

FM3
32-BIT MICROCONTROLLER
MB9Axxxx / MB9Bxxx Series
PERIPHERAL MANUAL

For the information for microcontroller supports, see the following web site.

<http://edevice.fujitsu.com/micom/en-support/>

FUJITSU SEMICONDUCTOR LIMITED

Preface

Thank you for your continued use of Fujitsu semiconductor products.
Read this manual and "Data Sheet" thoroughly before using products in this series.

■ Purpose of this manual and intended readers

This manual explains the functions and operations of this series and describes how it is used. The manual is intended for engineers engaged in the actual development of products using this series.

- * This manual explains the architecture and operation of the peripheral modules, but does not cover the specifics of each device in the series.
It is not intended to replace the device data sheets, but complement them.
Users should refer to the respective data sheets of devices for device-specific details.

■ Trademark

ARM and Cortex-M3 are the trademarks of ARM Limited in the EU and other countries.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Sample programs and development environment

Fujitsu Semiconductor offers sample programs free of charge for using the peripheral functions of the FM3 family. Fujitsu Semiconductor also makes available descriptions of the development environment required for this series. Feel free to use them to verify the operational specifications and usage of this Fujitsu Semiconductor microcontroller.

- Microcontroller support information:
<http://edevice.fujitsu.com/micom/en-support/>

- * : Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.
Fujitsu Semiconductor assumes no responsibility for any damage that may occur as a result of using a sample program.

- The contents of this document are subject to change without notice.
Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU SEMICONDUCTOR device; FUJITSU SEMICONDUCTOR does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU SEMICONDUCTOR assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU SEMICONDUCTOR or any third party or does FUJITSU SEMICONDUCTOR warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU SEMICONDUCTOR assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
Please note that FUJITSU SEMICONDUCTOR will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Related Manuals

The manuals related to this series are listed below. See the manual appropriate to the applicable conditions.

The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

■ Peripheral Manual

- **FM3 FAMILY MB9Axxx / MB9Bxxx SERIES PERIPHERAL MANUAL (this manual)**

■ Data sheet

For details about device-specific, electrical characteristics, package dimensions, ordering information etc., see the following document.

- **MICROCONTROLLER 32-bit ORIGINAL FM3 FAMILY DATA SHEET**

- * The data sheets for each series are provided.
See the appropriate data sheet for the series that you are using.

■ CPU Programming manual

For details about ARM Cortex-M3 core, see the following documents that can be obtained from <http://www.arm.com/>.

- **Cortex-M3 Technical Reference Manual**
- **ARMv7-M Architecture Application Level Reference Manual**

■ Flash Programming manual

For details about the functions and operations of the built-in flash memory, see the following document.

- **FM3 FAMILY FLASH PROGRAMMING MANUAL**

- * This manual is provided for each series.
See the appropriate manual for the series that you are using.

How to Use This Manual

■ Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- Search from the table of the contents
The table of the contents lists the manual contents in the order of description.
- Search from the register
The address where each register is located is not described in the text. To verify the address of a register, see "Register Map" in Appendixes.

■ About the chapters

Basically, this manual explains 1 peripheral function per chapter.

■ Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

■ Notations

- The notations in bit configuration of the register explanation of this manual are written as follows.
 - bit : bit number
 - Field : bit field name
 - Attribute : Attributes for read and write of each bit
 - R : Read only
 - W : Write only
 - R/W : Readable/Writable
 - - : Undefined
 - Initial value : Initial value of the register after reset
 - 0 : Initial value is "0"
 - 1 : Initial value is "1"
 - X : Initial value is undefined
- The multiple bits are written as follows in this manual.
Example : bit7:0 indicates the bits from bit7 to bit0
- The values such as for addresses are written as follows in this manual.
 - Hexadecimal number : "0x" is attached in the beginning of a value as a prefix (example : 0xFFFF)
 - Binary number : "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
 - Decimal number : Written using numbers only (example : 1000)

CONTENTS

CHAPTER 1: System Overview	1
1. Bus Architecture	2
1.1. Bus Block Diagram.....	4
2. Memory Architecture	5
2.1. Memory Map	6
2.2. Peripheral Address Map.....	7
3. Cortex-M3 Architecture.....	9
4. Mode	11
4.1. How to Set Operating Mode.....	11
4.2. Startup Sequence.....	12
CHAPTER 2-1: Clock	13
1. Clock Generation Unit Overview	14
2. Clock Generation Unit Configuration/Block Diagram.....	15
3. Clock Generation Unit Operations.....	18
3.1. Selecting the clock mode	18
3.2. Internal bus clock frequency division control	19
3.3. PLL clock control.....	20
3.4. Oscillation stabilization wait time	22
3.5. Interrupt causes	23
4. Clock Setup Procedure Examples.....	24
5. Clock Generation Unit Register List	26
5.1. System Clock Mode Control Register (SCM_CTL).....	27
5.2. System Clock Mode Status Register (SCM_STR).....	29
5.3. Base Clock Prescaler Register (BSC_PSR).....	31
5.4. APB0 Prescaler Register (APBC0_PSR).....	32
5.5. APB1 Prescaler Register (APBC1_PSR).....	33
5.6. APB2 Prescaler Register (APBC2_PSR).....	34
5.7. Software Watchdog Clock Prescaler Register (SWC_PSR).....	35
5.8. Trace Clock Prescaler Register (TTC_PSR)	36
5.9. Clock Stabilization Wait Time Register (CSW_TMR).....	37
5.10. PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)	39
5.11. PLL Control Register 1 (PLL_CTL1)	40
5.12. PLL Control Register 2 (PLL_CTL2)	41
5.13. Debug Break Watchdog Timer Control Register (DBWDT_CTL)	42
5.14. Interrupt Enable Register (INT_ENR)	43
5.15. Interrupt Status Register (INT_STR).....	44
5.16. Interrupt Clear Register (INT_CLR)	45
6. Clock Generation Unit Usage Precautions	47

CHAPTER 2-2: High-Speed CR Trimming	49
1. High-Speed CR Trimming Function Overview	50
2. High-Speed CR Trimming Function Configuration and Block Diagram.....	51
3. High-Speed CR Trimming Function Operation	52
4. High-Speed CR Trimming Function Setup Procedure Example	53
5. High-Speed CR Trimming Function Register List	59
5.1. High-speed CR oscillation Frequency Division Setup Register (MCR_PSR).....	60
5.2. High-speed CR oscillation Frequency Trimming Register (MCR_FTRM)	61
5.3. High-Speed CR Oscillator Register Write-Protect Register (MCR_RLR)	62
6. High-Speed CR Trimming Function Usage Precautions	63
CHAPTER 3: Resets	65
1. Overview	66
2. Configuration	67
3. Explanation of Operations	68
3.1. Reset Causes.....	69
3.2. Resetting Inside the Device	72
3.2.1. Resets to Cortex-M3	73
3.2.2. Resets to Peripheral Circuit	74
3.3. Reset Sequence.....	75
3.4. Operations After Resets are Cleared	77
4. Registers	78
4.1. Reset Cause Register (RST_STR: ReSeT STatus Register)	79
CHAPTER 4: Low-voltage Detection.....	83
1. Overview	84
2. Configuration	85
3. Explanation of Operations	87
4. Setup Procedure Examples.....	90
5. Registers	91
5.1. Low-voltage Detection Voltage Control Register (LVD_CTL)	92
5.2. Low-voltage Detection Interrupt Register (LVD_STR)	94
5.3. Low-voltage Detection Interrupt Clear Register (LVD_CLR)	95
5.4. Low-voltage Detection Voltage Protection Register (LVD_RLR)	96
5.5. Low-voltage Detection Circuit Status Register (LVD_STR2)	97
CHAPTER 5: Low Power Consumption Mode	99
1. Overview of Low Power Consumption Mode.....	100
2. Configuration of CPU Operation Modes	105
3. Operations of Standby Modes	110
3.1. Operations of SLEEP modes (high speed CR sleep, main sleep, PLL sleep, low speed CR sleep, and sub sleep modes)	113
3.2. Operations of TIMER modes (high speed CR timer, main timer, PLL timer, low speed CR timer, and sub timer modes).....	115
3.3. Operations of STOP mode.....	117
4. Standby Mode Setting Procedure Examples	120
5. List of Low Power Consumption Registers	122
5.1. Standby Mode Control Register (STB_CTL).....	123

CHAPTER 6: Interrupts	125
1. Overview	126
2. Structure	127
3. Exception and Interrupt Vectors	128
4. Registers	131
4.1. DMA Request Selection Register (DRQSEL)	133
4.2. EXC02 Batch Read Register (EXC02MON)	136
4.3. IRQ00 Batch Read Register (IRQ00MON)	137
4.4. IRQ01 Batch Read Register (IRQ01MON)	138
4.5. IRQ02 Batch Read Register (IRQ02MON)	139
4.6. IRQ03 Batch Read Register (IRQ03MON)	140
4.7. IRQ04/05 Batch Read Register (IRQxxMON)	141
4.8. IRQ06 Batch Read Register (IRQ06MON)	142
4.9. IRQ07/09/11/13/15/17/19/21 Batch Read Register (IRQxxMON)	144
4.10. IRQ08/10/12/14/16/18/20/22 Batch Read Register (IRQxxMON)	145
4.11. IRQ23 Batch Read Register (IRQ23MON)	146
4.12. IRQ24 Batch Read Register (IRQ24MON)	147
4.13. IRQ25/26/27 Batch Read Register (IRQxxMON)	149
4.14. IRQ28 Batch Read Register (IRQ28MON)	150
4.15. IRQ29 Batch Read Register (IRQ29MON)	152
4.16. IRQ30 Batch Read Register (IRQ30MON)	153
4.17. IRQ31 Batch Read Register (IRQ31MON)	155
4.18. IRQ32/33 Batch Read Register (IRQxxMON)	157
4.19. IRQ34 Batch Read Register (IRQ34MON)	158
4.20. IRQ35 Batch Read Register (IRQ35MON)	159
4.21. IRQ36/37 Batch Read Register (IRQxxMON)	160
4.22. IRQ38/39/40/41/42/43/44/45 Batch Read Register (IRQxxMON)	161
4.23. IRQ46/47 Batch Read Register (IRQxxMON)	162
5. Usage Warnings	163
CHAPTER 7: External Interrupt and NMI Control Sections.....	165
1. Overview	166
2. Block Diagram	167
3. Operations and Setting Procedure Examples.....	168
3.1. Operations of external interrupt control section	169
3.2. Operations of NMI control section.....	171
3.3. Returning from timer or stop mode	172
4. Registers	174
4.1. Enable Interrupt Request Register [ENIR]	175
4.2. External Interrupt Request Register [EIRR]	176
4.3. External Interrupt Clear Register [EICL]	177
4.4. External Interrupt Level Register [ELVR]	178
4.5. Non Maskable Interrupt Request Register [NMIRR]	179
4.6. Non Maskable Interrupt Clear Register [NMICL]	180

CHAPTER 8: DMAC.....	181
1. Overview of DMAC	182
2. Configuration of DMAC.....	183
2.1. DMAC and System Configuration	184
2.2. I/O Signals of DMAC	186
3. Functions and Operations of DMAC	188
3.1. Software-Block Transfer.....	189
3.2. Software-Burst Transfer	191
3.3. Hardware-Demand Transfer	192
3.4. Hardware-Block Transfer & Burst Transfer	193
3.5. Channel Priority Control	195
4. DMAC Control	196
4.1. Overview of DMAC Control.....	197
4.2. DMAC Operation and Control Procedure for Software Transfer	198
4.3. DMAC Operation and Control Procedure for Hardware (EM=0) Transfer	205
4.4. DMAC Operation and Control Procedure for Hardware (EM=1) Transfer	214
5. Registers of DMAC.....	218
5.1. List of Registers	219
5.2. Entire DMAC Configuration Register (DMACR)	220
5.3. Configuration A Register (DMACA).....	222
5.4. Configuration B Register (DMACB)	225
5.5. Transfer Source Address Register (DMACSA)	229
5.6. Transfer Destination Address Register (DMACDA)	230
5.7. Notes on Register Setting	231
CHAPTER 9: I/O PORT	233
1. Overview	234
2. Configuration, Block Diagram, and Operation	235
3. Setup Procedure Example.....	241
4. Register List	242
4.1. Port Function Setting Register (PFRx).....	244
4.2. Pull-up Setting Register (PCR _x)	245
4.3. Port input/output Direction Setting Register (DDR _x).....	246
4.4. Port Input Data Register (PDIR _x).....	247
4.5. Port Output Data Register x (PDOR _x)	248
4.6. Analog Input Setting Register (ADE).....	249
4.7. Extended Pin Function Setting Register (EPFR _x)	250
4.8. Extended Pin Function Setting Register 00 (EPFR00)	251
4.9. Extended Pin Function Setting Register 01 (EPFR01)	253
4.10. Extended Pin Function Setting Register 02 (EPFR02).....	257
4.11. Extended Pin Function Setting Register 04 (EPFR04)	261
4.12. Extended Pin Function Setting Register 05 (EPFR05)	265
4.13. Extended Pin Function Setting Register 06 (EPFR06)	269
4.14. Extended Pin Function Setting Register 07 (EPFR07)	273
4.15. Extended Pin Function Setting Register 08 (EPFR08)	277
4.16. Extended Pin Function Setting Register 09 (EPFR09)	282
4.17. Extended Pin Function Setting Register 10 (EPFR10)	286
4.18. Special Port Setting Register (SPSR)	292
5. Usage Precautions	294

