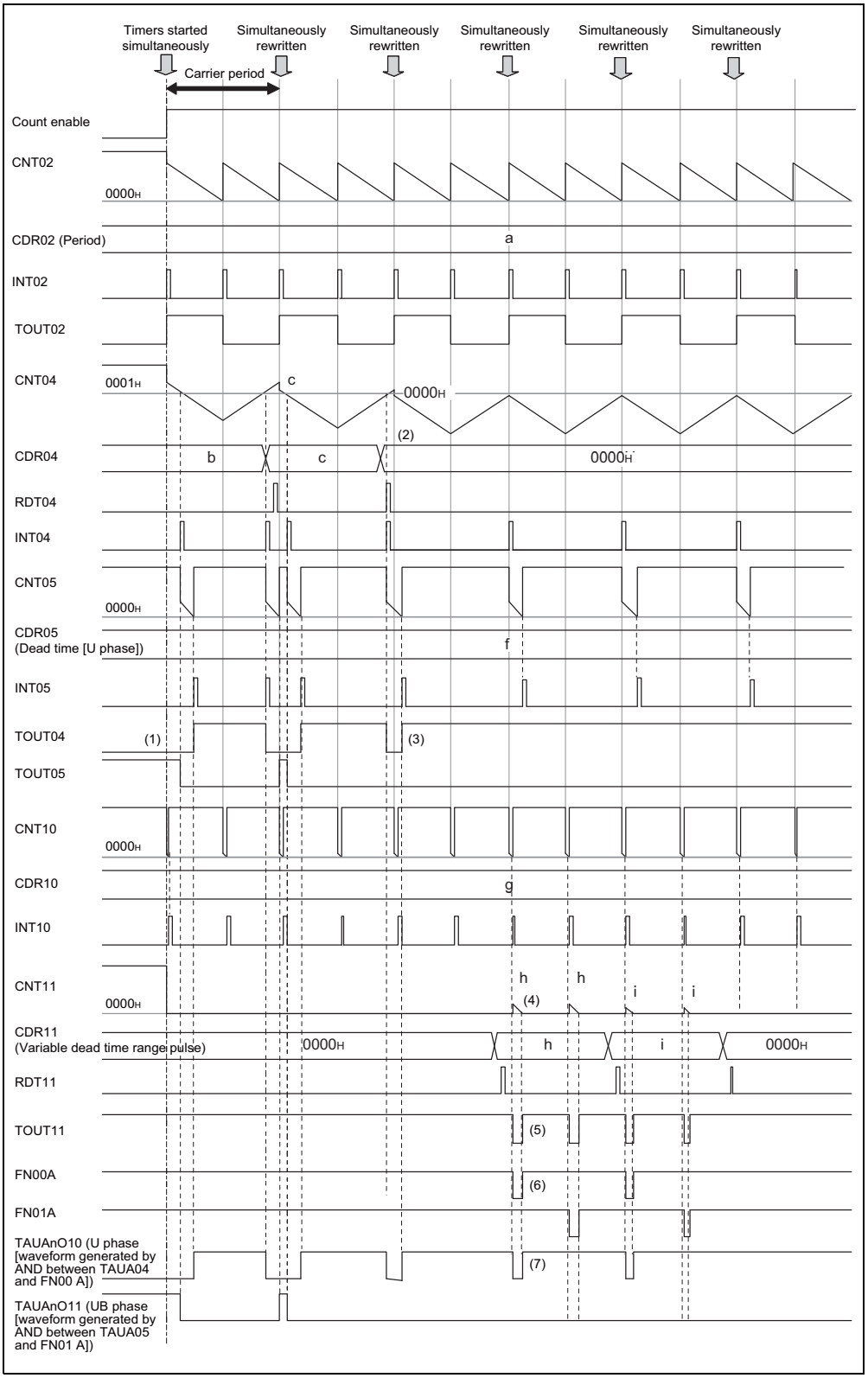


Figure 26-17 High Accuracy Triangle Wave PWM (U Phase Duty Cycle: 100%; UB Phase Duty Cycle: 0%) Output Example with Dead Time (When TOL4 = 0 (Active High) and TOL5 = 0 (Active High))

The following describes the operation example shown in Figure 26-17, where duty cycles of U-phase and UB-phase outputs change to 100% and 0%, respectively, using the timer configuration for U phase PWM output. The output of the triangle wave PWM output function with the dead time is active high.

(1) Timer operation from the timer start to triangle wave PWM output with the dead time is the same as the operation shown in Figure 26-16.

(2) A 100% duty cycle value (0000_H) is set in CDR04 for U phase output.

(3) The setting described in (2) sets the TOUT04 output to the active level and the TOUT05 output to the inactive level. Note that, however, the variable dead time range pulse is not output by this operation.

(4) To generate the variable dead time range pulse, a value for the variable dead time range pulse width is set in CDR11 one period after setting 100% duty cycle for U phase output (step (2)).
In this example, the value of CDR11 is fixed to 0000_H until the entry into the variable dead time range to minimize the influence on the output PWM.

(5) The variable dead time range pulse of the width set in CDR11 is output at the edge of TOUT02 after the elapse of the delay time set in CDR10.

(6) The pulse that is output in step (5) is converted to the variable dead time range pulse for U phase (FN00A) or UB phase (FN01A) by the combinational circuit PFN001.

(7) The pulses generated in step (6) are synthesized with the TOUT04 and TOUT05 output pulses in the logical operation circuits (FN00 and FN01) and are output from TAUAnO10 (U phase output) and TAUAnO11 (UB phase output).

Note As shown in Figure 26-18, if a value is set for the variable dead time range pulse width in CDR11 at the same time as a 100% duty cycle value is set for U phase output in CDR04, the variable dead time range pulse affects the last PWM output from TOUT04 (indicated by (1)) by the amount of time indicated by (2), which is due to the functional specification.
To eliminate this influence, CDR11 must be set one period after setting CDR04.

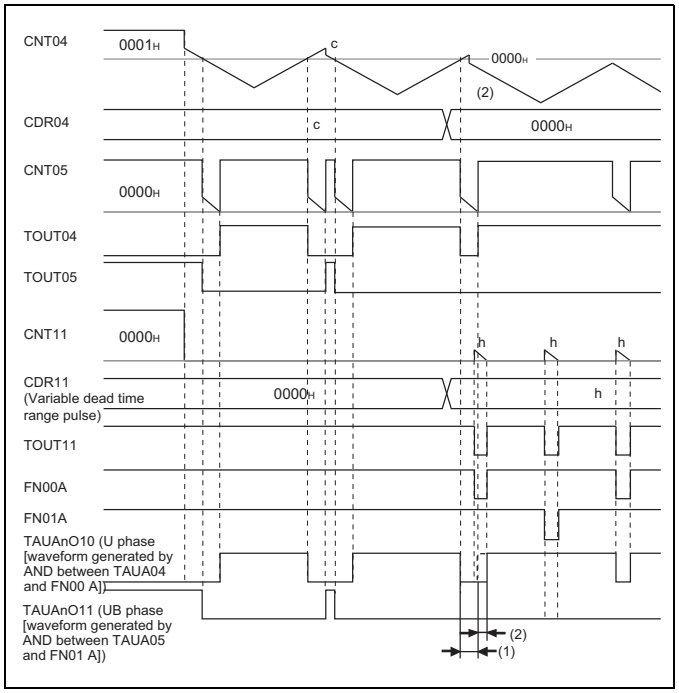


Figure 26-18 An Example of Variable Dead Time Range Pulse Giving Influence on Triangle Wave PWM Output with Dead Time

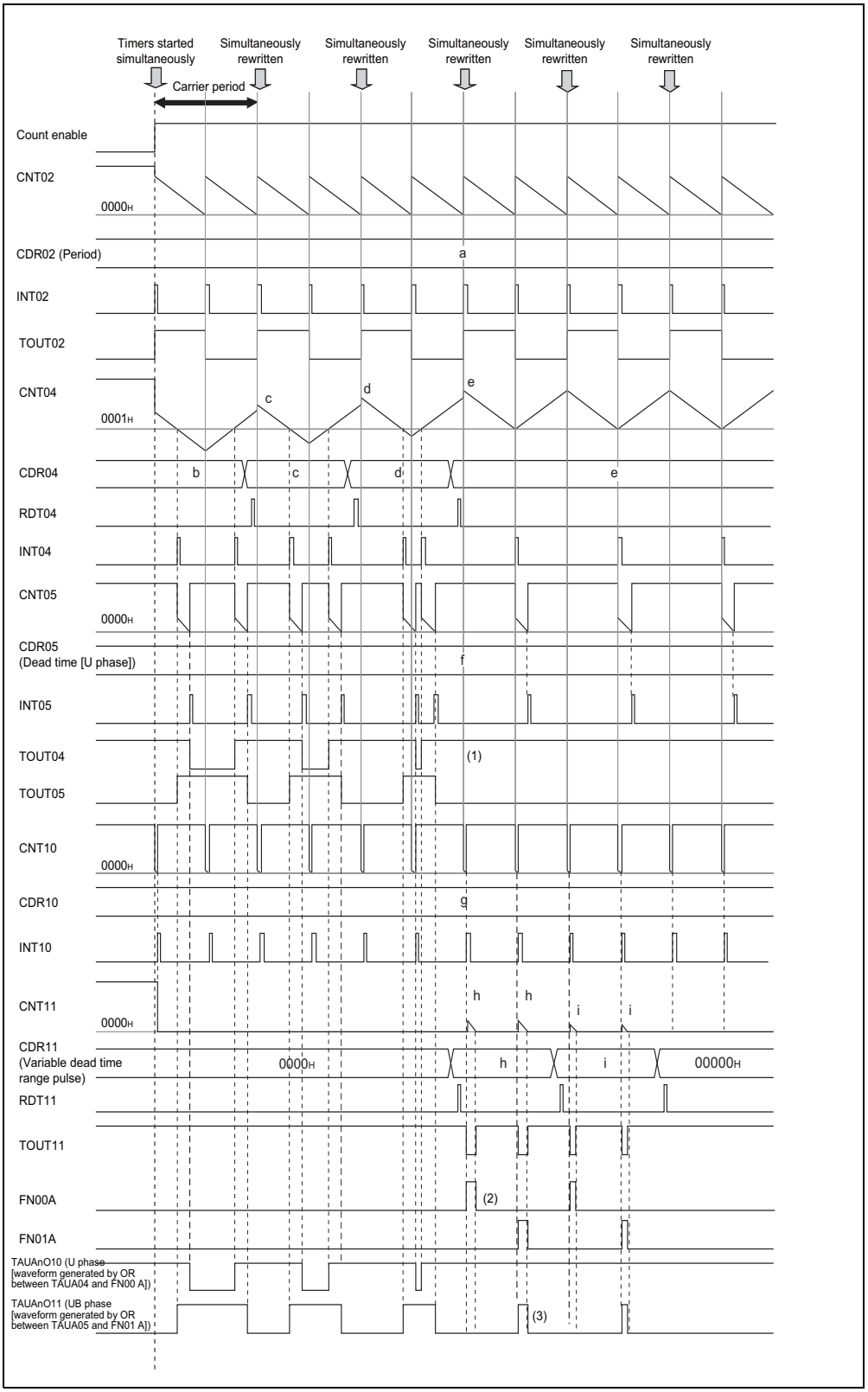


Figure 26-19 High Accuracy Triangle Wave PWM (U Phase Duty Cycle = 100%; UB Phase Duty Cycle = 0%) Output Example with Dead Time (When TOL4 = 1 (Active Low) and TOL5 = 1 (Active Low))

The following describes the operation example shown in Figure 26-19, where duty cycles of U-phase and UB-phase outputs change to 100% and 0%, respectively, using the timer configuration for U phase PWM output. The output of the triangle wave PWM output function with the dead time is active low.

(1) Timer operation from the timer start to triangle wave PWM output with the dead time is the same as the operation shown in *figure26-16 High Accuracy Triangle Wave PWM (U Phase Duty Cycle = 0%; UB Phase Duty Cycle = 100%) Output Example with Dead Time (When TOL4 = 0 (Active High) and TOL5 = 0 (Active High))*. Note that, however, the PWM outputs from TOUT04 and TOUT05 are active low.

(2) Accordingly, the combinational circuit setting (PIC0REG2n116, PIC0REG2n117 and PIC0REG2n118, PIC0REG2n119) should be made so that the active low output is selected (the same level as the PWM outputs). With this setting, variable dead time range pulses for active low are output as FN00A (for U phase) and FN01A (for UB phase).

(3) Similarly, the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304) should be made so that active low output is selected (the same level as the PWM outputs). The pulses generated in step (2) are synthesized with the TOUT04 and TOUT05 output waveforms, and are output as PWM for active low from TAUAnO10 (U phase output) and TAUAnO11 (UB phase output).

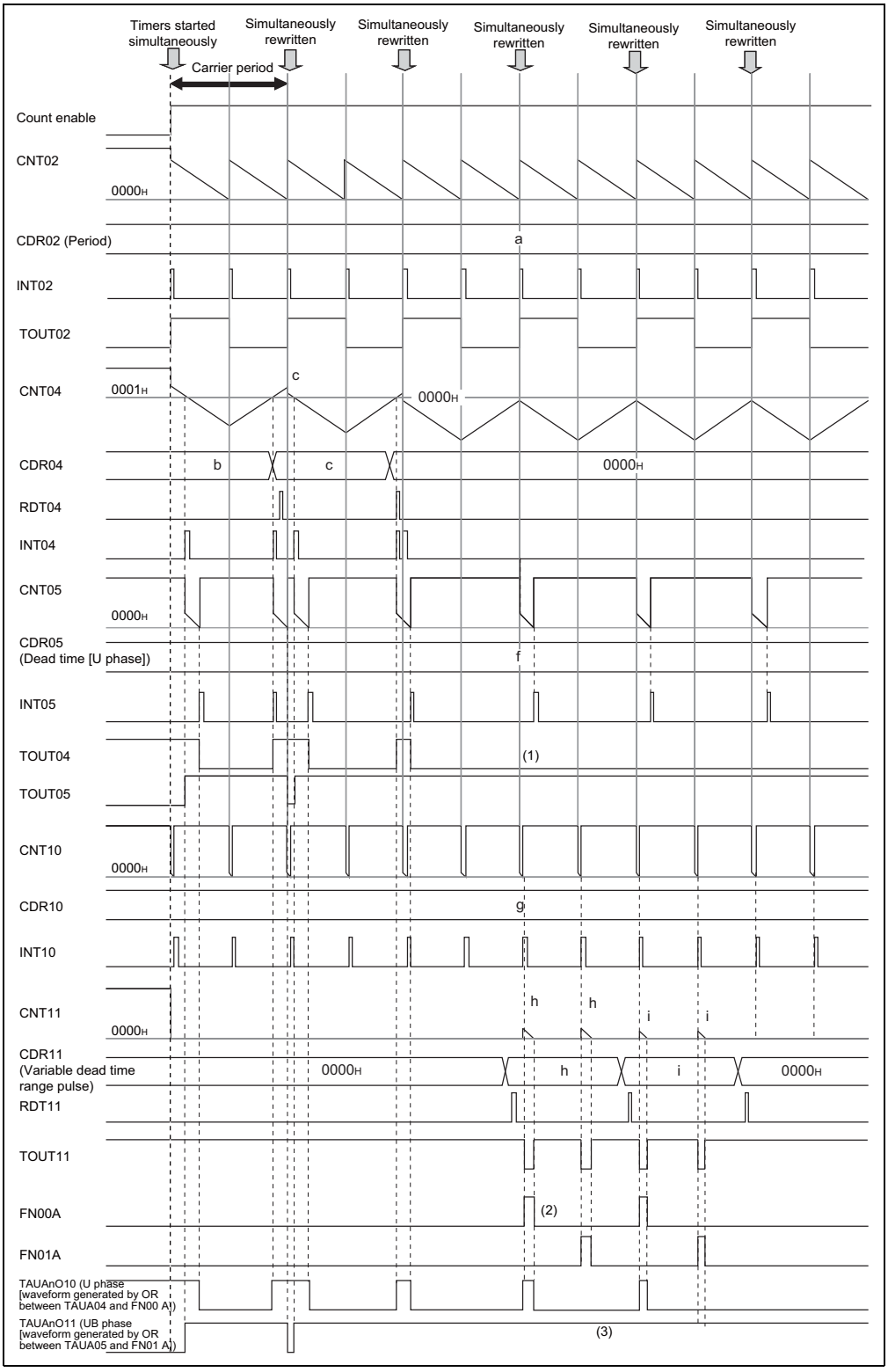


Figure 26-20 High Accuracy Triangle Wave PWM (U Phase Duty Cycle = 0%; UB Phase Duty Cycle = 100%) Output Example with Dead Time (When TOL4 = 0 (Active Low) and TOL5 = 0 (Active Low))

The following describes the operation example shown in Figure 26-20, where duty cycles of U-phase and UB-phase outputs change to 0% and 100%, respectively, using the timer configuration for U phase PWM output. The output of the triangle wave PWM output function with the dead time is active low.

(1) Timer operation from the timer start to triangle wave PWM output with the dead time is the same as the operation shown in *figure26-17 High Accuracy Triangle Wave PWM (U Phase Duty Cycle: 100%; UB Phase Duty Cycle: 0%) Output Example with Dead Time (When TOL4 = 0 (Active High) and TOL5 = 0 (Active High))*. Note that, however, the PWM outputs are active low.

(2) Accordingly, the combinational circuit setting (PIC0REG2n116, PIC0REG2n117 and PIC0REG2n118, PIC0REG2n119) should be made so that the active low output is selected (the same level as the PWM outputs). With this setting, variable dead time range pulses for active low are output as FN00A (for U phase) and FN01A (for UB phase).

(3) Similarly, the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304) should be made so that active low output is selected (the same level as the PWM outputs). The pulses generated in step (2) are synthesized with the TOUT04 and TOUT05 output waveforms, and are output as PWM for active low from TAUAnO10 (U phase output) and TAUAnO11 (UB phase output).

Caution If a value is set for the variable dead time range pulse width in CDR11 at the same time as a 100% duty cycle value is set for U phase output in CDR04, the last PWM output from TOUT04 is affected due to the functional specification. To eliminate this influence, CDR11 must be set one period after setting CDR04.
For details, refer to *figure26-18 An Example of Variable Dead Time Range Pulse Giving Influence on Triangle Wave PWM Output with Dead Time*.

26.4.4.5 Operation Flow

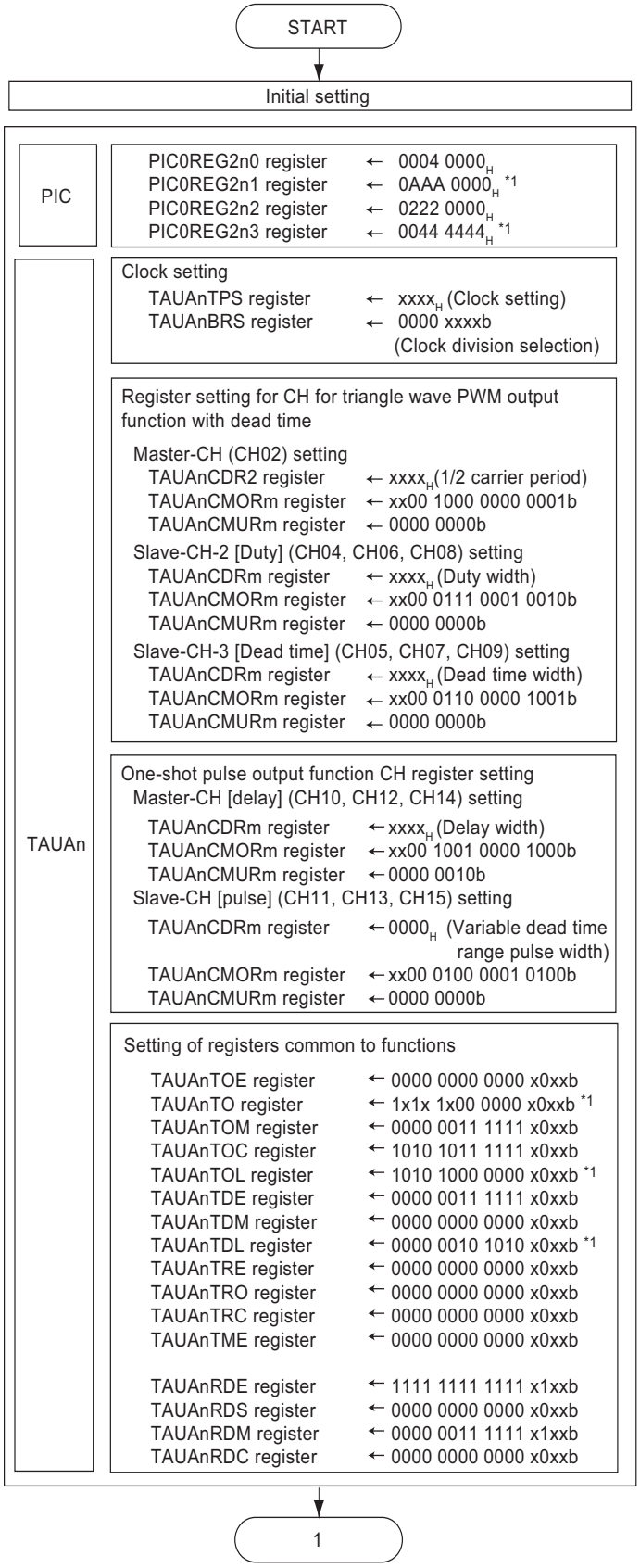


Figure 26-21 Setting Flow (Active High)

Note 1. Change the set value depending on the active level of the PWM output.

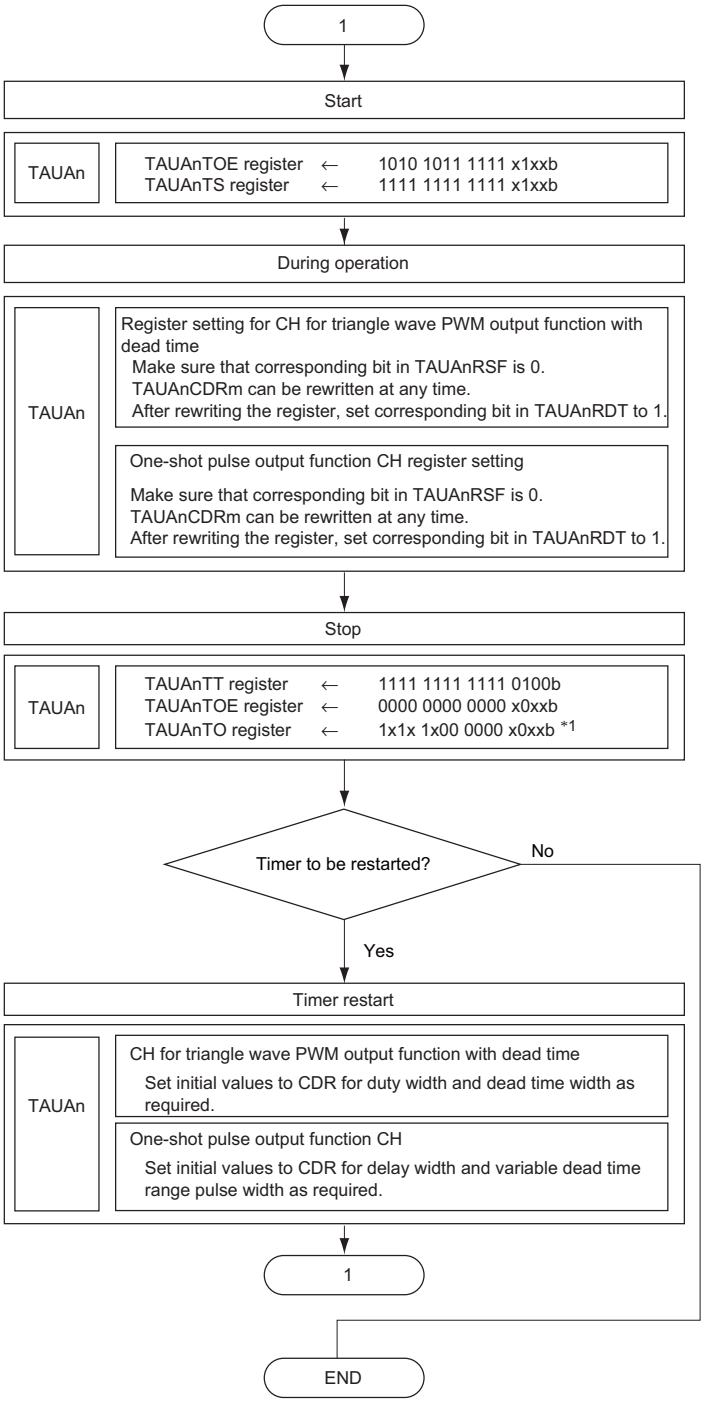


Figure 26-22 Setting Flow (Active High) (cont.)

Note 1. Change the set value depending on the active level of the PWM output.

26.4.4.6 Register Setting Examples for Various Functions

Register setting

Table 26-22 TAUAn Setting (Active High)

Registers related to TAUAn CH02 (Master-CH*1 for triangle wave PWM output function with dead time)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMOR2	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 0	
		7, 6	COS1 COS0	0 0	
		5		0	
		4	MD4	0	
		3	MD3	0	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	1: Outputs INTn and toggles TOUTn at the start of operation.
	TAUAn CMUR2	1, 0	TIS1 TIS0	0 0	Fixed

Note 1. Master-CH and Slave-CH are the names defined with the TAUA triangle wave PWM output function with the dead time. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH.

Note: In TAUAnCMORm of Master-CH for the triangle wave PWM output function with the dead time, only the CKS (operating clock select) bits and the MD0 bit can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Set the MD0 bit to 1 to perform this PIC function.

Registers related to TAUAn CH04, CH06, and CH08 (triangle wave PWM output function with dead time Slave-CH2*1) (m = 4, 6, and 8)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 1 1	
		7, 6	COS1 COS0	0 0	
		5		0	
		4	MD4	1	
		3	MD3	0	
		2	MD2	0	
		1	MD1	1	
		0	MD0	0	
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Registers related to TAUAn CH05, CH07, and CH09 (triangle wave PWM output function with dead time Slave-CH3*1) (m = 5, 7, and 9)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 1 0	
		7, 6	COS1 COS0	0 0	
		5		0	
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Note 1. Master-CH and Slave-CH are the names defined with the TAUA triangle wave PWM output function with the dead time. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH.

Note: In TAUAnCMORm for Slave-CH2 and Slave-CH3 for the triangle wave PWM output function with the dead time, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn CH10, CH12, and CH14 (One-shot pulse output function Master-CH[*1]) (m = 10, 12, and 14)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care[*2]	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 1	
		7, 6	COS1 COS0	0 0	
		5		0	
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	0	A start trigger is disabled during counting.
	TAUAn CMURm	1, 0	TIS1 TIS0	1 0	Detects both edges as effective edge.

Registers related to TAUAn CH11, CH13, and CH15 (One-shot pulse output function Slave-CH*[1]) (m = 11, 13, and 15)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care[*2]	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 0 0	
		7, 6	COS1 COS0	0 0	
		5		0	
		4	MD4	1	
		3	MD3	0	
		2	MD2	1	
		1	MD1	0	
		0	MD0	0	A start trigger is disabled during counting.
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Note 1. Master-CH and Slave-CH are the names defined with the TAUA one-shot pulse output function. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH. Set the same operating clock as the clock for Master-CH (CH02) for the triangle wave PWM output function with the dead time.

Note: In TAUAnCMORm for the one-shot pulse output function, only the CKS (operating clock select) bits and the MD0 bit can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Set the MD0 bit to 0 for this function.

Registers related to TAUAn channels in common

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTOE	15	TOE15	0 1	Timer operation is stopped. Timer operation is started.
		14	TOE14	0	
		13	TOE13	0 1	Timer operation is stopped. Timer operation is started.
		12	TOE12	0	
		11	TOE11	0 1	Timer operation is stopped. Timer operation is started.
		10	TOE10	0	
		9 to 4	TOE09 to TOE04	0 1	Timer operation is stopped. Timer operation is started.
		3	TOE03	Don't care	
		2	TOE02	0 1	Timer operation is stopped. Timer operation is started.
		1, 0	TOE01 TOE00	Don't care	
	TAUAnTO	15	TO15	1*1	High level is output from the TOUT15 pin.
		14	TO14	Don't care	
		13	TO13	1*1	High level is output from the TOUT13 pin.
		12	TO12	Don't care	
		11	TO11	1*1	High level is output from the TOUT11 pin.
		10	TO10	Don't care	
		9 to 4	TO09 to TO04	0*1	Low level is output from the TOUT09 to TOUT04 pins.
		3	TO03	Don't care	
		2	TO02	0	Low level is output from the TOUT02 pin.
		1, 0	TO01 TO00	Don't care	
	TAUAnTOM	15 to 10	TOM15 to TOM10	0	Independent operation mode
		9 to 4	TOM09 to TOM04	1	Synchronous operation mode
		3	TOM03	Don't care	
		2	TOM02	0	Independent operation mode
		1, 0	TOM01 TOM00	Don't care	
	TAUAnTOC	15	TOC15	1	Set/reset mode
		14	TOC14	0	
		13	TOC13	1	Set/reset mode
		12	TOC12	0	
		11	TOC11	1	Set/reset mode
		10	TOC10	0	
		9 to 4	TOC09 to TOC04	1	Synchronous operation mode 2
		3	TOC03	Don't care	
		2	TOC02	0	Toggle mode
		1, 0	TOC01 TOC00	Don't care	

Note 1. Change the set value depending on the system used.

Registers related to TAUAn channels in common (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTOL	15	TOL15	1*1	Negative logic output (active low)
		14	TOL14	Don't care	
		13	TOL13	1*1	Negative logic output (active low)
		12	TOL14	Don't care	
		11	TOL11	1*1	Negative logic output (active low)
		10	TOL10	Don't care	
		9 to 4	TOL09 to TOL04	0*1	Positive logic output (active high)
		3	TOL03	Don't care	
		2	TOL02	0	Positive logic output (active high)
		1, 0	TOL01 TOL00	Don't care	
	TAUAnTDE	15 to 10	TDE15 to TDE10	0	Dead time control is disabled.
		9 to 4	TDE09 to TDE04	1	Dead time control is enabled.*2
		3	TDE03	Don't care	
		2	TDE02	0	Dead time control is disabled.
		1, 0	TDE01 TDE00	Don't care	
	TAUAnTDM	15 to 9	TDM15 to TDM09	0	
		3	TDM03	Don't care	
		2	TDM02	0	Invalid since dead time control is disabled.
		1, 0	TDM01 TDM00	Don't care	
	TAUAnTDL	15 to 10	TDL15 to TDL10	0	Invalid since dead time control is disabled.
		9	TDL09	1*1	Operates for negative phase of W phase.
		8	TDL08	0*1	Operates for positive phase of W phase.
		7	TDL07	1*1	Operates for negative phase of V phase.
		6	TDL06	0*1	Operates for positive phase of V phase.
		5	TDL05	1*1	Operates for negative phase of U phase.
		4	TDL04	0*1	Operates for positive phase of U phase.
		3	TDL03	Don't care	
		2	TDL02	0	Invalid since dead time control is disabled.
		1, 0	TDL01 TDL00	Don't care	
	TAUAnTRE	15 to 4	TRE15 to TRE04	0	Realtime output is disabled.
		3	TRE03	Don't care	
		2	TER02	0	Realtime output is disabled.
		1, 0	TRE01 TRE00	Don't care	
	TAUAnTRO	15 to 4	TRO15 to TRO04	0	Invalid since realtime output is disabled.
		3	TRO03	Don't care	
		2	TRO02	0	Invalid since realtime output is disabled.
		1, 0	TRO01 TRO00	Don't care	

Note 1. Change the set value depending on the system used.
Note 2. For dead time control, an even numbered CH and an odd numbered CH are used as a pair to output the positive and negative phase waveforms. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn channels in common (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTRC	15 to 4	TRC15 to TRC04	0	Does not operate as realtime output trigger generating CH.
		3	TRC03	Don't care	
		2	TRC02	0	Does not operate as realtime output trigger generating CH.
		1, 0	TRC01 TRC00	Don't care	
	TAUAnTME	15 to 4	TME15 to TME04	0	Modulation and output of timer output and realtime output is disabled.
		3	TME03	Don't care	
		2	TME02	0	Modulation and output of timer output and realtime output is disabled.
		1, 0	TME01 TME00	Don't care	
	TAUAnRDE	15 to 4	RDE15 to RDE04	1	Simultaneous rewrite control is enabled.
		3	RDE03	Don't care	
		2	RDE02	1	Simultaneous rewrite control is enabled.
		1, 0	RDE01 RDE00	Don't care	
	TAUAnRDS	15 to 4	RDS15 to RDS04	0	Simultaneous rewrite control through another upper CH is disabled.
		3	RDS03	Don't care	
		2	RDS02	0	Simultaneous rewrite control through another upper CH is disabled.
		1, 0	RDS01 RDS00	Don't care	
	TAUAnRDM	15 to 10	RDM15 to RDM10	0	Simultaneous rewrite control is performed at the start of Master-CH.
		9 to 4	RDM09 to RDM04	1	Simultaneous rewrite control is performed at the start of Master-CH that operates at the triangle wave period.
		3	RDM03	Don't care	
		2	RDM02	1	Simultaneous rewrite control is performed at the start of Master-CH that operates at the triangle wave period.
		1, 0	RDM01 RDM00	Don't care	
	TAUAnRDC	15 to 4	RDC15 to RDC04	0	Does not operate as simultaneous rewrite trigger generating CH.
		3	RDC03	Don't care	
		2	RDC02	0	Does not operate as simultaneous rewrite trigger generating CH.
		1, 0	RDC01 RDC00	Don't care	

Table 26-23 PIC Setting (Active High)

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG2n0	18	PIC0REG2n018	1	Selects TOUT of TAUAn CH02.
	PIC0REG2n1	27, 26	PIC0REG2n127 PIC0REG2n126	1 0	Negative W phase active high output from the combinational circuit
		25, 24	PIC0REG2n125 PIC0REG2n124	1 0	Positive W phase active high output from the combinational circuit
		23, 22	PIC0REG2n123 PIC0REG2n122	1 0	Negative V phase active high output from the combinational circuit
		21, 20	PIC0REG2n121 PIC0REG2n120	1 0	Positive V phase active high output from the combinational circuit
		19, 18	PIC0REG2n119 PIC0REG2n118	1 0	Negative U phase active high output from the combinational circuit
		17, 16	PIC0REG2n117 PIC0REG2n116	1 0	Positive U phase active high output from the combinational circuit
	PIC0REG2n2	25, 24	PIC0REG2n225 PIC0REG2n224	1 0	The input selected with PIC0REG2n018 bit is selected.
		21, 20	PIC0REG2n221 PIC0REG2n220	1 0	The input selected with PIC0REG2n018 bit is selected.
		17, 16	PIC0REG2n217 PIC0REG2n216	1 0	The input selected with PIC0REG2n018 bit is selected.
	PIC0REG2n3	22 to 20	PIC0REG2n322 PIC0REG2n321 PIC0REG2n320	1 0 0	Negative W phase active high output from the logical operation circuit
		18 to 16	PIC0REG2n318 PIC0REG2n317 PIC0REG2n316	1 0 0	Positive W phase active high output from the logical operation circuit
		14 to 12	PIC0REG2n314 PIC0REG2n313 PIC0REG2n312	1 0 0	Negative V phase active high output from the logical operation circuit
		10 to 8	PIC0REG2n310 PIC0REG2n309 PIC0REG2n308	1 0 0	Positive V phase active high output from the logical operation circuit
		6 to 4	PIC0REG2n306 PIC0REG2n305 PIC0REG2n304	1 0 0	Negative U phase active high output from the logical operation circuit
		2 to 0	PIC0REG2n302 PIC0REG2n301 PIC0REG2n300	1 0 0	Positive U phase active high output from the logical operation circuit

26.4.5 Delay Pulse Output Function with Dead Time

26.4.5.1 Functional Overview

This function allows to generate three-phase PWM output with the dead time, that is, delayed as specified from the period timing.

With this function, PWM output can be reset in the next period unlike with the function described in section 26.4.3, Three-Phase PWM Output Function with Dead Time.

26.4.5.2 Configuration

Note: In this section, signal names are abbreviated as shown below.

INTm → TAUAnINTm

TINm → TAUAnTINm

TOUTm → TAUAnTOUTm

CDRm → TAUAnCDRm

CNTm → TAUAnCNTm

Configuration/ Timer Function	TAUA	TAPA
Timer configuration 1	TAUA0	TAPA0
Timer configuration 2	TAUA1	TAPA1

- Setting functions of TAUAn channels
  Since the CDRn value of CH03 does not influence TOUT10 to TOUT15, it is possible to use INTn of CH03 for other uses such as AD conversion trigger generation.

CH	Function Name	M/S*	CDR Set Value	Description
02	Delay pulse output function (CH02 is master for CH03 to CH09)	M	Period	
03		S		Reserved
04		S	Delay (U phase)	
05		S	Pulse width (U phase)	
06		S	Delay (V phase)	
07		S	Pulse width (V phase)	
08		S	Delay (W phase)	
09		S	Pulse width (W phase)	
10	Any function not using TOUTn	S		TOUT: U phase output
11	One-phase PWM output function	S	Dead time (U phase)	TOUT: UB phase output
12	Any function not using TOUTn	S		TOUT: V phase output
13	One-phase PWM output function	S	Dead time (V phase)	TOUT: VB phase output
14	Any function not using TOUTn	S		TOUT: W phase output
15	One-phase PWM output function	S	Dead time (W phase)	TOUT: WB phase output

Note * M = master channel, S = slave channel

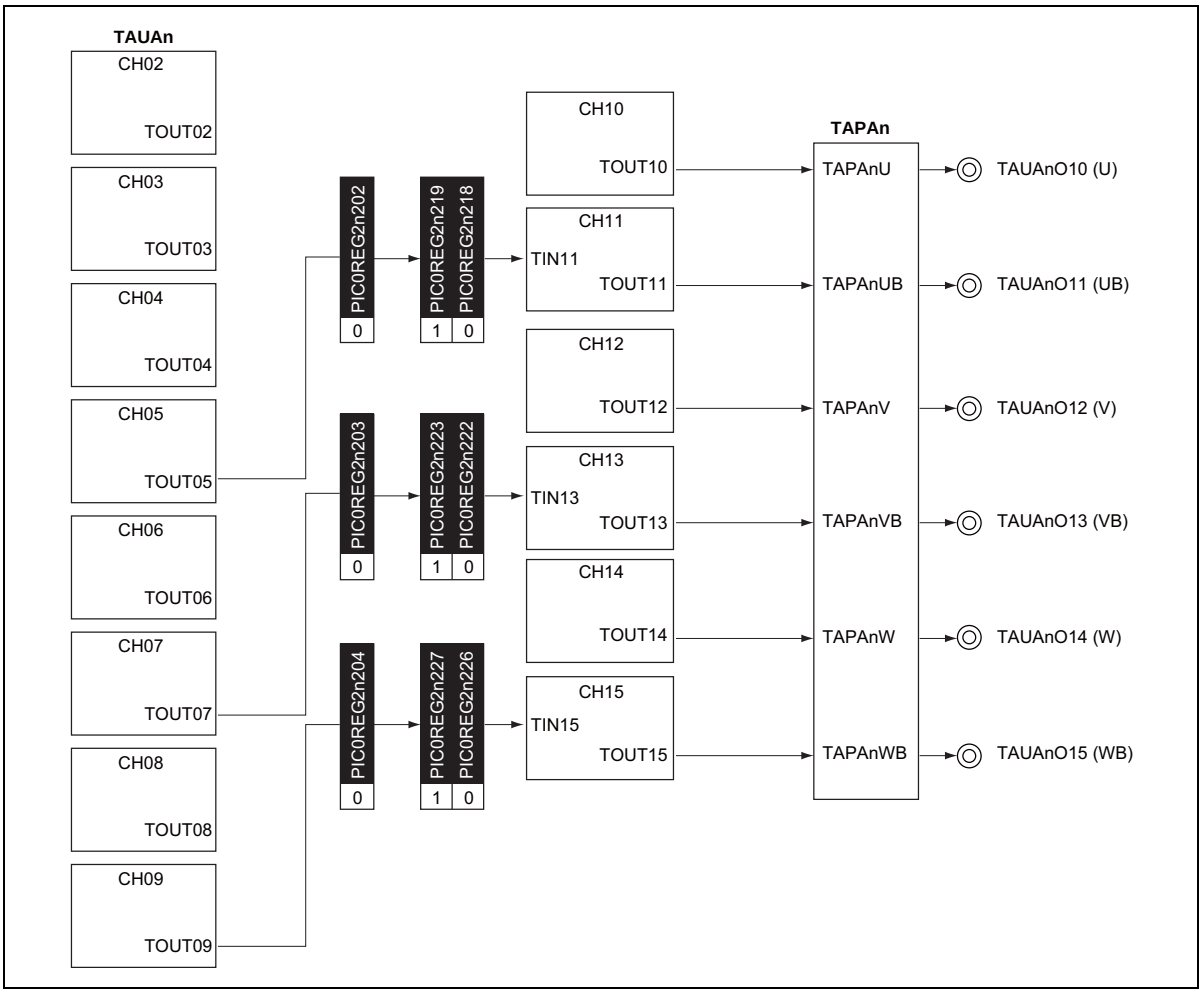


Figure 26-23 Block Diagram

26.4.5.3 Registers

(1) Timer I/O Control Register 2n2 (PIC0REG2n2)

Access This register can be read/written in 32-bit units.

Address FF81 C094$_H$ (n = 0), FF81 C0A4$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	PIC0REG2n227	PIC0REG2n226	0	0
R	R	R	R	R/W	R/W	R	R

23	22	21	20	19	18	17	16
PIC0REG2n223	PIC0REG2n222	0	0	PIC0REG2n219	PIC0REG2n218	0	0
R/W	R/W	R	R	R/W	R/W	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	0	0
R	R	R	R/W	R/W	R/W	R	R

Table 26-24 PIC0REG2n2 Contents (1/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>27
26</td><td>PIC0REG2n227
PIC0REG2n226</td><td>Selects the TIN input signal of TAUAn CH15.
<table>
<tr><th>PIC0REG2n227</th><th>PIC0REG2n226</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Signal selected with PIC0REG2n204 bit (TAUAn CH09 from TOUT)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>23
22</td><td>PIC0REG2n223
PIC0REG2n222</td><td>Selects the TIN input signal of TAUAn CH13.
<table>
<tr><th>PIC0REG2n223</th><th>PIC0REG2n222</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Signal selected with PIC0REG2n203 bit (TAUAn CH07 from TOUT)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>19
18</td><td>PIC0REG2n219
PIC0REG2n218</td><td>Selects the TIN input signal of TAUAn CH11
<table>
<tr><th>PIC0REG2n219</th><th>PIC0REG2n218</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Signal selected with PIC0REG2n202 bit (TAUAn CH05 from TOUT)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>4</td><td>PIC0REG2n204</td><td>Selects the signal to be supplied to TIN of TAUAn CH15.
0: TOUT from TAUAn CH09
1: Setting prohibited</td></tr>
</table>

Table 26-24 PIC0REG2n2 Contents (2/2)

Bit Position	Bit Name	Function
3	PIC0REG2n203	Selects the signal to be supplied to TIN of TAUAn CH13. 0: TOUT from TAUAn CH07 1: Setting prohibited
2	PIC0REG2n202	Selects the signal to be supplied to TIN of TAUAn CH11. 0: TOUT from TAUAn CH05 1: Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

(2) Timer I/O Control Register 2n3 (PIC0REG2n3)

Access This register can be read/written in 32-bit units.

Address FF81 C098$_H$ (n = 0), FF81 C0A8$_H$ (n = 1)

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

26.4.5.4 Example of Operation

The delay pulse output function with the dead time is achieved by combining the following TAUA functions:

- Delay pulse output function
- One-phase PWM output function

A PWM that is delayed from the period timing is generated using the delay pulse output function. Then the one-phase PWM is output with the dead time inserted using the one-phase PWM output function.

The delay pulse output function with the dead time is actually implemented by assigning the PWM output obtained by the above functions to the U, V, and W phases, which allows independent dead time setting to the PWM output of each phase. Since the operation of the three phases is identical except for the channel assigned, only the operation of the U phase is explained below.

(1) Delay Pulse Output Function

CH02, CH04, and CH05 are combined to allow the base PWM for one-phase PWM output to be output from TOUT05, which has a delay amount generated using CH04 with respect to the period set using CH02.

CH03 is a reserved timer for function implementation; do not use it for any other function.

Caution Do not set the delay amount larger than the period.

(2) One-Phase PWM Output Function

CH10 and CH11 are combined to allow the one-phase PWM to be output from TOUT10 and TOUT11.

By setting the dead time to CDR11, the one-phase PWM with the dead time inserted in the TIN11 input is output.

Similarly, the one-phase PWM with the dead time can be output for the V and W phases using CH12 and CH13 (V phase), and CH14 and CH15 (W phase).

Caution Set the same operating clock for all the TAUAn channels that are used for the delay pulse output function and one-phase PWM output function.

For details on TAUA functions, refer to section 13, Timer Array Unit A (TAUA).

The following describes the differences between the delay pulse output function with the dead time and three-phase PWM output function with the dead time.

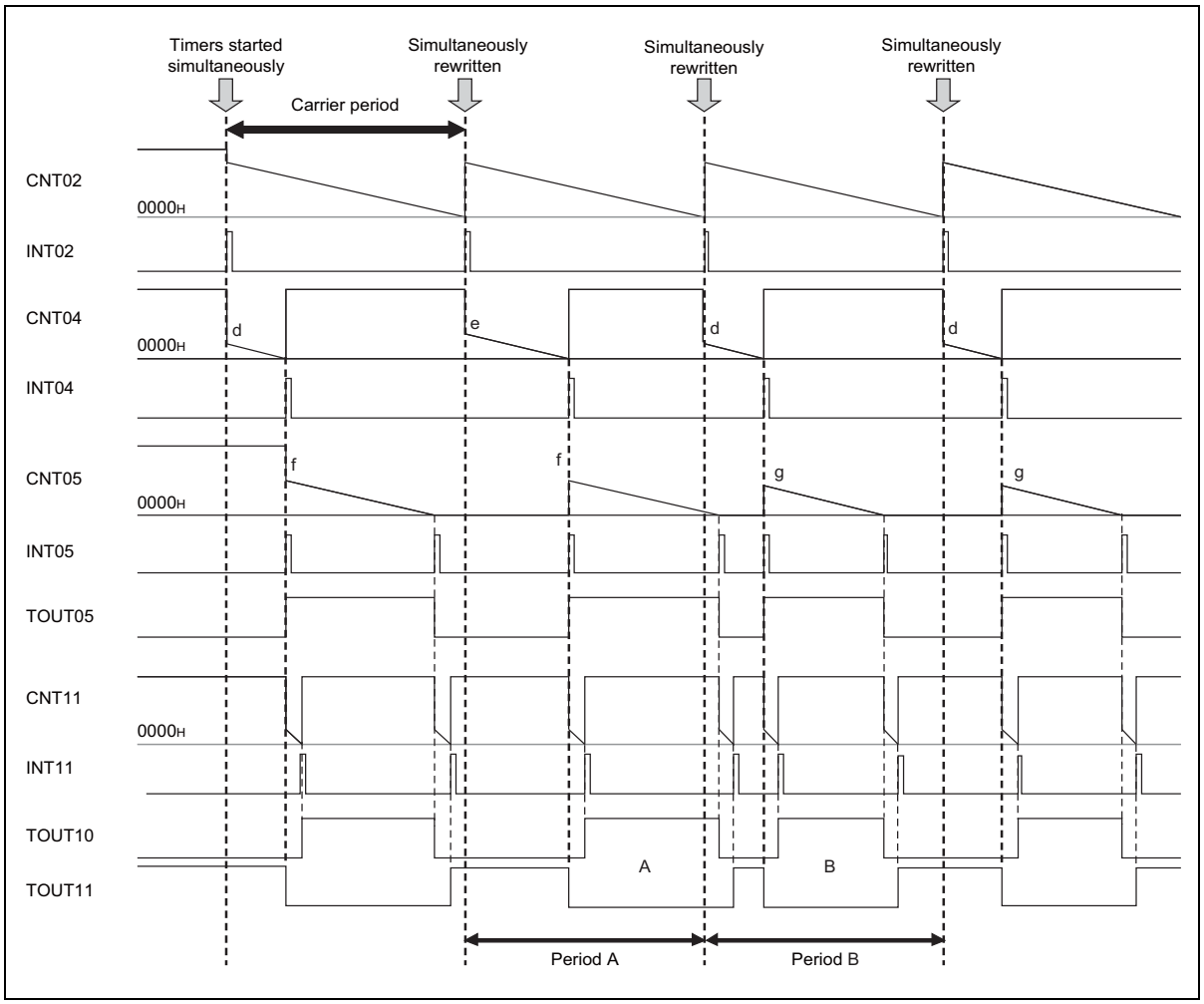


Figure 26-24 PWM Output Using Delay Pulse Output Function with Dead Time

In the above figure, although PWM waveform A is normally output within period A, the PWM clear position actually appears after period A because of the extended delay time. PWM waveform B is then output, which should be output within period B.

The following shows the operation when the three-phase PWM output function with the dead time is used to achieve what is described in the *figure26-24 PWM Output Using Delay Pulse Output Function with Dead Time* in the previous page.

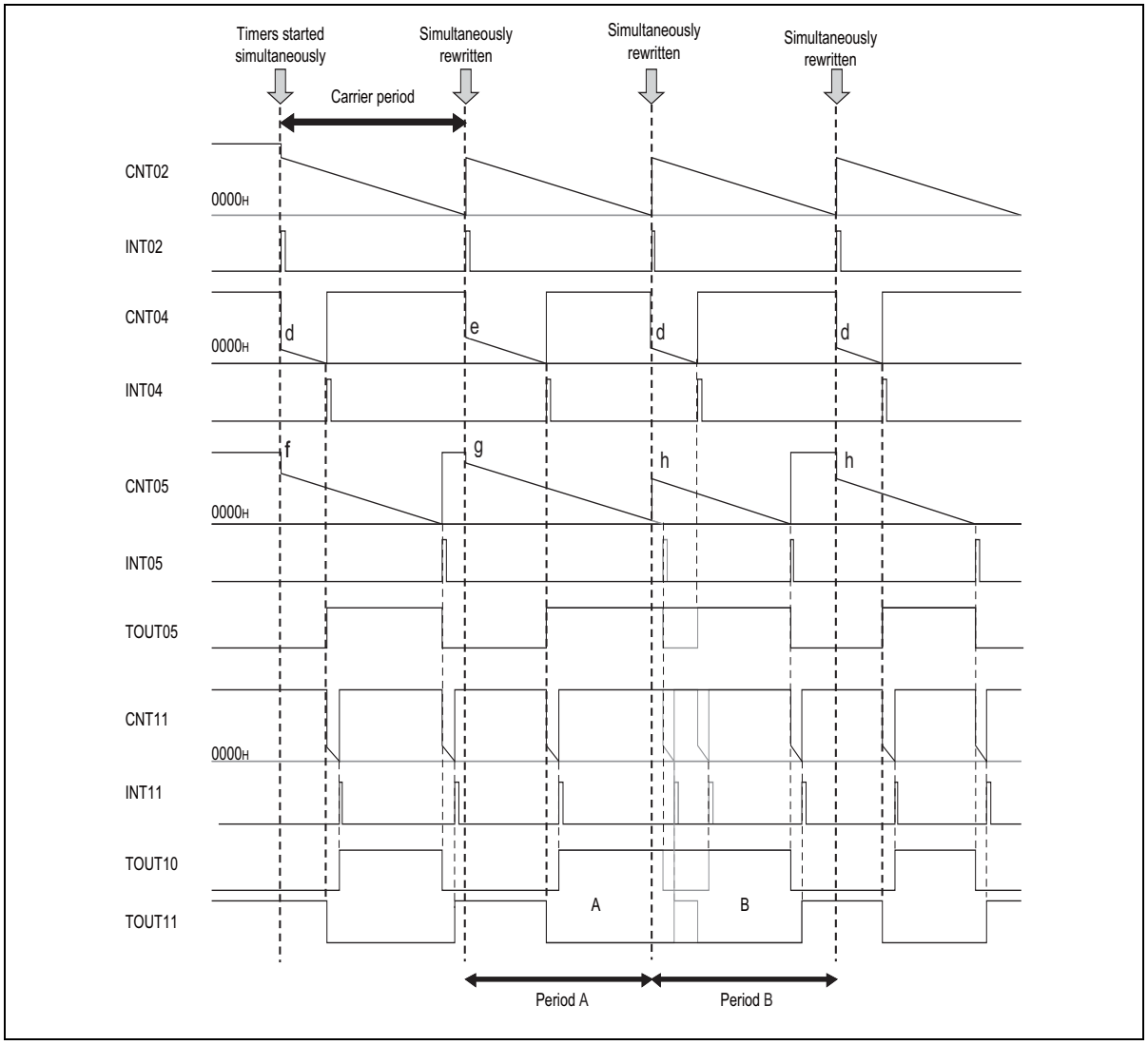


Figure 26-25 PWM Output Using Three-Phase PWM Output Function with Dead Time 1

In the example shown in the above figure, the set timing is delayed and the longer clear time than the carrier period is secured using the three-phase PWM output function with the dead time to allow the PWM output to be extended over carrier period A.

Although the set timing of PWM waveform A with respect to period A is the same as that in the figure in the previous page, the longer clear time than period A here causes periodic generation of the reload operation to precede, thus preventing generation of the clear timing of PWM waveform A.

Moreover, the set timing of PWM waveform B with respect to period B is ignored because the PWM waveform is already in the set state. Therefore, the PWM waveform will not change until the clear timing in period B, resulting in the output of the PWM waveform that is synthesized from PWM waveforms A and B.

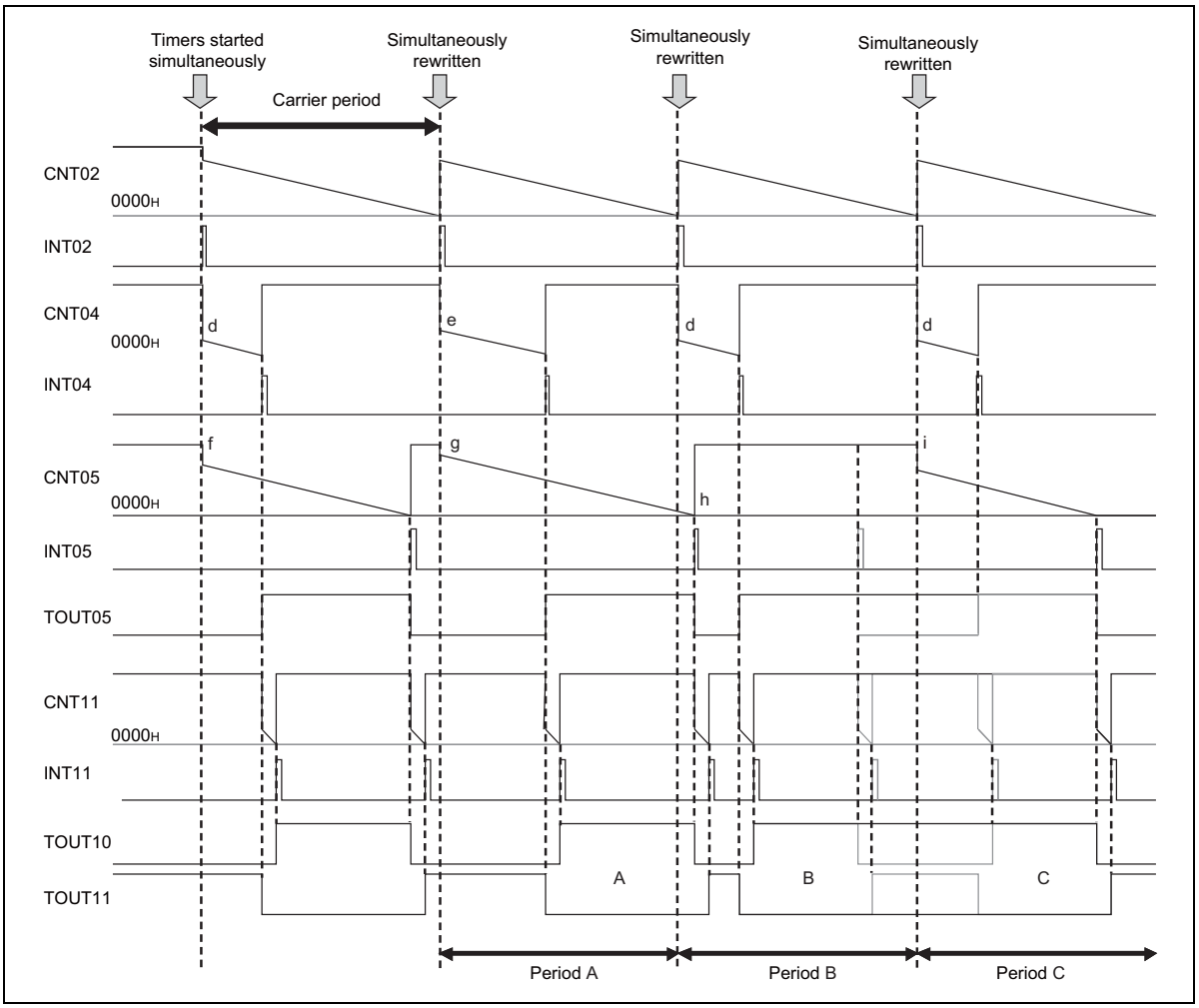


Figure 26-26 PWM Output Using Three-Phase PWM Output Function with Dead Time 2

In the example shown in the above figure, the counter operation based on the longer clear time than period A is continued on period B using the three-phase PWM output function with the dead time to allow the PWM output A to be cleared at the beginning of period B.

Although PWM waveform A with respect to period A is the same as that obtained using the delay pulse output function with the dead time, the clear timing is used at the beginning of period B, thus preventing generation of the clear timing of PWM output B, which should be output in period B.

Moreover, the set timing of PWM waveform C with respect to period C is ignored because the PWM waveform is already in the set state. Therefore, the PWM waveform will not change until the clear timing in period C, resulting in the output of the PWM waveform that is synthesized from PWM waveforms B and C.

As described above, using the delay pulse output function with the dead time enables generation of more flexible PWM output timing than using the three-phase PWM output function with the dead time.

The PIC provides the connection for using the PWM output timing generated by the delay pulse output function as the input of the one-phase PWM output function.

The timing chart based on the delay pulse output function with the dead time is shown in the next page.

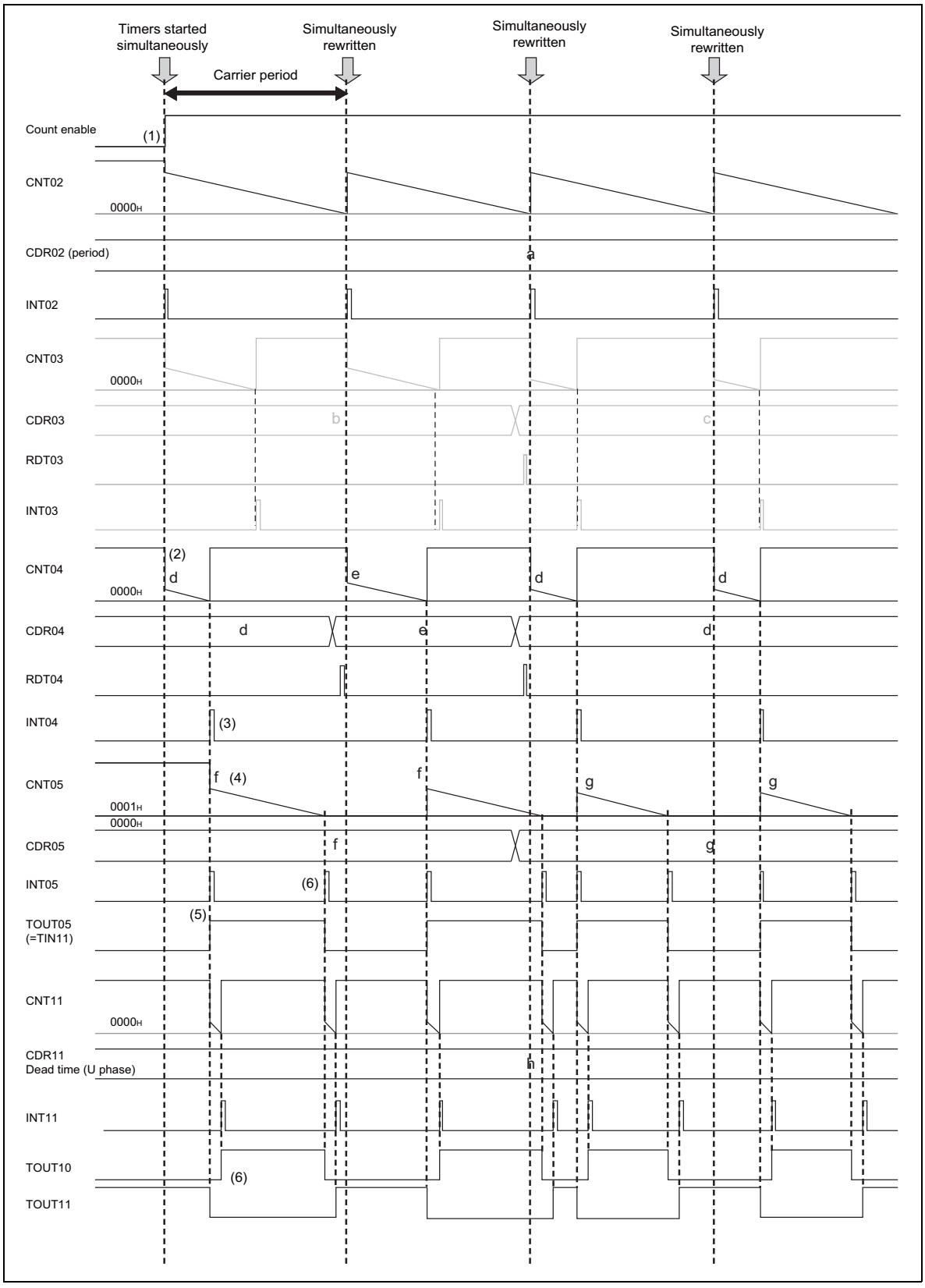


Figure 26-27 Delay Pulse Output Example with Dead Time

The following describes the delay pulse output example with the dead time shown in the Figure 26-27.

(1) Using the simultaneous timer start function, CH02 (carrier period timer) and CH04 (delay timing timer) are started.
Although CH05 (PWM duty timer) and CH11 (dead time timer) are started as well, they will not start counting until the INT04 edge and TIN11 edge are detected, where the former is the CH05 count start timing and the latter is the CH11 count start timing.
CH03 is not described here because it has no influence on the PWM output with this function.

(2) For CH04, CH02 underflow allows the set value in CDR04 to be reloaded to CNT04.

(3) CH04 underflow allows the delay timing signal (INT04) to be generated.

(4) INT04 generation allows the set value to be reloaded from CDR05 to CNT05, thus starting the count operation of CH05 (PWM duty timer).

(5) Here, INT05 is generated and the TOUT05 output level changes to the active level.

(6) Upon CH05 underflow, INT05 is generated again and the TOUT05 output level changes to the inactive level. TOUT05, which changes according to the CH04 and CH05 underflows like this, is supplied to the input TIN11 of the one-phase PWM output function.

(7) With the one-phase PWM output function, the PWM waveform with the dead time is generated and output according to the TIN11 edges detected.

26.4.5.5 Operation Flow

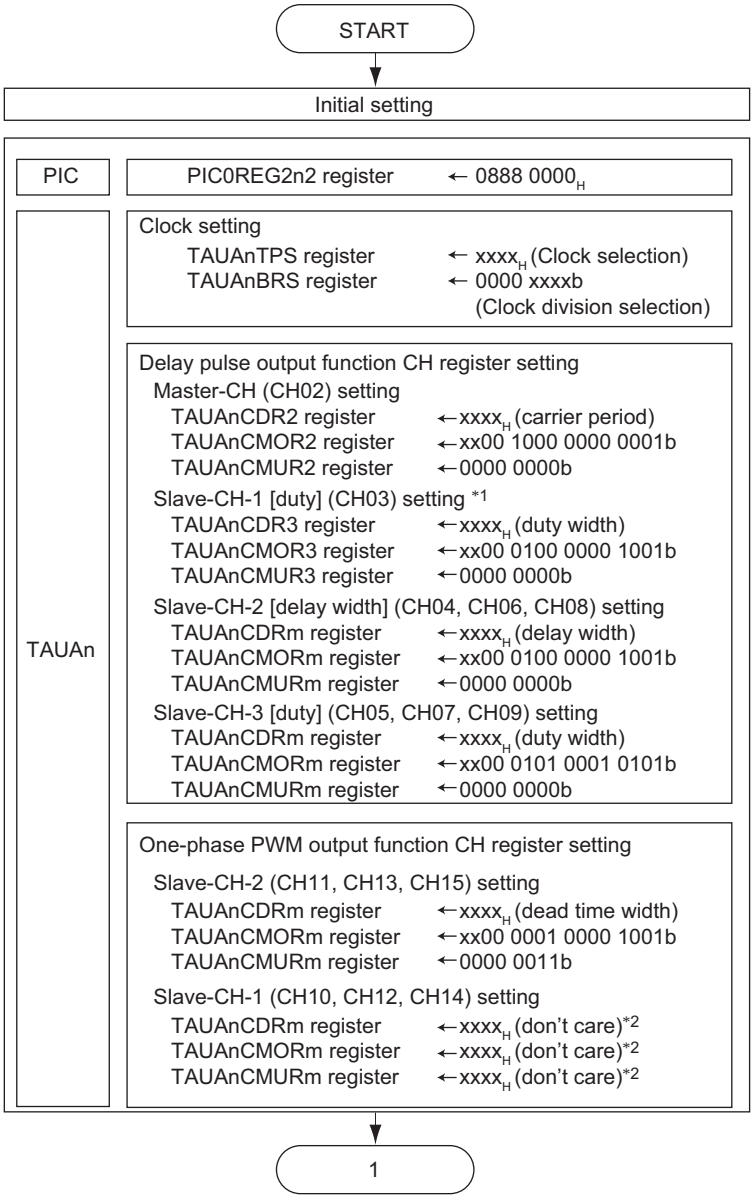


Figure 26-28 Setting Flow (Active High)

Note 1. Although this Slave-CH has no influence on the operation with this function, it should be set because it is one of the channels necessary for the delay pulse output function.

Note 2. Set the function not using TOUTm.

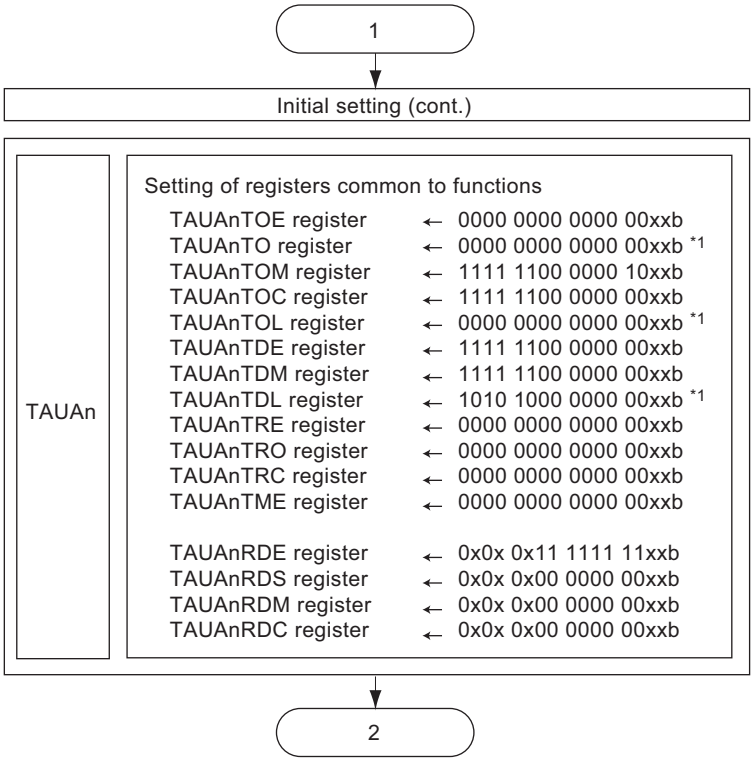


Figure 26-29 Setting Flow (Active High) (cont.)

Note 1. Change the set value depending on the active level of the PWM output.

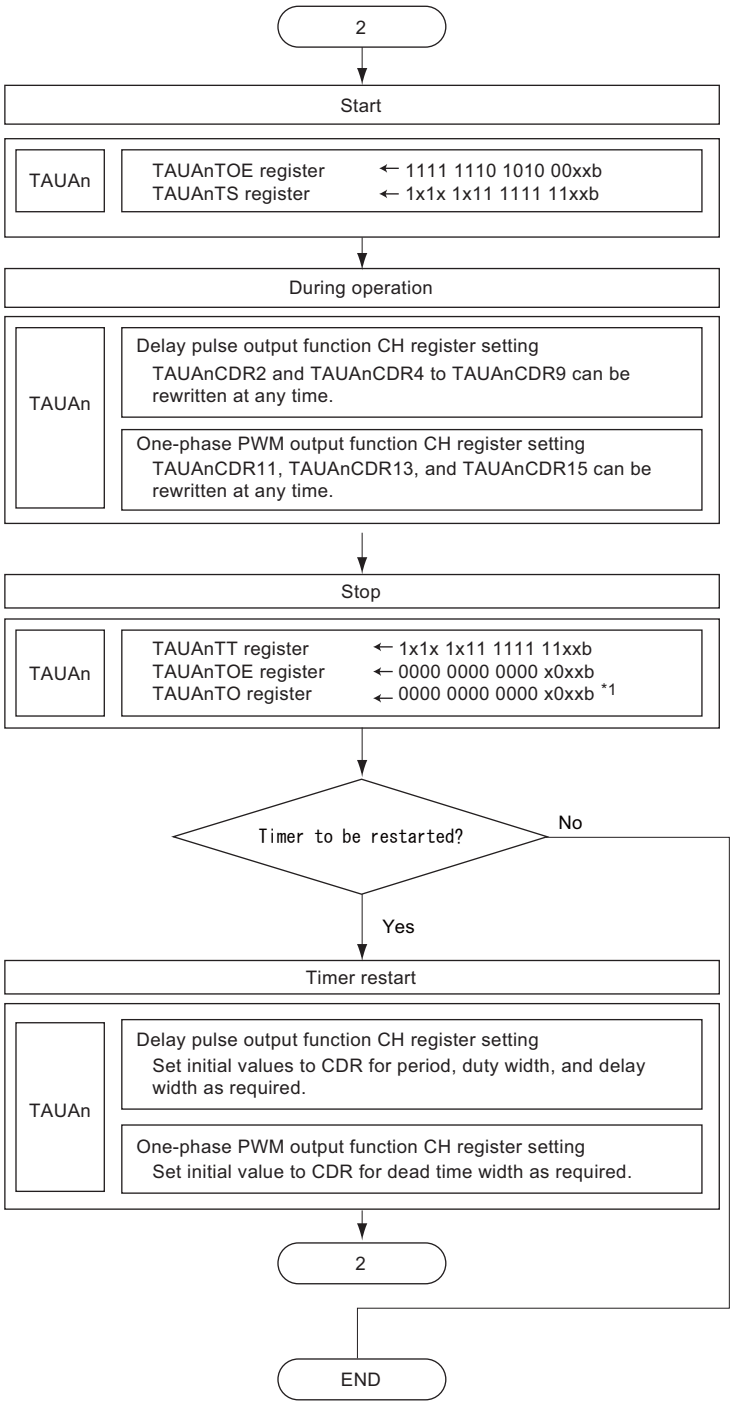


Figure 26-30 Setting Flow (Active High) (cont.)

Note 1. Change the set value depending on the active level of the PWM output.

26.4.5.6 Register Setting Examples for Various Functions

Register setting

Table 26-25 TAUAn Setting

Registers related to TAUAn CH02 (delay pulse output function Master-CH*1)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMOR2	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 0	
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	0	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	INTn is output at start of operation.
	TAUAn CMUR2	1, 0	TIS1 TIS0	0 0	

Registers related to TAUAn CH03 (delay pulse output function Slave CH-1*1, *3)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMOR3	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 0 0	Start trigger: INTn detection of Master-CH
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	Start trigger is enabled during counting.
	TAUAn CMUR3	1, 0	TIS1 TIS0	0 0	

Note 1. Master-CH and Slave-CH are the names defined with the TAUA delay pulse output function. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH.

Note 3. Although this CH has no influence on the operation with this function, it should be set because it is one of the channels necessary for the delay pulse output function.

Note: In TAUAnCMORm for the delay pulse output function, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn CH04, CH06, and CH08 (delay pulse output function Slave CH-2*1) (m = 4, 6, and 8)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 0 0	Start trigger: INTn detection of Master-CH
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	Start trigger is enabled during counting.
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Registers related to TAUAn CH05, CH07, and CH09 (delay pulse output function Slave CH-3*1) (m = 5, 7, and 9)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*2	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	1 0 1	Start trigger: INTn detection of upper CH
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	1	
		3	MD3	0	
		2	MD2	1	
		1	MD1	0	
		0	MD0	1	Start trigger is enabled during counting.
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Note 1. Master-CH and Slave-CH are the names defined with the TAUA delay pulse output function. For details, refer to section 13, Timer Array Unit A (TAUA).

Note 2. The same operating clock should be set for Master-CH and Slave-CH.

Note: In TAUAnCMORm for the delay pulse output function, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn CH11, CH13, and CH15 (one-phase PWM output function) (m = 11, 13, and 15)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care*1	The operating clock is set.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	0	
		10 to 8	STS2 STS1 STS0	0 0 1	Start trigger: Detection of TINn input effective edge.
		7, 6	COS1 COS0	0 0	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	1	
		2	MD2	0	
		1	MD1	0	
		0	MD0	1	Start trigger is enabled during counting.
	TAUAn CMURm	1, 0	TIS1 TIS0	1 1	Detects both edges of TINm as effective edge (high level width)

Note 1. Set the same operating clock as the clock for Master-CH (CH02) for the PWM output function.

Note: In TAUAnCMORm for the one-phase PWM output function, only the CKS (operating clock select) bits can be set arbitrarily; the other control bits are fixed to the values specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

CH10, CH12, and CH14 can be used for any function that does not use TOUTm output (such as the AD trigger output function).

Registers related to TAUAn channels in common

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTOE	15 to 10	TOE15 to TOE10	0 1	Timer operation is stopped. Timer operation is started.
		9	TOE09	0 1	Timer operation is stopped. Timer operation is started.
		8	TOE08	0	Fixed to 0 since TOUT08 is not used.
		7	TOE07	0 1	Timer operation is stopped. Timer operation is started.
		6	TOE06	0	Fixed to 0 since TOUT06 is not used.
		5	TOE05	0 1	Timer operation is stopped. Timer operation is started.
		4	TOE04	0	Fixed to 0 since TOUT04 is not used.
		3	TOE03	0	Fixed to 0 since TOUT03 is not used.
		2	TOE02	0	Fixed to 0 since TOUT02 is not used.
		1, 0	TOE01 TOE00	Don't care	
	TAUAnTO	15 to 10	TO15 to TO10	0*1	Low level is output to TOUT15 to TOUT10 pins.
		9 to 2	TO09 to TO02	0	Low level is output to TOUT09 to TOUT02 pins.
		1, 0	TO01 TO00	Don't care	
	TAUAnTOM	15 to 10	TOM15 to TOM10	1	Synchronous operation mode
		9 to 4	TOM09 to TOM04	0	Independent operation mode
		3	TOM03	1	Synchronous operation mode
		2	TOM02	0	Independent operation mode
		1, 0	TOM01 TOM00	Don't care	
	TAUAnTOC	15 to 10	TOC15 to TOC10	1	Synchronous operation mode 2
		9 to 4	TOC09 to TOC04	0	Independent operation mode
		3	TOC03	0	Synchronous operation mode 1
		2	TOC02	0	Toggle mode
		1, 0	TOC01 TOC00	Don't care	
	TAUAnTOL	15 to 10	TOL15 to TOL10	0*1	Positive logic output (active high)
		9 to 2	TOL09 to TOL02	0	Positive logic output (active high)
		1, 0	TOL01 TOL00	Don't care	
	TAUAnTDE	15 to 10	TDE15 to TDE10	1	Dead time control is enabled.*2
		9 to 2	TDE09 to TDE02	0	Dead time control is disabled.
		1, 0	TDE01 TDE00	Don't care	

Note 1. Change the set value depending on the system used.

Note 2. For dead time control, an even numbered CH and an odd numbered CH are used as a pair to output the positive and negative phase waveforms. For details, refer to section 13, Timer Array Unit A (TAUA).

Registers related to TAUAn channels in common (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnTDM	15 to 10	TDM15 to TDM10	1	Dead time is generated upon detection of TINm input edge of lower odd numbered CH.
		9 to 2	TDM09 to TDM02	0	Invalid since dead time control is disabled.
		1, 0	TDM01 TDM00	Don't care	
	TAUAnTDL	15	TDL15	1*1	Operates for negative phase of W phase.
		14	TDL14	0*1	Operates for positive phase of W phase.
		13	TDL13	1*1	Operates for negative phase of V phase.
		12	TDL12	0*1	Operates for positive phase of V phase.
		11	TDL11	1*1	Operates for negative phase of U phase.
		10	TDL10	0*1	Operates for positive phase of U phase.
		9 to 2	TDL09 to TDL02	0	Invalid since dead time control is disabled.
		1, 0	TDL01 TDL00	Don't care	
	TAUAnTRE	15 to 2	TRE15 to TRE02	0	Realtime output is disabled.
		1, 0	TRE01 TRE00	Don't care	
	TAUAnTRO	15 to 2	TRO15 to TRO02	0	Invalid since realtime output is disabled.
		1, 0	TRO01 TRO00	Don't care	
	TAUAnTRC	15 to 2	TRC15 to TRC02	0	Does not operate as realtime output trigger generating CH.
		1, 0	TRC01 TRC00	Don't care	
	TAUAnTME	15 to 2	TME15 to TME02	0	Modulation and output of timer output and realtime output is disabled.
		1, 0	TME01 TME00	Don't care	

Note 1. Change the set value depending on the system used.

Registers related to TAUAn channels in common (cont.)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAnRDE	15	RDE15	0	Simultaneous rewrite control is disabled.
		14	RDE14	Don't care	
		13	RDE13	0	Simultaneous rewrite control is disabled.
		12	RDE12	Don't care	
		11	RDE11	0	Simultaneous rewrite control is disabled.
		10	RDE10	Don't care	
		9 to 2	RDE09 to RDE02	1	Simultaneous rewrite control is enabled.
		1, 0	RDE01 RDE00	Don't care	
	TAUAnRDS	15	RDS15	0	Simultaneous rewrite control through another upper CH is disabled.
		14	RDS14	Don't care	
		13	RDS13	0	Simultaneous rewrite control through another upper CH is disabled.
		12	RDS12	Don't care	
		11	RDS11	0	Simultaneous rewrite control through another upper CH is disabled.
		10	RDS10	Don't care	
		9 to 2	RDS09 to RDS02	0	Simultaneous rewrite control through Master-CH is enabled.
		1, 0	RDS01 RDS00	Don't care	
	TAUAnRDM	15	RDM15	0	Invalid since simultaneous rewrite control is disabled.
		14	RDM14	Don't care	
		13	RDM13	0	Invalid since simultaneous rewrite control is disabled.
		12	RDM12	Don't care	
		11	RDM11	0	Invalid since simultaneous rewrite control is disabled.
		10	RDM10	Don't care	
		9 to 2	RDM09 to RDM02	0	Simultaneous load is executed at start of Master-CH.
		1, 0	RDM01 RDM00	Don't care	
	TAUAnRDC	15	RDC15	0	Invalid since simultaneous rewrite control is disabled.
		14	RDC14	Don't care	
		13	RDC13	0	Invalid since simultaneous rewrite control is disabled.
		12	RDC12	Don't care	
		11	RDC11	0	Invalid since simultaneous rewrite control is disabled.
		10	RDC10	Don't care	
		9 to 2	RDC09 to RDC02	0	Does not operate as simultaneous rewrite trigger generating CH.
		1, 0	RDC01 RDC00	Don't care	

Table 26-26 PIC setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0 REG2n2	27, 26	PIC0REG2n227 PIC0REG2n226	1 0	The input selected with PIC0REG2n204 bit is selected.
		23, 22	PIC0REG2n223 PIC0REG2n222	1 0	The input selected with PIC0REG2n203 bit is selected.
		19, 18	PIC0REG2n219 PIC0REG2n218	1 0	The input selected with PIC0REG2n202 bit is selected.
		4	PIC0REG2n204	0	TAUAnTOUT09 is selected.
		3	PIC0REG2n203	0	TAUAnTOUT07 is selected.
		2	PIC0REG2n202	0	TAUAnTOUT05 is selected.

26.4.6 Trigger and Pulse Width Measurement Function

26.4.6.1 Functional Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCAn to TAUJ0 and TAUAn.

26.4.6.2 Configuration

Note: In this section, signal names are abbreviated as shown below.

INTm → TAUAnINTm or TAUJnINTm

TINm → TAUAnTINm or TAUJnTINm

TOUTm → TAUAnTOUTm or TAUJnTOUTm

CDRm → TAUAnCDRm or TAUJnCDRm

CNTm → TAUAnCNTm or TAUJnCNTm

Configuration/ Timer Function	ENCA	TAUA	TAUJ
Timer configuration 1	ENCA0, ENCA1	TAUA0, TAUA1	TAUJ0
Timer configuration 2	ENCA0	TAUA0	TAUJ0
Timer configuration 3	ENCA1	TAUA1	TAUJ0

- Setting functions of TAUJ/TAUAn channels

TAU	CH	Function Name	M/S*	Target Trigger for Pulse Width Measurement
TAUJ0	00	TINn input pulse interval measurement function	S	ENCAT0IEC
	01	TINn input pulse interval measurement function	S	ENCAT0IEC
	02	TINn input pulse interval measurement function	S	ENCAT1IEC
	03	TINn input pulse interval measurement function	S	ENCAT1IEC
TAUA0	00	TINn input pulse interval measurement function	S	ENCAT0EQ0 or ENCAT0EQ1
	01	TINn input pulse interval measurement function	S	ENCAT0EQ1
	02	TINn input pulse interval measurement function	S	ENCAT0EQ0
TAUA1	00	TINn input pulse interval measurement function	S	ENCAT1EQ0 or ENCAT1EQ1
	01	TINn input pulse interval measurement function	S	ENCAT1EQ1
	02	TINn input pulse interval measurement function	S	ENCAT1EQ0

Note * M = master channel, S = slave channel

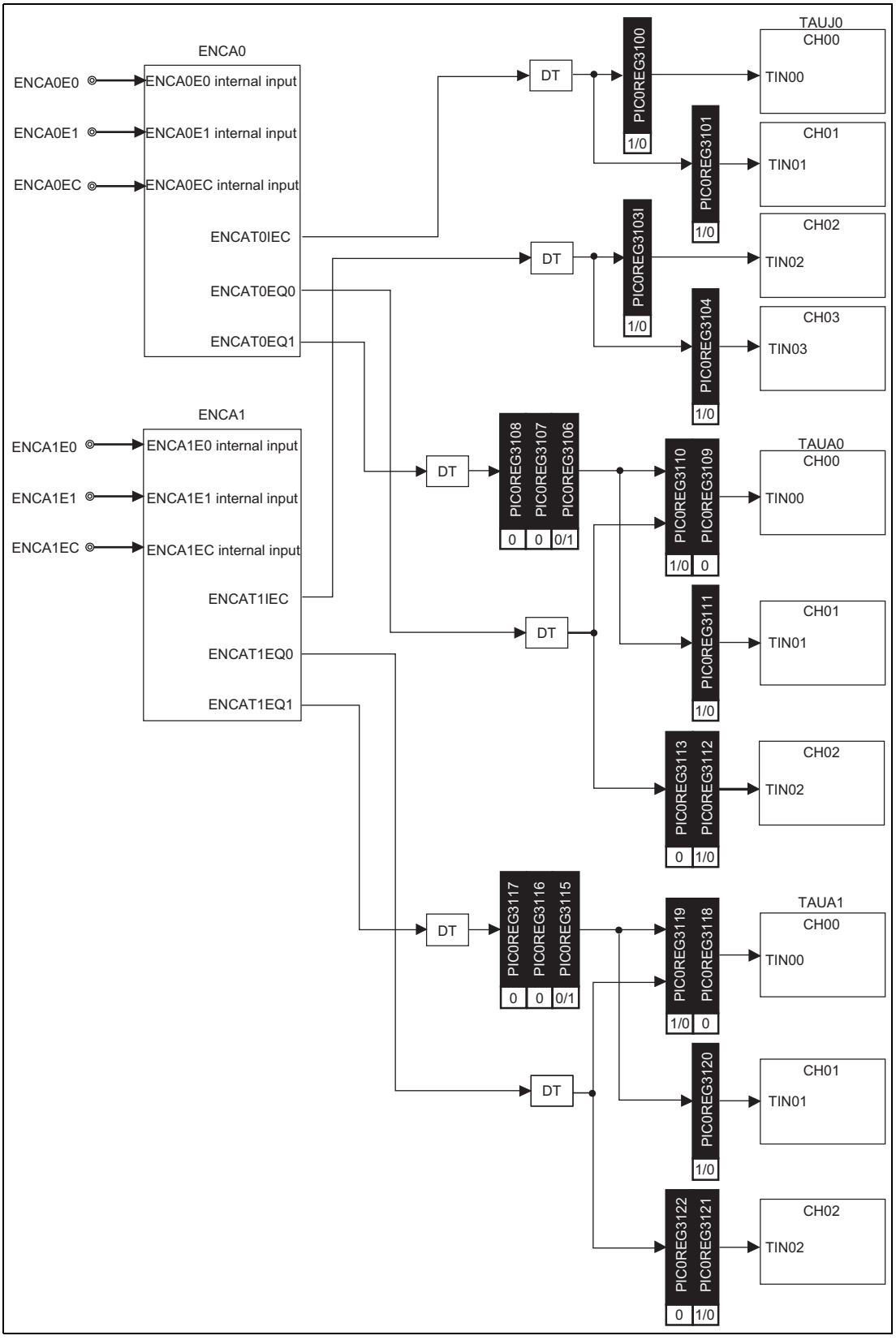


Figure 26-31 Block Diagram

26.4.6.3 Registers

(1) Timer I/O Control Register 31 (PIC0REG31)

Access This register can be read/written in 32-bit units.