CHAPTER 10: Clock supervisor.....	297
1. Overview	298
2. Configurations and Block Diagrams	299
3. Explanation of Operations	301
4. Setup Procedure Examples.....	302
5. Operation Examples	304
6. Register list.....	308
6.1. CSV control register (CSV_CTL)	309
6.2. CSV status register (CSV_STR)	311
6.3. Frequency detection window setting register (Upper) (FCSWH_CTL).....	312
6.4. Frequency detection window setting register (Lower) (FCSWL_CTL)	313
6.5. Frequency detection counter register (FCSWD_CTL).....	314
7. Usage Precautions	315
CHAPTER 11: Watchdog timer	317
1. Overview	318
2. Configuration and Block Diagram.....	319
3. Operations.....	321
4. Setting Procedure Example.....	326
5. Operation Example.....	328
6. Registers	331
6.1. Software Watchdog Timer Load Register (WdogLoad)	332
6.2. Software Watchdog Timer Value Register (WdogValue)	333
6.3. Software Watchdog Timer Control Register (WdogControl)	334
6.4. Software Watchdog Timer Clear Register (WdogIntClr)	335
6.5. Software Watchdog Timer Interrupt Status Register (WdogRIS).....	336
6.6. Software Watchdog Timer Lock Register (WdogLock)	337
6.7. Hardware Watchdog Timer Load Register (WDG_LDR)	338
6.8. Hardware Watchdog Timer Value Register (WDG_VLR).....	339
6.9. Hardware Watchdog Timer Control Register (WDG_CTL)	340
6.10. Hardware Watchdog Timer Clear Register (WDG_ICL)	341
6.11. Hardware Watchdog Timer Interrupt Status Register (WDG_RIS)	342
6.12. Hardware Watchdog Timer Lock Register (WDG_LCK)	343
7. Notes.....	344

CHAPTER 12: Dual Timer.....	345
1. Overview	346
2. Architecture	347
3. Operation Description.....	348
3.1. Timer Operating Mode	349
3.2. Default.....	353
3.3. Interrupt Behavior.....	354
4. Setting Procedure Example.....	355
5. Register	357
5.1. Load Register (TimerXLoad) X=1 or 2	358
5.2. Value Register (TimerXValue) X=1 or 2	359
5.3. Control Register (TimerXControl) X=1 or 2.....	360
5.4. Interrupt Clear Register (TimerXIntClr) X=1 or 2	362
5.5. Interrupt Status Register (TimerXRIS) X=1 or 2	363
5.6. Masked Interrupt Status Register (TimerXMIS) X=1 or 2	364
5.7. Background Load Register (TimerXBGLoad) X=1 or 2	365
CHAPTER 13-1: Watch Counter Prescaler.....	367
1. Overview of the Watch Counter Prescaler.....	368
2. Configuration of the Watch Counter Prescaler	369
3. Explanation of Operations and Setting Procedure Examples of the Watch Counter Prescaler	370
4. Registers of the Watch Counter Prescaler	372
4.1. Clock Selection Register (CLK_SEL).....	373
4.2. Division Clock Enable Register (CLK_EN)	374
CHAPTER 13-2: Watch Counter	375
1. Overview of the Watch Counter.....	376
2. Configuration of the Watch Counter	377
3. Interrupts of the Watch Counter.....	378
4. Explanation of Operations and Setting Procedure Examples of the Watch Counter	379
5. Registers of the Watch Counter.....	381
5.1. Watch Counter Read Register (WCRD)	382
5.2. Watch Counter Reload Register (WCRL)	383
5.3. Watch Counter Control Register (WCCR)	384
CHAPTER 14-1: Base Timer I/O Select Function	387
1. Overview	388
2. Configuration	389
3. I/O Mode.....	390
3.1. Pins	391
3.2. I/O mode	393
4. Registers	408
4.1. I/O Select Register (BTSEL0123).....	409
4.2. I/O Select Register (BTSEL4567)	411
4.3. Software-based Simultaneous Startup Register (BTSSSR)	413

CHAPTER 14-2: Base Timer.....	415
1. Overview of Base Timer	416
2. Block Diagram Of Base Timer	418
3. Operations of the Base Timer.....	421
4. 32-bit mode operations.....	423
5. Base Timer Interrupt.....	425
6. Starting the DMA Controller (DMAC)	426
7. Base Timer Registers	427
8. Notes on using the base timer.....	428
9. Descriptions of base timer functions.....	429
9.1. PWM timer function.....	430
9.1.1. 16-bit PWM timer operations.....	431
9.1.2. One-shot operation	432
9.1.3. Interrupt causes and timing chart.....	433
9.1.4. Output waveforms	434
9.1.5. PWM timer operation flowchart.....	435
9.1.6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWM timer is selected	436
9.1.7. PWM Cycle Set Register (PCSR)	443
9.1.8. PWM Duty Set Register (PDUT)	444
9.1.9. Timer Register (TMR).....	445
9.2. PPG timer function	446
9.2.1. 16-bit PPG timer operations.....	447
9.2.2. Continuous operation	448
9.2.3. One-shot operation	449
9.2.4. Interrupt causes and timing chart.....	451
9.2.5. PPG timer operation flowchart	452
9.2.6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PPG timer is selected	453
9.2.7. LOW Width Reload Register (PRLL).....	460
9.2.8. HIGH Width Reload Register (PRLH)	461
9.2.9. Timer Register (TMR).....	462
9.3. Reload timer function	463
9.3.1. Operations of the 16-bit reload timer.....	464
9.3.2. Reload timer operation flowchart	467
9.3.3. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the reload timer is selected	468
9.3.4. PWM Cycle Set Register (PCSR)	476
9.3.5. Timer Register (TMR).....	477
9.4. PWC timer function	478
9.4.1. Operations of the PWC timer	479
9.4.2. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWC timer is selected	486
9.4.3. Data Buffer Register (DTBF)	493

CHAPTER 15: Multifunction Timer	495
1. Overview of Multifunction Timer	496
2. Configuration of Multifunction Timer	497
2.1. Block Diagram of Multifunction Timer	498
2.2. Description of Each Function Block	499
2.3. I/O Pins of Multifunction Timer Unit.....	505
3. Operations of Multifunction Timer.....	508
3.1. Example of Operation of Multifunction Timer - 1	509
3.2. Example of Operation of Multifunction Timer - 2.....	516
4. Registers of Multifunction Timer	524
4.1. Individual Notation and Common Notation of Channel Numbers in Descriptions of Functions	525
4.2. List of Registers of Multifunction Timer	527
4.3. Details of Register Functions	529
4.3.1. FRT Control Register A (TCSA)	530
4.3.2. FRT Control Register B (TCSB).....	538
4.3.3. FRT Cycle Setting Register (TCCP).....	540
4.3.4. FRT Count Value Register (TCDT)	542
4.3.5. OCU Connecting FRT Select Register (OCFS)	543
4.3.6. OCU Control Register A (OCSA).....	544
4.3.7. OCU Control Register B (OCSB)	548
4.3.8. OCU Control Register C (OCSC).....	551
4.3.9. OCU Compare Value Store Register (OCCP)	552
4.3.10. WFG Control Register A (WFSA)	554
4.3.11. WFG Timer Value Register (WFTM).....	559
4.3.12. NZCL Control Register (NZCL)	560
4.3.13. WFG Interrupt Control Register (WFIR).....	564
4.3.14. ICU Connecting FRT Select Register (ICFS)	569
4.3.15. ICU Control Register A (ICSA)	570
4.3.16. ICU Control Register B (ICSB).....	573
4.3.17. ICU Capture value store register (ICCP).....	574
4.3.18. ADCMP Control Register A (ACSA)	575
4.3.19. ADCMP Control Register B (ACSB)	580
4.3.20. ADCMP Compare Value Store Register (ACCP).....	583
4.3.21. ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN)....	585
4.3.22. ADC Start Trigger Select Register (ATSA)	587
4.4. Details of OCU Output Waveform	590
4.5. Details of WFG Output Waveform.....	597
5. Other Matters	605
5.1. Connection of Model Containing Multiple MFT's	606
5.1.1. Selection of FRT Connected to OCU and ICU	607
5.1.2. PPG Timer Unit Connected to WFG	609
5.2. Treatment of Event Detect Register and Interrupt	610

CHAPTER 16-1: PPG Configuration	615
1. Configuration	616
CHAPTER 16-2: PPG	619
1. Overview	620
2. Configuration and Block Diagrams	621
3. Operations	623
3.1. PPG circuit operations	624
3.2. Timing generator circuit operations	628
4. Setup Procedure Example	630
5. Registers	632
5.1. PPG Start Trigger Control Register 0 (TTCR0)	634
5.2. PPG Start Trigger Control Register 1 (TTCR1)	636
5.3. PPG Compare Register "n" (COMP _n , where n=0 to 7)	638
5.4. PPG Start Register (TRG)	639
5.5. Output Reverse Register (REVC)	640
5.6. PPG Operation Mode Control Register (PPGC)	641
5.7. PPG Reload Registers (PRLH, PRLL)	643
5.8. PPG Gate Function Control Registers (GATEC0/GATEC4/GATEC4/GATEC8/GATEC12)	646
6. Notes	648
CHAPTER 17: Quad Position & Revolution Counter	649
1. Overview	650
2. Configuration	651
3. Operations	652
4. Registers	665
4.1. Quad Position & Revolution Counter Position Count Register (QPCR)	666
4.2. QPRC Revolution Count Register (QRCR)	668
4.3. QPRC Position Counter Compare Register (QPCCR)	669
4.4. QPRC Position and Revolution Counter Compare Register (QPRCR)	670
4.5. QPRC Control Register (QCR)	671
4.6. QPRC Extension Control Register (QECSR)	676
4.7. Low-Order Bytes of QPRC Interrupt Control Register (QICRL)	678
4.8. High-Order Bytes of QPRC Interrupt Control Register (QICRH)	682
4.9. QPRC Maximum Position Register(QMPR)	686

CHAPTER 18-1: A/D Converter	687
1. Configuration	688
2. Functions and Operations	689
3. Notes	690
CHAPTER 18-2: 10-bit A/D Converter	691
1. Overview	692
2. Configuration	693
3. Explanation of Operations	694
3.1. A/D conversion operation	695
3.1.1. Scan conversion operation	696
3.1.2. Priority conversion operation	698
3.1.3. Priority levels and state transitions	699
3.2. FIFO operations	701
3.2.1. FIFO operations in scan conversion	702
3.2.2. Interrupts in scan conversion	703
3.2.3. FIFO operations in priority conversion	705
3.2.4. Interrupts in priority conversion	706
3.2.5. Restrictions on reading FIFO data registers in empty state	707
3.2.6. Bit placement selection for FIFO data registers	708
3.3. A/D comparison function	709
3.4. Starting DMA	710
4. Setup procedure examples	711
4.1. Scan conversion setup procedure example	712
4.2. Priority conversion setup procedure example	713
4.3. Setting the conversion time	714
5. Registers	715
5.1. A/D Control Register (ADCR)	716
5.2. A/D Status Register (ADSR)	718
5.3. Scan Conversion Control Register (SCCR)	720
5.4. Scan Conversion FIFO Stage Count Setup Register (SFNS)	722
5.5. Scan Conversion FIFO Data Register (SCFD)	723
5.6. Scan Conversion Input Selection Register (SCIS)	724
5.7. Priority Conversion Control Register (PCCR)	725
5.8. Priority Conversion FIFO Stage Count Setup Register (PFNS)	727
5.9. Priority Conversion FIFO Data Register (PCFD)	728
5.10. Priority Conversion Input Selection Register (PCIS)	729
5.11. A/D Comparison Value Setup Register (CMRD)	730
5.12. A/D Comparison Control Register (CMPCR)	731
5.13. Sampling Time Selection Register (ADSS)	732
5.14. Sampling Time Setup Register (ADST)	733
5.15. Comparison Time Setup Register (ADCT)	735

CHAPTER 18-3: 12-bit A/D Converter	737
1. Overview	738
2. Configuration	739
3. Explanation of Operations	740
3.1. Enabling operations of the A/D converter	741
3.2. A/D conversion operation	742
3.2.1. Scan conversion operation.....	743
3.2.2. Priority conversion operation.....	745
3.2.3. Priority levels and state transitions.....	746
3.3. FIFO operations	748
3.3.1. FIFO operations in scan conversion	749
3.3.2. Interrupts in scan conversion	750
3.3.3. FIFO operations in priority conversion	752
3.3.4. Interrupts in priority conversion	753
3.3.5. Validity of FIFO data.....	754
3.3.6. Bit placement selection for FIFO data registers	755
3.4. A/D comparison function	756
3.5. Starting DMA	757
4. Setup procedure examples.....	758
4.1. A/D Operation Enable Setup Procedure Example	759
4.2. Scan conversion setup procedure example.....	760
4.3. Priority conversion setup procedure example.....	761
4.4. Setting the conversion time.....	762
5. Registers	763
5.1. A/D Control Register (ADCR).....	764
5.2. A/D Status Register (ADSR)	766
5.3. Scan Conversion Control Register (SCCR).....	768
5.4. Scan Conversion FIFO Stage Count Setup Register (SFNS)	770
5.5. Scan Conversion FIFO Data Register (SCFD)	771
5.6. Scan Conversion Input Selection Register (SCIS)	773
5.7. Priority Conversion Control Register (PCCR).....	774
5.8. Priority Conversion FIFO Stage Count Setup Register (PFNS)	776
5.9. Priority Conversion FIFO Data Register (PCFD)	777
5.10. Priority Conversion Input Selection Register (PCIS)	779
5.11. A/D Comparison Value Setup Register (CMPD)	780
5.12. A/D Comparison Control Register (CMPCR)	781
5.13. Sampling Time Selection Register (ADSS).....	782
5.14. Sampling Time Setup Register (ADST)	783
5.15. Comparison Time Setup Register (ADCT).....	785
5.16. A/D Operation Enable Setup Register (ADCEN)	786
CHAPTER 18-4: A/D Timer Trigger Selection	789
1. Overview	790
2. Registers	791
2.1. Scan Conversion Timer Trigger Selection Register (SCTSL).....	792
2.2. Priority Conversion Timer Trigger Selection Register (PRTSL)	793