Address FF81 C0C0$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	PIC0REG3122	PIC0REG3121	PIC0REG3120	PIC0REG3119	PIC0REG3118	PIC0REG3117	PIC0REG3116
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
PIC0REG3115	0	PIC0REG3113	PIC0REG3112	PIC0REG3111	PIC0REG3110	PIC0REG3109	PIC0REG3108
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0REG3107	PIC0REG3106	0	PIC0REG3104	PIC0REG3103	0	PIC0REG3101	PIC0REG3100
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Table 26-27 PIC0REG31 Contents (1/3)

Bit Position	Bit Name	Function
22 21	PIC0REG3122 PIC0REG3121	Selects the TIN signal input of TAUA1CH02. **PIC0REG3122** / **PIC0REG3121** / **Input Signal** 0 / 0 / TIN pin input (TAUA1CH02 is not used for trigger width measurement) 0 / 1 / DT output signal from ENCAT1EQ0 Other than above / Setting prohibited
20	PIC0REG3120	Selects the TIN signal input of TAUA1CH01. 0: TIN pin input (TAUA1CH01 is not used for trigger width measurement) 1: Signal selected with PIC0REG3115 to PIC0REG3117 (ENCAT1EQ1 signal is measured)
19 18	PIC0REG3118 PIC0REG3119	Selects the TIN signal input of TAUA1CH00. **PIC0REG3119** / **PIC0REG3118** / **Input Signal** 0 / 0 / Signal selected with PIC0REG3115 to PIC0REG3117 (ENCAT1EQ1 signal is measured) 1 / 0 / DT output signal from ENCAT1EQ0 Other than above / Setting prohibited

Table 26-27 PIC0REG31 Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17
16
15</td><td>PIC0REG3117
PIC0REG3116
PIC0REG3115</td><td>Selects the TIN signal input of TAUA1CH00 and TAUA1CH01.
<table>
<tr><th>PIC0REG3117</th><th>PIC0REG3116</th><th>PIC0REG3115</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>TIN pin input (TAUA1CH00 and TAUA1CH01 are not used for trigger width measurement)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>DT output signal from ENCAT1EQ1</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>13
12</td><td>PIC0REG3113
PIC0REG3112</td><td>Selects the TIN signal input of TAUA0CH02.
<table>
<tr><th>PIC0REG3113</th><th>PIC0REG3112</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>TIN pin input (TAUA0CH02 is not used for trigger width measurement)</td></tr>
<tr><td>0</td><td>1</td><td>DT output signal from ENCAT0EQ0</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>11</td><td>PIC0REG3111</td><td>Selects the TIN signal input of TAUA0CH01.
0: TIN pin input (TAUA0CH01 is not used for trigger width measurement)
1: Signal selected with PIC0REG3106 to PIC0REG3108 (ENCAT0EQ1 signal is measured)</td></tr>
<tr><td>10
9</td><td>PIC0REG3110
PIC0REG3109</td><td>Selects the TIN signal input of TAUA0CH00.
<table>
<tr><th>PIC0REG3110</th><th>PIC0REG3109</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>Signal selected with PIC0REG3106 to PIC0REG3108 (ENCAT0EQ1 signal is measured)</td></tr>
<tr><td>1</td><td>0</td><td>DT output signal from ENCAT0EQ0</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>8
7
6</td><td>PIC0REG3108
PIC0REG3107
PIC0REG3106</td><td>Selects the TIN signal input of TAUA0CH00 and TAUA0CH01.
<table>
<tr><th>PIC0REG3108</th><th>PIC0REG3107</th><th>PIC0REG3106</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>TIN pin input (TAUA0CH00 and TAUA0CH01 are not used for trigger width measurement)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>DT output signal from ENCAT0EQ1</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>4</td><td>PIC0REG3104</td><td>Selects the TIN signal input of TAUJ0CH03.
0: TIN pin input (TAUJ0CH03 is not used for trigger width measurement)
1: DT output signal from ENCAT1IEC</td></tr>
<tr><td>3</td><td>PIC0REG3103</td><td>Selects the TIN signal input of TAUJ0CH02.
0: TIN pin input (TAUJ0CH02 is not used for trigger width measurement)
1: DT output signal from ENCAT1IEC</td></tr>
<tr><td>1</td><td>PIC0REG3101</td><td>Selects the TIN signal input of TAUJ0CH01.
0: TIN pin input (TAUJ0CH01 is not used for trigger width measurement)
1: DT output signal from ENCAT0IEC</td></tr>
</table>

Table 26-27 PIC0REG31 Contents (3/3)

Bit Position	Bit Name	Function
0	PIC0REG3100	Selects the TIN signal input of TAUJ0CH00. 0: TIN pin input (TAUJ0CH00 is not used for trigger width measurement) 1: DT output signal from ENCAT0IEC

Note The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

26.4.6.4 Example of Operation

The trigger and pulse width measurement function is achieved by combining the ENCAn interrupt trigger signals (ENCATnIEC, ENCATnEQ0, ENCATnEQ1) and TINn input pulse interval measurement function of TAUAn and TAUJ0.

Also, the DT circuit in the PIC is used to convert the interrupt trigger signal input to TINn into the level-sensitive toggle signal.

The trigger and pulse width measurement function implements measurement of the ENCAn output interrupt trigger signal interval using the TINn input pulse interval measurement function of TAUAn and TAUJ0.

(1) TINn Input Pulse Interval Measurement Function

When the valid TINn edge of TAUAn or TAUJ0 is detected, the CNTn value is captured into CDRn and the CNTn is cleared.

Caution Set the both edges (rising and falling edges) of TINn to be detected as valid for this function.

For details of the TAUAn and TAUJ0 functions, see the corresponding sections.

(2) DT Circuit

The DT circuit is used to convert the interrupt trigger signal output from ENCAn into the level-sensitive toggle signal.

As shown in *figure26-32 DT Circuit Operation*, the output signal is toggled on each input interrupt trigger signal generation.

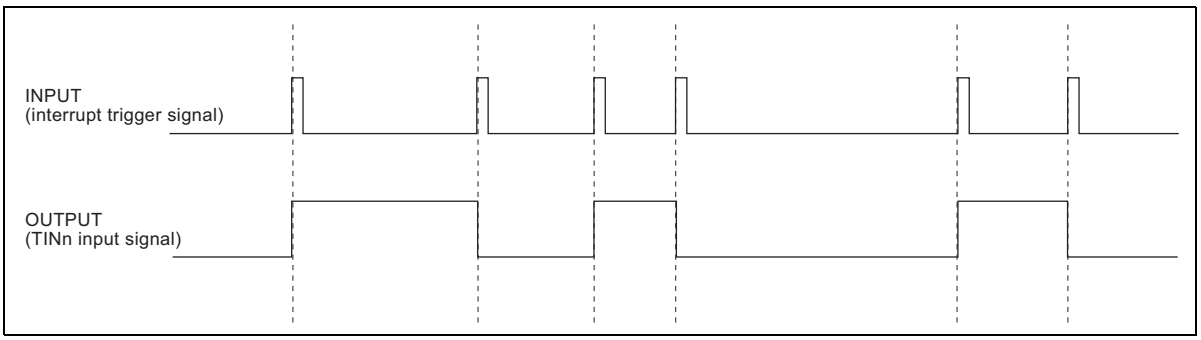


Figure 26-32 DT Circuit Operation

The PIC provides the input signal conversion and signal connection to TAUAn and TAUJ0 to measure the generation interval of interrupt trigger signals from ENCAn.

Figure 26-33 shows the timing chart of the trigger and pulse width measurement function.

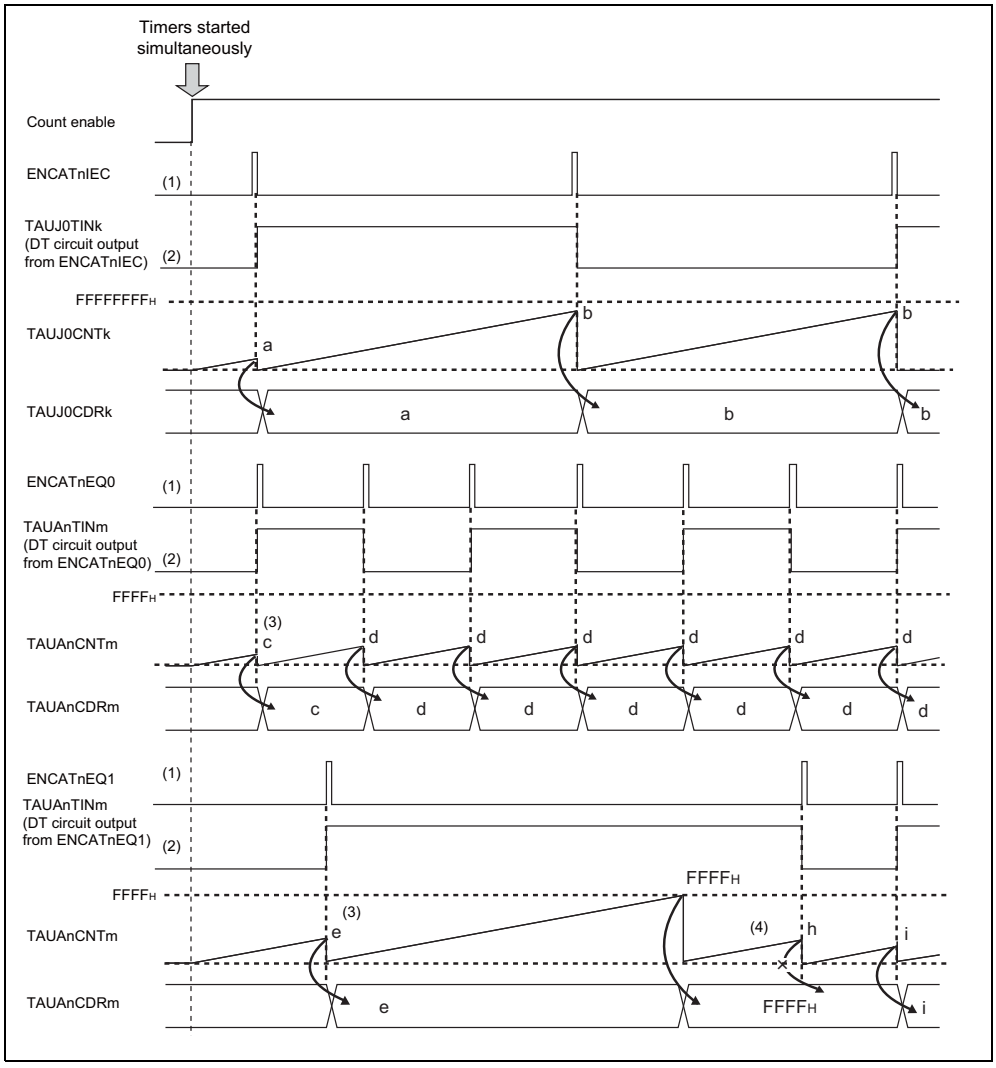


Figure 26-33 Operation Example of Trigger and Pulse Width Measurement Function (m = 0 to 2, k = 0 to 3)

(1) Following signal is output from encoder timer ENCAn as a trigger:
- ENCATnIEC (interrupt trigger signal output when timer counter value is cleared by ENCAnEC input)
- ENCATnEQ0 (interrupt trigger signal output according to timing of a match of timer counter value and value of compare register 0)
- ENCATnEQ1 (interrupt trigger signal output according to timing of a match of timer counter value and value of compare register 1)

(2) The interrupt trigger signal output from ENCAn is converted to the level-sensitive toggle signal by the DT circuit and is output to TINn of TAUAn and TAUJ0.

(3) By setting the both edges of TINn of TAUAn and TAUJ0 as valid, the CNTn value is captured into CDRn on the TINn toggle timing and cleared to 0000_H. This operation is repeated.

(4) When an overflow occurs, the count value $FFFF_H$ ($FFFFFFFF_H$ for TAUJ) is captured but the count value is not captured on the first trigger after the overflow.

With the above operation, the trigger period can be measured.

The following shows the combinations of the interrupt trigger signals and measurement timers and the pertinent PIC register bits for setting the signal paths. Appropriately set the measurement timer and signal path with the PIC register bits according to the interrupt trigger signal to be measured.

Table 26-28 Combinations of Interrupt Trigger Signals and Measurement Timers

Encoder Timer	Interrupt Trigger Signal	Measurement Timer Channel	PIC Register Bit
ENCA0	ENCAT0IEC	TAUJ0-CH00	PIC0REG3100
		TAUJ0-CH01	PIC0REG3101
	ENCAT0EQ0	TAUA0-CH00	PIC0REG3109, PIC0REG3110
		TAUA0-CH02	PIC0REG3112, PIC0REG3113
	ENCAT0EQ1	TAUA0-CH00	PIC0REG3106, PIC0REG3107 PIC0REG3108, PIC0REG3109 PIC0REG3110
		TAUA0-CH01	PIC0REG3106, PIC0REG3107 PIC0REG3108, PIC0REG3111
ENCA1	ENCAT1IEC	TAUJ0-CH02	PIC0REG3103
		TAUJ0-CH03	PIC0REG3104
	ENCAT1EQ0	TAUA1-CH00	PIC0REG3118, PIC0REG3119
		TAUA1-CH02	PIC0REG3121, PIC0REG3122
	ENCAT1EQ1	TAUA1-CH00	PIC0REG3115, PIC0REG3116 PIC0REG3117, PIC0REG3118 PIC0REG3119
		TAUA1-CH01	PIC0REG3115, PIC0REG3116 PIC0REG3117, PIC0REG3120

26.4.6.5 Operation Flow

Figure 26-34 and Figure 26-35 in this section show the general setting flow to measure the pulse interval, which applies to all the following combinations. Change the register settings indicated with the symbol "*" depending on the interrupt trigger signal to be measured and timer to be used. For the combinations of the interrupt trigger signals and timers, see table 26-28, Combinations of Interrupt Trigger Signals and Measurement Timers.

Encoder Timer	Interrupt Trigger Signal	Measurement Timer Channel
ENCA0	ENCAT0IEC	TAUJ0-CH00, TAUJ0-CH01
	ENCAT0EQ0	TAUA0-CH00, TAUA0-CH02
	ENCAT0EQ1	TAUA0-CH01
ENCA1	ENCAT1IEC	TAUJ0-CH02, TAUJ0-CH03
	ENCAT1EQ0	TAUA1-CH00, TAUA1-CH02
	ENCAT1EQ1	TAUA1-CH01

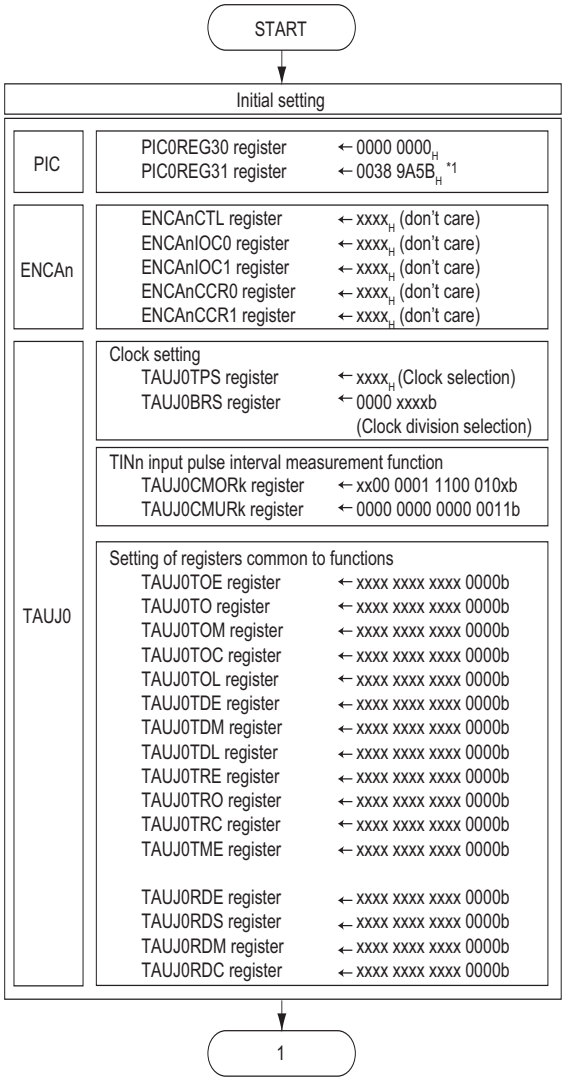


Figure 26-34 Setting Flow (k = 0 to 3)

Note 1. Change the settings depending on the combination of the interrupt trigger signal to be measured and timer to be used.

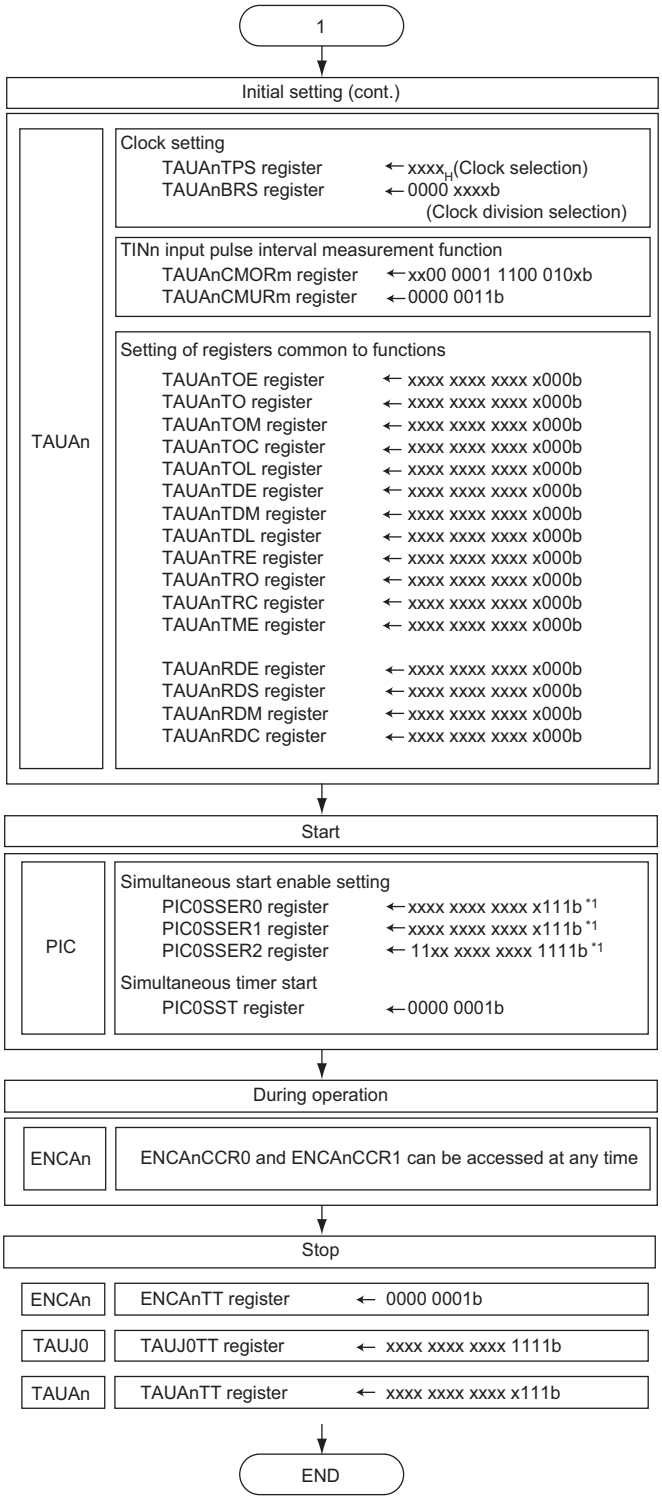


Figure 26-35 Setting Flow (cont.) (m = 0 to 2)

Note 1. Change the settings depending on the combination of the interrupt trigger signal to be measured and timer to be used.

26.4.6.6 Register Setting Examples for Various Functions

Register setting

Tables in this section show the setting example to measure the pulse interval, which applies to all the following combinations. For the combinations of the interrupt trigger signals and timers, see table 26-28, Combinations of Interrupt Trigger Signals and Measurement Timers.

Encoder Timer	Interrupt Trigger Signal	Measurement Timer Channel
ENCA0	ENCAT0IEC	TAUJ0-CH00, TAUJ0-CH01
	ENCAT0EQ0	TAUA0-CH00, TAUA0-CH02
	ENCAT0EQ1	TAUA0-CH01
ENCA1	ENCAT1IEC	TAUJ0-CH02, TAUJ0-CH03
	ENCAT1EQ0	TAUA1-CH00, TAUA1-CH02
	ENCAT1EQ1	TAUA1-CH01

Table 26-29 ENCAn Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
ENCAn	ENCAnCTL	15	ENCAnCME	Don't care	Enables or disables compare match interrupt detection mask.
		14	ENCAnMCS	Don't care	Selects a cancelation trigger of compare match interrupt detection mask.
		13 to 10		0	Fixed to 0
		9	ENCAnCRM1	Don't care	Selects the ENCAnCCR1 function.
		8	ENCAnCRM0	Don't care	Selects the ENCAnCCR0 function
		7	ENCAnCTS	Don't care	Selects trigger of capture operation of ENCAnCCR1
		6, 5		0	Fixed to 0
		4	ENCAnLDE	Don't care	Enables or disables reload operation when underflow is generated.
		3	ENCAnECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR1.
		2	ENCAnECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR0.
		1, 0	ENCAnUDS1, ENCAnUDS0	Don't care	Selects the counter up/down control by ENCAnE0/ENCAnE1.
	ENCAnIOC0	7 to 4		0	Fixed to 0
		3, 2	ENCAnTIS3, ENCAnTIS2	Don't care	Selects the effective edge for capture trigger 1 (ENCAnI1).
		1, 0	ENCAnTIS3, ENCAnTIS2	Don't care	Selects the effective edge for capture trigger 0 (ENCAnI0).
	ENCAnIOC1	7	ENCAnSCE	Don't care	Enables the special encoder clear.
		6	ENCAnZCL	Don't care	Selects the clear level of Z phase for a special encoder clear.
		5	ENCAnBCL	Don't care	Selects the clear level of B phase for a special encoder clear.
		4	ENCAnACL	Don't care	Selects the clear level of A phase for a special encoder clear.
		3, 2	ENCAnECS1 ENCAnECS0	Don't care	Selects encoder clear input (Z phase) edge.
		1, 0	ENCAnEIS1 ENCAnEIS0	Don't care	Selects encoder input (A and B phase) edge.

Table 26-30 TAUJ0 Setting (k = 0 to 3)

TAUJ0 (TINn input pulse interval measurement function)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUJ0	TAUJ0 CMORk	15, 14	CKS1 CKS0	Don't care	Sets the operating clock.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 0	
		7, 6	COS1 COS0	1 1	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	0	
		2	MD2	1	
		1	MD1	0	
		0	MD0	Don't care	
	TAUJ0 CMURk	1, 0	TIS1 TIS0	0 0	

Note: When TAUJ0CMORk is used for the TINn input pulse interval measurement function, the CKS (operating clock selection) and MD0 (INTn output control at the start of counting) bits can be set arbitrarily.

Though the COS (overflow mode selection) bit can also be set arbitrarily, it should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, refer to section 14, Timer Array Unit J (TAUJ).

For TAUJ common registers (TOE, TO, TOM, TOC, TOL, TDE, TDM, TDL, TRE, TRO, TRC, TME, RDE, RDS, RDM, and RDC), only set the bits corresponding to the used channels to 0.

Table 26-31 TAUAn Setting (m = 0 to 2)

TAUAn (TINn input pulse interval measurement function)

Function	Register	Bit Position	Bit Name	Set Value	Note
TAUAn	TAUAn CMORm	15, 14	CKS1 CKS0	Don't care	Sets the operating clock.
		13, 12	CCS1 CCS0	0 0	
		11	MAS	1	
		10 to 8	STS2 STS1 STS0	0 0 0	
		7, 6	COS1 COS0	1 1	
		5		0	Fixed to 0
		4	MD4	0	
		3	MD3	0	
		2	MD2	1	
		1	MD1	0	
		0	MD0	Don't care	
	TAUAn CMURm	1, 0	TIS1 TIS0	0 0	

Note: When TAUAnCMORm is used for the TINn input pulse interval measurement function, the CKS (operating clock selection) and MD0 (INTn output control at the start of counting) bits can be set arbitrarily.

Though the COS (overflow mode selection) bit can also be set arbitrarily, it should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, refer to section 13, Timer Array Unit A (TAUA).

For TAUA common registers (TOE, TO, TOM, TOC, TOL, TDE, TDM, TDL, TRE, TRO, TRC, TME, RDE, RDS, RDM, and RDC), only set the bits corresponding to the used channels to 0.

Table 26-32 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG31	22, 21	PIC0REG3122 PIC0REG3121	0 1	Selects the DT output signal from ENCAT1EQ0 as TIN input signal of TAUA1CH02.
		20	PIC0REG3120	1	Selects the signal selected with PIC0REG3115 to PIC0REG3117 (DT output signal from ENCAT1EQ1) as TIN input signal of TAUA1CH01.
		19, 18	PIC0REG3119 PIC0REG3118	1 0	Selects DT output signal from ENCAT1EQ0 as TIN input signal of TAUA1CH00.
		17 to 15	PIC0REG3117 PIC0REG3116 PIC0REG3115	0 0 1	Selects DT output signal from ENCAT1EQ1 as TIN input signal of TAUA1CH01 and TAUA1CH00.
		13, 12	PIC0REG3113 PIC0REG3112	0 1	Selects DT output signal from ENCAT0EQ0 as TIN input signal of TAUA0CH02.
		11	PIC0REG3111	1	Selects the signal selected with PIC0REG3106 to PIC0REG3108 (DT output signal from ENCAT0EQ1) as TIN input signal of TAUA0CH01.
		10, 9	PIC0REG3110 PIC0REG3109	1 0	Selects DT output signal from ENCAT1EQ0 as TIN input signal of TAUA0CH00.
		8 to 6	PIC0REG3108 PIC0REG3107 PIC0REG3106	0 0 1	Selects DT output signal from ENCAT0EQ1 as TIN input signal of TAUA0CH01 and TAUA0CH00.
		4	PIC0REG3104	1	Selects DT output signal from ENCAT1IEC as TIN input signal of TAUJ0CH03.
		3	PIC0REG3103	1	Selects DT output signal from ENCAT1IEC as TIN input signal of TAUJ0CH02.
		1	PIC0REG3101	1	Selects DT output signal from ENCAT0IEC as TIN input signal of TAUJ0CH01.
		0	PIC0REG3100	1	Selects DT output signal from ENCAT0IEC as TIN input signal of TAUJ0CH00.

26.4.7 A/D Trigger Encoder Capture Function

26.4.7.1 Functional Overview

The encoder count value can be captured synchronously with AD conversion by using the ADC trigger signal as the ENCAn capture signal.

26.4.7.2 Configuration

Configuration/ Timer Function	ENCA
Timer configuration	ENCA0, ENCA1

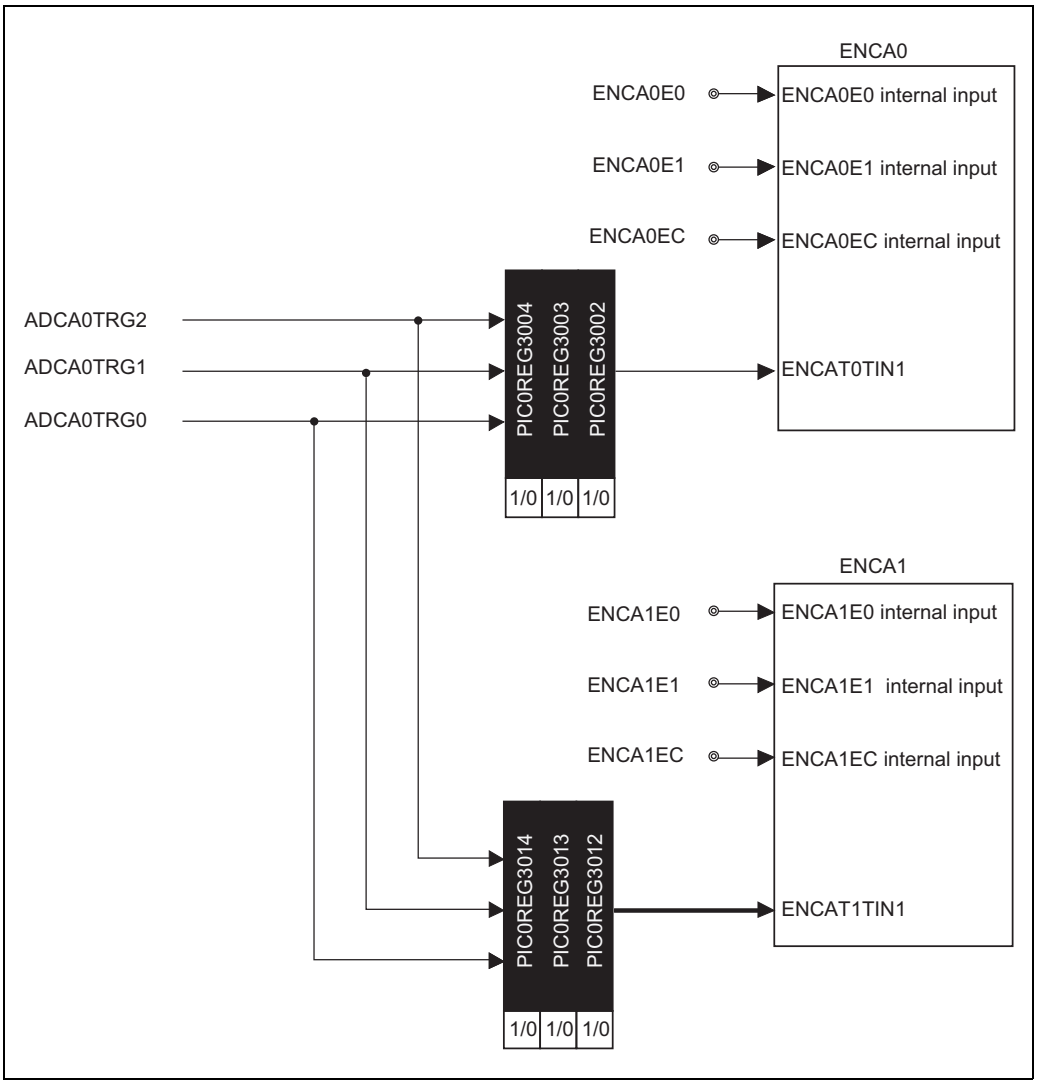


Figure 26-36 Block Diagram

26.4.7.3 Registers

(1) Timer I/O Control Register 30 (PIC0REG30)

Access This register can be read/written in 32-bit units.

Address FF81 C0BC$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	PIC0REG3014	PIC0REG3013	PIC0REG3012	0	0	0	0
R	R/W	R/W	R/W	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	PIC0REG3004	PIC0REG3003	PIC0REG3002	0	0
R	R	R	R/W	R/W	R/W	R	R

Table 26-33 PIC0REG30 Contents (1/2)

Bit Position	Bit Name	Function
14 13 12	PIC0REG3014 PIC0REG3013 PIC0REG3012	Selects the signal to be input to ENCAT1TIN1.

PIC0REG3014	PIC0REG3013	PIC0REG3012	Input Signal
0	0	0	ENCA1 capture operation based on AD trigger signal is not performed.
0	1	0	ADCA0TRG2
0	1	1	ADCA0TRG1
1	0	0	ADCA0TRG0
Other than above			Setting prohibited

Table 26-33 PIC0REG30 Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4
3
2</td><td>PIC0REG3004
PIC0REG3003
PIC0REG3002</td><td>Selects the signal to be input to ENCAT0TIN1.
<table>
<tr><th>PIC0REG3004</th><th>PIC0REG3003</th><th>PIC0REG3002</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>ENCA0 capture operation based on AD trigger signal is not performed.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>ADCA0TRG2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ADCA0TRG1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ADCA0TRG0</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Note The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding PIC connection function.

26.4.7.4 Example of Operation

The AD trigger encoder capture function is implemented by connecting the AD conversion start trigger signals (ADCA0TRG0, ADCA0TRG1, ADCA0TRG2) to ENCAn.

The PIC provides the signal path to connect the above AD conversion start trigger signals to the ENCAn capture trigger input.

Caution When using this function, the ENCAn interrupt trigger signal EQ1 (ENCATnEQ1) should not be selected as the ADC trigger described in section 26.4.2, ADC Trigger Selection Function. If selected, the correct operation cannot be performed because the following loop occurs: ADCAnTRG1 generation → ENCAn capture operation → ENCATnEQ1 generation by capture operation → ADCAnTRG1 generation.

The following shows a timing chart of the AD trigger encoder capture function using the ADCAnTRG1 as a trigger.

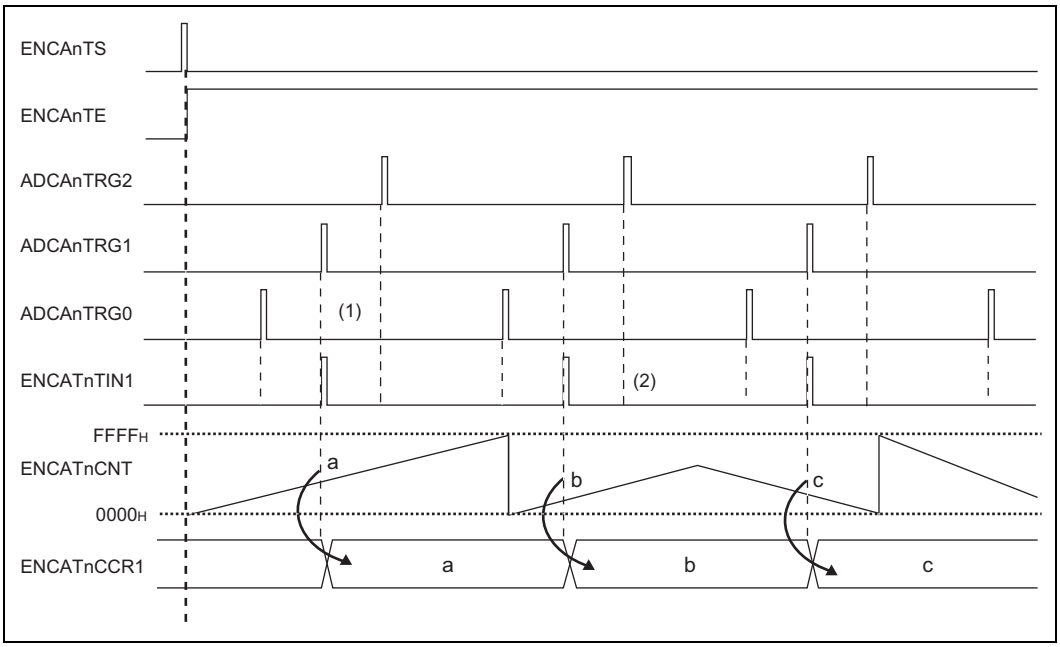


Figure 26-37 Operation Example of AD Trigger Encoder Capture Function

(1) When ADCAnTRG1 is selected as the ENCA1 capture trigger 1 signal ENCATnTIN1, the valid ADCAnTRG1 is input to ENCAn as the ENCATnTIN1 signal and ENCAn is captured.

(2) When an AD trigger signal other than ADCAnTRG1 (ADCAnTRG0, ADCAnTRG2) is generated, the ENCATnTIN1 signal is not generated and ENCAn is not captured.

26.4.7.5 Operation Flow

The figure in this section shows the setting flow to perform capture operation of ENCA0 and ENCA1 encoder timers based on the ADCA0TRG1 signal. Change the setting indicated with the symbol "*" depending on the encoder timer to be captured.

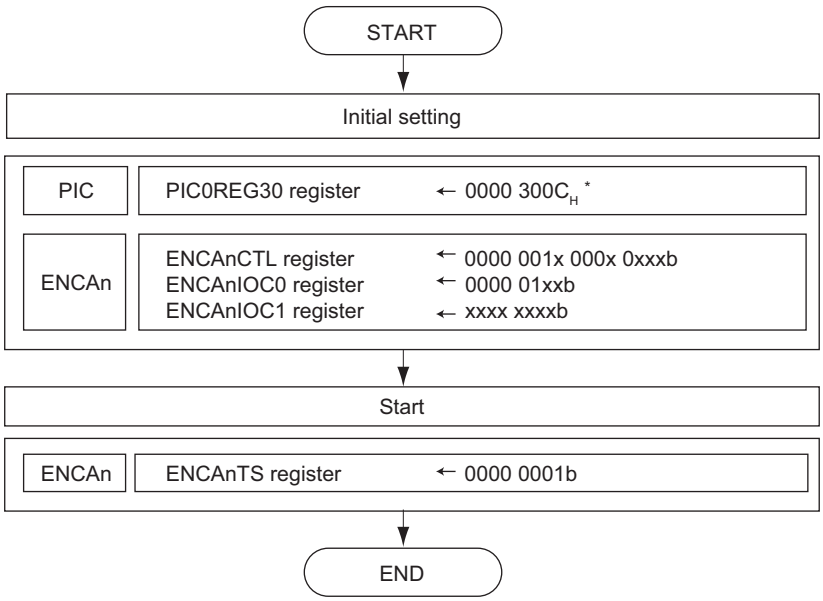


Figure 26-38 Setting Flow

26.4.7.6 Register Setting Examples for Various Functions

Register setting

The tables in this section show the setting example to perform capture operation of ENCA0 and ENCA1 encoder timers based on the ADCA0TRG1 signal. Change the settings depending on the encoder timer to be captured.

Table 26-34 ENCAn Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
ENCAn	ENCAnCTL	15	ENCAnCME	0	Disables compare match interrupt detection mask.
		14	ENCAnMCS	0	Selects a cancelation trigger of compare match interrupt detection mask.
		13 to 10		0	Fixed to 0
		9	ENCAnCRM1	1	Sets ENCAnCCR1 for capture operation.
		8	ENCAnCRM0	Don't care	Selects the ENCAnCCR0 function
		7	ENCAnCTS	0*1	Selects ENCAnI1 input as trigger of capture operation.
		6, 5		0	Fixed to 0
		4	ENCAnLDE	Don't care	Enables or disables reload operation when underflow is generated.
		3	ENCAnECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR1.
		2	ENCAnECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR0.
		1, 0	ENCAnUDS1, ENCAnUDS0	Don't care	Selects the counter up/down control by ENCAnE0/ENCAnE1.
	ENCAnIOC0	7 to 4		0	Fixed to 0
		3, 2	ENCAnTIS3, ENCAnTIS2	0*1 1*1	Sets the rising edge of capture trigger 1 (ENCAnI1) as valid.
		1, 0	ENCAnTIS1, ENCAnTIS0	Don't care	Selects the effective edge for capture trigger 0 (ENCAnI0).
	ENCAnIOC1	7	ENCAnSCE	Don't care	Enables the special encoder clear.
		6	ENCAnZCL	Don't care	Selects the clear level of Z phase for a special encoder clear.
		5	ENCAnBCL	Don't care	Selects the clear level of B phase for a special encoder clear.
		4	ENCAnACL	Don't care	Selects the clear level of A phase for a special encoder clear.
		3, 2	ENCAnECS1 ENCAnECS0	Don't care	Selects encoder clear input (Z phase) edge.
		1, 0	ENCAnEIS1 ENCAnEIS0	Don't care	Selects encoder input (A and B phase) edge.

Note 1. Change the setting depending on the encoder timer to be captured.

Note: All the registers can be set arbitrarily except the bits CRM1 = 1 (ENCAnCCR1 register function) and CTS = 0 (trigger of capture to ENCAnCCR1) in ENCAnCTL for the AD trigger encoder capture function.

Table 26-35 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG30	14 to 12	PIC0REG3014 PIC0REG3013 PIC0REG3012	0*1 1*1 1*1	Selects ADCA0TRG1 as the input signal to ENCAT1TIN1 of ENCA1.
		4 to 2	PIC0REG3004 PIC0REG3003 PIC0REG3002	0*1 1*1 1*1	Selects ADCA0TRG1 as the input signal to ENCAT0TIN1 of ENCA0.

Note 1. Change the setting depending on the encoder timer for captured operation.

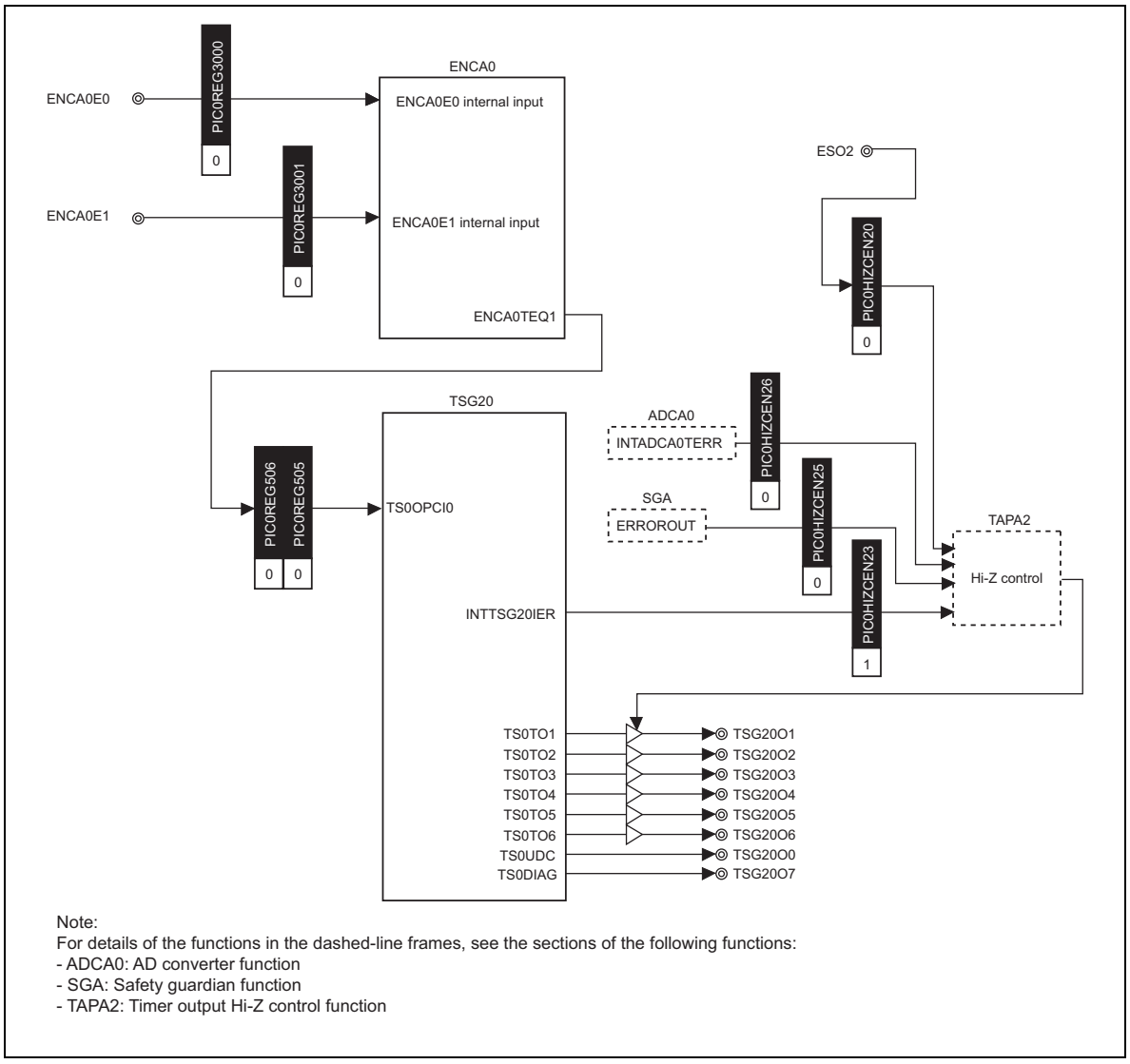
26.4.8 Two-Phase Encoder Control Function (Control Method 1)

26.4.8.1 Functional Overview

This function allows switching of the output pattern of TSG2n using the two-phase encoder (ENCA timer).

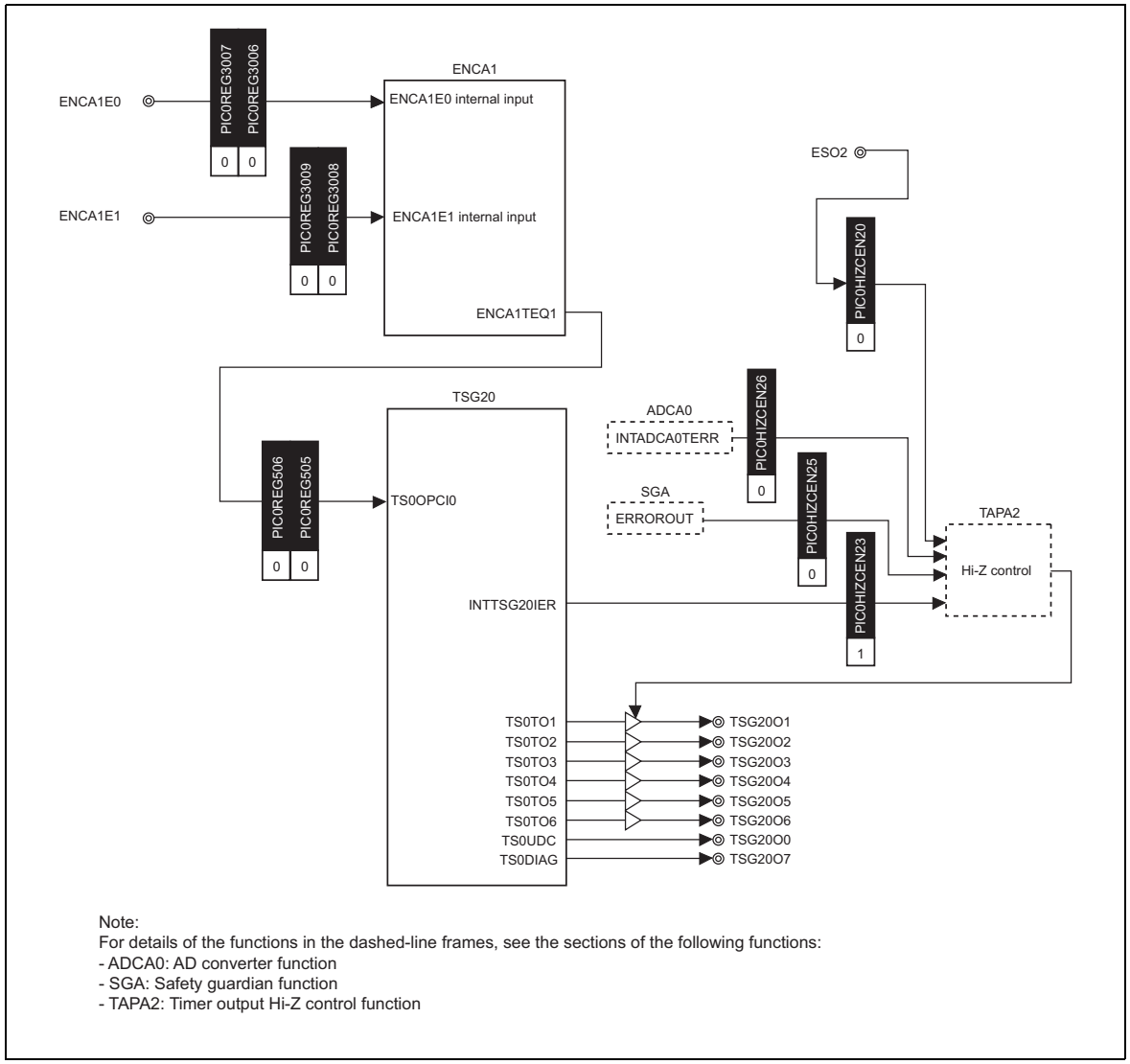
26.4.8.2 Configuration

Configuration/Timer Function	ENCA	TSG2	TAPA
Timer configuration 1	ENCA0	TSG20	TAPA2
Timer configuration 2	ENCA1	TSG20	TAPA2
Timer configuration 3	ENCA1	TSG21	TAPA3



Note:
For details of the functions in the dashed-line frames, see the sections of the following functions:
- ADCA0: AD converter function
- SGA: Safety guardian function
- TAPA2: Timer output Hi-Z control function

Figure 26-39 Block Diagram of Timer Configuration 1



Note:
For details of the functions in the dashed-line frames, see the sections of the following functions:
- ADCA0: AD converter function
- SGA: Safety guardian function
- TAPA2: Timer output Hi-Z control function

Figure 26-40 Block Diagram of Timer Configuration 2

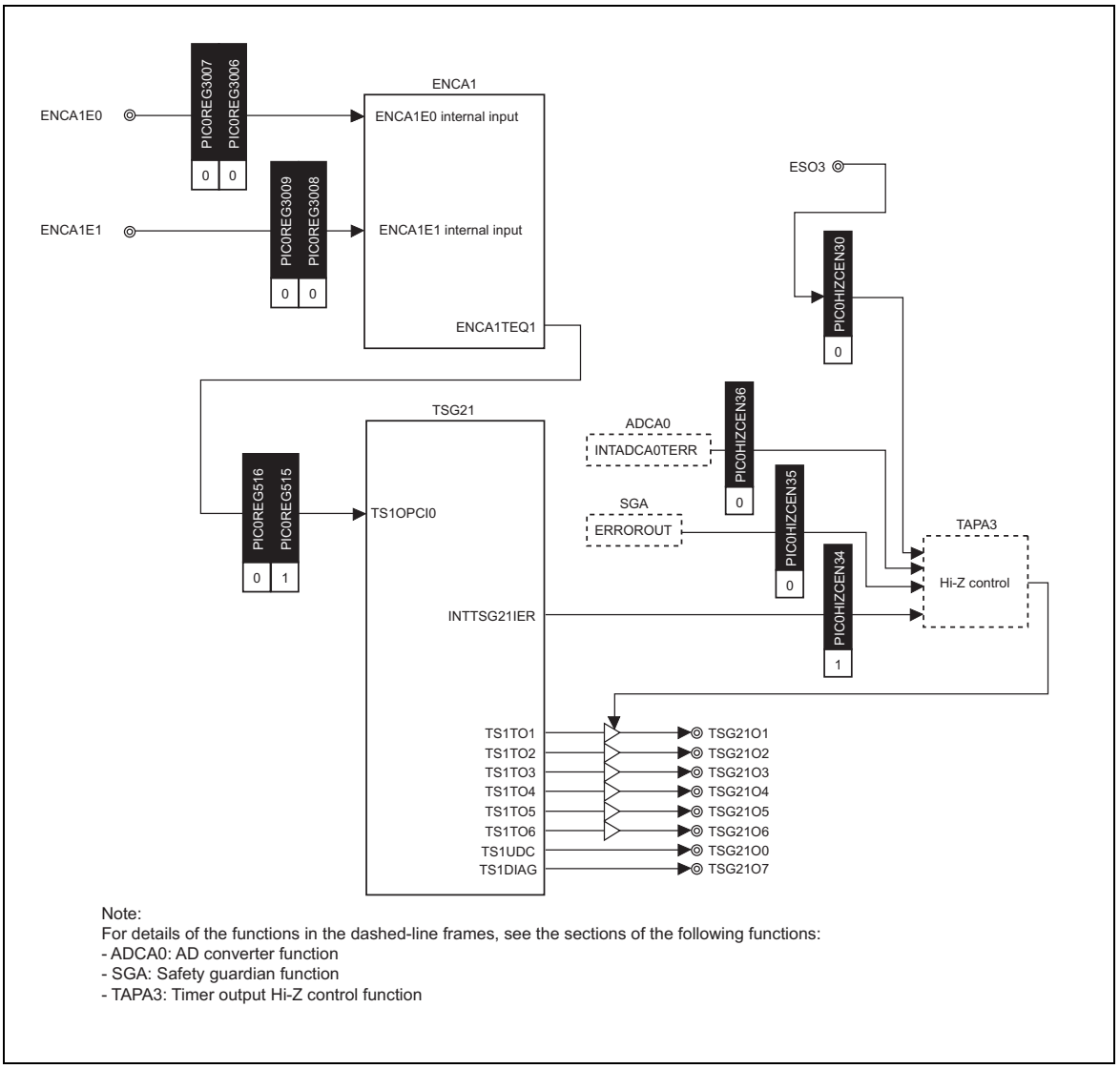


Figure 26-41 Block Diagram of Timer Configuration 3

26.4.8.3 Registers

(1) Timer I/O Control Register 30 (PIC0REG30)

PIC0REG30 is a 32-bit register that selects the timer input signal to control the two-phase encoder.

Access This register can be read/written in 32-bit units.

Address FF81 C0BC$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PIC0REG3009	PIC0REG3008
R	R	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
PIC0REG3007	PIC0REG3006	0	0	0	0	PIC0REG3001	PIC0REG3000
R/W	R/W	R	R	R	R	R/W	R/W

Table 26-36 PIC0REG30 Contents

Bit Position	Bit Name	Function
9, 8	PIC0REG3009, PIC0REG3008	Selects the ENCA1E1 pin input of timer ENCA1. **PIC0REG3009** / **PIC0REG3008** / **Input Signal** 0 / 0 / ENCA1E1 input (ENCA1 timer) Other than above / Setting prohibited
7, 6	PIC0REG3007, PIC0REG3006	Selects the ENCA1E0 pin input of timer ENCA1. **PIC0REG3007** / **PIC0REG3006** / **Input Signal** 0 / 0 / ENCA1E0 input (ENCA1 timer) Other than above / Setting prohibited
1	PIC0REG3001	Selects the TS0PUD signal of TSG20 and selects ENCA0 timer input. 0: ENCA0E1 pin input 1: Setting prohibited
0	PIC0REG3000	Selects the TS0PEC signal of TSG20 and selects ENCA0 timer input. 0: ENCA0E0 pin input 1: Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(2) Timer I/O Control Register 50 (PIC0REG50)

PIC0REG50 selects the timer input signal to control the two-phase encoder.

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

Address FF81 C0E0$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG506	PIC0REG505	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 26-37 PIC0REG50 Contents

Bit Position	Bit Name	Function
6 5	PIC0REG506, PIC0REG505	Selects the signal to be input to TS0OPCI0.

PIC0REG506	PIC0REG505	Input Signal
0	0	ENCA0TEQ1 input (ENCA0 timer)
0	1	ENCA1TEQ1 input (ENCA1 timer)
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(3) Timer I/O Control Register 51 (PIC0REG51)

PIC0REG51 selects the timer input signal to control the two-phase encoder.

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

Address FF81 C0E8$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG516	PIC0REG515	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 26-38 PIC0REG51 Contents

Bit Position	Bit Name	Function
6 5	PIC0REG516, PIC0REG515	Selects the signal to be input to TS0OPCI0.

PIC0REG516	PIC0REG515	Input Signal
0	0	ENCA0TEQ1 input (ENCA0 timer)
0	1	ENCA1TEQ1 input (ENCA1 timer)
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(4) Hi-Z Output Control Register 2 (PIC0HIZCEN2)

PIC0HIZCEN2 selects the input signal to control Hi-Z output.

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

Address FF81 C0DC$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN26	PIC0HIZ CEN25	0	PIC0HIZ CEN23	0	0	PIC0HIZ CEN20
R	R/W	R/W	R	R/W	R	R	R/W

Table 26-39 PIC0HIZCEN2 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN26	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN25	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
3	PIC0HIZCEN23	Enables or disables Hi-Z output control with the INTTSG20IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN20	Enables or disables Hi-Z output control with the ESO2 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(5) Hi-Z Output Control Register 3 (PIC0HIZCEN3)

PIC0HIZCEN3 selects the input signal to control Hi-Z output.

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

Address FF81 C0E4$_H$

Initial value 00$_H$

This register is initialized by any reset

7	6	5	4	3	2	1	0
0	PIC0HIZCEN36	PIC0HIZCEN35	PIC0HIZCEN34	0	0	0	PIC0HIZCEN30
R	R/W	R/W	R/W	R	R	R	R/W

Table 26-40 PIC0HIZCEN3 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN36	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN35	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
4	PIC0HIZCEN34	Enables or disables Hi-Z output control with the INTTSG21IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN30	Enables or disables Hi-Z output control with the ESO3 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

26.4.8.4 Example of Operation

By setting an arbitrary encoder count value to the compare value (ENCAnCCR1) of the ENCAn timer and enabling compare match interrupt (INTENCAnI1) of the ENCAn timer as the trigger of the pattern switch of timer TSG2n, the control of TSG2n output can be achieved.

In this control method, the encoder counter is cleared by a compare match of ENCAnCCR0 with the ENCAn timer value.

It is necessary to set compare value (ENCAnCCR1) at each pattern switch (each INTENCAnI1 interrupt). It is necessary to match the initial output pattern of timer TSG2n to the set value of the compare register (ENCAnCCR1) of the ENCAn timer before start because clear by Z phase input is not performed. Switching between normal and reverse rotations of output patterns should be set with the TSnPSC bit in TSnOPT0.

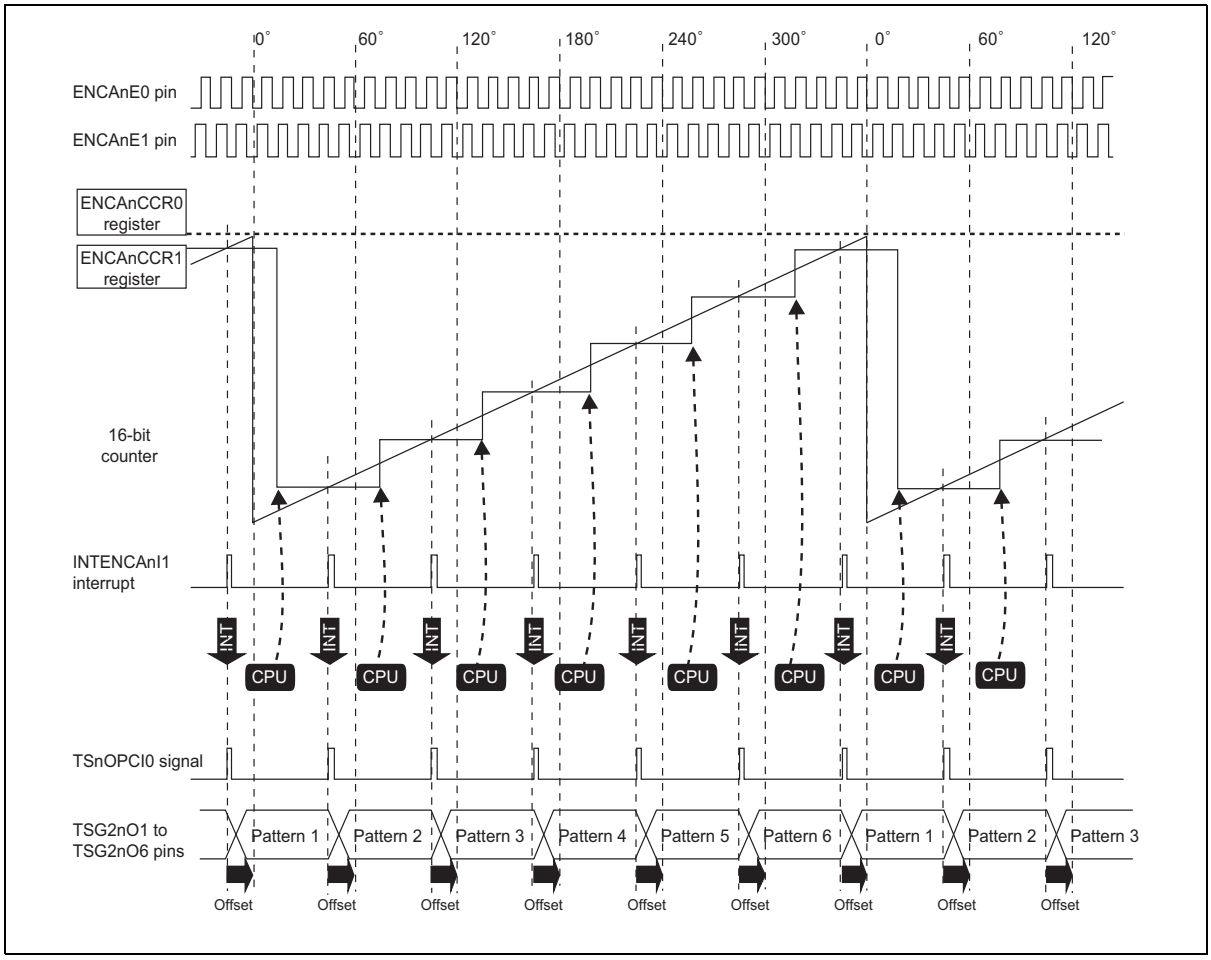


Figure 26-42 Operation Example of Control Method 1 at Up Count

Note ▲ : Write access with CPU

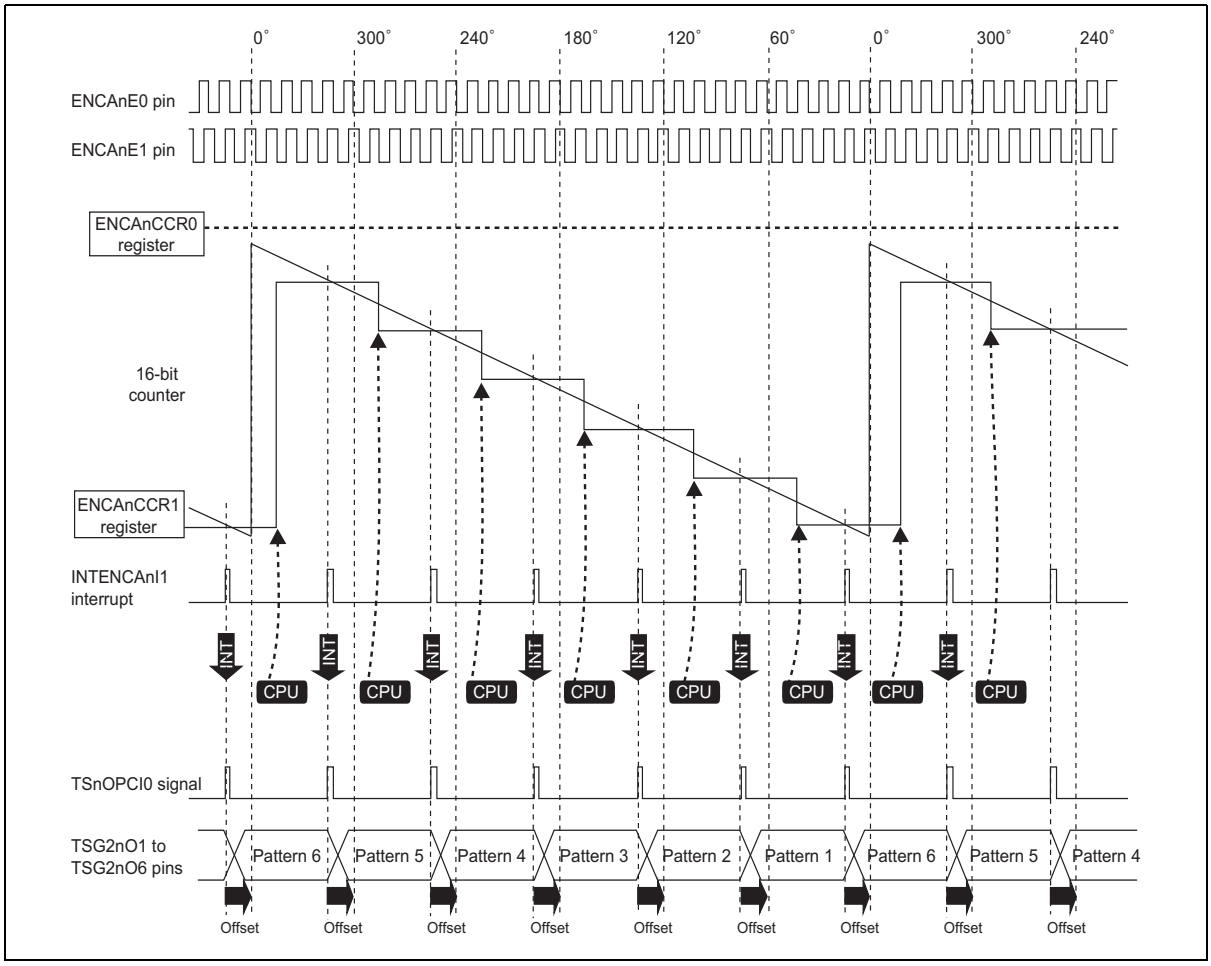


Figure 26-43 Operation Example of Control Method 1 at Down Count

Note ▲ : Write access with CPU

26.4.8.5 Operation Flow

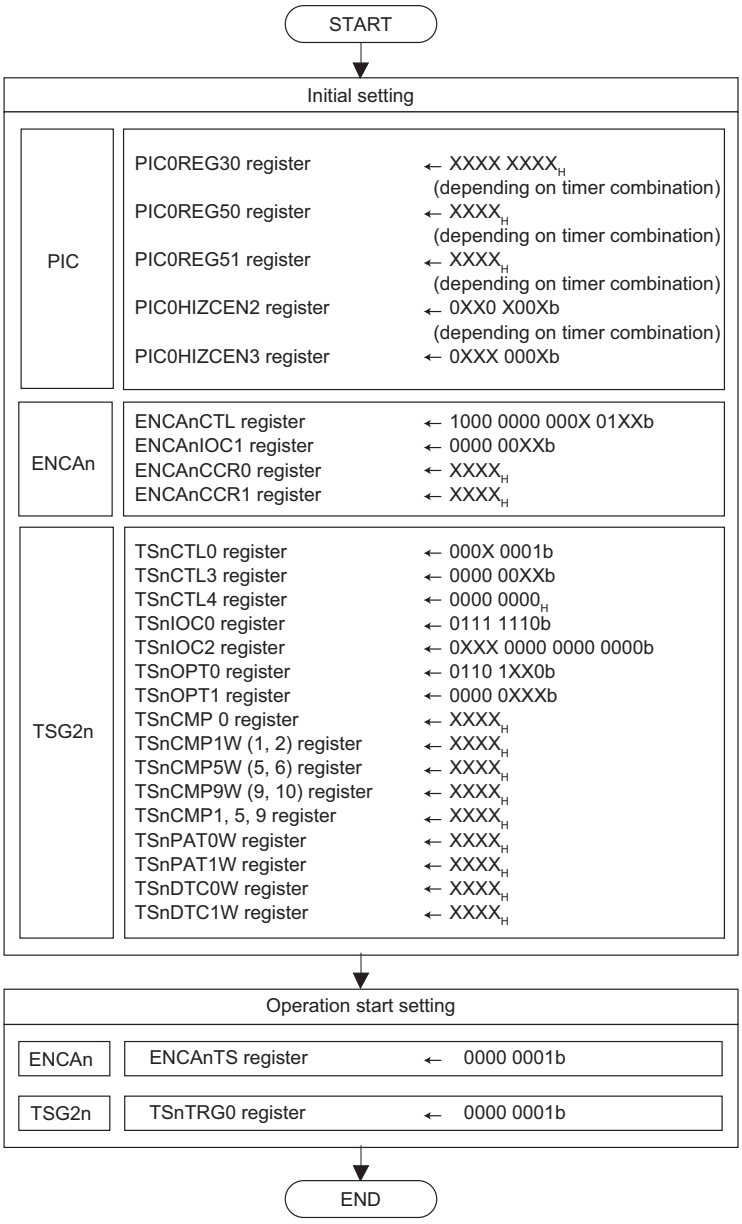


Figure 26-44 Initial Setting and Operation Start Flow

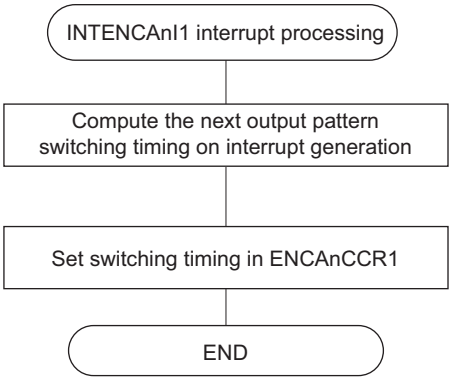


Figure 26-45 ENCAnCCR1 Rewrite Flow during Operation

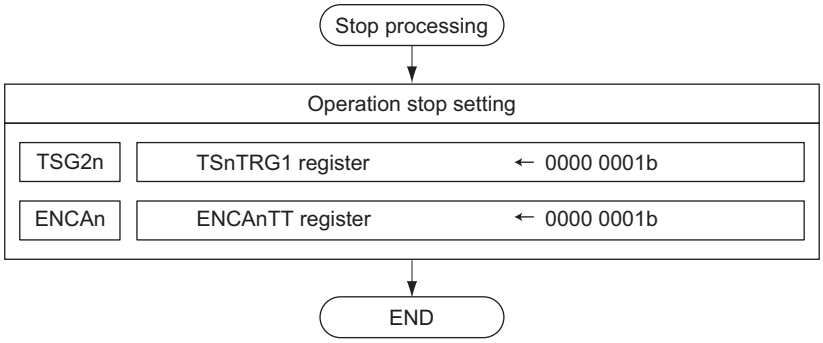


Figure 26-46 Operation Stop Flow

26.4.8.6 Function Setting

The following table shows the register settings.

Register bits not listed in the table are used with the initial values.

Table 26-41 ENCAn Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
ENCAn	ENCAnCTL	15	ENCAnCME	1	Enables or disables compare match interrupt detection mask.
		14	ENCAnMCS	0	Selects a cancelation trigger of compare match interrupt detection mask.
		9	ENCAnCRM1	0	Selects the ENCAnCCR1 function (capture/ compare)
		8	ENCAnCRM0	0	Selects the ENCAnCCR0 function (capture/ compare)
		7	ENCAnCTS	0	Selects trigger of capture operation of ENCAnCCR1
		4	ENCAnLDE	Don't care	Enables or disables reload operation when underflow is generated.
		3	ENCAnECM1	0	Enables or disables clearing of the counter on compare match of ENCAnCCR1.
		2	ENCAnECM0	1	Enables or disables clearing of the counter on compare match of ENCAnCCR0.
		1, 0	ENCAnUDS1, ENCAnUDS0	Don't care	Selects the counter up/down control by ENCAnE0/ENCAnE1.
	ENCAnIOC1	7	ENCAnSCE	0	Enables the special encoder clear.
		6	ENCAnZCL	0	Selects the clear level of Z phase for a special encoder clear.
		5	ENCAnBCL	0	Selects the clear level of B phase for a special encoder clear.
		4	ENCAnACL	0	Selects the clear level of A phase for a special encoder clear.
		3, 2	ENCAnECS1 ENCAnECS0	0	Selects encoder clear input (Z phase) edge.
		1, 0	ENCAnEIS1 ENCAnEIS0	Don't care	Selects encoder input (A and B phase) edge.
	ENCAn CCR0	—	—	Don't care	Compare register Sets a compare value.
	ENCAn CCR1	—	—	Don't care	Compare register Sets a compare value (should be rewritten during timer operation).
	ENCAnCNT	—	—	Don't care	Timer counter register

Table 26-42 TSG2n Setting (1/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCTL0	4	TSnDWD	Don't care	Selects the diag output pulse width.
		1, 0	TSnMD1, TSnMD0	1, 1	Selects operating mode (120-DC).
	TSnCTL3	1	TSnRIA	Don't care	Selects the reload timing of the compare register.
		0	TSnRMC	Don't care	Selects the transfer timing of the compare register.
	TSnCTL4	8	TSnPRE	Don't care	Enables or disables peak reload timing.
		7	TSnVRE	Don't care	Enables or disables valley reload timing.
		6	TSnPIE	0	Enables or disables generation of the peak interrupt.
		5	TSnVIE	0	Enables or disables generation of the valley interrupt.
		4 to 0	TSnRCC4, TSnRCC3, TSnRCC2, TSnRCC1, TSnRCC0	0, 0, 0, 0, 0	Sets the thinning out rate of reload timing and interrupt.
	TSnIOC0	6	TSnTOE6	1	Enables or disables rewriting of the TSnOL6 and TSnTO6 bits in TSnIOC2.
		5	TSnTOE5	1	Enables or disables rewriting of the TSnOL5 and TSnTO5 bits in TSnIOC2.
		4	TSnTOE4	1	Enables or disables rewriting of the TSnOL4 and TSnTO4 bits in TSnIOC2.
		3	TSnTOE3	1	Enables or disables rewriting of the TSnOL3 and TSnTO3 bits in TSnIOC2.
		2	TSnTOE2	1	Enables or disables rewriting of the TSnOL2 and TSnTO2 bits in TSnIOC2.
		1	TSnTOE1	1	Enables or disables rewriting of the TSnOL1 and TSnTO1 bits in TSnIOC2.
	TSnIOC2	14	TSnOL6	Don't care	Selects an active level of the TSG2nO6 output.
		13	TSnOL5	Don't care	Selects an active level of the TSG2nO5 output.
		12	TSnOL4	Don't care	Selects an active level of the TSG2nO4 output.
		11	TSnOL3	Don't care	Selects an active level of the TSG2nO3 output.
		10	TSnOL2	Don't care	Selects an active level of the TSG2nO2 output.
		9	TSnOL1	Don't care	Selects an active level of the TSG2nO1 output.
	TSnOPT0	6	TSnSOC	0	Selects the control of the timer output with software.
		5	TSnSTE	1	Enables or disables the control with the pattern output trigger.
		4	TSnPOT	1	Selects the pattern output trigger. Selects either the output pattern switching by the external pattern input pin (TSG2nPTSI0 to TSG2nPTSI2) or output switching by the TSnOPCI1 and TSnOPCI0 rising edge as the pattern output trigger.
		3	TSnPSS	1	Selects the pattern output order switching method. The pattern output order is switched with TSnPSC.
		2	TSnIDC	Don't care	Selects the direction of the motor rotation.
		1	TSnPSC	Don't care	Selects the order of the timer output pattern during semi-automatic cruise control.
	TSnOPT1	2 to 0	TSnSPC2 to TSnSPC0	Don't care	Sets the timer output pattern in software output function mode.

Table 26-42 TSG2n Setting (2/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCMP0	—	—	Don't care	Compare register Sets the PWM period.
	TSnCMP1W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP5W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP9W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP 1, 5, 9	—	—	Don't care	Compare register Sets the compare value.
	TSnPAT0W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnPAT1W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnDTC0W	—	—	Don't care	Dead time set register Sets the dead time value.
	TSnDTC1W	—	—	Don't care	Dead time set register Sets the dead time value.

Table 26-43 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG30	9, 8	PIC0REG3009, PIC0REG3008	See the block diagram.	Selects the ENCA1E1 pin input.
		7, 6	PIC0REG3007, PIC0REG3006	See the block diagram.	Selects the ENCA1E0 pin input.
		1	PIC0REG3001	See the block diagram.	Selects the ENCA0E1 pin input.
		0	PIC0REG3000	See the block diagram.	Selects the ENCA0E0 pin input.
	PIC0REG50	6, 5	PIC0REG506, PIC0REG505	See the block diagram.	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		4	PIC0REG504	See the block diagram.	Selects the ENCA0TCSO/ENCA1TCSO signal.
	PIC0REG51	6, 5	PIC0REG516 PIC0REG515	See the block diagram.	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		4	PIC0REG514	See the block diagram.	Selects the ENCA0TCSO/ENCA1TCSO signal.
	PIC0HIZ CEN2	6	PIC0HIZ CEN26	See the block diagram.	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZ CEN25	See the block diagram.	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		3	PIC0HIZ CEN23	See the block diagram.	Enables or disables the INTTSG20IER interrupt as Hi-Z control signal input.
		0	PIC0HIZ CEN20	See the block diagram.	Enables or disables the ESO2 pin input as Hi-Z control signal input.
	PIC0HIZ CEN3	6	PIC0HIZ CEN36	See the block diagram.	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZ CEN35	See the block diagram.	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		4	PIC0HIZ CEN34	See the block diagram.	Enables or disables the INTTSG21IER interrupt as Hi-Z control signal input.
		0	PIC0HIZ CEN30	See the block diagram.	Enables or disables the ESO3 pin input as Hi-Z control signal input.

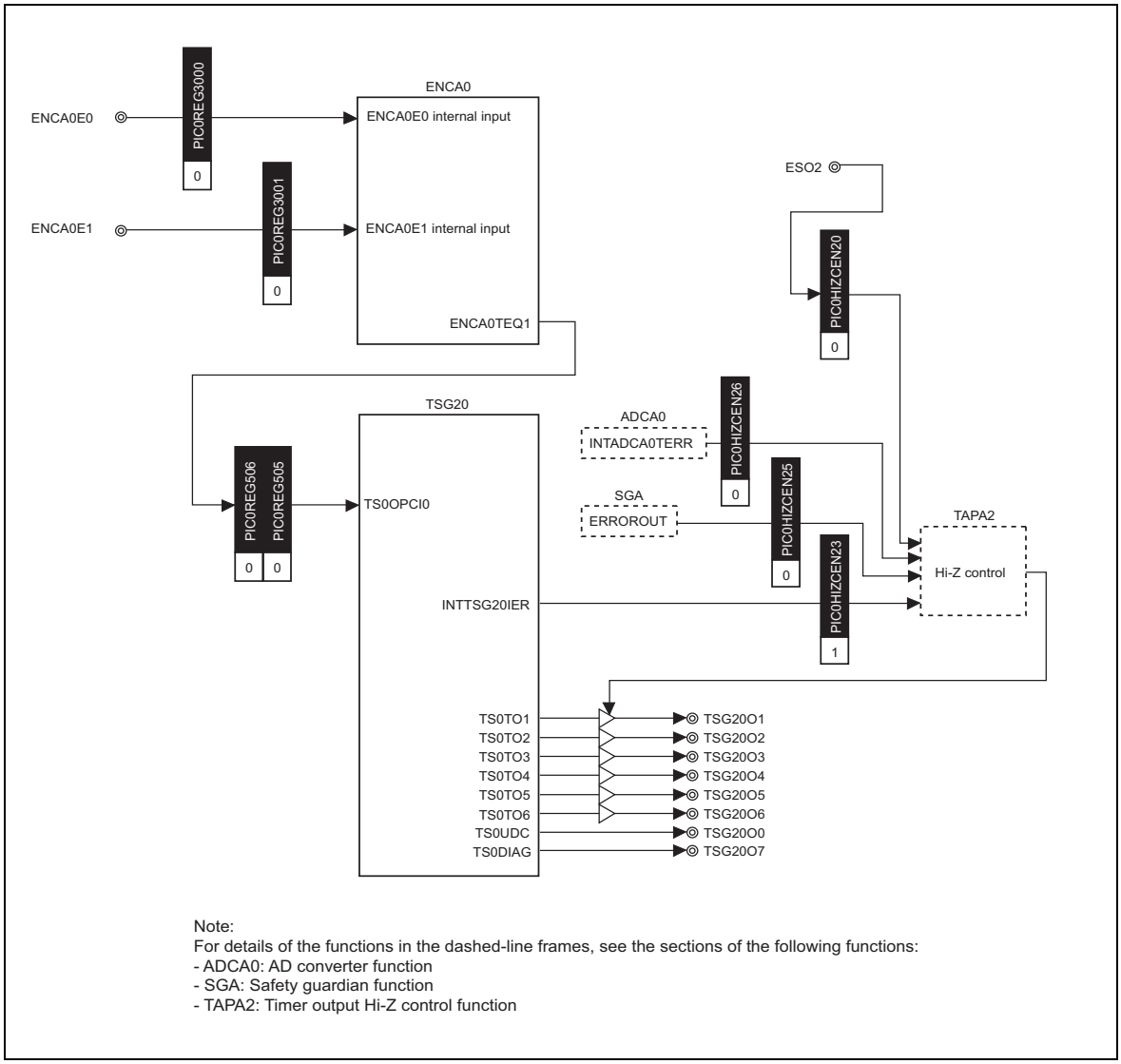
26.4.9 Two-Phase Encoder Control Function (Control Method 2)

26.4.9.1 Functional Overview

This function allows changing the TSG2n output patterns using the two-phase encoder (ENCAn timer). The output patterns can also be switched with the advance and retard control.