CHAPTER 19-1: Multi-function Serial Interface	795
1. Overview of the Multi-function Serial Interface	796
CHAPTER 19-2: UART (Async Serial Interface)	797
1. Overview of UART (Async Serial Interface).....	798
2. UART Interrupt	799
2.1. Receive interrupt and flag set timing.....	800
2.2. Interrupt and flag set timing when receive FIFO is used	802
2.3. Transmit interrupt and flag set timing.....	804
2.4. Interrupt and flag set timing when transmit FIFO is used	805
3. UART Operation	806
4. Dedicated Baud Rate Generator	814
4.1. Baud rate settings	815
5. Setting Procedure and Program Flow in Operation Mode 0 (Async Normal Mode).....	819
6. Setting Procedure and Program Flow in Operation Mode 1 (Async Multiprocessor Mode).....	822
7. UART (Async Serial Interface) Registers	826
7.1. Serial Control Register (SCR).....	827
7.2. Serial Mode Register (SMR)	830
7.3. Serial Status Register (SSR).....	833
7.4. Extended Communication Control Register (ESCR)	836
7.5. Receive Data Register/Transmit Data Register (RDR/TDR)	839
7.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)	841
7.7. FIFO Control Register 1 (FCR1)	843
7.8. FIFO Control Register 0 (FCR0)	846
7.9. FIFO Byte Register (FBYTE)	850

CHAPTER 19-3: CSIO (Clock Sync Serial Interface).....	853
1. Outline of CSIO (Clock Sync Serial Interface)	854
2. CSIO (Clock Sync Serial Interface) interrupts	855
2.1. Receive interrupt and flag set timing.....	856
2.2. Interrupt occurrence and flag set timing when receive FIFO is used	858
2.3. Transmit interrupt occurrence and flag set timing.....	860
2.4. Interrupt occurrence and flag set timing when transmit FIFO is used	861
3. CSIO (Clock Sync Serial Interface) operations.....	862
3.1. Normal transfer (I)	862
3.2. Normal transfer (II)	867
3.3. SPI transfer (I).....	872
3.4. SPI transfer (II).....	877
4. Dedicated baud rate generator.....	882
4.1. Baud rate settings	883
4.2. CSIO (Clock Sync Serial Interface) setup procedure and program flow	885
5. CSIO (Clock Sync Serial Interface) registers.....	887
5.1. Serial Control Register (SCR).....	888
5.2. Serial Mode Register (SMR).....	891
5.3. Serial Status Register (SSR).....	894
5.4. Extended Communication Control Register (ESCR)	896
5.5. Receive Data Register/Transmit Data Register (RDR/TDR)	898
5.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)	900
5.7. FIFO Control Register 1 (FCR1)	902
5.8. FIFO Control Register 0 (FCR0)	905
5.9. FIFO Byte Register (FBYTE)	909
CHAPTER 19-4: LIN Interface (Ver. 2.1)	
(LIN Communication Control Interface Ver. 2.1)	911
1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1) ...	912
2. LIN Interface (Ver. 2.1) Interrupts	913
2.1. Receive interrupt and flag set timing.....	914
2.2. Interrupt and flag set timing when receive FIFO is used	916
2.3. Transmit interrupt and flag set timing.....	918
2.4. Interrupt and flag set timing when transmit FIFO is used	919
3. Dedicated Baud Rate Generator	920
3.1. Baud rate settings	921
4. LIN Interface (Ver. 2.1) Operations	925
5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow.....	935
6. LIN Interface (ver. 2.1) Registers.....	940
6.1. Serial Control Register (SCR).....	941
6.2. Serial Mode Register (SMR)	945
6.3. Serial Status Register (SSR).....	947
6.4. Extended Communication Control Register (ESCR)	951
6.5. Receive Data Register/Transmit Data Register (RDR/TDR)	953
6.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)	955
6.7. FIFO Control Register 1 (FCR1)	957
6.8. FIFO Control Register 0 (FCR0)	960
6.9. FIFO Byte Register (FBYTE)	964

CHAPTER 19-5: I ² C Interface (I ² C Communications Control Interface).....	967
1. Overview of I ² C Interface (I ² C Communications Control Interface)	968
2. I ² C Interface interrupt	969
2.1. I ² C interface operation	971
2.2. Master mode	972
2.3. Slave mode	1003
2.4. Bus error	1011
3. Dedicated Baud Rate Generator	1012
4. I ² C communication operation flowchart examples.....	1014
5. I ² C Interface Registers	1021
5.1. I ² C Bus Control Register (IBCR).....	1022
5.2. Serial Mode Register (SMR)	1029
5.3. I ² C Bus Status Register (IBSR).....	1031
5.4. Serial Status Register (SSR).....	1036
5.5. Receive Data Register/Transmit Data Register (RDR/TDR)	1040
5.6. 7-bit Slave Address Mask Register (ISMK).....	1042
5.7. 7-bit Slave Address Register (ISBA).....	1043
5.8. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)	1044
5.9. FIFO Control Register 1 (FCR1)	1045
5.10. FIFO Control Register 0 (FCR0)	1048
5.11. FIFO Byte Register (FBYTE)	1053
CHAPTER 20-1: USB Clock Generation.....	1055
1. Overview	1056
2. Configuration and Block Diagram.....	1057
3. Explanation of Operation	1058
4. Setup Procedure Example.....	1060
5. Register List	1061
5.1. USB Clock Setup Register (UCCR).....	1062
5.2. USB-PLL Control Register-1 (UPCR1)	1063
5.3. USB-PLL Control Register-2 (UPCR2)	1064
5.4. USB-PLL Control Register 3 (UPCR3)	1065
5.5. USB-PLL Control Register 4 (UPCR4)	1066
5.6. USB-PLL Macro Status Register (UP_STR)	1067
5.7. USB-PLL Interrupt Enable Register (UPINT_ENR)	1068
5.8. USB-PLL Interrupt Status Register (UPINT_STR).....	1069
5.9. USB-PLL Interrupt Clear Register (UPINT_CLR)	1070
5.10. USB Enable Request Register (USBEN).....	1071
6. Usage Precautions	1072

CHAPTER 20-2: USB Function	1073
1. Overview of USB Function	1074
1.1. Features of USB function	1074
2. Configuration of USB Function	1075
3. Operations of USB Function.....	1076
3.1. USB function operation	1077
3.2. Detection of connection and disconnection	1080
3.3. Operation of each register in response to a command.....	1083
3.4. Suspend function	1086
3.5. Wake-up function	1087
3.6. DMA transfer function	1089
3.7. NULL transfer function	1094
3.8. STALL response/release of endpoint 0.....	1095
3.9. Stall response/release of endpoints 1 to 5	1097
4. Examples of USB Function Setting Procedures	1101
5. USB Function Registers	1108
5.1. UDC Control Register (UDCC)	1110
5.2. EP0 Control Register (EP0C)	1113
5.3. EP1 to 5 Control Registers (EP1C to EP5C)	1115
5.4. Time Stamp Register (TMSP)	1119
5.5. UDC Status Register (UDCS)	1120
5.6. UDC Interrupt Enable Register (UDCIE).....	1123
5.7. EP0I Status Register (EP0IS)	1125
5.8. EP0O Status Register (EP0OS).....	1127
5.9. EP1 to 5 Status Registers (EP1S to EP5S)	1129
5.10. EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)	1133

CHAPTER 20-3: USB Host	1135
1. Overview of USB host	1136
2. USB host configuration.....	1138
3. USB host operations	1139
3.1. Device connection.....	1140
3.2. USB bus resetting	1142
3.3. Token packet	1143
3.4. Data packet	1146
3.5. Handshake packet	1147
3.6. Retry function	1148
3.7. SOF interrupt.....	1149
3.8. Error status.....	1151
3.9. End of packet	1152
3.10. Suspend and resume operations	1153
3.11. Device disconnection	1156
4. USB host setting procedure examples	1157
5. USB host registers	1162
5.1. Host Control Registers 0 and 1 (HCNT0 and HCNT1)	1164
5.2. Host Interrupt Register (HIRQ)	1169
5.3. Host Error Status Register (HERR).....	1173
5.4. Host Status Register (HSTATE)	1176
5.5. SOF Interrupt Frame Compare Register (HFCOMP)	1179
5.6. Retry Timer Setup Register (HRTIMER).....	1180
5.7. Host Address Register (HADR).....	1181
5.8. EOF Setup Register (HEOF)	1182
5.9. Frame Setup Register (HFRAME)	1183
5.10. Host Token Endpoint Register (HTOKEN)	1184

CHAPTER 21-1: CAN Prescaler	1187
1. Overview and configuration	1188
2. CAN Prescaler Register	1189
2.1. CAN Prescaler Register (CANPRE)	1190
CHAPTER 21-2: CAN Controller.....	1193
1. Overview	1194
2. Configuration	1195
3. CAN Controller Operations	1196
3.1. Message objects	1197
3.2. Message transmission	1199
3.3. Message reception.....	1202
3.4. FIFO buffer function	1205
3.5. Interrupt function	1207
3.6. Bit timing	1208
3.7. Test mode.....	1210
3.8. Software initialization	1214
4. CAN Registers.....	1215
4.1. CAN register functions	1218
4.2. Total control registers	1219
4.2.1. CAN Control Register (CTRLR)	1220
4.2.2. CAN Status Register (STATR).....	1224
4.2.3. CAN Error Counter (ERRCNT)	1227
4.2.4. CAN Bit Timing Register (BTR).....	1229
4.2.5. CAN Interrupt Register (INTR)	1230
4.2.6. CAN Test Register (TESTR)	1231
4.2.7. CAN Prescaler Extension Register (BRPER)	1233
4.3. Message interface registers.....	1234
4.3.1. IFx Command Request Register (IFxCREQ)	1235
4.3.2. IFx Command Mask Register (IFxCMSK).....	1237
4.3.3. IFx Mask Registers 1, 2 (IFxMSK1 ,IFxMSK2)	1242
4.3.4. IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2).....	1243
4.3.5. IFx Message Control Register (IFxMCTR).....	1244
4.3.6. IFx Data Registers A1, A2, B1, and B2 (IFxDTA1, IFxDTA2, IFxDTB1, and IFxDTB2)	1245
4.4. Message objects	1246
4.5. Message handler registers.....	1252
4.5.1. CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2).....	1253
4.5.2. CAN New Data Registers 1, 2 (NEWDT1, NEWDT2).....	1255
4.5.3. CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2).....	1257
4.5.4. CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)	1259

CHAPTER 22: CRC (Cyclic Redundancy Check)	1261
1. CRC Overview.....	1262
2. CRC Operations	1263
2.1. CRC calculation sequence.....	1264
2.2. CRC use examples	1265
3. CRC Registers	1269
3.1. CRC Control Register (CRCCR).....	1270
3.2. Initial Value Register (CRCINIT)	1273
3.3. Input Data Register (CRCIN)	1274
3.4. CRC Register (CRCR)	1275
CHAPTER 23: External Bus Interface.....	1277
1. External Bus Interface Features	1278
2. Block Diagram	1279
3. Operation.....	1280
3.1. Fundamental SRAM and flash memory accesses.....	1281
3.2. NAND flash memory access	1282
3.2.1. Read access to NAND flash memory.....	1283
3.2.2. Write (auto program) access	1284
3.2.3. Auto block erase access	1285
3.2.4. Error reply	1286
4. Example waveforms of external memory access	1287
4.1. Word read access to 8-bit wide SRAM	1288
4.2. Serial word write/read access to 16-bit wide SRAM	1289
4.3. Issue of an 8-bit NAND flash memory read/write command.....	1290
4.4. 8-bit NAND flash memory status read	1291
4.5. 8-bit NAND flash memory data write.....	1292
4.6. 16-bit NOR flash memory page read	1293
5. Endianness and Valid Byte Lanes	1294
6. Connection Examples	1297
7. Registers	1299
7.1. Mode Register 0 to Mode Register 7	1300
7.2. Timing Register 0 to Timing Register 7	1302
7.3. Area Register 0 to Area Register 7	1306
CHAPTER 24: Debug Interface.....	1309
1. Overview	1310
2. Pin Description	1311
2.1. Pins for Debug Purposes	1312
2.2. ETM Pins.....	1313
2.3. Functions Initially Assigned to Pins.....	1314
2.4. Internal Pull-Ups of JTAG Pins.....	1315
APPENDICES	1317
1. Register Map	1318

MAJOR CHANGES IN THIS EDITION

Page	Section	Change Results
-	-	MB9BFxxx Series, MB9BF500 Series → this Series
9	CHAPTER 1 System Overview	<p>3. Cortex-M3 Architecture</p> <ul style="list-style-type: none"> · Corrected the description of ". Cortex-M3 Core Version" of "■ Cortex-M3 Core".
11		<p>4. Mode</p> <ul style="list-style-type: none"> · external reset → INITX pin input reset
17	CHAPTER 2-1 Clock	<p>2. Clock Generation Unit Configuration/Block Diagram</p> <ul style="list-style-type: none"> · Corrected Figure 2-1. (Deleted USB-PLL block.)
21		<p>3.3. PLL clock control</p> <ul style="list-style-type: none"> · Corrected Table 3-2. · Added <Notes>.
22		<p>3.4. Oscillation stabilization wait time</p> <ul style="list-style-type: none"> · Corrected <Notes>.
24, 25		<p>4. Clock Setup Procedure Examples</p> <ul style="list-style-type: none"> · Corrected Figure 4-1 and Figure 4-2. · Main CR → High-speed CR · Sub CR → Low-speed CR · Corrected <Notes>.
28		<p>5.1. System Clock Mode Control Register (SCM_CTL)</p> <ul style="list-style-type: none"> · Corrected <Notes>.
29, 30		<p>5.2. System Clock Mode Status Register (SCM_STR)</p> <ul style="list-style-type: none"> · Corrected the description of [bit2] and [bit0]. (Deleted "Set this bit to "0" when writing.".)
37		<p>5.9. Clock Stabilization Wait Time Register (CSW_TMR)</p> <ul style="list-style-type: none"> · Corrected the description of [bit6:4].
47, 48		<p>6. Clock Generation Unit Usage Precautions</p> <ul style="list-style-type: none"> · Corrected ". Switching clock modes". · Corrected the description of ". Correlation between the clock mode switching and the oscillation stable bit". (Main CR → High-speed CR, Sub CR → Low-speed CR) · If the standby mode is released by an interrupt, the clock mode is switched to that indicated by the RCM bit in the SCM_CTL. <p style="text-align: center;">→</p> <p>If the standby mode is released by an interrupt, the device restarts in the clock mode that indicated by the RCM bit in the SCM_CTL.</p>

Page	Section	Change Results
All	CHAPTER 2-2 High-Speed CR Trimming	Separated from the "Clock" chapter.
51		<p>2. High-Speed CR Trimming Function Configuration and Block Diagram</p> <ul style="list-style-type: none"> · Corrected Figure 2-1. (CRTRIM → CLKHC_div)
53 to 58		<p>4. High-Speed CR Trimming Function Setup Procedure Example</p> <ul style="list-style-type: none"> · Corrected Figure 4-1. (TRD[7:0] → TRD) · Corrected the description of "■ Frequency trimming setup". (...the register. → ...the MCR_FTRM Register.) · Corrected the whole description in "■ How to calculate the frequency trimming data". · Corrected the whole description in "■ Example trimming data acquisition using input capture". · Added "■ Frequency trimming procedure example". · Added "■ Xtrm calculation procedure example". · Added "■ Procedure example of using "CR trimming" area storage data inside flash memory".
60		<p>5.1. High-speed CR oscillation Frequency Division Setup Register (MCR_PSR)</p> <ul style="list-style-type: none"> · Added the explanation to the summaries.
61		<p>5.2. High-speed CR oscillation Frequency Trimming Register (MCR_FTRM)</p> <ul style="list-style-type: none"> · Changed the whole description.
62		<p>5.3. High-Speed CR Oscillator Register Write-Protect Register (MCR_RLR)</p> <ul style="list-style-type: none"> · Corrected the summaries. · Corrected the description of [bit31:0].
67	CHAPTER 3 Resets	<p>2. Configuration</p> <ul style="list-style-type: none"> · Corrected the figures of "■ Block Diagram of Resets".
69		<p>3.1. Reset Causes</p> <ul style="list-style-type: none"> · Corrected the table of "■ Low-voltage Detection Reset, External Voltage Monitoring (LVDH)". (Deleted "Note:" of "Initialization target".) (bit3(LVDH) → bit0(PONR))
73		<p>3.2.1. Resets to Cortex-M3</p> <ul style="list-style-type: none"> · Power initialization reset → Power-on reset
74		<p>3.2.2. Resets to Peripheral Circuit</p> <ul style="list-style-type: none"> · Power initialization reset → Power-on reset · Corrected <Notes>. (APVC1 → APBC1)
79 to 81		<p>4.1. Reset Cause Register (RST_STR: ReSeT STatus Register)</p> <ul style="list-style-type: none"> · Corrected bit3. (LVDH → Reserved) · Corrected the description of [bit0]. (Added low-voltage detection reset.) · Corrected <Note>. (Added low-voltage detection reset.)

Page	Section	Change Results
84	CHAPTER 4: Low-voltage Detection	<p>1. Overview</p> <ul style="list-style-type: none"> · Corrected the description of "● Operations of Low-voltage Reset Circuit". · Corrected <Notes>.
85		<p>2. Configuration</p> <ul style="list-style-type: none"> · Corrected the figures of "■ Block diagram of Low-voltage Detection Circuit". · Corrected the description of "● Low-voltage Detection Voltage Control Register". · Added "● Low-voltage Detection Circuit Status Register".
87		<p>3. Explanation of Operations</p> <ul style="list-style-type: none"> · "■ Operations of Low-Voltage Detection Reset Circuit" · Corrected the description. · Corrected the figure. · Deleted <Notes>.
88, 89		<ul style="list-style-type: none"> · "■ Operations of Low-voltage Detection Interrupt Circuit" · Corrected the description of "● Operations". · Corrected the figures of "● Canceling a low-voltage detection interrupt request". · Corrected <Note>.
90		<p>4. Setup Procedure Examples</p> <ul style="list-style-type: none"> · Deleted "Stopping a low-voltage detection reset". · Deleted "Restarting a low-voltage detection reset ". · Corrected Figure 4-1.
92, 93		<p>5.1. Low-voltage Detection Voltage Control Register (LVD_CTL)</p> <ul style="list-style-type: none"> · Corrected the summaries. · Corrected bit6. (LVDRE → Reserved) · Corrected <Notes>.
97		Added "5.5. Low-voltage Detection Circuit Status Register (LVD_STR2)".
116	CHAPTER 5 Low Power Consumption Mode	<p>3.2. Operations of TIMER modes (high speed CR timer, main timer, PLL timer, low speed CR timer, and sub timer modes)</p> <ul style="list-style-type: none"> · Added <Notes>.
119		<p>3.3. Operations of STOP mode</p> <ul style="list-style-type: none"> · Added <Notes>.
121		<p>4. Standby Mode Setting Procedure Examples</p> <ul style="list-style-type: none"> · Added <Note>.

Page	Section	Change Results
Whole chapter	CHAPTER 6: Interrupts	<p>Corrected register name.</p> <ul style="list-style-type: none"> · Interrupt Request Batch Read Register 00 (IRQMON00) → EXC02 Batch Read Register (EXC02MON) · Interrupt Request Batch Read Register 01 to 48 (IRQMON01 to 48) → IRQ00 Batch Read Register to IRQ47 Batch Read Register (IRQ00MON to IRQ47MON)
128 to 130		<p>3. Exception and Interrupt Vectors</p> <ul style="list-style-type: none"> · Corrected the header of Table 3-1. (Interrupt No. → IRQ No.)
133		<p>4.1. DMA Request Selection Register (DRQSEL)</p> <ul style="list-style-type: none"> · Corrected the summaries. (IRQMONxx, xx=00 to 48 → IRQxxMON, xx=00 to 47)
158		<p>4.19. IRQ34 Batch Read Register (IRQ34MON)</p> <ul style="list-style-type: none"> · Corrected the description of [bit4:0]. (IRQ2 to 6 → DRQ)
159		<p>4.20. IRQ35 Batch Read Register (IRQ35MON)</p> <ul style="list-style-type: none"> · Corrected the description of [bit5:0]. (IRQ7 to 10 → USB Interrupt Source, IRQ1 → DRQO, IRQ0 → DRQI)
176	CHAPTER 7 External Interrupt and NMI Control Sections	<p>4.2. External Interrupt Request Register [EIRR]</p> <ul style="list-style-type: none"> · Corrected the initial value of "■ Register configuration". (0 → X) · Added <Notes>.
191	CHAPTER 8 DMAC	<p>3.2. Software-Burst Transfer</p> <ul style="list-style-type: none"> · Corrected the description of ". Transfer data size". (TW=00 → TW=10)
225		<p>5.4. Configuration B Register (DMACB)</p> <ul style="list-style-type: none"> · Corrected bit15:8. (SP[3:0], DP[3:0] → Reserved)

Page	Section	Change Results
Whole chapter	CHAPTER 9 I/O PORT	<p>The descriptions have been changed by sharing USB pin/sub oscillation pin and the ports.</p> <ul style="list-style-type: none"> · Added the description of USB pin and sub oscillation pin. · The following registers are added. Port function setting register 8 (PFR8) Port input/output direction setting register 8 (DDR8) Port input data register 8 (PDIR8) Port output data register 8 (PDOR8) Special Port Setting Register (SPSR)
235		<p>2. Configuration, Block Diagram, and Operation</p> <ul style="list-style-type: none"> · Corrected Figure 2-1. (Deleted a signal line that extends from the PDOR to EPFR register.)
244		<p>4.1. Port Function Setting Register (PFRx)</p> <ul style="list-style-type: none"> · Corrected <Notes>.
245		<p>4.2. Pull-up Setting Register (PCR_x)</p> <ul style="list-style-type: none"> · Added <Notes>.
246		<p>4.3. Port input/output Direction Setting Register (DDRx)</p> <ul style="list-style-type: none"> · Added <Notes>.
247		<p>4.4. Port Input Data Register (PDIR_x)</p> <ul style="list-style-type: none"> · Added <Notes>.
248		<p>4.5. Port Output Data Register x (PDOR_x)</p> <ul style="list-style-type: none"> · Added <Notes>.
250		<p>4.7. Extended Pin Function Setting Register (EPFR_x)</p> <ul style="list-style-type: none"> · Corrected the initial value.
252		<p>4.8. Extended Pin Function Setting Register 00 (EPFR00)</p> <ul style="list-style-type: none"> · Corrected the description of [bit1]. (Internal CR → Internal high-speed CR)
260		<p>4.10. Extended Pin Function Setting Register 02 (EPFR02)</p> <ul style="list-style-type: none"> · Corrected the error in description. (RTO02 → RTO12)
264		<p>4.11. Extended Pin Function Setting Register 04 (EPFR04)</p> <ul style="list-style-type: none"> · Added <Notes>.
268		<p>4.12. Extended Pin Function Setting Register 05 (EPFR05)</p> <ul style="list-style-type: none"> · Added <Notes>.
295		<p>5. Usage Precautions</p> <ul style="list-style-type: none"> · Corrected the description of "● Product Specifications and Peripheral Function Assignment".
315	CHAPTER 10 Clock supervisor	<p>7. Usage Precautions</p> <ul style="list-style-type: none"> · Added ". The settings for CSV OFF and external reset.".

Page	Section	Change Results
321 to 325	CHAPTER 11 Watchdog timer	<p>3. Operations</p> <ul style="list-style-type: none"> · Corrected Table 3-1. · Deleted the following descriptions from "● Value register (WdogValue)". " However, during tool break, the value can be read only when the watchdog timer is stopped." · Corrected the description of "● Reload and lock of the register of the software watchdog timer". · Corrected the description of "● Value register (WDG_VLR)". · Corrected Table 3-6.
326, 327		<p>4. Setting Procedure Example</p> <ul style="list-style-type: none"> · Corrected Figure 4-1 and Figure 4-2.
332		<p>6.1. Software Watchdog Timer Load Register (WdogLoad)</p> <ul style="list-style-type: none"> · Corrected the description of [bit31:0]. <p>In case of writing</p> <p>When "0" is written, an interrupt will be generated right away. → When "0" is written, an interrupt will be generated.</p> <p>In case of reading</p> <p>Added "A set value can be read.".</p> <ul style="list-style-type: none"> · Corrected <Notes>. (Deleted "right away".)
333		<p>6.2. Software Watchdog Timer Value Register (WdogValue)</p> <ul style="list-style-type: none"> · Deleted the following description from <Note>. "Reading of this register is possible only if the watchdog timer is set as it stops at tool break. Other read value is not guaranteed."
336		<p>6.5. Software Watchdog Timer Interrupt Status Register (WdogRIS)</p> <ul style="list-style-type: none"> · Corrected the attribute of "■ Register configuration". (bit0 : R/W → R) · Corrected the description of [bit0].
337		<p>6.6. Software Watchdog Timer Lock Register (WdogLock)</p> <ul style="list-style-type: none"> · Corrected the initial value of "■ Register configuration". (0x00000001 → 0x00000000) · Corrected the description of "In case of writing" of [bit31:0]. · Corrected <Notes>.
338		<p>6.7. Hardware Watchdog Timer Load Register (WDG_LDR)</p> <ul style="list-style-type: none"> · Corrected the description of [bit31:0]. <p>In case of writing</p> <p>An interrupt is generated immediately after "0" is written. → An interrupt is generated after "0" is written.</p> <p>In case of reading</p> <p>Added "A set value can be read.".</p> <ul style="list-style-type: none"> · Corrected <Notes>.
339		<p>6.8. Hardware Watchdog Timer Value Register (WDG_VLR)</p> <ul style="list-style-type: none"> · Corrected <Notes>.
344		<p>7. Notes</p> <ul style="list-style-type: none"> · Added ". Hardware watchdog and interrupt handler". · See the section for Interrupt Source Register in the chapter "Clock" for an interrupt source. → See the section for Interrupt Source Register in the chapter "Interrupt" for an interrupt source.