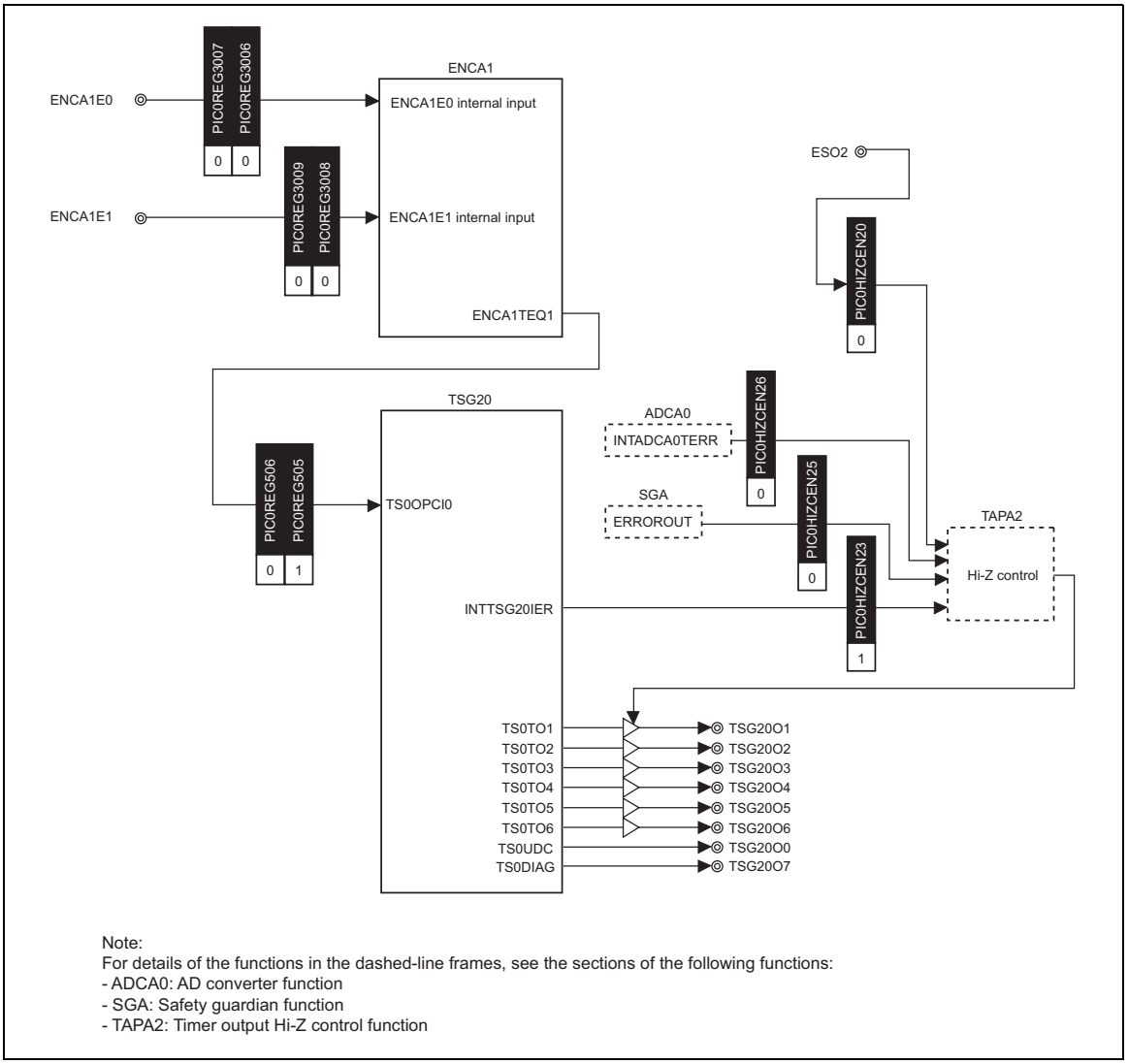
26.4.9.2 Configuration

Configuration/Timer Function	ENCA	TSG2	TAPA
Timer configuration 1	ENCA0	TSG20	TAPA2
Timer configuration 2	ENCA1	TSG20	TAPA2
Timer configuration 3	ENCA1	TSG21	TAPA3



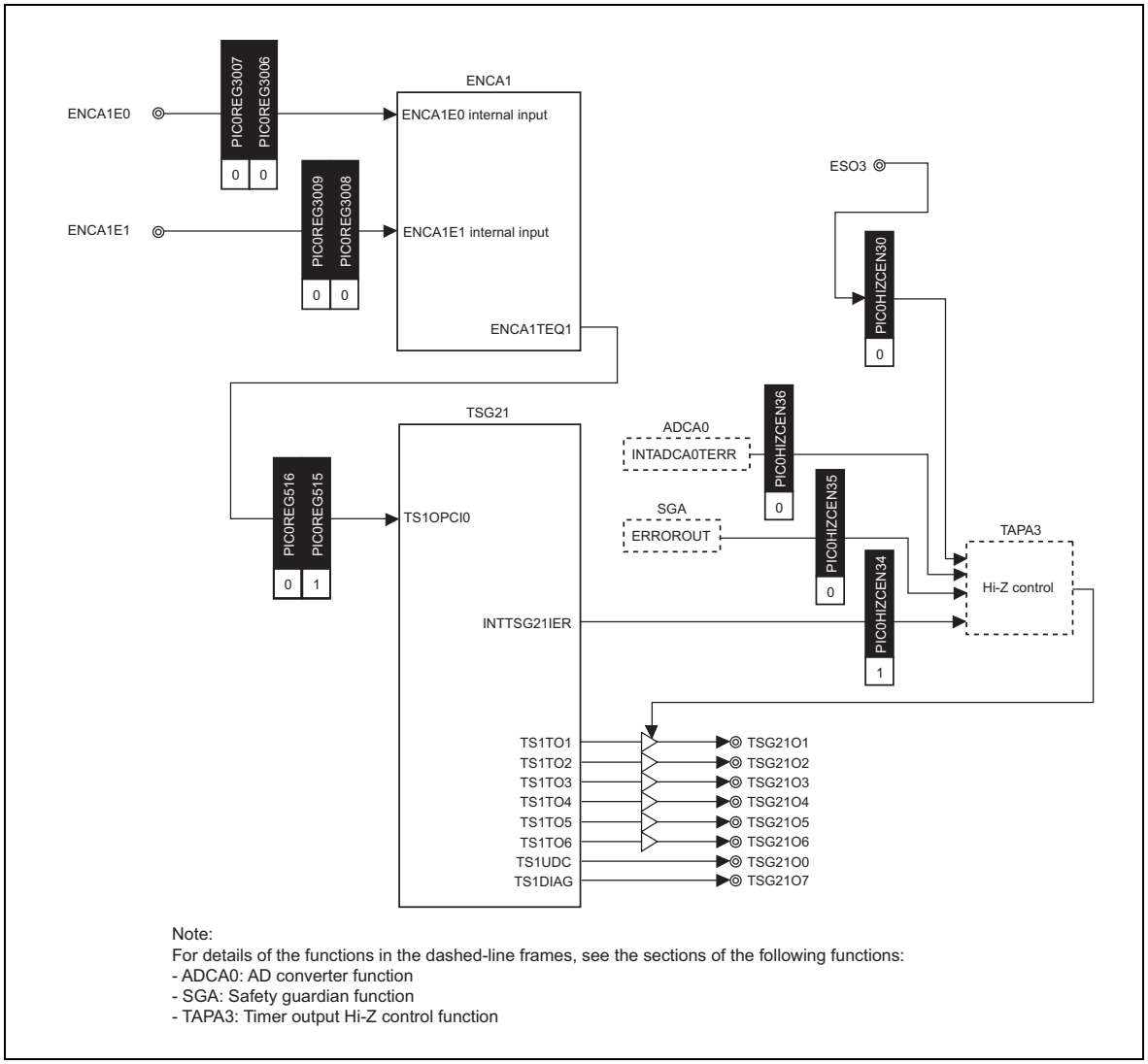
Note:
For details of the functions in the dashed-line frames, see the sections of the following functions:
- ADCA0: AD converter function
- SGA: Safety guardian function
- TAPA2: Timer output Hi-Z control function

Figure 26-47 Block Diagram of Timer Configuration 1



Note:
For details of the functions in the dashed-line frames, see the sections of the following functions:
- ADCA0: AD converter function
- SGA: Safety guardian function
- TAPA2: Timer output Hi-Z control function

Figure 26-48 Block Diagram of Timer Configuration 2



Note:
For details of the functions in the dashed-line frames, see the sections of the following functions:
- ADCA0: AD converter function
- SGA: Safety guardian function
- TAPA3: Timer output Hi-Z control function

Figure 26-49 Block Diagram of Timer Configuration 3

26.4.9.3 Registers

(1) Timer I/O Control Register 30 (PIC0REG30)

PIC0REG30 is a 32-bit register that selects the timer input signal to control the two-phase encoder.

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

Address FF81 C0BC$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PIC0REG3009	PIC0REG3008
R	R	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
PIC0REG3007	PIC0REG3006	0	0	0	0	PIC0REG3001	PIC0REG3000
R/W	R/W	R	R	R	R	R/W	R/W

Table 26-44 PIC0REG30 Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>PIC0REG3009,
PIC0REG3008</td><td>Selects the ENCA1E1 pin input of timer ENCA1.
<table>
<tr><th>PIC0REG3009</th><th>PIC0REG3008</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>ENCA1E1 input (ENCA1 timer)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>PIC0REG3007,
PIC0REG3006</td><td>Selects the ENCA1E0 pin input of timer ENCA1.
<table>
<tr><th>PIC0REG3007</th><th>PIC0REG3006</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>ENCA1E0 input (ENCA1 timer)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1</td><td>PIC0REG3001</td><td>Selects the TS0PUD signal of TSG20 and selects ENCA0 timer input.
0: ENCA0E1 pin input
1: Setting prohibited</td></tr>
<tr><td>0</td><td>PIC0REG3000</td><td>Selects the TS0PEC signal of TSG20 and selects ENCA0 timer input.
0: ENCA0E0 pin input
1: Setting prohibited</td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(2) Timer I/O Control Register 50 (PIC0REG50)

PIC0REG50 selects the timer input signal to control the two-phase encoder.

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

Address FF81 C0E0$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG506	PIC0REG505	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 26-45 PIC0REG50 Contents

Bit Position	Bit Name	Function
6 5	PIC0REG506, PIC0REG505	Selects the signal to be input to TS0OPCI0.

PIC0REG506	PIC0REG505	Input Signal
0	0	ENCA0TEQ1 input (ENCA0 timer)
0	1	ENCA1TEQ1 input (ENCA1 timer)
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(3) Timer I/O Control Register 51 (PIC0REG51)

PIC0REG51 selects the timer input signal to control the two-phase encoder.

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

Address FF81 C0E8$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG516	PIC0REG515	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 26-46 PIC0REG51 Contents

Bit Position	Bit Name	Function
6 5	PIC0REG516, PIC0REG515	Selects the signal to be input to TS1OPCI0.

PIC0REG516	PIC0REG515	Input Signal
0	0	ENCA0TEQ1 input (ENCA0 timer)
0	1	ENCA1TEQ1 input (ENCA1 timer)
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(4) Hi-Z Output Control Register 2 (PIC0HIZCEN2)

PIC0HIZCEN2 selects the input signal to control Hi-Z output.

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

Address FF81 C0DC$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN26	PIC0HIZ CEN25	0	PIC0HIZ CEN23	0	0	PIC0HIZ CEN20
R	R/W	R/W	R	R/W	R	R	R/W

Table 26-47 PIC0HIZCEN2 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN26	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN25	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
3	PIC0HIZCEN23	Enables or disables Hi-Z output control with the INTTSG20IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN20	Enables or disables Hi-Z output control with the ESO2 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(5) Hi-Z Output Control Register 3 (PIC0HIZCEN3)

PIC0HIZCEN3 selects the input signal to control Hi-Z output.

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

Address FF81 C0E4$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0HIZCEN36	PIC0HIZCEN35	PIC0HIZCEN34	0	0	0	PIC0HIZCEN30
R	R/W	R/W	R/W	R	R	R	R/W

Table 26-48 PIC0HIZCEN3 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN36	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN35	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
4	PIC0HIZCEN34	Enables or disables Hi-Z output control with the INTTSG21IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN30	Enables or disables Hi-Z output control with the ESO3 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

26.4.9.4 Example of Operation

The encoder count value at which the TSG2n output pattern is switched and encoder count value which defines an offset of the output pattern switching position (advance or retard) are set in ENCAnCCR0 and ENCAnCCR1 of the ENCAn timer, respectively. Using compare match interrupt (INTENCAnI1) of the ENCAn timer as a trigger of the pattern switch of timer TSG2n, the control of TSG2n output can be achieved.

In control method 2, the encoder counter is cleared by a compare match of ENCAnCCR0. It is not necessary to set the compare value for each interrupt, unlike in control method 1.

The ENCAnCCR1 value can be rewritten with the INTENCAnI1 interrupt. It is necessary to match the initial output pattern of TSG2n to the set value of ENCAnCCR0 before start because clear by encoder clear input is not performed. Switching between normal and reverse rotations of output patterns should be set with the TSnPSC bit in TSnOPT0.

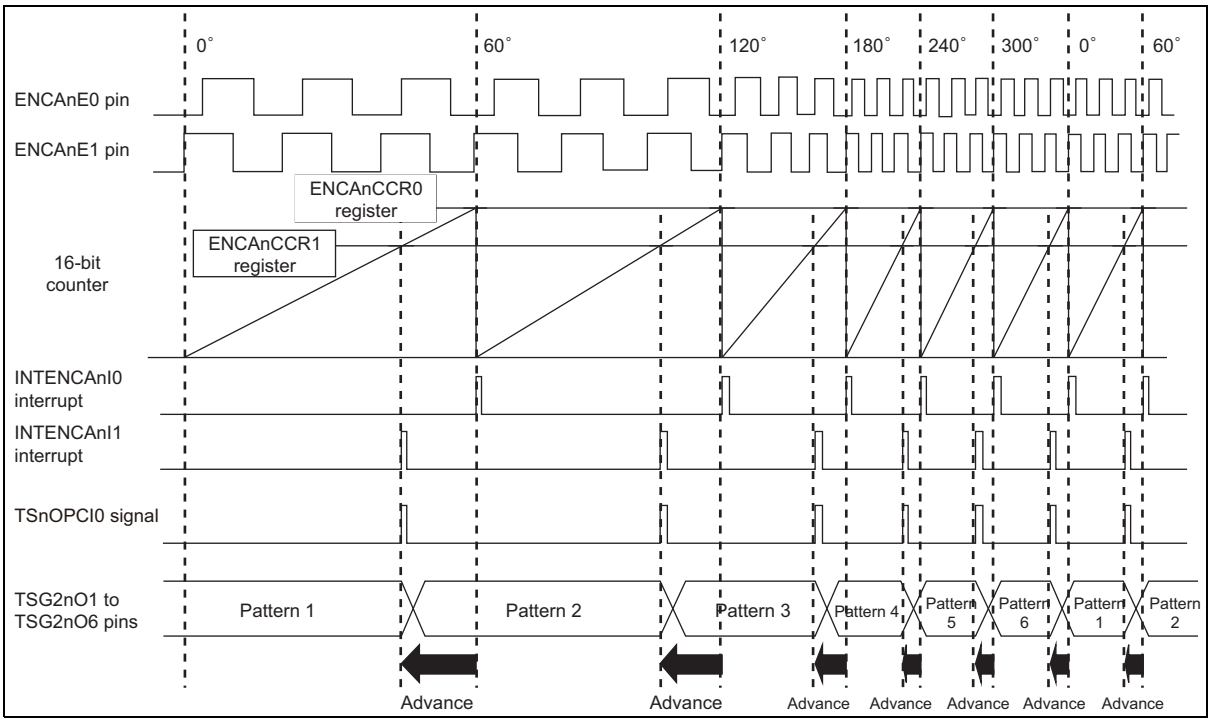


Figure 26-50 Operation Example of Control Method 2: at Advance

By advancing the output pattern switching ahead of the timing set in ENCAnCCR0, the TSG2n output pattern phase can be advanced.

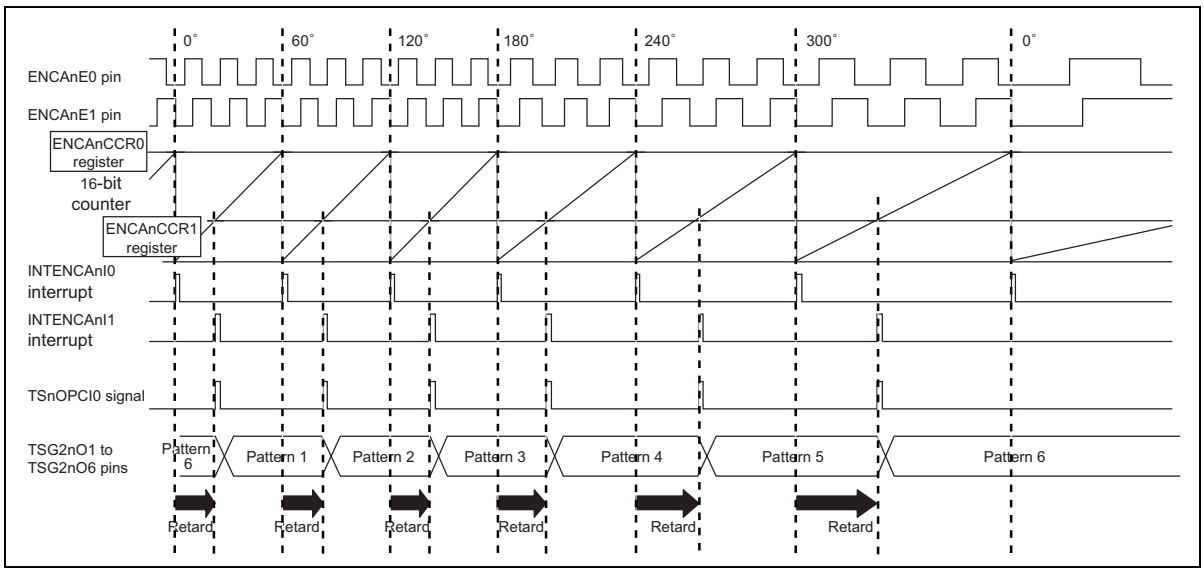


Figure 26-51 Operation Example of Control Method 2: at Retard

By delaying the output pattern switching behind the timing set in ENCAnCCR0, the TSG2n output pattern phase can be retarded.

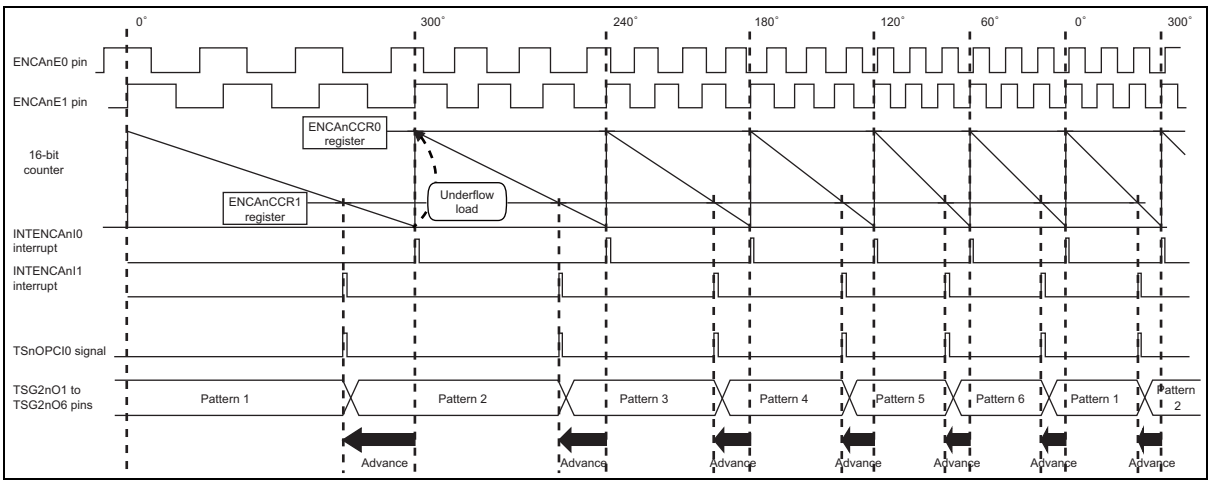


Figure 26-52 Operation Example of Control Method 2: at Down Count

26.4.9.5 Operation Flow

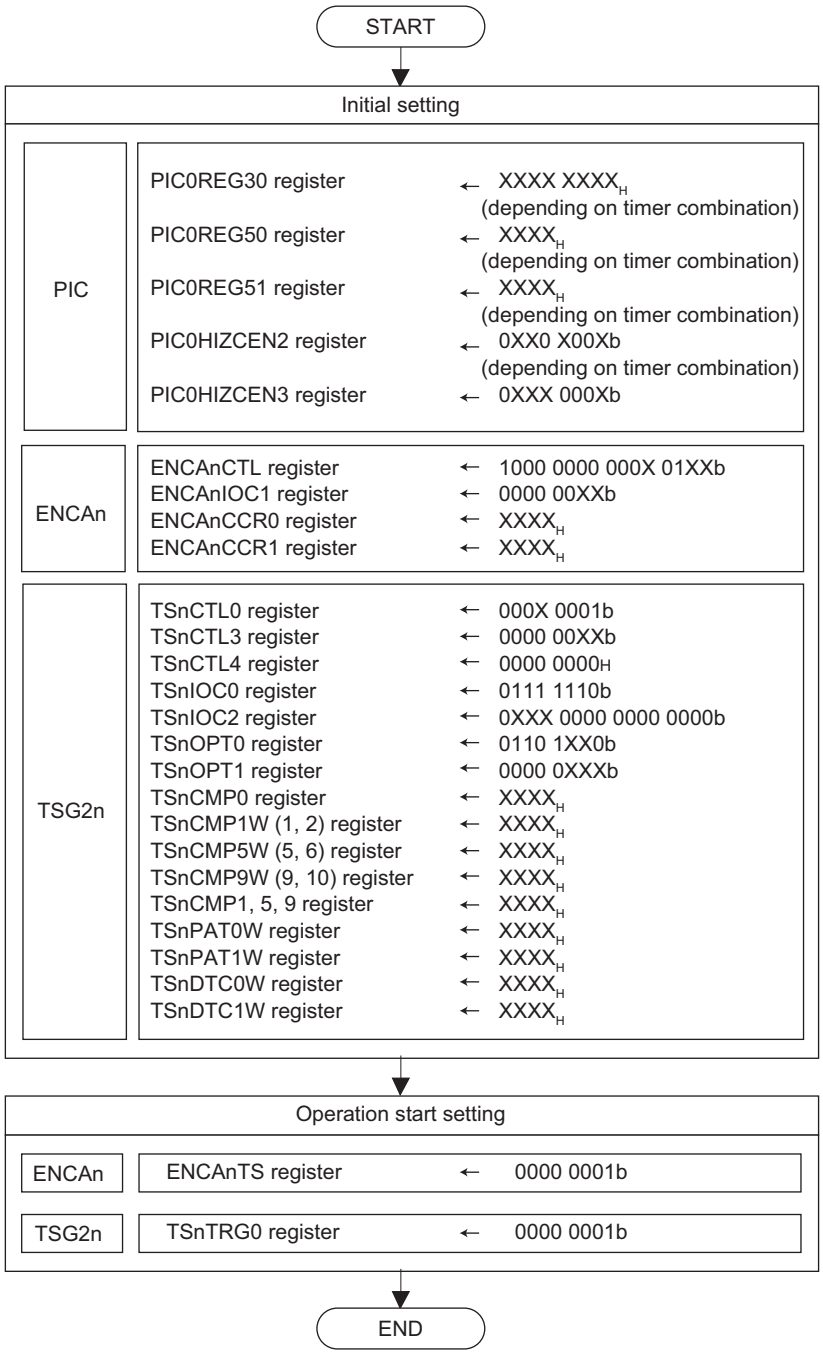


Figure 26-53 Initial Setting and Operation Start Flow

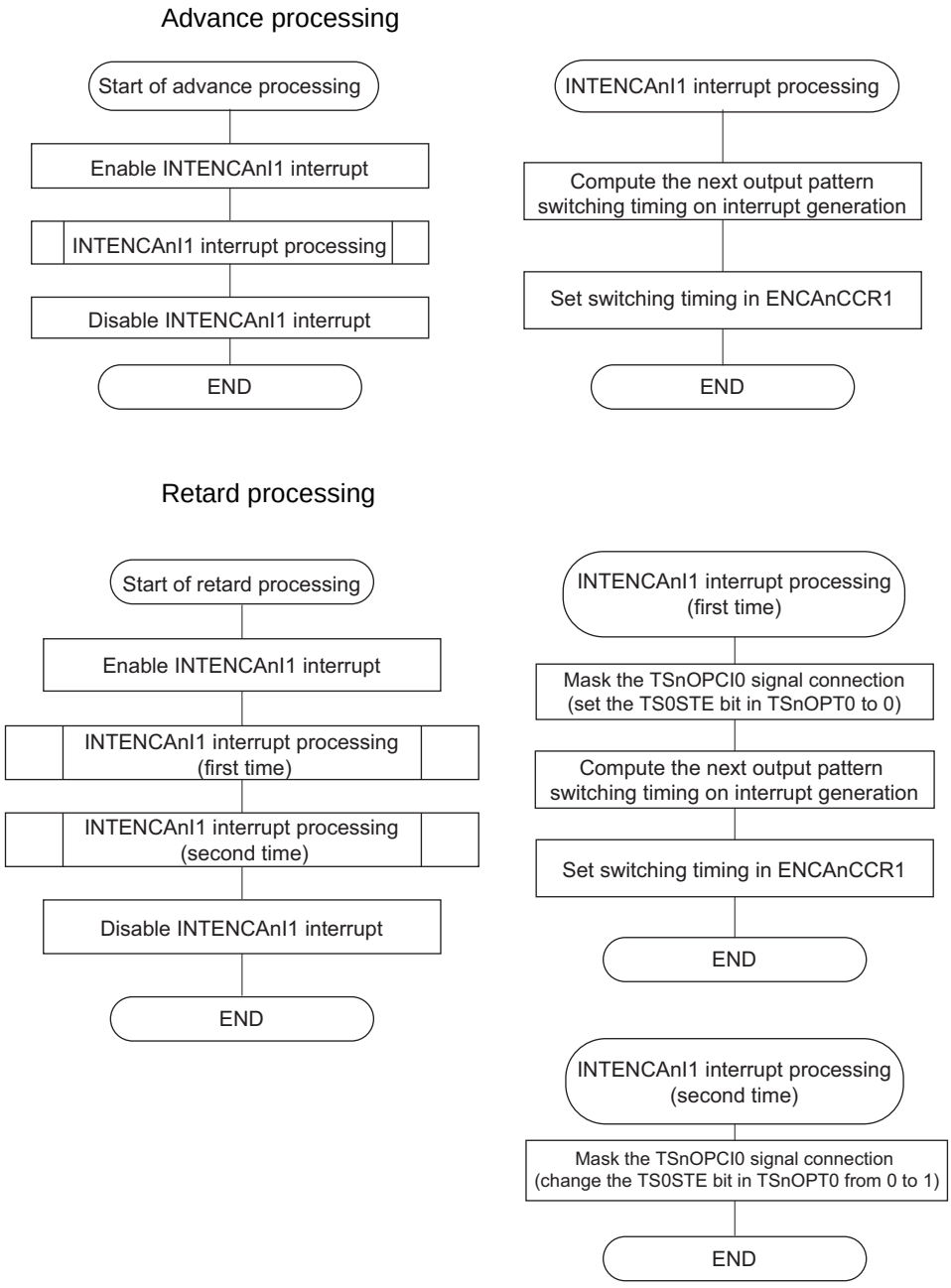


Figure 26-54 ENCAnCCR1 Rewrite Flow during Operation

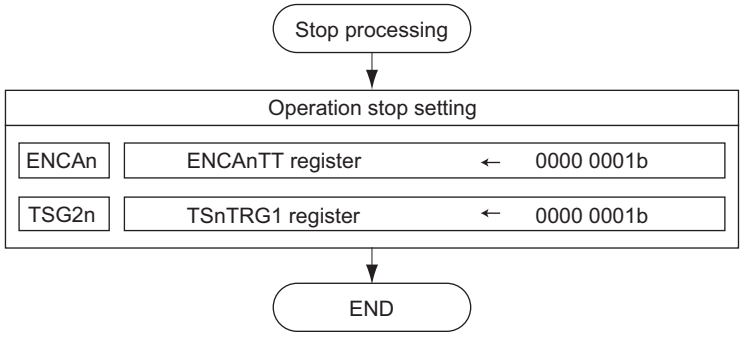


Figure 26-55 Operation Stop Flow

26.4.9.6 Function Setting

Timer register settings

Register bits not listed here are used with the initial values.

Table 26-49 ENCAn Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
ENCAn	ENCAnCTL	15	ENCAnCME	1	Enables or disables compare match interrupt detection mask.
		14	ENCAnMCS	0	Selects a cancelation trigger of compare match interrupt detection mask.
		9	ENCAnCRM1	0	Selects the ENCAnCCR1 function (capture/compare)
		8	ENCAnCRM0	0	Selects the ENCAnCCR0 function (capture/compare)
		7	ENCAnCTS	0	Selects trigger of capture operation of ENCAnCCR1
		4	ENCAnLDE	Don't care	Enables or disables reload operation when underflow is generated.
		3	ENCAnECM1	0	Enables or disables clearing of the counter on compare match of ENCAnCCR1.
		2	ENCAnECM0	1	Enables or disables clearing of the counter on compare match of ENCAnCCR0.
		1, 0	ENCAnUDS1, ENCAnUDS0	Don't care	Selects the counter up/down control by ENCAnE0/ENCAnE1.
	ENCAnIOC1	7	ENCAnSCE	0	Enables the special encoder clear.
		6	ENCAnZCL	0	Selects the clear level of Z phase for a special encoder clear.
		5	ENCAnBCL	0	Selects the clear level of B phase for a special encoder clear.
		4	ENCAnACL	0	Selects the clear level of A phase for a special encoder clear.
		3, 2	ENCAnECS1 ENCAnECS0	0	Selects encoder clear input (Z phase) edge.
		1, 0	ENCAnEIS1 ENCAnEIS0	Don't care	Selects encoder input (A and B phase) edge.
	ENCAnCCR0	—	—	Don't care	Compare register Sets a compare value.
	ENCAnCCR1	—	—	Don't care	Compare register Sets a compare value.
	ENCAnCNT	—	—	Don't care	Timer counter register

Table 26-50 TSG2n Setting (1/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCTL0	4	TSnDWD	Don't care	Selects the diag output pulse width.
		1, 0	TSnMD1, TSnMD0	1, 1	Selects operating mode (120-DC).
	TSnCTL3	1	TSnRIA	Don't care	Selects the reload timing of the compare register.
		0	TSnRMC	Don't care	Selects the transfer timing of the compare register.
	TSnCTL4	8	TSnPRE	Don't care	Enables or disables peak reload timing.
		7	TSnVRE	Don't care	Enables or disables valley reload timing.
		6	TSnPIE	0	Enables or disables generation of the peak interrupt.
		5	TSnVIE	0	Enables or disables generation of the valley interrupt.
		4 to 0	TSnRCC4, TSnRCC3, TSnRCC2, TSnRCC1, TSnRCC0	0	Sets the thinning out rate of reload timing and interrupt.
	TSnIOC0	6	TSnTOE6	1	Enables or disables rewriting of the TSnOL6 and TSnTO6 bits in TSnIOC2.
		5	TSnTOE5	1	Enables or disables rewriting of the TSnOL5 and TSnTO5 bits in TSnIOC2.
		4	TSnTOE4	1	Enables or disables rewriting of the TSnOL4 and TSnTO4 bits in TSnIOC2.
		3	TSnTOE3	1	Enables or disables rewriting of the TSnOL3 and TSnTO3 bits in TSnIOC2.
		2	TSnTOE2	1	Enables or disables rewriting of the TSnOL2 and TSnTO2 bits in TSnIOC2.
		1	TSnTOE1	1	Enables or disables rewriting of the TSnOL1 and TSnTO1 bits in TSnIOC2.
	TSnIOC2	14	TSnOL6	Don't care	Selects an active level of the TSG2nO6 output.
		13	TSnOL5	Don't care	Selects an active level of the TSG2nO5 output.
		12	TSnOL4	Don't care	Selects an active level of the TSG2nO4 output.
		11	TSnOL3	Don't care	Selects an active level of the TSG2nO3 output.
		10	TSnOL2	Don't care	Selects an active level of the TSG2nO2 output.
		9	TSnOL1	Don't care	Selects an active level of the TSG2nO1 output.
	TSnOPT0	6	TSnSOC	0	Selects the control of the timer output with software.
		5	TSnSTE	1	Enables or disables the control with the pattern output trigger.
		4	TSnPOT	0	Selects pattern output trigger. Selects either the output pattern switching by the external pattern input pin (TSG2nPTSI0 to TSG2nPTSI2) or output switching by the TSnOPCI1 and TSnOPCI0 rising edge as the pattern output trigger.
		3	TSnPSS	1	Selects the pattern output order switching method. The pattern output order is switched with TSnPSC.
		2	TSnIDC	Don't care	Selects the direction of the motor rotation.
		1	TSnPSC	Don't care	Selects the order of the timer output pattern during semi-automatic cruise control.
	TSnOPT1	2 to 0	TSnSPC2 to TSnSPC0	Don't care	Sets the timer output pattern in software output function mode.

Table 26-50 TSG2n Setting (2/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCMP0	—	—	Don't care	Compare register Sets the PWM period.
	TSnCMP1W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP5W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP9W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP 1, 5, 9	—	—	Don't care	Compare register Sets the compare value.
	TSnPAT0W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnPAT1W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnDTC0W	—	—	Don't care	Dead time set register Sets the dead time value.
	TSnDTC1W	—	—	Don't care	Dead time set register Sets the dead time value.

Table 26-51 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG30	9, 8	PIC0REG3009, PIC0REG3008	See the block diagram	Selects the ENCA1E1 pin input.
		7, 6	PIC0REG3007, PIC0REG3006	See the block diagram	Selects the ENCA1E0 pin input.
		1	PIC0REG3001	See the block diagram	Selects the ENCA0E1 pin input.
		0	PIC0REG3000	See the block diagram	Selects the ENCA0E0 pin input.
	PIC0REG50	6, 5	PIC0REG506, PIC0REG505	See the block diagram	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		4	PIC0REG504	See the block diagram	Selects the ENCA0TCSO/ENCA1TCSO signal.
	PIC0REG51	6, 5	PIC0REG516 PIC0REG515	See the block diagram	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		4	PIC0REG514	See the block diagram	Selects the ENCA0TCSO/ENCA1TCSO signal.
	PIC0HIZ CEN2	6	PIC0HIZCEN26	See the block diagram	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZCEN25	See the block diagram	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		3	PIC0HIZCEN23	See the block diagram	Enables or disables the INTTSG20IER interrupt as Hi-Z control signal input.
		0	PIC0HIZCEN20	See the block diagram	Enables or disables the ESO2 pin input as Hi-Z control signal input.
	PIC0HIZ CEN3	6	PIC0HIZCEN36	See the block diagram	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZCEN35	See the block diagram	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		4	PIC0HIZCEN34	See the block diagram	Enables or disables the INTTSG21IER interrupt as Hi-Z control signal input.
		0	PIC0HIZCEN30	See the block diagram	Enables or disables the ESO3 pin input as Hi-Z control signal input.

26.4.10 Three-Phase Pulse Input Control Function

26.4.10.1 Functional Overview

By combining three-phase pulse input (TSG2nPTSI2 to TSG2nPTSI0 pin inputs) and TAUAn offset trigger mode, variable phase control of TSG2n pattern output can be performed.

Table 26-52 Three-Phase Pulse Input Control Method

Control Method	Function
Fixed phase control 1 (pattern switching method)	Outputs a fixed pattern at a constant rotation angle.
Variable phase control (trigger switching method)	Varies the phase by arbitrary angle (time) up to ± 60 degrees with reference to the rotation angle and outputs the pattern.

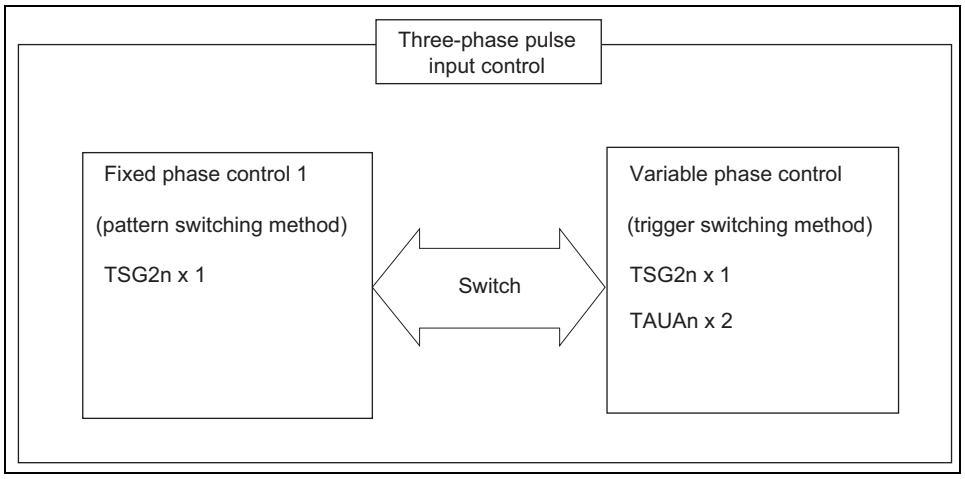


Figure 26-56 Example of Fixed Phase Control 1 (Pattern Switching Method) and Variable Phase Control

26.4.10.2 Configuration

Configuration/ Timer Function	TAUA	TSG2	TAPA
Timer configuration 1	TAUA0	TSG20	TAPA2
Timer configuration 2	TAUA1	TSG21	TAPA3

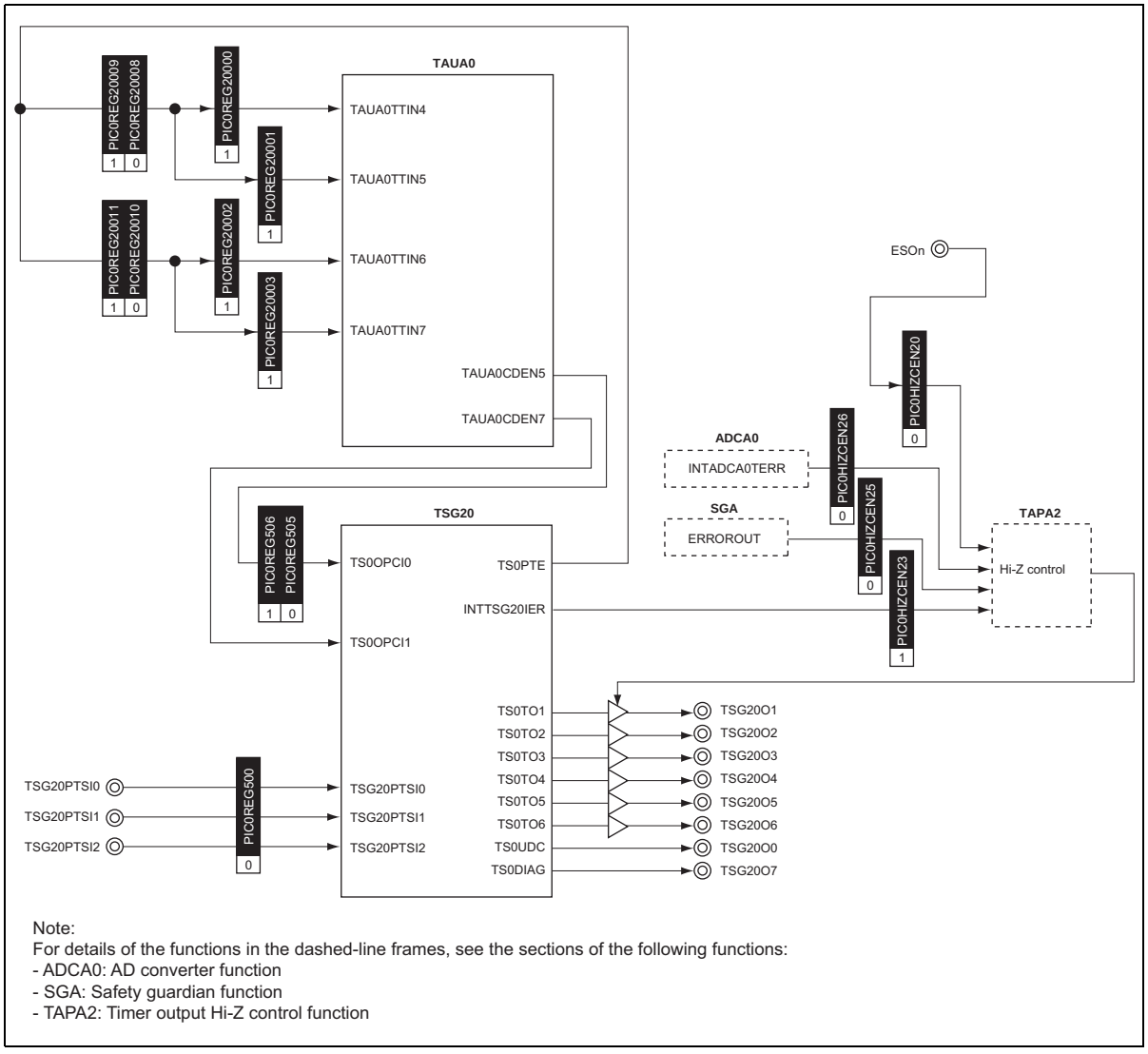


Figure 26-57 Block Diagram of Timer Configuration 1

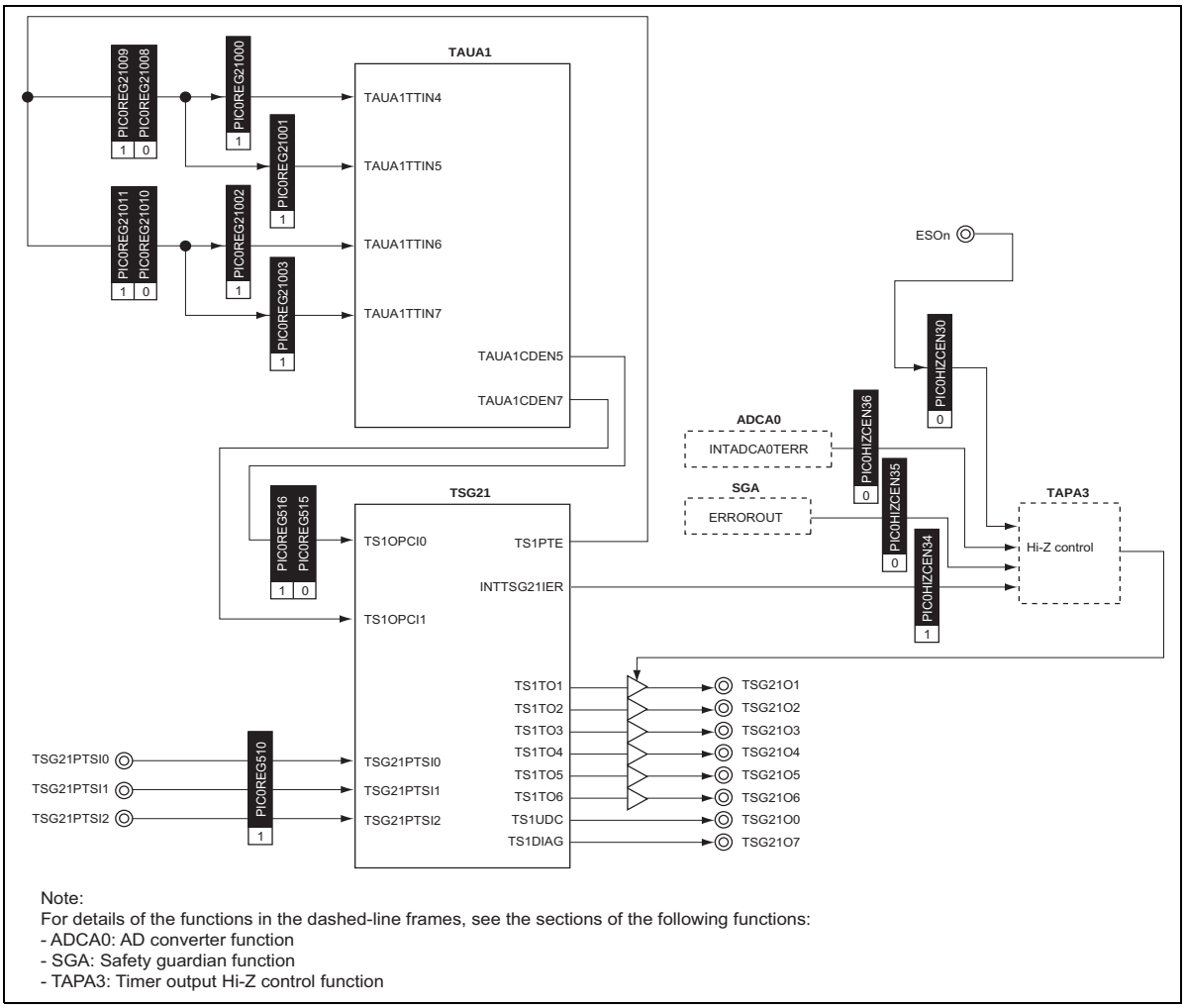


Figure 26-58 Block Diagram of Timer Configuration 2

26.4.10.3 Registers

(1) Timer I/O Control Register 200 (PIC0REG200)

PIC0REG200 is a 32-bit register that selects the timer input signal to control three-phase pulse input.

Access This register can be read/written in 32-bit units.

Address FF81 C08C$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	PIC0REG20011	PIC0REG20010	PIC0REG20009	PIC0REG20008
R	R	R	R	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
0	0	0	0	PIC0REG20003	PIC0REG20002	PIC0REG20001	PIC0REG20000
R	R	R	R	R/W	R/W	R/W	R/W

Table 26-53 PIC0REG200 Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11, 10</td><td>PIC0REG20011,
PIC0REG20010</td><td>Selects TS0PTE signal input of TSG20.
<table>
<tr><th>PIC0REG20011</th><th>PIC0REG20010</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>TS0PTE output</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>9, 8</td><td>PIC0REG20009,
PIC0REG20008</td><td>Selects TS0PTE signal input of TSG20.
<table>
<tr><th>PIC0REG20009</th><th>PIC0REG20008</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>TS0PTE output</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3</td><td>PIC0REG20003</td><td>Selects TS0PTE signal input of TSG20.
0: Setting prohibited
1: TS0PTE signal input of TSG20</td></tr>
<tr><td>2</td><td>PIC0REG20002</td><td>Selects TS0PTE signal input of TSG20.
0: Setting prohibited
1: TS0PTE signal input of TSG20</td></tr>
<tr><td>1</td><td>PIC0REG20001</td><td>Selects TS0PTE signal input of TSG20.
0: Setting prohibited
1: TS0PTE signal input of TSG20</td></tr>
<tr><td>0</td><td>PIC0REG20000</td><td>Selects TS0PTE signal input of TSG20.
0: Setting prohibited
1: TS0PTE signal input of TSG20</td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(2) Timer I/O Control Register 210 (PIC0REG210)

PIC0REG210 is a 32-bit register that selects the timer input signal to control three-phase pulse input.

Access This register can be read/written in 32-bit units.

Address FF81 C09C$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	PIC0REG21011	PIC0REG21010	PIC0REG21009	PIC0REG21008
R	R	R	R	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
0	0	0	0	PIC0REG21003	PIC0REG21002	PIC0REG21001	PIC0REG21000
R	R	R	R	R/W	R/W	R/W	R/W

Table 26-54 PIC0REG210 Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11, 10</td><td>PIC0REG21011,
PIC0REG21010</td><td>Selects TS1PTE signal input of TSG21.
<table>
<tr><th>PIC0REG21011</th><th>PIC0REG21010</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>TS1PTE output</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>9, 8</td><td>PIC0REG21009,
PIC0REG21008</td><td>Selects TS1PTE signal input of TSG21.
<table>
<tr><th>PIC0REG21009</th><th>PIC0REG21008</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>TS1PTE output</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3</td><td>PIC0REG21003</td><td>Selects TS1PTE signal input of TSG21.
0: Setting prohibited
1: TS1PTE signal input of TSG21</td></tr>
<tr><td>2</td><td>PIC0REG21002</td><td>Selects TS1PTE signal input of TSG21.
0: Setting prohibited
1: TS1PTE signal input of TSG21</td></tr>
<tr><td>1</td><td>PIC0REG21001</td><td>Selects TS1PTE signal input of TSG21.
0: Setting prohibited
1: TS1PTE signal input of TSG21</td></tr>
<tr><td>0</td><td>PIC0REG21000</td><td>Selects TS1PTE signal input of TSG21.
0: Setting prohibited
1: TS1PTE signal input of TSG21</td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(3) Timer I/O Control Register 50 (PIC0REG50)

PIC0REG50 is an 8-bit register that selects the timer input signal to control three-phase pulse input.

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

Address FF81 C0E0$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG506	PIC0REG505	0	0	0	0	PIC0REG500
R	R/W	R/W	R	R	R	R	R/W

Table 26-55 PIC0REG50 Contents

Bit Position	Bit Name	Function
6, 5	PIC0REG506, PIC0REG505	Selects TAUA0CDEN5 signal input of TAUA0. **PIC0REG506** / **PIC0REG505** / **Input Signal** 1 / 0 / TAUA0CDEN5 output Other than above / Setting prohibited
0	PIC0REG500	Selects TSG20PTSI0 to TSG20PTSI2 pin inputs. 0: TSG20PTSI0 to TSG20PTSI2 pins are selected. 1: Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(4) Timer I/O Control Register 51 (PIC0REG51)

PIC0REG51 is an 8-bit register that selects the timer input signal to control three-phase pulse input.

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

Address FF81 C0E8$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0REG516	PIC0REG515	0	0	0	0	PIC0REG510
R	R/W	R/W	R	R	R	R	R/W

Table 26-56 PIC0REG51 Contents

Bit Position	Bit Name	Function
6, 5	PIC0REG516, PIC0REG515	Selects TAUA1CDEN5 signal input of TAUA1.
0	PIC0REG510	Selects TSG21PTSI0 to TSG21PTSI2 pin signal inputs. 0: Setting prohibited 1: TSG21PTSI0 to TSG21PTSI2 pins are selected

PIC0REG516	PIC0REG515	Input Signal
1	0	TAUA1CDEN5 output
Other than above		Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(5) Hi-Z Output Control Register 2 (PIC0HIZCEN2)

PIC0HIZCEN2 selects the input signal to control Hi-Z output.

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

Address FF81 C0DC$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0HIZ CEN26	PIC0HIZ CEN25	0	PIC0HIZ CEN23	0	0	PIC0HIZ CEN20
R	R/W	R/W	R	R/W	R	R	R/W

Table 26-57 PIC0HIZCEN2 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN26	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN25	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
3	PIC0HIZCEN23	Enables or disables Hi-Z output control with the INTTSG20IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN20	Enables or disables Hi-Z output control with the ESO2 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(6) Hi-Z Output Control Register 3 (PIC0HIZCEN3)

PIC0HIZCEN3 selects the input signal to control Hi-Z output.

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

Address FF81 C0E4$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	PIC0HIZCEN36	PIC0HIZCEN35	PIC0HIZCEN34	0	0	0	PIC0HIZCEN30
R	R/W	R/W	R/W	R	R	R	R/W

Table 26-58 PIC0HIZCEN3 Contents

Bit Position	Bit Name	Function
6	PIC0HIZCEN36	Enables or disables Hi-Z output control with the INTADCA0TERR interrupt signal. 0: Disable 1: Enable
5	PIC0HIZCEN35	Enables or disables Hi-Z output control with the ERROROUT signal. 0: Disable 1: Enable
4	PIC0HIZCEN34	Enables or disables Hi-Z output control with the INTTSG21IER interrupt signal. 0: Disable 1: Enable
0	PIC0HIZCEN30	Enables or disables Hi-Z output control with the ESO3 pin input. 0: Disable 1: Enable

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

26.4.10.4 Example of Operation

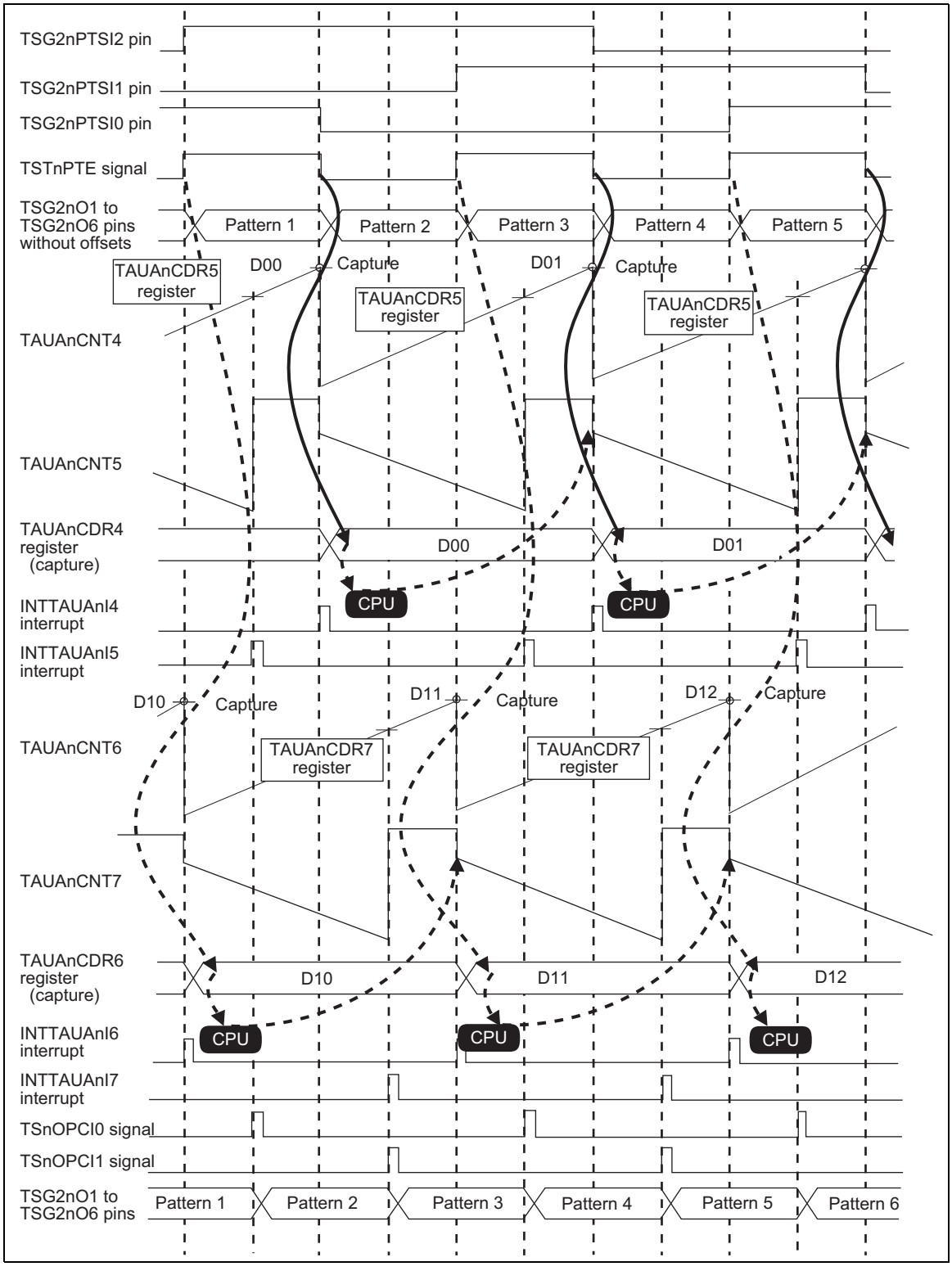


Figure 26-59 Example of Operation

The timing of TSG2n pattern output is set in the TAUAnCDR5 and TAUAnCDR5 (m+1) registers of TAUAn.

The TAUAn capture values (TAUAnCDR4 and TAUAnCDR4 (m+1)) indicate the timing of rising and falling edges of the TSG2nPTSI2 to TSG2nPTSI0 pins. The variable phase control can be achieved by clearing and starting the timer TAUAn at the rising and falling edge of TSG2nPTSI2 to TSG2nPTSI0 pins and changing the TSG2n pattern output timing, in other words, changing the TAUAnCDR5 and TAUAnCDR5 (m+1) register values.

Table 26-59 Relation between TAUAn Compare Register Setting and TSG2n Pattern Output Timing

TAUAn Compare Register Setting	TSG2n Pattern Output Timing
Compare value = 0000_H	Patterns are changed upon TSG2nPTSI0 to TSG2nPTSI2 pin edge detection (a maximum of one TAUAn counter clock delay is generated).
Compare value < capture value	Patterns are changed in advance of TSG2nPTSI0 to TSG2nPTSI2 pin edge detection.
Compare value > capture value	Setting prohibited.

26.4.10.5 Operation Flow

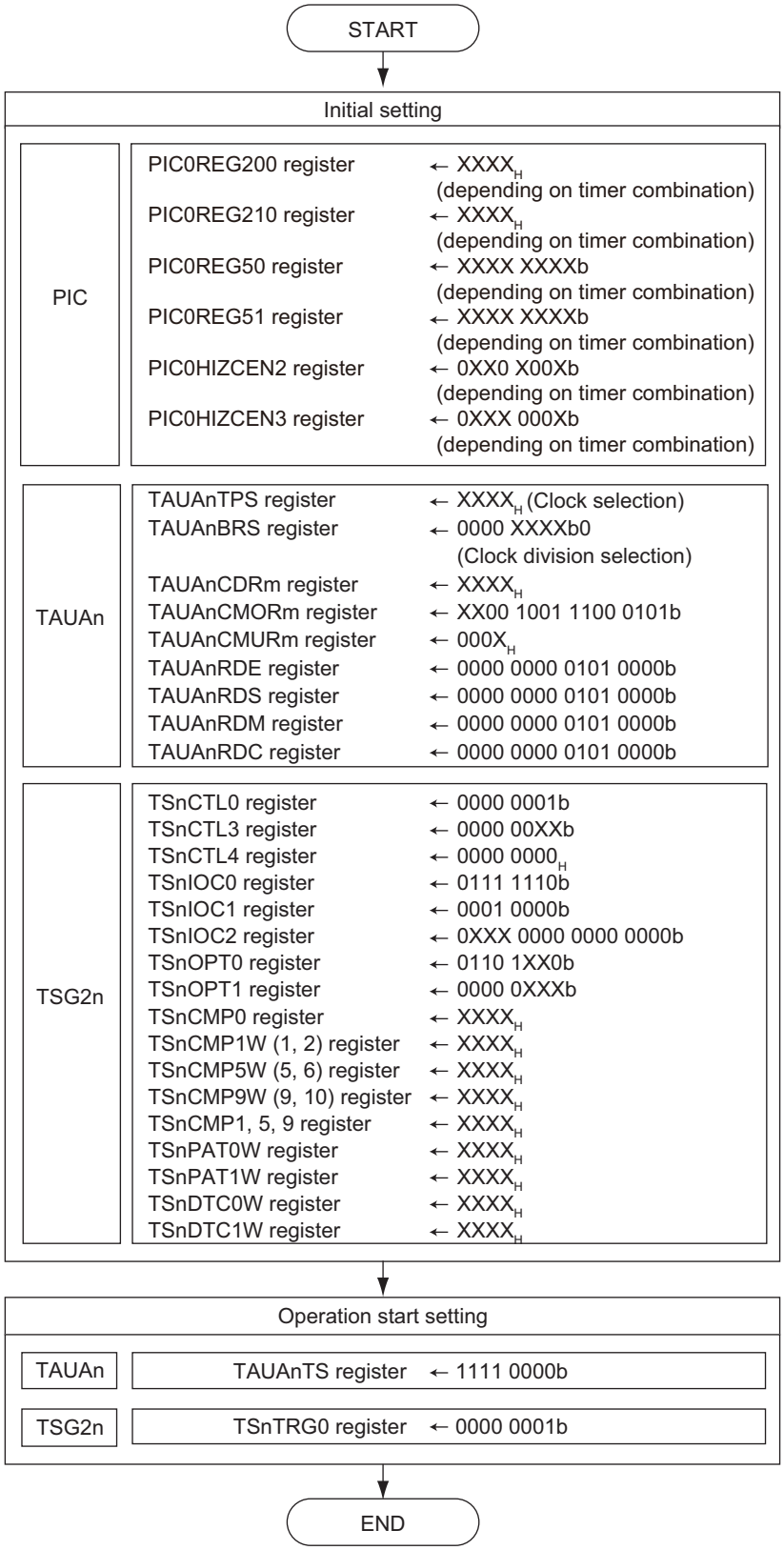


Figure 26-60 Initial Setting and Operation Starting Flow

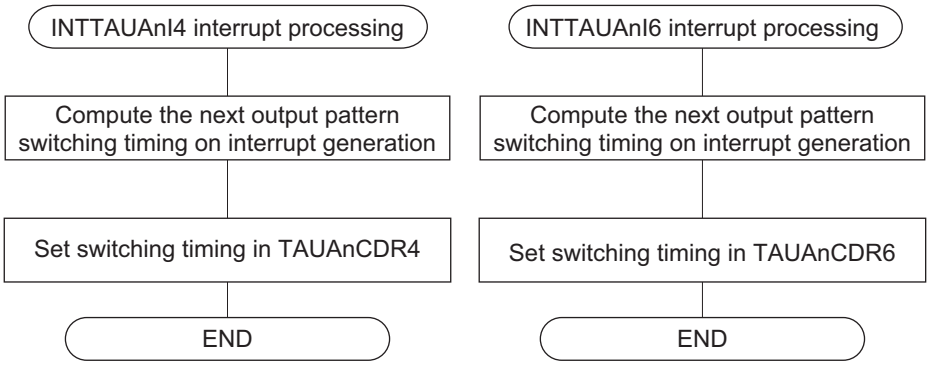


Figure 26-61 TAUAnCDR4 and TAUAnCDR6 Rewrite Flow during Operation

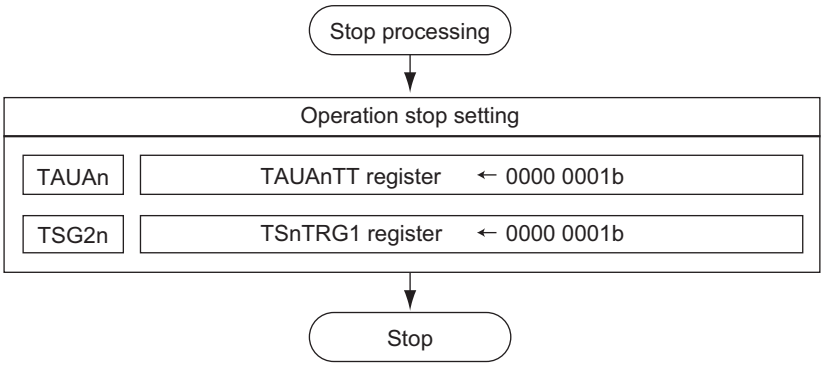


Figure 26-62 Operation Stop Flow (In HT-PWM Mode)

26.4.10.6 Function Setting

The following tables show the register settings.

Register bits not listed in the tables are used with the initial values.

Table 26-60 TSG2n Setting (1/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCTL0	1, 0	TSnMD1, TSnMD0	1, 1	Selects operating mode (120-DC).
	TSnCTL3	1	TSnRIA	Don't care	Selects the reload timing of the compare register.
		0	TSnRMC	Don't care	Selects the transfer timing of the compare register.
	TSnCTL4	8	TSnPRE	Don't care	Enables or disables peak reload timing.
		7	TSnVRE	Don't care	Enables or disables valley reload timing.
		6	TSnPIE	0	Enables or disables generation of the peak interrupt.
		5	TSnVIE	0	Enables or disables generation of the valley interrupt.
		4 to 0	TSnRCC4, TSnRCC3, TSnRCC2, TSnRCC1, TSnRCC0	0	Sets the thinning out rate of reload timing and interrupt.
	TSnIOC0	6	TSnTOE6	1	Enables or disables rewriting of the TSnOL6 and TSnTO6 bits in TSnIOC2.
		5	TSnTOE5	1	Enables or disables rewriting of the TSnOL5 and TSnTO5 bits in TSnIOC2.
		4	TSnTOE4	1	Enables or disables rewriting of the TSnOL4 and TSnTO4 bits in TSnIOC2.
		3	TSnTOE3	1	Enables or disables rewriting of the TSnOL3 and TSnTO3 bits in TSnIOC2.
		2	TSnTOE2	1	Enables or disables rewriting of the TSnOL2 and TSnTO2 bits in TSnIOC2.
		1	TSnTOE1	1	Enables or disables rewriting of the TSnOL1 and TSnTO1 bits in TSnIOC2.
	TSnIOC1	4	TSnPTS	1	Enables toggle signal output on edge detection of TSG2nPTSI0 to TSG2nPTSI2.
	TSnIOC2	14	TSnOL6	Don't care	Selects an active level of the TSG2nO6 output.
		13	TSnOL5	Don't care	Selects an active level of the TSG2nO5 output.
		12	TSnOL4	Don't care	Selects an active level of the TSG2nO4 output.
		11	TSnOL3	Don't care	Selects an active level of the TSG2nO3 output.
		10	TSnOL2	Don't care	Selects an active level of the TSG2nO2 output.
		9	TSnOL1	Don't care	Selects an active level of the TSG2nO1 output.
	TSnOPT0	6	TSnSOC	0	Selects the control of the timer output with software.
		5	TSnSTE	1	Enables or disables the control with the pattern output trigger.
		4	TSnPOT	0	Selects the pattern output trigger. Selects either the output pattern switching by the external pattern input pin (TSG2nPTSI0 to TSG2nPTSI2) or output switching by the TSnOPCI1 and TSnOPCI0 rising edge as the pattern output trigger.
		3	TSnPSS	1	Selects the pattern output order switching method. The pattern output order is switched with TSnPSC.
		2	TSnIDC	Don't care	Selects the direction of the motor rotation.
		1	TSnPSC	Don't care	Selects the order of the timer output pattern during semi-automatic cruise control.

Table 26-60 TSG2n Setting (2/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnOPT1	2 to 0	TSnSPC2 to TSnSPC0	Don't care	Sets the timer output pattern in software output function mode.
	TSnCMP0	—	—	Don't care	Compare register Sets the PWM period.
	TSnCMP1W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP5W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP9W	—	—	Don't care	Compare register Sets the compare value.
	TSnCMP 1, 5, 9	—	—	Don't care	Compare register Sets the compare value.
	TSnPAT0W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnPAT1W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnDTC0W	—	—	Don't care	Dead time set register Sets the dead time value.
	TSnDTC1W	—	—	Don't care	Dead time set register Sets the dead time value.

Table 26-61 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG200	11, 10	PIC0REG20011, PIC0REG20010	depending on timer combination	Selects TS0PET (TSG20) signal.
		9, 8	PIC0REG20009, PIC0REG20008	depending on timer combination	Selects TS0PET (TSG20) signal.
		3	PIC0REG20003	depending on timer combination	Selects TAUA0 external pin TAI007 input or internal signal input (TS0PTE signal of TSG20).
		2	PIC0REG20002	depending on timer combination	Selects TAUA0 external pin TAI006 input or internal signal input (TS0PTE signal of TSG20).
		1	PIC0REG20001	depending on timer combination	Selects TAUA0 external pin TAI005 input or internal signal input (TS0PTE signal of TSG20).
		0	PIC0REG20000	depending on timer combination	Selects TAUA0 external pin TAI004 input or internal signal input (TS0PTE signal of TSG20).
	PIC0REG210	11, 10	PIC0REG21011, PIC0REG21010	depending on timer combination	Selects TS1PET (TSG21) signal.
		9, 8	PIC0REG21009, PIC0REG21008	depending on timer combination	Selects TS1PET (TSG21) signal.
		3	PIC0REG21003	depending on timer combination	Selects TAUA1 external pin TAI107 input or internal signal input (TS1PTE signal of TSG21).
		2	PIC0REG21002	depending on timer combination	Selects TAUA1 external pin TAI106 input or internal signal input (TS1PTE signal of TSG21).
		1	PIC0REG21001	depending on timer combination	Selects TAUA1 external pin TAI105 input or internal signal input (TS1PTE signal of TSG21).
		0	PIC0REG21000	depending on timer combination	Selects TAUA1 external pin TAI104 input or internal signal input (TS1PTE signal of TSG21).
	PIC0REG050	6, 5	PIC0REG506, PIC0REG505	See the block diagram	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		0	PIC0REG500	1	Selects TSG20PTSI0 to TSG20PTSI2 pin input.
	PIC0REG051	6, 5	PIC0REG516, PIC0REG515	See the block diagram	Selects the ENCA0TEQ1/ENCA1TEQ1 signal.
		0	PIC0REG510	1	Selects TSG21PTSI0 to TSG21PTSI2 pin input.
PIC	PIC0HIZCEN2	6	PIC0HIZCEN26	See the block diagram	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZCEN25	See the block diagram	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		3	PIC0HIZCEN23	See the block diagram	Enables or disables the INTTSG20IER interrupt as Hi-Z control signal input.
		0	PIC0HIZCEN20	See the block diagram	Enables or disables the ESO2 pin input as Hi-Z control signal input.
	PIC0HIZCEN3	6	PIC0HIZCEN36	See the block diagram	Enables or disables the INTADC0TERR interrupt as Hi-Z control signal input.
		5	PIC0HIZCEN35	See the block diagram	Enables or disables the ERROROUT signal as Hi-Z control signal input.
		4	PIC0HIZCEN34	See the block diagram	Enables or disables the INTTSG21IER interrupt as Hi-Z control signal input.
		0	PIC0HIZCEN30	See the block diagram	Enables or disables the ESO3 pin input as Hi-Z control signal input.

Table 26-62 TAUAn Setting (Master CHm (m = 4, 6))

Function	Register	Bit Position	Bit Name	Set Value	Note
TAU	TAUAnTPS	15 to12	PRS33 to PRS30	Don't care	Selects the CK3 clock.
		11 to 8	PRS23 to PRS20	Don't care	Selects the CK2 clock.
		7 to 4	PRS13 to PRS10	Don't care	Selects the CK1 clock.
		3 to 0	PRS03 to PRS00	Don't care	Selects the CK0 clock.
	TAUAnBRS	7 to 0	BRS07 to BRS00	Don't care	Selects CK3 division value.
	TAUAnCDRm			Don't care	Capture/compare register of CHm The function is automatically changed between capture and compare depending on the operating mode.
	TAUAnCMORm	15, 14	CKS1, CKS0	Don't care	Selects the operating clock of CHm.
		13, 12	CCS1, CCS0	0, 0	Selects the counter clock of CHm.
		11	MAS	1	Selects independent or synchronous operation mode.
		10 to 8	STS2 to STS0	0, 0, 1	Selects external trigger signal.
		7, 6	COS1, COS0	1, 1	Controls the capture operation.
		4 to 0	MD4 to MD0	0, 0, 1, 0, 1	Sets operating mode (offset trigger mode)
	TAUAnCMURm	1, 0	TIS1, TIS0	Don't care (other than 0, 0)	Selects the effective edge of TIN input.
	TAUAnRDE	6	RDE06	0	Disables simultaneous rewrite control of CH06.
		4	RDE04	0	Disables simultaneous rewrite control of CH04.
	TAUAnRDS	6	RDS06	0	Disables simultaneous rewrite control of CH06 through another upper CH.
		4	RDS04	0	Disables simultaneous rewrite control of CH04 through another upper CH.
	TAUAnRDM	6	RDM06	0	Selects CH06 simultaneous rewrite control timing from start of Master-CH and start of triangle wave period.
		4	RDM04	0	Selects CH04 simultaneous rewrite control timing from start of Master-CH and start of triangle wave period.
	TAUAnRDC	6	RDC06	0	Disables operation as simultaneous rewrite trigger generating CH for CH06.
		4	RDC04	0	Disables operation as simultaneous rewrite trigger generating CH for CH04.

Table 26-63 TAUAn Setting (Slave CHm (m = 5, 7))

Function	Register	Bit Position	Bit Name	Set Value	Note
TAU	TAUAnTPS	15 to 12	PRS33 to PRS30	Same as master setting	Selects the CK3 clock.
		11 to 8	PRS23 to PRS20		Selects the CK2 clock.
		7 to 4	PRS13 to PRS10		Selects the CK1 clock.
		3 to 0	PRS03 to PRS00		Selects the CK0 clock.
	TAUAnBRS	7 to 0	BRS07 to BRS00		Selects CK3 division value.
	TAUAnCDRm			Don't care	Capture/compare register of CHm The function is automatically changed between capture and compare depending on the operating mode.
	TAUAn CMORm	15, 14	CKS1, CKS0	Don't care	Selects the operating clock of CHm.
		13, 12	CCS1, CCS0	0, 0	Selects the counter clock of CHm.
		11	MAS	0	Selects independent or synchronous operation mode.
		10 to 8	STS2 to STS0	1, 0, 0	Selects external trigger signal.
		7, 6	COS1, COS0	0, 0	Controls capture function.
		4 to 0	MD4 to MD0	0, 1, 0, 0, 1	Sets operating mode (offset trigger mode)
	TAUAn CMURm	1, 0	TIS1, TIS0	0, 0	Selects the effective edge of TIN input.
	TAUAnRDE	7	RDE07	0	Disables simultaneous rewrite control of CH07.
		5	RDE05	0	Disables simultaneous rewrite control of CH05.
	TAUAnRDS	7	RDS07	0	Disables simultaneous rewrite control of CH07 through another upper CH.
		5	RDS05	0	Disables simultaneous rewrite control of CH05 through another upper CH.
	TAUAnRDM	7	RDM07	0	Selects CH07 simultaneous rewrite control timing from start of Master-CH and start of triangle wave period.
		5	RDM05	0	Selects CH05 simultaneous rewrite control timing from start of Master-CH and start of triangle wave period.
	TAUAnRDC	7	RDC07	0	Disables operation as simultaneous rewrite trigger generating CH for CH07.
		5	RDC05	0	Disables operation as simultaneous rewrite trigger generating CH for CH05.

26.4.11 Three-Phase Encoder Control Function

26.4.11.1 Functional Overview

This function allows connecting the signal input to the TSG2nPTSI0 to TSG2nPTSI2 pins to the ENCAn timer to perform count operation.

26.4.11.2 Configuration

Configuration/ Timer Function	ENCA	TSG2
Timer configuration 1	ENCA0	TSG20
Timer configuration 2	ENCA1	TSG21

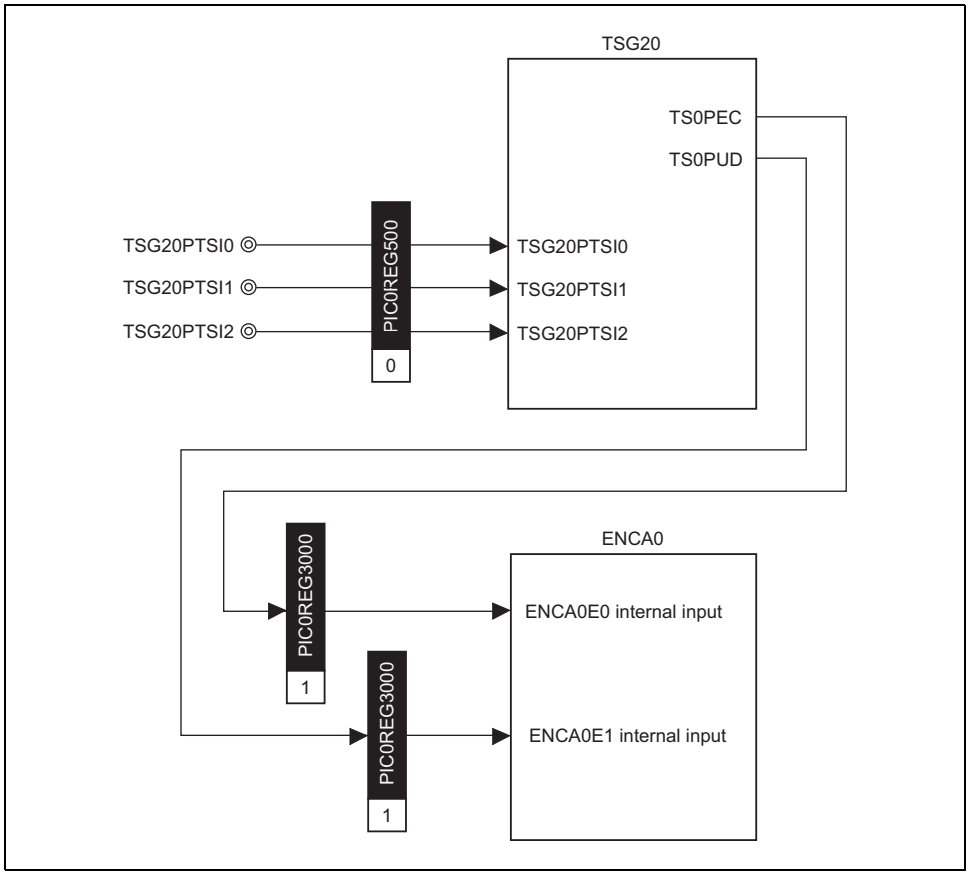


Figure 26-63 Block Diagram of Timer Configuration 1

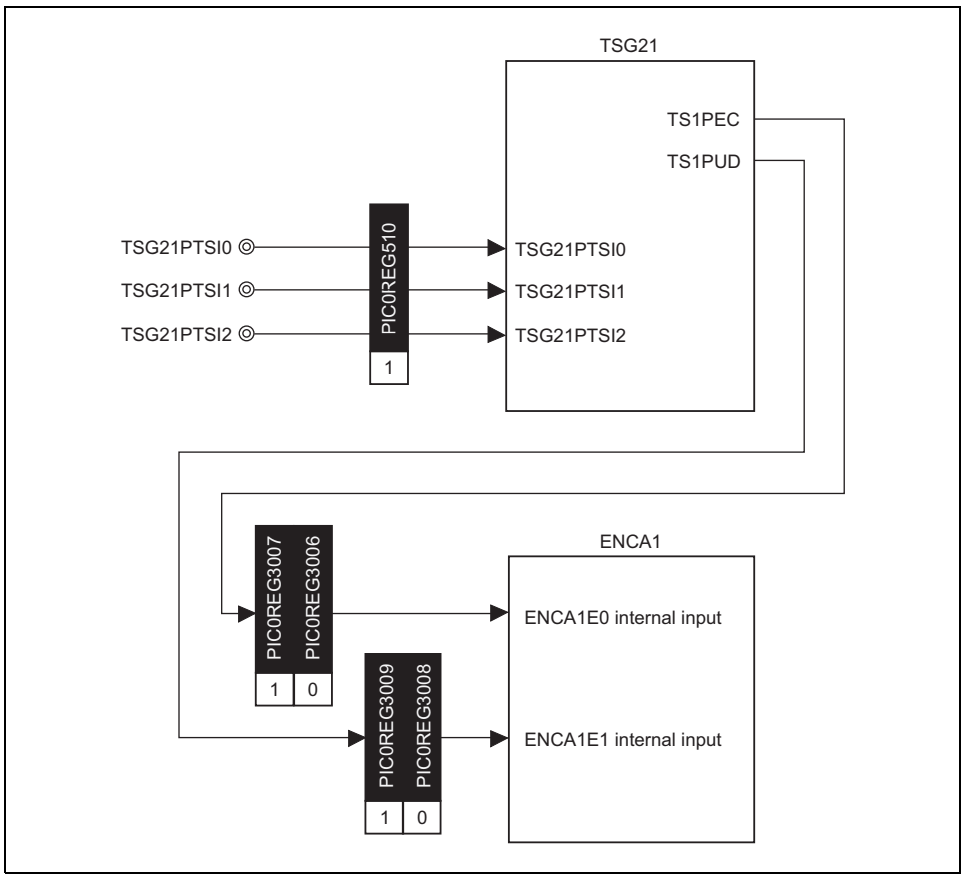


Figure 26-64 Block Diagram of Timer Configuration 2

26.4.11.3 Registers

(1) Timer I/O Control Register 30 (PIC0REG30)

PIC0REG30 is a 32-bit register that selects the timer input signal to control the three-phase encoder.

Access This register can be read/written in 32-bit units.

Address FF81 C0BC$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PIC0REG3009	PIC0REG3008
R	R	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
PIC0REG3007	PIC0REG3006	0	0	0	0	PIC0REG3001	PIC0REG3000
R/W	R/W	R	R	R	R	R/W	R/W

Table 26-64 PIC0REG30 Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>PIC0REG3009,
PIC0REG3008</td><td>Outputs the TS1PUD signal of TSG21 and selects the ENCA1 timer input.
<table>
<tr><th>PIC0REG3009</th><th>PIC0REG3008</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>ENCA1E1 input (ENCA1 timer)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>PIC0REG3007,
PIC0REG3006</td><td>Selects the TS1PEC signal of TSG21 and selects ENCA1 timer input.
<table>
<tr><th>PIC0REG3007</th><th>PIC0REG3006</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>ENCA1E0 input (ENCA1 timer)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1</td><td>PIC0REG3001</td><td>Selects the TS0PUD signal of TSG20 and selects ENCA0 timer input.
0: Setting prohibited
1: TS0PUD signal input of TSG20</td></tr>
<tr><td>0</td><td>PIC0REG3000</td><td>Selects the TS0PUD signal of TSG20 and selects ENCA0 timer input.
0: Setting prohibited
1: TS0PUD signal input of TSG20</td></tr>
</table>

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(2) Timer I/O Control Register 50 (PIC0REG50)

PIC0REG50 is an 8-bit register that selects the timer input signal to control the three-phase encoder.

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

Address FF81 C0E0$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PIC0REG500
R	R	R	R	R	R	R	R/W

Table 26-65 PIC0REG50 Contents

Bit Position	Bit Name	Function
0	PIC0REG500	Selects TSG20PTSI0 to TSG20PTSI2 pin input. 0: TSG20PTSI0 to TSG20PTSI2 pins are selected. 1: Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

(3) Timer I/O Control Register 51 (PIC0REG51)

PIC0REG51 is an 8-bit register that selects the timer input signal to control the three-phase encoder.

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

Address FF81 C0E8$_H$

Initial value 00$_H$

This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PIC0REG510
R	R	R	R	R	R	R	R/W

Table 26-66 PIC0REG51 Contents

Bit Position	Bit Name	Function
0	PIC0REG510	Selects TSG21PTSI0 to TSG21PTSI2 pin signal to be input. 0: TSG21PTSI0 to TSG21PTSI2 pins are selected. 1: Setting prohibited

Caution The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

26.4.11.4 Example of Operation

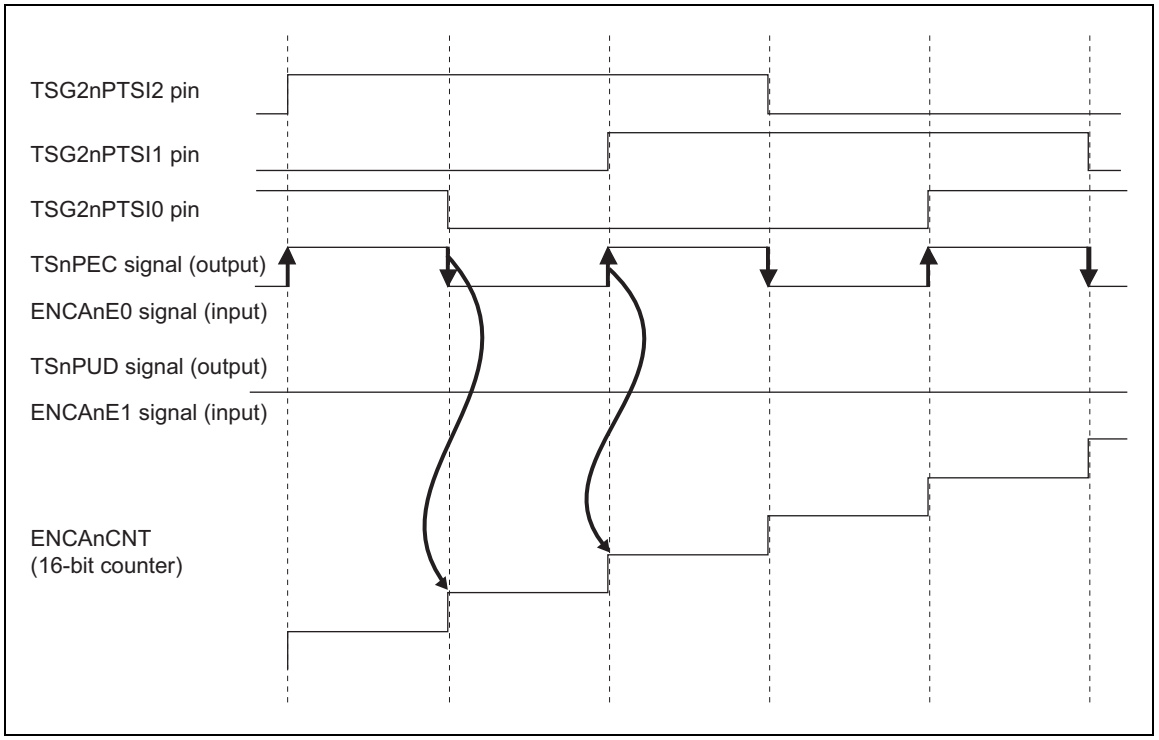


Figure 26-65 Operation Example of Three-Phase Encoder Function

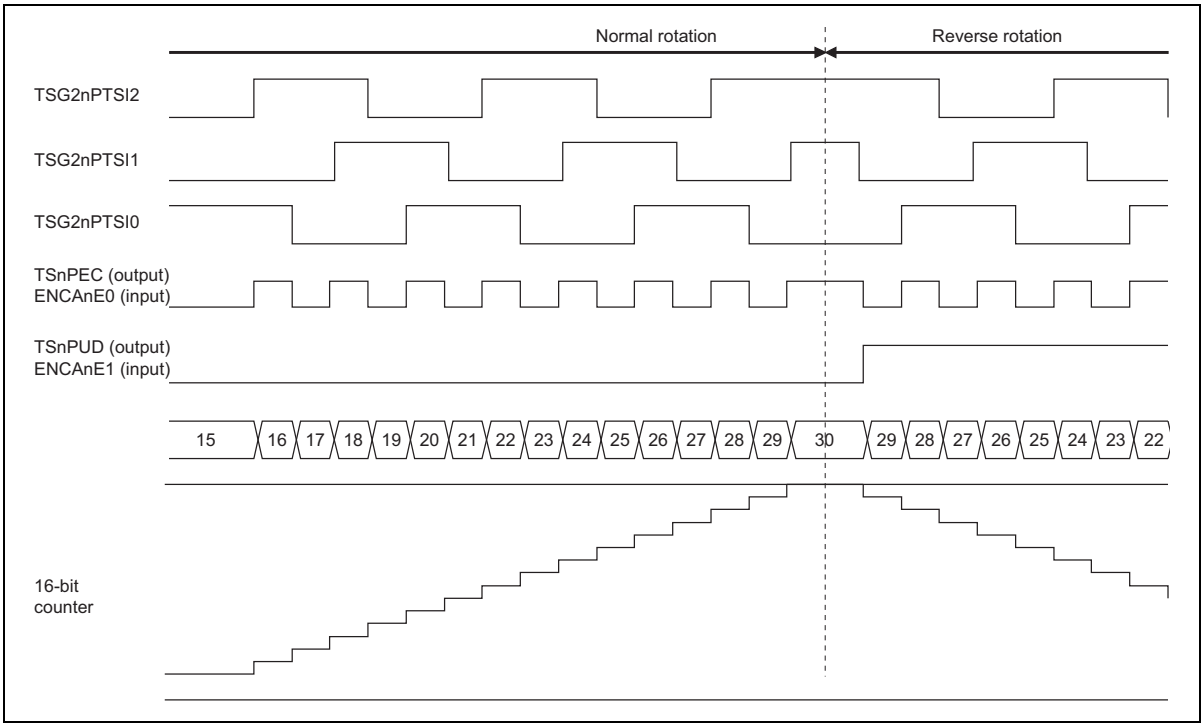


Figure 26-66 Example of Normal Rotation/Reverse Rotation using Three-Phase Encoder Function

Caution If noise is generated on the TSG2nPTSI2 to TSG2nPTSI0 pins, the 16-bit counter (ENCAnCNT) erroneously counts the noise edges.