Page	Section	Change Results
377	CHAPTER 13-2 Watch Counter	2. Configuration of the Watch Counter · Corrected the error in description. (Watch counter control register (WCRD) → (WCCR))
407	CHAPTER 14-1 Base Timer I/O Select Function	3.2. I/O mode · Corrected <Notes> of "■ I/O mode 8 (Shared channel signal trigger and timer start/stop mode". (· The odd channel stops... → · Base timer stops...)
409 to 412		4.1., 4.2. I/O Select Register (BTSEL0123, BTSEL4567) · Corrected the bit number of the register.
534	CHAPTER 15 Multifunction Timer	4.3.1 FRT Control Register A (TCSA) · Added the following to the description of item "2" in [bit6]. "Do not write TCSA.SCLR=0 until it can be checked that the counter value is cleared."
576		4.3.18. ADCMP Control Register A (ACSA) · Added the following to the description of [bit5:4]. "If the buffer function of ACCP and ACCPDN registers is to be used, use FRT-ch.0 for FRT to be connected."
580		4.3.19. ADCMP Control Register B (ACSB) · Added the following to the description of [bit2]. "If the buffer function of ACCP and ACCPDN registers is to be used, use FRT-ch.0 for FRT to be connected."
584		4.3.20. ADCMP Compare Value Store Register (ACCP) · Added <Notes>.
586		4.3.21. ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN) · Added <Notes>.
590, 591		4.4. Details of OCU Output Waveform · Corrected the whole description in "■ List of OCU Operation Modes".
Whole chapter	CHAPTER 16-1 PPG Configuration	Corrected the chapter name. (PPG → PPG Configuration) Deleted the description of "MB9BF500".
636	CHAPTER 16-2 PPG	5.2. PPG Start Trigger Control Register 1 (TTCR1) · Corrected the summaries. (PPG01/PPG03/PPG05/PPG07 → PPG8/PPG10/PPG12/PPG14)

Page	Section	Change Results
658	CHAPTER 17 Quad Position & Revolution Counter	<p>3. Operations</p> <ul style="list-style-type: none"> · "● RC_Mode1 (QCR:RCM[1:0] = "01"): Added <Notes>.
659		<ul style="list-style-type: none"> · "● RC_Mode3 (QCR:RCM[1:0] = "11"): Added <Notes>.
661		<ul style="list-style-type: none"> · "■ Operation example of QPRC Maximum Position Register (QMPPR) interrupt" The following ... RC_Mode2 (QCR:RCM[1:0]). → The following ... RC_Mode2 (QCR:RCM[1:0] = "10").
662		<ul style="list-style-type: none"> · "■ Position counter reset mask function" Deleted the description of "the ZIN function is set to the counter clear function (QCR:CGSC = "0") and when".
664		<ul style="list-style-type: none"> · "■ Position counter reset mask function" Added <Notes>.
667		<p>4.1. Quad Position & Revolution Counter Position Count Register (QPCR)</p> <ul style="list-style-type: none"> · Corrected <Notes>.
668		<p>4.2. QPRC Revolution Count Register (QRCR)</p> <ul style="list-style-type: none"> · Added <Notes>.
672, 673		<p>4.5. QPRC Control Register (QCR)</p> <ul style="list-style-type: none"> · "■ Low-Order Bytes of QPRC Control Register (QCRL)" · Added "(RC_Mode0 to 3)" to the description of [bit3:2]. · Added "(PC_Mode0 to 3)" to the description of [bit1:0]. · Added <Note>.
675		<ul style="list-style-type: none"> · "■ High-Order Bytes of QPRC Control Register (QCRH)" · Corrected the description of [bit9:8]. · Corrected <Notes>.
676, 677		<p>4.6. QPRC Extension Control Register (QEGR)</p> <ul style="list-style-type: none"> · Corrected the description of [bit1]. · Deleted <Notes> of [bit1]. · Corrected the description of [bit0]. (2K → 8K, "0x8000" → "0xFFFF")
680, 681		<p>4.7. Low-Order Bytes of QPRC Interrupt Control Register (QICRL)</p> <ul style="list-style-type: none"> · Corrected <Notes> of [bit3] and [bit1].
683, 684		<p>4.8. High-Order Bytes of QPRC Interrupt Control Register (QICRH)</p> <ul style="list-style-type: none"> · Corrected <Note> of [bit13]. · Added <Note> to [bit10] and [bit9].
Whole chapter	CHAPTER 18-1 A/D Converter	Added the description of 12 bit A/D converter.
690		<p>3. Notes</p> <ul style="list-style-type: none"> · Corrected the description of "■ Notes on 10-bit A/D converter".

Page	Section	Change Results
696	CHAPTER 18-2 10-bit A/D Converter	<p>3.1.1. Scan conversion operation</p> <ul style="list-style-type: none"> Corrected the title of Figure 3-1, Figure 3-2. (SCISO=0x04 → 0x08)
706		<p>3.2.4. Interrupts in priority conversion</p> <ul style="list-style-type: none"> Corrected <Note>. (PFS[3:0] → PFS[1:0])
710		<p>3.4. Starting DMA</p> <ul style="list-style-type: none"> Corrected the whole description and the figure.
713		<p>4.2. Priority conversion setup procedure example</p> <ul style="list-style-type: none"> Corrected Figure 4-2. (SFCLR → PFCLR)
714		<p>4.3. Setting the conversion time</p> <ul style="list-style-type: none"> Corrected the description of "■ Example of setting the sampling time". ((ADSS2 to 0) → (ADSS3 to 0))
Whole chapter	CHAPTER 18-3 12-bit A/D Converter	Added a new section.
Whole chapter	CHAPTER 18-4 A/D Timer Trigger Selection	Added the description of 12 bit A/D converter.
796	CHAPTER 19-1 Multi-function Serial Interface	<p>1. Overview of the Multi-function Serial Interface</p> <ul style="list-style-type: none"> Added "LIN(LIN bus interface)" to "■ Interface Mode" and Table 1-1. Added "■ LIN Sync field Detection: LSYN".
798	CHAPTER 19-2 UART	<p>1. Overview of UART (Async Serial Interface)</p> <ul style="list-style-type: none"> 128 bytes → 128 × 9 bits
817, 818		<p>4.1. Baud rate settings</p> <ul style="list-style-type: none"> Corrected the computation expression in "■ Allowable baud rate range for data reception". Corrected the value of "Minimum allowable baud rate error" in the table.
830		<p>7.2. Serial Mode Register (SMR)</p> <ul style="list-style-type: none"> Corrected the summaries.
844		<p>7.7. FIFO Control Register 1 (FCR1)</p> <ul style="list-style-type: none"> Added <Note> to [bit11].
849		<p>7.8. FIFO Control Register 0 (FCR0)</p> <ul style="list-style-type: none"> Corrected the description of [bit1] and [bit0].
851		<p>7.9. FIFO Byte Register (FBYTE)</p> <ul style="list-style-type: none"> Added <Notes>.
854	CHAPTER 19-3 CSIO	<p>1. Outline of CSIO (Clock Sync Serial Interface)</p> <ul style="list-style-type: none"> 128 bytes → 128 × 9 bits
903		<p>5.7. FIFO Control Register 1 (FCR1)</p> <ul style="list-style-type: none"> Added <Note> to [bit11].
907, 908		<p>5.8. FIFO Control Register 0 (FCR0)</p> <ul style="list-style-type: none"> Corrected the description of [bit1] and [bit0].
910		<p>5.9. FIFO Byte Register (FBYTE)</p> <ul style="list-style-type: none"> Added <Notes>.

Page	Section	Change Results
912	CHAPTER 19-4 LIN Interface	1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1) · 128 bytes → 128 × 9 bits
923, 924		3.1. Baud rate settings · Corrected Figure 3-1. (11xFL → 10xFL) · Corrected the computation expression in "■ Allowable baud rate range for data reception". · Corrected the value in the table.
941, 942		6.1. Serial Control Register (SCR) · Added <Notes> of [bit15]. · Corrected the description of [bit12]. (SSR:LER, FRE, ORE → SSR:FRE, ORE)
945, 946		6.2. Serial Mode Register (SMR) · Corrected the summaries. · Corrected the description of [bit2]. (Unused bit → Reserved bit)
958		6.7. FIFO Control Register 1 (FCR1) · Added <Note> to [bit11].
963		6.8. FIFO Control Register 0 (FCR0) · Corrected the description of [bit1] and [bit0].
965		6.9. FIFO Byte Register (FBYTE) · Added <Notes>.
968	CHAPTER 19-5 I ² C Interface	1. Overview of I ² C Interface (I ² C Communications Control Interface) · 128 bytes → 128 × 9 bits
1028		5.1. I ² C Bus Control Register (IBCR) · Added <Notes>.
1040		5.5. Receive Data Register/Transmit Data Register (RDR/TDR) · Corrected the description of "■ Receive Data Register (RDR)".
1046		5.9. FIFO Control Register 1 (FCR1) · Added <Note> to [bit11].
1051, 1052		5.10. FIFO Control Register 0 (FCR0) · Corrected the description of [bit1] and [bit0].
1054		5.11. FIFO Byte Register (FBYTE) · Corrected <Notes>.
1057		2. Configuration and Block Diagram · Corrected Figure 2-1.
1064	CHAPTER 20-1 USB Clock Generation	5.3. 5.3. USB-PLL Control Register-2 (UPCR2) · Corrected the description of [bit2:0].
1137		1. Overview of USB host · Corrected Table 1-1. (Added missed lines in the table)
1190	CHAPTER 21-1 CAN Prescaler	2.1. CAN Prescaler Register (CANPRE) · Corrected the initial value of "■ Register configuration". (bit3, 1, 0 : 0 → 1)

Page	Section	Change Results
1319	APPENDIXES 1. Register Map	FLASH_IF <ul style="list-style-type: none"> Corrected the initial value of FSTR(0x008). (-----00X → -----0X) Added CRTRMM(0x100).
1320		Clock/Reset <ul style="list-style-type: none"> Corrected the initial value of RST_STR(0x00C). (-----0 00000-01 → -----0 0000--01)
1321		SW WDT <ul style="list-style-type: none"> Corrected the initial value of WdogLock(0xC00). (00000000 00000000 00000000 00000001 → 00000000 00000000 00000000 00000000)
1325		PPG <ul style="list-style-type: none"> Corrected "Base_Address + Address". (0x038 - 0x03F → 0x038 - 0x0FF, 0x108 - 0x13F → 0x108 - 0x1FF)
1327		Corrected the whole description of Base Timer I/O Select Function.
1328		QPRC <ul style="list-style-type: none"> Corrected the initial value of QICRH(0x0014). (--111111 → --000000)
1329		10bit A/DC <ul style="list-style-type: none"> Added "10bit" to the register name.
1330		12bit A/DC <ul style="list-style-type: none"> Added a new section. CR Trim Corrected configuration and initial value of MCR_FTRM(0x004). (0111111 → -----01 1000000)
1331		EXTI <ul style="list-style-type: none"> Corrected the initial value of EIRR(0x0004). 00000000 00000000 → XXXXXXXX XXXXXXXX
1332, 1333		INT-Req. READ <ul style="list-style-type: none"> IRQMON0 → EXC02MON IRQMON1 to 48 → IRQ00MON to IRQ47MON
1334, 1335		GPIO <ul style="list-style-type: none"> Added PFR8(0x020), DDR8(0x220), PDIR8(0x320), PDOR8(0x420) and SPSR(0x580).
1336		LVD <ul style="list-style-type: none"> Added LVD_STR2(0x010).
1337		CAN_Prescaler <ul style="list-style-type: none"> Corrected "Base_Address + Address". (0x04E → 0x000) Corrected the initial value of CANPRE(0x000). (----0000 → ----1011)
1340		USB ch0 <ul style="list-style-type: none"> Corrected configuration, access unit and initial value of UDCC(0x2120). (B → B,H,W 10100-00 → ----- 10100-00)

In the previous revision, the number at the upper-right of the page is CM91-10101-2.

CHAPTER: System Overview

This chapter explains this series system overview.

1. Bus Architecture
2. Memory Architecture
3. Cortex-M3 Architecture
4. Mode

1. Bus Architecture

This chapter explains this series bus architecture.

For this series bus, AHB Bus Matrix circuit actualizes a multi-layer bus. Master and slave architectures are shown below:

● Master

- Cortex-M3 CPU(I-code Bus, D-code Bus, System Bus)
- DMAC

● Slave

- Internal Flash Memory
- Internal SRAM(Code SRAM, On-chip SRAM)
- External Bus
- USB ch0/1
- AHB-AHB Bus Bridge
- AHB-APB Bus Bridge (APB0, 1, 2)

See Figure 1-1 for the bus block diagram.

■ Features

● RAM Architecture

For this series SRAM layout, bus is divided into code area and on-chip area. This allows for preventing conflicts to RAM by multiple bus masters such as CPU and DMAC and allows for improving the performance.

Also, because the divided RAM address areas are serial, RAM area can be utilized to the maximum extent.

● APB Extension Bus

APB1 and APB2 Peripheral Buses are APB extension bus that the following functions are originally added based on AMBA3.0. (APB0 is not included.)

· Supporting Halfword (16 bits) and Byte(8 bits) Accesses

For supported registers, halfword access and byte access are enabled.
See "Register Map" for the supported registers.

· Adding Read-Modify-Write (RMW) Signal

HMASTLOCK signal in bit-band operations is used to generate.

RMW signal is a signal added to prevent that an unrelated flag is cleared mistakenly in read-modify-write process of bit-band operations.

The corresponding flag reads "1" in read during the read-modify-write process and is designed to ignore "1" write.

This prevents any unrelated flag from being mistakenly cleared in the next write when the flag is set immediately after the read in the sequence from read to modify to write.

For the corresponding flags and registers, it is described that "regardless of bit values, "1" can be read in "Read-Modify-Write.""

<Notes>

- Bit-band operation must not be performed to a register which RMW is prohibited.
- When Read-Modify-Write process is performed over the software without bit-band operation, RMW signal is not output.
Therefore, in this case, the flag value can be read in read operation although a register supports RMW process, and it is necessary not to be cleared an unrelated flag mistakenly in write operation.
- For the details of bit-band operations, see the "Cortex-M3 Technical Reference Manual".