26.4.11.5 Operation Flow

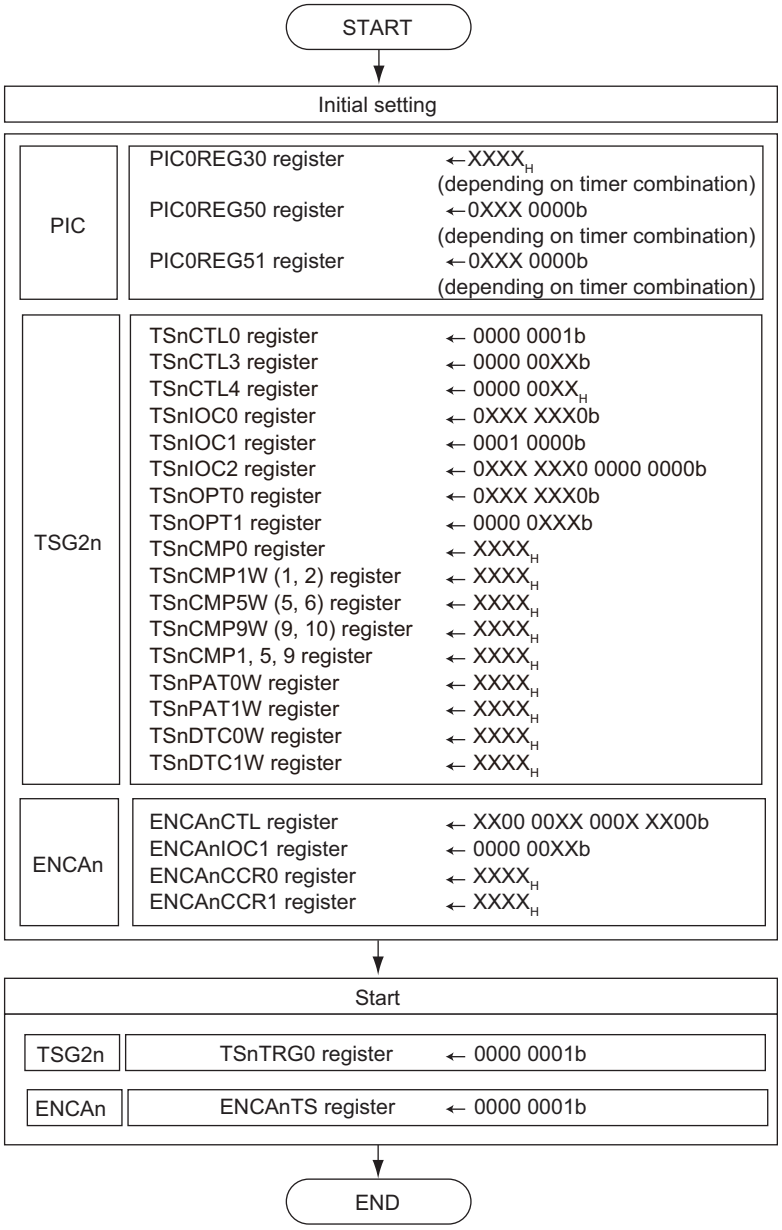


Figure 26-67 Initial Setting and Operation Start Flow

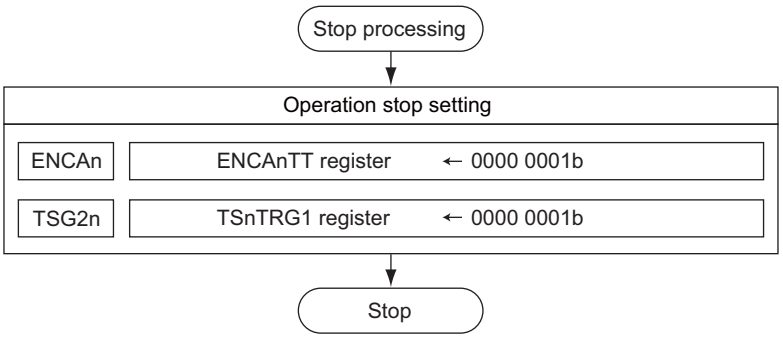


Figure 26-68 Operation Stop Flow

26.4.11.6 Function Setting

The following tables show the register settings.

Register bits not listed in the tables are used with the initial values.

Table 26-67 ENCAn Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
ENCAn	ENCAnCTL	15	ENCAnCME	Don't Care	Enables or disables compare match interrupt detection mask.
		14	ENCAnMCS	Don't Care	Selects a cancelation trigger of compare match interrupt detection mask.
		9	ENCAnCRM1	Don't Care	Selects the ENCAnCCR1 function (capture/compare)
		8	ENCAnCRM0	Don't Care	Selects the ENCAnCCR0 function (capture/compare)
		7	ENCAnCTS	0	Selects trigger of capture operation of ENCAnCCR1
		4	ENCAnLDE	Don't Care	Enables or disables reload operation when underflow is generated.
		3	ENCAnECM1	Don't Care	Enables or disables clearing of the counter on compare match of ENCAnCCR1.
		2	ENCAnECM0	Don't Care	Enables or disables clearing of the counter on compare match of ENCAnCCR0.
		1, 0	ENCAnUDS1, ENCAnUDS0	0, 0	Selects the counter up/down control by ENCAnE0/ ENCAnE1.
	ENCAnIOC1	7	ENCAnSCE	0	Enables the special encoder clear.
		6	ENCAnZCL	0	Selects the clear level of Z phase for a special encoder clear.
		5	ENCAnBCL	0	Selects the clear level of B phase for a special encoder clear.
		4	ENCAnACL	0	Selects the clear level of A phase for a special encoder clear.
		3, 2	ENCAnECS1 ENCAnECS0	0, 0	Selects encoder clear input (Z phase) edge.
		1, 0	ENCAnEIS1 ENCAnEIS0	Don't care (other than 0, 0)	Selects encoder input (A and B phase) edge.
	ENCAnCCR0	—	—	Don't Care	Compare register Sets a compare value.
	ENCAnCCR1	—	—	Don't Care	Compare register Sets a compare value.
	ENCAnCNT	—	—	Don't Care	Timer counter register

Table 26-68 TSG2n Setting (1/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCTL0	1, 0	TSnMD1, TSnMD0	0, 1	Selects operating mode (HT-PWM).
	TSnCTL3	1	TSnRIA	Don't care	Selects the reload timing of the compare register.
		0	TSnRMC	Don't care	Selects the transfer timing of the compare register.
	TSnCTL4	8	TSnPRE	Don't care	Enables or disables peak reload timing.
		7	TSnVRE	Don't care	Enables or disables valley reload timing.
		6	TSnPIE	Don't care	Enables or disables generation of the peak interrupt.
		5	TSnVIE	Don't care	Enables or disables generation of the valley interrupt.
		4 to 0	TSnRCC4, TSnRCC3, TSnRCC2, TSnRCC1, TSnRCC0	Don't care	Sets the thinning out rate of reload timing and interrupt.
	TSnIOC0	6	TSnTOE6	Don't care	Enables or disables rewrite of the TSnOL6 and TSnTO6 bits in TSnIOC2.
		5	TSnTOE5	Don't care	Enables or disables rewrite of the TSnOL5 and TSnTO5 bits in TSnIOC2.
		4	TSnTOE4	Don't care	Enables or disables rewrite of the TSnOL4 and TSnTO4 bits in TSnIOC2.
		3	TSnTOE3	Don't care	Enables or disables rewrite of the TSnOL3 and TSnTO3 bits in TSnIOC2.
		2	TSnTOE2	Don't care	Enables or disables rewrite of the TSnOL2 and TSnTO2 bits in TSnIOC2.
		1	TSnTOE1	Don't care	Enables or disables rewrite of the TSnOL1 and TSnTO1 bits in TSnIOC2.
	TSnIOC1	4	TSnPTS	1	Enables toggle signal output on edge detection of TSG2nPTSI0 to TSG2nPTSI2.
	TSnIOC2	14	TSnOL6	Don't care	Selects an active level of the TSG2nO6 output.
		13	TSnOL5	Don't care	Selects an active level of the TSG2nO5 output.
		12	TSnOL4	Don't care	Selects an active level of the TSG2nO4 output.
		11	TSnOL3	Don't care	Selects an active level of the TSG2nO3 output.
		10	TSnOL2	Don't care	Selects an active level of the TSG2nO2 output.
		9	TSnOL1	Don't care	Selects an active level of the TSG2nO1 output.
	TSnOPT0	6	TSnSOC	Don't care	Selects the control of the timer output with software.
		5	TSnSTE	Don't care	Enables or disables the control with the pattern output trigger.
		4	TSnPOT	Don't care	Selects the pattern output trigger. Selects either the output pattern switching by the external pattern input pin (TSG2nPTSI0 to TSG2nPTSI2) or output switching by the TSnOPCI1 and TSnOPCI0 rising edge as the pattern output trigger.
		3	TSnPSS	Don't care	Selects the pattern output order switching method. The pattern output order is switched with TSnPSC.
		2	TSnIDC	Don't care	Selects the direction of the motor rotation.
		1	TSnPSC	Don't care	Selects the order of the timer output pattern during semi-automatic cruise control.
	TSnOPT1	2 to 0	TSnSPC2 to TSnSPC0	Don't care	Sets the timer output pattern in software output function mode.

Table 26-68 TSG2n Setting (2/2)

Function	Register	Bit Position	Bit Name	Set Value	Note
TSG2n	TSnCMP0	—	—	Don't care	Compare register Sets the PWM period.
	TSnCMP1W	—	—	Don't care	Compare register Sets a compare value.
	TSnCMP5W	—	—	Don't care	Compare register Sets a compare value.
	TSnCMP9W	—	—	Don't care	Compare register Sets a compare value.
	TSnPAT0W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnPAT1W	—	—	Don't care	Pattern register Sets the output pattern.
	TSnDTC0W	—	—	Don't care	Dead time set register Sets the dead time value.
	TSnDTC1W	—	—	Don't care	Dead time set register Sets the dead time value.

Table 26-69 PIC Setting

Function	Register	Bit Position	Bit Name	Set Value	Note
PIC	PIC0REG030	9, 8	PIC0REG 3009, PIC0REG 3008	See the block diagram	Outputs the TS1PUD signal of TSG21 and selects the ENCA1E1 input of ENCA1 timer.
		7, 6	PIC0REG 3007, PIC0REG 3006	See the block diagram	Selects the TS1PEC signal of TSG21 and selects the ENCA1E0 input of ENCA1 timer.
		1	PIC0REG 3001	See the block diagram	Outputs the TS0PUD signal of TSG20 and selects the ENCA0E1 input of ENCA0 timer.
		0	PIC0REG 3000	See the block diagram	Selects the TS0PEC signal of TSG20 and selects the ENCA0E0 input of ENCA0 timer.
	PIC0REG050	0	PIC0REG500	See the block diagram	Selects TSG20PTSI0 to TSG20PTSI2 pin input.
	PIC0REG051	0	PIC0REG510	See the block diagram	Selects TSG21PTSI0 to TSG21PTSI2 pin input.

26.4.12 CAN Time Stamp Function

26.4.12.1 Functional Overview

The time stamp function can be set for message reception by combining the CAN controller and CH1 or CH3 of TAUJ0.

CH1 or CH3 of TAUJ0 captures a timer value according to the TSOUT signal that is output from the CAN controller on reception of a data frame. The CPU reads the captured value and obtains the time when the capture event occurred, in other words, the time stamp of the message received via the CAN bus.

26.4.12.2 Configuration

Configuration/ Timer Function	CAN	TAUJ
Configuration 1	CAN0	TAUJ0 CH1
Configuration 2	CAN1	TAUJ0 CH3

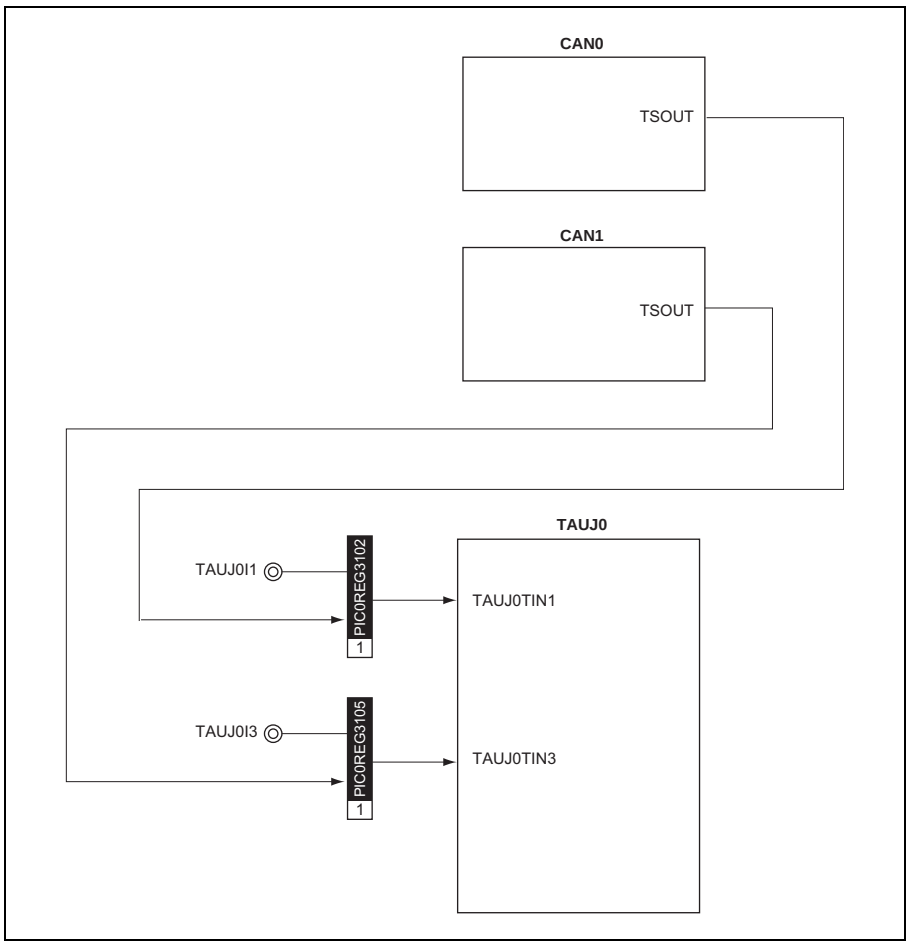


Figure 26-69 Block Diagram of CAN Time Stamp Function

26.4.12.3 Registers

(1) Timer I/O Control Register 31 (PIC0REG31)

PIC0REG31 is a 32-bit register that selects timer input signals.

Access This register can be read/written in 32-bit units.

Address FF81 C0C0$_H$

Initial value 0000 0000$_H$

This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	PIC0REG3105	PIC0REG3104	0	PIC0REG3102	PIC0REG3101	0
R	R	R/W	R/W	R	R/W	R/W	R

Table 26-70 PIC0REG31 Contents

Bit Position	Bit Name	Function
5, 4	PIC0REG3105, PIC0REG3104	Selects the input signal of CH3 of TAUJ0.
2, 1	PIC0REG3102, PIC0REG3101	Selects the input signal of CH1 of TAUJ0.

Selects the input signal of CH3 of TAUJ0.

PIC0REG3105	PIC0REG3104	Input Signal
0	0	TAUJ0TIN3 input
1	0	Trigger signal for CAN1 time stamp (TSOUT signal)
Other than above		Setting prohibited

Selects the input signal of CH1 of TAUJ0.

PIC0REG3102	PIC0REG3101	Input Signal
0	0	TAUJ0TIN1 input
1	0	Trigger signal for CAN0 time stamp (TSOUT signal)
Other than above		Setting prohibited

Caution 1. The bit might be set to 0 by other timer connection functions. In that case, apply the bit definition of the corresponding timer connection function.

Caution 2. Set the CAN0/CAN1 time stamp trigger signal (TSOUT signal) while the TSOUT signal toggle operation is disabled or before CAN operation is started. TSOUT signal toggle operation is disabled by setting FCNnCMTSCTL.FCNnCMTSTSGE to 0.

Section 27 On-chip Debugging Unit (OCD)

The on-chip debugging function is described in this section.

The on-chip debugging emulator allows the debugging of programs with the microcontroller mounted on the target system.

The debugging function incorporated in this microcontroller conforms to IEEE-ISTO 5001TM-2003 Class 1, a Nexus debugging interface standard.

Caution While the microcontroller supports the debugging functions described in this section, whether they are usable or not depends on the debugger. For details on debugging, see the user's manual for the debugger.

27.1 Functional Overview

An overview of the on-chip debugging function is given below.

(1) Debugging Interface

This interface is used to communicate with the host via the on-chip debugging emulator by using the DCUTRST, DCUTCK, DCUTMS, DCUTDI, DCUTDO, and DCUTRDY signals.

(2) Debugging Monitor

The basic debugging functions below can be used by running a monitor program in a memory space for debugging while the user-created program is suspended.

- Downloading of user-created programs
- Reading and writing memory and registers
- Running user-created programs starting at any address

(3) Hardware Breaks

Up to four breakpoints can be specified for instructions and data. If a breakpoint is specified for an instruction, execution will be suspended at any address. If a breakpoint is specified for an address, execution will be suspended by access to data at any address.
Break conditions can be combined in sequences with up to two steps.

(4) Software Breaks

Software breaks are used to suspend (cause a break in) the execution of user-created programs stored in RAM at any address.

(5) Forced Breaks

A user-created program can be forcibly suspended.

(6) Forced Resetting

The microcontroller can be forcibly reset.

(7) Real-Time RAM Monitoring (RRM)

Memory can be read while the program is running. Since this read access is through debugging-dedicated DMA, it barely affects program execution.

(8) Dynamic Memory Modification (DMM)

Memory can be rewritten while the program is running. Since this rewrite access is through debugging-dedicated DMA, it barely affects program execution.

(9) Measuring Times

Using a 32-bit counter, times taken to execute user-created programs can be measured based on a clock signal obtained by dividing the TCK signal frequency by two.

(10) Masking

The following signal can be masked:

$\overline{\text{RESET}}$

(11) Selecting if Peripheral I/O Modules Run or Stop during Breaks

Whether peripheral I/O modules run or stop is selectable.

- Peripheral I/O modules always stop operating during a break.
  - Watchdog timer (WDTA0)
- Selecting whether modules run or stop during a break
  - Timers (TSG2n, TAUAn, TAUJn, ENCAn, OSTMn)
  - Asynchronous serial interfaces (UARTHn)
  - Three-line serial interfaces (CSIGn, CSIHn)
  - CAN controller (FCNn)
  - A/D converter (ADCAn)

(12) Hot Plug-in

The on-chip debugging emulator can be connected and debugging can start without resetting the CPU while it is running (hot plug-out is not supported).

(13) Security

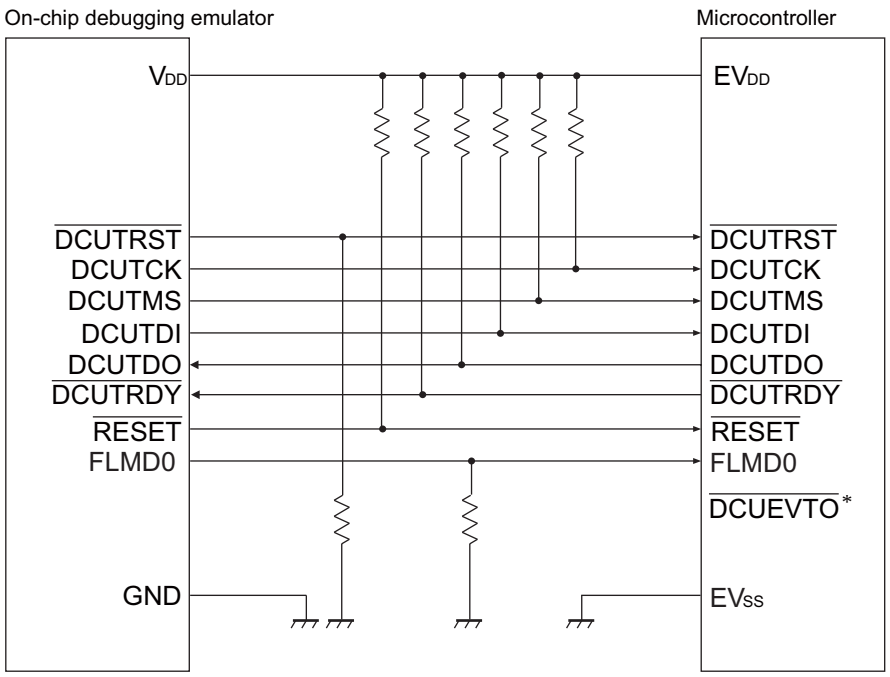
To prevent unauthorized parties reading the contents of the flash memory, a 96-bit ID code can be written to the microcontroller. If the code the user inputs does not match the ID code written to the microcontroller when starting the debugger, the flash memory is not accessible. If the last bit (bit 95) is set to 0, the flash memory is not accessible even if the ID codes match.

For details of how to set the ID code, see the user's manual for the software tools you are using.

(14) Trigger Events

The trigger events detect an event with four-step sequential execution on the execution address, the access address, the access data, and the range (comparison of large and small sizes).

27.2 Connection with the On-Chip Debugging Emulator



Note * Connect the DCUEVTO pin if the on-chip debugging emulator uses this signal.

Figure 27-1 Connection with the On-Chip Debugging Emulator

Table 27-1 Pins Used for Connection with the on-Chip Debugging Emulator

Pin Name	Description
V_{DD}	Signal used to detect power supply to the target system and power supply for the buffers in the on-chip debugging emulator
$\overline{\text{DCUTRST}}$	Signal for asynchronously resetting the debugging functions of the microcontroller
DCUTCK	Clock signal used in debugging
DCUTMS	Signal to select the mode of data transfer
DCUTDI	Input data signal for the microcontroller
DCUTDO	Output data signal from the microcontroller
$\overline{\text{DCUTRDY}}$	Synchronization signal for data transfer
$\overline{\text{RESET}}$	Reset signal for the microcontroller: this is connected so that the microcontroller is placed in the reset state from the supply of system power until the debugger starts.
FLMD0	Mode signal used in rewriting the contents of flash memory in the microcontroller

27.3 Notes on On-Chip Debugging

1. Handling of devices used for debugging
   Do not mount a device that was used in debugging on a mass-produced product, because the rewriting of flash memory during debugging will mean that the number of times flash memory is rewritable cannot be guaranteed.
2. If a pin reset is input while self-programming is in progress (the FLMD0 signal is at the high level) in debugging mode (the DCUTRST pin is at the high level), debugging will be suspended. In that case, input the pin reset again while FLMD0 is at the low level.
3. After placing the high level on DCUTRST, ensure that no resets are generated until ID authentication has been completed. If a reset is generated, it will cause self-diagnosis BIST to start, and this may cause a time-out and prevent communications with the tool.
4. After power is supplied to the microcomputer, place the high level on the RESET pin while the DCUTRST pin is low. If the internal NEXUS function has not been initialized yet so that the signal on the RESET pin changes from the low to the high level while the signal on the DCUTRST pin is still high, release from the reset state will not proceed because internal signals are undefined, and the CPU will not operate.
5. This product does not support hot plug-out, which would handle turning off the supply of power to the NEXUS tool (including disconnection of the connector) in debugging mode. If the chip is booted up in debugging mode, do not turn off the supply of power to the NEXUS tool (including disconnection of the connector) during operations.
6. After the chip is released from the RESET state and the transition of the level on the DCUTRST pin from the high to the low level, wait until at least 10 cycles of the clock signal on the DCUTCK pin are input while the DCUTMS pin is at the high level and the DCUTRST pin is at the low level before setting the DCUTRST pin to the high level.
7. Once the DCUTRST pin is at the high level, keep it at the high level (i.e. do not change the level back from high to low once the DCUTRST pin is set to the high level).
8. Terminating operation in on-chip debugging mode under any of the following conditions causes operation to become undefined. After terminating operations in the on-chip debugging mode, set the RESET pin and the DCUTRST pin to the low level so that the start-up sequence for Nexus is executed.
   - Power supply off
   - Low level on the DCUTRST pin
   - ID codes do not match
   - FLMD0 pin and RESET pin are set to the high and low levels, respectively
9. When the pin reset signal goes to the active level in on-chip debugging mode, the ERROROUT pin is fixed to the low level (when the pin reset signal is active in single-chip mode, the ERROROUT pin will be in the high-impedance state).
10. After release from the reset state in debugging mode (with resetting produced by RESET, WDTA0RES, CLMA0RES to CLMA2RES, LVIRES, SGARES, or DBRES), a self-diagnosis BIST is not executed, waiting for a time corresponding to that required to execute the self-diagnostic BIST. Then, the self-diagnosis BIST reset is generated so that the transition to the state of running the user-created program is possible.

11. Resets due to self-diagnostic BIST in debugging mode are not maskable by debugger settings.
12. A reset cannot be applied during the HEAPCLK stopping time (flash reset sequence time, PLL lock-up time, OSC stabilization time) in debugging mode, so if a reset is to be applied, ensure that HEAPCLK is operating (this applies to the μPD70F3508 and μPD70F3509 products).

Section 28 Electrical Characteristics

28.1 Electrical Characteristics of the µPD70F3505A

28.1.1 Absolute Maximum Ratings

(T_a = 25°C)

Item	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DDM}	$V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 1.6	V
	V_{DDC}		-0.5 to + 1.6	V
	$CPUPLLV_{DD}$		-0.5 to + 1.6	V
	EV_{DD}	$EV_{DD} = OSCV_{DD} = FV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 6.0	V
	$OSCV_{DD}$		-0.5 to + 6.0	V
	FV_{DD}		-0.5 to + 6.0	V
	AV_{DD0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 6.0	V
	V_{SSM}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	V_{SSC}		-0.5 to + 0.5	V
	$CPUPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	EV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	$OSCV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	FV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	AV_{SS0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
Input voltage	V_I		-0.5 to EV_{DD} + 0.5*	V
Analog input voltage	V_{IAN}	ADCA0I1 to ADCA0I18	-0.5 to AV_{DD0} + 0.5*	V
	AV_{REF0P}		-0.5 to AV_{DD0} + 0.5*	V
	AV_{REF0M}		-0.5 to AV_{DD0} + 0.5*	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total pins	-50	mA
Low-level output current	I_{OL}	1 pin	10	mA
		Total pins	50	mA
Operating ambient temperature	T_a		-40 to + 110	°C
Storage temperature	T_{stg}	Before writing to flash memory	-65 to + 150	°C
		After writing to flash memory	-65 to + 125	°C

Note * Be sure not to exceed the absolute maximum ratings of each supply voltage.

Caution 1. Make sure that the total the current flowing through these pins and output currents does not exceed 50 mA.

Caution 2. Do not directly connect output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open drain pins or open collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

Caution 3. Product quality may suffer if the absolute maximum rating is exceeded even momentary for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

28.1.2 Capacity

(T_a = 25°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz 0 V for non-measured pins, and excluding analog pins selected for use with S/H channels or AD converter channels.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

28.1.3 Operating Conditions

(T_a = -40 to + 110°C, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Internal Operation Clock Frequency	Operating Ambient Temperature (T_a)	Power Supply Voltage
fxx = 64 MHz	T_a = -40 to + 110°C	V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V AV_{DD0} = 4.5 to 5.5 V
fxx = 80 MHz		

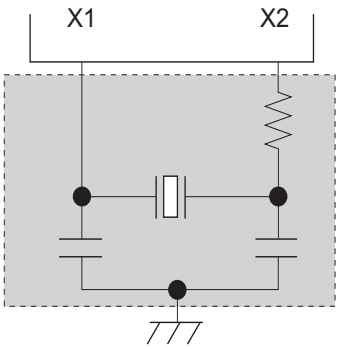
28.1.4 Characteristics of Oscillation Circuit

(T_a = -40 to + 110°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ = 1.1 to 1.3 V,
$EV_{DD} = OSCV_{DD} = FV_{DD}$ = 3.0 to 5.5V, AV_{DD0} = 4.5 to 5.5 V,
$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency*	fx		8	8	8	MHz

Note * The values are only applicable as oscillation circuit characteristics.

Caution 1. When placing wiring in the area enclosed by the broken line in the figure below during the design of the oscillator, avoid adverse effects from wiring capacitance by following the directions below.



- Bring the oscillation circuit as close as possible to the X1 and X2 terminals.
- Do not pass other signal conductors through the broken line part in the above figure.
- Do not route the wiring near signal lines through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as $OSCV_{SS}$.
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When selecting the crystal oscillator, designing the parameters of the oscillation circuit, or calculating the time required for oscillation to become stable, exhaustively evaluate the finished system (MCU, oscillator, and target board) for matching of the device and the oscillator or ask the manufacturer of the oscillator to do so.

28.1.5 DC Characteristics

DC Characteristics (1/3)

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$= EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-level input voltage	V_{IH1}	Port pins, FLMD0 pin	0.7 × EV_{DD}		EV_{DD}+0.3	V
	V_{IH2}	FLMD1 pin	0.8 × EV_{DD}		EV_{DD}+0.3	V
	V_{IH3} *5,*7	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.2		EV_{DD}+0.3	V
	V_{IH4}*5,*8	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.0		EV_{DD}+0.3	V
	V_{IH5}	$\overline{\text{RESET}}$ pin	0.8 × EV_{DD}		EV_{DD}+0.3	V
Low-level input voltage	V_{IL1}	Port pins, FLMD0 pin	-0.5		0.3 × EV_{DD}	V
	V_{IL2}	FLMD1 pin	-0.5		0.2 × EV_{DD}	V
	V_{IL3}*5	DCUTDI, DCUTMS DCUTCK	-0.5		0.8	V
	V_{IL4}*7	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
	V_{IL5}*8	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
Hysteresis width*1	V_{HYS1}	Port pins, FLMD0 pin		0.5		V
	V_{HYS2}	FLMD1 pin		1.0		V
	V_{HYS3}	$\overline{\text{RESET}}$ pin		1.0		V
High-level output voltage*2	V_{OH1}	I_{OH} = - 3 mA	EV_{DD}-1.0		EV_{DD}	V
		I_{OH} = - 1 mA	EV_{DD}-1.0		EV_{DD}	V
		I_{OH} = - 0.1 mA	EV_{DD}-0.5		EV_{DD}	V
Low-level output voltage*2	V_{OL1}	I_{OL} = 3 mA	0		0.4	V
		I_{OL} = 1 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V, other than for the pins below	20	40	100	KΩ
	R_2	V_I = 0 V *6	10	19	50	KΩ
Pull-down resistor	R_3	V_I = EV_{DD}*3	20	40	100	KΩ
	R_4	V_I = EV_{DD}*4	25	40	100	KΩ
	R_5	V_I = EV_{DD}*6	10	19	50	KΩ

Note 1. When the operation of an adjacent pin is halted.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.
Note 3. Only for the FLMD1 and $\overline{\text{DCUTRST}}$ pins
Note 4. Only for the $\overline{\text{RESET}}$ pin
Note 5. In debugging mode
Note 6. Only for the FLMD0 pin
Note 7. EV_{DD} = 4.5 to 5.5 V
Note 8. EV_{DD} = 3.0 to 4.5 V

Caution The characteristics of the alternative function pins are the same as those of the port pin unless otherwise specified.

DC Characteristics (2/3)

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
High-level input leakage current	I_{LIH1}	V_{IAN} = AV_{DD0}	Analog pins*1	EV_{DD} = 3.0 to 4.5 V			0.2	μA
				EV_{DD} = 4.5 to 5.5 V			0.3	μA
		V_I = EV_{DD}	Pins other than analog pins*2				5	μA
Low-level input leakage current	I_{LIL1}	V_{IAN}=0 V	Analog pins*1	EV_{DD} = 3.0 to 4.5 V			-0.2	μA
				EV_{DD} = 4.5 to 5.5 V			-0.3	μA
		V_I = 0 V	Pins other than analog pins*2				-5	μA
High-level output leakage current	I_{LOH1}	V_I = EV_{DD}					5	μA
Low-level output leakage current	I_{LOL1}	V_I = 0V					-5	μA

Note 1. ADCA0I1 to ADCA0I18
Note 2. The current flowing through the pull-up/pull-down resistors is not included.

DC Characteristics (3/3)

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power-supply current*1	Normal operating conditions	I_{DD}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$)	fxx = 64 MHz	PCLK= 1/2HEAPCLK		240	mA
					PCLK= HEAPCLK		270	mA
				fxx = 80 MHz	PCLK= 1/2HEAPCLK		280	mA
					PCLK= HEAPCLK		320	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			73.5 + α	mA
	When rewriting flash memory	I_{DDF}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$)	fxx = 64 MHz			260	mA
				fxx = 80 MHz			300	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			73.5 + α	mA
	When executing BIST	I_{DDB}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$)	fxx = 64 MHz			285	mA
				fxx = 80 MHz			340	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})				30	mA
	When HALT	I_{DDH}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$)	fxx = 64 MHz	PCLK= 1/2HEAPCLK		210	mA
					PCLK= HEAPCLK		250	mA
				fxx = 80 MHz	PCLK= 1/2HEAPCLK		240	mA
					PCLK= HEAPCLK		290	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			73.5 + α	mA

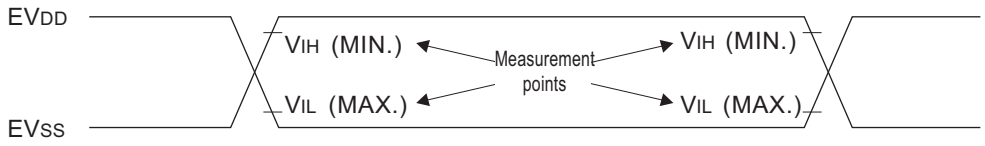
Note 1. The current flowing through the pull-up/pull-down resistors is not included.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.

Note MAX. is the maximum value of average current.

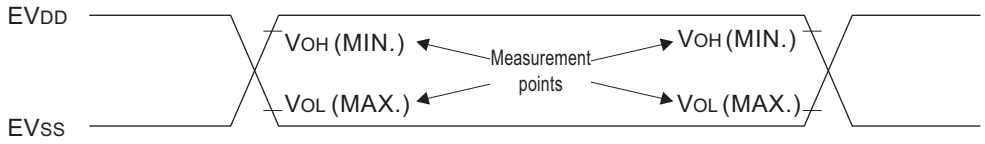
28.1.6 AC Characteristics

28.1.6.1 Measurement Conditions

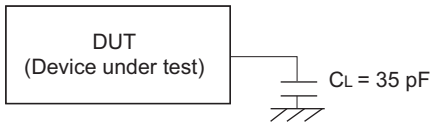
Input waveform for AC testing



AC test output measurement points



Load conditions

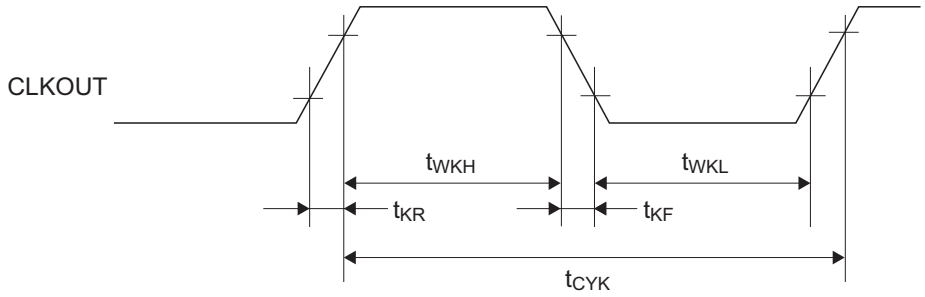


Caution If the load capacitance exceeds 35 pF due to the circuit configuration, bring the load capacitance of the device to 35 pF or less by inserting a buffer or by some other means.

28.1.6.2 Clock Timing (CLKOUT)

(T_a = -40 to + 110°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD}$ = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CLKOUT output cycle time	t_{CYK}		0.1	4	µs
CLKOUT high level width	t_{WKH}		$t_{CYK}/2$ - 13		ns
CLKOUT low level width	t_{WKL}		$t_{CYK}/2$ - 13		ns
CLKOUT rise time	t_{KR}			13	ns
CLKOUT fall time	t_{KF}			13	ns



28.1.6.3 Timing in Turning the Power Supply On and Off

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
V_{DDM},V_{DDC} ↑ to EV_{DD} ↑	t_{RLI}		0	1000	ms
EV_{DD} ↑ to V_{DDM}, V_{DDC} ↑	t_{RIL}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to AV_{DD0} ↑	t_{RLA}		0	1000	ms
AV_{DD0} ↑ to V_{DDM}, V_{DDC} ↑	t_{RAL}		0	1000	ms
EV_{DD} ↑ to AV_{DD0} ↑	t_{RIA}		0	1000	ms
AV_{DD0} ↑ to EV_{DD} ↑	t_{RAI}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to $\overline{\text{RESET}}$ ↑	t_{RLR}	*1	t_{OSC}+ 0.5		ms
EV_{DD} ↑ to $\overline{\text{RESET}}$ ↑	t_{RIR}	*1	t_{OSC}+ 0.5		ms
AV_{DD0}↑to $\overline{\text{RESET}}$ ↑	t_{RAR}	*1	t_{OSC}+ 0.5		ms
FLMD1,0 setup time (to $\overline{\text{RESET}}$ ↑)	t_{RMR0}		0		ns
FLMD1,0 hold time (to $\overline{\text{RESET}}$ ↓)	t_{FRM0}	*2	0		ns
$\overline{\text{RESET}}$ ↓ to V_{DDM}, V_{DDC} ↓	t_{FRL}		500		ns
$\overline{\text{RESET}}$ ↓ to EV_{DD} ↓	t_{FRI}		500		ns
$\overline{\text{RESET}}$ ↓ to AV_{DD0} ↓	t_{FRA}		500		ns
EV_{DD} ↓ to V_{DDM}, V_{DDC} ↓	t_{FIL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to EV_{DD} ↓	t_{FLI}		0	1000	ms
AV_{DD0} ↓ to V_{DDM}, V_{DDC} ↓	t_{FAL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to AV_{DD0} ↓	t_{FLA}		0	1000	ms
AV_{DD0} ↓ to EV_{DD} ↓	t_{FAI}		0	1000	ms
EV_{DD} ↓ to AV_{DD0} ↓	t_{FIA}		0	1000	ms

Note 1. t_{OSC}: Oscillation stabilization time

Note 2. Using the specified value is recommended for the FLMD1, 0 hold time (time until $\overline{\text{RESET}}$ falls: t_{FRM0}) . If using the recommended value is difficult, change the FLMD1, 0 pins once rewriting of the flash memory is complete.

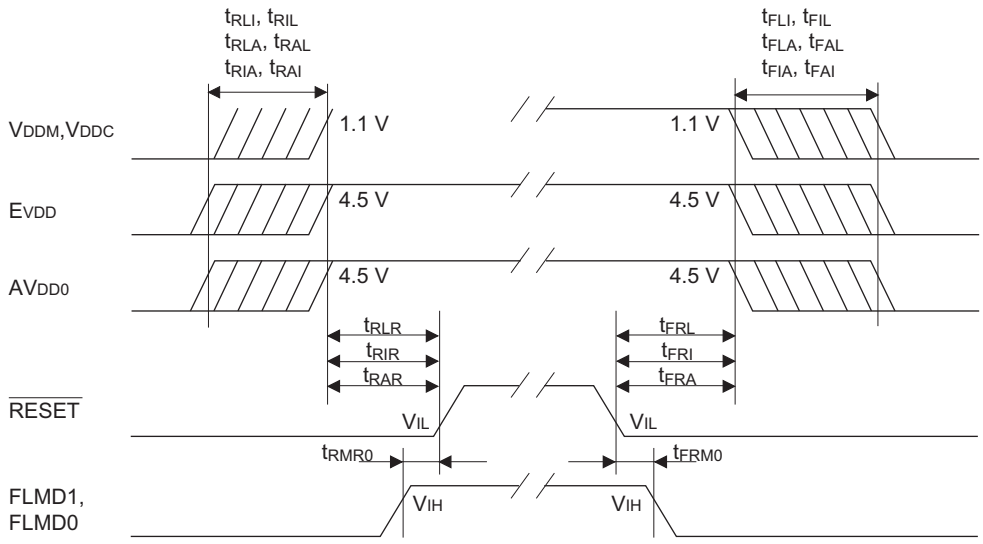
Caution 1. Turn on/off internal power system at the same time.

Caution 2. Turn on/off external power system at the same time.

Caution 3. Apply an external reset ($\overline{\text{RESET}}$) whenever the power supply voltage falls below the operating voltage.

Caution 4. If the power-on flag or self-diagnosis (BIST) control registers are to be used, boot the internal power system first (when the internal power supply voltage reaches 1.1 V, set the voltage of the external power system to a voltage below the detection voltage [min: 2.7 V] for POF).

Note Internal power system: V_{DDM}, V_{DDC}, $CPUPLLV_{DD}$
External power system: EV_{DD}, $OSCV_{DD}$, FV_{DD}


tRLI, tRIL
tRLA, tRAL
tRIA, tRAI
tFLI, tFIL
tFLA, tFAL
tFIA, tFAI
VDDM,VDDC
1.1 V
1.1 V
EVDD
4.5 V
4.5 V
AVDD0
4.5 V
4.5 V
tRLR
tRIR
tRAR
tFRL
tFRI
tFRA
RESET
VIL
VIL
tRMR0
tERM0
FLMD1,
FLMD0
VIH
VIH

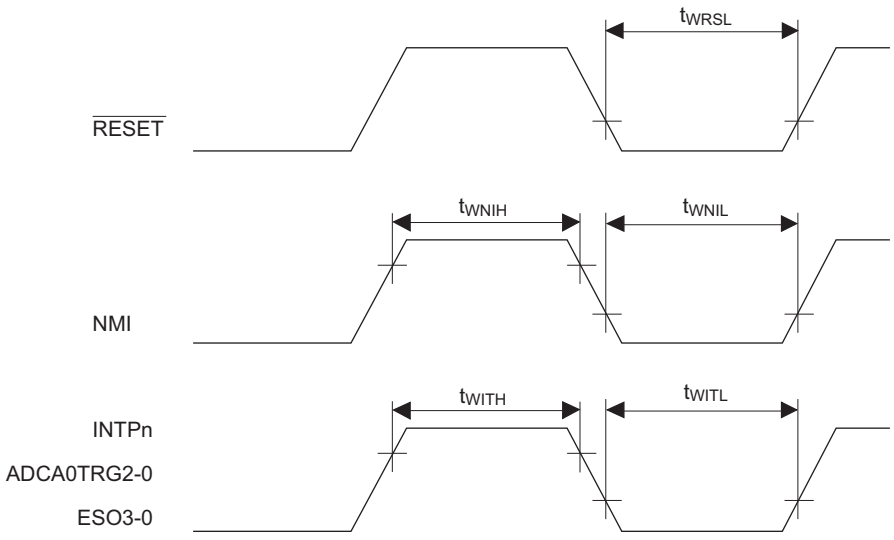
28.1.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing

(T_a = -40 to + 110°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD}$ = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}	Other than power supply turning on	500		ns
NMI input high level width	t_{WNIH}	Digital filter	*1		ns
NMI input low level width	t_{WNIL}	Digital filter	*1		ns
INTPn input high level width	t_{WITH}	Digital filter, n = 0 to 9	*1		ns
INTPn input low level width	t_{WITL}	Digital filter, n = 0 to 9	*1		ns
ADCA0TRG2-0 input high level width	t_{WITH}	Analog filter	250		ns
ADCA0TRG2-0 input low level width	t_{WITL}	Analog filter	250		ns
ESO3-0 input high level width	t_{WITH}	Analog filter	250		ns
ESO3-0 input low level width	t_{WITL}	Analog filter	250		ns

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.



Note n = 0 to 9

28.1.6.5 Timer Timing

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
TI input high level width	t_{TIH}	TAUA0I0-15, TAUA1I0-13, TAUJ0I0-3, TAUJ1I0-3, TSG20PTSI0-2, TSG21PTSI0-2, ENCA0E0,1, ENCA0EC, ENCA1E0,1, ENCA1EC	*1		ns
TI input low level width	t_{TIL}	TAUA0I0-15, TAUA1I0-13, TAUJ0I0-3, TAUJ1I0-3 TSG20PTSI0-2, TSG21PTSI0-2, ENCA0E0,1, ENCA0EC, ENCA1E0,1, ENCA1EC	*1		ns
TO output cycle time	t_{TCYK}	TAUA0O0-15, TAUA1O0-15 TAUJ0O0-3, TAUJ1O0-3 OSTM0O, OSTM1O TSG20O1-7, TSG21O0-7, TPB0O, TPB1O	0.25	1000000	µs

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.

28.1.6.6 CSIHn Timing

(a) Master mode

(T_a = –40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYM}	Output	125		ns
CSIHnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) – 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIM}		30		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIM}		10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOM}			10	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOM}	CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 0/0 or 1/1	t_{KWHM}		ns
		CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 1/0 or 0/1	t_{KWLM}		ns
CSIHnRYI setup time (to CSIHnSC)	t_{SRYI}		(2 × t_{PCLK}) + 30		ns
CSIHnCSx setup time	t_{SSCSB0}	C SIHnCFGx.CSIHnDAPx = 0	CS_{SETUP} × t_{KCYM} – 10		ns
	t_{SSCSB1}	CSIHnCFGx.CSIHnDAPx = 1	(CS_{SETUP} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx hold time	t_{HSCSB0}	CSIHnCTL1.CSIHnSIT = 0	CS_{HOLD} × t_{KCYM} – 10		ns
	t_{HSCSB1}	CSIHnCTL1.CSIHnSIT = 1	(CS_{HOLD} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx inactive level width	t_{WSCSB}	*1	(CS_{IDLE} + 0.5) × t_{KCYM} – 20		ns
		Other than the above	CS_{IDLE} × t_{KCYM} – 20		ns

Note 1. When the setting of CSIHnCFGx.CSIHnCKPx is changed while CS_{IDLE} = 000_B.

Note n = 0, x = 0 to 3

CS_{SETUP}: CSIHnCFGx.CSIHnSPx[3:0] bits

CS_{HOLD}: CSIHnCFGx.CSIHnHDx[3:0] bits

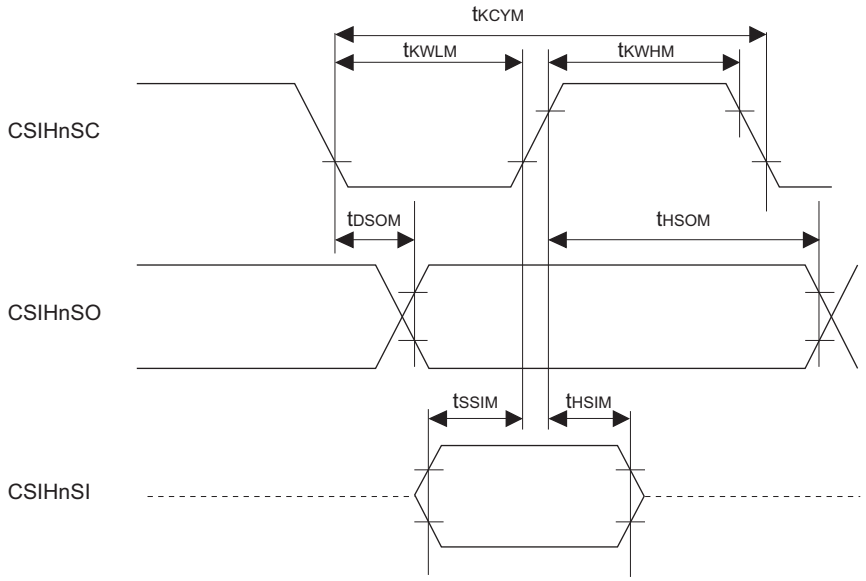
CS_{IDLE}: CSIHnCFGx.CSIHnIDx[2:0] bits

t_{PCLK}: PCLK cycle time

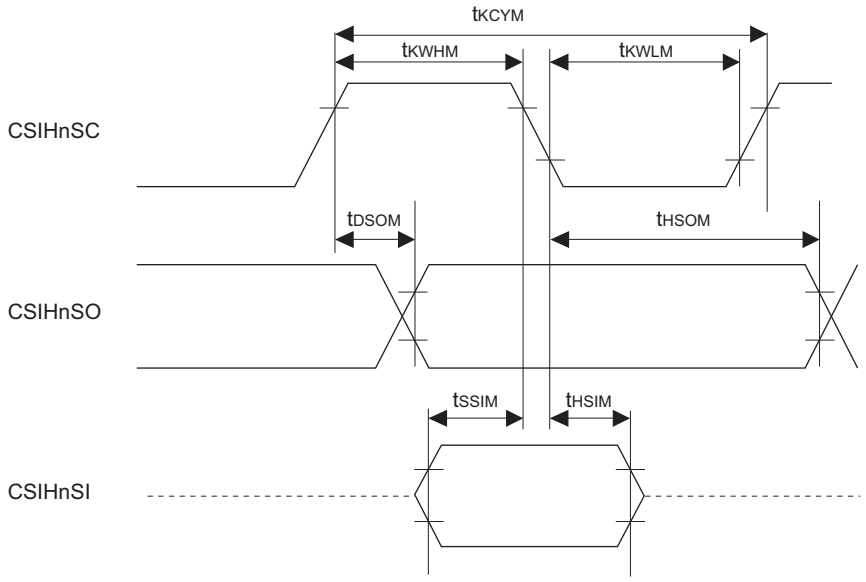
[CSIHnSC/CSIHnSI/CSIHnSO]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0 or 1/1)



Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 1/0 or 0/1)

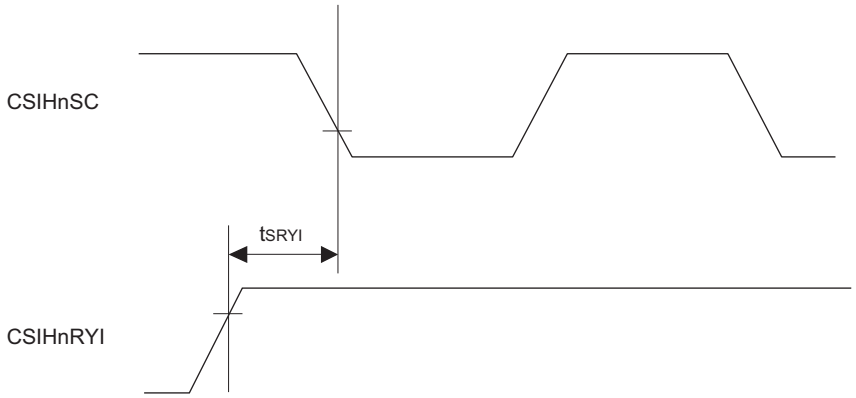


Note n = 0, x = 0 to 3

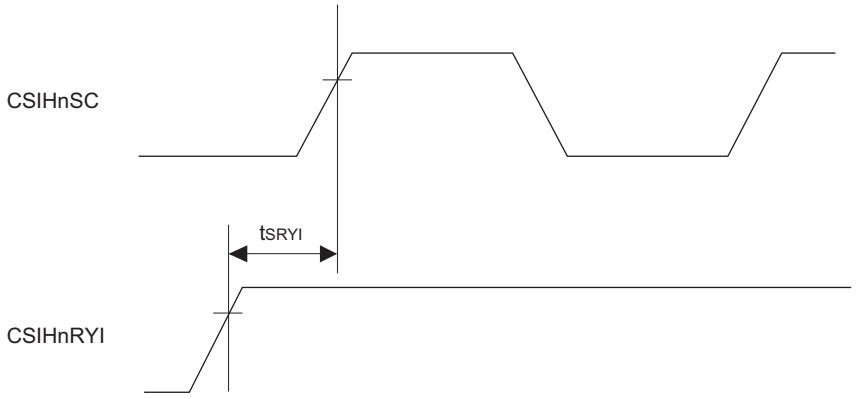
[CSIHnSC/CSIHnRYI]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx = 0)



Master mode: (CSIHnCFGx: CSIHnCKPx = 1)

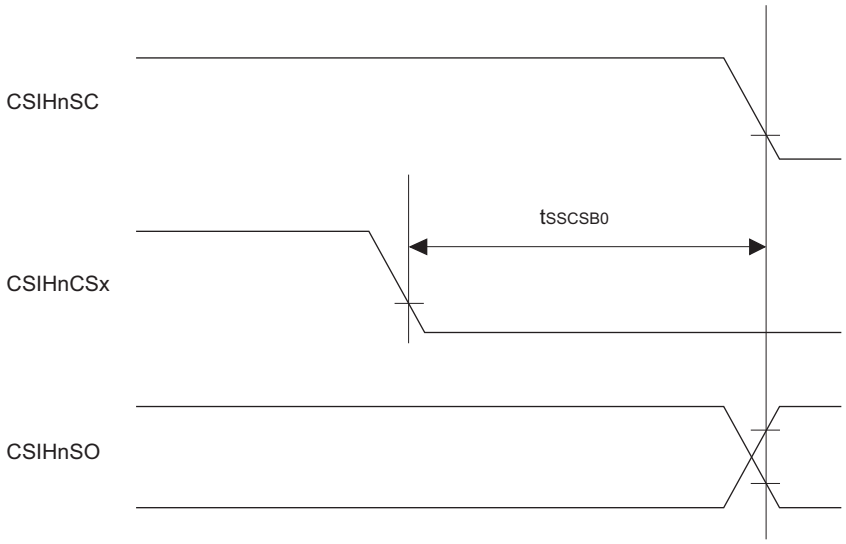


Note n = 0, x = 0 to 3

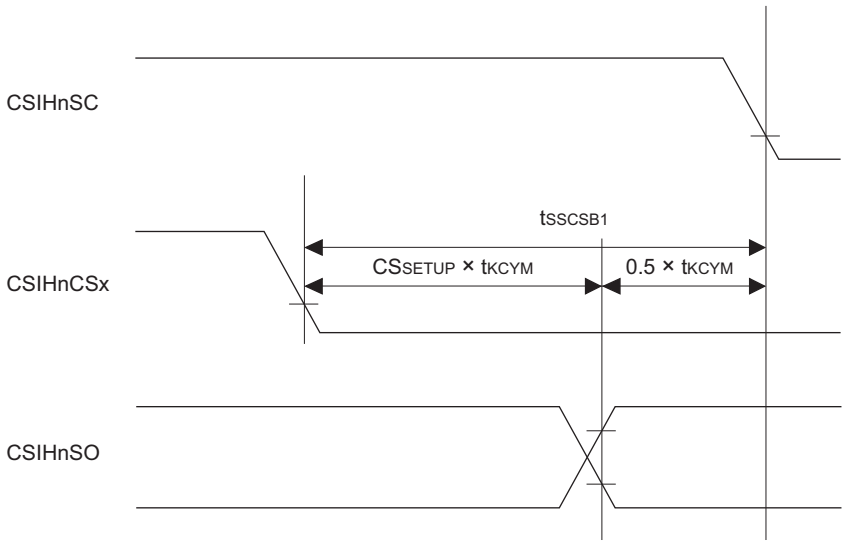
[CSIHnSC/CSIHnSO/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/1)



Note n = 0, x = 0 to 3

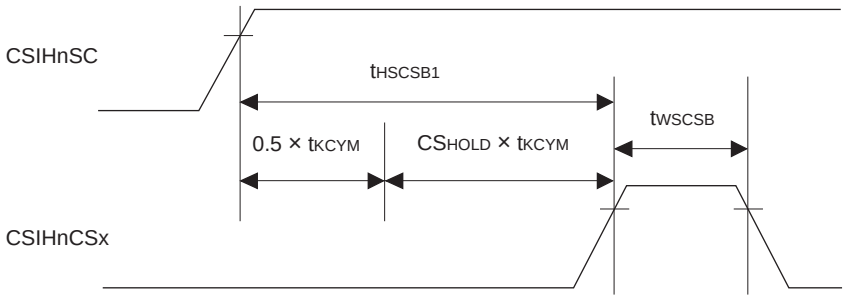
[CSIHnSC/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



Note n =0, x = 0 to 3

(b) Slave Mode

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

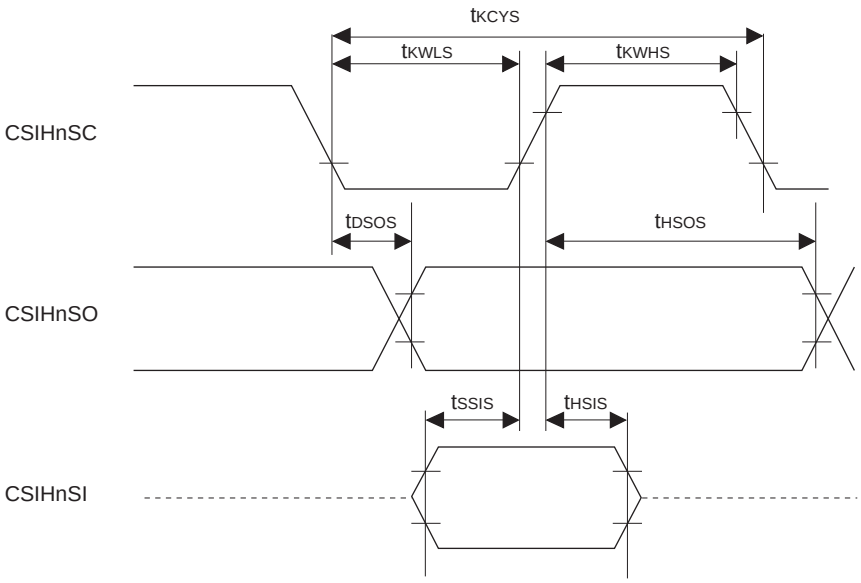
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIHnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIS}			15		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOS}				30	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOS}	CSIHnCTL1.CSIHnCKR / CSIHnCFG0.CSIHnDAP0 = 0/0 or 1/1		t_{KWHS}		ns
		CSIHnCTL1.CSIHnCKR / CSIHnCFG0.CSIHnDAP0 = 1/0 or 0/1		t_{KWLS}		ns
CSIHnRYO output delay	t_{SRYO}				30	ns

Note n = 0

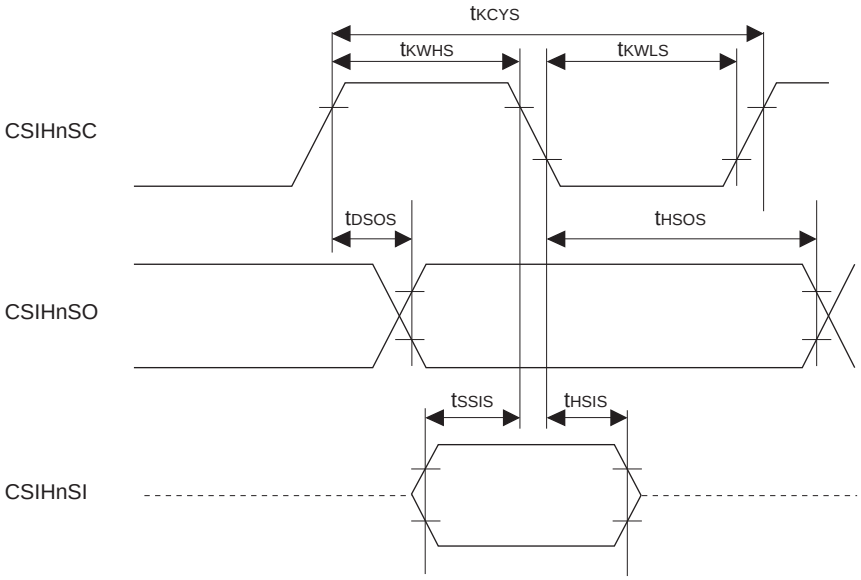
t_{PCLK}: PCLK cycle time

[CSIHnSC/CSIHnSI/CSIHnSO]

Slave mode: (CSIHnCTL: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



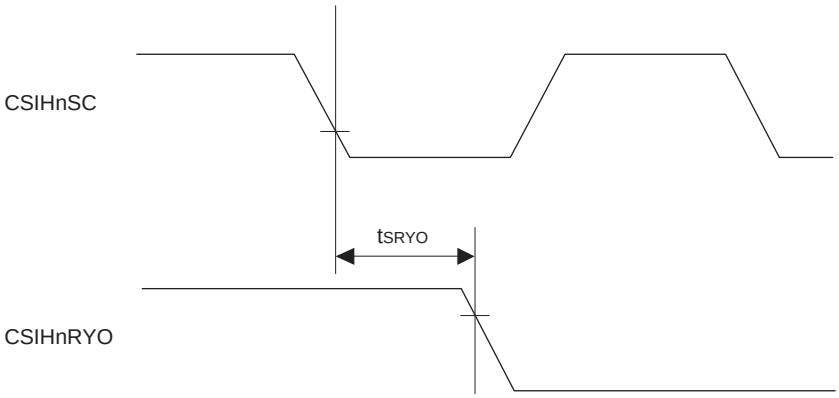
Slave mode: (CSIHnCTL: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 1/0 or 0/1)



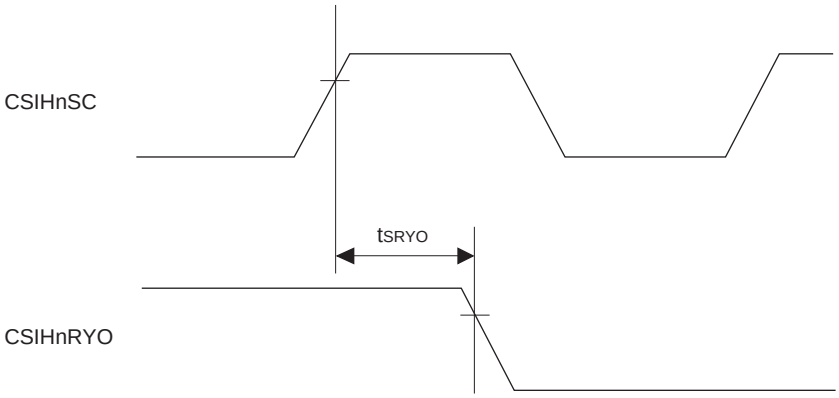
Note n = 0

[CSIHnSC/CSIHnRYO]

(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/1 or 1/0)



Note n = 0

28.1.6.7 CSIGn Timing

(a) Master Mode

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0V, C_L = 35 pF)

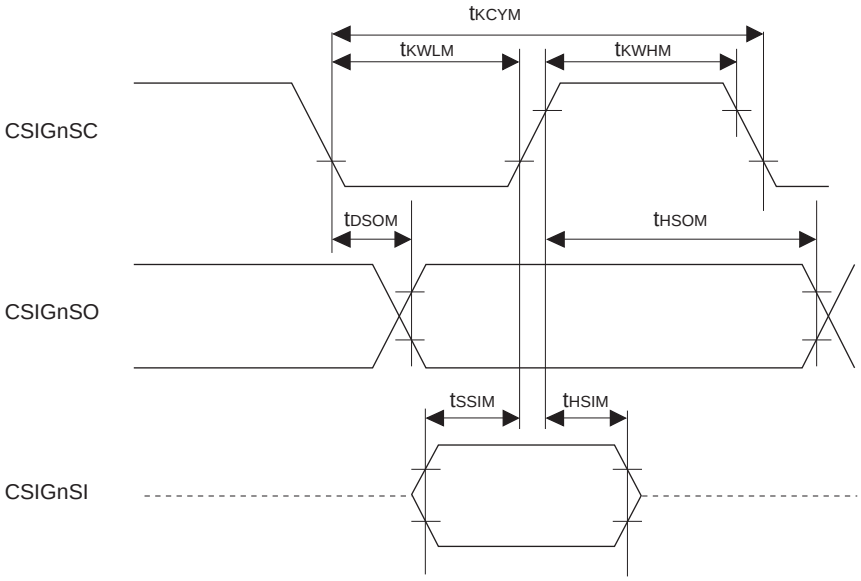
Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYM}	Output	125		ns
CSIGnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIM}		30		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIM}		10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOM}			10	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOM}	CSIGnCTL1.CSIGnCKR / CSIGnCFG0.CSIGnDAP = 0/0 or 1/1	t_{KWHM}		ns
		CSIGnCTL1.CSIGnCKR / CSIGnCFG0.CSIGnDAP = 1/0 or 0/1	t_{KWLM}		ns
CSIGnRYI setup time (to CSIGnSC)	t_{SRYI}		(2 × t_{PCLK}) + 30		ns

Note n = 0 to 2 (for CSIGnRYI, n = 0, 1)

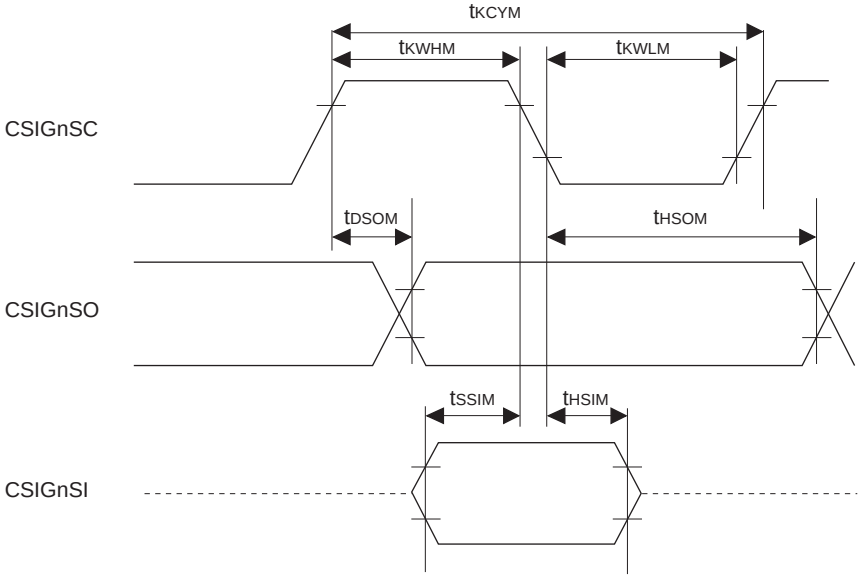
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



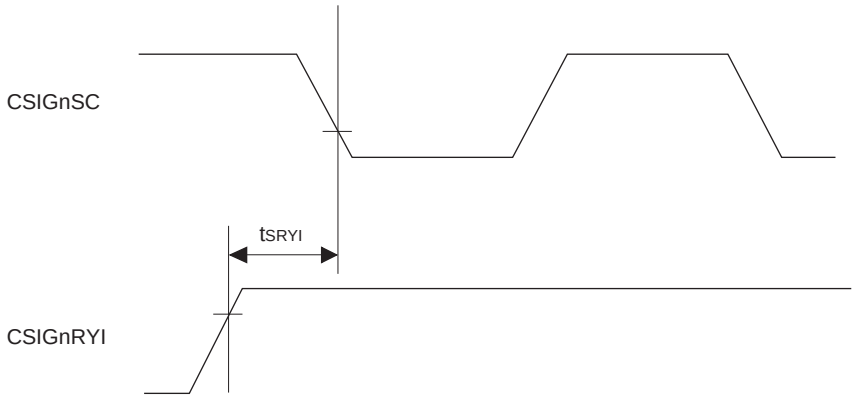
Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



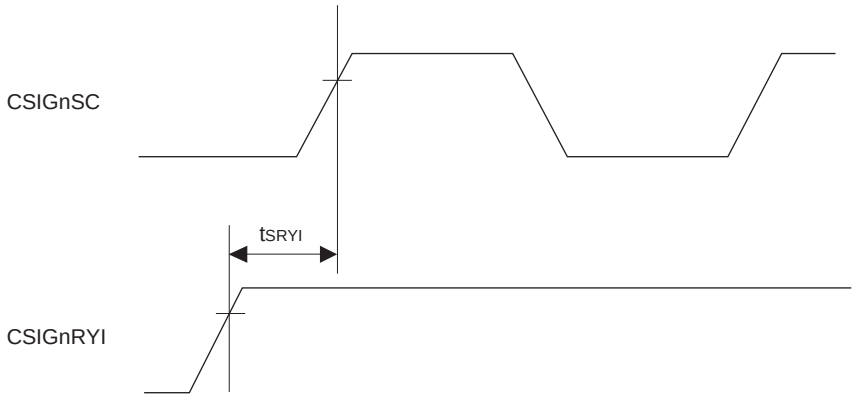
Note n = 0 to 2

[CSIGnSC/CSIGnRYI]

Master mode: (CSIGnCTL1: CSIGnCKR = 0)



Master mode: (CSIGnCTL1: CSIGnCKR = 1)



Note n = 0 to 2 (for CSIGnRYI, n = 0, 1)

(b) Slave Mode

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

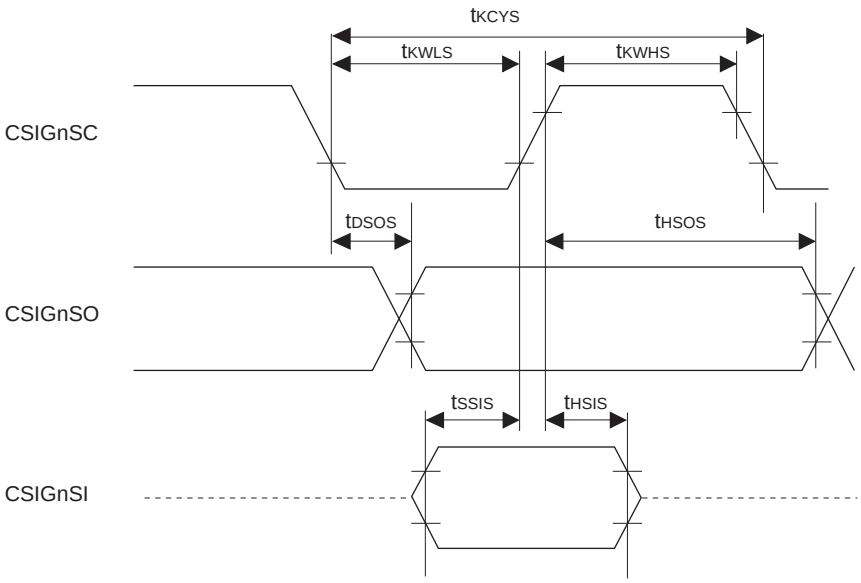
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIGnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIS}			15		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOS}				30	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOS}	CSIGnCTL1.CSIGnCKR / CSIGnCFG0.CSIGnDAP = 0/0 or 1/1		t_{KWHS}		ns
		CSIGnCTL1.CSIGnCKR / CSIGnCFG0.CSIGnDAP = 1/0 or 0/1		t_{KWLS}		ns
CSIGnRYO output delay	t_{SRYO}				30	ns

Note n = 0 to 2

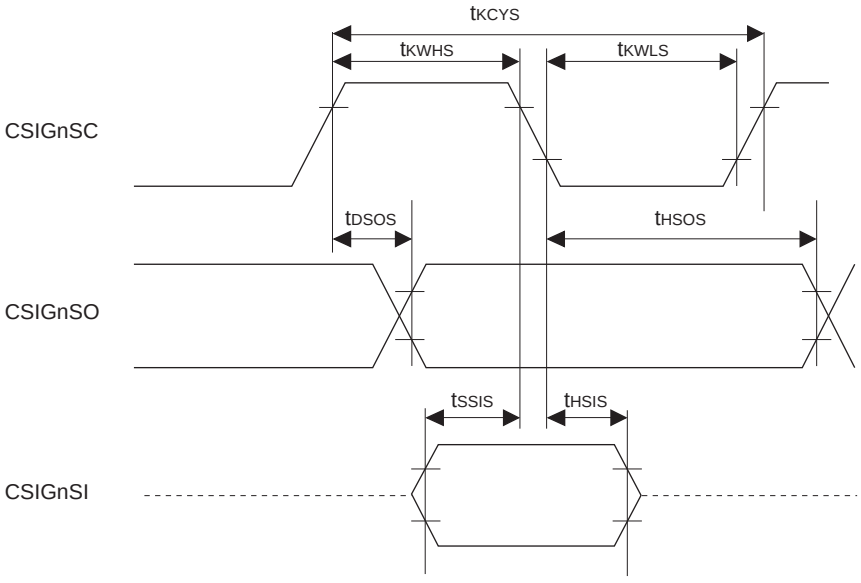
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Slave mode: (CSIGnCTL: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



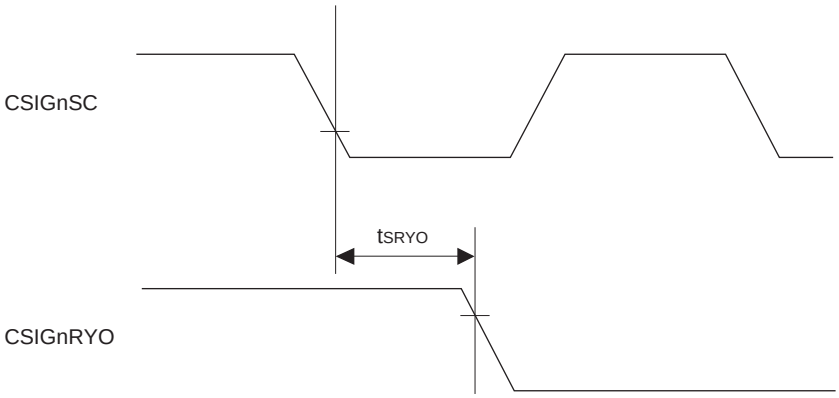
Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



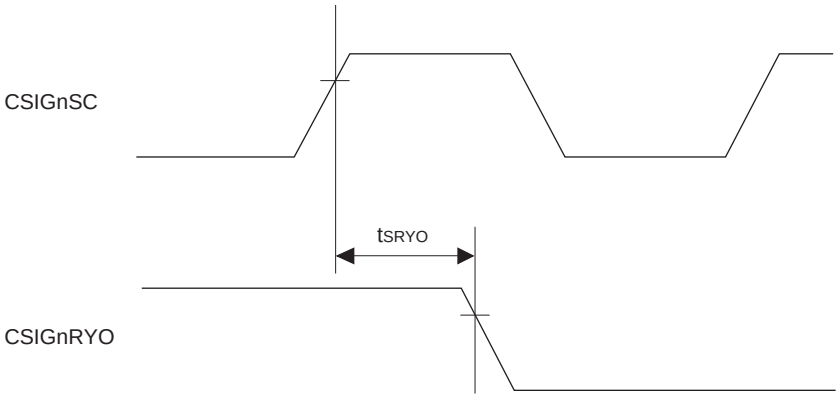
Note n = 0 to 2

[CSIGnSC/CSIGnRYO]

(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1 or 1/0)



Note n = 0 to 2

28.1.6.8 UARTHn Timing

(a) Asynchronous Mode

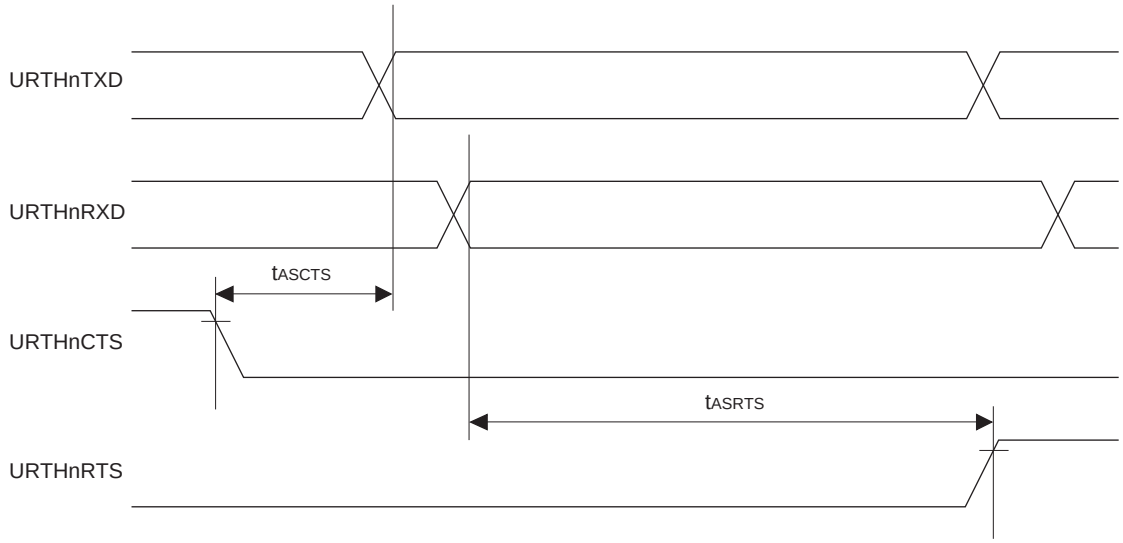
(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		MIN.	MAX.	Unit
UARTHn transfer rate				16	2 M	bps
URTHnCTS setup time (to URTHnTXD)	t_{ASCTS}	Input	URTHnCTS without noise filter	3T + 30		ns
			URTHnCTS with noise filter	5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{ASRTS}	Output	URTHnRXD without noise filter		(3BRS + 2) T + 30	ns
			URTHnRXD with noise filter		(3BRS + 4) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]



(b) Clock Synchronous, Master Mode (Normal Mode)

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM1}	Output		500		500		ns
URTHnSC high level width	t_{KWHM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnSC low level width	t_{KWLM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDM1}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDM1}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM1}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM1}	Output		t_{KCYM1}/2 -10		t_{KCYM1}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM1}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM1}	Output			T + 30		T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

(c) Clock Synchronous, Slave Mode (Normal Mode)

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnSC/URTHnRXD without Noise Filter		URTHnSC/URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		10		10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		2T + 10		2T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		-T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	2T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		2T + 30	3T + 30	2T + 30	3T + 30	ns

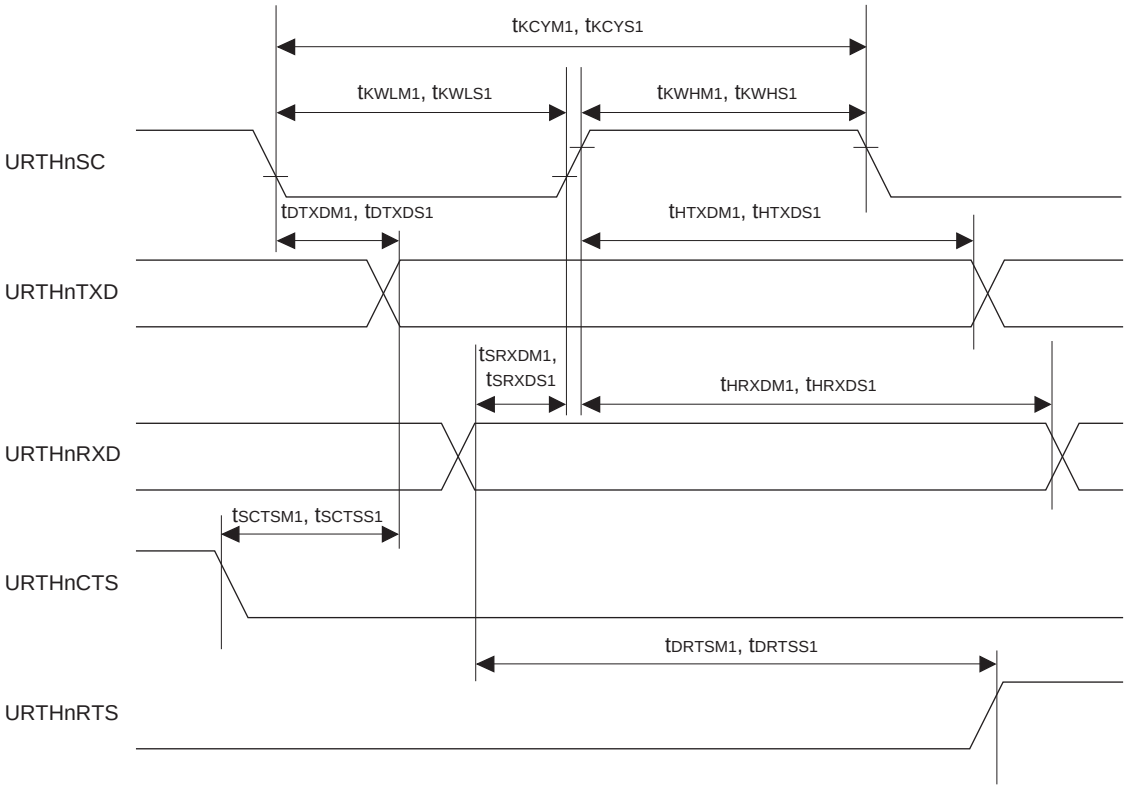
Note n = 0 to 2

T = prescaler cycle time

Item	Symbol	Conditions		URTHnSC without Noise Filter URTHnRXD with Noise Filter		URTHnSC with Noise Filter URTHnRXD without Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		2T + 10		-2T + 10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		10		4T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	3T + t_{KCYS1} + 30		3T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		4T + 30	5T + 30	30	T + 30	ns

Note n = 0 to 2

T = prescaler cycle time


tKCYM1, tKCYS1
tKWLM1, tKWLS1
tKWHM1, tKWHS1
URTHnSC
tDTXDM1, tDTXDS1
tHTXDM1, tHTXDS1
URTHnTXD
tSRXDM1,
tSRXDS1
tHRXDM1, tHRXDS1
URTHnRXD
tSCTSM1, tSCTSS1
URTHnCTS
tDRTSM1, tDRTSS1
URTHnRTS

Note n = 0 to 2

(d) Clock Synchronous, Master Mode (Delay Sampling Mode)

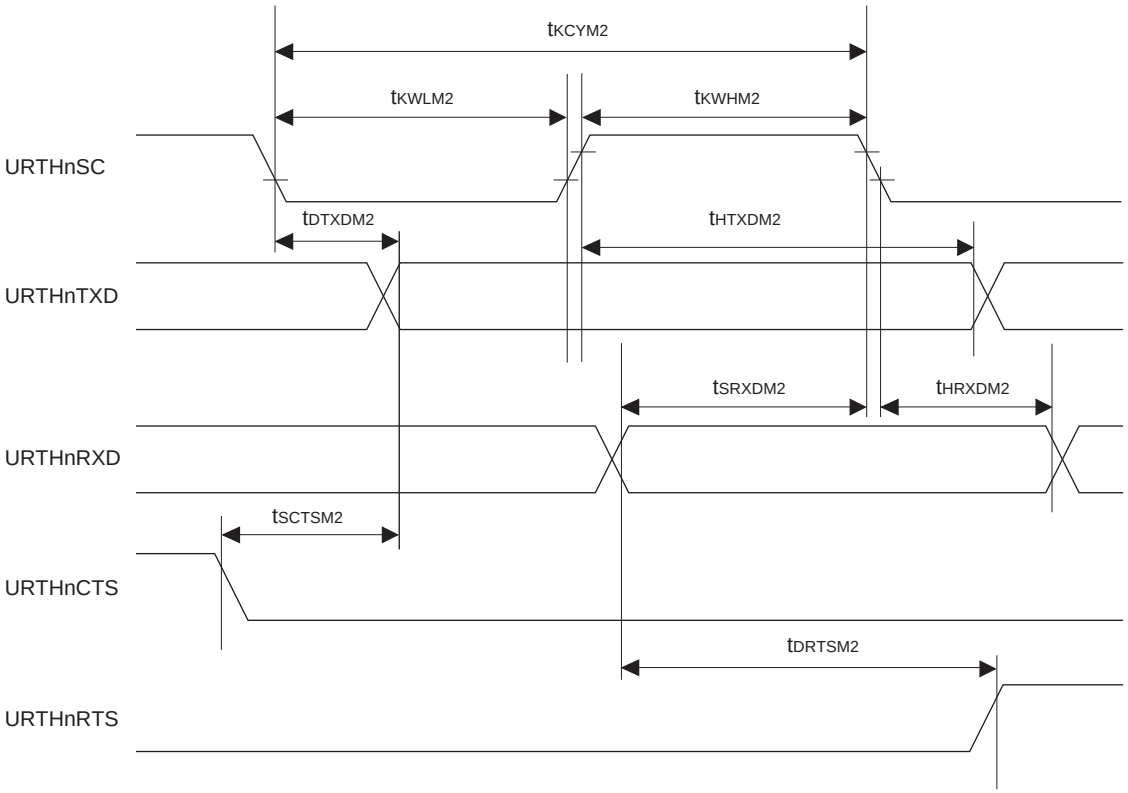
(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM2}	Output		500		500		ns
URTHnSC high level width	t_{KWHM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnSC low level width	t_{KWLM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↓)	t_{SRXDM2}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↓)	t_{HRXDM2}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM2}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM2}	Output		t_{KCYM2}/2 - 10		t_{KCYM2}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM2}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM2}	Output			(BRS + 1) T + 30		(BRS + 1) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]


tKCYM2
tKWLM2
tKWHM2
URTHnSC
tDTXDM2
tHTXDM2
URTHnTXD
tSRXDM2
tHRXDM2
URTHnRXD
tSCTSM2
URTHnCTS
tDRTSM2
URTHnRTS

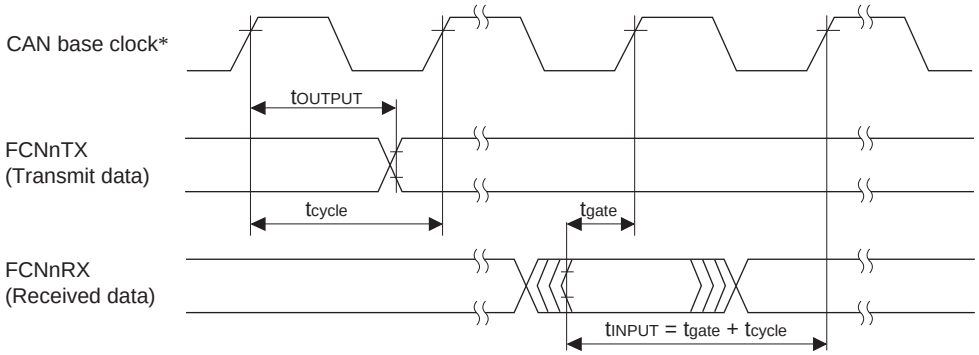
Note n = 0 to 2

28.1.6.9 CAN Timing

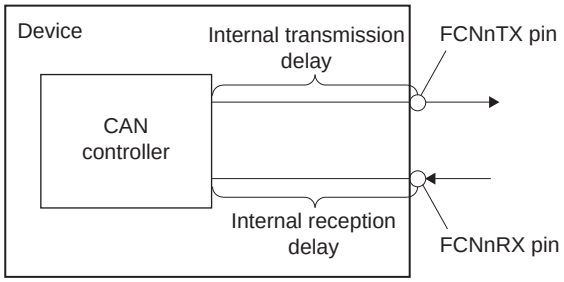
(T_a = -40 to + 110°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD}$ = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CAN transfer rate				1	Mbps
CAN internal delay time*[1]	t_{node}			87.5	ns

Note 1. CAN internal delay time (t_{node}) = t_{OUTPUT} (internal transmission delay) + t_{INPUT} (internal reception delay)



Note * CAN base clock frequency: f_{CAN}



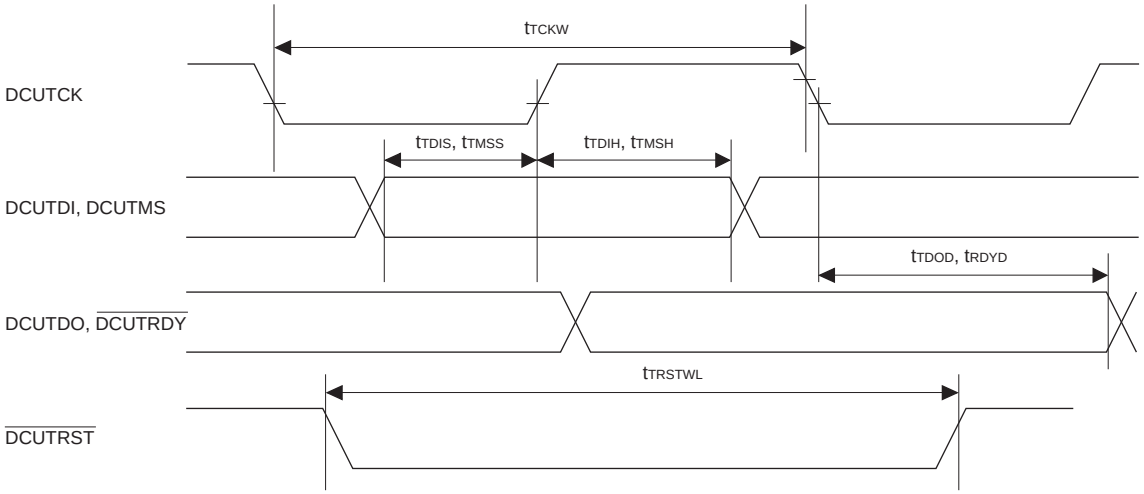
Schematic view of the CAN-internal delay

Note n = 0, 1

28.1.6.10 NEXUS I/F Timing

(T_a = 0 to + 40°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	MIN.	MAX.	Unit
DCUTCK cycle time	t_{TCKW}	40.0		ns
DCUTDI setup time (to DCUTCK ↑)	t_{TDIS}	t_{TCKW}/2 - 8.5		ns
DCUTMS setup time (to DCUTCK ↑)	t_{TMSS}	t_{TCKW}/2 - 8.5		ns
DCUTDI hold time (to DCUTCK ↑)	t_{TDIH}	2.0		ns
DCUTMS hold time (to DCUTCK ↑)	t_{TMSH}	2.0		ns
DCUTDO output delay time (to DCUTCK ↓)	t_{TDOD}		20.0	ns
$\overline{\text{DCUTRDY}}$ output delay time (to DCUTCK ↓)	t_{RDYD}		20.0	ns
$\overline{\text{DCUTRST}}$ low level width	t_{TRSTWL}	1000		ns



28.1.6.11 A/D Converter Characteristics

(a) 10-Bit Resolution, without S/H

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	μs
Accuracy*1	TOE					±2.0	LSB
Conversion time*2	t_{CONV}			1.5		10	μs
Zero-scale error*1	ZSE					±1.5	LSB
Full-scale error*1	FSE					±1.5	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	μA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when using the self diagnosis function	AV_{DD0} conversion	1003		1023	LSB
			2/3AV_{DD0} conversion	673	683	693	LSB
			1/2AV_{DD0} conversion	504	512	520	LSB
			1/3AV_{DD0} conversion	331	341	351	LSB
			AV_{SS0} conversion	0		20	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(b) 10-Bit Resolution, with S/H

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.5	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±2.0	LSB
Full-scale error*1	FSE					±2.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±2.0	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} conversion	671	683	695	LSB
			1/2AV_{DD0} conversion	502	512	522	LSB
			1/3AV_{DD0} conversion	329	341	353	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(c) 12-Bit Resolution, without S/H

(T_a = -40 to +110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	μs
Accuracy*1	TOE					±6.0	LSB
Conversion time*2	t_{CONV}			1.0		10	μs
Zero-scale error*1	ZSE					±5.0	LSB
Full-scale error*1	FSE					±5.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	μA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} conversion	4015		4095	LSB
			2/3AV_{DD0} conversion	2691	2731	2771	LSB
			1/2AV_{DD0} conversion	2018	2048	2078	LSB
			1/3AV_{DD0} conversion	1325	1365	1405	LSB
			AV_{SS0} conversion	0		80	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(d) 12-Bit Resolution, with S/H

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	µs
Accuracy*1	TOE					±9.0	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±7.0	LSB
Full-scale error*1	FSE					±7.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±3.5	LSB
Integral non-linearity error*1	INL					±5.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} conversion	2689	2731	2773	LSB
			1/2AV_{DD0} conversion	2016	2048	2080	LSB
			1/3AV_{DD0} conversion	1323	1365	1407	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

Caution 1. Implement the following countermeasures to minimize any adverse effect on precision of the A/D converter from the switching noise of the digital circuit.

i. Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

ii. Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, FV_{DD}, or AV_{DD0}) and ground pin (V_{SSM}, V_{SSC}, EV_{SS}, $OSCV_{SS}$, $CPUPLLV_{SS}$, FV_{SS}, or AV_{SS0}), and between the analog reference voltage pins (AV_{REF0P} or AV_{REF0M}) and the analog ground pin (AV_{SS0}).

iii. We recommend isolating the analog power supply (AV_{DD0}) from the digital power supply (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, or FV_{DD}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board. Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

Caution 2. Connect resistor Re and capacitor Ce to the analog input pins (ADCA0Im) externally to eliminate noise (m = 01 to 18).