● Priority Level

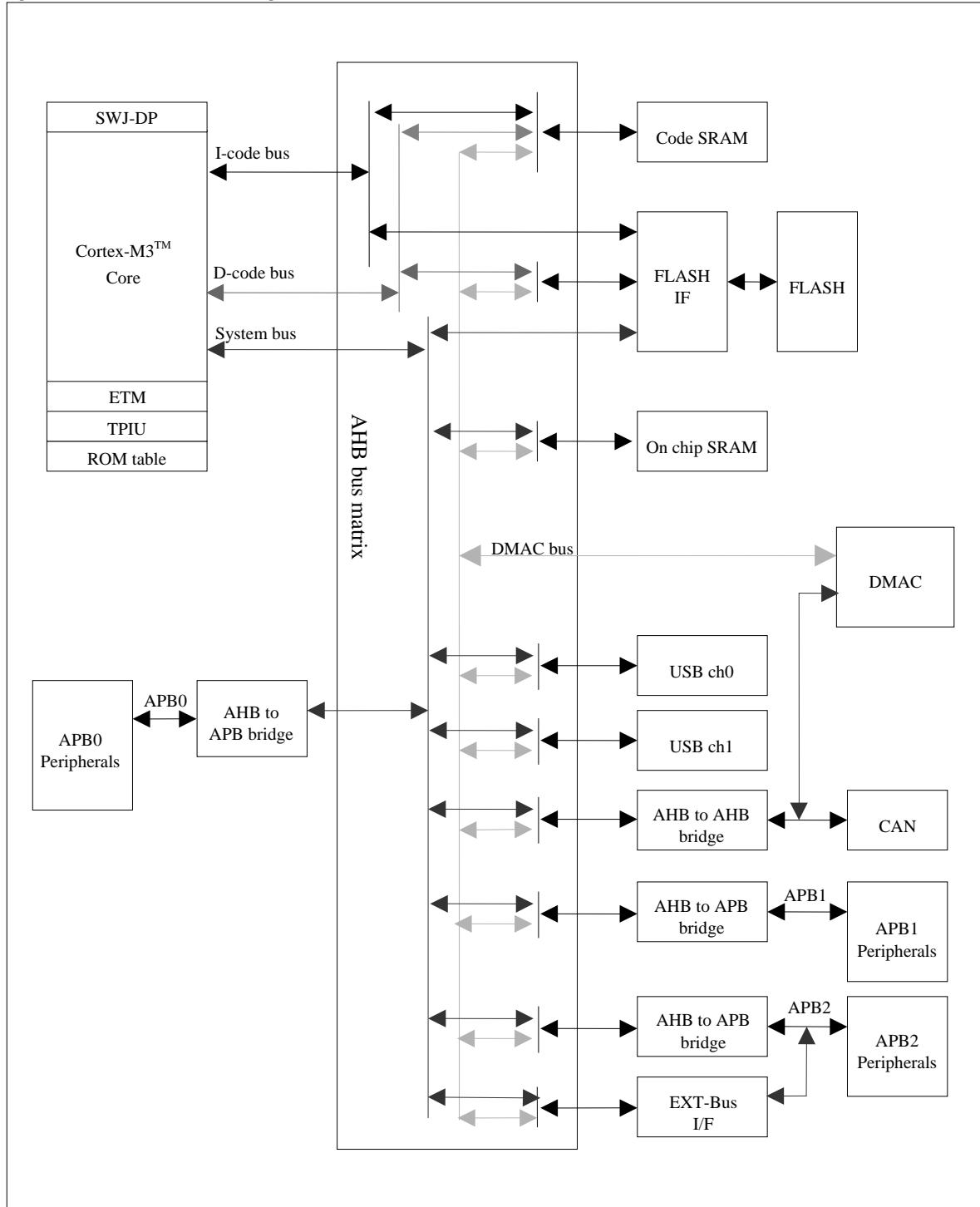
This series sets the bus right as "DMAC>CPU."

According to DMAC access settings such as a case where DMAC is always accessed by a burst transfer, access to CPU may be controlled. Please pay extra attention to DMAC transfer settings.

1.1. Bus Block Diagram

Figure 1-1 illustrates this series bus block diagram.

Figure 1-1 Bus Block Diagram



<Note>

There are some areas which no DMAC transfer can be performed. For details, see the DMAC Transfer column in Table 2-1.

2. Memory Architecture

This chapter shows this series memory architecture.

For this series, 4G-byte address space is available.

Maximum 1M-byte FLASH area, maximum 512K-byte on-chip SRAM area, and maximum 512K-byte code SRAM area are defined.

Also, as an external bus area, 2G-byte area from 0x60000000 to 0xDFFFFFFF is defined. An external memory device can be connected to this area.

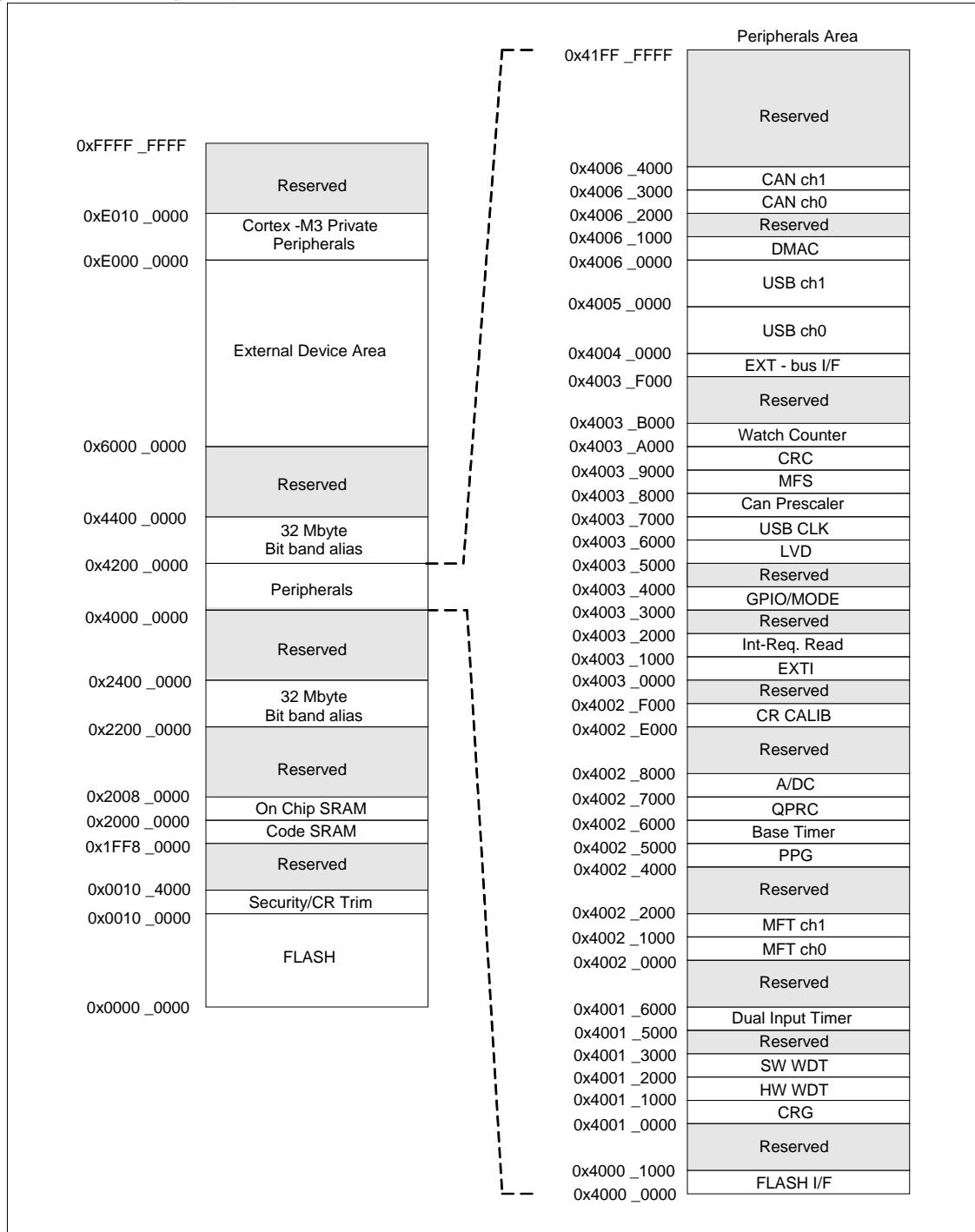
Clause 2.1 illustrates the memory map, and Clause 2.2 illustrates the peripheral memory map.

For the details of Cortex-M3 private peripheral area and bit-band area shown in Figure 2-1, see "Cortex-M3 Technical Reference Manual".

2.1. Memory Map

Figure 2-1 illustrates this series memory map.

Figure 2-1 Memory Map



<Notes>

- Do not access to reserved area.
- For the details of flash memory, see "Flash Programming Manual".

2.2. Peripheral Address Map

Table 2-1 shows this series peripheral address map.

Table 2-1 Peripheral Address Map

Start Address	End Address	Bus	DMAC Transfer	Peripheral	Register Map	Details
0x4000_0000	0x4000_0FFF	AHB	Disabled	FLASH IF Register	FLASH_IF	*1
0x4000_1000	0x4000_FFFF			Reserved	-	-
0x4001_0000	0x4001_0FFF	APB0	Disabled	Clock and Reset Control	Clock / Reset	Chapter 2 Chapter 3 Chapter 5 Chapter 10
0x4001_1000	0x4001_1FFF			Hardware Watchdog Timer	HWWDT	Chapter 11
0x4001_2000	0x4001_2FFF			Software Watchdog Timer	SWWDT	Chapter 11
0x4001_3000	0x4001_4FFF			Reserved	-	-
0x4001_5000	0x4001_5FFF			Dual Timer	Dual_Timer	Chapter 12
0x4001_6000	0x4001_FFFF			Reserved	-	-
0x4002_0000	0x4002_0FFF	APB1	Enabled	Multi-function Timer unit0	MFT	Chapter 15
0x4002_1000	0x4002_1FFF			Multi-function Timer unit1	MFT	Chapter 15
0x4002_2000	0x4002_3FFF			Reserved	-	-
0x4002_4000	0x4002_4FFF			PPG	PPG	Chapter 16
0x4002_5000	0x4002_5FFF			Base Timer	Base Timer Base Timer Selector	Chapter 14-1 Chapter 14-2
0x4002_6000	0x4002_6FFF			QPRC	QPRC	Chapter 17
0x4002_7000	0x4002_7FFF			A/D Converter	A/DC	Chapter 18-2 Chapter 18-3
0x4002_8000	0x4002_DFFF			Reserved	-	-
0x4002_E000	0x4002_EFFF			Internal CR Trimming	CR Trim	Chapter 2
0x4002_F000	0x4002_FFFF			Reserved	-	-

Start Address	End Address	Bus	DMAC Transfer	Peripheral	Register Map	Details
0x4003_0000	0x4003_0FFF	APB2 Enabled		External Interrupt	EXTI	Chapter 7
0x4003_1000	0x4003_1FFF			Interrupt Source Check Register	INT-Req READ	Chapter 6
0x4003_2000	0x4003_2FFF			Reserved	-	-
0x4003_3000	0x4003_3FFF			GPIO	GPIO	Chapter 9
0x4003_4000	0x4003_4FFF			Reserved	-	-
0x4003_5000	0x4003_5FFF			Low Voltage Detection	LVD	Chapter 4
0x4003_6000	0x4003_6FFF			USB Clock Generation Circuit	USB Clock	Chapter 20-1
0x4003_7000	0x4003_7FFF			CAN Pre-scaler	CAN Prescaler	Chapter 21-1
0x4003_8000	0x4003_8FFF			Multi-function Serial	MFS	Chapter 19-2 Chapter 19-3 Chapter 19-4 Chapter 19-5
0x4003_9000	0x4003_9FFF			CRC	CRC	Chapter 22
0x4003_A000	0x4003_AFFF			Watch Counter	Watch Counter	Chapter 13-1 Chapter 13-2
0x4003_B000	0x4003_EFFF			Reserved	-	-
0x4003_F000	0x4003_FFFF			External Bus I/F	EXT-Bus I/F	Chapter 23
0x4004_0000	0x4004_FFFF	AHB Enabled		USB ch0	USB	Chapter 20-2 Chapter 20-3
0x4005_0000	0x4005_FFFF			USB ch1	USB	Chapter 20-2 Chapter 20-3
0x4006_0000	0x4006_0FFF			DMAC Register	DMAC	Chapter 8
0x4006_1000	0x4006_1FFF			Reserved	-	-
0x4006_2000	0x4006_2FFF			CAN ch0	CAN	Chapter 21-2
0x4006_3000	0x4006_3FFF			CAN ch1	CAN	Chapter 21-2
0x4006_4000	0x41FF_FFFF			Reserved	-	-

*1 : For the details of "FLASH IF Register", see "Flash Programming Manual".

3. Cortex-M3 Architecture

This chapter explains the core architecture used in this series.

Cortex-M3 core block architecture used in this series is as follows:

- Cortex-M3 Core
- NVIC
- MPU
- DWT
- ITM
- FPB
- ETM
- SWJ-DP
- TPIU
- ROM Table

■ Cortex-M3 Core

High-performance 32-bit processor core (ARM Cortex-M3 core) is equipped with this series.

This peripheral manual does not describe the details of Cortex-M3 core.

For the details, see "Cortex-M3 Technical Reference Manual".

- Cortex-M3 Core Version
 - For the version of Cortex-M3 core, See "Data sheet".

■ NVIC(Nested Vectored Interrupt Controller)

For this series, 1 NMI(non-maskable interrupt) and 48 peripheral interrupts (IRQ0 to IRQ47)*1 can be used.

Also, interrupt priority register (from 0xE000E400) is comprised of 4 bits, and 16 interrupt priority levels can be configured.

For the details of peripheral interrupts, see another chapter "Interrupt", and for NMI operations, see also another chapter "External Interrupt and NMI Control Block".

NMI terminal is assigned for a combined use with a general-purpose port. Its default value after a reset release is set to the general-purpose port, and NMI input is masked.

When NMI is used, enable NMI in the port setting.

For the details, see another chapter "I/O Port".

*1 : "Cortex-M3 Technical Reference Manual" defines an exception type: IRQ as an external interrupt.
In this peripheral manual, to distinguish from an interrupt by an external terminal "External Interrupt and NMI Control Block," the exception type: IRQ is indicated as a peripheral interrupt.

- **SysTick Timer**

SysTick Timer is a system timer for OS task management integrated into NVIC.

This series generates STCLK through dividing HCLK by eight and sets the values of SysTick Calibration Value Register (0xE000E01C) as shown below:

[Bit 31] : NOREF = 0
[Bit 30] : SKEW = 1
[Bit 23:0] : TENMS = 0x0186A0 (100000)*1

*1 : TENMS value is set to a value which becomes 10ms when 1/8 clock of HCLK is input to STCLK and that HCLK is in 80MHz (10MHz in 1/8 case).

The value of TENMS is not always 10ms because HCLK can be changed to another frequency in the clock control block. Therefore, it is required to calculate an appropriate interrupt timing according to HCLK frequency.

■ **DWT(Data Watchpoint & Trace Unit)**

This series is equipped with DWT to use as the debug function.

DWT contains four comparators, and each comparator can be set as a hardware watchpoint.

■ **ITM(Instrumentation Trace Macrocell)**

This series is equipped with ITM as a debug function.

ITM is an optional application driven trace source that supports printf style debugging. The operation system (OS) and application event are traced, and the system diagnostic information is sent.

■ **FPB(Flash Patch & Breakpoint)**

FPB has the following functions:

- Hardware Breakpoint function
- The function of remapping from Code memory space (FLASH) to System space (On-chip SRAM).

FPB is equipped with six instruction comparators and two literal comparators.

■ **MPU(Memory Protection Unit)**

This series is equipped with a Cortex-M3 optional component MPU, and maximum eight areas can be defined.

■ **ETM(Embedded Trace Macrocell)**

This series is equipped with a Cortex-M3 optional component ETM to support instruction trace.

■ **SWJ-DP**

This series is equipped with SWJ-DP to support both serial wire protocol and JTAG protocol.

■ **TPIU(Trace Port Interface Unit)**

ETM/ITM trace information is output via TPIU.

■ **ROM Table**

ROM table provides the address information of a debug component to an external debug tool.

4. Mode

This chapter explains operating modes.

In this product line, the following operating modes can be used:

- User Mode
Internal ROM(Flash) Startup : CPU obtains a reset vector from Flash and starts operations.
- Serial Writer Mode
Flash serial write is enabled.
* : For the details of this mode, see "Flash Programming Manual".

Operating modes are determined after a release of respective power-on reset, low voltage detection reset, and INITX pin input reset.

* : For the details of power consumption control and clock selection modes, see other chapters "Low-power Mode" and "Clock".

4.1. How to Set Operating Mode

Operating modes are configured by MD pins' (MD1 and MD0) inputs.

MD Pin		Mode
MD1	MD0	
0	0	User Mode Internal ROM(Flash) Startup
0	1	Serial Writer Mode
1	0	Configuration Prohibited
1	1	Configuration Prohibited

4.2. Startup Sequence

Processes to determine operating modes in the startup sequence are shown below:

1. MD Terminal Sampling
2. Determining Operating Mode and Retaining Mode Data

The descriptions of these processes are as follows:

1. MD Terminal Sampling
Operating mode is configured by MD terminal inputs (MD1, MD0). These inputs are sampled by power-on reset, low-voltage detection reset, and INITX pin input reset.
Until each reset, which is the sampling factor, is released, MD1 and MD0 terminal inputs need to be determined.
2. Determining Operating Mode and Retaining Mode Data
MD1 and MD0 which are sampled by respective resets are retained until respective resets are input again.
Operating modes are determined by the retained MD1 and MD0. Therefore, even MD1 and MD0 are changed after a reset is released, it does not affect an operating mode.

Chapter: Clock

This chapter explains the operating clock.

1. Clock Generation Unit Overview
2. Clock Generation Unit Configuration/Block Diagram
3. Clock Generation Unit Operations
4. Clock Setup Procedure Examples
5. Clock Generation Unit Register List
6. Clock Generation Unit Usage Precautions

1. Clock Generation Unit Overview

This section provides an overview of the clock generation unit.

The clock generation unit generates various types of clocks used to operate the MCU.

Source clock is the generic name for external and internal oscillation clocks of this MCU.

The following five types of clocks are source clocks:

- Main clock (CLKMO)
- Sub clock (CLKSO)
- High-speed CR clock (CLKHC)
- Low-speed CR clock (CLKLC)
- PLL clock (CLKPLL)

Select one from the source clock. In this chapter, the selected clock is referred to as the master clock. The master clock is a source of internal bus clocks used to operate this MCU.

Dividing the master clock frequency can generate a base clock. In addition, dividing the base clock can generate each bus clock.

In this chapter, the base clock and bus clocks are referred to as internal bus clocks. The following five types of clocks are internal bus clocks:

- Base clock (FCLK/HCLK)
- APB0 bus clock (PCLK0)
- APB1 bus clock (PCLK1)
- APB2 bus clock (PCLK2)
- TRACE clock (TPIUCLK)

In addition to source clocks, the master clock, and internal bus clocks, the following clocks are provided:

- USB-PLL clock
- CAN prescaler clock
- Software watchdog timer count clock

The following shows the features of the clock generation unit.

- It can be set the oscillation stabilization wait time of the main clock (CLKMO).
- It can be set the interrupt which generates at completing the oscillation stabilization wait time of the main clock (CLKMO).
- It can be set the oscillation stabilization wait time of the sub clock (CLKSO).
- It can be set the interrupt which generates at completing the oscillation stabilization wait time of the sub clock (CLKSO).
- It can be set the oscillation stabilization wait time of the PLL clock (CLKPLL).
- It can be set the interrupt which generates at completing the oscillation stabilization wait time of the PLL clock (CLKPLL).
- It can be set the PLL multiplication ratio.
- It can be selected the master clock.
- It can be set the frequency division ratio of each internal bus clock frequency.
- It can be selected run or stop of the APB1 and APB2 bus clocks.
- It can be set the frequency division ratio of the software watchdog timer count clock frequency.
- It can be set run/stop of the software watchdog timer count clock.
- It can be set the software watchdog timer count operation in debug mode.
- It includes registers for enabling clock-related interrupts, checking interrupt status, and clearing interrupt causes.

2. Clock Generation Unit Configuration/Block Diagram

This section explains configuration of the clock generation unit.

■ Source clock

Source clock is the generic name for external and internal oscillation clocks of this MCU. The following five types of clocks are source clocks:

- **Main clock (CLKMO)**

CLKMO is generated by connecting a crystal oscillator to the main oscillation pins (X0, X1), or input using an external clock.

- **Sub clock (CLKSO)**

CLKSO is generated by connecting a crystal oscillation to the sub oscillator pins (X0A, X1A), or input using an external clock.

- **High-speed CR clock (CLKHC)**

CLKHC is an output clock for the internal high-speed CR oscillator.

- **Low-speed CR clock (CLKLC)**

CLKLC is an output clock for the internal low-speed CR oscillator.

- **PLL clock (CLKPLL)**

CLKPLL is generated by multiplying an oscillation clock using the PLL Clock Multiplication Circuit (PLL Oscillation Circuit).

■ Master clock

The signal selected from source clocks are referred to as the master clock.

The master clock is a source for all bus clocks.

■ Internal bus clocks

The following signals are bus clocks generated internally.

- **Base clock (HCLK/FCLK)**

HCLK and FCLK are collectively called the base clock. Both FCLK and HCLK are supplied to the CPU.
HCLK is a clock for macro connected to the AHB bus.

The clock frequency can be set to between 1/1 and 1/16 frequency of the master clock.

This clock stops in timer mode or stop mode.

In sleep mode, the CPU stops the supply of HCLK while continuing the supply of FCLK.

- **APB0 bus clock (PCLK0)**

PCLK0 is a clock for peripheral macro connected to the APB0 bus.

The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.

This clock stops in timer mode or stop mode.

- **APB1 bus clock (PCLK1)**

PCLK1 is a clock for peripheral macro connected to the APB1 bus.

The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.

This clock stops in timer mode or stop mode.

The supply of the clock can be also stopped by setting a register.

● APB2 bus clock (PCLK2)

PCLK2 is a clock for peripheral macro connected to the APB2 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode or stop mode.
The supply of the clock can be also stopped by setting a register.

● TPIU clock (TPIUCLK)

TPIUCLK is a clock for TRACE.
The clock frequency can be set to between 1/1 and 1/2 frequency of the base clock.
This clock stops in timer mode or stop mode.
This clock output is enabled only for the products equipped with ETM.

■ Clocks other than source clocks and internal bus clocks

● USB-PLL clock

This clock runs at 48 MHz, used for operating USB.
The PLL clock is generated by setting the USB-PLL oscillator.
This clock stops in timer mode or stop mode.
This clock can be set the frequency independently without depending on the frequency of master clock setting.

For USB-PLL operation settings, see Chapter "USB clock generation".

● CAN prescaler clock

This clock is the same clock as CLKPLL, used for CAN prescaler.
The frequency division used for the clock must be configured on the prescaler side.
This clock stops in stop mode.
The supply of the clock can be also stopped by setting a register.

For operation settings of CAN prescaler, see Chapter "CAN Prescaler".

● Software watchdog timer counter clock (SWDOGCLK)

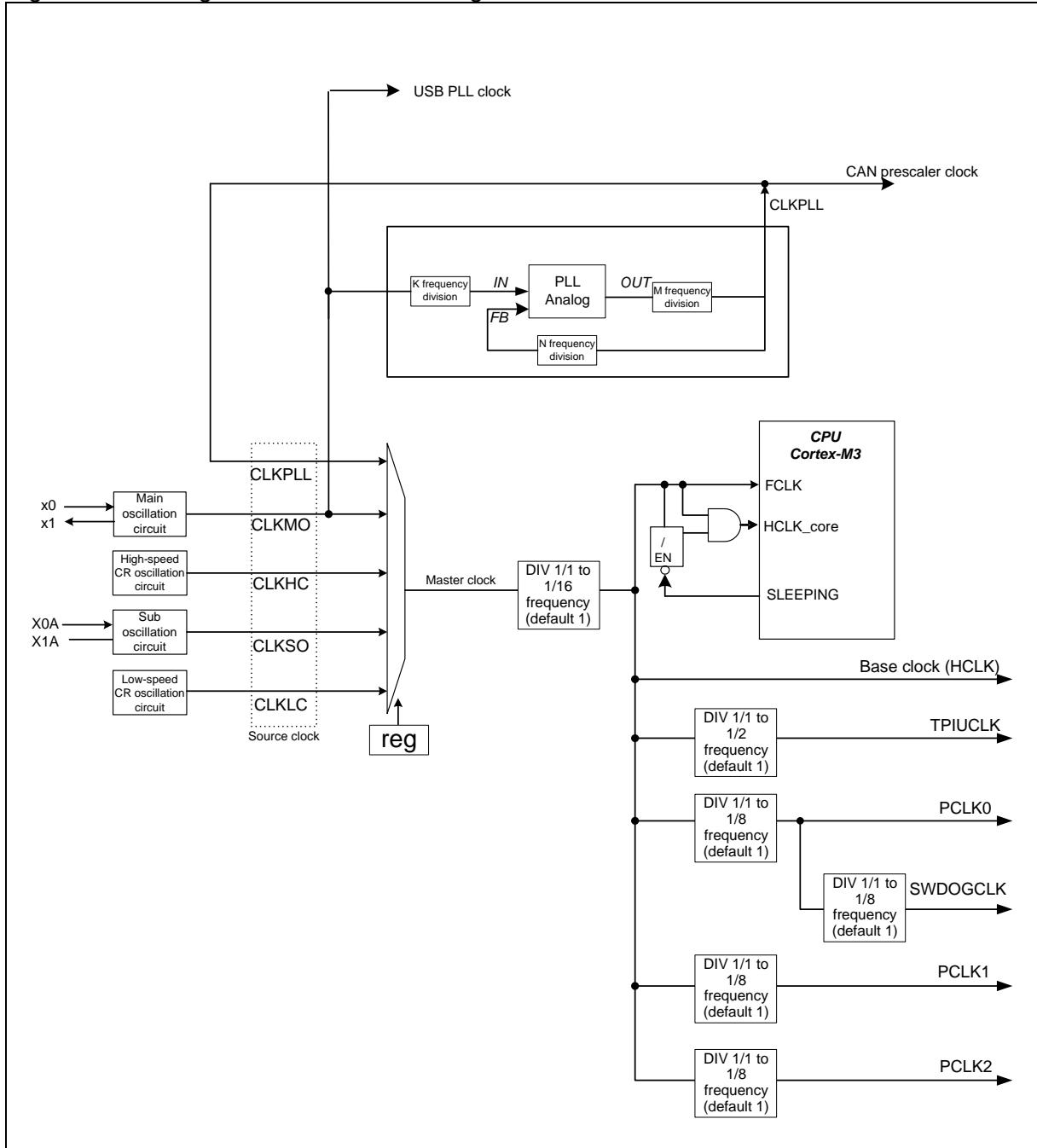
SWDOGCLK is a clock for the software watchdog timer connected to the APB0 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the APB0 bus clock.
This clock stops in timer mode or stop mode.

For operation settings of the software watchdog timer, see Chapter "Watchdog Timer".

■ Block diagram

Figure 2-1 shows the block diagram of the clock generation unit.

Figure 2-1 Clock generation unit block diagram



3. Clock Generation Unit Operations

This section explains the clock generation unit.

3.1. Selecting the clock mode

■ Definition of clock mode (selecting the master clock)

The MCU clock mode is defined by the source clock selected by the system clock mode control register. Five types of clock modes are provided: Main clock mode, sub clock mode, high-speed CR clock mode, low-speed CR clock mode, and PLL clock mode.