Note Full-scale range (%FSR) is the ratio, as a percentage, of the extremes of the range of analog input voltages that can be converted, and is expressed as follows regardless of the resolution.

1%FSR = (Maximum value of convertible analog input voltage - Minimum value of convertible analog input voltage)/100

= (AV_{REF0P} - 0)/100

= AV_{REF0P}/100

1 LSB is as follows at a resolution of 10 bits:

1 LSB = $1/2^{10}$

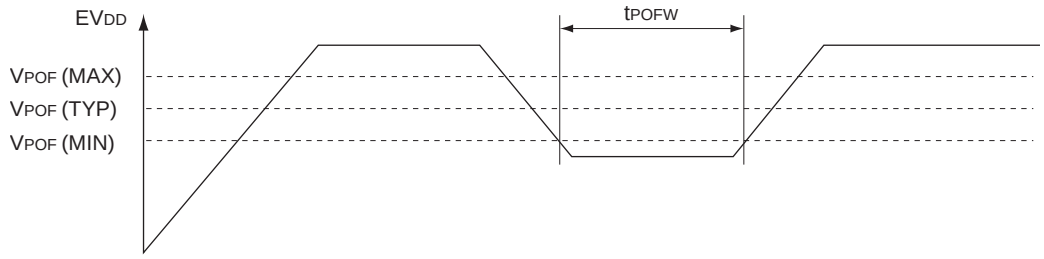
= 0.098%FSR

28.1.6.12 POF/LVI Characteristics

POF characteristics

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{POF}		2.7	2.8	2.9	V
Minimum detection width	t_{POFW}		0.2			ms

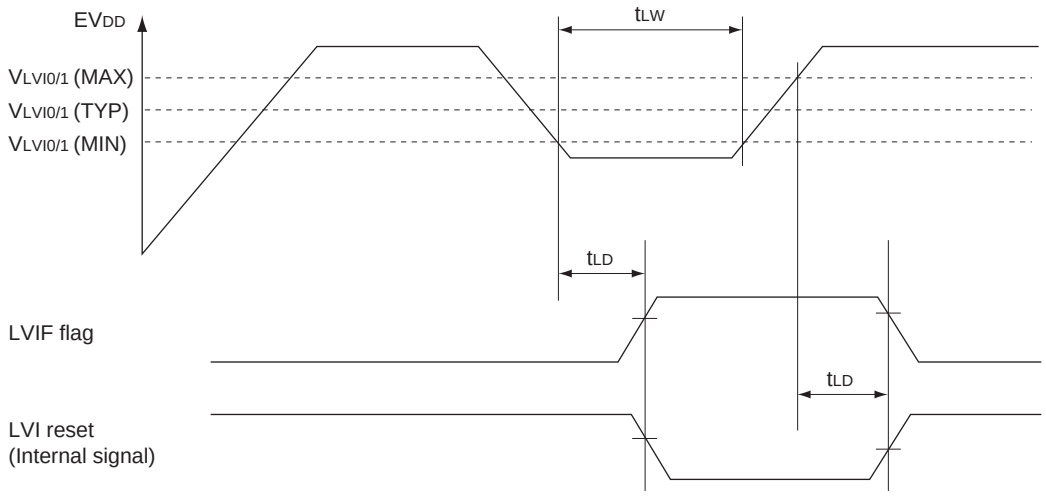


LVI characteristics

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{LVI0}	LVICNT.LVICNT[1:0] = 01_B	4.5	4.6	4.7	V
	V_{LVI1}	LVICNT.LVICNT[1:0] = 10_B	4.2	4.3	4.4	V
Response time*1	t_{LD}	After EV_{DD} reaches V_{LVI0},V_{LVI1}（MAX.） After EV_{DD} falls V_{LVI0}, V_{LVI1}（MIN.） Voltage gradient:1.8 V/us to 1.8 V/s			2	ms
Minimum detection width	t_{LW}		2.0			ms
Stabilization waiting time*2					350	µs

Note 1. Time until output of an interrupt or reset after voltage detection.
Note 2. Time until LVI is detected in accord with the specification after enabling of the LVI function.


EVDD
tLW
VLVI0/1 (MAX)
VLVI0/1 (TYP)
VLVI0/1 (MIN)
tLD
LVIF flag
tLD
LVI reset
(Internal signal)

28.1.6.13 Flash Memory Programming Characteristics

(a) Basic Characteristics

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}			64		80	MHz
Number of times rewritable (per block)*1	C_{ERWR}	Code Flash	Data are retained for 20 years			100	Times/ block
		Data Flash	Data are retained for 5 years			15000	Times/ block
			Data are retained for 15 years			5000	Times/ block
			Data are retained for 20 years			1000	Times/ block

Note 1. In writing to a delivered product, either "erasing → writing" or "writing alone" is considered as a "time" of rewriting.

Example: (P: writing, E: erasing)

Delivered product→ P→E→P→E→P: 3 times

Delivered product→E→P→E→P→E→P: 3 times

28.1.6.14 FLMD0 Pulse Timing Characteristics

(T_a = -40 to + 110°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD}$ = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (to VDD)	t_{DP}		1			ms
$\overline{\text{RESET}}$ release (to FLMD0)	t_{PR}		2			ms
$\overline{\text{RESET}}\uparrow \rightarrow$ FLMD0 set time	t_{RP}		1200			µs
FLMD0 high/low level width	t_{PW}		10		100	µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns
Serial clock input frequency	t_{SCCY}		9.8		5000	KHz
High level width	t_{WHSC}		85		50000	ns
Low level width	t_{WLSC}		85		50000	ns
SI setup time (to SCK ↑)	t_{PLHSC}		75			ns
SI hold time (to SCK ↑)	t_{PHLSC}		0			ns
SO output delay time	t_{DSC}				23	ns
UART transfer rate			9.6		1000	Kbps

Note VDD: $V_{DDM} = V_{DDC} = CPUPLLV_{DD}$

28.1.6.15 BIST Execution Time

(T_a = -40 to + 110°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = 3.0 to 5.5 V, AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition 1	Condition 2	MIN.	TYP.	MAX.	Unit
BIST execution time		All HW-BIST Total fxx = 64 MHz				26	ms
		All HW-BIST Total fxx = 80 MHz				21	ms

28.2 Electrical Characteristics of the µPD70F3506

28.2.1 Absolute Maximum Ratings

(T_a = 25°C)

Item	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DDM}	$V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 1.6	V
	V_{DDC}		-0.5 to + 1.6	V
	$CPUPLLV_{DD}$		-0.5 to + 1.6	V
	$FRPLLV_{DD}$		-0.5 to + 1.6	V
	EV_{DD}	$EV_{DD} = OSCV_{DD} = FV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 4.1	V
	$OSCV_{DD}$		-0.5 to + 4.1	V
	FV_{DD}		-0.5 to + 4.1	V
	AV_{DD0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 4.1	V
	V_{SSM}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	V_{SSC}		-0.5 to + 0.5	V
	$CPUPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	$FRPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	EV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	$OSCV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	FV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	AV_{SS0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
Input voltage	V_I		-0.5 to EV_{DD} + 0.5*1	V
Analog input voltage	V_{IAN}	ADCA0I1 to ADCA0I22	-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0P}		-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0M}		-0.5 to AV_{DD0} + 0.5*1	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total pins	-50	mA
Low-level output current	I_{OL}	1 pin	10	mA
		Total pins	50	mA
Operating ambient temperature	T_a		-40 to + 125	°C
Storage temperature	T_{stg}	Before writing to flash memory	-65 to + 150	°C
		After writing to flash memory	-65 to + 125	°C

Note 1. Be sure not to exceed the absolute maximum ratings of each supply voltage.

Caution 1. Make sure that the total the current flowing through these pins and output currents does not exceed 50 mA.

Caution 2. Do not directly connect output (or I/O) pins of IC products to each other, or to VDD, VCC, and GND. Open drain pins or open collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

Caution 3. Product quality may suffer if the absolute maximum rating is exceeded even momentary for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

28.2.2 Capacity

(T_a = 25°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz 0 V for non-measured pins, and excluding analog pins selected for use with S/H channels or AD converter channels.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

28.2.3 Operating Conditions

(T_a = -40 to +125°C, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Internal Operation Clock Frequency	Operating Ambient Temperature (T_a)	Power Supply Voltage
fxx = 64 MHz	T_a = -40 to +125°C	V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V
fxx = 80 MHz*1		

Note 1. Set PCLK to 64 MHz or a lower frequency when using the FlexRay function.

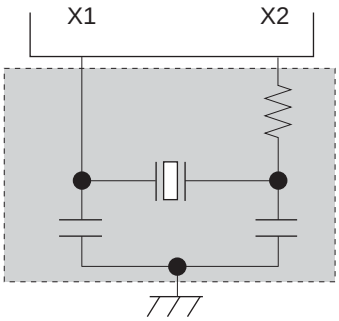
28.2.4 Characteristics of Oscillation Circuit

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency*1	fx		16	16	16	MHz

Note 1. The values are only applicable as oscillation circuit characteristics.

Caution 1. When placing wiring in the area enclosed by the broken line in the figure below during the design of the oscillator, avoid adverse effects from wiring capacitance by following the directions below.



- Bring the oscillation circuit as close as possible to the X1 and X2 terminals.
- Do not pass other signal conductors through the broken line part in the above figure.
- Do not route the wiring near signal lines through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as $OSCV_{SS}$.
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When selecting the crystal oscillator, designing the parameters of the oscillation circuit, or calculating the time required for oscillation to become stable, exhaustively evaluate the finished system (MCU, oscillator, and target board) for matching of the device and the oscillator or ask the manufacturer of the oscillator to do so.

28.2.5 DC Characteristics

DC Characteristics (1/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$= $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-level input voltage	V_{IH1}	Port pins, FLMD0 pin	0.7 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH2}	FLMD1 pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH3}*5	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.0		EV_{DD} + 0.3	V
	V_{IH4}	$\overline{\text{RESET}}$ pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
Low-level input voltage	V_{IL1}	Port pins, FLMD0 pin	-0.5		0.3 × EV_{DD}	V
	V_{IL2}	FLMD1 pin	-0.5		0.2 × EV_{DD}	V
	V_{IL3}*5	DCUTDI, DCUTMS DCUTCK	-0.5		0.8	V
	V_{IL4}	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
Hysteresis width*1	V_{HYS1}	Port pins, FLMD0 pin		0.5		V
	V_{HYS2}	FLMD1 pin		1.0		V
	V_{HYS3}	$\overline{\text{RESET}}$ pin		1.0		V
High-level output voltage*2	V_{OH1}	I_{OH} = - 3 mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 1 mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 0.1 mA	EV_{DD} - 0.5		EV_{DD}	V
Low-level output voltage*2	V_{OL1}	I_{OL} = 3 mA	0		0.4	V
		I_{OL} = 1 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V, other than for the pins below	20	40	100	KΩ
	R_2	V_I = 0 V*6	10	19	50	KΩ
Pull-down resistor	R_3	V_I = EV_{DD}*3	20	40	100	KΩ
	R_4	V_I = EV_{DD}*4	25	40	100	KΩ
	R_5	V_I = EV_{DD}*6	10	19	50	KΩ

Note 1. When the operation of an adjacent pin is halted.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.
Note 3. Only for the $\overline{\text{FLMD1}}$ and $\overline{\text{DCUTRST}}$ pins.
Note 4. Only for the $\overline{\text{RESET}}$ pin.
Note 5. In debugging mode
Note 6. Only for the FLMD0 pin

Caution The characteristics of the alternative function pins are the same as those of the port pin unless otherwise specified.

DC Characteristics (2/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-level input leakage current	I_{LIH1}	V_{IAN} = AV_{DD0}	Analog pins*1			0.2	µA
		V_I = EV_{DD}	Pins other than analog pins*2			5	µA
Low-level input leakage current	I_{LIL1}	V_{IAN} = 0 V	Analog pins*1			-0.2	µA
		V_I = 0 V	Pins other than analog pins*2			-5	µA
High-level output leakage current	I_{LOH1}	V_I = EV_{DD}				5	µA
Low-level output leakage current	I_{LOL1}	V_I = 0 V				-5	µA

Note 1. ADCA0I1 to ADCA0I22
Note 2. The current flowing through the pull-up/pull-down resistors is not included.

DC Characteristics (3/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power-supply current*1	Normal operating conditions	I_{DD}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 48 MHz	PCLK = 1/2HEAPCLK			240	mA
					PCLK = HEAPCLK			260	mA
				fxx = 64 MHz	PCLK = 1/2HEAPCLK			260	mA
					PCLK = HEAPCLK			300	mA
				fxx = 80 MHz	PCLK = 1/2HEAPCLK			299	mA
					PCLK = HEAPCLK			330	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA
	When rewriting flash memory	I_{DDF}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx = 64 MHz				280	mA
				fxx = 80 MHz				320	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA
	When executing BIST	I_{DDB}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx = 48MHz				240	mA
				fxx = 64 MHz				290	mA
				fxx = 80 MHz				340	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})					30	mA
	When HALT	I_{DDH}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx = 64 MHz	PCLK = 1/2HEAPCLK			230	mA
					PCLK = HEAPCLK			280	mA
				fxx = 80 MHz	PCLK = 1/2HEAPCLK			260	mA
					PCLK = HEAPCLK			310	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA

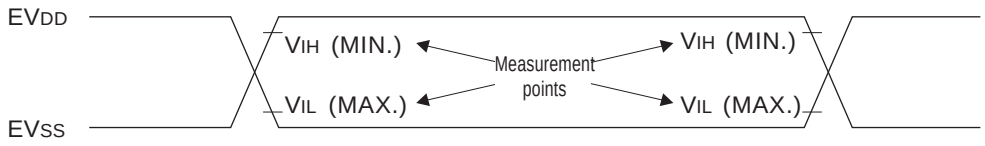
Note 1. The current flowing through the pull-up/pull-down resistors is not included.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.

Note MAX. is the maximum value of average current.

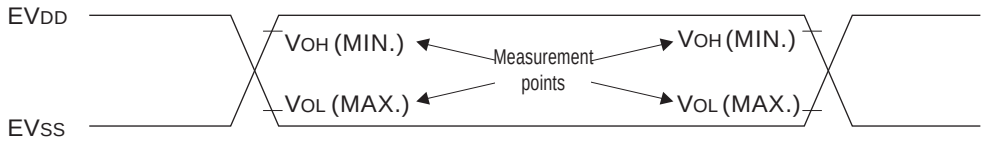
28.2.6 AC Characteristics

28.2.6.1 Measurement Conditions

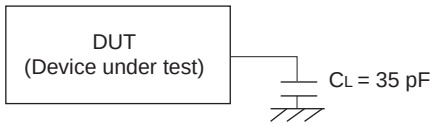
Input waveform for AC testing



AC test output measurement points



Load conditions

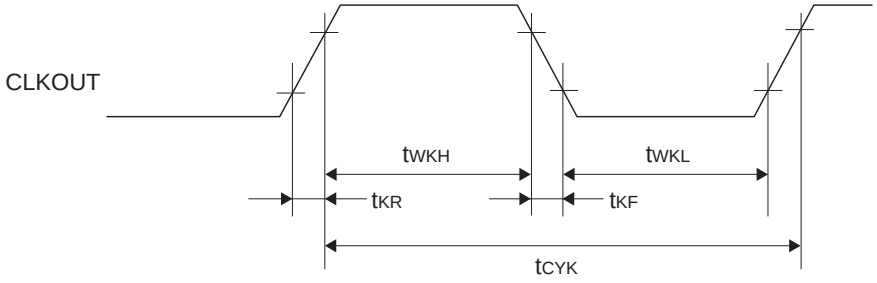


Caution If the load capacitance exceeds 35 pF due to the circuit configuration, bring the load capacitance of the device to 35 pF or less by inserting a buffer or by some other means.

28.2.6.2 Clock Timing (CLKOUT)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CLKOUT output cycle time	t_{CYK}		0.1	4	μs
CLKOUT high level width	t_{WKH}		t_{CYK}/2 - 13		ns
CLKOUT low level width	t_{WKL}		t_{CYK}/2 - 13		ns
CLKOUT rise time	t_{KR}			13	ns
CLKOUT fall time	t_{KF}			13	ns



28.2.6.3 Timing in Turning the Power Supply On and Off

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
V_{DDM},V_{DDC} ↑ to EV_{DD} ↑	t_{RLI}		0	1000	ms
EV_{DD} ↑ to V_{DDM}, V_{DDC} ↑	t_{RIL}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to AV_{DD0} ↑	t_{RLA}		0	1000	ms
AV_{DD0} ↑ to V_{DDM}, V_{DDC} ↑	t_{RAL}		0	1000	ms
EV_{DD} ↑ to AV_{DD0} ↑	t_{RIA}		0	1000	ms
AV_{DD0} ↑ to EV_{DD} ↑	t_{RAI}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to $\overline{RESET}$ ↑	t_{RLR}	*1	t_{OSC} + 0.5		ms
EV_{DD} ↑ to $\overline{RESET}$ ↑	t_{RIR}	*1	t_{OSC} + 0.5		ms
AV_{DD0} ↑ to $\overline{RESET}$ ↑	t_{RAR}	*1	t_{OSC} + 0.5		ms
FLMD1,0 setup time (to $\overline{RESET}$ ↑)	t_{RMR0}		0		ns
FLMD1,0 hold time(to $\overline{RESET}$ ↓)	t_{FRM0}	*2	0		ns
$\overline{RESET}$ ↓ to V_{DDM}, V_{DDC} ↓	t_{FRL}		500		ns
$\overline{RESET}$ ↓ to EV_{DD} ↓	t_{FRI}		500		ns
$\overline{RESET}$ ↓ to AV_{DD0} ↓	t_{FRA}		500		ns
EV_{DD} ↓ to V_{DDM}, V_{DDC} ↓	t_{FIL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to EV_{DD} ↓	t_{FLI}		0	1000	ms
AV_{DD0} ↓ to V_{DDM}, V_{DDC} ↓	t_{FAL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to AV_{DD0} ↓	t_{FLA}		0	1000	ms
AV_{DD0} ↓ to EV_{DD} ↓	t_{FAI}		0	1000	ms
EV_{DD} ↓ to AV_{DD0} ↓	t_{FIA}		0	1000	ms

Note 1. t_{OSC}: Oscillation stabilization time

Note 2. Using the specified value is recommended for the FLMD1, 0 hold time (time until $\overline{RESET}$ falls: t_{FRM0}). If using the recommended value is difficult, change the FLMD1, 0 pins once rewriting of the flash memory is complete.

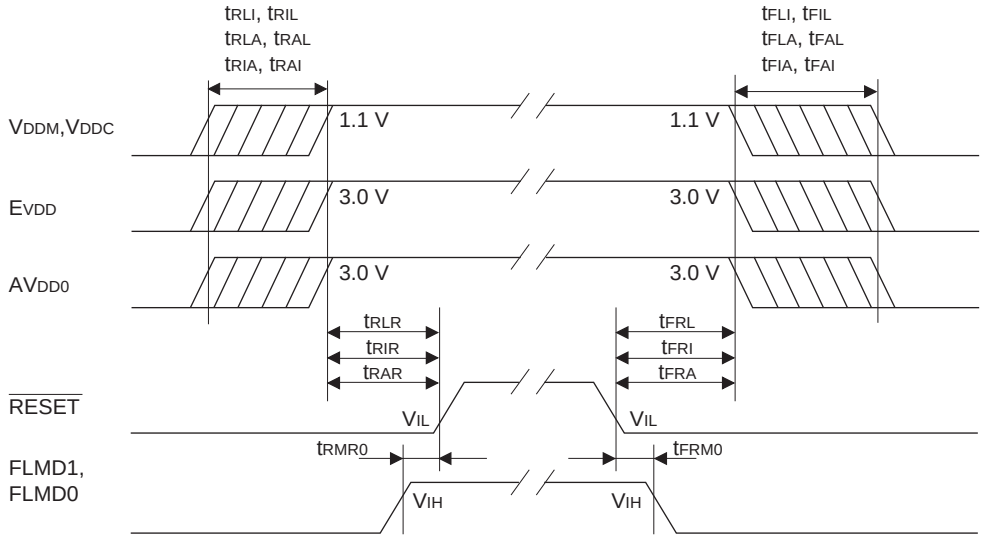
Caution 1. Turn on/off internal power system at the same time.

Caution 2. Turn on/off external power system at the same time.

Caution 3. Apply an external reset ($\overline{RESET}$) whenever the power supply voltage falls below the operating voltage.

Caution 4. If the power-on flag or self-diagnosis (BIST) control registers are to be used, boot the internal power system first (when the internal power supply voltage reaches 1.1 V, set the voltage of the external power system to a voltage below the detection voltage [min: 2.7 V] for POF).

Note Internal power system: V_{DDM}, V_{DDC}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$
External power system: EV_{DD}, $OSCV_{DD}$, FV_{DD}


tRLI, tRIL
tRLA, tRAL
tRIA, tRAI
tFLI, tFIL
tFLA, tFAL
tFIA, tFAI
VDDM,VDDC
EVDD
AVDD0
1.1 V
3.0 V
3.0 V
1.1 V
3.0 V
3.0 V
tRLR
tRIR
tRAR
tFRL
tFRI
tFRA
RESET
VIL
VIL
tRMR0
tFRM0
FLMD1,
FLMD0
VIH
VIH

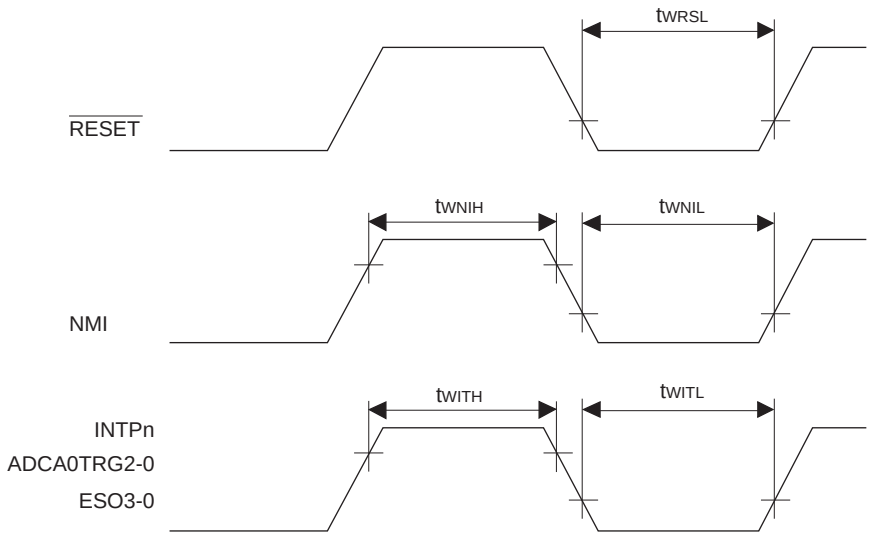
28.2.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}	Other than power supply turning on	500		ns
NMI input high level width	t_{WNIH}	Digital filter	*1		ns
NMI input low level width	t_{WNIL}	Digital filter	*1		ns
INTPn input high level width	t_{WITH}	Digital filter, n = 0 to 10	*1		ns
INTPn input low level width	t_{WITL}	Digital filter, n = 0 to 10	*1		ns
ADCA0TRG2-0 input high level width	t_{WITH}	Analog filter	250		ns
ADCA0TRG2-0 input low level width	t_{WITL}	Analog filter	250		ns
ESO3-0 input high level width	t_{WITH}	Analog filter	250		ns
ESO3-0 input low level width	t_{WITL}	Analog filter	250		ns

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.



Note n = 0 to 10

28.2.6.5 Timer Timing

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
TI input high level width	t_{TIH}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3, TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TI input low level width	t_{TIL}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3 TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TO output cycle time	t_{TCYK}	TAUA0O0-15, TAUA1O0-15 TAUJ0O0-3, TAUJ1O0-3 OSTM0O, OSTM1O TSG20O0-7, TSG21O0-7, TPB0O, TPB1O	0.25	1000000	µs

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.

28.2.6.6 CSIHn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYM}	Output	125		ns
CSIHnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	$(t_{KCYM}/2) - 20$		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIM}		30		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIM}		10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOM}			10	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOM}	CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 0/0 or 1/1	t_{KWHM}		ns
		CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 1/0 or 0/1	t_{KWLM}		ns
CSIHnRYI setup time (to CSIHnSC)	t_{SRYI}		$(2 \times t_{PCLK}) + 30$		ns
CSIHnCSx setup time	t_{SSCSB0}	CSIHnCFGx.CSIHnDAPx = 0	$CS_{SETUP} \times t_{KCYM} - 10$		ns
	t_{SSCSB1}	CSIHnCFGx.CSIHnDAPx = 1	$(CS_{SETUP} + 0.5) \times t_{KCYM} - 10$		ns
CSIHnCSx hold time	t_{HSCSB0}	CSIHnCTL1.CSIHnSIT = 0	$CS_{HOLD} \times t_{KCYM} - 10$		ns
	t_{HSCSB1}	CSIHnCTL1.CSIHnSIT = 1	$(CS_{HOLD} + 0.5) \times t_{KCYM} - 10$		ns
CSIHnCSx inactive level width	t_{WSCSB}	*[1]	$(CS_{IDLE} + 0.5) \times t_{KCYM} - 20$		ns
		Other than the above	$CS_{IDLE} \times t_{KCYM} - 20$		ns

Note 1. When the setting of CSIHnCFGx.CSIHnCKPx is changed while CS_{IDLE} = 000_B.

Note n = 0,1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

CS_{SETUP}: CSIHnCFGx.CSIHnSPx[3:0] bits

CS_{HOLD}: CSIHnCFGx.CSIHnHDx[3:0] bits

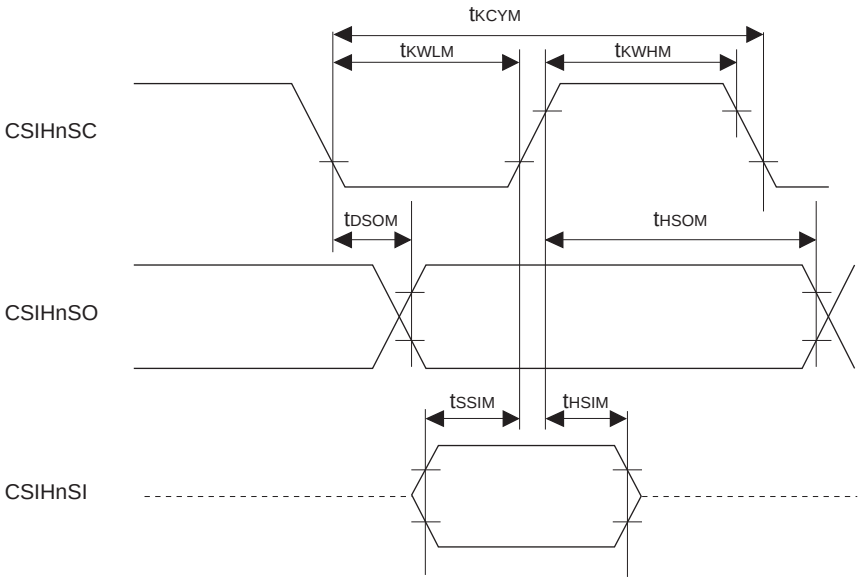
CS_{IDLE}: CSIHnCFGx.CSIHnIDx[2:0] bits

t_{PCLK}: PCLK cycle time

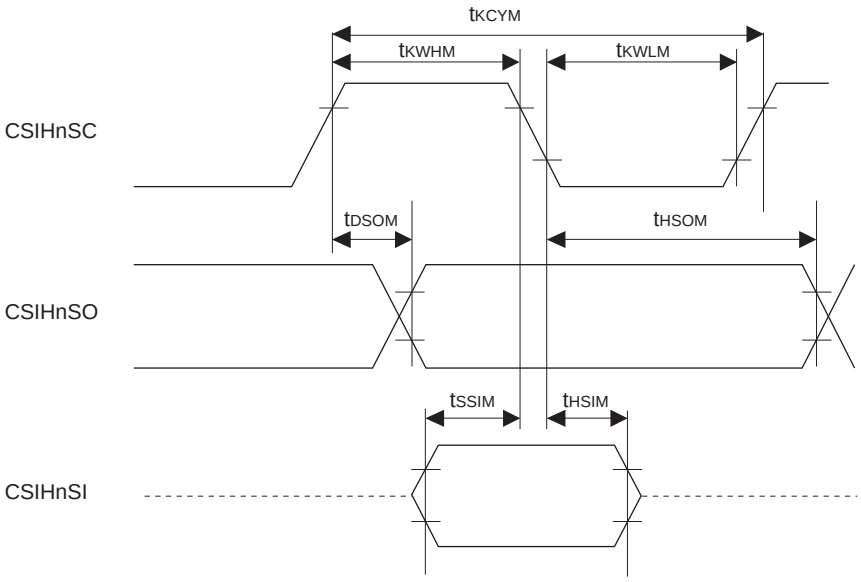
[CSIHnSC/CSIHnSI/CSIHnSO]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0 or 1/1)



Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 1/0 or 0/1)

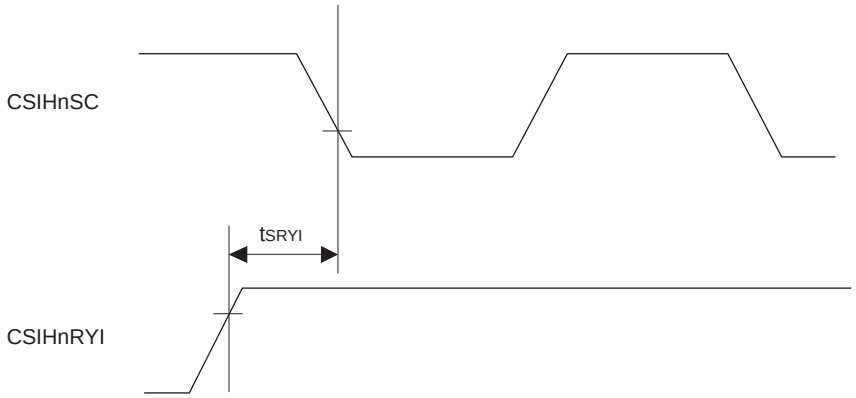


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

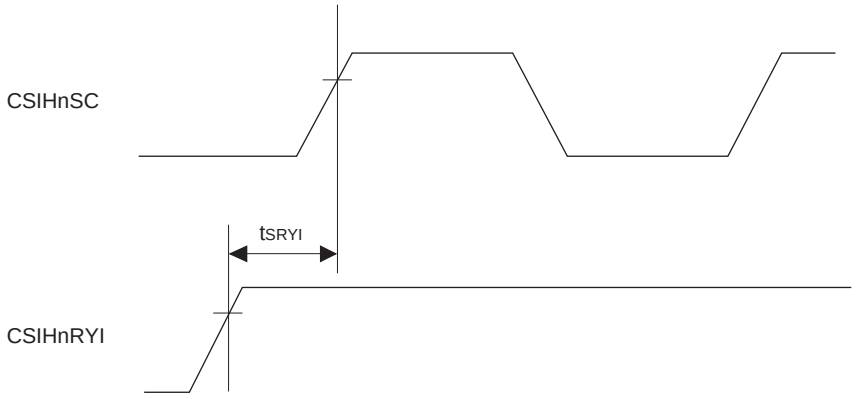
[CSIHnSC/CSIHnRYI]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx = 0)



Master mode: (CSIHnCFGx: CSIHnCKPx = 1)

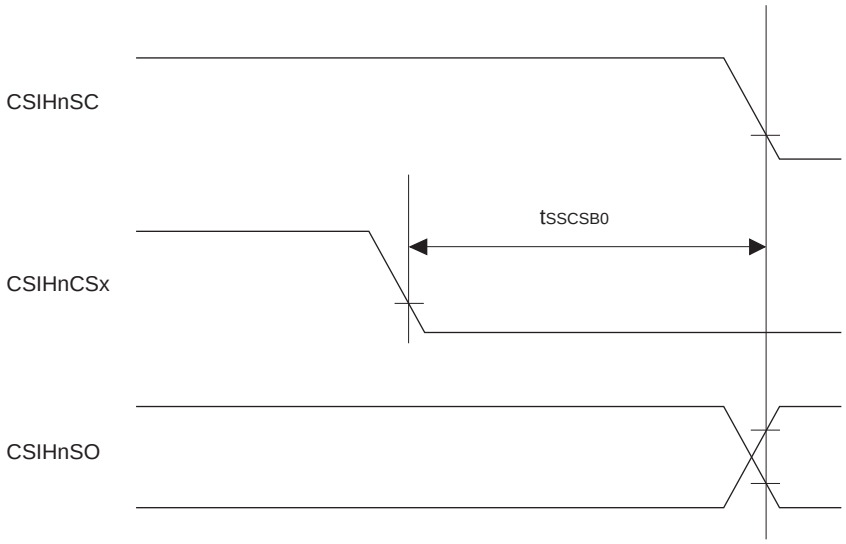


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

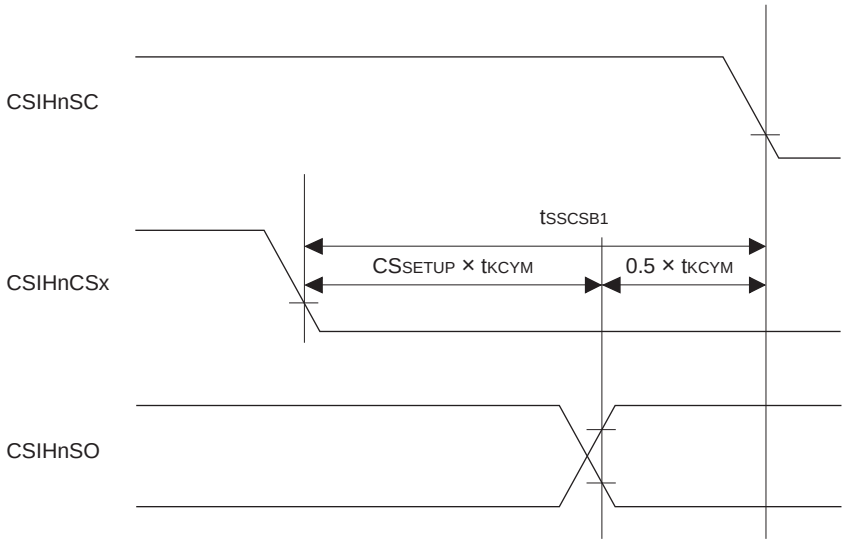
[CSIHnSC/CSIHnSO/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/1)

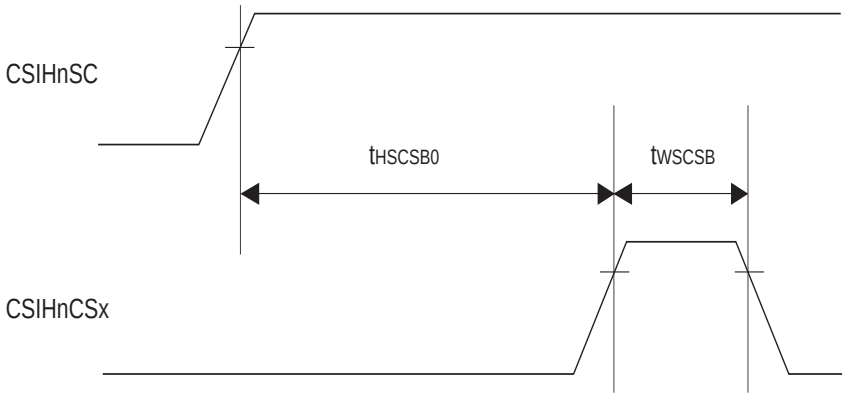


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

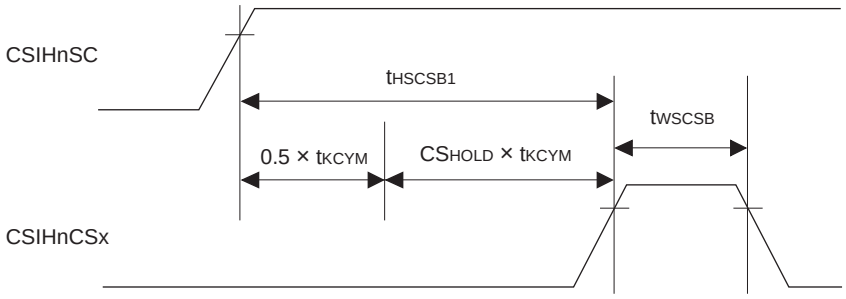
[CSIHnSC/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

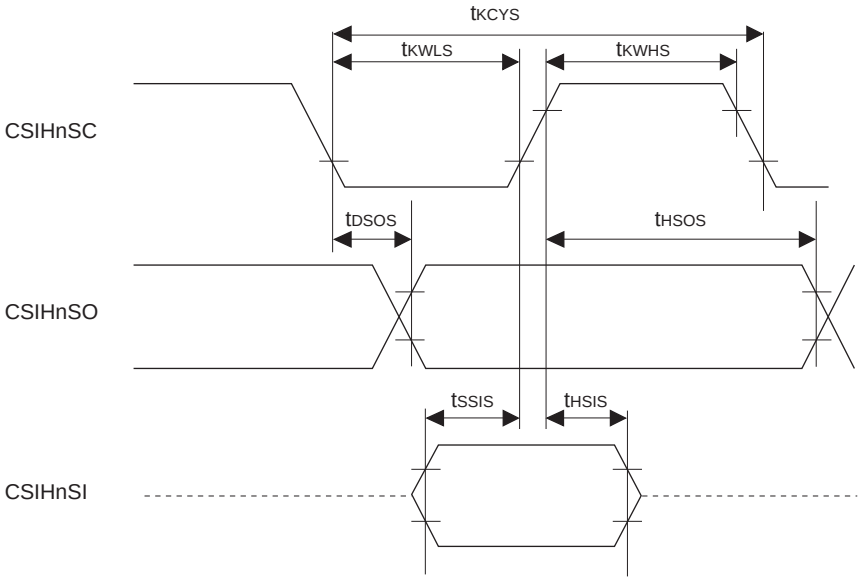
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIHnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIS}			15		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOS}				30	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOS}	CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 0/0 or 1/1		t_{KWHS}		ns
		CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 1/0 or 0/1		t_{KWLS}		ns
CSIHnRYO output delay	t_{SRYO}				30	ns

Note n = 0, 1

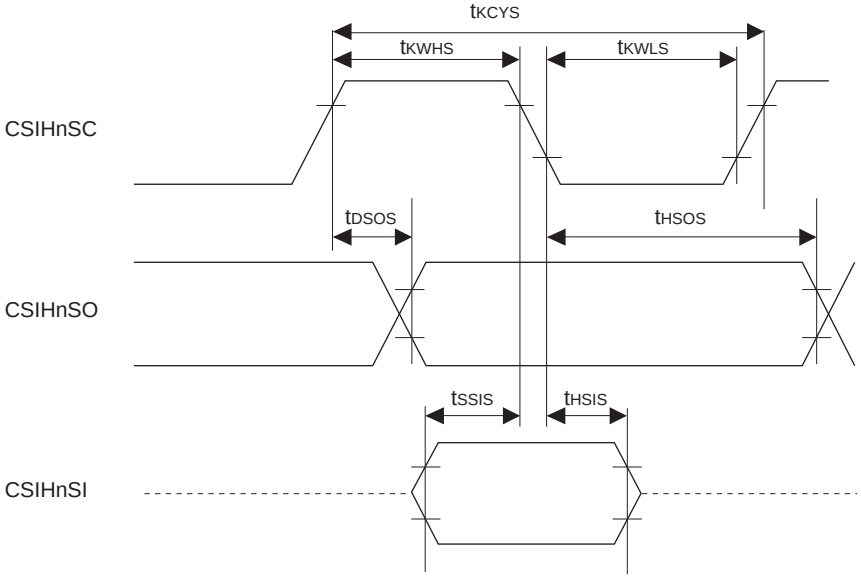
t_{PCLK}: PCLK cycle time

[CSIHnSC/CSIHnSI/CSIHnSO]

Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



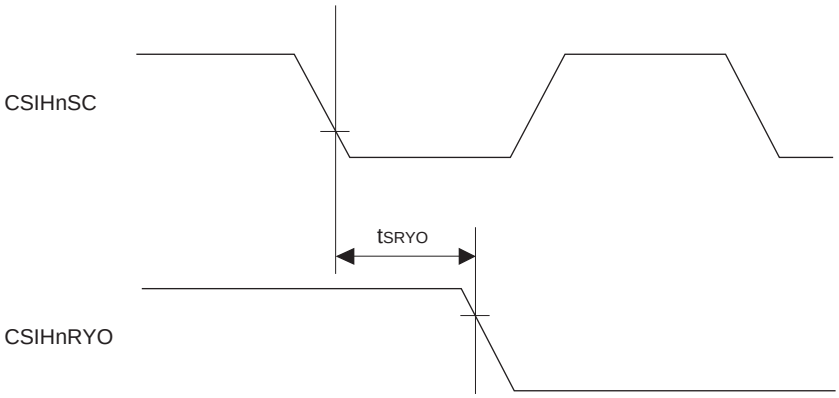
Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 1/0 or 0/1)



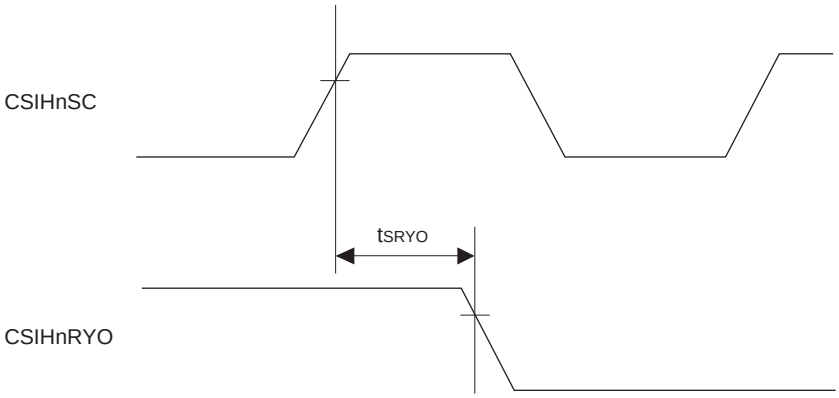
Note n = 0, 1

[CSIHnSC/CSIHnRYO]

(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/1 or 1/0)



Note n = 0, 1

28.2.6.7 CSIGn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

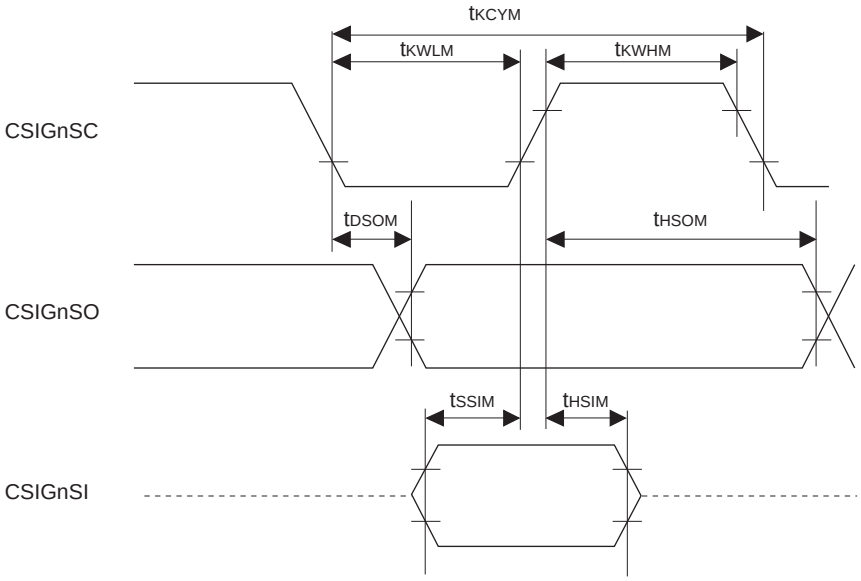
Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYM}	Output	125		ns
CSIGnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIM}		30		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIM}		10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOM}			10	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOM}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1	t_{KWHM}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1	t_{KWLM}		ns
CSIGnRYI setup time (to CSIGnSC)	t_{SRYI}		(2 × t_{PCLK}) + 30		ns

Note n = 0 to 2

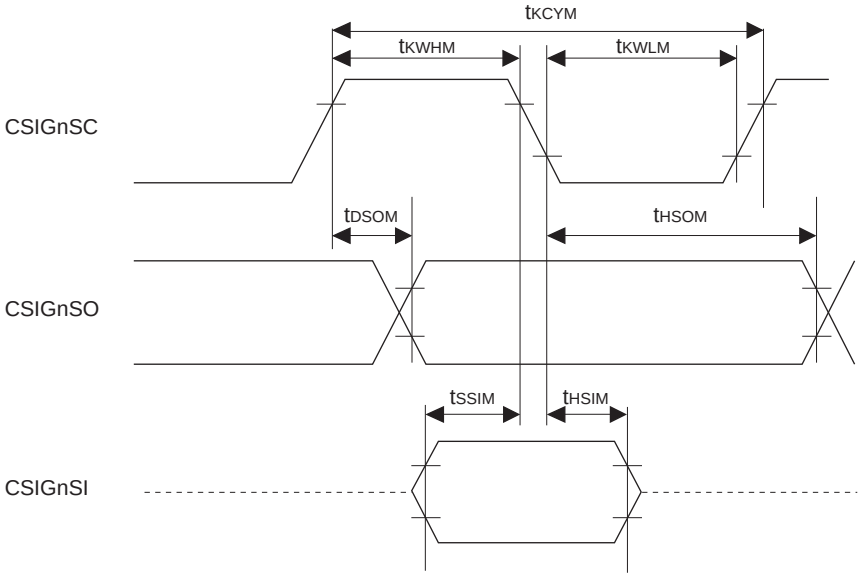
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



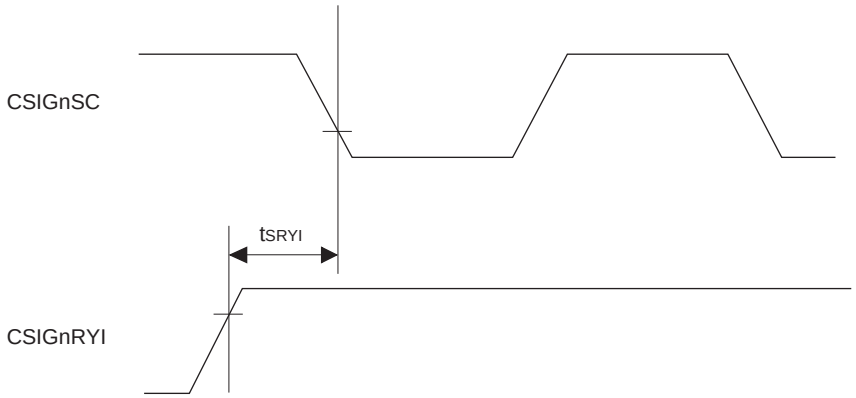
Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



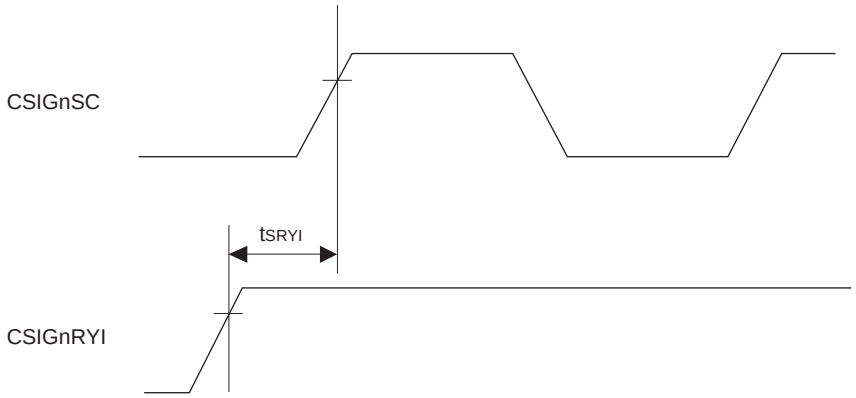
Note n = 0 to 2

[CSIGnSC/CSIGnRYI]

Master mode: (CSIGnCTL1: CSIGnCKR = 0)



Master mode: (CSIGnCTL1: CSIGnCKR = 1)



Note n = 0 to 2

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

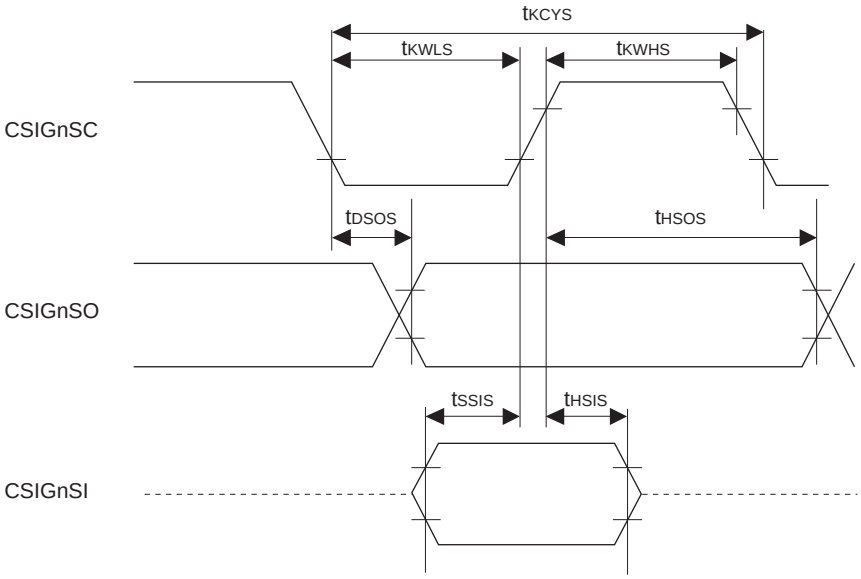
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIGnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIS}			15		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOS}				30	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOS}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1		t_{KWHS}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1		t_{KWLS}		ns
CSIGnRYO output delay	t_{SRYO}				30	ns

Note n = 0 to 2

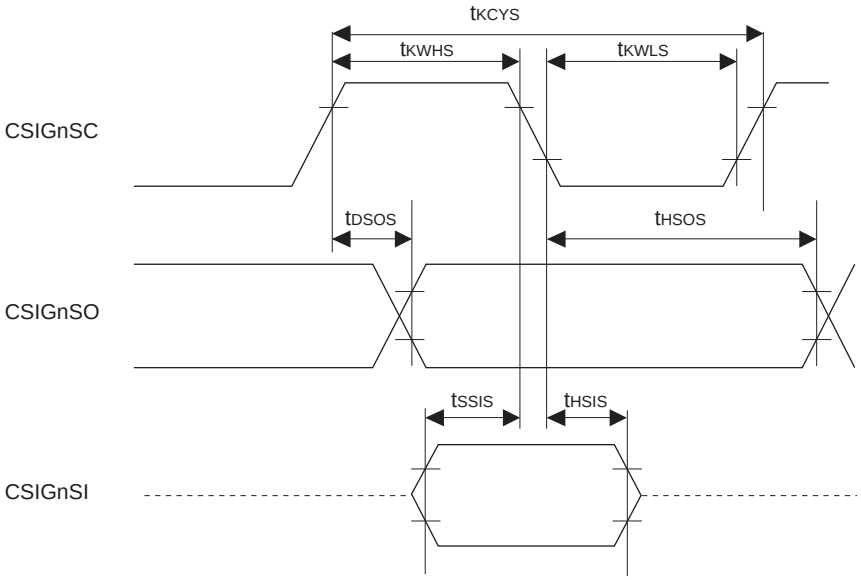
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



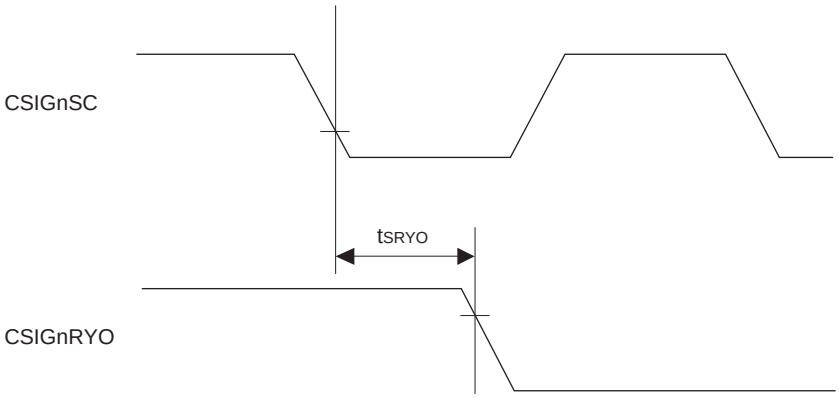
Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



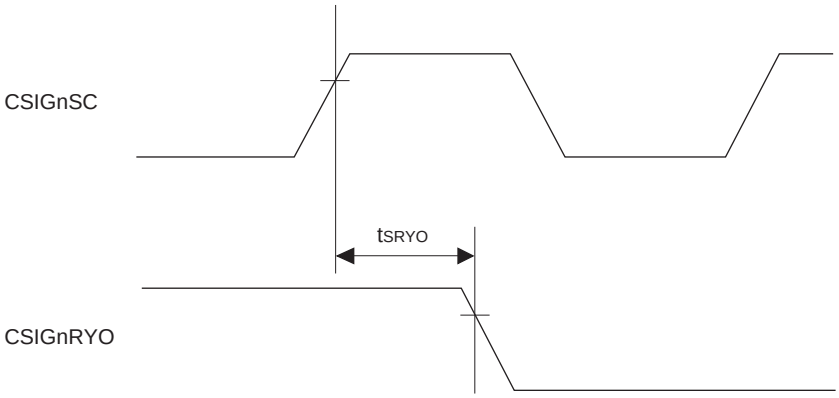
Note n = 0 to 2

[CSIGnSC/CSIGnRYO]

(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1 or 1/0)



Note n = 0 to 2

28.2.6.8 UARTHn Timing

(a) Asynchronous Mode

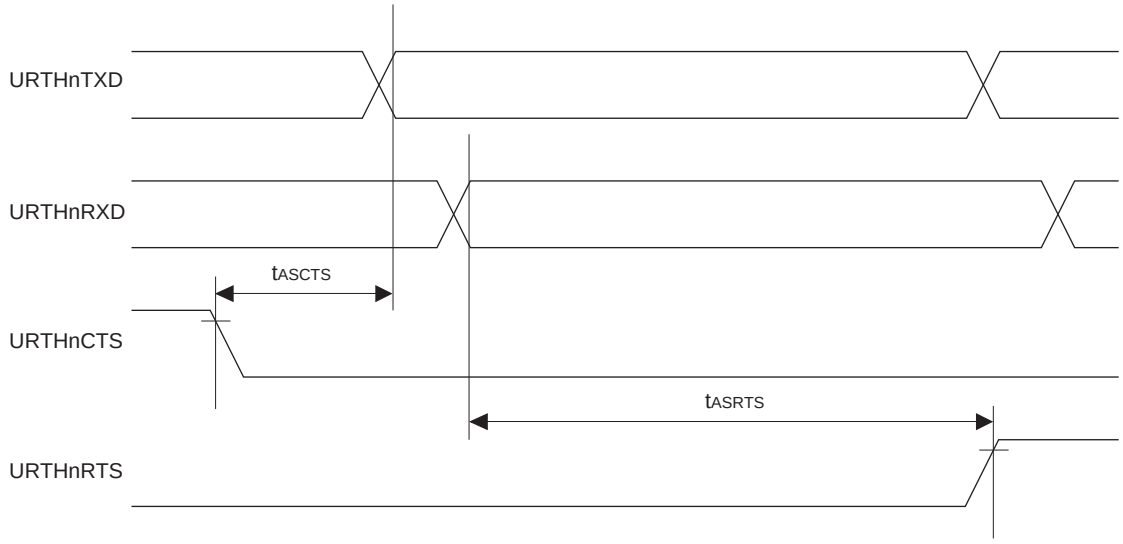
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		MIN.	MAX.	Unit
UARTHn transfer rate				16	2 M	bps
URTHnCTS setup time (to URTHnTXD)	t_{ASCTS}	Input	URTHnCTS without noise filter	3T + 30		ns
			URTHnCTS with noise filter	5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{ASRTS}	Output	URTHnRXD without noise filter		(3BRS + 2) T + 30	ns
			URTHnRXD with noise filter		(3BRS + 4) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]



(b) Clock Synchronous, Master Mode (Normal Mode)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM1}	Output		500		500		ns
URTHnSC high level width	t_{KWHM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnSC low level width	t_{KWLM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDM1}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDM1}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM1}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM1}	Output		t_{KCYM1}/2 - 10		t_{KCYM1}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM1}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM1}	Output			T + 30		T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

(c) Clock Synchronous, Slave Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35pF)

Item	Symbol	Conditions		URTHnSC/URTHnRXD without Noise Filter		URTHnSC/URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		10		10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		2T + 10		2T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		-T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	2T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		2T+ 30	3T + 30	2T + 30	3T + 30	ns

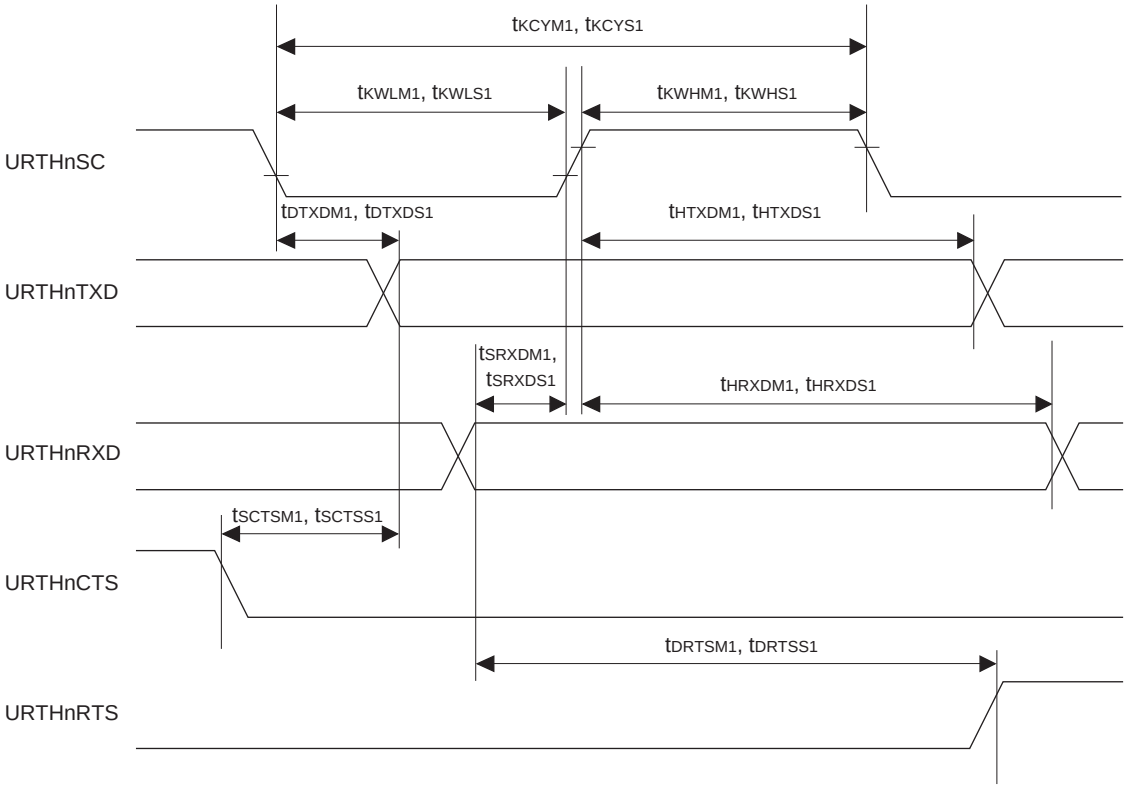
Note n = 0 to 2

T = prescaler cycle time

Item	Symbol	Conditions		URTHnSC without Noise Filter URTHnRXD with Noise Filter		URTHnSC with Noise Filter URTHnRXD without Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		2T + 10		-2T + 10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		10		4T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	3T + t_{KCYS1} + 30		3T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		4T + 30	5T + 30	30	T + 30	ns

Note n = 0 to 2

T = prescaler cycle time


tKCYM1, tKCYS1
tKWLM1, tKWLS1
tKWHM1, tKWHS1
URTHnSC
tDTXDM1, tDTXDS1
tHTXDM1, tHTXDS1
URTHnTXD
tSRXDM1,
tSRXDS1
tHRXDM1, tHRXDS1
URTHnRXD
tSCTSM1, tSCTSS1
URTHnCTS
tDRTSM1, tDRTSS1
URTHnRTS

Note n = 0 to 2

(d) Clock Synchronous, Master Mode (Delay Sampling Mode)

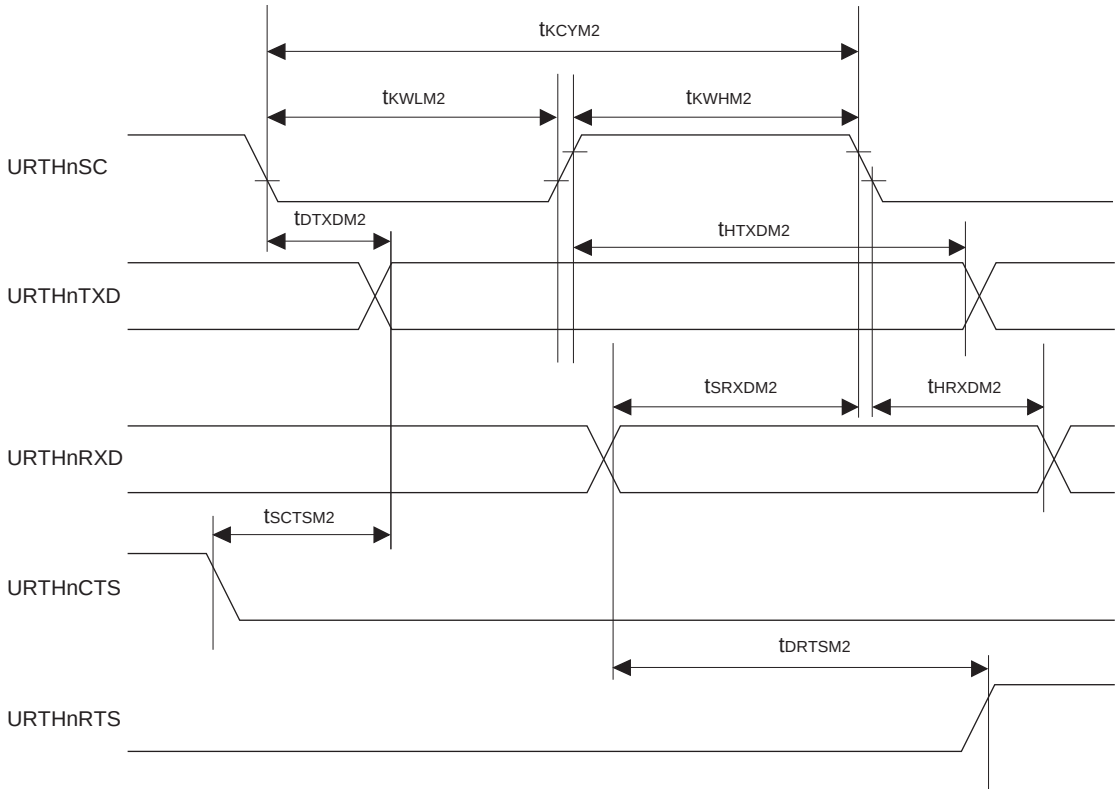
(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM2}	Output		500		500		ns
URTHnSC high level width	t_{KWHM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnSC low level width	t_{KWLM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↓)	t_{SRXDM2}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↓)	t_{HRXDM2}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM2}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM2}	Output		t_{KCYM2}/2 - 10		t_{KCYM2}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM2}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM2}	Output			(BRS + 1) T + 30		(BRS + 1) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]


tKCYM2
tKWLM2
tKWHM2
URTHnSC
tDTXDM2
tHTXDM2
URTHnTXD
tSRXDM2
tHRXDM2
URTHnRXD
tSCTSM2
URTHnCTS
tDRTSM2
URTHnRTS

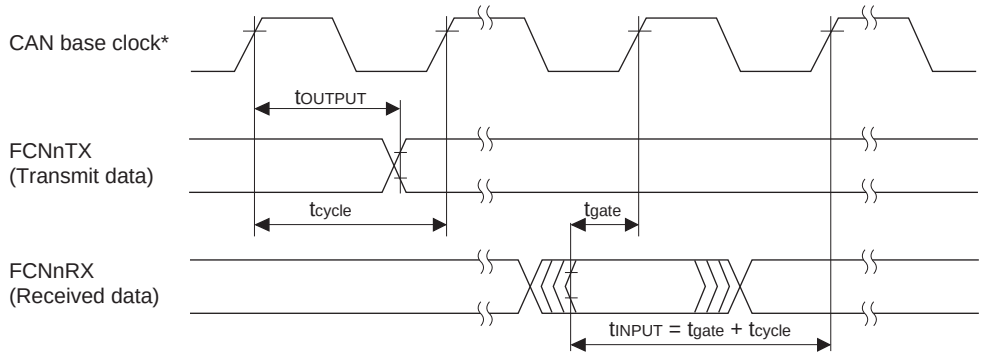
Note n = 0 to 2

28.2.6.9 CAN Timing

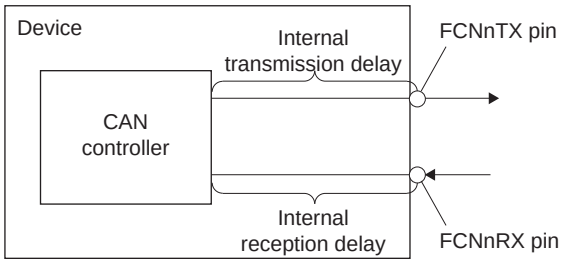
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CAN transfer rate				1	Mbps
CAN internal delay time*[1]	t_{node}			87.5	ns

Note 1. CAN internal delay time (t_{node}) = t_{OUTPUT} (internal transmission delay) + t_{INPUT} (internal reception delay)



Note * CAN base clock frequency: f_{CAN}



Schematic view of the CAN-internal delay

Note n = 0, 1

28.2.6.10 FlexRay Timing

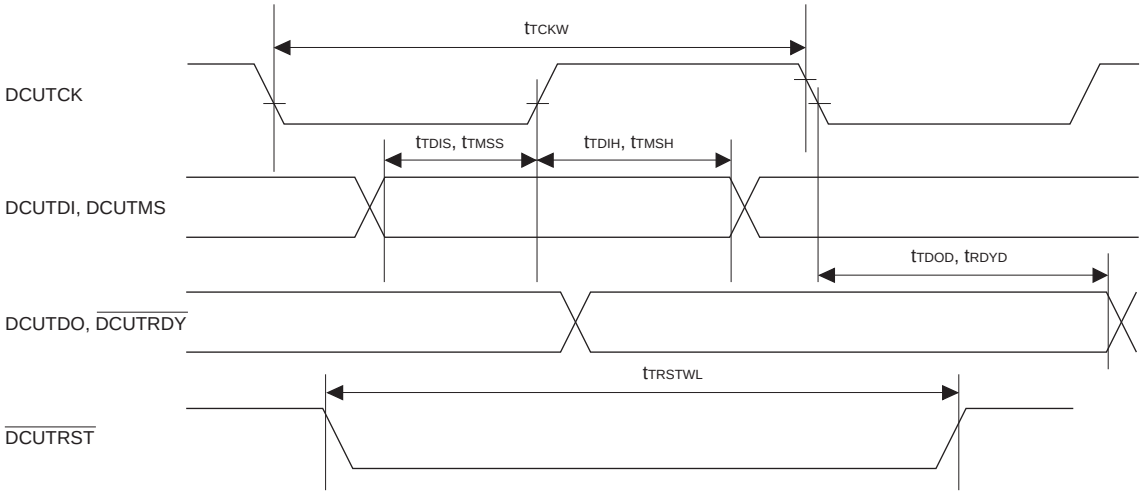
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				10	Mbps

28.2.6.11 NEXUS I/F Timing

(T_a = 0 to +40°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.
DCUTCK cycle time	t_{TCKW}	40.0		ns
DCUTDI setup time (to DCUTCK ↑)	t_{TDIS}	t_{TCKW}/2 - 8.5		ns
DCUTMS setup time (to DCUTCK ↑)	t_{TMSS}	t_{TCKW}/2 - 8.5		ns
DCUTDI hold time (to DCUTCK ↑)	t_{TDIH}	2.0		ns
DCUTMS hold time (to DCUTCK ↑)	t_{TMSH}	2.0		ns
DCUTDO output delay time (to DCUTCK ↓)	t_{TDOD}		20.0	ns
$\overline{\text{DCUTRDY}}$ output delay time (to DCUTCK ↓)	t_{RDYD}		20.0	ns
$\overline{\text{DCUTRST}}$ low level width	t_{TRSTWL}	1000		ns



28.2.6.12 A/D Converter Characteristics

(a) 10-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.0	LSB
Conversion time*2	t_{CONV}			1.5		10	ms
Zero-scale error*1	ZSE					±1.5	LSB
Full-scale error*1	FSE					±1.5	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} conversion	1003		1023	LSB
			2/3AV_{DD0} conversion	673	683	693	LSB
			1/2AV_{DD0} conversion	504	512	520	LSB
			1/3AV_{DD0} conversion	331	341	351	LSB
			AV_{SS0} conversion	0		20	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(b) 10-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.5	LSB
Conversion time*2	t_{CONV}			1.5		10	ms
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±2.0	LSB
Full-scale error*1	FSE					±2.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±2.0	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} conversion	671	683	695	LSB
			1/2AV_{DD0} conversion	502	512	522	LSB
			1/3AV_{DD0} conversion	329	341	353	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

Caution 1. Implement the following countermeasures to minimize any adverse effect on precision of the A/D converter from the switching noise of the digital circuit.

i. Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

ii. Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, FV_{DD}, or AV_{DD0}) and ground pin (V_{SSM}, V_{SSC}, EV_{SS}, $OSCV_{SS}$, $CPUPLLV_{SS}$, $FRPLLV_{SS}$, FV_{SS}, or AV_{SS0}), and between the analog reference voltage pins (AV_{REF0P} or AV_{REF0M}) and the analog ground pin (AV_{SS0}).

iii. We recommend isolating the analog power supply (AV_{DD0}) from the digital power supply (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, or FV_{DD}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board. Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

Caution 2. Connect resistor Re and capacitor Ce to the analog input pins (ADCA0Im) externally to eliminate noise (m = 01 to 22).

Note Full-scale range (%FSR) is the ratio, as a percentage, of the extremes of the range of analog input voltages that can be converted, and is expressed as follows regardless of the resolution.

1 %FSR = (Maximum value of convertible analog input voltage - Minimum value of convertible analog input voltage)/100

= (AV_{REF0P} - 0)/100

= AV_{REF0P}/100

1 LSB is as follows at a resolution of 10 bits:

1 LSB = $1/2^{10}$

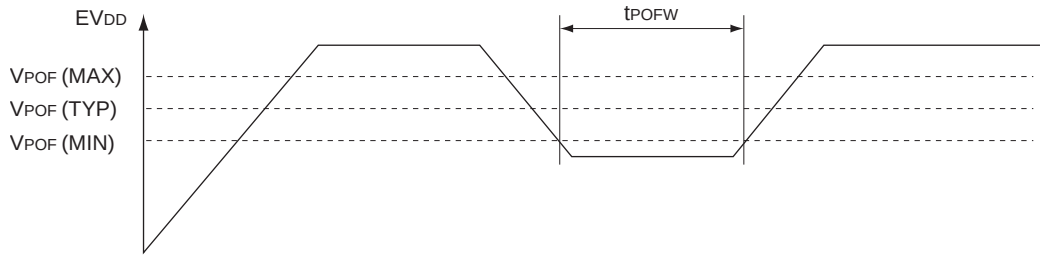
= 0.098%FSR

28.2.6.13 POF/LVI Characteristics

POF characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{POF}		2.7	2.8	2.9	V
Minimum detection width	t_{POFW}		0.2			ms

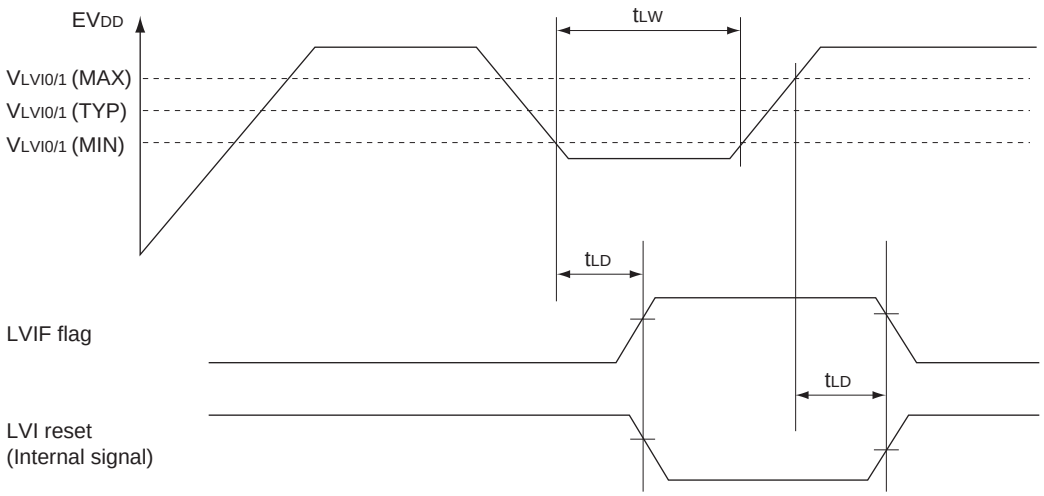


LVI characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{LVI2}	LVICNT.LVICNT[1:0]=11_B	3	3.1	3.2	V
Response time*[1]	t_{LD}	After EV_{DD} reaches V_{LVI2} (MAX.) After EV_{DD} falls V_{LVI2} (MIN.) Voltage gradient: 1.8 V/us to 1.8 V/s			2	ms
Minimum detection width	t_{LW}		2.0			ms
Stabilization waiting time*[2]					350	µs

Note 1. Time until output of an interrupt or reset after voltage detection.
Note 2. Time until LVI is detected in accord with the specification after enabling of the LVI function.


EVDD
tLW
VLVI0/1 (MAX)
VLVI0/1 (TYP)
VLVI0/1 (MIN)
tLD
LVIF flag
tLD
LVI reset
(Internal signal)

28.2.6.14 Flash Memory Programming Characteristics

(a) Basic Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}			48		80	MHz
Number of times rewritable (per block)*1	C_{ERWR}	Code Flash	Data are retained for 20 years			100	Times/ block
		Data Flash	Data are retained for 5 years			15000	Times/ block
			Data are retained for 15 years			5000	Times/ block
			Data are retained for 20 years			1000	Times/ block

Note 1. In writing to a delivered product, either "erasing → writing" or "writing alone" is considered as a "time" of rewriting.

Example: (P: writing, E: erasing)

Delivered product→ P→E→P→E→P : 3 times

Delivered product→E→P→E→P→E→P : 3 times

28.2.6.15 FLMD0 Pulse Timing Characteristics

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (to VDD)	t_{DP}		1			ms
$\overline{\text{RESET}}$ release (to FLMD0)	t_{PR}		2			ms
$\overline{\text{RESET}}\uparrow \rightarrow$ FLMD0 set time	t_{RP}		1200			µs
FLMD0 high/low level width	t_{PW}		10		100	µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns
Serial clock input frequency	t_{SCCY}		9.8		5000	KHz
High level width	t_{WHSC}		85		50000	ns
Low level width	t_{WLSC}		85		50000	ns
SI setup time (to SCK ↑)	t_{PLHSC}		75			ns
SI hold time (to SCK ↑)	t_{PHLSC}		0			ns
SO output delay time	t_{DSC}				23	ns
UART transfer rate			9.6		1000	Kbps

Note VDD: $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$

28.2.6.16 BIST Execution Time

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition 1	Condition 2	MIN.	TYP.	MAX.	Unit
BIST execution time		All HW-BIST Total fxx=48MHz				34	ms
		All HW-BIST Total fxx = 64 MHz				26	ms
		All HW-BIST Total fxx = 80 MHz				21	ms

28.3 Electrical Characteristics of the µPD70F3507

28.3.1 Absolute Maximum Ratings

(T_a = 25°C)

Item	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DDM}	$V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 1.6	V
	V_{DDC}		-0.5 to + 1.6	V
	$CPUPLLV_{DD}$		-0.5 to + 1.6	V
	$FRPLLV_{DD}$		-0.5 to + 1.6	V
	EV_{DD}	$EV_{DD} = OSCV_{DD} = FV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 6.0	V
	$OSCV_{DD}$		-0.5 to + 6.0	V
	FV_{DD}		-0.5 to + 6.0	V
	AV_{DD0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to + 6.0	V
	V_{SSM}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	V_{SSC}		-0.5 to + 0.5	V
	$CPUPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	$FRPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	EV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	$OSCV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	FV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
	AV_{SS0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to + 0.5	V
Input voltage	V_I		-0.5 to EV_{DD} + 0.5*1	V
Analog input voltage	V_{IAN}	ADCA0I1 to ADCA0I22	-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0P}		-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0M}		-0.5 to AV_{DD0} + 0.5*1	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total pins	-50	mA
Low-level output current	I_{OL}	1 pin	10	mA
		Total pins	50	mA
Operating ambient temperature	T_a		-40 to + 125	°C
Storage temperature	T_{stg}	Before writing to flash memory	-65 to + 150	°C
		After writing to flash memory	-65 to + 125	°C

Note 1. Be sure not to exceed the absolute maximum ratings of each supply voltage.

Caution 1. Make sure that the total the current flowing through these pins and output currents does not exceed 50 mA.

Caution 2. Do not directly connect output (or I/O) pins of IC products to each other, or to VDD, VCC, and GND. Open drain pins or open collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

Caution 3. Product quality may suffer if the absolute maximum rating is exceeded even momentary for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

28.3.2 Capacity

(T_a = 25°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz 0 V for non-measured pins, and excluding analog pins selected for use with S/H channels or AD converter channels.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

28.3.3 Operating Conditions

(T_a = -40 to +125°C, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Internal Operation Clock Frequency	Operating Ambient Temperature (T_a)	Power Supply Voltage
fxx = 64 MHz	T_a = -40 to +125°C	V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V
fxx = 80 MHz*1		

Note 1. Set PCLK to 64 MHz or a lower frequency when using the FlexRay function.

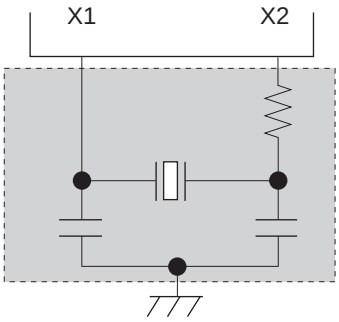
28.3.4 Characteristics of Oscillation Circuit

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency*1	fx	µPD70F3507M1	8	8	8	MHz
		µPD70F3507M2	16	16	16	MHz

Note 1. The values are only applicable as oscillation circuit characteristics.

Caution 1. When placing wiring in the area enclosed by the broken line in the figure below during the design of the oscillator, avoid adverse effects from wiring capacitance by following the directions below.



- Bring the oscillation circuit as close as possible to the X1 and X2 terminals.
- Do not pass other signal conductors through the broken line part in the above figure.
- Do not route the wiring near signal lines through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as $OSCV_{SS}$.
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When selecting the crystal oscillator, designing the parameters of the oscillation circuit, or calculating the time required for oscillation to become stable, exhaustively evaluate the finished system (MCU, oscillator, and target board) for matching of the device and the oscillator or ask the manufacturer of the oscillator to do so.

28.3.5 DC Characteristics

DC Characteristics (1/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$= $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-level input voltage	V_{IH1}	Port pins, FLMD0 pin	0.7 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH2}	FLMD1 pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH3}*5	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.2		EV_{DD} + 0.3	V
	V_{IH4}	$\overline{\text{RESET}}$ pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
Low-level input voltage	V_{IL1}	Port pins, FLMD0 pin	-0.5		0.3 × EV_{DD}	V
	V_{IL2}	FLMD1 pin	-0.5		0.2 × EV_{DD}	V
	V_{IL3}*5	DCUTDI, DCUTMS DCUTCK	-0.5		0.8	V
	V_{IL4}	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
Hysteresis width*1	V_{HYS1}	Port pins, FLMD0 pin		0.5		V
	V_{HYS2}	FLMD1 pin		1.0		V
	V_{HYS3}	$\overline{\text{RESET}}$ pin		1.0		V
High-level output voltage*2	V_{OH1}	I_{OH} = - 3 mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 1 mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 0.1 mA	EV_{DD} - 0.5		EV_{DD}	V
Low-level output voltage*2	V_{OL1}	I_{OL} = 3 mA	0		0.4	V
		I_{OL} = 1 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V, other than for the pins below	20	40	100	KΩ
	R_2	V_I = 0 V*6	10	19	50	KΩ
Pull-down resistor	R_3	V_I = EV_{DD}*3	20	40	100	KΩ
	R_4	V_I = EV_{DD}*4	25	40	100	KΩ
	R_5	V_I = EV_{DD}*6	10	19	50	KΩ

Note 1. When the operation of an adjacent pin is halted.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.
Note 3. Only for the $\overline{\text{FLMD1}}$ and $\overline{\text{DCUTRST}}$ pins.
Note 4. Only for the $\overline{\text{RESET}}$ pin.
Note 5. In debugging mode
Note 6. Only for the FLMD0 pin

Caution The characteristics of the alternative function pins are the same as those of the port pin unless otherwise specified.

DC Characteristics (2/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-level input leakage current	I_{LIH1}	V_{IAN} = AV_{DD0}	Analog pins*1			0.3	µA
		V_I = EV_{DD}	Pins other than analog pins*2			5	µA
Low-level input leakage current	I_{LIL1}	V_{IAN} = 0 V	Analog pins*1			-0.3	µA
		V_I = 0 V	Pins other than analog pins*2			-5	µA
High-level output leakage current	I_{LOH1}	V_I = EV_{DD}				5	µA
Low-level output leakage current	I_{LOL1}	V_I = 0 V				-5	µA

Note 1. ADCA0I1 to ADCA0I22
Note 2. The current flowing through the pull-up/pull-down resistors is not included.

DC Characteristics (3/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power-supply current*1	Normal operating conditions	I_{DD}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 64 MHz	PCLK = 1/2HEAPCLK			260	mA
					PCLK = HEAPCLK			300	mA
				fxx = 80 MHz	PCLK = 1/2HEAPCLK			299	mA
					PCLK = HEAPCLK			330	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA
	When rewriting flash memory	I_{DDF}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx=64MHz				280	mA
				fxx=80MHz				320	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA
	When executing BIST	I_{DDB}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx = 64 MHz				290	mA
				fxx = 80 MHz				340	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})					30	mA
	When HALT	I_{DDH}	Internal power supply current (V_{DDM},V_{DDC} + $CPUPLLV_{DD}$+ $FRPLLV_{DD}$)	fxx = 64 MHz	PCLK = 1/2HEAPCLK			230	mA
					PCLK = HEAPCLK			280	mA
				fxx = 80 MHz	PCLK = 1/2HEAPCLK			260	mA
					PCLK = HEAPCLK			310	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2				73.5 + α	mA

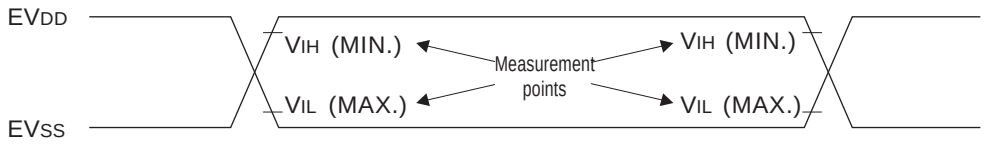
Note 1. The current flowing through the pull-up/pull-down resistors is not included.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.

Note MAX. is the maximum value of average current.

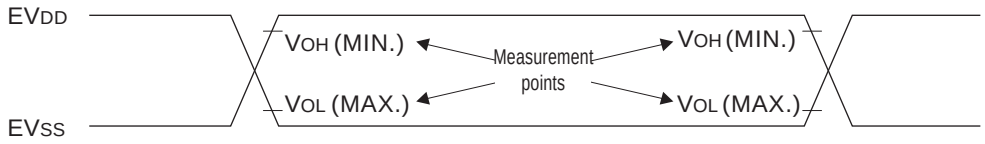
28.3.6 AC Characteristics

28.3.6.1 Measurement Conditions

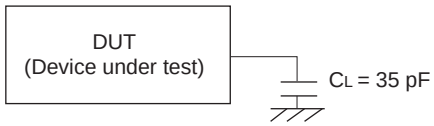
Input waveform for AC testing



AC test output measurement points



Load conditions

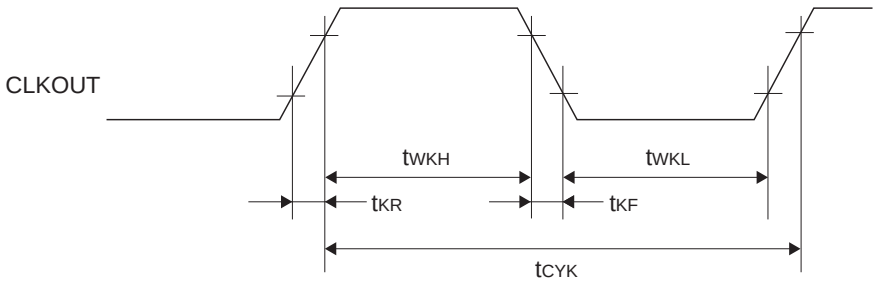


Caution If the load capacitance exceeds 35 pF due to the circuit configuration, bring the load capacitance of the device to 35 pF or less by inserting a buffer or by some other means.

28.3.6.2 Clock Timing (CLKOUT)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CLKOUT output cycle time	t_{CYK}		0.1	4	μs
CLKOUT high level width	t_{WKH}		t_{CYK}/2 - 13		ns
CLKOUT low level width	t_{WKL}		t_{CYK}/2 - 13		ns
CLKOUT rise time	t_{KR}			13	ns
CLKOUT fall time	t_{KF}			13	ns



28.3.6.3 Timing in Turning the Power Supply On and Off

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
V_{DDM},V_{DDC} ↑ to EV_{DD} ↑	t_{RLI}		0	1000	ms
EV_{DD} ↑ to V_{DDM}, V_{DDC} ↑	t_{RIL}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to AV_{DD0} ↑	t_{RLA}		0	1000	ms
AV_{DD0} ↑ to V_{DDM}, V_{DDC} ↑	t_{RAL}		0	1000	ms
EV_{DD} ↑ to AV_{DD0} ↑	t_{RIA}		0	1000	ms
AV_{DD0} ↑ to EV_{DD} ↑	t_{RAI}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to $\overline{RESET}$ ↑	t_{RLR}	*1	t_{OSC} + 0.5		ms
EV_{DD} ↑ to $\overline{RESET}$ ↑	t_{RIR}	*1	t_{OSC} + 0.5		ms
AV_{DD0} ↑ to $\overline{RESET}$ ↑	t_{RAR}	*1	t_{OSC} + 0.5		ms
FLMD1,0 setup time (to $\overline{RESET}$ ↑)	t_{RMR0}		0		ns
FLMD1,0 hold time(to $\overline{RESET}$ ↓)	t_{FRM0}	*2	0		ns
$\overline{RESET}$ ↓ to V_{DDM}, V_{DDC} ↓	t_{FRL}		500		ns
$\overline{RESET}$ ↓ to EV_{DD} ↓	t_{FRI}		500		ns
$\overline{RESET}$ ↓ to AV_{DD0} ↓	t_{FRA}		500		ns
EV_{DD} ↓ to V_{DDM}, V_{DDC} ↓	t_{FIL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to EV_{DD} ↓	t_{FLI}		0	1000	ms
AV_{DD0} ↓ to V_{DDM}, V_{DDC} ↓	t_{FAL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to AV_{DD0} ↓	t_{FLA}		0	1000	ms
AV_{DD0} ↓ to EV_{DD} ↓	t_{FAI}		0	1000	ms
EV_{DD} ↓ to AV_{DD0} ↓	t_{FIA}		0	1000	ms

Note 1. t_{OSC}: Oscillation stabilization time

Note 2. Using the specified value is recommended for the FLMD1, 0 hold time (time until $\overline{RESET}$ falls: t_{FRM0}). If using the recommended value is difficult, change the FLMD1, 0 pins once rewriting of the flash memory is complete.

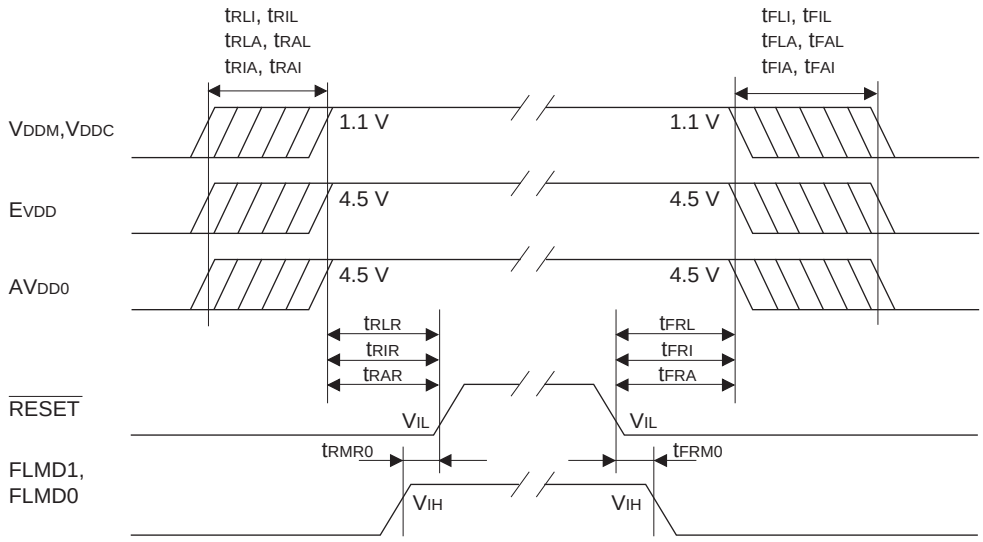
Caution 1. Turn on/off internal power system at the same time.

Caution 2. Turn on/off external power system at the same time.

Caution 3. Apply an external reset ($\overline{RESET}$) whenever the power supply voltage falls below the operating voltage.

Caution 4. If the power-on flag or self-diagnosis (BIST) control registers are to be used, boot the internal power system first (when the internal power supply voltage reaches 1.1 V, set the voltage of the external power system to a voltage below the detection voltage [min: 2.7 V] for POF).

Note Internal power system: V_{DDM}, V_{DDC}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$
External power system: EV_{DD}, $OSCV_{DD}$, FV_{DD}


tRLI, tRIL
tRLA, tRAL
tRIA, tRAI
tFLI, tFIL
tFLA, tFAL
tFIA, tFAI
VDDM,VDDC
EVDD
AVDD0
1.1 V
4.5 V
tRLR
tRIR
tRAR
tFRL
tFRI
tFRA
RESET
VIL
tRMR0
tFRM0
FLMD1,
FLMD0
VIH

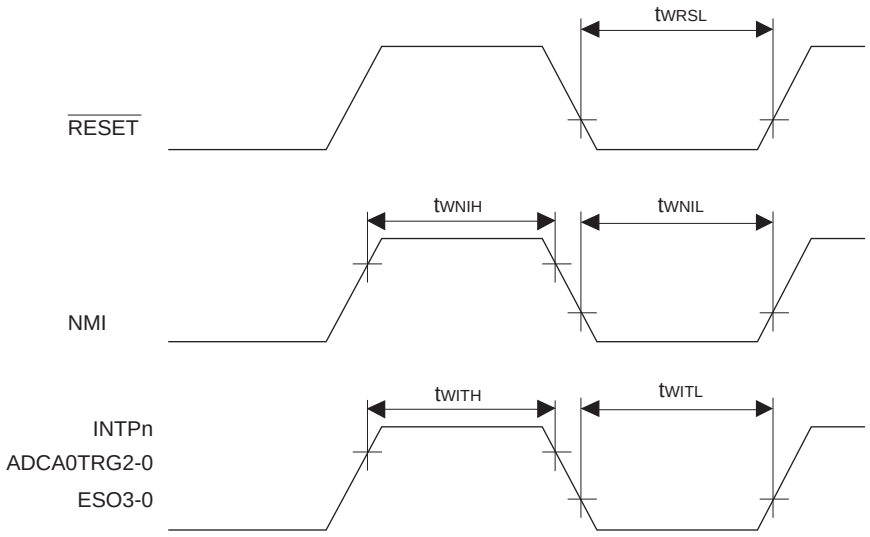
28.3.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}	Other than power supply turning on	500		ns
NMI input high level width	t_{WNIH}	Digital filter	*1		ns
NMI input low level width	t_{WNIL}	Digital filter	*1		ns
INTPn input high level width	t_{WITH}	Digital filter, n = 0 to 10	*1		ns
INTPn input low level width	t_{WITL}	Digital filter, n = 0 to 10	*1		ns
ADCA0TRG2-0 input high level width	t_{WITH}	Analog filter	250		ns
ADCA0TRG2-0 input low level width	t_{WITL}	Analog filter	250		ns
ESO3-0 input high level width	t_{WITH}	Analog filter	250		ns
ESO3-0 input low level width	t_{WITL}	Analog filter	250		ns

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.



Note n = 0 to 10

28.3.6.5 Timer Timing

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
TI input high level width	t_{TIH}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3, TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TI input low level width	t_{TIL}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3 TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TO output cycle time	t_{TCYK}	TAUA0O0-15, TAUA1O0-15 TAUJ0O0-3, TAUJ1O0-3 OSTM0O, OSTM1O TSG20O0-7, TSG21O0-7, TPB0O, TPB1O	0.25	1000000	µs

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.

28.3.6.6 CSIHn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYM}	Output	125		ns
CSIHnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	$(t_{KCYM}/2) - 20$		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIM}		30		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIM}		10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOM}			10	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOM}	CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 0/0 or 1/1	t_{KWHM}		ns
		CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 1/0 or 0/1	t_{KWLM}		ns
CSIHnRYI setup time (to CSIHnSC)	t_{SRYI}		$(2 \times t_{PCLK}) + 30$		ns
CSIHnCSx setup time	t_{SSCSB0}	CSIHnCFGx.CSIHnDAPx = 0	$CS_{SETUP} \times t_{KCYM} - 10$		ns
	t_{SSCSB1}	CSIHnCFGx.CSIHnDAPx = 1	$(CS_{SETUP} + 0.5) \times t_{KCYM} - 10$		ns
CSIHnCSx hold time	t_{HSCSB0}	CSIHnCTL1.CSIHnSIT = 0	$CS_{HOLD} \times t_{KCYM} - 10$		ns
	t_{HSCSB1}	CSIHnCTL1.CSIHnSIT = 1	$(CS_{HOLD} + 0.5) \times t_{KCYM} - 10$		ns
CSIHnCSx inactive level width	t_{WSCSB}	*1	$(CS_{IDLE} + 0.5) \times t_{KCYM} - 20$		ns
		Other than the above	$CS_{IDLE} \times t_{KCYM} - 20$		ns

Note 1. When the setting of CSIHnCFGx.CSIHnCKPx is changed while CS_{IDLE} = 000_B.

Note n = 0,1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

CS_{SETUP}: CSIHnCFGx.CSIHnSPx[3:0] bits

CS_{HOLD}: CSIHnCFGx.CSIHnHDx[3:0] bits

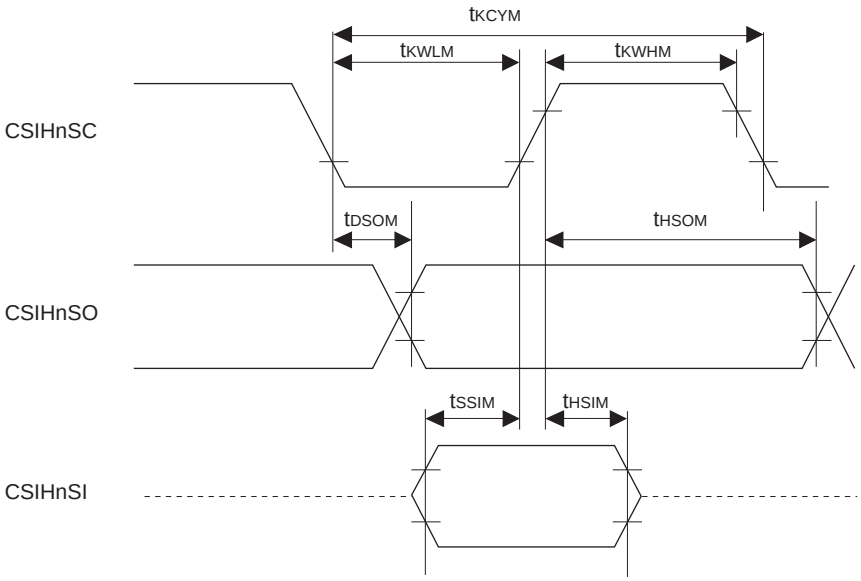
CS_{IDLE}: CSIHnCFGx.CSIHnIDx[2:0] bits

t_{PCLK}: PCLK cycle time

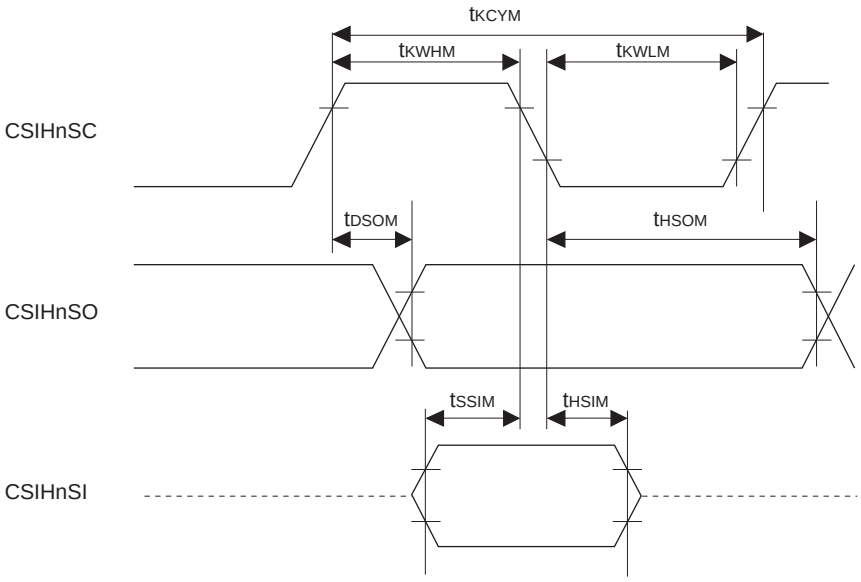
[CSIHnSC/CSIHnSI/CSIHnSO]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0 or 1/1)



Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 1/0 or 0/1)

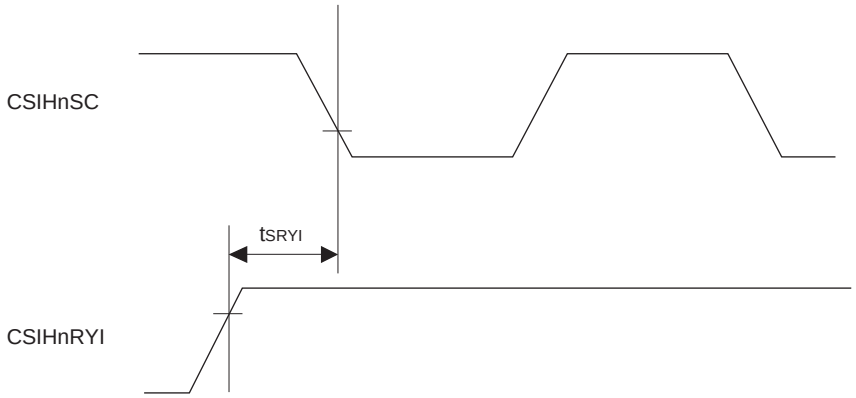


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

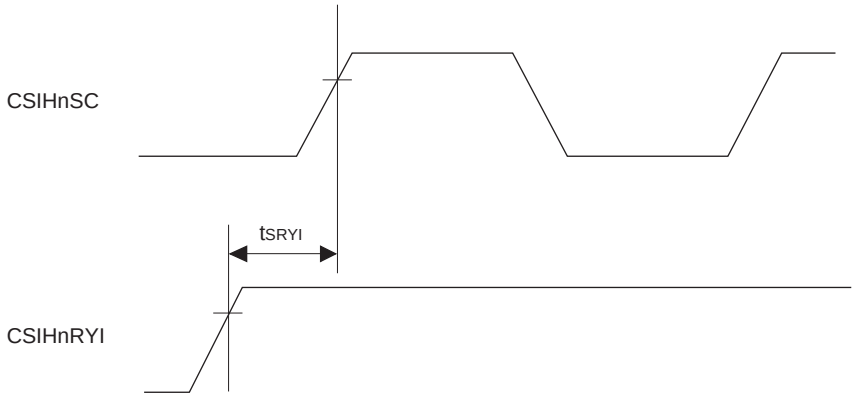
[CSIHnSC/CSIHnRYI]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx = 0)



Master mode: (CSIHnCFGx: CSIHnCKPx = 1)

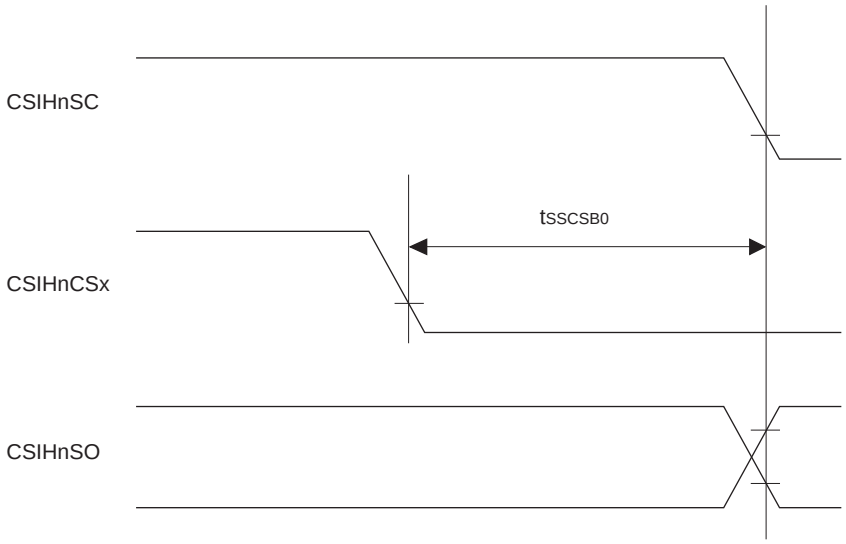


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

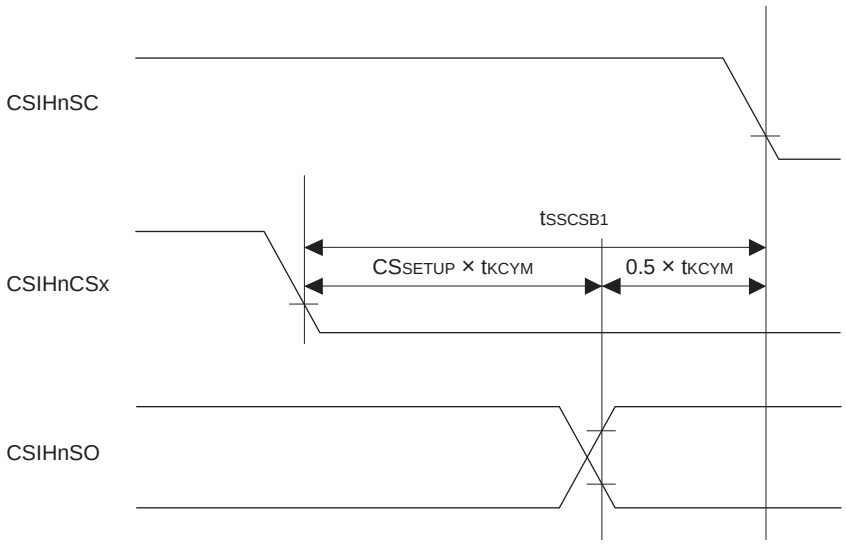
[CSIHnSC/CSIHnSO/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/1)

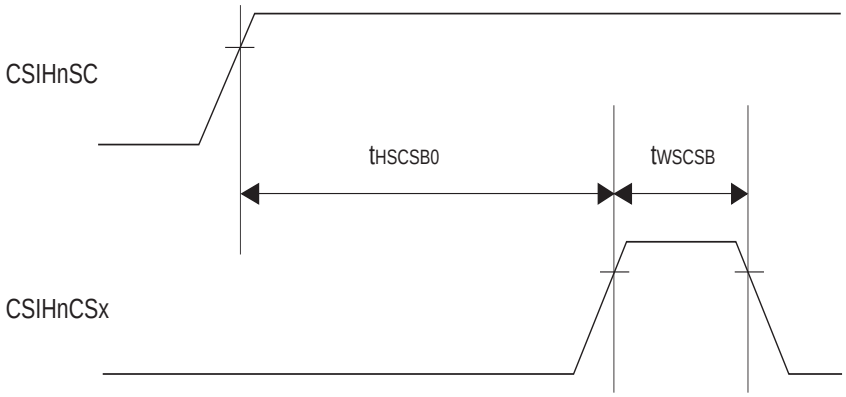


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

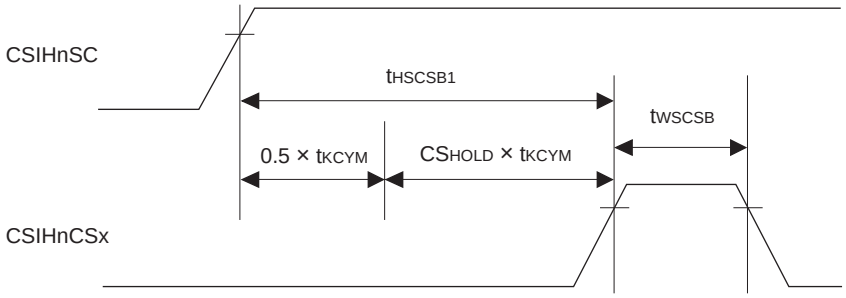
[CSIHnSC/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

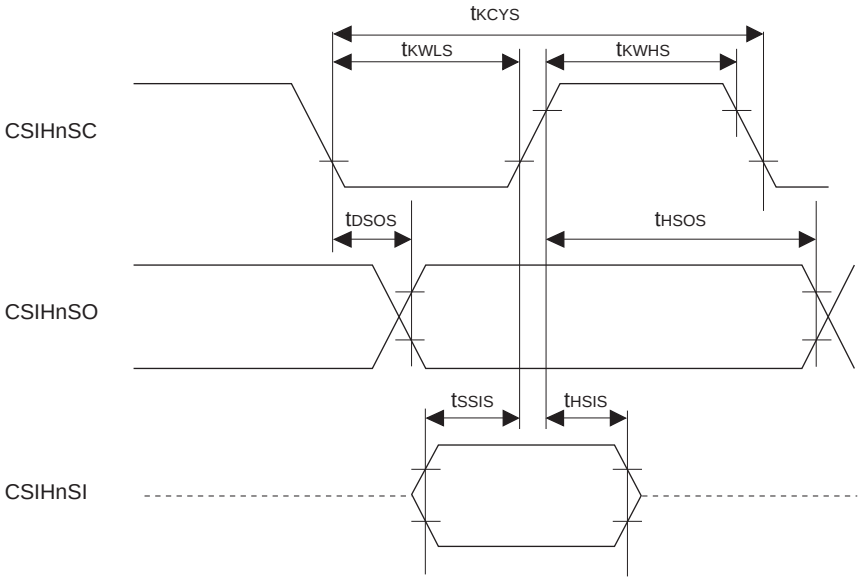
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIHnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIS}			15		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOS}				30	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOS}	CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 0/0 or 1/1		t_{KWHS}		ns
		CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 1/0 or 0/1		t_{KWLS}		ns
CSIHnRYO output delay	t_{SRYO}				30	ns

Note n = 0, 1

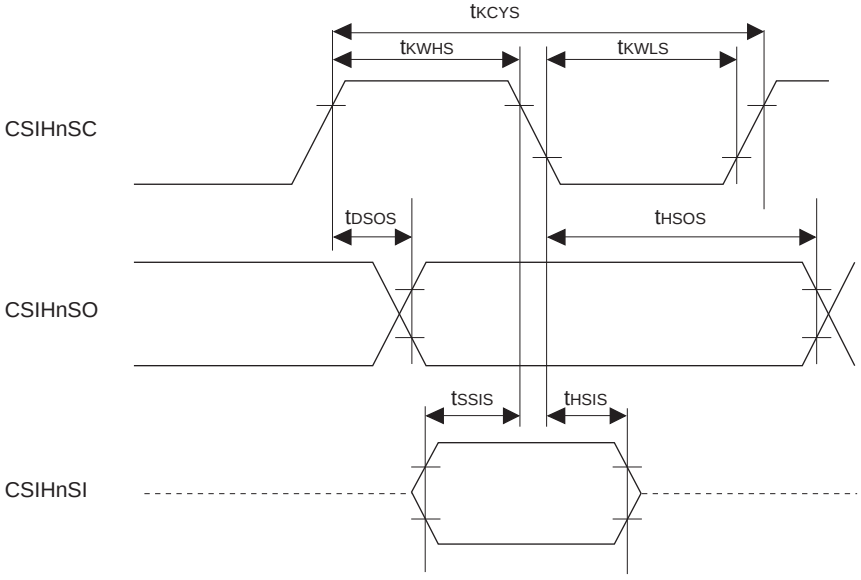
t_{PCLK}: PCLK cycle time

[CSIHnSC/CSIHnSI/CSIHnSO]

Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



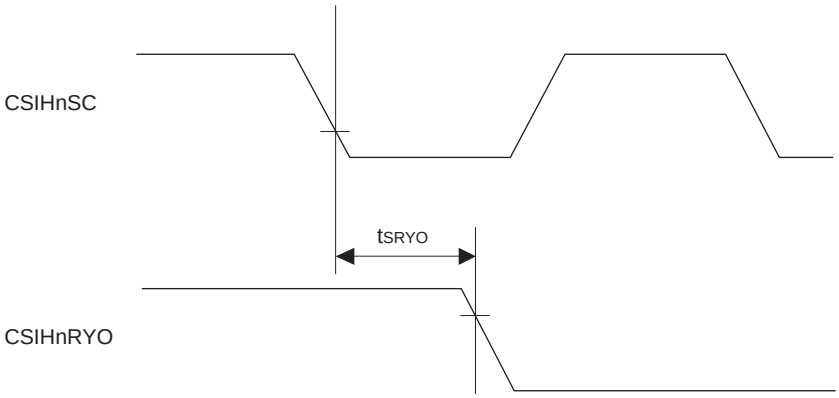
Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 1/0 or 0/1)



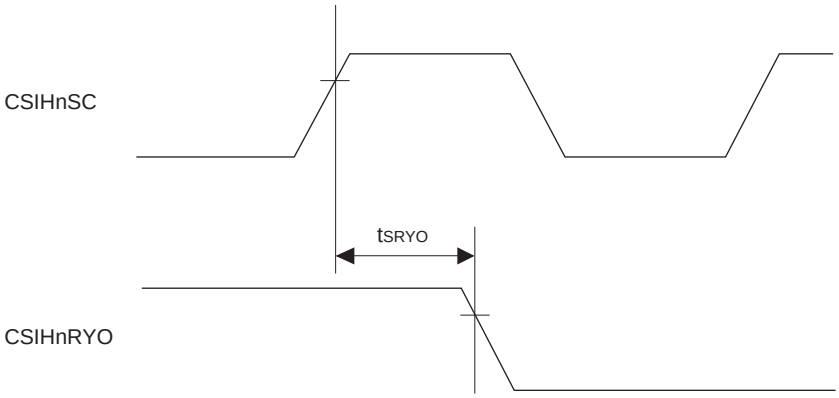
Note n = 0, 1

[CSIHnSC/CSIHnRYO]

(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/1 or 1/0)



Note n = 0, 1

28.3.6.7 CSIGn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

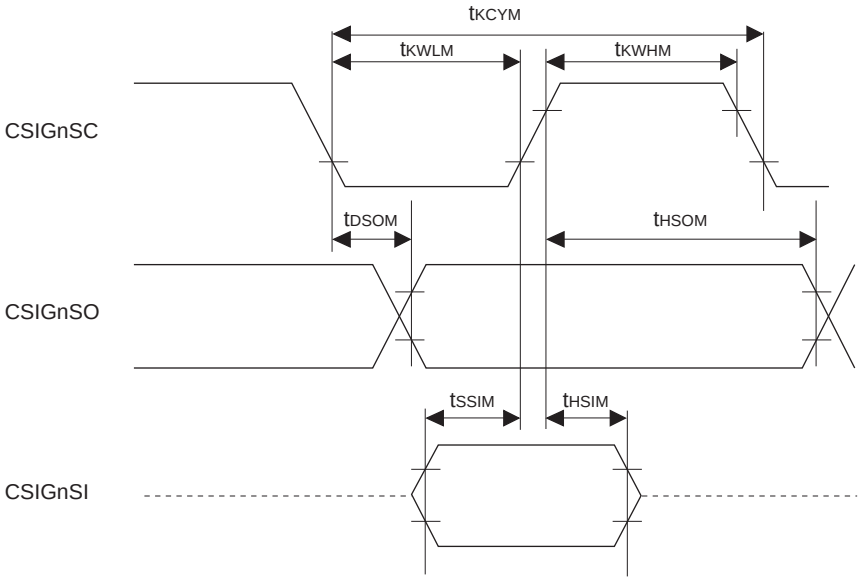
Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYM}	Output	125		ns
CSIGnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIM}		30		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIM}		10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOM}			10	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOM}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1	t_{KWHM}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1	t_{KWLM}		ns
CSIGnRYI setup time (to CSIGnSC)	t_{SRYI}		(2 × t_{PCLK}) + 30		ns

Note n = 0 to 2

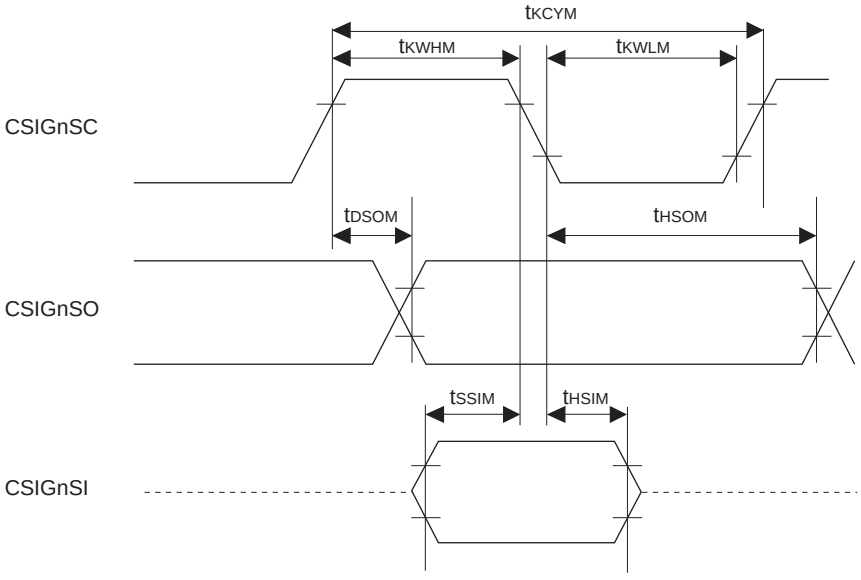
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



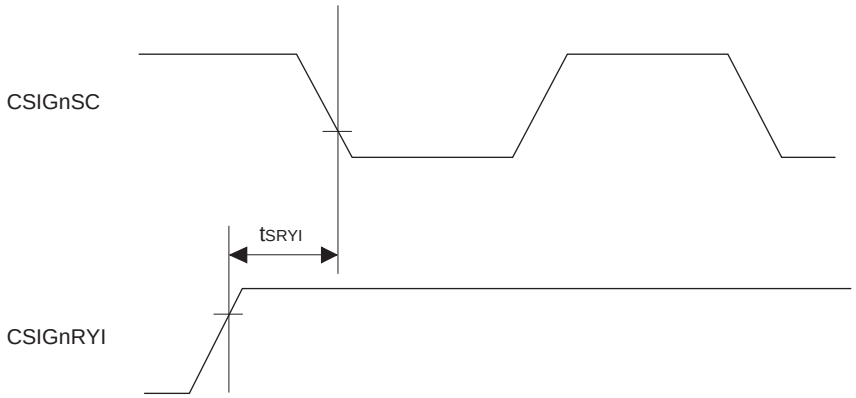
Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



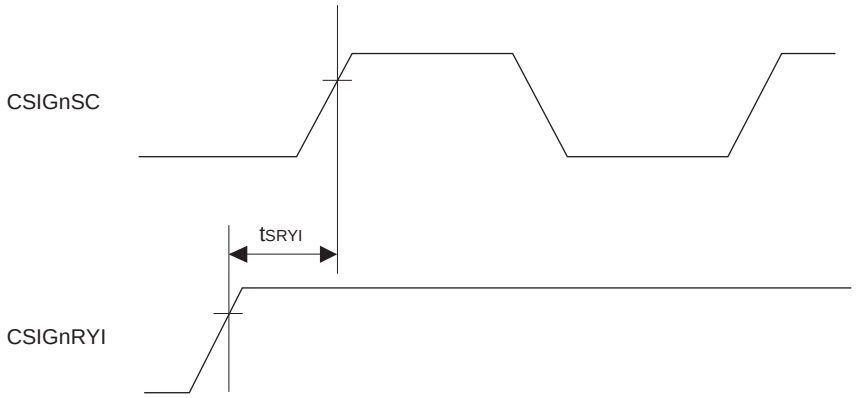
Note n = 0 to 2

[CSIGnSC/CSIGnRYI]

Master mode: (CSIGnCTL1: CSIGnCKR = 0)



Master mode: (CSIGnCTL1: CSIGnCKR = 1)



Note n = 0 to 2

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

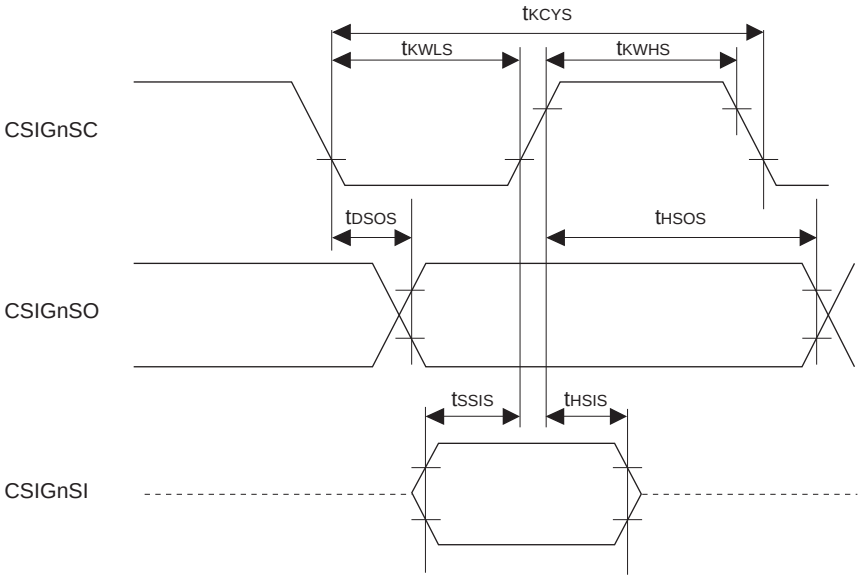
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYS}	Input	PCLK = 64 MHz	150		ns
		Input	PCLK = 80 MHz	150		ns
CSIGnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIS}			15		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOS}				30	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOS}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1		t_{KWHS}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1		t_{KWLS}		ns
CSIGnRYO output delay	t_{SRYO}				30	ns

Note n = 0 to 2

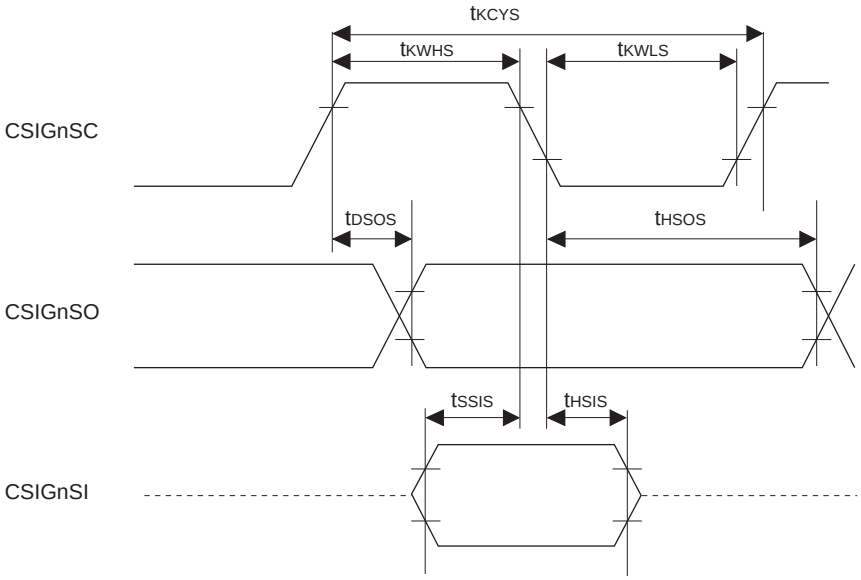
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



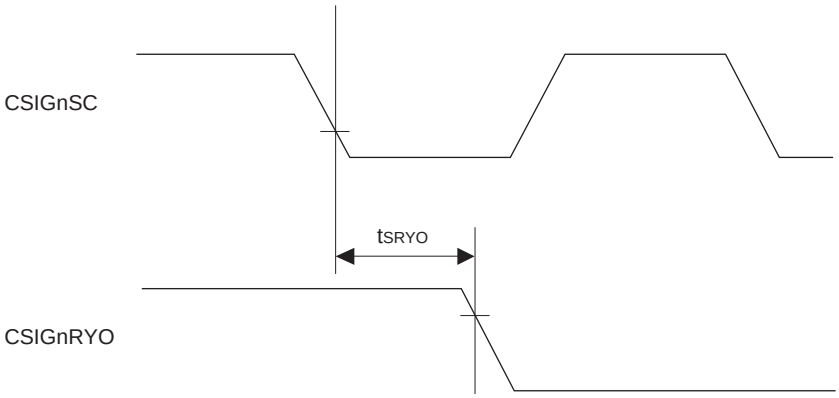
Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



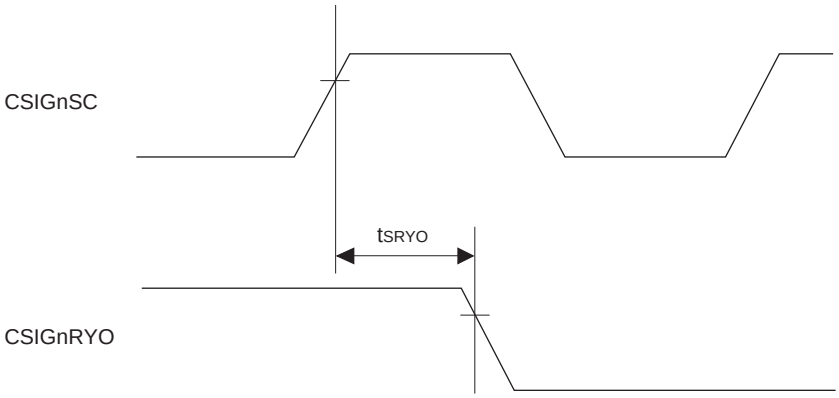
Note n = 0 to 2

[CSIGnSC/CSIGnRYO]

(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1 or 1/0)



Note n = 0 to 2

28.3.6.8 UARTHn Timing

(a) Asynchronous Mode

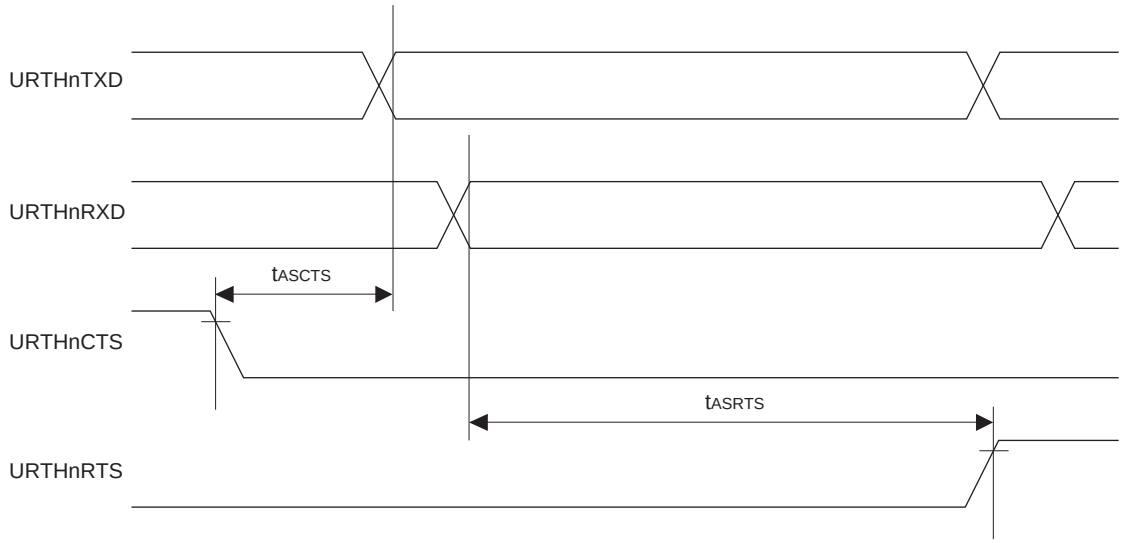
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		MIN.	MAX.	Unit
UARTHn transfer rate				16	2 M	bps
URTHnCTS setup time (to URTHnTXD)	t_{ASCTS}	Input	URTHnCTS without noise filter	3T + 30		ns
			URTHnCTS with noise filter	5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{ASRTS}	Output	URTHnRXD without noise filter		(3BRS + 2) T + 30	ns
			URTHnRXD with noise filter		(3BRS + 4) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]



(b) Clock Synchronous, Master Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM1}	Output		500		500		ns
URTHnSC high level width	t_{KWHM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnSC low level width	t_{KWLM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDM1}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDM1}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM1}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM1}	Output		t_{KCYM1}/2 - 10		t_{KCYM1}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM1}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM1}	Output			T + 30		T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

(c) Clock Synchronous, Slave Mode (Normal Mode)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35pF)

Item	Symbol	Conditions		URTHnSC/URTHnRXD without Noise Filter		URTHnSC/URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		10		10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		2T + 10		2T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		-T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	2T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		2T+ 30	3T + 30	2T + 30	3T + 30	ns

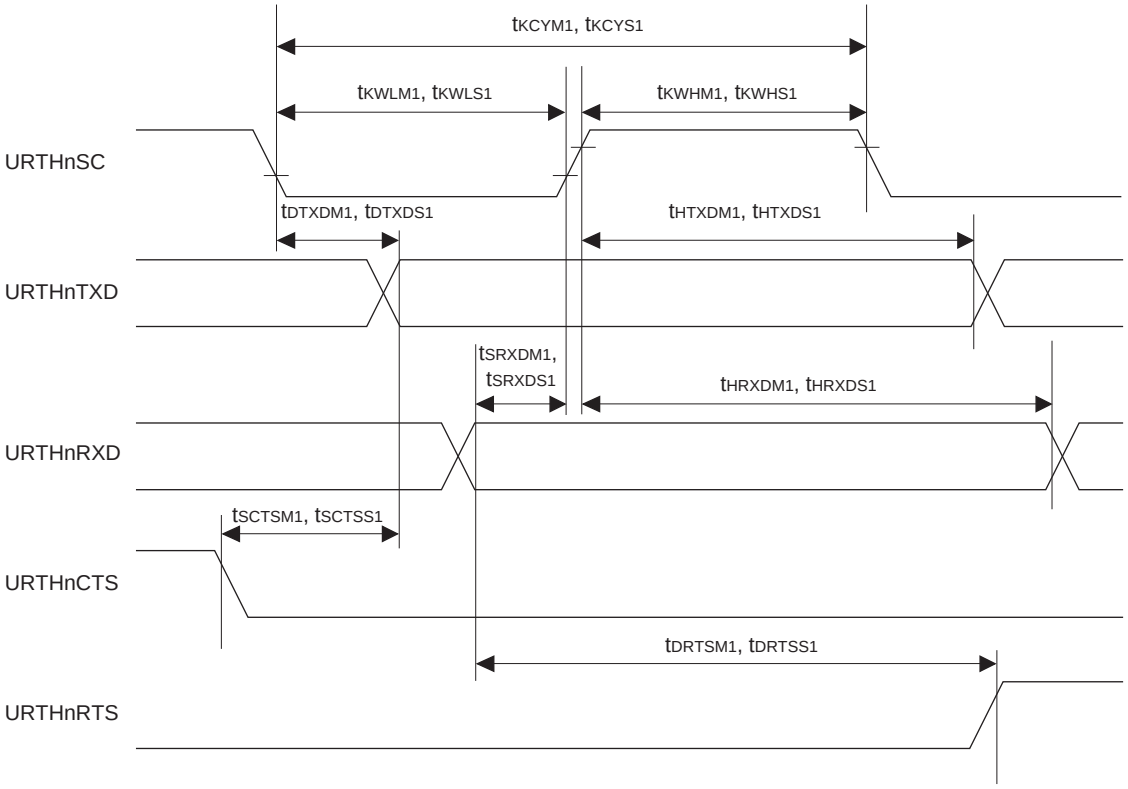
Note n = 0 to 2

T = prescaler cycle time

Item	Symbol	Conditions		URTHnSC without Noise Filter URTHnRXD with Noise Filter		URTHnSC with Noise Filter URTHnRXD without Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		2T + 10		-2T + 10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		10		4T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	3T + t_{KCYS1} + 30		3T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		4T + 30	5T + 30	30	T + 30	ns

Note n = 0 to 2

T = prescaler cycle time


tKCYM1, tKCYS1
tKWLM1, tKWLS1
tKWHM1, tKWHS1
URTHnSC
tDTXDM1, tDTXDS1
tHTXDM1, tHTXDS1
URTHnTXD
tSRXDM1,
tSRXDS1
tHRXDM1, tHRXDS1
URTHnRXD
tSCTSM1, tSCTSS1
URTHnCTS
tDRTSM1, tDRTSS1
URTHnRTS

Note n = 0 to 2

(d) Clock Synchronous, Master Mode (Delay Sampling Mode)

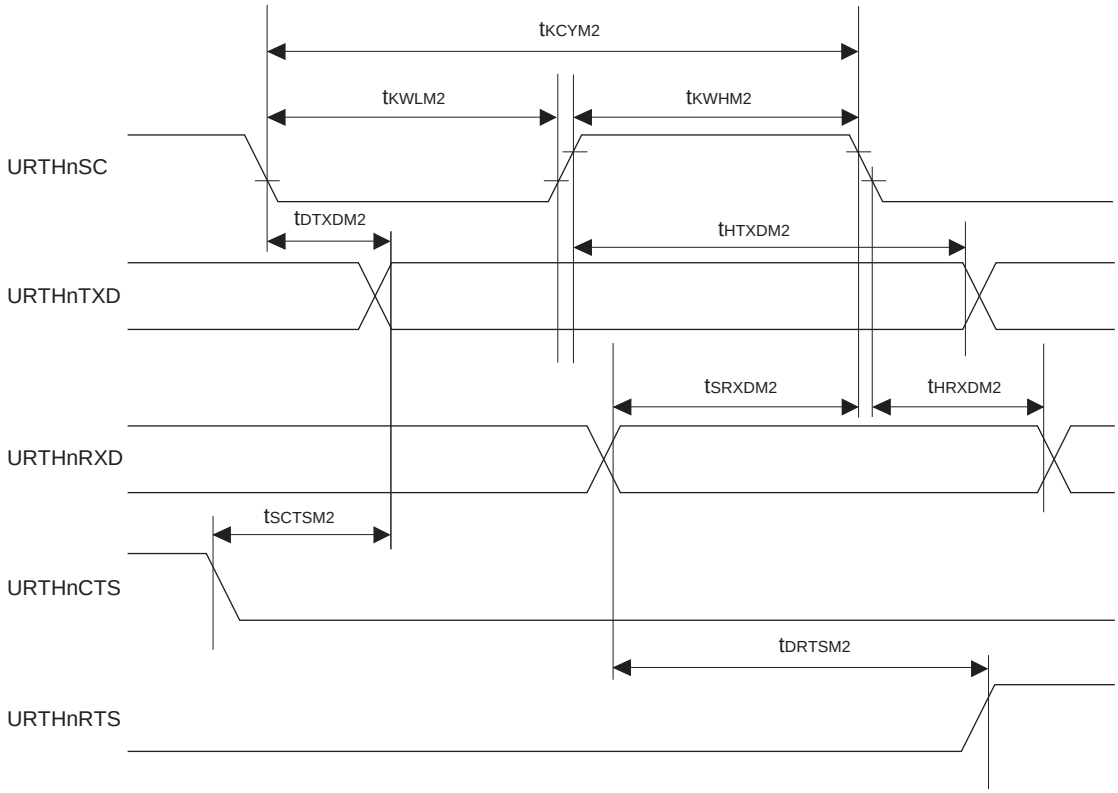
(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM2}	Output		500		500		ns
URTHnSC high level width	t_{KWHM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnSC low level width	t_{KWLM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↓)	t_{SRXDM2}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↓)	t_{HRXDM2}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM2}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM2}	Output		t_{KCYM2}/2 - 10		t_{KCYM2}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM2}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM2}	Output			(BRS + 1) T + 30		(BRS + 1) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]


tKCYM2
tKWLM2
tKWHM2
URTHnSC
tDTXDM2
tHTXDM2
URTHnTXD
tSRXDM2
tHRXDM2
URTHnRXD
tSCTSM2
URTHnCTS
tDRTSM2
URTHnRTS

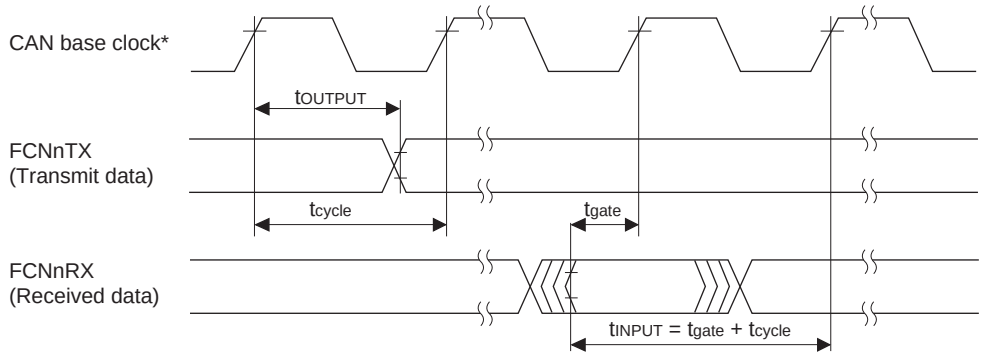
Note n = 0 to 2

28.3.6.9 CAN Timing

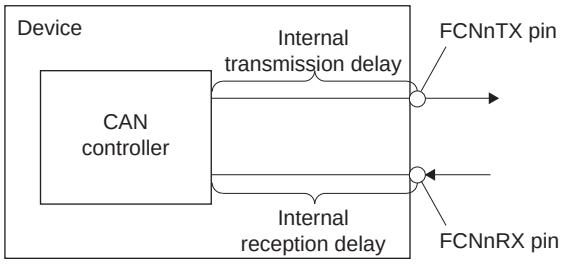
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CAN transfer rate				1	Mbps
CAN internal delay time*[1]	t_{node}			87.5	ns

Note 1. CAN internal delay time (t_{node}) = t_{OUTPUT} (internal transmission delay) + t_{INPUT} (internal reception delay)



Note * CAN base clock frequency: f_{CAN}



Schematic view of the CAN-internal delay

Note n = 0, 1

28.3.6.10 FlexRay Timing

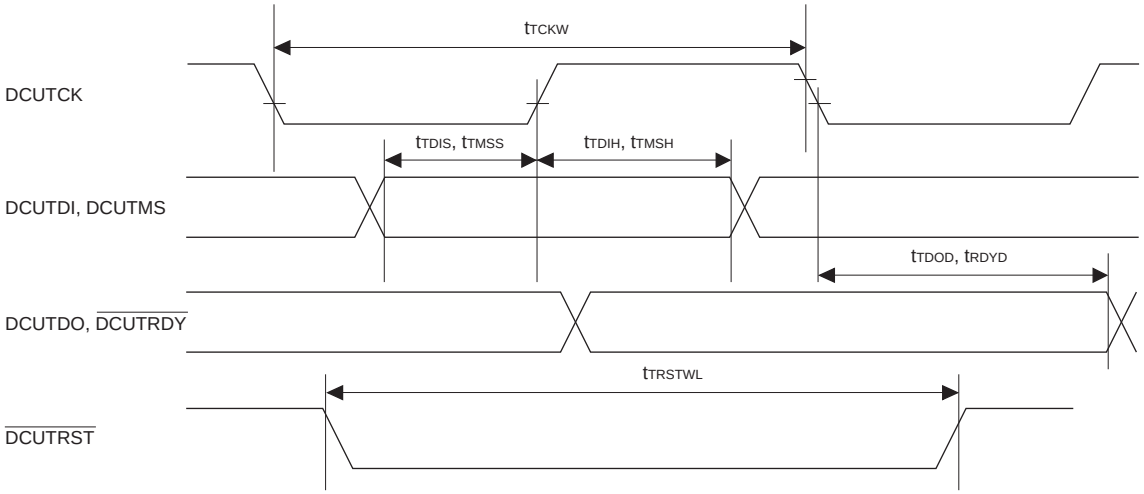
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				10	Mbps

28.3.6.11 NEXUS I/F Timing

(T_a = 0 to +40°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.
DCUTCK cycle time	t_{TCKW}	40.0		ns
DCUTDI setup time (to DCUTCK ↑)	t_{TDIS}	t_{TCKW}/2 - 8.5		ns
DCUTMS setup time (to DCUTCK ↑)	t_{TMSS}	t_{TCKW}/2 - 8.5		ns
DCUTDI hold time (to DCUTCK ↑)	t_{TDIH}	2.0		ns
DCUTMS hold time (to DCUTCK ↑)	t_{TMSH}	2.0		ns
DCUTDO output delay time (to DCUTCK ↓)	t_{TDOD}		20.0	ns
$\overline{\text{DCUTRDY}}$ output delay time (to DCUTCK ↓)	t_{RDYD}		20.0	ns
$\overline{\text{DCUTRST}}$ low level width	t_{TRSTWL}	1000		ns



28.3.6.12 A/D Converter Characteristics

(a) 10-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	μs
Accuracy*1	TOE					±2.0	LSB
Conversion time*2	t_{CONV}			1.5		10	ms
Zero-scale error*1	ZSE					±1.5	LSB
Full-scale error*1	FSE					±1.5	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	μA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} conversion	1003		1023	LSB
			2/3AV_{DD0} conversion	673	683	693	LSB
			1/2AV_{DD0} conversion	504	512	520	LSB
			1/3AV_{DD0} conversion	331	341	351	LSB
			AV_{SS0} conversion	0		20	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(b) 10-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.5	LSB
Conversion time*2	t_{CONV}			1.5		10	ms
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±2.0	LSB
Full-scale error*1	FSE					±2.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±2.0	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} conversion	671	683	695	LSB
			1/2AV_{DD0} conversion	502	512	522	LSB
			1/3AV_{DD0} conversion	329	341	353	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(c) 12-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	µs
Accuracy*1	TOE					±6.0	LSB
Conversion time*2	t_{CONV}			1.0		10	ms
Zero-scale error*1	ZSE					±5.0	LSB
Full-scale error*1	FSE					±5.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} conversion	4015		4095	LSB
			$2/3AV_{DD0}$ conversion	2691	2731	2771	LSB
			$1/2AV_{DD0}$ conversion	2018	2048	2078	LSB
			$1/3AV_{DD0}$ conversion	1325	1365	1405	LSB
			AV_{SS0} conversion	0		80	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(d) 12-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	µs
Accuracy*1	TOE					±9.0	LSB
Conversion time*2	t_{CONV}			1.5		10	ms
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±7.0	LSB
Full-scale error*1	FSE					±7.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±3.5	LSB
Integral non-linearity error*1	INL					±5.0	LSB
AVREF0P input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS=0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} conversion	2689	2731	2773	LSB
			1/2AV_{DD0} conversion	2016	2048	2080	LSB
			1/3AV_{DD0} conversion	1323	1365	1407	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

Caution 1. Implement the following countermeasures to minimize any adverse effect on precision of the A/D converter from the switching noise of the digital circuit.

i. Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

ii. Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, FV_{DD}, or AV_{DD0}) and ground pin (V_{SSM}, V_{SSC}, EV_{SS}, $OSCV_{SS}$, $CPUPLLV_{SS}$, $FRPLLV_{SS}$, FV_{SS}, or AV_{SS0}), and between the analog reference voltage pins (AV_{REF0P} or AV_{REF0M}) and the analog ground pin (AV_{SS0}).

iii. We recommend isolating the analog power supply (AV_{DD0}) from the digital power supply (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, or FV_{DD}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board. Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

Caution 2. Connect resistor Re and capacitor Ce to the analog input pins (ADCA0Im) externally to eliminate noise (m = 01 to 22).

Note Full-scale range (%FSR) is the ratio, as a percentage, of the extremes of the range of analog input voltages that can be converted, and is expressed as follows regardless of the resolution.

1 %FSR = (Maximum value of convertible analog input voltage - Minimum value of convertible analog input voltage)/100

= (AV_{REF0P} - 0)/100

= AV_{REF0P}/100

1 LSB is as follows at a resolution of 10 bits:

1 LSB = $1/2^{10}$

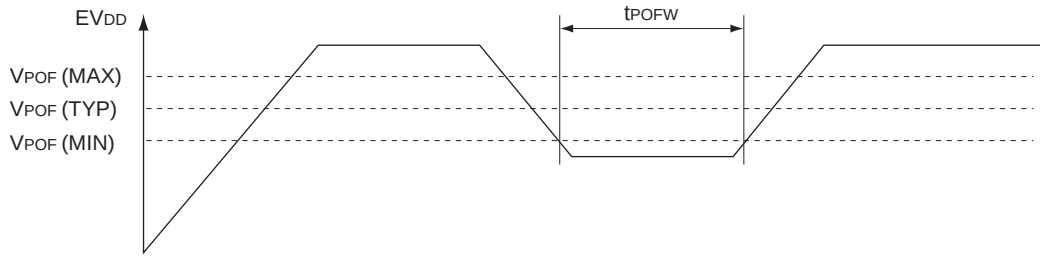
= 0.098%FSR

28.3.6.13 POF/LVI Characteristics

POF characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{POF}		2.7	2.8	2.9	V
Minimum detection width	t_{POFW}		0.2			ms

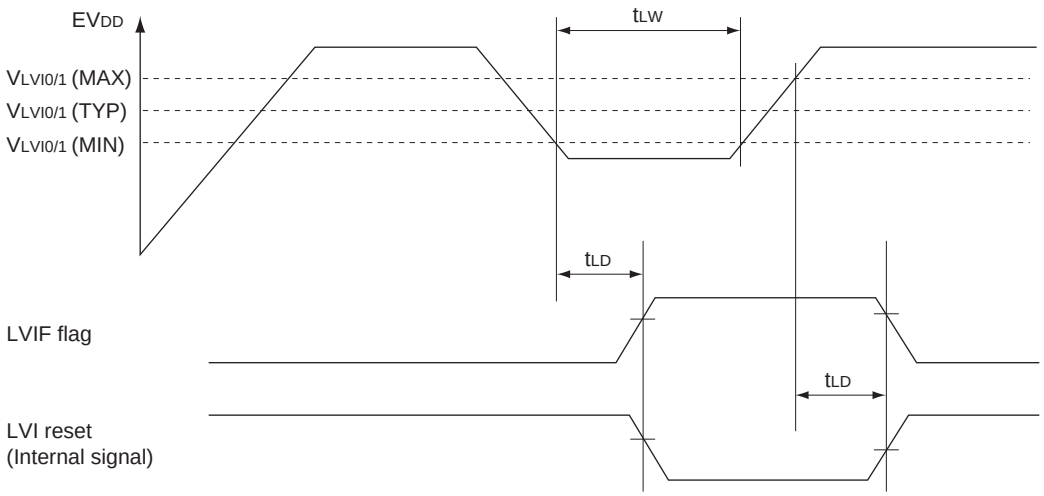


LVI characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{LVI0}	LVICNT.LVICNT[1:0]=01_B	4.5	4.6	4.7	V
	V_{LVI1}	LVICNT.LVICNT[1:0]=10_B	4.2	4.3	4.4	V
Response time*1	t_{LD}	After EV_{DD} reaches V_{LVI0}, V_{LVI1} (MAX.) After EV_{DD} falls V_{LVI0}, V_{LVI1} (MIN.) Voltage gradient: 1.8 V/us to 1.8 V/s			2	ms
Minimum detection width	t_{LW}		2.0			ms
Stabilization waiting time*2					350	µs

Note 1. Time until output of an interrupt or reset after voltage detection.
Note 2. Time until LVI is detected in accord with the specification after enabling of the LVI function.


EVDD
tLW
VLVI0/1 (MAX)
VLVI0/1 (TYP)
VLVI0/1 (MIN)
tLD
LVIF flag
tLD
LVI reset
(Internal signal)

28.3.6.14 Flash Memory Programming Characteristics

(a) Basic Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}			64		80	MHz
Number of times rewritable (per block)*1	C_{ERWR}	Code Flash	Data are retained for 20 years			100	Times/ block
		Data Flash	Data are retained for 5 years			15000	Times/ block
			Data are retained for 15 years			5000	Times/ block
			Data are retained for 20 years			1000	Times/ block

Note 1. In writing to a delivered product, either "erasing → writing" or "writing alone" is considered as a "time" of rewriting.

Example: (P: writing, E: erasing)

Delivered product→ P→E→P→E→P : 3 times

Delivered product→E→P→E→P→E→P : 3 times

28.3.6.15 FLMD0 Pulse Timing Characteristics

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (to VDD)	t_{DP}		1			ms
$\overline{\text{RESET}}$ release (to FLMD0)	t_{PR}		2			ms
$\overline{\text{RESET}}\uparrow \rightarrow$ FLMD0 set time	t_{RP}		1200			µs
FLMD0 high/low level width	t_{PW}		10		100	µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns
Serial clock input frequency	t_{SCCY}		9.8		5000	KHz
High level width	t_{WHSC}		85		50000	ns
Low level width	t_{WLSC}		85		50000	ns
SI setup time (to SCK ↑)	t_{PLHSC}		75			ns
SI hold time (to SCK ↑)	t_{PHLSC}		0			ns
SO output delay time	t_{DSC}				23	ns
UART transfer rate			9.6		1000	Kbps

Note VDD: $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$

28.3.6.16 BIST Execution Time

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition 1	Condition 2	MIN.	TYP.	MAX.	Unit
BIST execution time		All HW-BIST Total fxx = 64 MHz				26	ms
		All HW-BIST Total fxx = 80 MHz				21	ms

28.4 Electrical Characteristics of the µPD70F3508 (Target)

28.4.1 Absolute Maximum Ratings

(T_a = 25°C)

Item	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DDM}	$V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +1.6	V
	V_{DDC}		-0.5 to +1.6	V
	$CPUPLLV_{DD}$		-0.5 to +1.6	V
	$FRPLLV_{DD}$		-0.5 to +1.6	V
	EV_{DD}	$EV_{DD} = OSCV_{DD} = FV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +4.1	V
	$OSCV_{DD}$		-0.5 to +4.1	V
	FV_{DD}		-0.5 to +4.1	V
	AV_{DD0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +4.1	V
	V_{SSM}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	V_{SSC}		-0.5 to +0.5	V
	$CPUPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	$FRPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	EV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	$OSCV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	FV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	AV_{SS0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
Input voltage	V_I		-0.5 to EV_{DD} + 0.5*[1]	V
Analog input voltage	V_{IAN}	ADCA0I1 to ADCA0I22	-0.5 to AV_{DD0} + 0.5*[1]	V
	AV_{REF0P}		-0.5 to AV_{DD0} + 0.5*[1]	V
	AV_{REF0M}		-0.5 to AV_{DD0} + 0.5*[1]	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total pins	-50	mA
Low-level output current	I_{OL}	1 pin	10	mA
		Total pins	50.	mA
Operating ambient temperature	T_a		-40 to +125	°C
Storage temperature	T_{stg}	Before writing to flash memory	-65 to +150	°C
		After writing to flash memory	-65 to +125	°C

Note 1. Be sure not to exceed the absolute maximum ratings of each supply voltage.

Caution 1. Make sure that the total the current flowing through these pins and output currents does not exceed 50 mA.

Caution 2. Do not directly connect output (or I/O) pins of IC products to each other, or to VDD, VCC, and GND. Open drain pins or open collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

Caution 3. Product quality may suffer if the absolute maximum rating is exceeded even momentary for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

28.4.2 Capacity

(T_a = 25°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz 0 V for non-measured pins, and excluding analog pins selected for use with S/H channels or AD converter channels.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

28.4.3 Operating Conditions

(T_a = -40 to +125°C,V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Internal Operation Clock Frequency	Operating Ambient Temperature (T_a)	Power Supply Voltage
fxx = 80 MHz	T_a = -40 to +125°C	V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V
fxx = 160 MHz		

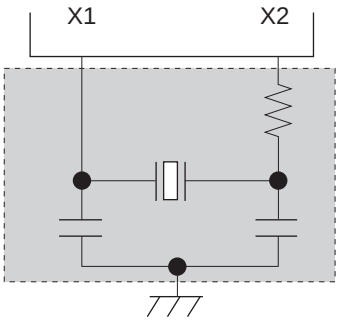
28.4.4 Characteristics of Oscillation Circuit

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency*1	fx		16	16	16	MHz

Note 1. The values are only applicable as oscillation circuit characteristics.

Caution 1. When placing wiring in the area enclosed by the broken line in the figure below during the design of the oscillator, avoid adverse effects from wiring capacitance by following the directions below.



- Bring the oscillation circuit as close as possible to the X1 and X2 terminals.
- Do not pass other signal conductors through the broken line part in the above figure.
- Do not route the wiring near signal lines through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as $OSCV_{SS}$.
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When selecting the crystal oscillator, designing the parameters of the oscillation circuit, or calculating the time required for oscillation to become stable, exhaustively evaluate the finished system (MCU, oscillator, and target board) for matching of the device and the oscillator or ask the manufacturer of the oscillator to do so.

28.4.5 DC Characteristics

DC Characteristics (1/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-level input voltage	V_{IH1}	Port pins, FLMD0 pin	0.7 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH2}	FLMD1 pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH3}*5	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.0		EV_{DD} + 0.3	V
	V_{IH4}	$\overline{\text{RESET}}$ pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
Low-level input voltage	V_{IL1}	Port pins, FLMD0 pin	-0.5		0.3 × EV_{DD}	V
	V_{IL2}	FLMD1 pin	-0.5		0.2 × EV_{DD}	V
	V_{IL3}*5	DCUTDI, DCUTMS DCUTCK	-0.5		0.8	V
	V_{IL4}	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
Hysteresis width*1	V_{HYS1}	Port pins, FLMD0 pin		0.5		V
	V_{HYS2}	FLMD1 pin		1.0		V
	V_{HYS3}	$\overline{\text{RESET}}$ pin		1.0		V
High-level output voltage*2	V_{OH1}	I_{OH} = - 3mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 1mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 0.1mA	EV_{DD} - 0.5		EV_{DD}	V
Low-level output voltage*2	V_{OL1}	I_{OL} = 3mA	0		0.4	V
		I_{OL} = 1mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V, other than for the pins below	20	40	100	KΩ
	R_2	V_I = 0 V *6	10	19	50	KΩ
Pull-down resistor	R_3	V_I = EV_{DD}*3	20	40	100	KΩ
	R_4	V_I = EV_{DD}*4	25	40	100	KΩ
	R_5	V_I = EV_{DD}*6	10	19	50	KΩ

Note 1. When the operation of an adjacent pin is halted.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.
Note 3. Only for the FLMD1 and $\overline{\text{DCUTRST}}$ pins
Note 4. Only for the $\overline{\text{RESET}}$ pin
Note 5. In debugging mode
Note 6. Only for the FLMD0 pin

Caution The characteristics of the alternative function pins are the same as those of the port pin unless otherwise specified.

DC Characteristics (2/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-level input leakage current	I_{LIH1}	V_{IAN} = AV_{DD0}	Analog pins*1			0.2	µA
		V_I = EV_{DD}	Pins other than analog pins*2			5	µA
Low-level input leakage current	I_{LIL1}	V_{IAN}= 0 V	Analog pins*1			-0.2	µA
		V_I = 0 V	Pins other than analog pins*2			-5	µA
High-level output leakage current	I_{LOH1}	V_I = EV_{DD}				5	µA
Low-level output leakage current	I_{LOL1}	V_I = 0 V				-5	µA

Note 1. ADCA0I1 to ADCA0I22
Note 2. The current flowing through the pull-up/pull-down resistors is not included.

DC Characteristics (3/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power-supply current*1	Normal operating conditions	I_{DD}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			350	mA
				fxx = 160 MHz			500	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA
	When rewriting flash memory	I_{DDF}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			325	mA
				fxx = 160 MHz			450	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA
	When executing BIST	I_{DDB}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			460	mA
				fxx = 160 MHz			730	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})				40	mA
	When HALT	I_{DDH}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			315	mA
				fxx = 160 MHz			440	mA
		I_{DD3}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA

Note 1. The current flowing through the pull-up/pull-down resistors is not included. The value for current is applicable when PLCK = 1/2 HEAPCLK.

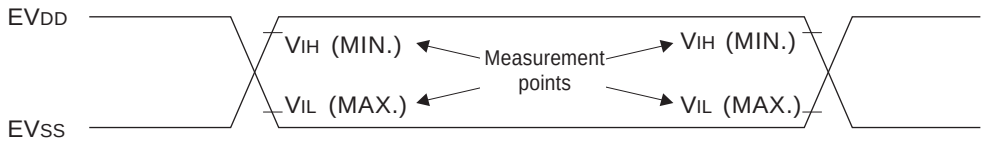
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.

Note MAX. is the maximum value of average current.

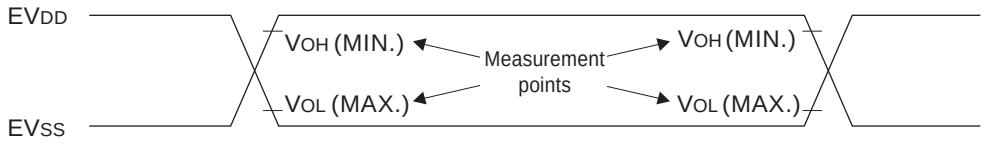
28.4.6 AC Characteristics

28.4.6.1 Measurement Conditions

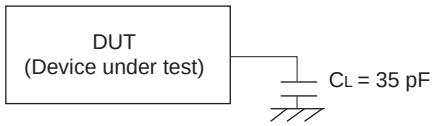
Input waveform for AC testing



AC test output measurement points



Load conditions

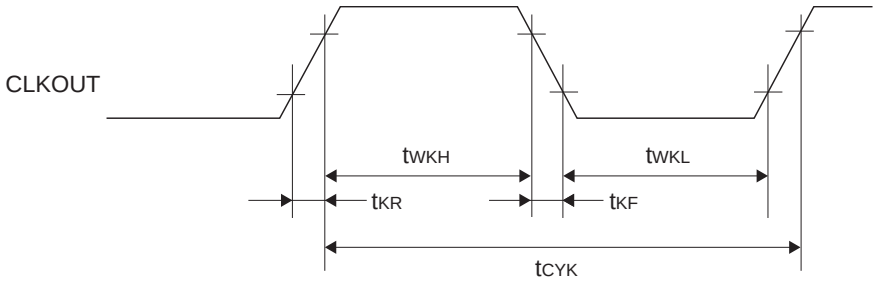


Caution If the load capacitance exceeds 35 pF due to the circuit configuration, bring the load capacitance of the device to 35 pF or less by inserting a buffer or by some other means.

28.4.6.2 Clock Timing (CLKOUT)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CLKOUT output cycle time	t_{CYK}		0.1	4	µs
CLKOUT high level width	t_{WKH}		t_{CYK}/2 - 13		ns
CLKOUT low level width	t_{WKL}		t_{CYK}/2 - 13		ns
CLKOUT rise time	t_{KR}			13	ns
CLKOUT fall time	t_{KF}			13	ns



28.4.6.3 Timing in Turning the Power Supply On and Off

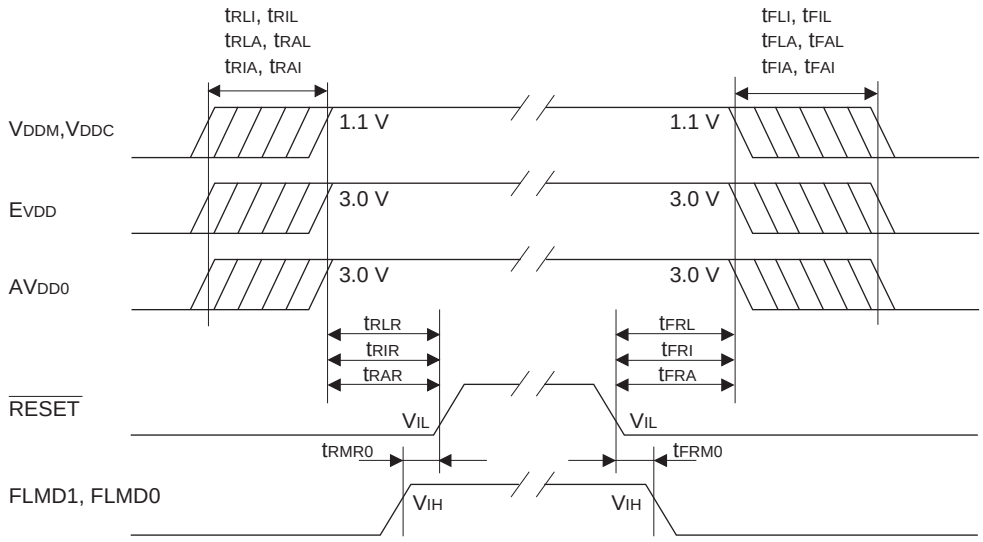
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
V_{DDM}, V_{DDC} ↑ to EV_{DD} ↑	t_{RLI}		0	1000	ms
EV_{DD} ↑ to V_{DDM}, V_{DDC} ↑	t_{RIL}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to AV_{DD0} ↑	t_{RLA}		0	1000	ms
AV_{DD0} ↑ to V_{DDM}, V_{DDC} ↑	t_{RAL}		0	1000	ms
EV_{DD} ↑ to AV_{DD0} ↑	t_{RIA}		0	1000	ms
AV_{DD0} ↑ to EV_{DD} ↑	t_{RAI}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to $\overline{RESET}$ ↑	t_{RLR}	*1	t_{OSC} + 0.5		ms
EV_{DD} ↑ to $\overline{RESET}$ ↑	t_{RIR}	*1	t_{OSC} + 0.5		ms
AV_{DD0} ↑ to $\overline{RESET}$ ↑	t_{RAR}	*1	t_{OSC} + 0.5		ms
FLMD1,0 setup time (to $\overline{RESET}$ ↑)	t_{RMR0}		0		ns
FLMD1,0 hold time(to $\overline{RESET}$ ↓)	t_{FRM0}	*2	0		ns
$\overline{RESET}$ ↓ to V_{DDM}, V_{DDC} ↓	t_{FRL}		500		ns
$\overline{RESET}$ ↓ to EV_{DD} ↓	t_{FRI}		500		ns
$\overline{RESET}$ ↓ to AV_{DD0} ↓	t_{FRA}		500		ns
EV_{DD} ↓ to V_{DDM}, V_{DDC} ↓	t_{FIL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to EV_{DD} ↓	t_{FLI}		0	1000	ms
AV_{DD0} ↓ to V_{DDM}, V_{DDC} ↓	t_{FAL}		0	1000	ms
V_{DDM}, V_{DDC}↓to AV_{DD0} ↓	t_{FLA}		0	1000	ms
AV_{DD0} ↓ to EV_{DD} ↓	t_{FAI}		0	1000	ms
EV_{DD} ↓ to AV_{DD0} ↓	t_{FIA}		0	1000	ms

Note 1. t_{OSC}: Oscillation stabilization time
Note 2. Using the specified value is recommended for the FLMD1, 0 hold time (time until $\overline{RESET}$ falls: t_{FRM0}). If using the recommended value is difficult, change the FLMD1, 0 pins once rewriting of the flash memory is complete.

Caution 1. Turn on/off internal power system at the same time.
Caution 2. Turn on/off external power system at the same time
Caution 3. Apply an external reset ($\overline{RESET}$) whenever the power supply voltage falls below the operating voltage.
Caution 4. If the power-on flag or self-diagnosis (BIST) control registers are to be used, boot the internal power system first (when the internal power supply voltage reaches 1.1 V, set the voltage of the external power system to a voltage below the detection voltage [min: 2.7 V] for POF).

Note Internal power system: V_{DDM}, V_{DDC}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$
External power system: EV_{DD}, $OSCV_{DD}$, FV_{DD}


tRLI, tRIL
tRLA, tRAL
tRIA, tRAI
tFLI, tFIL
tFLA, tFAL
tFIA, tFAI
VDDM,VDDC
EVDD
AVDD0
1.1 V
3.0 V
tRLR
tRIR
tRAR
tFRL
tFRI
tFRA
RESET
VIL
tRMR0
tFRM0
FLMD1, FLMD0
VIH

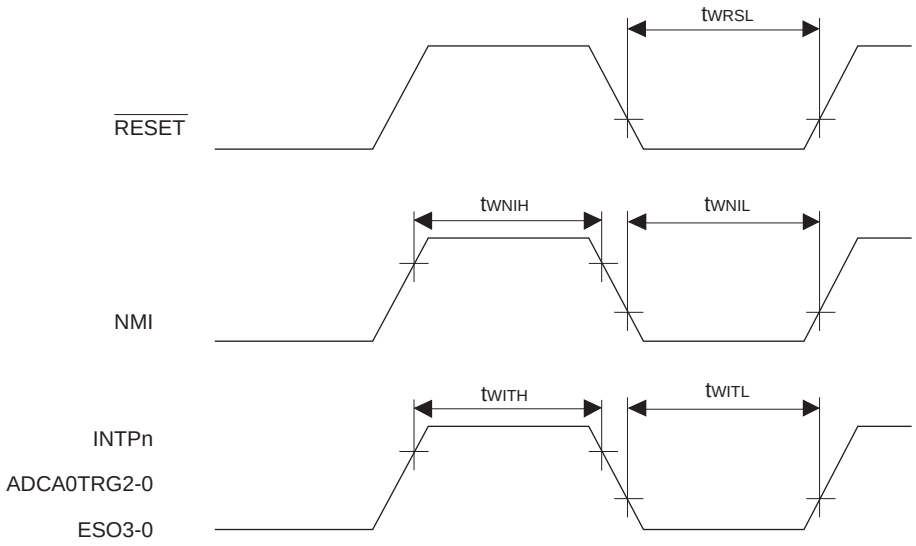
28.4.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}	Other than power supply turning on	500		ns
NMI input high level width	t_{WNIH}	Digital filter	*1		ns
NMI input low level width	t_{WNIL}	Digital filter	*1		ns
INTPn input high level width	t_{WITH}	Digital filter, n = 0 to 10	*1		ns
INTPn input low level width	t_{WITL}	Digital filter, n = 0 to 10	*1		ns
ADCA0TRG2-0 input high level width	t_{WITH}	Analog filter	250		ns
ADCA0TRG2-0 input low level width	t_{WITL}	Analog filter	250		ns
ESO3-0 input high level width	t_{WITH}	Analog filter	250		ns
ESO3-0 input low level width	t_{WITL}	Analog filter	250		ns

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.



Note n = 0 to 10

28.4.6.5 Timer Timing

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
TI input high level width	t_{TIH}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3, TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TI input low level width	t_{TIL}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3 TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TO output cycle time	t_{TCYK}	TAUA0O0-15, TAUA1O0-15 TAUJ0O0-3, TAUJ1O0-3 OSTM0O, OSTM1O TSG20O0-7, TSG21O0-7, TPB0O, TPB1O	0.25	1000000	µs

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.

28.4.6.6 CSIHn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYM}	Output	125		ns
CSIHnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) – 20		ns
CSIHnSII setup time (to CSIHnSC)	t_{SSIM}		30		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIM}		10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOM}			10	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOM}	CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 0/0 or 1/1	t_{KWHM}		ns
		CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 1/0 or 0/1	t_{KWLM}		ns
CSIHnRYI setup time (to CSIHnSC)	t_{SRYI}		(2 × t_{PCLK}) + 30		ns
CSIHnCSx setup time	t_{SSCSB0}	CSIHnCFGx.CSIHnDAPx = 0	CS_{SETUP} × t_{KCYM} – 10		ns
	t_{SSCSB1}	CSIHnCFGx.CSIHnDAPx = 1	(CS_{SETUP} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx hold time	t_{HSCSB0}	CSIHnCTL1.CSIHnSIT = 0	CS_{HOLD} × t_{KCYM} – 10		ns
	t_{HSCSB1}	CSIHnCTL1.CSIHnSIT = 1	(CS_{HOLD} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx inactive level width	t_{WSCSB}	*1	(CS_{IDLE} + 0.5) × t_{KCYM} – 20		ns
		Other than the above	CS_{IDLE} × t_{KCYM} – 20		ns

Note 1. When the setting of CSIHnCFGx.CSIHnCKPx is changed while CS_{IDLE} = 000_B.

Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

CS_{SETUP}: CSIHnCFGx.CSIHnSPx[3:0] bits

CS_{HOLD}: CSIHnCFGx.CSIHnHDx[3:0] bits

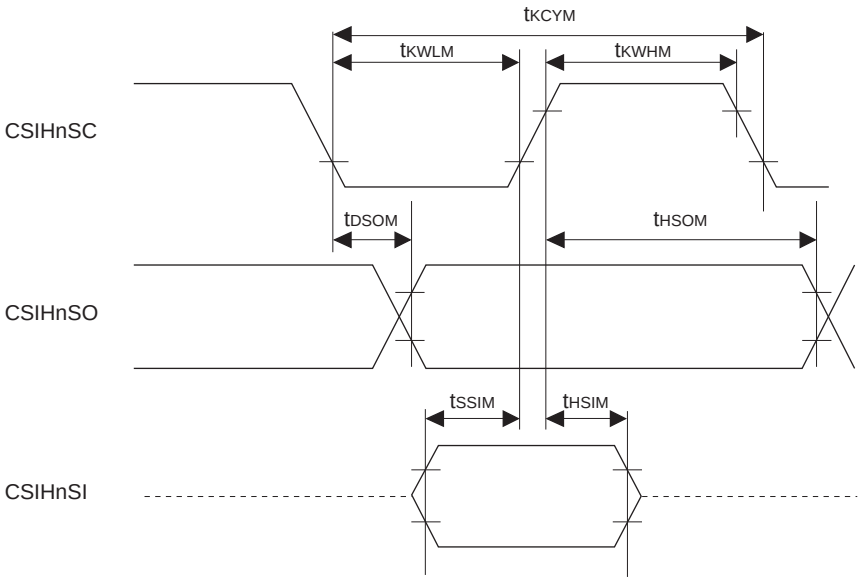
CS_{IDLE}: CSIHnCFGx.CSIHnIDx[2:0] bits

t_{PCLK}: PCLK cycle time

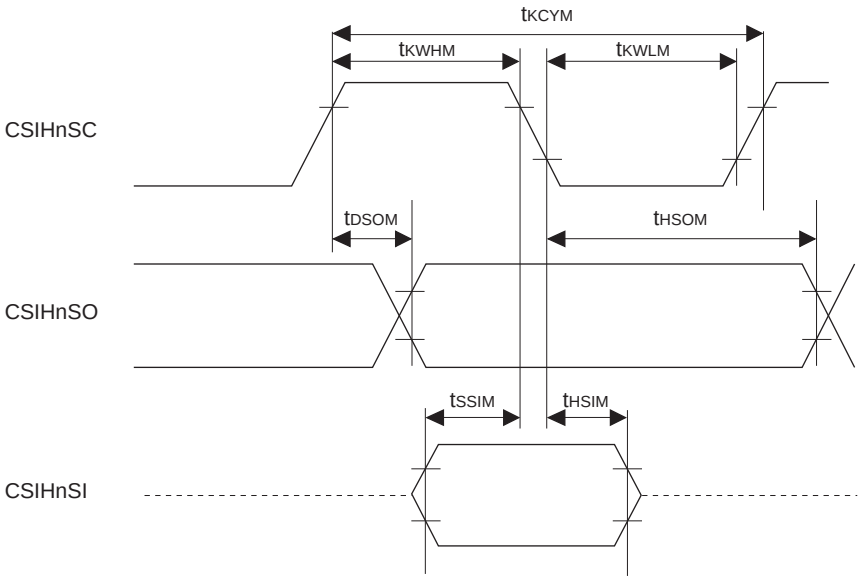
[CSIHnSC/CSIHnSI/CSIHnSO]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0 or 1/1)



Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 1/0 or 0/1)

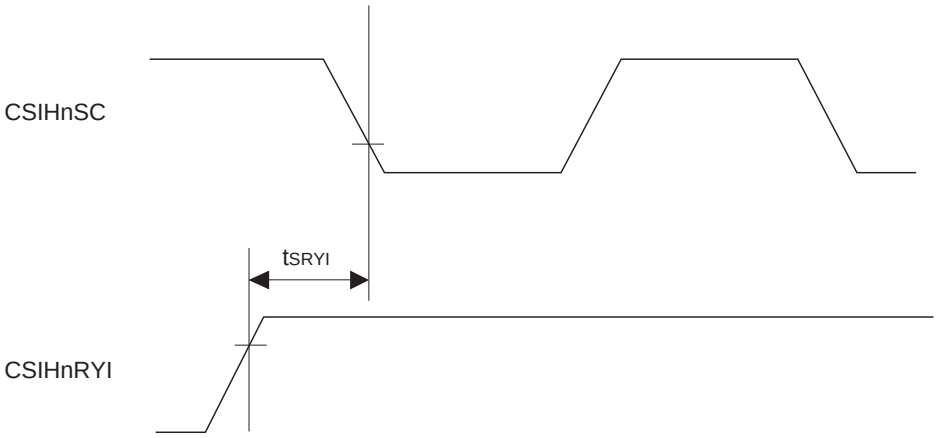


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

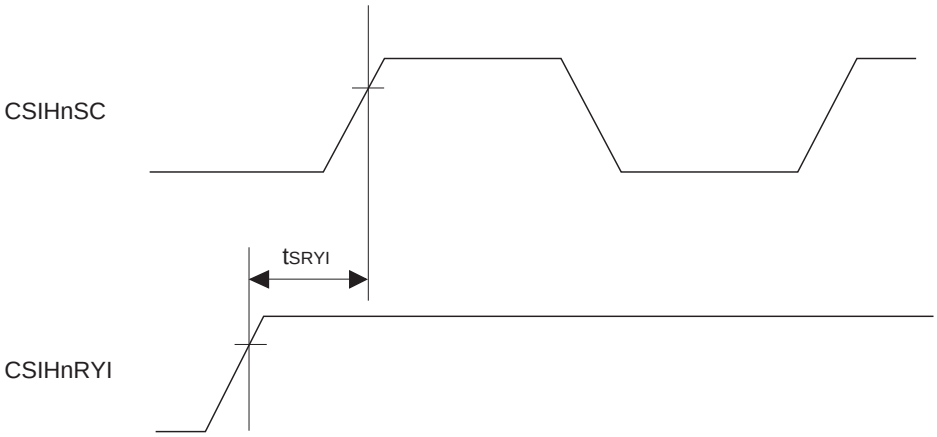
[CSIHnSC/CSIHnRYI]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx = 0)



Master mode: (CSIHnCFGx: CSIHnCKPx = 1)

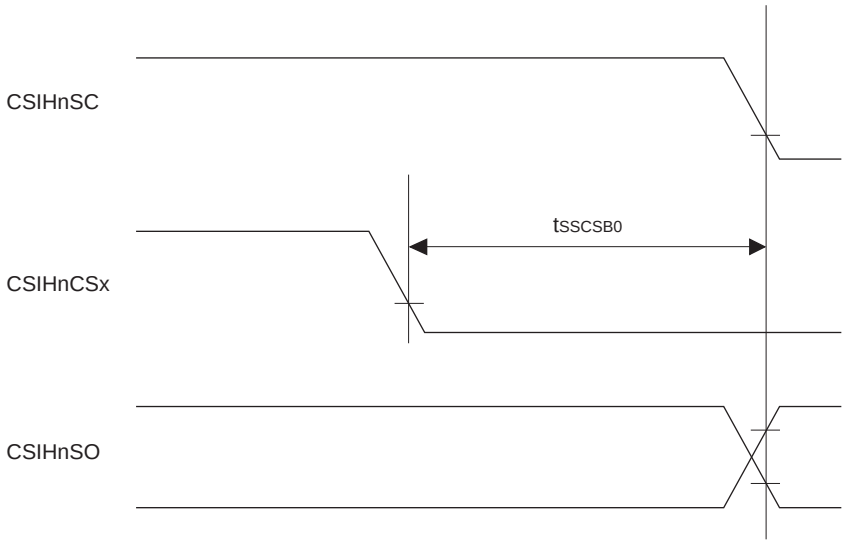


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

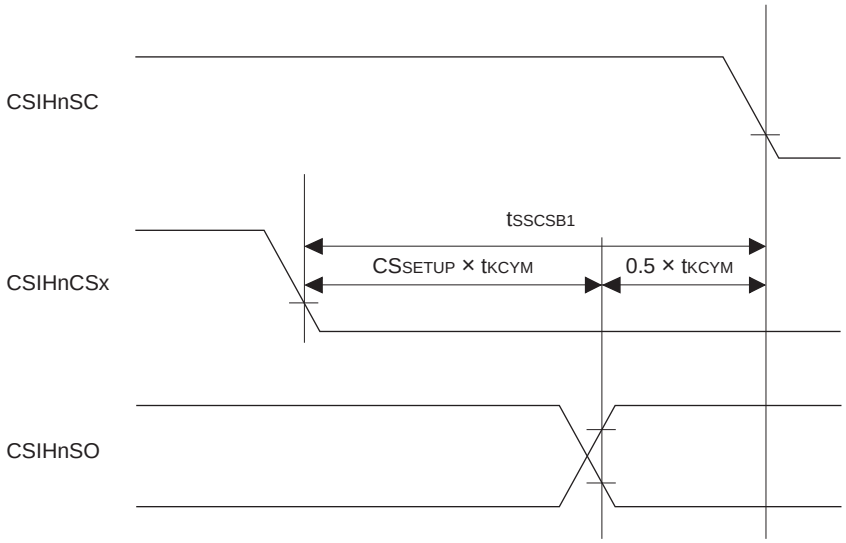
[CSIHnSC/CSIHnSO/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/1)

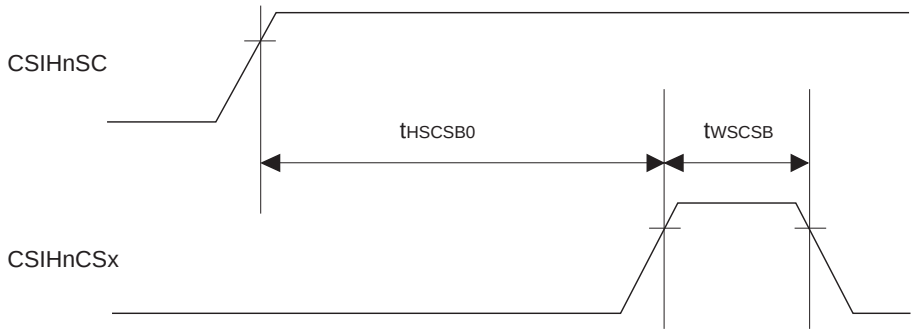


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

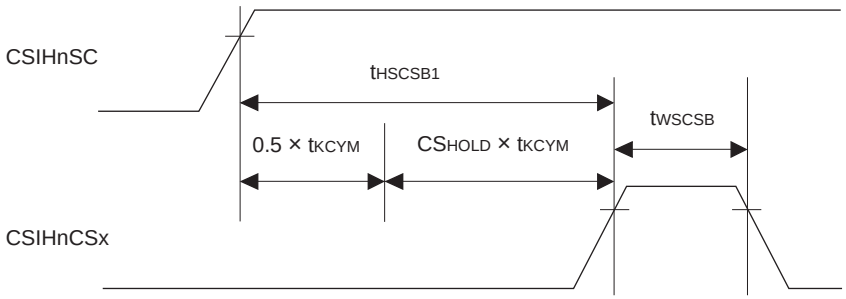
[CSIHnSC/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

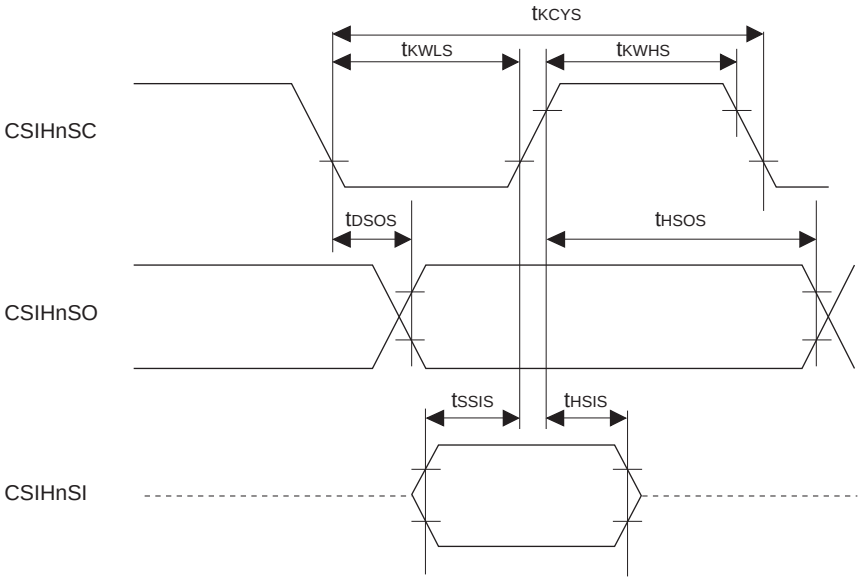
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYS}	Input	PCLK = 80 MHz	150		ns
CSIHnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIS}			15		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOS}				30	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOS}	CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 0/0 or 1/1		t_{KWHS}		ns
		CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 1/0 or 0/1		t_{KWLS}		ns
CSIHnRYO output delay	t_{SRYO}				30	ns

Note n = 0, 1

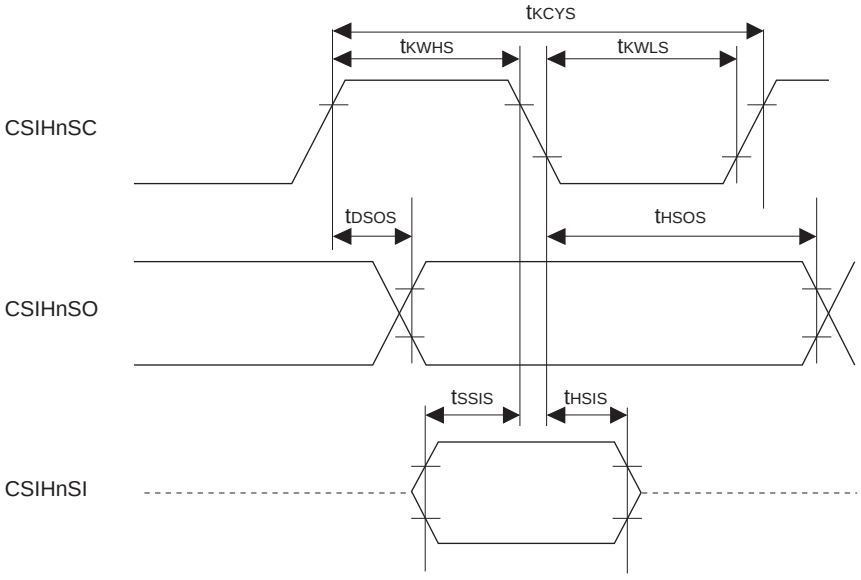
t_{PCLK}: PCLK cycle time

[CSIHnSC/CSIHnSI/CSIHnSO]

Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



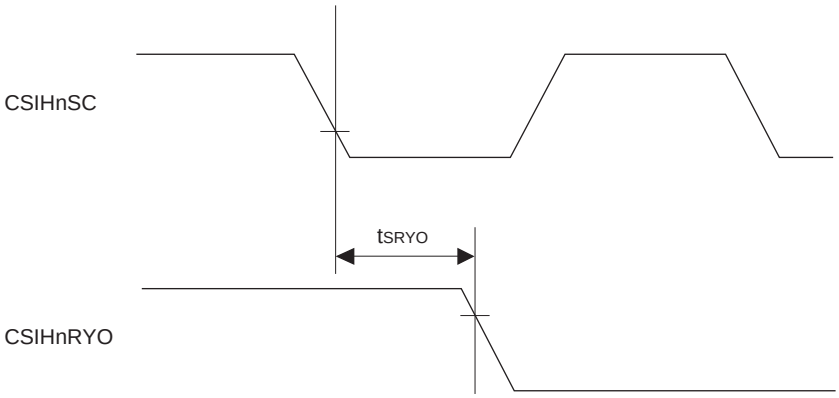
Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 1/0 or 0/1)



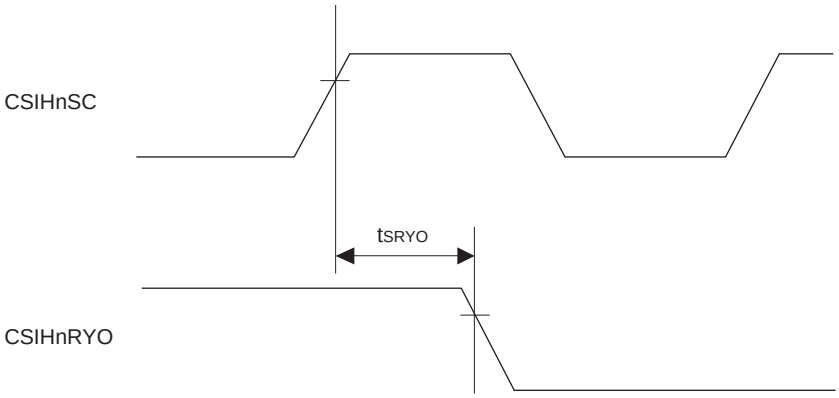
Note n = 0, 1

[CSIHnSC/CSIHnRYO]

(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/1 or 1/0)



Note n = 0, 1

28.4.6.7 CSIGn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

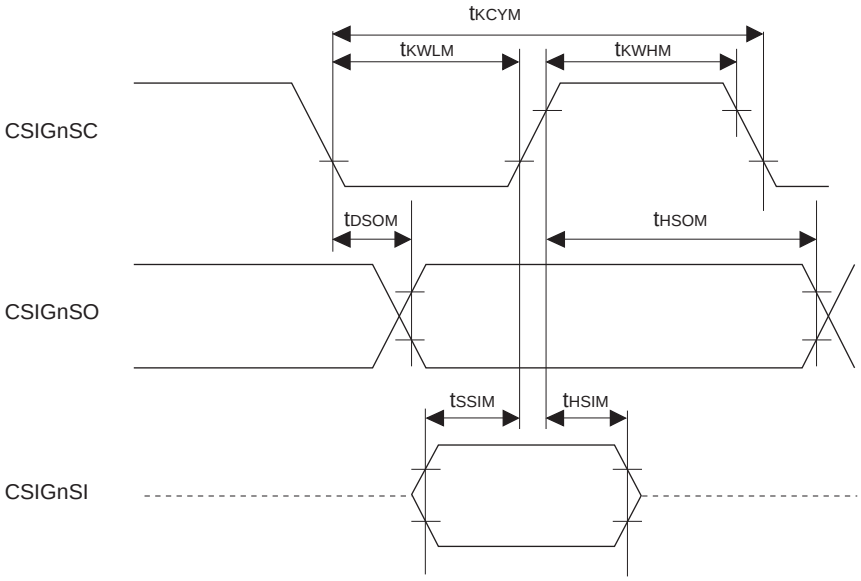
Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYM}	Output	125		ns
CSIGnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIM}		30		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIM}		10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOM}			10	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOM}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1	t_{KWHM}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1	t_{KWLM}		ns
CSIGnRYI setup time (to CSIGnSC)	t_{SRYI}		($2 \times t_{PCLK}$) + 30		ns

Note n = 0 to 2

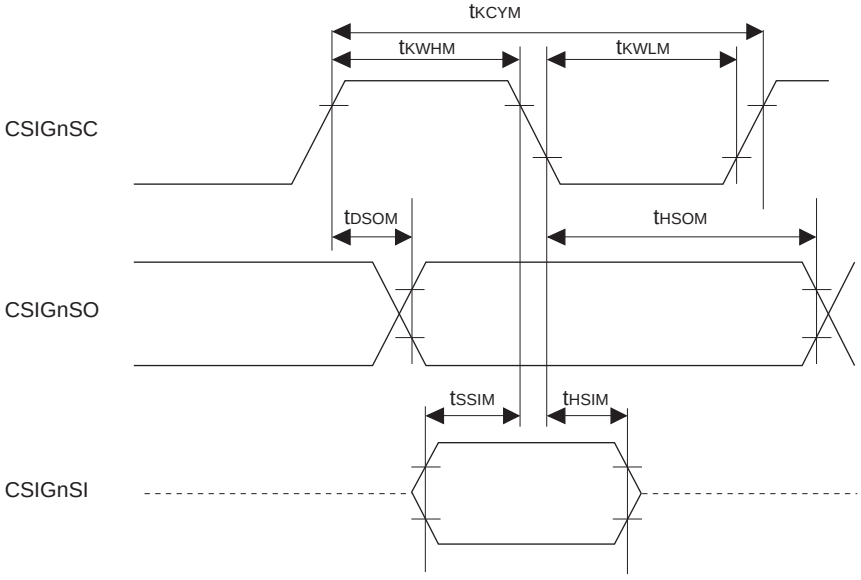
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



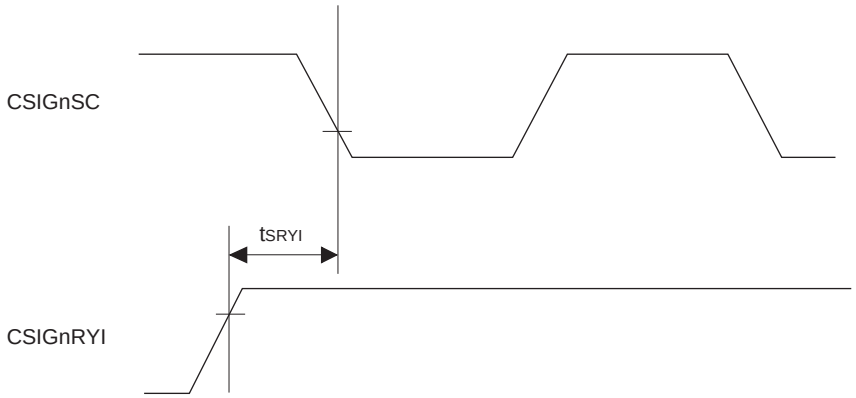
Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



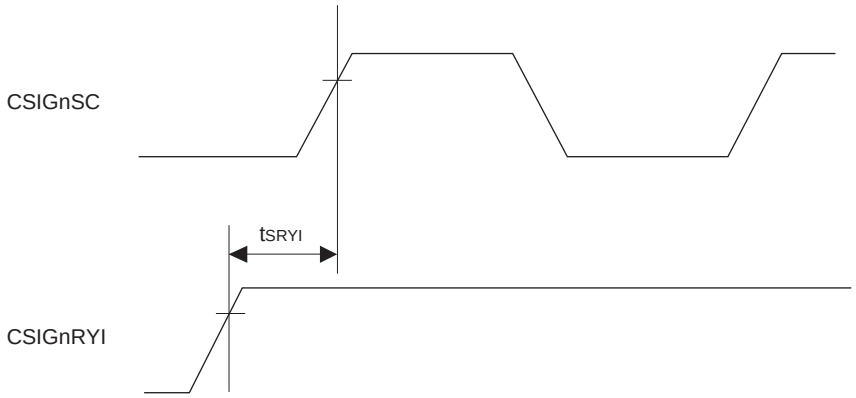
Note n = 0 to 2

[CSIGnSC/CSIGnRYI]

Master mode: (CSIGnCTL1: CSIGnCKR = 0)



Master mode: (CSIGnCTL1: CSIGnCKR = 1)



Note n = 0 to 2

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

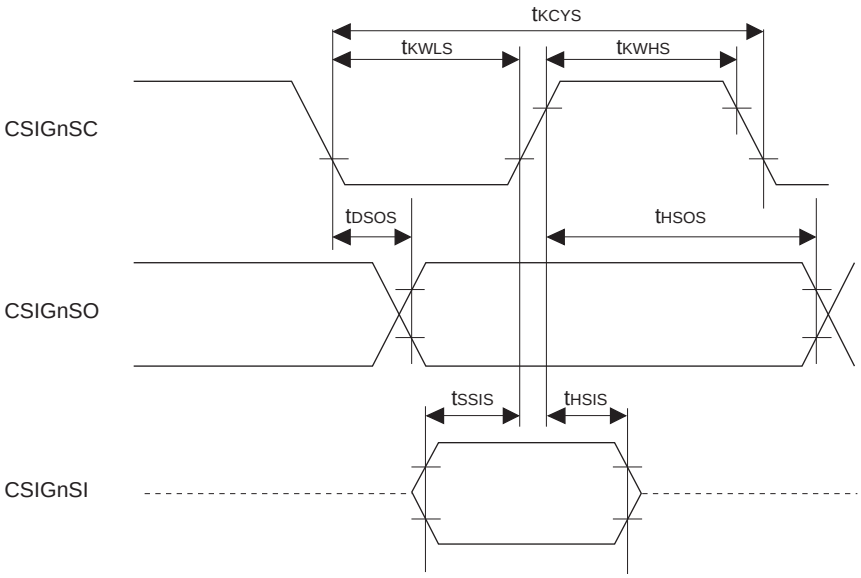
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYS}	Input	PCLK = 80 MHz	150		ns
CSIGnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIS}			15		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOS}				30	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOS}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1		t_{KWHS}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1		t_{KWLS}		ns
CSIGnRYO output delay	t_{SRYO}				30	ns

Note n = 0 to 2

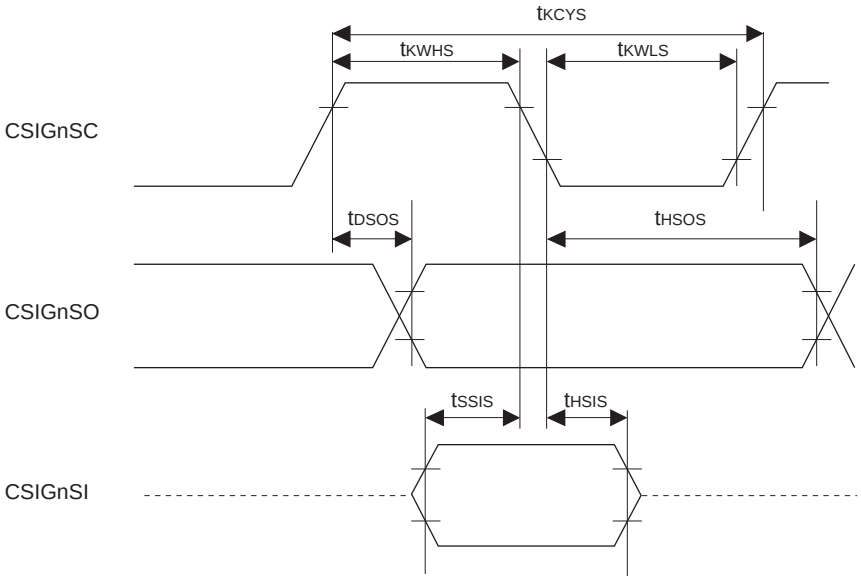
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



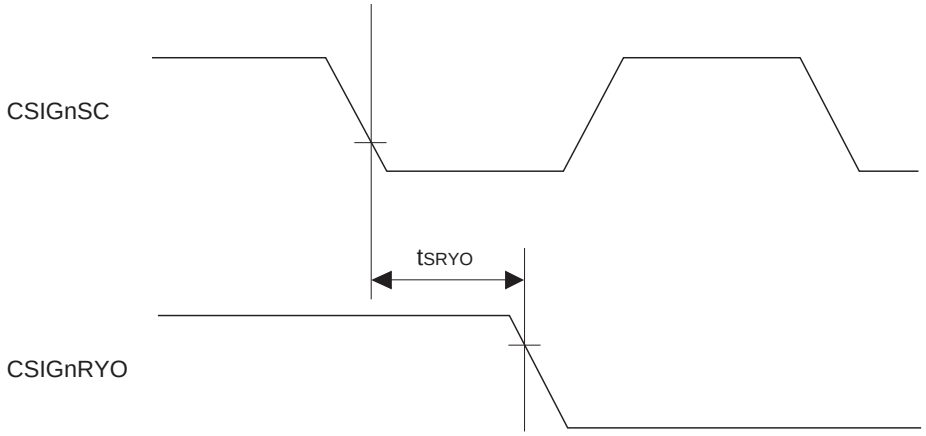
Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



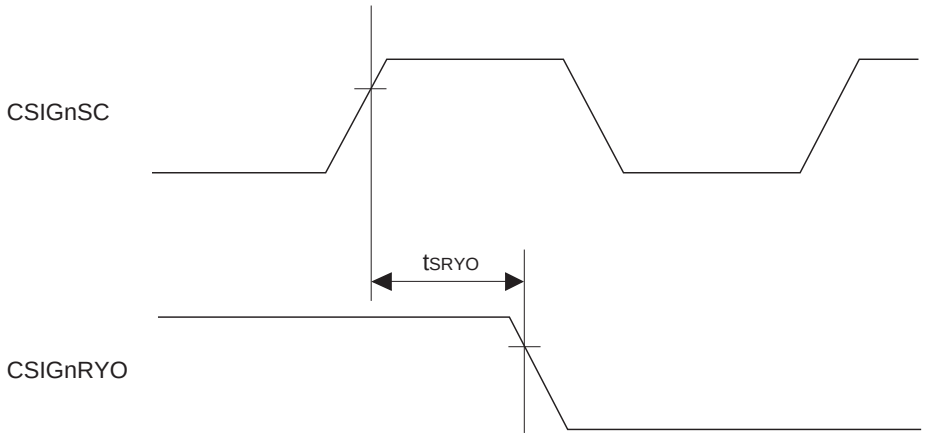
Note n = 0 to 2

[CSIGnSC/CSIGnRYO]

(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1 or 1/0)



Note n = 0 to 2

28.4.6.8 UARTHn Timing

(a) Asynchronous Mode

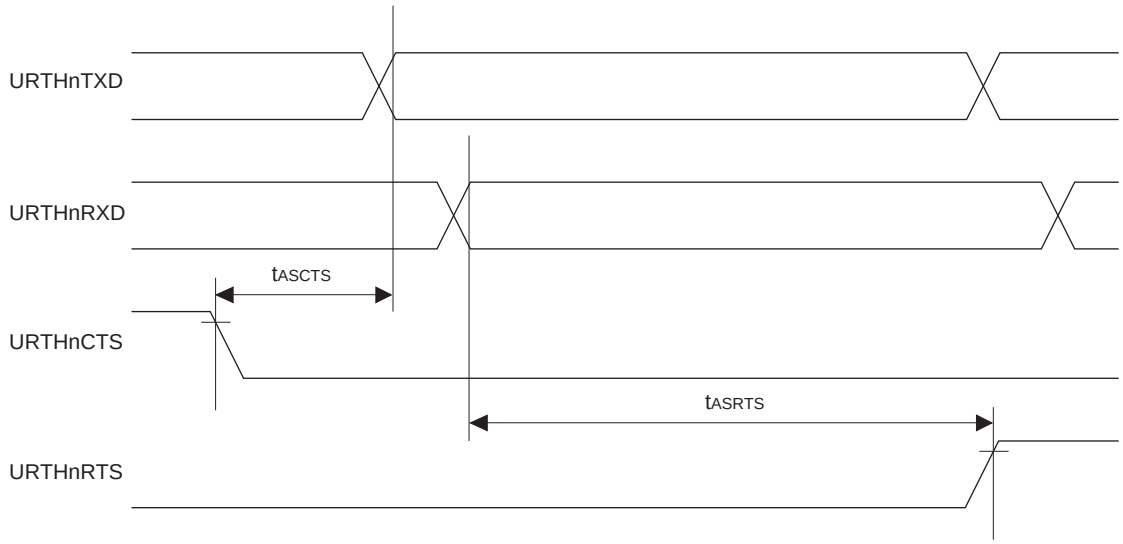
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		MIN.	MAX.	Unit
UARTHn transfer rate				16	2M	bps
URTHnCTS setup time (to URTHnTXD)	t_{ASCTS}	Input	URTHnCTS without noise filter	3T + 30		ns
			URTHnCTS with noise filter	5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{ASRTS}	Output	URTHnRXD without noise filter		(3BRS + 2) T + 30	ns
			URTHnRXD with noise filter		(3BRS + 4) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]



(b) Clock Synchronous, Master Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM1}	Output		500		500		ns
URTHnSC high level width	t_{KWHM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnSC low level width	t_{KWLM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDM1}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDM1}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM1}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM1}	Output		t_{KCYM1}/2 - 10		t_{KCYM1}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM1}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM1}	Output			T + 30		T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

(c) Clock Synchronous, Slave Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnSC/URTHnRXD without Noise Filter		URTHnSC/URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		10		10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		2T + 10		2T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		-T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	2T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		2T + 30	3T + 30	2T + 30	3T + 30	ns

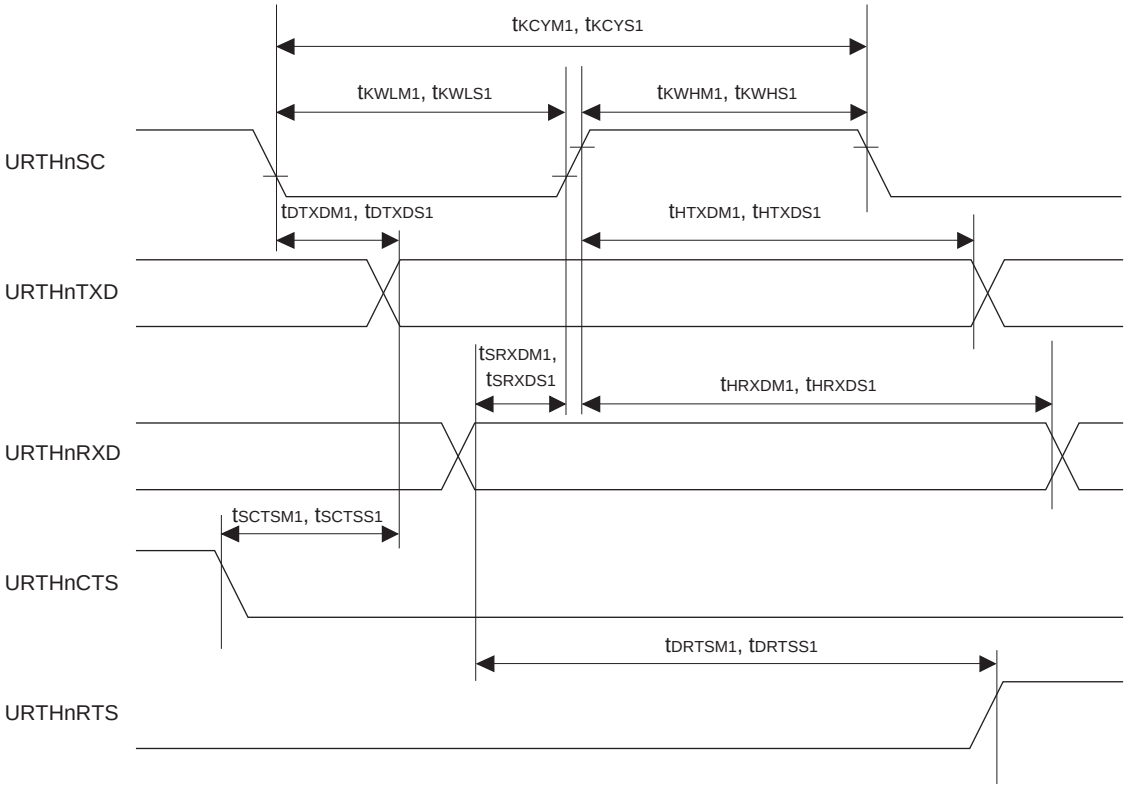
Note n = 0 to 2

T = prescaler cycle time

Item	Symbol	Conditions		URTHnSC without Noise Filter URTHnRXD with Noise Filter		URTHnSC with Noise Filter URTHnRXD without Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		2T + 10		-2T + 10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		10		4T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	3T + t_{KCYS1} + 30		3T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		4T + 30	5T + 30	30	T + 30	ns

Note n = 0 to 2

T = prescaler cycle time


tKCYM1, tKCYS1
tKWLM1, tKWLS1
tKWHM1, tKWHS1
URTHnSC
tDTXDM1, tDTXDS1
tHTXDM1, tHTXDS1
URTHnTXD
tSRXDM1,
tSRXDS1
tHRXDM1, tHRXDS1
URTHnRXD
tSCTSM1, tSCTSS1
URTHnCTS
tDRTSM1, tDRTSS1
URTHnRTS

Note n = 0 to 2

(d) Clock Synchronous, Master Mode (Delay Sampling Mode)

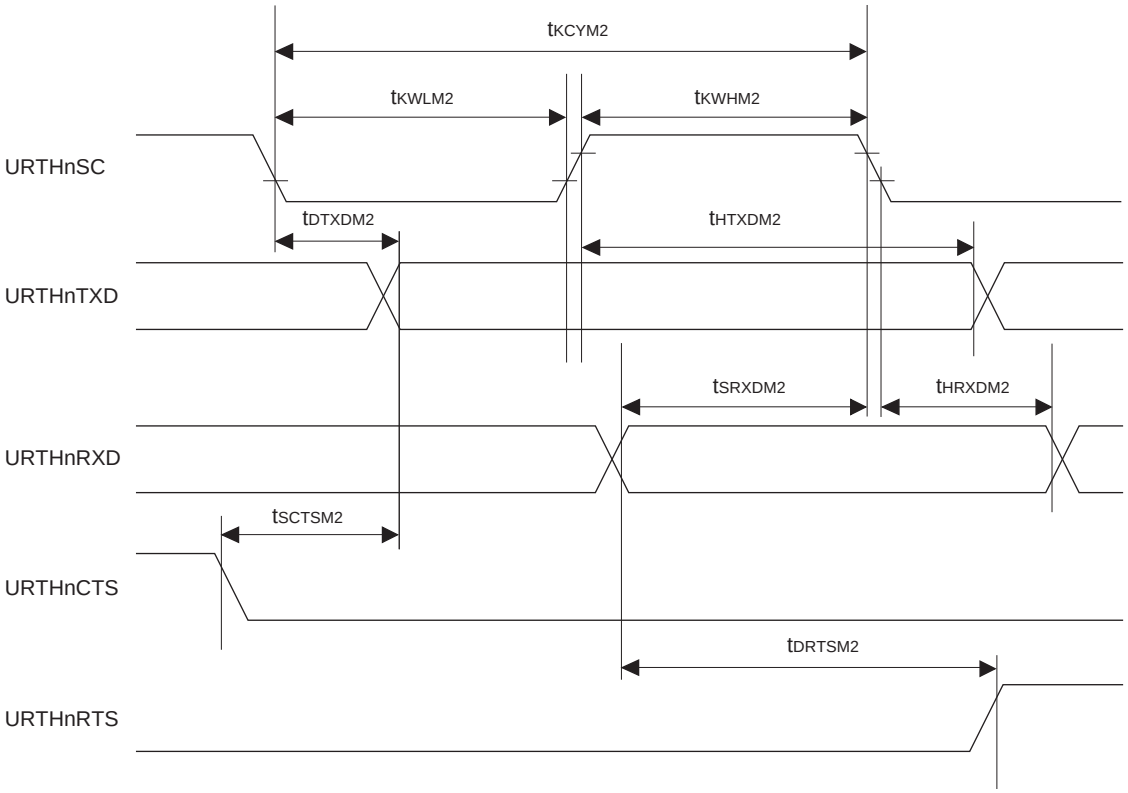
(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM2}	Output		500		500		ns
URTHnSC high level width	t_{KWHM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnSC low level width	t_{KWLM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↓)	t_{SRXDM2}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↓)	t_{HRXDM2}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM2}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM2}	Output		t_{KCYM2}/2 - 10		t_{KCYM2}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM2}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM2}	Output			(BRS + 1) T + 30		(BRS + 1) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]


tKCYM2
tKWLM2
tKWHM2
URTHnSC
tDTXDM2
tHTXDM2
URTHnTXD
tSRXDM2
tHRXDM2
URTHnRXD
tSCTSM2
URTHnCTS
tDRTSM2
URTHnRTS

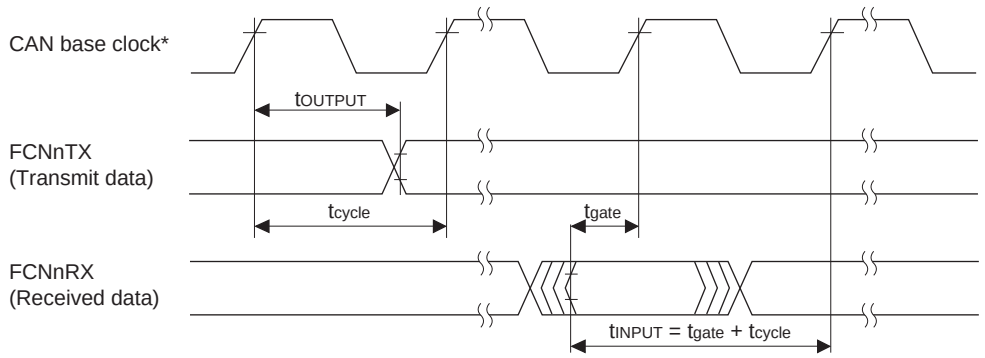
Note n = 0 to 2

28.4.6.9 CAN Timing

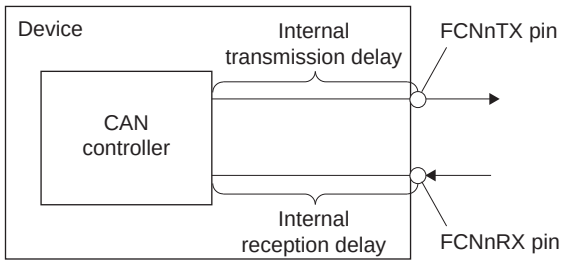
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CAN transfer rate				1	Mbps
CAN internal delay time*[1]	t_{node}			87.5	ns

Note 1. CAN internal delay time (t_{node}) = t_{OUTPUT} (internal transmission delay) + t_{INPUT} (internal reception delay)



Note * CAN base clock frequency: f_{CAN}



Schematic view of the CAN-internal delay

Note n = 0, 1

28.4.6.10 FlexRay Timing

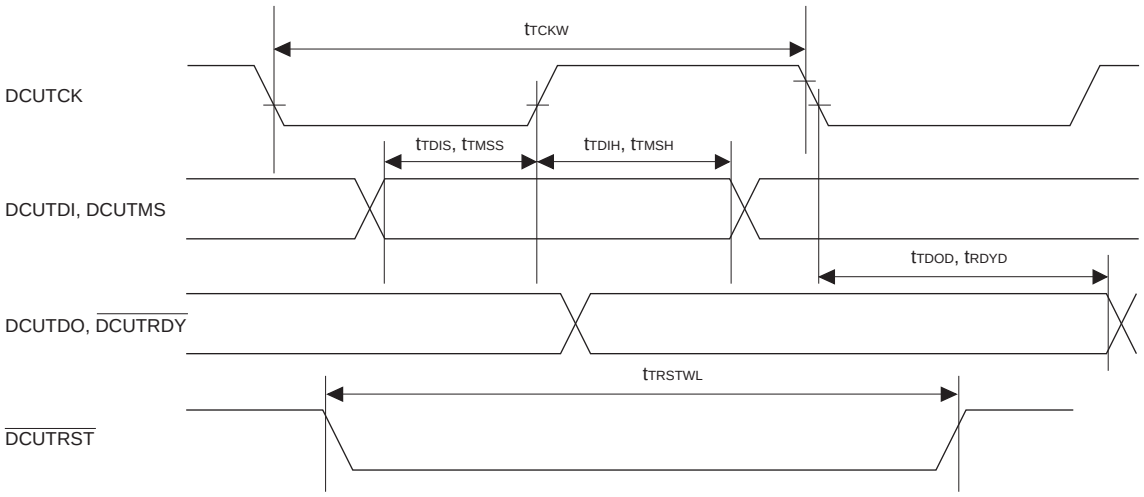
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				10	Mbps

28.4.6.11 NEXUS I/F Timing

(T_a = 0 to +40°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	MIN.	MAX.	Unit
DCUTCK cycle time	t_{TCKW}	40.0		ns
DCUTDI setup time (to DCUTCK ↑)	t_{TDIS}	t_{TCKW}/2 - 8.5		ns
DCUTMS setup time (to DCUTCK ↑)	t_{TMSS}	t_{TCKW}/2 - 8.5		ns
DCUTDI hold time (to DCUTCK ↑)	t_{TDIH}	2.0		ns
DCUTMS hold time (to DCUTCK ↑)	t_{TMSH}	2.0		ns
DCUTDO output delay time (to DCUTCK ↓)	t_{TDOD}		20.0	ns
$\overline{\text{DCUTRDY}}$ output delay time (to DCUTCK ↓)	t_{RDYD}		20.0	ns
$\overline{\text{DCUTRST}}$ low level width	t_{TRSTWL}	1000		ns



28.4.6.12 A/D Converter Characteristics

(a) 10-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.0	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Zero-scale error*1	ZSE					±1.5	LSB
Full-scale error*1	FSE					±1.5	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} Conversion	1003		1023	LSB
			2/3AV_{DD0} Conversion	673	683	693	LSB
			1/2AV_{DD0} Conversion	504	512	520	LSB
			1/3AV_{DD0} Conversion	331	341	351	LSB
			AV_{SS0} Conversion	0		20	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(b) 10-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.5	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±2.0	LSB
Full-scale error*1	FSE					±2.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±2.0	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} Conversion	671	683	695	LSB
			1/2AV_{DD0} Conversion	502	512	522	LSB
			1/3AV_{DD0} Conversion	329	341	353	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

Caution 1. Implement the following countermeasures to minimize any adverse effect on precision of the A/D converter from the switching noise of the digital circuit.

i. Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

ii. Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, FV_{DD}, or AV_{DD0}) and ground pin (V_{SSM}, V_{SSC}, EV_{SS}, $OSCV_{SS}$, $CPUPLLV_{SS}$, $FRPLLV_{SS}$, FV_{SS}, or AV_{SS0}), and between the analog reference voltage pins (AV_{REF0P}, or AV_{REF0M}) and the analog ground pin (AV_{SS0}).

iii. We recommend isolating the analog power supply (AV_{DD0}) from the digital power supply (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, or FV_{DD}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board. Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

Caution 2. Connect resistor Re and capacitor Ce to the analog input pins (ADCA0Im) externally to eliminate noise (m = 01 to 22).

Note Full-scale range (%FSR) is the ratio, as a percentage, of the extremes of the range of analog input voltages that can be converted, and is expressed as follows regardless of the resolution.

1%FSR = (Maximum value of convertible analog input voltage - Minimum value of convertible analog input voltage)/100

= (AV_{REF0P} - 0)/100

= AV_{REF0P}/100

1 LSB is as follows at a resolution of 10 bits:

1 LSB = $1/2^{10}$

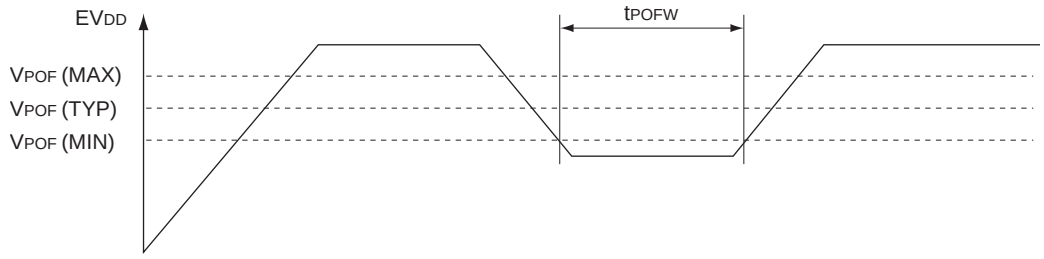
= 0.098%FSR

28.4.6.13 POF/LVI Characteristics

POF Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{POF}		2.7	2.8	2.9	V
Minimum detection width	t_{POFW}		0.2			ms

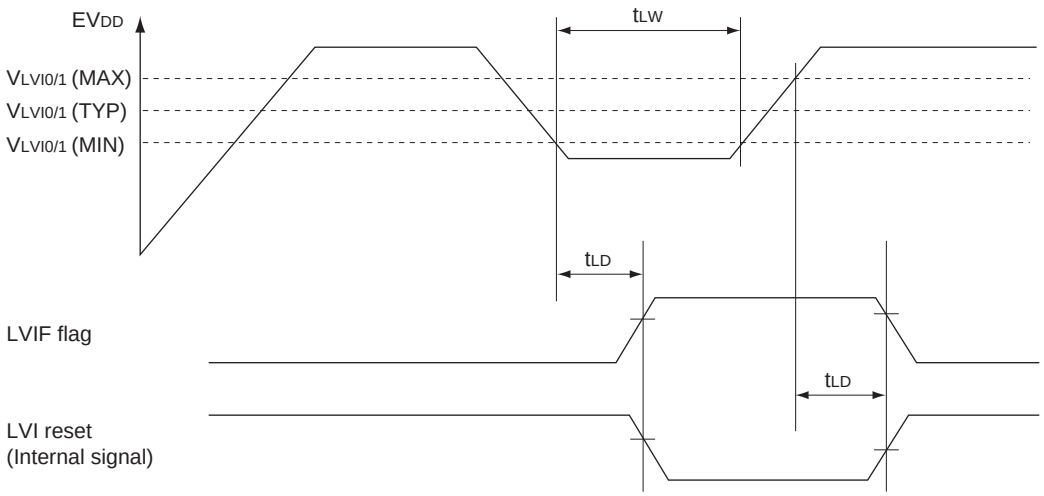


LVI characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{LVI2}	LVICNT.LVICNT[1:0] = 11_B	3	3.1	3.2	V
Response time*[1]	t_{LD}	After EV_{DD} reaches V_{LVI2} (MAX.) After EV_{DD} falls V_{LVI2} (MIN.) Voltage gradient: 1.8 V/us to 1.8 V/s			2	ms
Minimum detection width	t_{LW}		2.0			ms
Stabilization waiting time*[2]					350	µs

Note 1. Time until output of an interrupt or reset after voltage detection.
Note 2. Time until LVI is detected in accord with the specification after enabling of the LVI function.


EVDD
tLW
VLVI0/1 (MAX)
VLVI0/1 (TYP)
VLVI0/1 (MIN)
tLD
LVIF flag
tLD
LVI reset
(Internal signal)

28.4.6.14 Flash Memory Programming Characteristics

(a) Basic Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}			80		160	MHz
Number of times rewritable (per block)*1	C_{ERWR}	Code Flash	Data are retained for 20 years			100	Times/ block
		Data Flash	Data are retained for 5 years			15000	Times/ block
			Data are retained for 15 years			5000	Times/ block
			Data are retained for 20 years			1000	Times/ block

Note 1. In writing to a delivered product, either "erasing → writing" or "writing alone" is considered as a "time" of rewriting.

Example: (P: writing, E: erasing)

Delivered product→ P→E→P→E→P: 3 times

Delivered product→ E→P→E→P→E→P: 3 times

28.4.6.15 FLMD0 Pulse Timing Characteristics

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 3.0 to 3.6 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (to VDD)	t_{DP}		1			ms
$\overline{\text{RESET}}$ release (to FLMD0)	t_{PR}		2			ms
$\overline{\text{RESET}}\uparrow \rightarrow$ FLMD0 set time	t_{RP}		1200			µs
FLMD0 high/low level width	t_{PW}		10		100	µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns
Serial clock input frequency	t_{SCCY}		9.8		5000	KHz
High level width	t_{WHSC}		85		50000	ns
Low level width	t_{WLSC}		85		50000	ns
SI setup time (to SCK ↑)	t_{PLHSC}		75			ns
SI hold time (to SCK ↑)	t_{PHLSC}		0			ns
SO output delay time	t_{DSC}				23	ns
UART transfer rate			9.6		1000	Kbps

Note VDD: $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$

28.4.6.16 BIST Execution Time

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 3.0 to 3.6 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition 1	Condition2	MIN.	TYP.	MAX.	Unit
BIST execution time		All HW-BIST Total fxx = 80 MHz				40	ms
		All HW-BIST Total fxx = 160 MHz				20	ms

28.5 Electrical Characteristics of the µPD70F3509 (Target)

28.5.1 Absolute Maximum Ratings

(T_a = 25°C)

Item	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DDM}	$V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +1.6	V
	V_{DDC}		-0.5 to +1.6	V
	$CPUPLLV_{DD}$		-0.5 to +1.6	V
	$FRPLLV_{DD}$		-0.5 to +1.6	V
	EV_{DD}	$EV_{DD} = OSCV_{DD} = FV_{DD}$ $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +6.0	V
	$OSCV_{DD}$		-0.5 to +6.0	V
	FV_{DD}		-0.5 to +6.0	V
	AV_{DD0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V	-0.5 to +6.0	V
	V_{SSM}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	V_{SSC}		-0.5 to +0.5	V
	$CPUPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	$FRPLLV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	EV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	$OSCV_{SS}$	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	FV_{SS}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
	AV_{SS0}	$V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$	-0.5 to +0.5	V
Input voltage	V_I		-0.5 to EV_{DD} + 0.5*1	V
Analog input voltage	V_{IAN}	ADCA0I1 to ADCA0I22	-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0P}		-0.5 to AV_{DD0} + 0.5*1	V
	AV_{REF0M}		-0.5 to AV_{DD0} + 0.5*1	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total pins	-50	mA
Low-level output current	I_{OL}	1 pin	10	mA
		Total pins	50.	mA
Operating ambient temperature	T_a		-40 to +125	°C
Storage temperature	T_{stg}	Before writing to flash memory	-65 to +150	°C
		After writing to flash memory	-65 to +125	°C

Note 1. Be sure not to exceed the absolute maximum ratings of each supply voltage.

Caution 1. Make sure that the total the current flowing through these pins and output currents does not exceed 50 mA.

Caution 2. Do not directly connect output (or I/O) pins of IC products to each other, or to VDD, VCC, and GND. Open drain pins or open collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

Caution 3. Product quality may suffer if the absolute maximum rating is exceeded even momentary for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

28.5.2 Capacity

(T_a = 25°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz 0 V for non-measured pins, and excluding analog pins selected for use with S/H channels or AD converter channels.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

28.5.3 Operating Conditions

(T_a = -40 to +125°C,V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Internal Operation Clock Frequency	Operating Ambient Temperature (T_a)	Power Supply Voltage
fxx = 80 MHz	T_a = -40 to +125°C	V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V
fxx = 160 MHz		

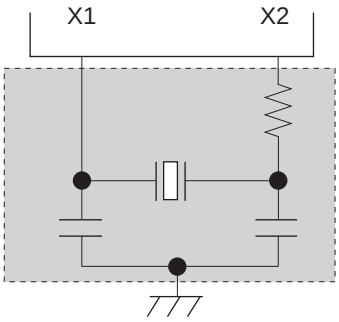
28.5.4 Characteristics of Oscillation Circuit

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency*1	fx	µPD70F3509M1	8	8	8	MHz
		µPD70F3509M2	16	16	16	MHz

Note 1. The values are only applicable as oscillation circuit characteristics.

Caution 1. When placing wiring in the area enclosed by the broken line in the figure below during the design of the oscillator, avoid adverse effects from wiring capacitance by following the directions below.



- Bring the oscillation circuit as close as possible to the X1 and X2 terminals.
- Do not pass other signal conductors through the broken line part in the above figure.
- Do not route the wiring near signal lines through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as $OSCV_{SS}$.
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When selecting the crystal oscillator, designing the parameters of the oscillation circuit, or calculating the time required for oscillation to become stable, exhaustively evaluate the finished system (MCU, oscillator, and target board) for matching of the device and the oscillator or ask the manufacturer of the oscillator to do so.

28.5.5 DC Characteristics

DC Characteristics (1/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-level input voltage	V_{IH1}	Port pins, FLMD0 pin	0.7 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH2}	FLMD1 pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
	V_{IH3}*5	DCUTDI, DCUTMS DCUTCK, $\overline{\text{DCUTRST}}$	2.2		EV_{DD} + 0.3	V
	V_{IH4}	$\overline{\text{RESET}}$ pin	0.8 × EV_{DD}		EV_{DD} + 0.3	V
Low-level input voltage	V_{IL1}	Port pins, FLMD0 pin	-0.5		0.3 × EV_{DD}	V
	V_{IL2}	FLMD1 pin	-0.5		0.2 × EV_{DD}	V
	V_{IL3}*5	DCUTDI, DCUTMS DCUTCK	-0.5		0.8	V
	V_{IL4}	$\overline{\text{RESET}}$ pin, $\overline{\text{DCUTRST}}$	-0.5		0.6	V
Hysteresis width*1	V_{HYS1}	Port pins, FLMD0 pin		0.5		V
	V_{HYS2}	FLMD1 pin		1.0		V
	V_{HYS3}	$\overline{\text{RESET}}$ pin		1.0		V
High-level output voltage*2	V_{OH1}	I_{OH} = - 3mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 1mA	EV_{DD} - 1.0		EV_{DD}	V
		I_{OH} = - 0.1mA	EV_{DD} - 0.5		EV_{DD}	V
Low-level output voltage*2	V_{OL1}	I_{OL} = 3mA	0		0.4	V
		I_{OL} = 1mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V, other than for the pins below	20	40	100	KΩ
	R_2	V_I = 0 V *6	10	19	50	KΩ
Pull-down resistor	R_3	V_I = EV_{DD}*3	20	40	100	KΩ
	R_4	V_I = EV_{DD}*4	25	40	100	KΩ
	R_5	V_I = EV_{DD}*6	10	19	50	KΩ

Note 1. When the operation of an adjacent pin is halted.
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.
Note 3. Only for the FLMD1 and $\overline{\text{DCUTRST}}$ pins
Note 4. Only for the $\overline{\text{RESET}}$ pin
Note 5. In debugging mode
Note 6. Only for the FLMD0 pin

Caution The characteristics of the alternative function pins are the same as those of the port pin unless otherwise specified.

DC Characteristics (2/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-level input leakage current	I_{LIH1}	V_{IAN} = AV_{DD0}	Analog pins*1			0.3	µA
		V_I = EV_{DD}	Pins other than analog pins*2			5	µA
Low-level input leakage current	I_{LIL1}	V_{IAN}= 0 V	Analog pins*1			-0.3	µA
		V_I = 0 V	Pins other than analog pins*2			-5	µA
High-level output leakage current	I_{LOH1}	V_I = EV_{DD}				5	µA
Low-level output leakage current	I_{LOL1}	V_I = 0 V				-5	µA

Note 1. ADCA0I1 to ADCA0I22
Note 2. The current flowing through the pull-up/pull-down resistors is not included.

DC Characteristics (3/3)

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power-supply current*1	Normal operating conditions	I_{DD}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			350	mA
				fxx = 160 MHz			500	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA
	When rewriting flash memory	I_{DDF}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			325	mA
				fxx = 160 MHz			450	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA
	When executing BIST	I_{DDB}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			460	mA
				fxx = 160 MHz			730	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})				40	mA
	When HALT	I_{DDH}	Internal power supply current (V_{DDM}, V_{DDC} + $CPUPLLV_{DD}$ + $FRPLLV_{DD}$)	fxx = 80 MHz			315	mA
				fxx = 160 MHz			440	mA
		I_{DD5}	External power supply current (EV_{DD} + $OSCV_{DD}$ + FV_{DD} + AV_{DD0})	α: Port drive current*2			80 + α	mA

Note 1. The current flowing through the pull-up/pull-down resistors is not included. The value for current is applicable when PLCK = 1/2 HEAPCLK.

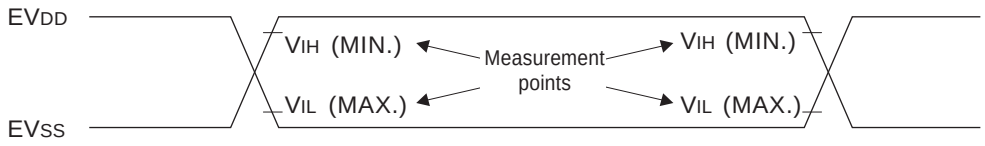
Note 2. Make sure that the total current flowing through these pins and output currents does not exceed 50 mA.

Note MAX. is the maximum value of average current.

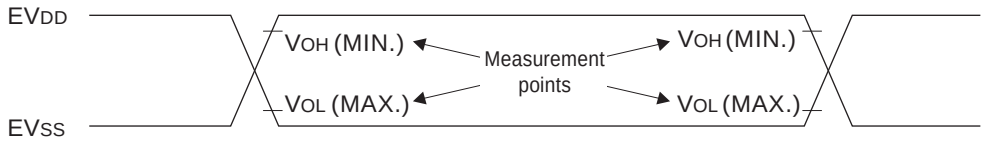
28.5.6 AC Characteristics

28.5.6.1 Measurement Conditions

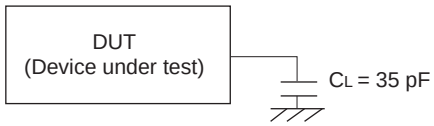
Input waveform for AC testing



AC test output measurement points



Load conditions

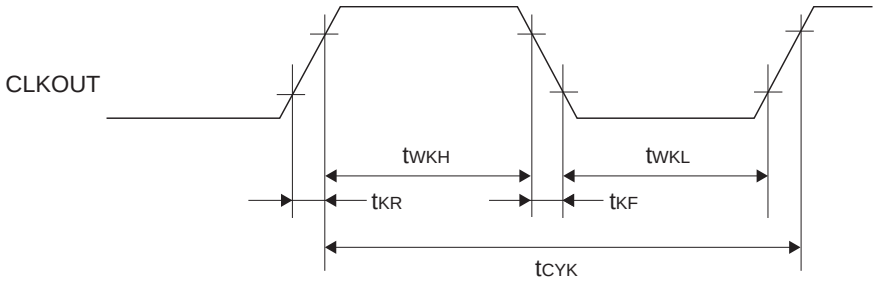


Caution If the load capacitance exceeds 35 pF due to the circuit configuration, bring the load capacitance of the device to 35 pF or less by inserting a buffer or by some other means.

28.5.6.2 Clock Timing (CLKOUT)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CLKOUT output cycle time	t_{CYK}		0.1	4	μs
CLKOUT high level width	t_{WKH}		t_{CYK}/2 - 13		ns
CLKOUT low level width	t_{WKL}		t_{CYK}/2 - 13		ns
CLKOUT rise time	t_{KR}			13	ns
CLKOUT fall time	t_{KF}			13	ns



28.5.6.3 Timing in Turning the Power Supply On and Off

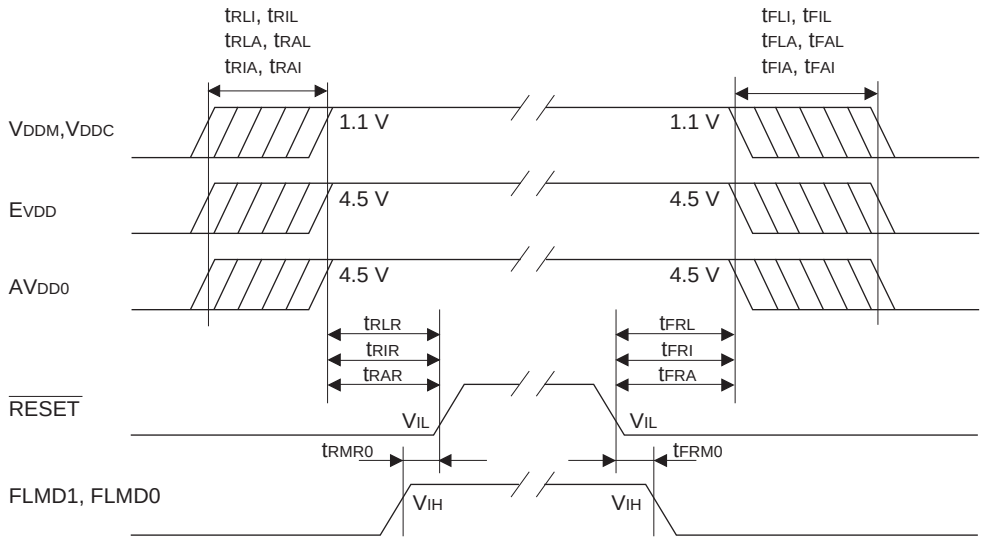
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
V_{DDM}, V_{DDC} ↑ to EV_{DD} ↑	t_{RLI}		0	1000	ms
EV_{DD} ↑ to V_{DDM}, V_{DDC} ↑	t_{RIL}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to AV_{DD0} ↑	t_{RLA}		0	1000	ms
AV_{DD0} ↑ to V_{DDM}, V_{DDC} ↑	t_{RAL}		0	1000	ms
EV_{DD} ↑ to AV_{DD0} ↑	t_{RIA}		0	1000	ms
AV_{DD0} ↑ to EV_{DD} ↑	t_{RAI}		0	1000	ms
V_{DDM}, V_{DDC} ↑ to $\overline{\text{RESET}}$ ↑	t_{RLR}	*1	t_{OSC} + 0.5		ms
EV_{DD} ↑ to $\overline{\text{RESET}}$ ↑	t_{RIR}	*1	t_{OSC} + 0.5		ms
AV_{DD0} ↑ to $\overline{\text{RESET}}$ ↑	t_{RAR}	*1	t_{OSC} + 0.5		ms
FLMD1,0 setup time (to $\overline{\text{RESET}}$ ↑)	t_{RMR0}		0		ns
FLMD1,0 hold time(to $\overline{\text{RESET}}$ ↓)	t_{FRM0}	*2	0		ns
$\overline{\text{RESET}}$ ↓ to V_{DDM}, V_{DDC} ↓	t_{FRL}		500		ns
$\overline{\text{RESET}}$ ↓ to EV_{DD} ↓	t_{FRI}		500		ns
$\overline{\text{RESET}}$ ↓ to AV_{DD0} ↓	t_{FRA}		500		ns
EV_{DD} ↓ to V_{DDM}, V_{DDC} ↓	t_{FIL}		0	1000	ms
V_{DDM}, V_{DDC} ↓ to EV_{DD} ↓	t_{FLI}		0	1000	ms
AV_{DD0} ↓ to V_{DDM}, V_{DDC} ↓	t_{FAL}		0	1000	ms
V_{DDM}, V_{DDC}↓to AV_{DD0} ↓	t_{FLA}		0	1000	ms
AV_{DD0} ↓ to EV_{DD} ↓	t_{FAI}		0	1000	ms
EV_{DD} ↓ to AV_{DD0} ↓	t_{FIA}		0	1000	ms

Note 1. t_{OSC}: Oscillation stabilization time
Note 2. Using the specified value is recommended for the FLMD1, 0 hold time (time until $\overline{\text{RESET}}$ falls: t_{FRM0}). If using the recommended value is difficult, change the FLMD1, 0 pins once rewriting of the flash memory is complete.

Caution 1. Turn on/off internal power system at the same time.
Caution 2. Turn on/off external power system at the same time
Caution 3. Apply an external reset ($\overline{\text{RESET}}$) whenever the power supply voltage falls below the operating voltage.
Caution 4. If the power-on flag or self-diagnosis (BIST) control registers are to be used, boot the internal power system first (when the internal power supply voltage reaches 1.1 V, set the voltage of the external power system to a voltage below the detection voltage [min: 2.7 V] for POF).

Note Internal power system: V_{DDM}, V_{DDC}, $CPUPLLV_{DD}$, $FRPLLV_{DD}$
External power system: EV_{DD}, $OSCV_{DD}$, FV_{DD}


tRLI, tRIL
tRLA, tRAL
tRIA, tRAI
tFLI, tFIL
tFLA, tFAL
tFIA, tFAI
VDDM,VDDC
EVDD
AVDD0
1.1 V
4.5 V
tRLR
tRIR
tRAR
tFRL
tFRI
tFRA
RESET
VIL
tRMR0
tFRM0
FLMD1, FLMD0
VIH

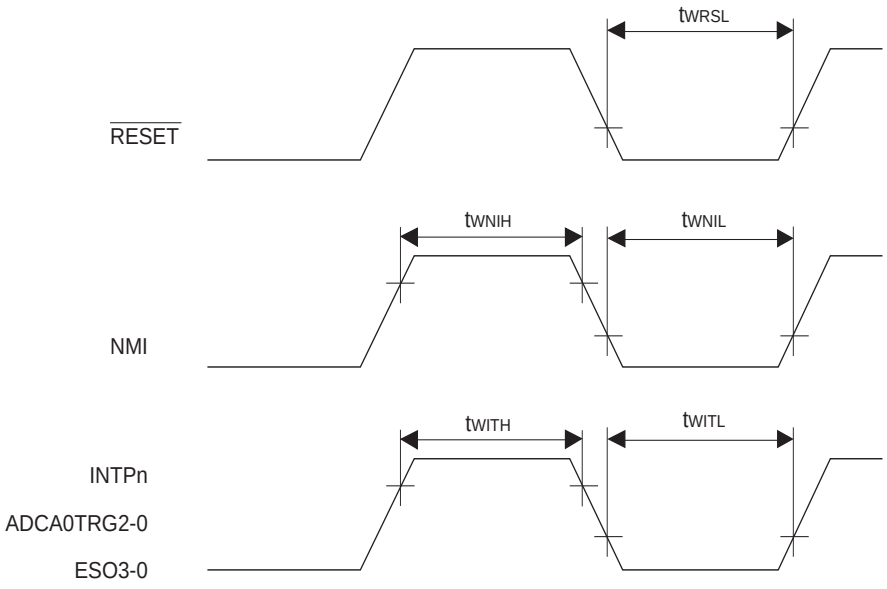
28.5.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}	Other than power supply turning on	500		ns
NMI input high level width	t_{WNIH}	Digital filter	*1		ns
NMI input low level width	t_{WNIL}	Digital filter	*1		ns
INTPn input high level width	t_{WITH}	Digital filter, n = 0 to 10	*1		ns
INTPn input low level width	t_{WITL}	Digital filter, n = 0 to 10	*1		ns
ADCA0TRG2-0 input high level width	t_{WITH}	Analog filter	250		ns
ADCA0TRG2-0 input low level width	t_{WITL}	Analog filter	250		ns
ESO3-0 input high level width	t_{WITH}	Analog filter	250		ns
ESO3-0 input low level width	t_{WITL}	Analog filter	250		ns

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.



Note n = 0 to 10

28.5.6.5 Timer Timing

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
TI input high level width	t_{TIH}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3, TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TI input low level width	t_{TIL}	TAUA0I0-15, TAUA1I0-15, TAUJ0I0-3, TAUJ1I0-3 TSG20PTSI0-2, TSG20CLKI, TSG21PTSI0-2, TSG21CLKI, ENCA0I0,1, ENCA0E0,1, ENCA0EC, ENCA1I0,1, ENCA1E0,1, ENCA1EC	*1		ns
TO output cycle time	t_{TCYK}	TAUA0O0-15, TAUA1O0-15 TAUJ0O0-3, TAUJ1O0-3 OSTM0O, OSTM1O TSG20O0-7, TSG21O0-7, TPB0O, TPB1O	0.25	1000000	µs

Note 1. (2, 3, 4, or 5) Tsamp + 10 (Tsamp is the sampling interval of the noise rejection sampling clock cycle)

Caution The above minimum values are for pulse widths that ensure detection of an effective edge. An effective edge may also be detected even if the input pulse width is less than the above minimum specification.

28.5.6.6 CSIHn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYM}	Output	125		ns
CSIHnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) – 20		ns
CSIHnSII setup time (to CSIHnSC)	t_{SSIM}		30		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIM}		10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOM}			10	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOM}	CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 0/0 or 1/1	t_{KWHM}		ns
		CSIHnCFGx.CSIHnCKPx/ CSIHnDAPx = 1/0 or 0/1	t_{KWLM}		ns
CSIHnRYI setup time (to CSIHnSC)	t_{SRYI}		($2 \times t_{PCLK}$) + 30		ns
CSIHnCSx setup time	t_{SSCSB0}	CSIHnCFGx.CSIHnDAPx = 0	$CS_{SETUP} \times t_{KCYM}$ – 10		ns
	t_{SSCSB1}	CSIHnCFGx.CSIHnDAPx = 1	(CS_{SETUP} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx hold time	t_{HSCSB0}	CSIHnCTL1.CSIHnSIT = 0	$CS_{HOLD} \times t_{KCYM}$ – 10		ns
	t_{HSCSB1}	CSIHnCTL1.CSIHnSIT = 1	(CS_{HOLD} + 0.5) × t_{KCYM} – 10		ns
CSIHnCSx inactive level width	t_{WSCSB}	*[1]	(CS_{IDLE} + 0.5) × t_{KCYM} – 20		ns
		Other than the above	$CS_{IDLE} \times t_{KCYM}$ – 20		ns

Note 1. When the setting of CSIHnCFGx.CSIHnCKPx is changed while CS_{IDLE} = 000_B.

Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

CS_{SETUP}: CSIHnCFGx.CSIHnSPx[3:0] bits

CS_{HOLD}: CSIHnCFGx.CSIHnHDx[3:0] bits

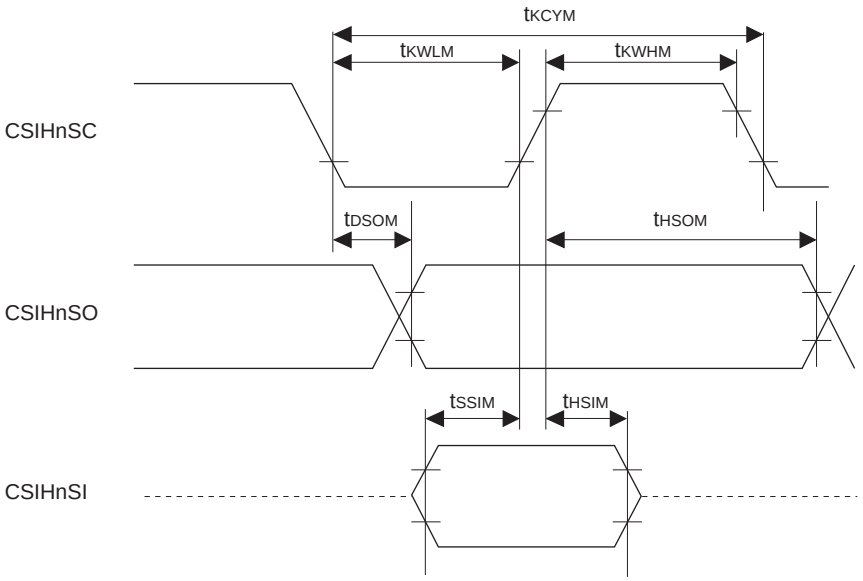
CS_{IDLE}: CSIHnCFGx.CSIHnIDx[2:0] bits

t_{PCLK}: PCLK cycle time

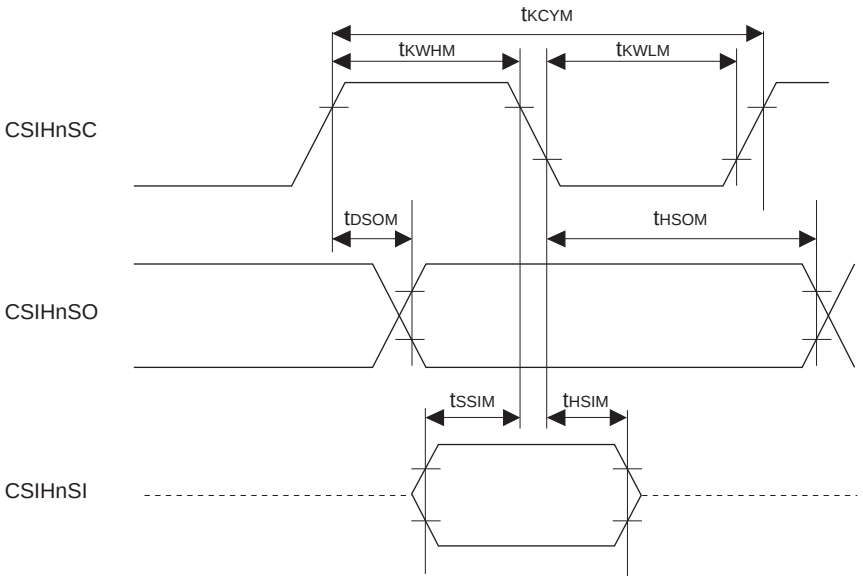
[CSIHnSC/CSIHnSI/CSIHnSO]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0 or 1/1)



Master mode: (CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 1/0 or 0/1)

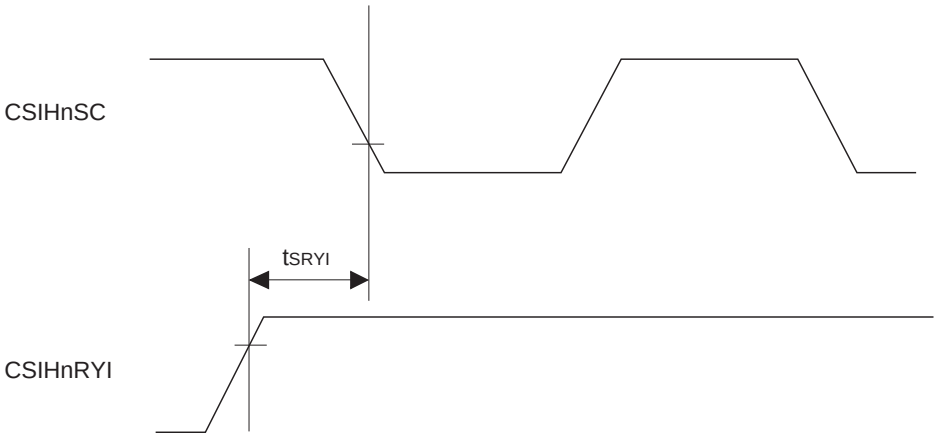


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

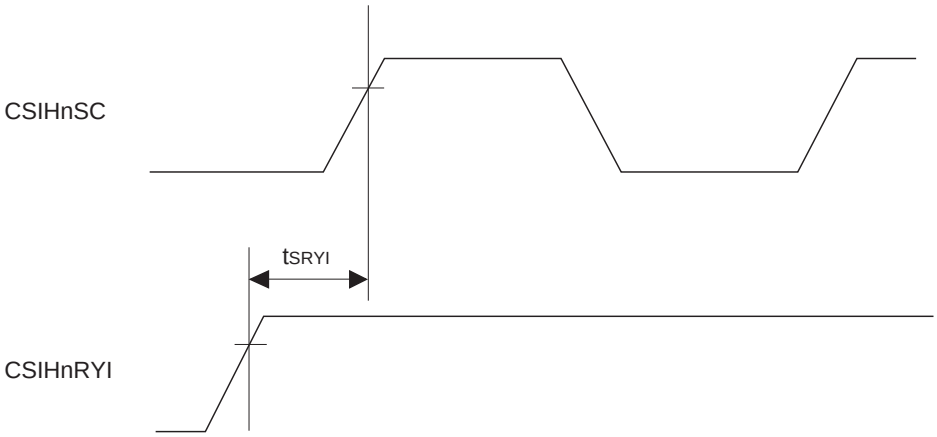
[CSIHnSC/CSIHnRYI]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

Master mode: (CSIHnCFGx: CSIHnCKPx = 0)



Master mode: (CSIHnCFGx: CSIHnCKPx = 1)

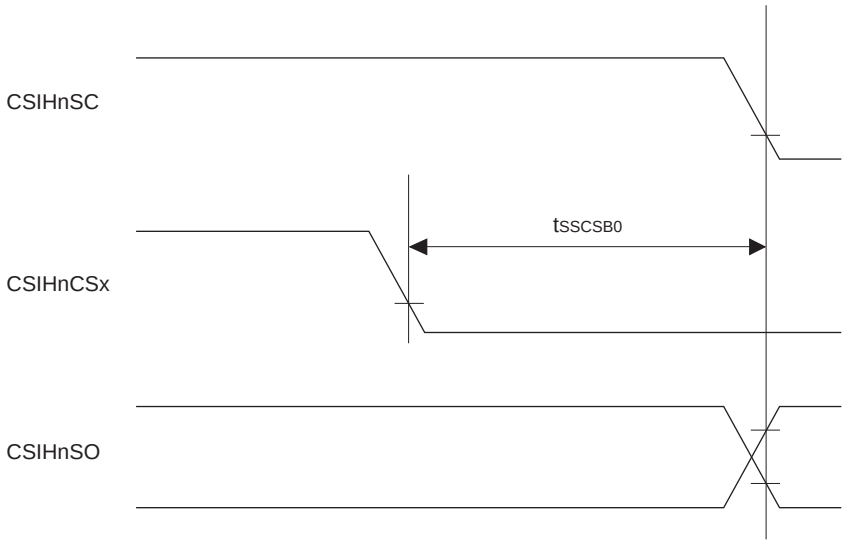


Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

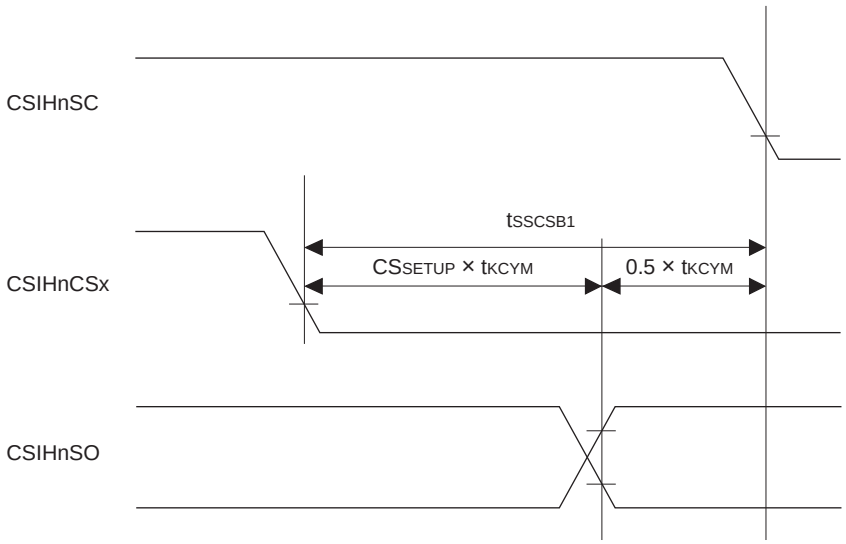
[CSIHnSC/CSIHnSO/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/1)



Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

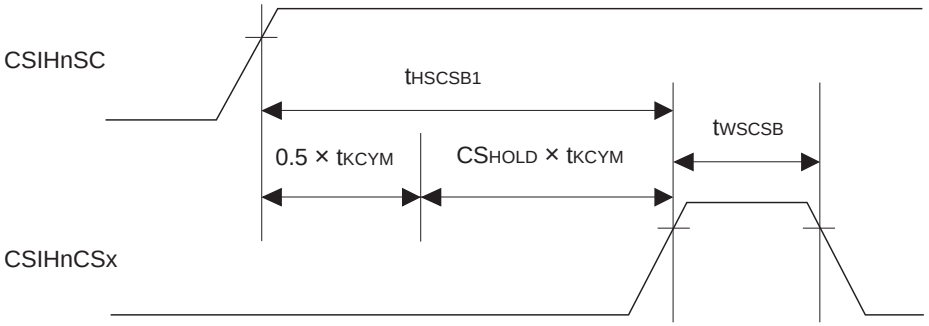
[CSIHnSC/CSIHnCSx]

Caution If you are not using the chip-select function, read the CSIHnCKPx bits as the CSIHnCTL1.CSIHnCKR bits.

(CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



(CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx/CSIHnDAPx = 0/0)



Note n = 0, 1, x = 0 to 7 (n = 0), x = 0 to 3 (n = 1)

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

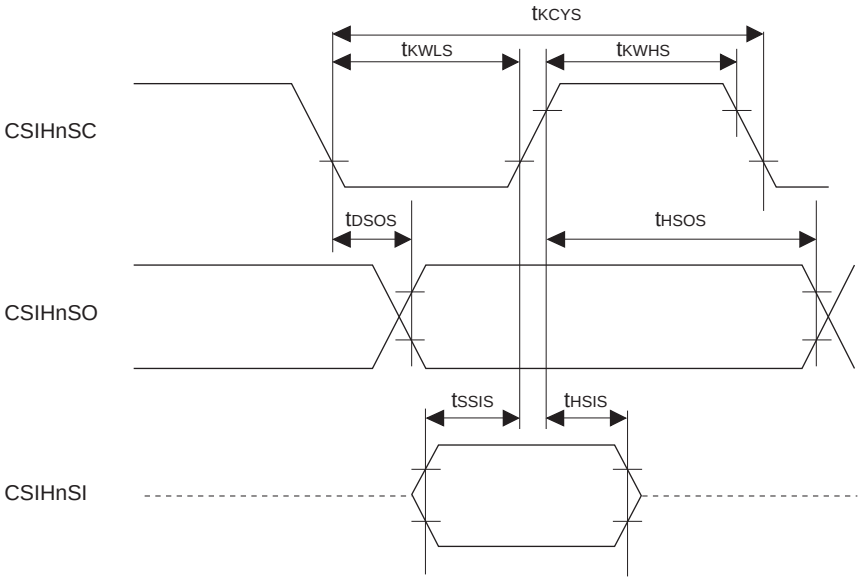
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIHnSC cycle time	t_{KCYS}	Input	PCLK = 80 MHz	150		ns
CSIHnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIHnSI setup time (to CSIHnSC)	t_{SSIS}			15		ns
CSIHnSI hold time (to CSIHnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIHnSO output delay time (to CSIHnSC)	t_{DSOS}				30	ns
CSIHnSO hold time (to CSIHnSC)	t_{HSOS}	CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 0/0 or 1/1		t_{KWHS}		ns
		CSIHnCTL1.CSIHnCKR/ CSIHnCFG0.CSIHnDAP0 = 1/0 or 0/1		t_{KWLS}		ns
CSIHnRYO output delay	t_{SRYO}				30	ns

Note n = 0, 1

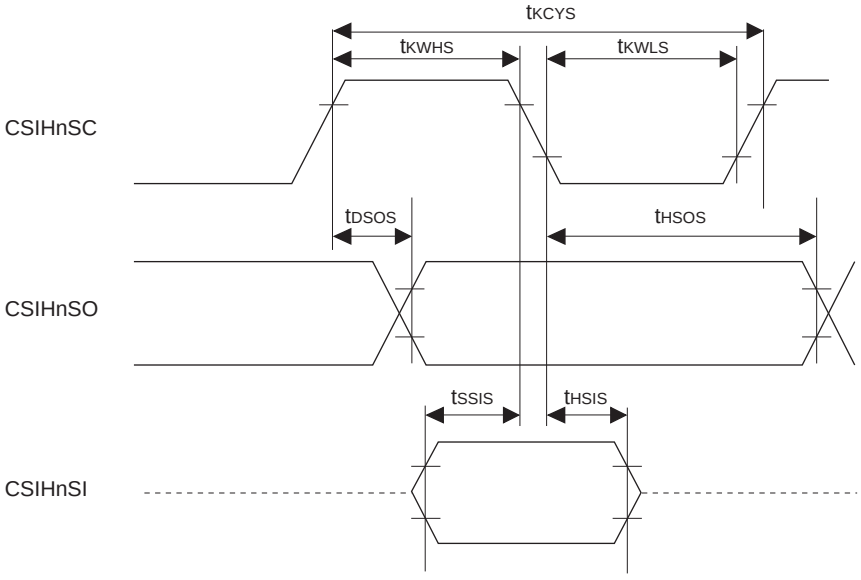
t_{PCLK}: PCLK cycle time

[CSIHnSC/CSIHnSI/CSIHnSO]

Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



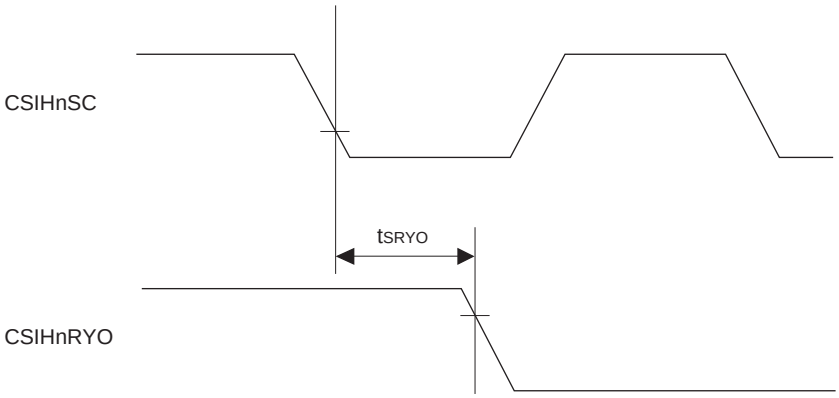
Slave mode: (CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 1/0 or 0/1)



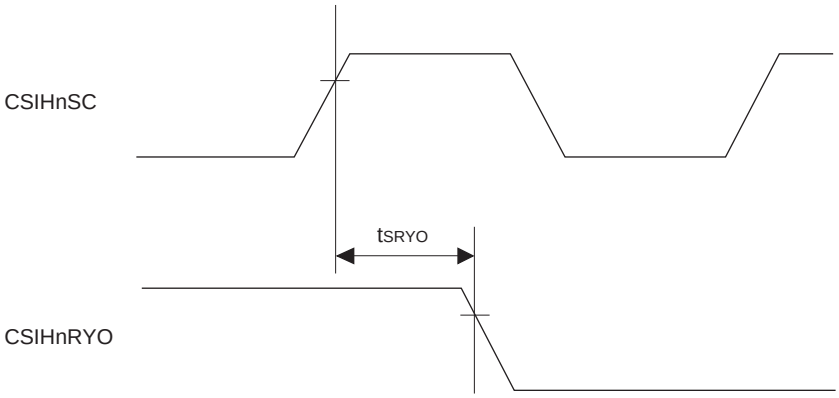
Note n = 0, 1

[CSIHnSC/CSIHnRYO]

(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/0 or 1/1)



(CSIHnCTL1: CSIHnCKR/CSIHnCFG0: CSIHnDAP0 = 0/1 or 1/0)



Note n = 0, 1

28.5.6.7 CSIGn Timing

(a) Master Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

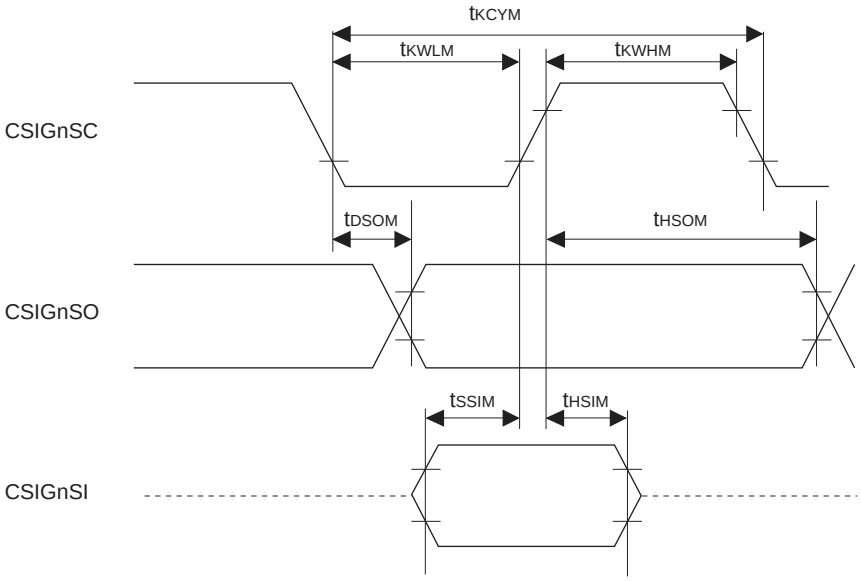
Item	Symbol	Conditions	MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYM}	Output	125		ns
CSIGnSC high/low level width	t_{KWHM}, t_{KWLM}	Output	(t_{KCYM}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIM}		30		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIM}		10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOM}			10	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOM}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1	t_{KWHM}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1	t_{KWLM}		ns
CSIGnRYI setup time (to CSIGnSC)	t_{SRYI}		($2 \times t_{PCLK}$) + 30		ns

Note n = 0 to 2

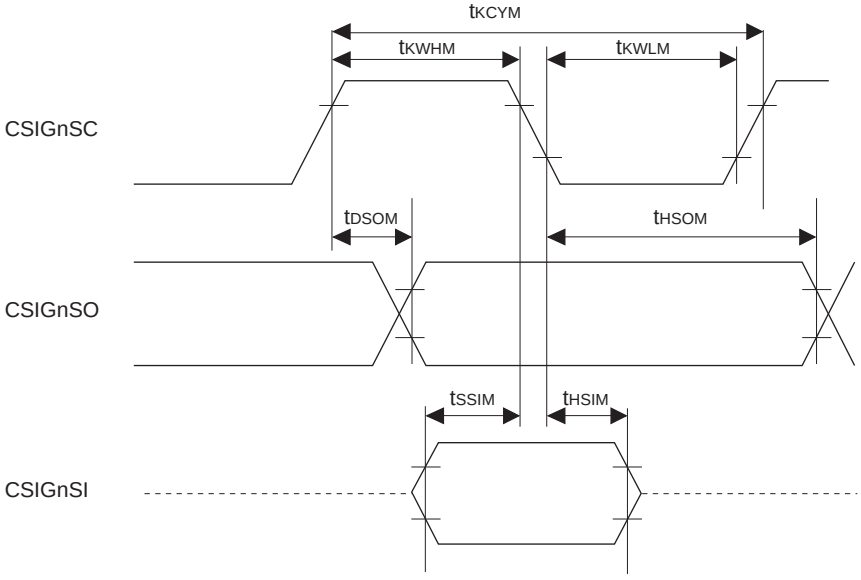
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



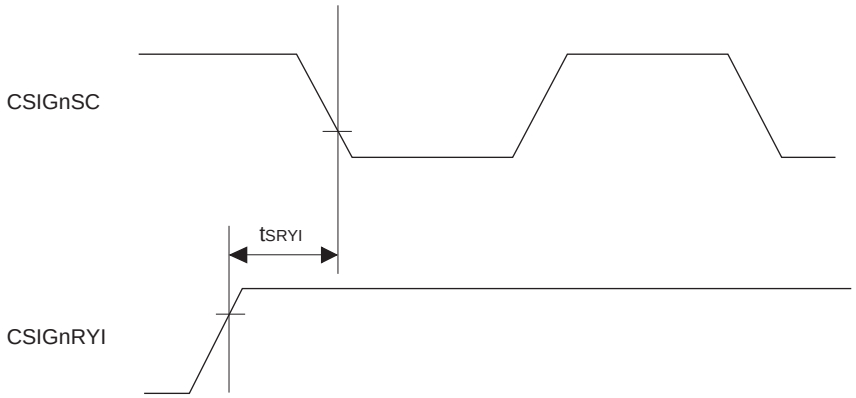
Master mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



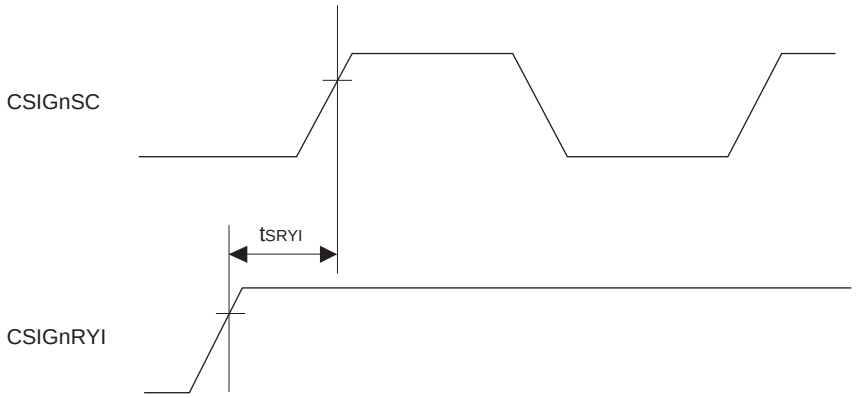
Note n = 0 to 2

[CSIGnSC/CSIGnRYI]

Master mode: (CSIGnCTL1: CSIGnCKR = 0)



Master mode: (CSIGnCTL1: CSIGnCKR = 1)



Note n = 0 to 2

(b) Slave Mode

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

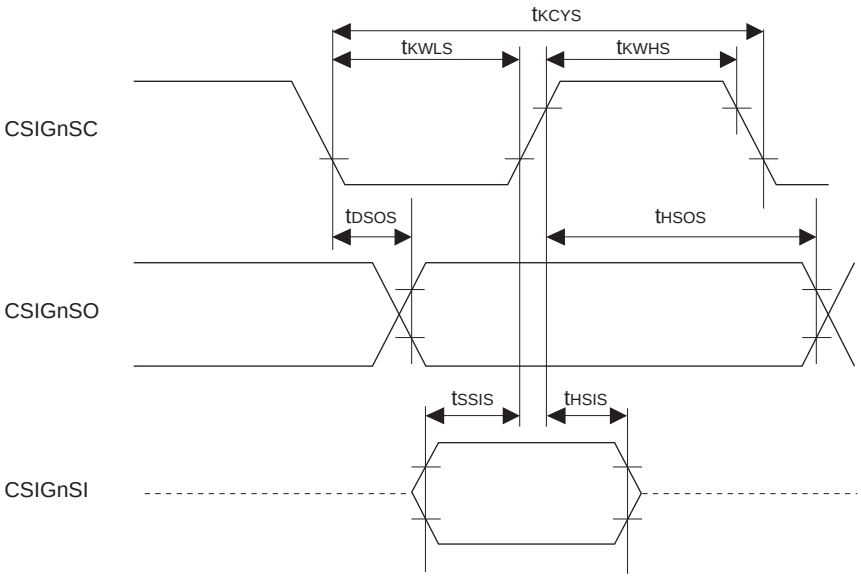
Item	Symbol	Conditions		MIN.	MAX.	Unit
CSIGnSC cycle time	t_{KCYS}	Input	PCLK = 80 MHz	150		ns
CSIGnSC high/low level width	t_{KWHS}, t_{KWLS}	Input		(t_{KCYS}/2) - 20		ns
CSIGnSI setup time (to CSIGnSC)	t_{SSIS}			15		ns
CSIGnSI hold time (to CSIGnSC)	t_{HSIS}			t_{PCLK} + 10		ns
CSIGnSO output delay time (to CSIGnSC)	t_{DSOS}				30	ns
CSIGnSO hold time (to CSIGnSC)	t_{HSOS}	CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 0/0 or 1/1		t_{KWHS}		ns
		CSIGnCTL1.CSIGnCKR/ CSIGnCFG0.CSIGnDAP = 1/0 or 0/1		t_{KWLS}		ns
CSIGnRYO output delay	t_{SRYO}				30	ns

Note n = 0 to 2

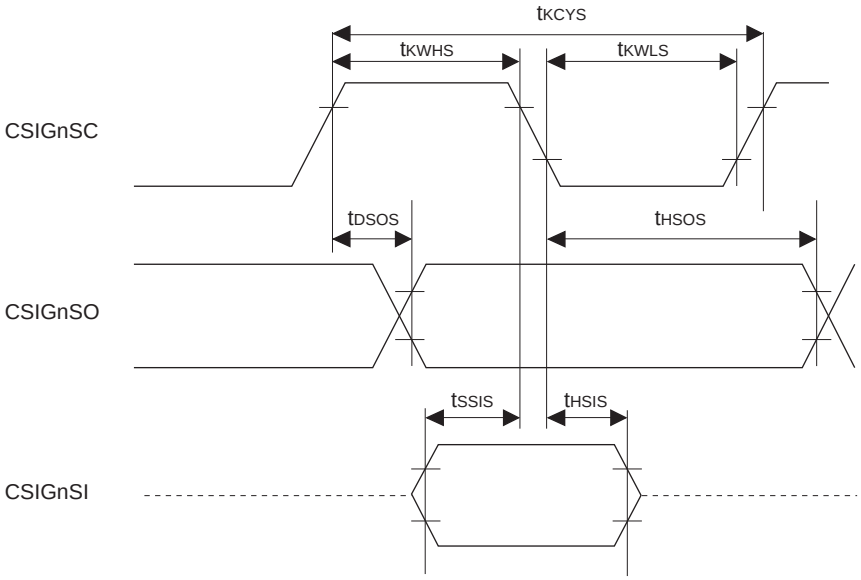
t_{PCLK}: PCLK cycle time

[CSIGnSC/CSIGnSI/CSIGnSO]

Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



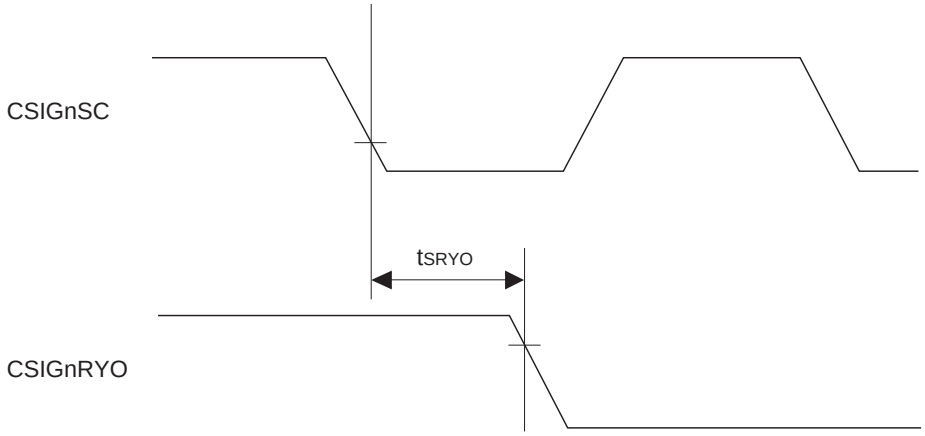
Slave mode: (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)



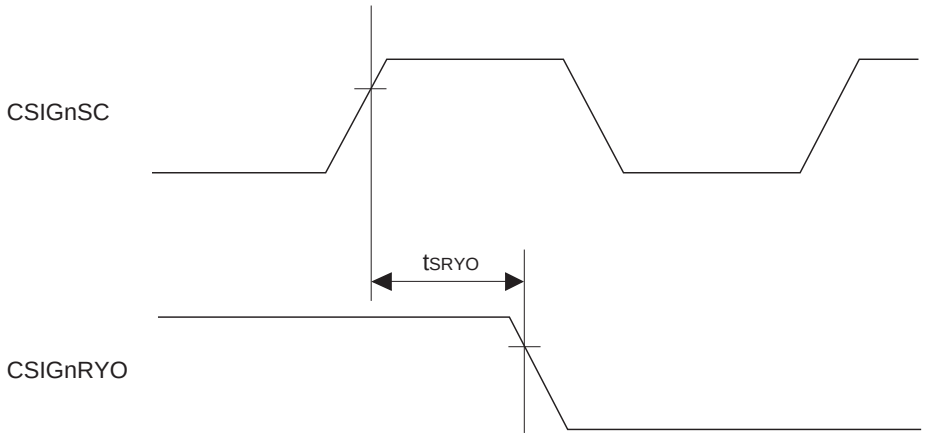
Note n = 0 to 2

[CSIGnSC/CSIGnRYO]

(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)



(CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1 or 1/0)



Note n = 0 to 2

28.5.6.8 UARTHn Timing

(a) Asynchronous Mode

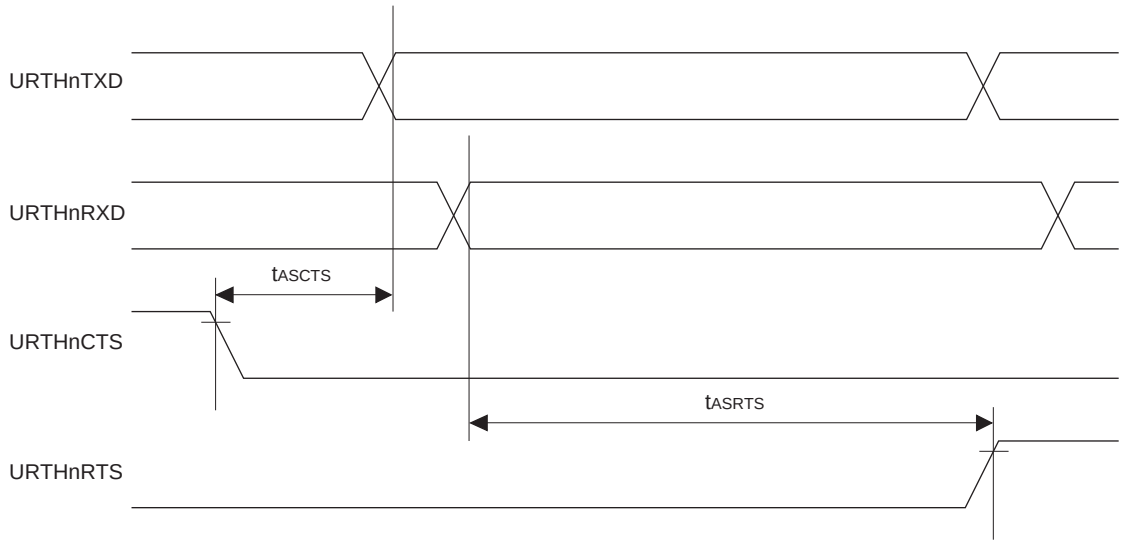
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		MIN.	MAX.	Unit
UARTHn transfer rate				16	2M	bps
URTHnCTS setup time (to URTHnTXD)	t_{ASCTS}	Input	URTHnCTS without noise filter	3T + 30		ns
			URTHnCTS with noise filter	5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{ASRTS}	Output	URTHnRXD without noise filter		(3BRS + 2) T + 30	ns
			URTHnRXD with noise filter		(3BRS + 4) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]



(b) Clock Synchronous, Master Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM1}	Output		500		500		ns
URTHnSC high level width	t_{KWHM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnSC low level width	t_{KWLM1}	Output		t_{KCYM1}/2 - 20		t_{KCYM1}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDM1}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDM1}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM1}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM1}	Output		t_{KCYM1}/2 - 10		t_{KCYM1}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM1}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM1}	Output			T + 30		T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

(c) Clock Synchronous, Slave Mode (Normal Mode)

(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	URTHnSC/URTHnRXD without Noise Filter		URTHnSC/URTHnRXD with Noise Filter		Unit
			MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input	500		500		ns
URTHnSC high level width	t_{KWHS1}	Input	4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input	4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input	10		10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input	2T + 10		2T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output		30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output	t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input, URTHnCTS without noise filter	T + t_{KCYS1} + 30		-T + t_{KCYS1} + 30		ns
		Input, URTHnCTS with noise filter	2T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output	2T + 30	3T + 30	2T + 30	3T + 30	ns

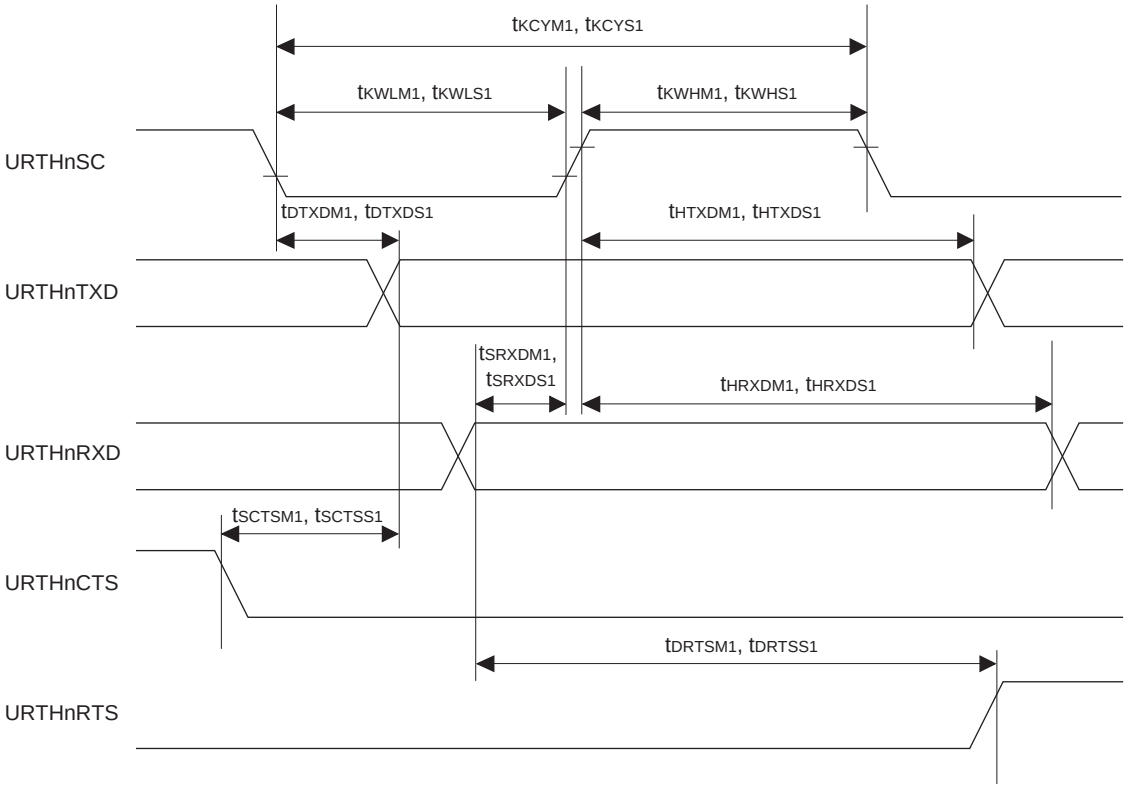
Note n = 0 to 2

T = prescaler cycle time

Item	Symbol	Conditions		URTHnSC without Noise Filter URTHnRXD with Noise Filter		URTHnSC with Noise Filter URTHnRXD without Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYS1}	Input		500		500		ns
URTHnSC high level width	t_{KWHS1}	Input		4T - 20		4T - 20		ns
URTHnSC low level width	t_{KWLS1}	Input		4T - 20		4T - 20		ns
URTHnRXD setup time (to URTHnSC ↑)	t_{SRXDS1}	Input		2T + 10		-2T + 10		ns
URTHnRXD hold time (to URTHnSC ↑)	t_{HRXDS1}	Input		10		4T + 10		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDS1}	Output			30		30	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDS1}	Output		t_{KWHS1} - 10		t_{KWHS1} - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSS1}	Input	URTHnCTS without noise filter	T + t_{KCYS1} + 30		T + t_{KCYS1} + 30		ns
			URTHnCTS with noise filter	3T + t_{KCYS1} + 30		3T + t_{KCYS1} + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSS1}	Output		4T + 30	5T + 30	30	T + 30	ns

Note n = 0 to 2

T = prescaler cycle time


tKCYM1, tKCYS1
tKWLM1, tKWLS1
tKWHM1, tKWHS1
URTHnSC
tDTXDM1, tDTXDS1
tHTXDM1, tHTXDS1
URTHnTXD
tSRXDM1,
tSRXDS1
tHRXDM1, tHRXDS1
URTHnRXD
tSCTSM1, tSCTSS1
URTHnCTS
tDRTSM1, tDRTSS1
URTHnRTS

Note n = 0 to 2

(d) Clock Synchronous, Master Mode (Delay Sampling Mode)

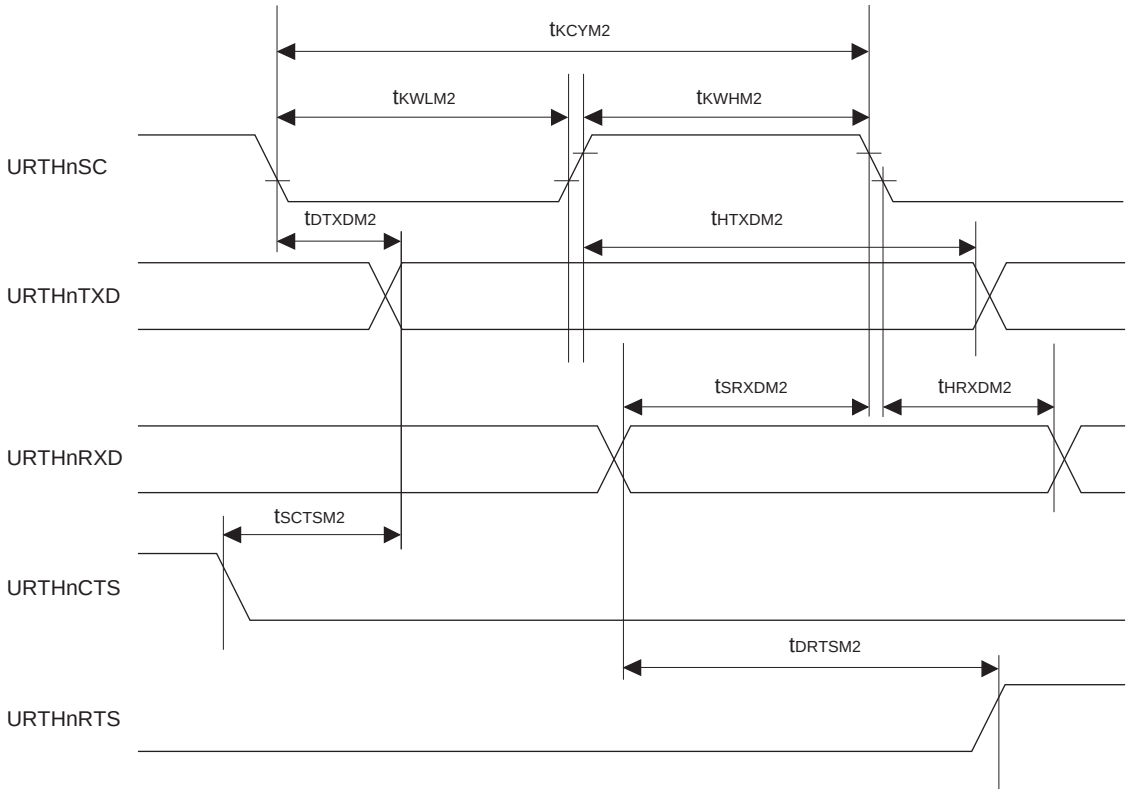
(T_a = -40 to +125°C, $V_{DDM} = V_{DDC} = CPUPLLV_{DD} = FRPLLV_{DD}$ = 1.1 to 1.3 V, $EV_{DD} = OSCV_{DD} = FV_{DD} = AV_{DD0}$ = 4.5 to 5.5 V, $V_{SSM} = V_{SSC} = CPUPLLV_{SS} = FRPLLV_{SS} = EV_{SS} = OSCV_{SS} = FV_{SS} = AV_{SS0}$ = 0V, C_L = 35 pF)

Item	Symbol	Conditions		URTHnRXD without Noise Filter		URTHnRXD with Noise Filter		Unit
				MIN.	MAX.	MIN.	MAX.	
URTHnSC cycle time	t_{KCYM2}	Output		500		500		ns
URTHnSC high level width	t_{KWHM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnSC low level width	t_{KWLM2}	Output		t_{KCYM2}/2 - 20		t_{KCYM2}/2 - 20		ns
URTHnRXD setup time (to URTHnSC ↓)	t_{SRXDM2}	Input		T + 30		3T + 30		ns
URTHnRXD hold time (to URTHnSC ↓)	t_{HRXDM2}	Input		0		-2T		ns
URTHnTXD output delay time (to URTHnSC ↓)	t_{DTXDM2}	Output			10		10	ns
URTHnTXD output hold time (to URTHnSC ↑)	t_{HTXDM2}	Output		t_{KCYM2}/2 - 10		t_{KCYM2}/2 - 10		ns
URTHnCTS setup time (to URTHnTXD)	t_{SCTSM2}	Input	URTHnCTS without noise filter	3T + 30		3T + 30		ns
			URTHnCTS with noise filter	5T + 30		5T + 30		ns
URTHnRTS output delay time (to URTHnRXD)	t_{DRTSM2}	Output			(BRS + 1) T + 30		(BRS + 1) T + 30	ns

Note n = 0 to 2

T = prescaler cycle time

BRS: Setting of URTHnCTL2.URTHnBRS[11:0]


tKCYM2
tKWLM2
tKWHM2
URTHnSC
tDTXDM2
tHTXDM2
URTHnTXD
tSRXDM2
tHRXDM2
URTHnRXD
tSCTSM2
URTHnCTS
tDRTSM2
URTHnRTS

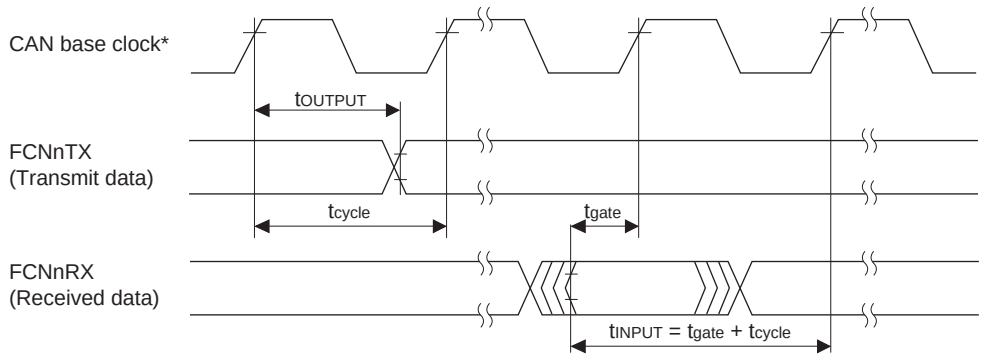
Note n = 0 to 2

28.5.6.9 CAN Timing

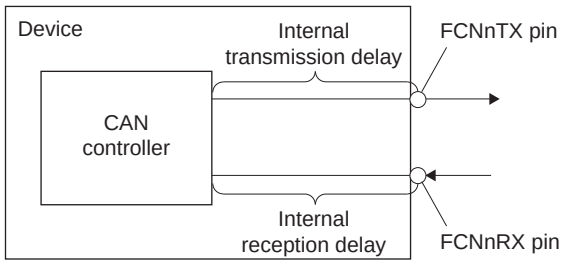
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
CAN transfer rate				1	Mbps
CAN internal delay time*[1]	t_{node}			87.5	ns

Note 1. CAN internal delay time(t_{node}) = t_{OUTPUT} (internal transmission delay) + t_{INPUT} (internal reception delay)



Note * CAN base clock frequency: f_{CAN}



Schematic view of the CAN-internal delay

Note n = 0, 1

28.5.6.10 FlexRay Timing

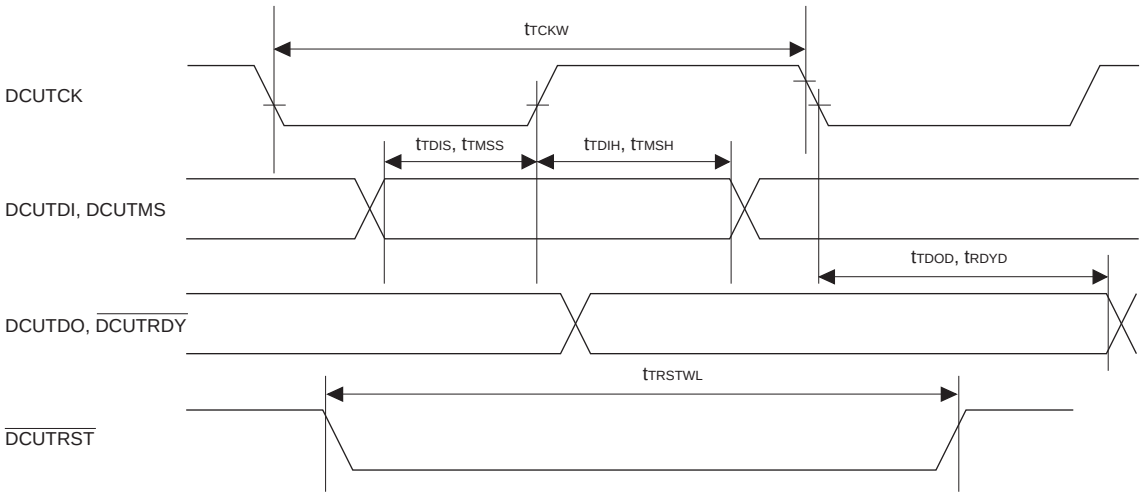
(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				10	Mbps

28.5.6.11 NEXUS I/F Timing

(T_a = 0 to +40°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	MIN.	MAX.	Unit
DCUTCK cycle time	t_{TCKW}	40.0		ns
DCUTDI setup time (to DCUTCK ↑)	t_{TDIS}	t_{TCKW}/2 - 8.5		ns
DCUTMS setup time (to DCUTCK ↑)	t_{TMSS}	t_{TCKW}/2 - 8.5		ns
DCUTDI hold time (to DCUTCK ↑)	t_{TDIH}	2.0		ns
DCUTMS hold time (to DCUTCK ↑)	t_{TMSH}	2.0		ns
DCUTDO output delay time (to DCUTCK ↓)	t_{TDOD}		20.0	ns
$\overline{\text{DCUTRDY}}$ output delay time (to DCUTCK ↓)	t_{RDYD}		20.0	ns
$\overline{\text{DCUTRST}}$ low level width	t_{TRSTWL}	1000		ns



28.5.6.12 A/D Converter Characteristics

(a) 10-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.0	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Zero-scale error*1	ZSE					±1.5	LSB
Full-scale error*1	FSE					±1.5	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} Conversion	1003		1023	LSB
			2/3AV_{DD0} Conversion	673	683	693	LSB
			1/2AV_{DD0} Conversion	504	512	520	LSB
			1/3AV_{DD0} Conversion	331	341	351	LSB
			AV_{SS0} Conversion	0		20	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(b) 10-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Power-down return time	t_{DPU}					1.5	µs
Accuracy*1	TOE					±2.5	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±2.0	LSB
Full-scale error*1	FSE					±2.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±2.0	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} Conversion	671	683	695	LSB
			1/2AV_{DD0} Conversion	502	512	522	LSB
			1/3AV_{DD0} Conversion	329	341	353	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(c) 12-Bit Resolution, without S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	µs
Accuracy*1	TOE					±6.0	LSB
Conversion time*2	t_{CONV}			1.0		10	µs
Zero-scale error*1	ZSE					±5.0	LSB
Full-scale error*1	FSE					±5.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			AV_{REF0M}		AV_{REF0P}	V
Differential non-linearity error*1	DNL					±1.5	LSB
Integral non-linearity error*1	INL					±2.5	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			3.5	9.5	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			4.9	11	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			4.6	10.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			6.0	12	mA
	A_{IDDPD}	ADCA0GPS=0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	AV_{DD0} Conversion	4015		4095	LSB
			2/3AV_{DD0} Conversion	2691	2731	2771	LSB
			1/2AV_{DD0} Conversion	2018	2048	2078	LSB
			1/3AV_{DD0} Conversion	1325	1365	1405	LSB
			AV_{SS0} Conversion	0		80	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

(d) 12-Bit Resolution, with S/H

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = AV_{REF0P} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = AV_{REF0M} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					12	bit
Power-down return time	t_{DPU}					1.0	µs
Accuracy*1	TOE					±9.0	LSB
Conversion time*2	t_{CONV}			1.5		10	µs
Channel S/H hold time	t_{HOLD}					50	µs
Zero-scale error*1	ZSE					±7.0	LSB
Full-scale error*1	FSE					±7.0	LSB
Sampling time	t_{AS}			6.5			clocks
Analog input voltage	VIAN			0.2		AV_{REF0P} - 0.2	V
Differential non-linearity error*1	DNL					±3.5	LSB
Integral non-linearity error*1	INL					±5.0	LSB
AV_{REF0P} input current	A_{IREF1}				0.6	1	mA
AV_{DD0} current	A_{IDD}	ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 0			13.6	22	mA
		ADCA0GPS = 1, ADCA0BPC = 0, ADCA0DIAG = 1			16.2	24	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 0			14.3	23.5	mA
		ADCA0GPS = 1, ADCA0BPC = 1, ADCA0DIAG = 1			17.4	25.5	mA
	A_{IDDPD}	ADCA0GPS = 0			0.8	16	µA
Conversion result		AV_{REF0P} = AV_{DD0} AV_{REF0M} = AV_{SS0} when self diagnosis is in use	2/3AV_{DD0} Conversion	2689	2731	2773	LSB
			1/2AV_{DD0} Conversion	2016	2048	2080	LSB
			1/3AV_{DD0} Conversion	1323	1365	1407	LSB

Note 1. Excluding quantization error.
Note 2. Be sure to set the related register within the range from minimum to maximum conversion time.

Caution 1. Implement the following countermeasures to minimize any adverse effect on precision of the A/D converter from the switching noise of the digital circuit.

i. Use one full side of the board for the power-supply wiring or connect the wiring in as large a grid pattern as possible.

ii. Insert a bypass capacitor in the vicinity of the pin lead between power supply pins (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, FV_{DD}, or AV_{DD0}) and ground pin (V_{SSM}, V_{SSC}, EV_{SS}, $OSCV_{SS}$, $CPUPLLV_{SS}$, $FRPLLV_{SS}$, FV_{SS}, or AV_{SS0}), and between the analog reference voltage pins (AV_{REF0P}, or AV_{REF0M}) and the analog ground pin (AV_{SS0}).

iii. We recommend isolating the analog power supply (AV_{DD0}) from the digital power supply (V_{DDM}, V_{DDC}, EV_{DD}, $OSCV_{DD}$, $CPUPLLV_{DD}$, $FRPLLV_{DD}$, or FV_{DD}) and supplying power via a series-connected regulator. When a common power supply is to be used, connect the analog power-supply line, digital power-supply line, and an electrolytic capacitor at a point in the source of the supply but wire the power supplies in isolation from each other on the board. Insert a chip inductor at the point of entry for the analog power supply voltage. In addition, earth the analog ground, digital ground, and an electrolytic capacitor at a point in the source of the power supply, but wire them in isolation from each other on the board.

Caution 2. Connect resistor Re and capacitor Ce to the analog input pins (ADCA0Im) externally to eliminate noise (m = 01 to 22).

Note Full-scale range (%FSR) is the ratio, as a percentage, of the extremes of the range of analog input voltages that can be converted, and is expressed as follows regardless of the resolution.

1%FSR = (Maximum value of convertible analog input voltage - Minimum value of convertible analog input voltage)/100

= (AV_{REF0P} - 0)/100

= AV_{REF0P}/100

1 LSB is as follows at a resolution of 10 bits:

1 LSB = $1/2^{10}$

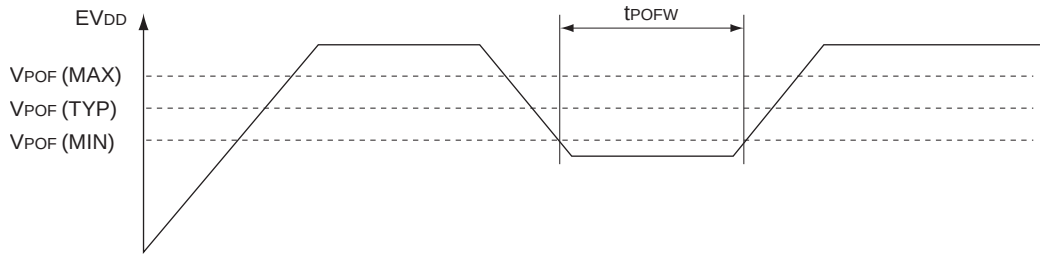
= 0.098%FSR

28.5.6.13 POF/LVI Characteristics

POF Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{POF}		2.7	2.8	2.9	V
Minimum detection width	t_{POFW}		0.2			ms

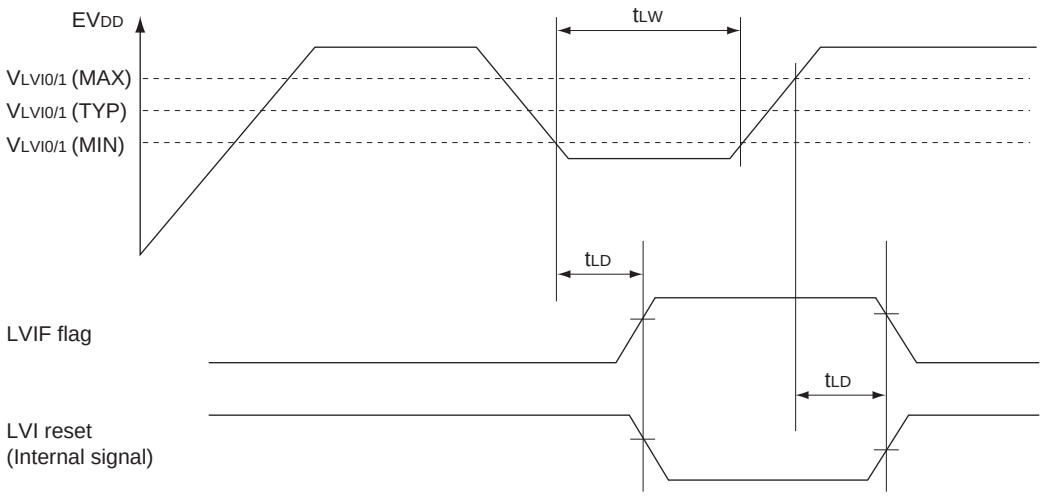


LVI characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detected voltage	V_{LVI0}	LVICNT.LVICNT[1:0] = 01_B	4.5	4.6	4.7	V
	V_{LVI1}	LVICNT.LVICNT[1:0] = 10_B	4.2	4.3	4.4	V
Response time*1	t_{LD}	After EV_{DD} reaches V_{LVI0}, V_{LVI1} (MAX.) After EV_{DD} falls below V_{LVI0}, V_{LVI1} (MIN.) Voltage gradient: 1.8 V/us to 1.8 V/s			2	ms
Minimum detection width	t_{LW}		2.0			ms
Stabilization waiting time*2					350	μs

Note 1. Time until output of an interrupt or reset after voltage detection.
Note 2. Time until LVI is detected in accord with the specification after enabling of the LVI function.


EVDD
tLW
VLVI0/1 (MAX)
VLVI0/1 (TYP)
VLVI0/1 (MIN)
tLD
LVIF flag
tLD
LVI reset
(Internal signal)

28.5.6.14 Flash Memory Programming Characteristics

(a) Basic Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}			80		160	MHz
Number of times rewritable (per block)*1	C_{ERWR}	Code Flash	Data are retained for 20 years			100	Times/ block
		Data Flash	Data are retained for 5 years			15000	Times/ block
			Data are retained for 15 years			5000	Times/ block
			Data are retained for 20 years			1000	Times/ block

Note 1. In writing to a delivered product, either "erasing → writing" or "writing alone" is considered as a "time" of rewriting.

Example: (P: writing, E: erasing)

Delivered product→ P→E→P→E→P: 3 times

Delivered product→ E→P→E→P→E→P: 3 times

28.5.6.15 FLMD0 Pulse Timing Characteristics

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V, C_L = 35 pF)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 setup time (to VDD)	t_{DP}		1			ms
$\overline{\text{RESET}}$ release (to FLMD0)	t_{PR}		2			ms
$\overline{\text{RESET}}$↑ → FLMD0 set time	t_{RP}		1200			μs
FLMD0 high/low level width	t_{PW}		10		100	μs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns
Serial clock input frequency	t_{SCCY}		9.8		5000	KHz
High level width	t_{WHSC}		85		50000	ns
Low level width	t_{WLSC}		85		50000	ns
SI setup time (to SCK ↑)	t_{PLHSC}		75			ns
SI hold time (to SCK ↑)	t_{PHLSC}		0			ns
SO output delay time	t_{DSC}				23	ns
UART transfer rate			9.6		1000	Kbps

Note VDD: V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$

28.5.6.16 BIST Execution Time

(T_a = -40 to +125°C, V_{DDM} = V_{DDC} = $CPUPLLV_{DD}$ = $FRPLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = $OSCV_{DD}$ = FV_{DD} = AV_{DD0} = 4.5 to 5.5 V, V_{SSM} = V_{SSC} = $CPUPLLV_{SS}$ = $FRPLLV_{SS}$ = EV_{SS} = $OSCV_{SS}$ = FV_{SS} = AV_{SS0} = 0 V)

Item	Symbol	Condition 1	Condition2	MIN.	TYP.	MAX.	Unit
BIST execution time		All HW-BIST Total fxx = 80 MHz				40	ms
		All HW-BIST Total fxx = 160 MHz				20	ms

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.50	Jan. 27, 2012	—	First Edition issued
0.51	Feb. 22, 2012	p.30	2.2.1 Terms; corrected
		p.36	2.2.3 Pin Data Input/Output; PBDCn.PBDCn_m, corrected
		p.37	2.2.3 Pin Data Input/Output; Pn register write, corrected
		p.39	Table 2-7 Port Group Configuration Registers; Register Name, <PORTn_base>, corrected
		p.49	Table 2-18 Pn Register Contents; Note, corrected
		p.53	2.3.4 Configuration of Electrical Characteristics Registers; register name of (2) PDSCn, corrected
			Table 2-22 PDSCn Register Contents; corrected
		p.54	2.3.4 Configuration of Electrical Characteristics Registers; bit 10 in (3) PODCn bit chart, corrected
		p.56	2.3.5 Port Register Protection; (1) PPCMDn, corrected
		p.63	2.4.1 List of Ports and Pins; (2), title of Table 2-28, corrected
		p.67	Table 2-28 List of Pins Other than Port Pins; functions of ENCA1I0 and ENCA1I1, corrected
		p.77	Table 2-30 Handling of Unused Pins Other than Port Pins; Handling in DCUTCK to DCUEVTO and ERROROUT, corrected
		p.86	Table 2-32 List of States of Pins Other than Port Pins; Note 3, corrected
		p.96	Table 2-34 List of Pull-Up and Pull-Down Resistors for Pins Other than Port Pins; FLMD1, $\overline{\text{RESET}}$, $\overline{\text{DCUTRST}}$ in Pull-Up Resistor and Pull Down Resistor, corrected
		p.115	2.15.1 Details of Noise Cancellation; (1) Digital Filter Function, corrected
		p.121	2.15.1 Details of Noise Cancellation; (3), title of Table 2-53, corrected
		p.126	Table 2-56 The List of Registers for the Digital Noise Canceller; Register Function, corrected
		p.127	Table 2-57 Contents of Register DNFAnCTL; Function of bits DNFAnNFSTS[1:0], corrected
		p.133	2.16 Edge Detection; corrected
1.00	Jul. 25, 2012	Throughout	Addition of μPD70F3506 and μPD70F3508
			Addition of $\overline{\text{RESETOUT}}$ signal
			Deletion of Boundary scan mode
			Correction of $\overline{\text{ERROROUT}}$ to ERROROUT
			Correction of signal names about TAPA
			Section 1 Introduction
		p.2	Addition of Minimum instruction execution times
		p.23	Modification of 1.7.1 Internal Block Diagram
			Section 2 Port Functions
		p.38	Addition of 2.2.4 Logic Diagram of Port Control
		p.41	Addition of Table 2-8 JTAG Port Group Configuration Registers
		p.44	Modification of Caution 1 in 2.3.2 (3) PIPCn – Port IP Control Register
		p.55	Modification of Table 2-23 in 2.3.4 (2) PDSCn – Port Drive Strength Control Register
		p.57	Modification of Table 2-25 in 2.3.4 (4) PISn – Port Input Buffer Selection Register
		p.59	Modification of Figure 2-2 Sequence for Writing to a Protected Port Register
		p.65, 68 to 71, 73	Modification of Table 2-29 List of Pins Other than Port Pins
		pp.74 to 82	Addition of I/O circuit type in 2.4.1 (3) Handling of Unused Port Pins and (4) Handling of Unused Pins Other than Port Pins

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012	p.90	Addition of JP0 in Table 2-32 List of States of Port Pins
		p.91	Modification of Table 2-33 List of States of Pins Other than Port Pins
		p.100	Addition of JP0 in Table 2-34 List of Pull-Up and Pull-Down Resistors for Port Pins
		p.101	Modification of Table 2-35 List of Pull-Up and Pull-Down Resistors for Pins Other than Port Pins
		p.111	Modification of Table 2-43 List of Port 3 Control Registers
		p.117	Modification of Table 2-49 List of Port 6 Control Registers
		p.121	Addition of 2.15 JTAG Port 0
		p.139	Modification of Table 2-62 DNFSCTL Register Contents
			Section 3 CPU System Function
		pp.144 to 147	Modifocation of Table 3-2 PPU Protected Areas and Modules
		p.148	Addition of 3.1.2 (2) Branching Instructions and Loading Instructions
		p.159	Modification of Table 3-7 DCLKWAIT Register
			Section 4 Interrupt Functions
		p.165	Addition of Note 4 to Table 4-1 List of Exception Sources
		p.174	Addition of Note 6 to Table 4-2 List of Interrupt Sources
		p.184	Correction of ICFLX0I2L and ICSW2L address in Table 4-3 (9/9)
		pp.185 to 194	Modification of bit names in 4.3.2 IMRm
		p.203	Addition of Caution 2 in 4.3.11 INTSTR0B
		p.207, 209, 212	Correction of description in Figure 4-1 Processing upon Occurrence of FENMI Interrupt Request, Figure 4-2 Processing upon Occurrence of FEINT Interrupt Request and Figure 4-4 Processing upon Occurrence of EIINT Interrupt Request
		p.214, 221	Modification of bit name in 4.5.1 Mask Function of EI Level Maskable Interrupt (EIINT) and 4.5.4 Pending Interrupt Report Function
			Section 5 DMA/DTS Module
		p.232	Correction of description in 5.3.8 Arbitration of Transfer Requests
		p.278	Correction of register name in Table 5-8 List of DTS Control Registers
		p.299	Deletion of description in 5.11.1 Transfer information (TI)
		p.302	Correction of Figure 5-13 DTS Transfer Setting Flow
		p.303	Correction of Figure 5-14 Example of TI Fetch and Write Back
		p.308, 310	Correction of description in 5.11.5 Transfer Types
		p.323, 324	Correction of description in 5.11.6 Special Functions
		p.326	Correction of description in 5.11.8 Chain Function
		p.330	Correction of Table 5-11 Conditions for Skipping Write-back
		p.331	Correction of description in 5.11.14 Valid DTS Transfer Request Conditions
		p.340	Deletion of Caution in 5.15.1 DTSSELn
			Section 6 Memory Modules
		p.347	Correction of description in 6.5 Option-Setting Bytes
		p.349	Addition of 6.5.1(2) μPD70F3506
			Section 7 Clock Generation
		p.361	Modification of 7.4 Clock Generating Circuit
		p.372	Correction of equation in 7.6.4 (1) (a) Calculating the Thresholds of CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]
		p.375	Modification of Table 7-16 Examples of CLMAnCMPH and CLMAnCMPL Register Settings
		pp.377 to 381	Correction of description in 7.6.5 Clock Monitor A Registers
		p.378	Addition of Caution in 7.6.5 (2) CLMAnCTL1 - CLMAn Control Register 1

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012		**Section 8 Reset Controller**
		p.383	Addition of Note 5 to Table 8-1 Reset Sources and Timing of Operations
		p.384	Addition of Caution in 8.1 (1) Internal Reset Signals
		p.389	Modification of 8.2.6 Clock Monitor Reset
		p.395	Addition of Cautions in 8.3.2 (1) RESF
		pp.395 to 397	Modification of Table 8-4RESFC Register Contents and Table 8-5 RESFS Register Contents
		p.398	Modification of description in 8.3.3 Details of Software Reset Control Registers, and addition of Caution in 8.3.3 (1) SWRESA - Software Reset Register
		pp.401 to 403	Modification of Figure 8-5 Example of POF Operations to Figure 8-7 Example of LVI Operation (when the LVICNT.LVIMD Bit = 0)
		p.405, 408	Modification of 8.4.4 (1) POF and (5) LVISF
			Section 9 Safety Functions
		p.412	Modification of description in 9.3.2 (2) APES
		p.417	Correction of register names in Table 9-7 List of Registers Related to Self-Diagnostic BIST
		p.420	Addition of Caution in 9.5.2 (3) BSEQ0STRHBT - Self-Diagnostic BIST Status Register
		p.422	Modification of Table 9-12 Contents of the BSEQ0STCHBT Register
		p.438	Addition of Caution in 9.7.1 On-chip RAM
			Section 10 Safety Guardian (SGA)
		p.445	Correction of Table 10-5 SGA Error Input Signals (3/3) and Table 10-6 SGA Input Signals
		p.451	Addition of Note in (1) Dynamic Mode Enable
		p.466	Addition of Caution in (9) SGAICFG1
		p.470	Addition of Caution in (13) SGAEMK1
			Section 12 Window Watchdog Timer A (WDTA)
		p.495	Addition of description in 12.4.2 (1) WDTA Enable Register (WDTAnWDTE)
			Section 13 Timer Array Unit A (TAUA)
		p.501	Correction of register name in Figure 13-2 TAUA Block Diagram
		p.512	Addition of Note for Table 13-8 Simultaneous Rewrite and Trigger Timing
		p.513	Correction of Figure 13-4 General Procedure for Simultaneous Rewrite
		p.551, 555	Addition of description in 13.13.3 and 13.13.4 (1) Overview - Prerequisites
		p.560	Modification of description in 13.13.5 (1) Overview - Functionaldescription
		p.561	Correction of equation in 13.13.5 (2) Equations
		p.564, 571, 578, 630	Addition of description in Table 13-34 Operating Procedure for TAUAnTTINm Input Pulse Interval Measurement Function, Table 13-39 Operating Procedure for TAUAnTTINm Input Signal Width MeasurementFunction, Table 13-43 Operating Procedure for TAUAnTTINm Input Period Count Detection Function and Table 13-87 Operating Procedure for TAUAnTTINm Input Position Detection Function
		p.575	Correction of equation in 13.13.7 (2) Equations
		p.580, 584	Modification of description in 13.13.8 and 13.13.9 (1) Overview - Functionaldescription
		p.605	Modification of description in 13.13.12 (3) Block Diagram and General Timing Diagram
		p.616, 619	Correction of signal name in Figure 13-61 General Timing Diagram of External Event Count Function and Figure 13-62 TAUAnCDRm = 0000H, TAUAnCMURm.TAUAnTIS[1:0] = 01B
		p.626	Correction of signal name in 13.13.15 (6) (a) TAUAnCDRm = 0000H
		p.627	Correction of equation in 13.13.16 (2) Equations

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012	p.636	Correction of description in 13.14.1 (4) (d) Simultaneous Rewrite for Master Channels - Note
		p.638	Correction of title Table 13-93 Control Bit Settings in Synchronous Channel Output Mode 1
		p.644	Correction of description in 13.14.2 (1) Note 2
		p.659	Correction of description in 13.14.4 (1) Overview - Prerequisites
		p.666	Correction of title Table 13-116 Control Bit Settings in Independent Channel Output Mode 2
		p.667	Modification of description in Table 13-118 Operating Procedure for One-Shot Pulse Output Function
		p.675	Modification of description inTable 13-126 Operating Procedure for Offset Trigger Output Function
		p.677	Correction of equation in 13.14.6 (2) Equations
		pp.685 to 686	Addition of description in 13.14.6 (7) Specific Timing Diagrams
		p.689	Correction of equation in 13.14.7 (2) Equations
		p.692	Correction of value in Figure 13-90 General Timing Diagram of Triangle PWM Output Function with Dead Time
		pp.700 to 701	Addition of description in 13.14.7 (8) Specific Timing Diagrams
		p.706	Correction of description in Figure 13-95 Block Diagram of Non-Complementary Modulation Output Function Type 1
		p.726	Modification of description inTable 13-170 TAUAnCMORm Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2
		p.757	Correction of Table 13-200 Overview of TAUAn Registers (1/2)
		p.768	Correction of bit7, 6 in Table 13-206 Description of TAUAnCMORm Register (2/3)
		p.772, 773, 784	Modification of description inTable 13-209 Description of TAUAnCSCm Register, Table 13-210 Description of TAUAnTS Register, Table 13-212 Description of TAUAnTT Register, Table 13-229 Description of TAUAnRDT Register and Table 13-230 Description of TAUAnRSF Register
		p.785	Correction of description in 13.15.7 (1) TAUAnDASi - TAUAn DMA Window Address Setting Register i (i = 0 to 7)
			Section 14 Timer Array Unit J (TAUJ)
		p.791	Correction of register name in Figure 14-2 Block Diagram of TAUJ
		p.801	Correction of Figure 14-4 General Procedure for Simultaneous Rewrite
		p.802	Modification of description in14.7.2 (1) Initial Settings
		p.827	Correction of bit name in Table 14-20 Effects of an Overflow Modification of description in17.12.3 (1) Overview - Functional description
		p.832, 840, 847	Addition of description in Table 14-24 Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function, Table 14-29 Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function and Table 14-33 Operating Procedure for TAUJnTTINm Input Period Count Detection Function
		p.836	Correction of bit name in Table 14-25 Effects of Overflow
		p.844, 849	Correction of equation in 14.12.5, 14.12.6 (2) Equations
		p.861	Correction of title Table 14-43 Control Bit Settings in Synchronous Channel Output Mode 1
		p.871, 872	Correction of description in 14.14.3 (1) TAUJnCDRm and (2) TAUJnCNTm - Access
		p.880, 881, 886	Modification of Table 14-55 Description of TAUJnCSCm Register, Table 14-56 Description of TAUJnTS Register, Table 14-58 Description of TAUJnTT Register and Table 14-66 Description of TAUJnRDT Register
		p.883, 885	Modification of Table 14-61 Description of TAUJnTOC Register and Table 14-65 Description of TAUJnRDM Register

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012		**Section 15 TSG2 (TSG20, TSG21)**
		p.887	Modification of Table 15-3 TSG2 Clock Supply
		pp.895 to 896	Modification of Table 15-8 TSnCTL1 Register Contents
		p.908	Correction of bit 4 in Table 15-15 TSnIOC1 Register Contents (1/2)
		p.910	Addition of Note in 15.4.2 (10) TSG2n I/O Control Register 2 (TSnIOC2)
		p.912	Correction of description in 15.4.2 (12) TSG2n Status Register 0 (TSnSTR0) - Access
		p.913	Correction of bit 3 in Table 15-19 TSnSTR1 Register Contents
		p.947	Correction of Table 15-38 Compare Register Functions in Each Mode (2/4)
		p.956	Correction of description in 15.5.4 (1) Timer Output in Each Mode
		p.974, 977	Correction of signal name in Figure 15-22 Example of Operation immediately after TSnTE Flag in TSnSTR0 is Set to 1, Figure 15-25 Example of Pattern Difference Detection Flag Operation and 15.7.10 Caution 2
		p.989, 990	Correction of description in Figure 15-32 When TSnPIE = 1, TSnVIE = 1, and TSnRCC04 to TSnRCC00 = 00B in TSnCTL4, and TSnACC01 and TSnACC00 = 00B in TSnCTL5 (HT-PWM Mode) and Figure 15-33 When TSnPIE = 0, TSnVIE = 1, and TSnRCC04 to TSnRCC00 = 02B in TSnCTL4 and TSnACC01 and TSnACC00 = 00B in TSnCTL5 (HT-PWM Mode)
		p.1034	Correction of description in Figure 15-59 Basic Operation Flow in SP-PWM Mode
		p.1039	Correction of equation in 15.11.3 (3) Various Settings of SP-PWM Mode - Carrier period
		p.1046	Correction of signal name in Table 15-73 Timer Input Function in 120-DC Mode
		p.1049	Correction of equation in 15.11.4 (2) Various Settings of 120-DC Mode - Carrier period
			Section 16 TPBA (TPBA0 and TPBA1)
		p.1091, 1094	Correction of description in 16.4.2 (8) TPBAn Period Setting Register (TPBAnCMP0) and (11) TPBAn Timer Counter Register (TPBAnCNT0)
		p.1099	Correction of signal name in 16.5.2 Compare Register Rewrite Operation - Caution
		p.1100	Correction of Figure 16-3 Simultaneous Rewrite Timing (TPBAnDPS = 0, TPBAnRDM = 0, and TPBAnTOL = 0)
		p.1105	Correction of description in Table 16-23 Timer Output Function
		p.1106	Correction of signal name in Table 16-25 Compare Match Timing
		p.1108	Correction of description in Figure 16-5 Example of Basic Timing (2/2) - Note 2
			Section 17 OS Timer (OSTM)
		p.1116	Correction of description in 17.3.4 Starting and Stopping the Timer - Starting the timer
		p.1124	Correction of description in Table 17-8 List of OS Timer Registers
		p.1133	Correction of bit 15 in Table 17-19 IC0CKSEL1 Register Contents
			Section 18 Encoder Timer (ENCA)
		p.1146	Correction of bit 0 in Table 18-10 ENCAnFLG Register Contents
		p.1152, 1153	Modification of Table 18-16 ENCAnTS Register Contents and Table 18-17 ENCAnTT Register Contents
		p.1169	Correction of description in 18.4.4 (2) Capture Function - 1
		p.1171	Correction of signal name in 18.4.5 (2) Capture Function
		p.1174	Correction of description in 18.4.6 (4) Stopping the Timers in a Dual Encoder-Timer Configuration
		p.1178	Correction of description in 18.5.1 (4) Settings for the ENCAnCCR1 Register
		p.1182, 1183	Correction of description in 18.6 (4) Timing of Basic Encoder Operation 4 (Encoder Capture Mode) and (5) Timing of Basic Encoder Operation 5 (Encoder Capture and Comparison Mode)

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012		**Section 19 Timer Option Module (TAPA)**
		p.1199	Correction of description in 19.3.2 (1) TAPAnCTL0 - TAPAn Control Register 0
		p.1201	Correction of description in 19.3.2 (3) TAPAnFLG - TAPAn Flag Register - Access, Initial value
		pp.1212 to 1215	Correction of description in 19.4.3 Selection of INT Signal Output
		p.1214	Correction of description in Figure 19-3 Example of Operations in Response to Master Channel 0 of TAUA
		p.1216, 1217, 1221	Correction of description in 19.4.4 Selecting a Trigger to Start Conversion by the A/D Converter
			Section 20 CAN Controller (FCN)
		p.1233	Correction of Table 20-14 FCN Message Buffer Register Bit Configuration (1/2)
		p.1265	Correction of description in 20.5.3 (1) FCNnMmDATxB/H/W - FCNn Message Data Byte Registers - Access
		p.1278	Correction of Figure 20-6 Reception Timing
		p.1281	Correction of 20.7.4 (3) Mask setting for FCN module 1 (mask 1) (example)
		p.1312	Correction of Figure 20-13 Initialization
			Section 21 Clocked Serial Interface G (CSIG)
		p.1343	Addition of Port Groups in 21.1 CSIG Features
		p.1357, 1358	Correction of signal name in 21.3.9 Handshake Function
		p.1365	Addition of Caution to Table 21-9 CSIGnCTL0 Register Contents
		p.1372	Correction of description in21.4 (6) CSIGnBCTL0 - CSIG Receive-Only Mode Control Register 0 - Address
			Section 22 Queue-Equipped Three-Line Serial Interface H (CSIHn)
		p.1383	Addition of Port Groups in 22.1 CSIHn Features
		p.1391	Correction of Figure 22-6 Chip Select and RCB Example
		p.1408	Correction of Table 22-11 CSIHnTIR Interrupt Generation
		p.1410	Modification of description in 22.3.10 (4) CSIHnTIJC (Job Completion Interrupt)
		p.1438	Correction of Table 22-24 CSIHnMCTL1 Register Contents
		p.1449	Correction of signal name in 22.4.1 (12) CSIHnTX0H — CSIHn Transmission Register 0 for Half Word Access - Caution 3
			Section 23 Synchronous/Asynchronous Serial Interface H (UARTH)
		p.1472	Addition of Table 23-6 Port Groups for UARTHn
		p.1479	Correction of bit 3 in Table 23-9 URTHnCTL1 Register Contents (2/3)
		pp.1484 to 1485	Correction of description in 23.4 (5) URTHnSTR0 – UARTHn Status Register 0 and (6) URTHnSTR1 – UARTHn Status Register 1 - Initial value
		p.1485, 1486	Correction of description in Table 23-13 URTHnSTR1 Register Contents
		p.1490	Correction of bit 3 in Table 23-15 URTHnOPT0 Register Contents (3/3)
		pp.1493 to 1496	Correction of description in 23.4 (11) URTHnRX - UARTHn Receive Data Register, Table 23-18 URTHnRX Register Contents, Table 23-19 URTHnRXHL Register Contents, Table 23-20 URTHnERX Register Contents and Table 23-21 URTHnERXW Register Contents
		p.1497	Addition of Table 23-22 Access in the Various Modes of Reception
		pp.1498 to 1501	Correction of description in 23.4 (15) URTHnTX – UARTHn Transmit Data Register, Table 23-23 URTHnTX Register Contents, 23.4 (16) URTHnTXHL – UARTHn Transmit Data Register HL, Table 23-24 URTHnTXHL Register Contents, 23.4 (17) URTHnETX – UARTHn Enhanced Bit Transmit Data Register, Table 23-25 URTHnETX Register Contents, 23.4 (18) URTHnETXW – UARTHn Enhanced Bit Transmit Word Data Register and Table 23-26 URTHnETXW Register Contents

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012	p.1502	Addition of Table 23-27 Access in the Various Modes of Transmission
		p.1510	Correction of signal name in 23.6.3 Handshake Mode
		p,1514, 1515, 1528	Correction of bit name in 23.6.5 Enhanced Bit Detection/ID Compare-Match Detection and 23.6.15 Digital Receive Data Noise Filter
			Section 24 FlexRay™ (FLX)
		p.1545	Correction of FLXnSUCC1 in Table 24-12 FlexRay Controller Register Map (2/4)
		p.1555	Correction of description in 24.4.3 (1) FLXnTEST1 – Test Register 1 - Loop back mode
		p.1560, 1564	Modification of description in 24.4.4 (1) FLXnEIR – Error Interrupt Register and (2) FLXnSIR – Status Interrupt Register
		p.1582	Correction of description in 24.4.5 (1) FLXnSUCC1 - SUC Configuration Register 1 - Initial value
		p.1594	Correction of description in 24.4.5 (7) FLXnMHDC - MHD Configuration Register - Initial value
		p.1607	Correction of bit 13 in Table 24-52 FLXnCCSV Register Contents (2/3)
		p.1621	Correction of description in 24.4.6 (10) FLXnESIDm - Even Sync ID Register
		p.1622	Correction of description in 24.4.6 (11) FLXnOSIDm – Odd Sync ID Register
		p.1654	Correction of title in Table 24-87 FLXnRDDSm Register Contents
		p.1655	Correction of description in 24.4.11 (2) FLXnRDHS1 - Read Header Section Register 1
		p.1667, 1668	Correction of description in 24.5.1 (3) Symbol Window, (4) Network Idle Time (NIT) and (5) Configuration of NIT Start and Offset Correction Start
		p.1669	Correction of description in 24.5.3 (1) Global Time and (2) Local Time
		p.1681	Correction of description in 24.5.5 (6) WAKEUP State - Wakeup pattern (WUP)
		p.1684	Correction of description in 24.5.5 (7) STARTUP State - Startup timeouts
		p.1705, 1707	Correction of description in 24.5.11 (2) Host Access to Message RAM - Data transfer from message RAM to output buffer
		p.1724	Correction of description in 24.7.3 Register FLXnRCV Displays Wrong Value - Effects
		pp.1730 to 1732	Addition of 24.7.12 The interval of cycles after startup commences is shortened by the READY command during startup and normal operation to 24.7.15 Initialization of the slot mode by the HALT or FREEZE command is immediately executed
			Section 25 A/D Converter
		p.1734, 1735	Correction of description in Table 25-2 Units to which Hardware Trigger Signals are Connected
		p.1739, 1740	Modification of description in 25.3.1 Basic Operation
		p.1776	Modification of description in 25.3.15 Stabilization Control
		p.1777	Correction of description in Table 25-7 List of ADCAn Registers (1/2) - A/D conversion result registers
		p.1779	Correction of bit 7 in Table 25-8 ADCAnCTL0 Register Contents, and addition of caution in 25.4.2 (1) ADCAnCTL0 – A/D Converter Mode Control Register 0
		p.1782	Addition of Caution in 25.4.2 (2) ADCAnCTL1 - A/D Converter Mode Control Register 1
		p.1785	Modification of Note in 25.4.2 (5) ADCAnCNT – A/D Converter Stabilization Counter
		p.1788, 1790 to 1792	Modification of Table 25-15 ADCAnSTC1 Register Contents, and Table 25-17 ADCAnSTC2 Register Contents to Table 25-19 ADCAnTRG4+i Register Contents
		p.1798	Addition of Note 1 in 25.4.5 (3) ADCAnDBiCR - CGi DMA buffer register
		p.1799	Modification of Table 25-23 ADCAnDBiCRL Register Contents
		p.1809	Addition of (e) in 25.5.4 (1) Analog Input Pin (ADCAnIm)
		p.1813, 1814	Description of reference values in Figure 25-23 Example of Measures against Noise on Analog Input Circuit and Note for Figure 25-24 Internal Equivalent Circuit of ADCAnIm Pin

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012		**Section 26 Peripheral Interconnection (PIC)**
		p.1844	Correction of description in 26.4.3.2 Configuration
		p.1856	Correction of description in (2)
		p.1857, 1858	Correction of description in Figure 26-12 Setting Flow (Active High)
		p.1867	Correction of title in Table 26-16 PIC0REG2n0 Contents
		p.1906	Correction of description in Figure 26-28 Setting Flow (Active High)
		p.1908	Correction of description in Figure 26-30 Setting Flow (Active High) (cont.)
		p.1917	Correction of description in Figure 26-31 Block Diagram
		p.1923	Correction of description in Table 26-28 Combinations of Interrupt Trigger Signals and Measurement Timers
		p.1927	Correction of Table 26-30 TAUJ0 Setting
		p.1931, 1932	Correction of description in Table 26-33 PIC0REG30 Contents
		p.1932, 1933	Correction of description in 26.4.7.4 Example of Operation
		p.1945, 1946	Correction of description in Figure 26-42 Operation Example of Control Method 1 at Up Count and Figure 26-43 Operation Example of Control Method 1 at Down Count
		p.1954	Correction of description in Figure 26-48 Block Diagram of Timer Configuration 2
		p.1957	Correction of bit 0 in Table 26-44 PIC0REG30 Contents
		p.1962, 1963, 1965	Correction of description in Figure 26-50 Operation Example of Control Method 2: at Advance to Figure 26-52 Operation Example of Control Method 2: at Down Count and Figure 26-54 ENCAnCCR1 Rewrite Flow during Operation
		p.1971	Correction of description in Figure 26-57 Block Diagram of Timer Configuration 1
		p.1981	Correction of description in Figure 26-59 Example of Operation
		p.1983	Correction of description in Figure 26-60 Initial Setting and Operation Starting Flow
		p.1987	Correction of description in Table 26-61 PIC Setting
		p.1988	Correction of Table 26-62 TAUAn Setting (Master CHm)
		p.1990, 1991	Correction of description in Figure 26-63 Block Diagram of Timer Configuration 1 and Figure 26-64 Block Diagram of Timer Configuration 2
		p.1996, 1997	Correction of description in Figure 26-65 Operation Example of Three-Phase Encoder Function and Figure 26-66 Example of Normal Rotation/Reverse Rotation using Three-Phase Encoder Function
		p.1998	Correction of Figure 26-67 Initial Setting and Operation Start Flow
		p.2002	Correction of description in Table 26-69 PIC Setting
			Section 27 On-chip Debugging Unit (OCD)
		p.2007	Correction of description in 27.1 (12) Hot Plug-in
		p.2009	Correction of description in 27.3 Notes on On-Chip Debugging - 5
		p.2011	Addition of 11 and 12 in 27.3 Notes on On-Chip Debugging
			Section 28 Electrical Characteristics
		p.2013	Correction of condition in 28.1.2 Capacity
		p.2014	Correction of circuit in Caution 1
		p.2015	Correction of description in DC Characteristics (1/3) - VIH4
		p.2017, 2018	Correction of DC Characteristics (3/3)
		p.2023	Correction of 28.1.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing
		p.2025, 2030, 2033, 2036	Correction of 28.1.6.6 CSIHn Timing and 28.1.6.7 CSIGn Timing
		pp.2040-2042, 2044	Modification of 28.1.6.8 (b) Clock Synchronous, Master Mode (Normal Mode), (c) Clock Synchronous, Slave Mode (Normal Mode) and (d) Clock Synchronous, Master Mode (Delay Sampling Mode)

REVISION HISTORY	V850E2/Px4 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 25, 2012	p.2046	Modification of Note 1 in 28.1.6.9 CAN Timing
		pp.2048 to 2051	Correction of 28.1.6.11 A/D Converter Characteristics
		p.2053	Correction of 28.1.6.12 POF/LVI Characteristics
		p.2104	Correction of condition in 28.3.2 Capacity
		p.2105	Correction of circuit in Caution 1
		p.2108	Correction of DC Characteristics (3/3)
		p.2114	Correction of 28.3.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing
		p.2116, 2121, 2124, 2127	Correction of 28.3.6.6 CSIHn Timing and 28.3.6.7 CSIGn Timing
		pp.2131 to 2133, 2135	Modification of 28.3.6.8 (b) Clock Synchronous, Master Mode (Normal Mode), (c) Clock Synchronous, Slave Mode (Normal Mode) and (d) Clock Synchronous, Master Mode (Delay Sampling Mode)
		p.2137	Modification of Note 1 in 28.3.6.9 CAN Timing
		pp.2140 to 2143	Correction of 28.3.6.12 A/D Converter Characteristics
		p.2145	Correction of 28.3.6.13 POF/LVI Characteristics
		p.2194	Correction of title in 28.5 Electrical Characteristics of the μPD70F3509 (Target)
		p.2195	Correction of condition in 28.5.2 Capacity
		p.2196	Correction of circuit in Caution 1
		p.2199	Correction of DC Characteristics (3/3)
		p.2204	Correction of 28.5.6.4 Reset / Interrupt / A/D Converter Trigger / ESO Timing
		p.2206, 2211, 2214, 2217	Correction of 28.5.6.6 CSIHn Timing and 28.5.6.7 CSIGn Timing
		pp.2221 to 2223, 2225	Modification of 28.5.6.8 (b) Clock Synchronous, Master Mode (Normal Mode), (c) Clock Synchronous, Slave Mode (Normal Mode) and (d) Clock Synchronous, Master Mode (Delay Sampling Mode)
		p.2227	Modification of Note 1 in 28.5.6.9 CAN Timing
		pp.2230 to 2233	Correction of 28.5.6.12 A/D Converter Characteristics
		p.2235	Correction of 28.5.6.13 POF/LVI Characteristics

V850E2/Px4 User’s Manual: Hardware

Publication Date: Rev.1.00 Jul 25, 2012

Published by: Renesas Electronics Corporation



SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2012 Renesas Electronics Corporation. All rights reserved.

Colophon 1.1

V850E2/Px4

Renesas Electronics Corporation

R01UH0098EJ0100