● Main clock mode

In main clock mode, the main clock (CLKMO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Status of the PLL clock (CLKPLL) differs depending on the setting of the PLLE bit in the System Clock Mode Control Register (SCM_CTL), and the sub clock (CLKSO) depends on the SOSCE bit in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

● Sub-clock mode

In sub clock mode, the sub clock (CLKSO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

The main clock (CLKMO), high-speed CR clock (CLKHC), and PLL clock (CLKPLL) are stopped by hardware. The low-speed CR clock (CLKLC) cannot be stopped by user program.

● High-speed CR clock mode

In high-speed CR clock mode, the internal high-speed CR clock (CLKHC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Statuses of the main clock (CLKMO), PLL clock (CLKPLL), and sub clock (CLKSO) differ depending on the settings of MOSCE, PLLE, and SOSCE bits in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

● Low-speed CR clock mode

In low-speed CR clock mode, the internal low-speed CR clock (CLKLC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

In low-speed CR clock mode, the main clock (CLKMO), high-speed CR clock (CLKHC), and PLL clock (CLKPLL) are stopped by hardware. Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL).

● PLL clock mode

In PLL clock mode, the PLL clock (CLKPLL) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

3.2. Internal bus clock frequency division control

This section explains the internal bus clock frequency division.

Frequency division ratio vs. the base clock can be set independently for each internal bus clock.
This function can set the operating frequency optimized for each circuit.

Table 3-1 shows the internal bus clock.

The following five types of internal bus clock frequency divisions can be selected.

Table 3-1 Internal bus clock list

	Internal bus clock name	Source clock	Maximum operating frequency
1	Base clock (HCLK/FCLK)	1/1 to 1/16 frequency of the master clock	80 MHz
2	APB0 bus clock (PCLK0)	1/1 to 1/8 frequency of the base clock	40 MHz
3	APB1 bus clock (PCLK1)	1/1 to 1/8 frequency of the base clock	40 MHz
4	APB2 bus clock (PCLK2)	1/1 to 1/8 frequency of the base clock	40 MHz
5	TRACE clock (TPIUCLK)	1/1 to 1/2 frequency of the base clock	80 MHz

To set the frequency division ratio of internal bus clocks, use the Base Clock Prescaler Register (BSC_PSR), APB0 Prescaler Register (APBC0_PSR), APB1 Prescaler Register (APBC1_PSR), APB2 Prescaler Register (APBC2_PSR), and Trace Clock Prescaler Register (TTC_PSR). For details on each register, see "5. Clock Generation Unit Register List".

■ Setting the bus clock frequency division

- The set frequency division ratio is not cleared by a software reset. The latest value is retained even after the software reset.
- The value is initialized by a reset other than software resets.
Before changing the initially set master clock to a faster source clock, be sure to set the frequency division ratio.
- If a combined value of master clock, PLL multiplication, and frequency division ratio settings exceeds the maximum operating frequency of each internal bus, the operation corresponding to the setting is not guaranteed.

3.3. PLL clock control

This section explains the PLL clock control.

The PLL Clock Control Circuit is used to generate the main clock. The PLL Oscillation Circuit can enable/disable operation (oscillation), select the input clock, set the stabilization wait time, and set the multiplication.

■ PLL operation

The following explains operation of the PLL clock.

- Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR).
 - Selecting the PLL input clock
 - Setting the PLL clock stabilization wait time
- The "PLL oscillation enable bit" of the System Clock Mode Control Register (SCM_CTL) must be enabled to let the PLL Circuit start oscillating.
- When the PLL clock stabilization wait time has elapsed, and the "PLL oscillation stable bit" of the System Clock Mode Status Register (SCM_STR) indicates a stable state, the preparation for transition to PLL clock mode completes.
- The "Master clock switch control bit" of the System Clock Mode Control Register (SCM_CTL) must be set to PLL clock mode to change to PLL clock mode.

■ Setting the PLL clock oscillation stabilization wait time

The details are given in "5.10 PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)".

<Notes>

- For block diagram of the PLL Clock Control Circuit, see "2.Clock Generation Unit Configuration/Block Diagram".
- For the order of frequency division settings for each internal bus clock, see "4 Clock Setup Procedure Examples".
- For the oscillation stabilization wait time, see "3.4 Oscillation stabilization wait time".
- Only the main oscillation can be selected for the PLL input clock.

■ Setting the multiplication ratio to generate the PLL clock

Each frequency division clock in the PLL Multiplication Circuit must be set using PLL Control Register 1 (PLL_CTL1) and PLL Control Register 2 (PLL_CTL2). The following Table 3-2 provides example frequency division settings.

Table 3-2 Example PLL multiplication ratio settings

Input clock	K	PLLlin	N	PLLout	M	CLKPLL
4MHz	1	4MHz	20	80MHz	1	80MHz
4MHz	1	4MHz	15	60MHz	1	60MHz
4MHz	1	4MHz	15	120MHz	2	60MHz
5MHz	1	5MHz	16	80MHz	1	80MHz
5MHz	1	5MHz	12	60MHz	1	60MHz
5MHz	1	5MHz	12	120MHz	2	60MHz
6MHz	1	6MHz	10	60MHz	1	60MHz
6MHz	1	6MHz	10	120MHz	2	60MHz
8MHz	1	8MHz	10	80MHz	1	80MHz
10MHz	1	10MHz	8	80MHz	1	80MHz
10MHz	1	10MHz	6	60MHz	1	60MHz
10MHz	1	10MHz	6	120MHz	2	60MHz
12MHz	1	12MHz	5	60MHz	1	60MHz
12MHz	1	12MHz	5	120MHz	2	60MHz
15MHz	1	15MHz	4	60MHz	1	60MHz
16MHz	1	16MHz	5	80MHz	1	80MHz
20MHz	1	20MHz	4	80MHz	1	80MHz
30MHz	1	30MHz	2	120MHz	2	60MHz
40MHz	2	20MHz	4	80MHz	1	80MHz
48MHz	3	16MHz	5	80MHz	1	80MHz
48MHz	4	12MHz	5	60MHz	1	60MHz

<Notes>

- For PLL characteristics, see "Data Sheet".
- Set the PLLin within the value "PLL input clock frequency: f_{PLL1} " shown in the data sheet.
- The value "MxN" is a multiplication ratio for the PLLin. Set this value within the range shown in the "PLL multiple rate" of the data sheet.
- The frequency of the PLLin multiplied by "MxN" becomes PLLout. Set this value within the range shown in the "PLL macro oscillation clock frequency: f_{PLLO} " of the data sheet.
- The value of the PLLout divided by "M" becomes CLKPLL.
- See Figure 2-1 for the configurations of PLL and divider.

3.4. Oscillation stabilization wait time

This section explains the oscillation stabilization wait time.

An oscillation stabilization wait time is required if the source clock is not in a stable operating state. During the oscillation stabilization wait time, internal and external clocks stop the supply, only the internal time counter operates to wait until the stabilization wait time passes, a time value set in the Clock Stabilization Wait Time Register (CSW_TMR) or PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). When the wait time has been passed, the corresponding oscillator is ready to operate, and the clock can be used as a master clock.

■ Setting the oscillation stabilization wait time

- Main clock (CLKMO)
Set the oscillation stabilization wait time of the main clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKHC.
- Sub clock (CLKSO)
Set the oscillation stabilization wait time of the sub clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKLC.
- PLL clock
Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). The set time value is counted by CLKHC.
 - Selecting the PLL input clock
 - Setting the PLL clock stabilization wait time

■ Cause of waiting for oscillation stability

- After the oscillation is enabled via software
If the PLLE, SOSCE, and MOSCE bits of the System Clock Mode Control Register (SCM_CTL) are set to active, each relevant oscillator waits during the oscillation stabilization wait time.
- When returning from stop mode using an external interrupt
The status returns to clock mode, a state before stop mode, using an external clock. During stop mode, all source clocks stop and, therefore, the hardware automatically waits during the oscillation stabilization wait time.
- After PLL operation is enabled
After PLL operation is enabled, the PLL oscillation stabilization wait time is spent.

<Notes>

- Each set value of the oscillation stabilization wait time must be changed before the clock is enabled.
- After software reset, the oscillation stabilization wait time is not applied.
- In the stabilization wait time for main clock, sub clock and PLL clock, the built-in CR counts the clock as set in the Clock Stabilization Wait Time Registers. Stabilization wait time flag will be activated when the counting is complete, so these wait times are independent of each oscillator statuses. The oscillation stabilization wait time may be completed before oscillator stabilization if the setting of the oscillation stabilization wait time is too short.
- As the stabilization wait times for main clock and sub clock oscillators depend on the type of the oscillator (crystal, ceramics, etc.), proper oscillation stabilization wait time must be chosen for the oscillator to be used.
- Set the PLL oscillation stabilization wait time by referring to PLL Clock LOCKUP Time of the electric characteristics described in "Data Sheet".

3.5. Interrupt causes

This section explains interrupt causes relevant to clocks.

The clock generation unit has the following interrupt causes.

■ Interrupt causes

The clock generation unit has the following four types of interrupt causes:

- FCS interrupt (anomalous frequency detection interrupt)
When the FCS (anomalous frequency detection) is enabled, and an anomalous frequency of the main clock is detected, an interrupt occurs.
- PLL clock oscillation stabilization completion interrupt
When the PLL clock oscillation stabilization wait time ends, an interrupt occurs.
- Sub clock oscillation stabilization completion interrupt
When the sub clock oscillation stabilization wait time ends, an interrupt occurs.
- Main clock oscillation stabilization completion interrupt
When the main clock oscillation stabilization wait time ends, an interrupt occurs.

■ Registers

The following three types of registers are provided for each interrupt cause:

- Interrupt Enable Register
Enables/disables each interrupt.
- Interrupt Status Register
Indicates each interrupt status. This register is read-only.
- Interrupt Clear Register
Clears each interrupt cause. This register is write-only.

4. Clock Setup Procedure Examples

This section explains procedure examples of setting up clocks.

■ Setup procedure examples

Figure 4-1 Example clock setup procedure (Power-on -> High-speed CR run mode -> Desired clock mode)

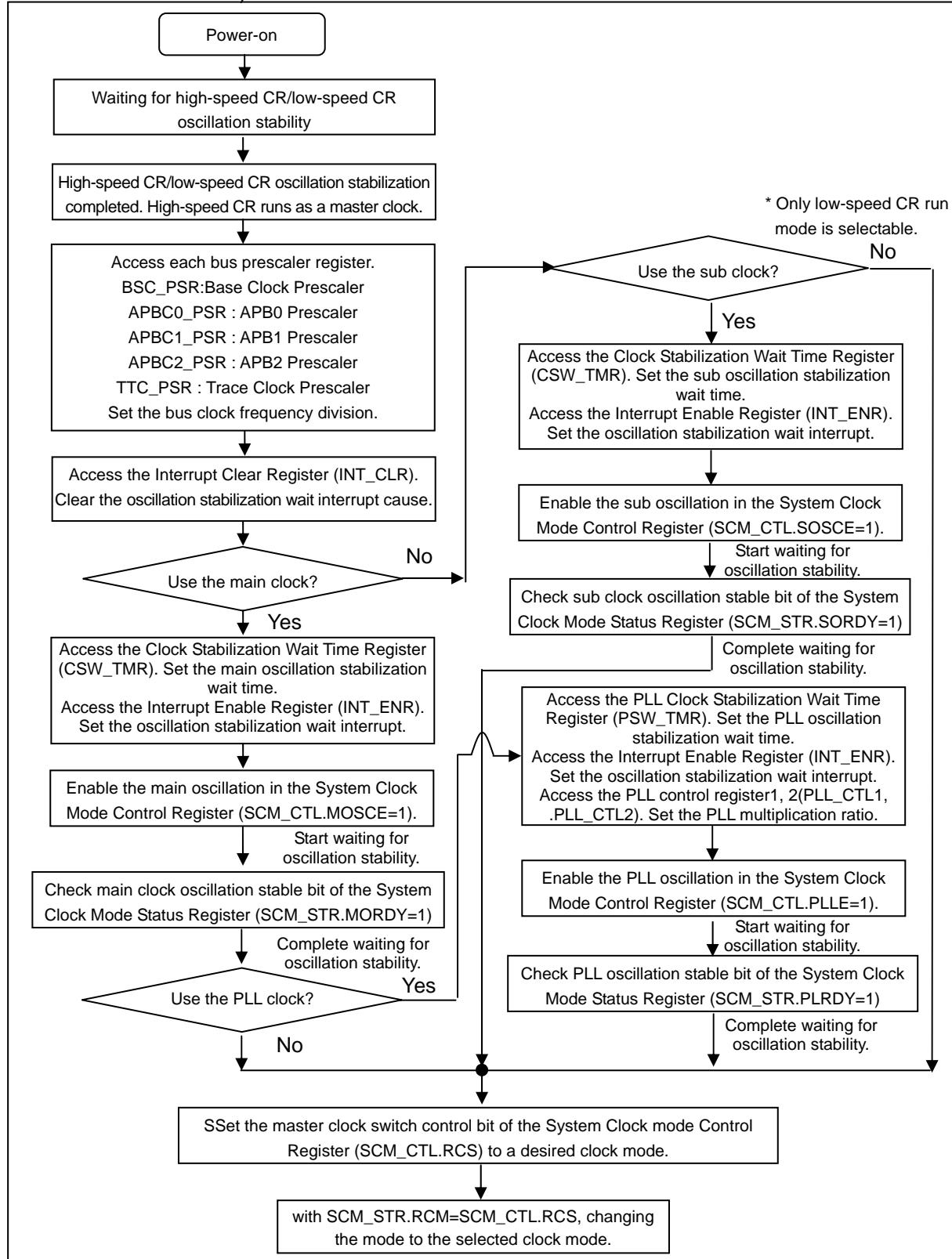
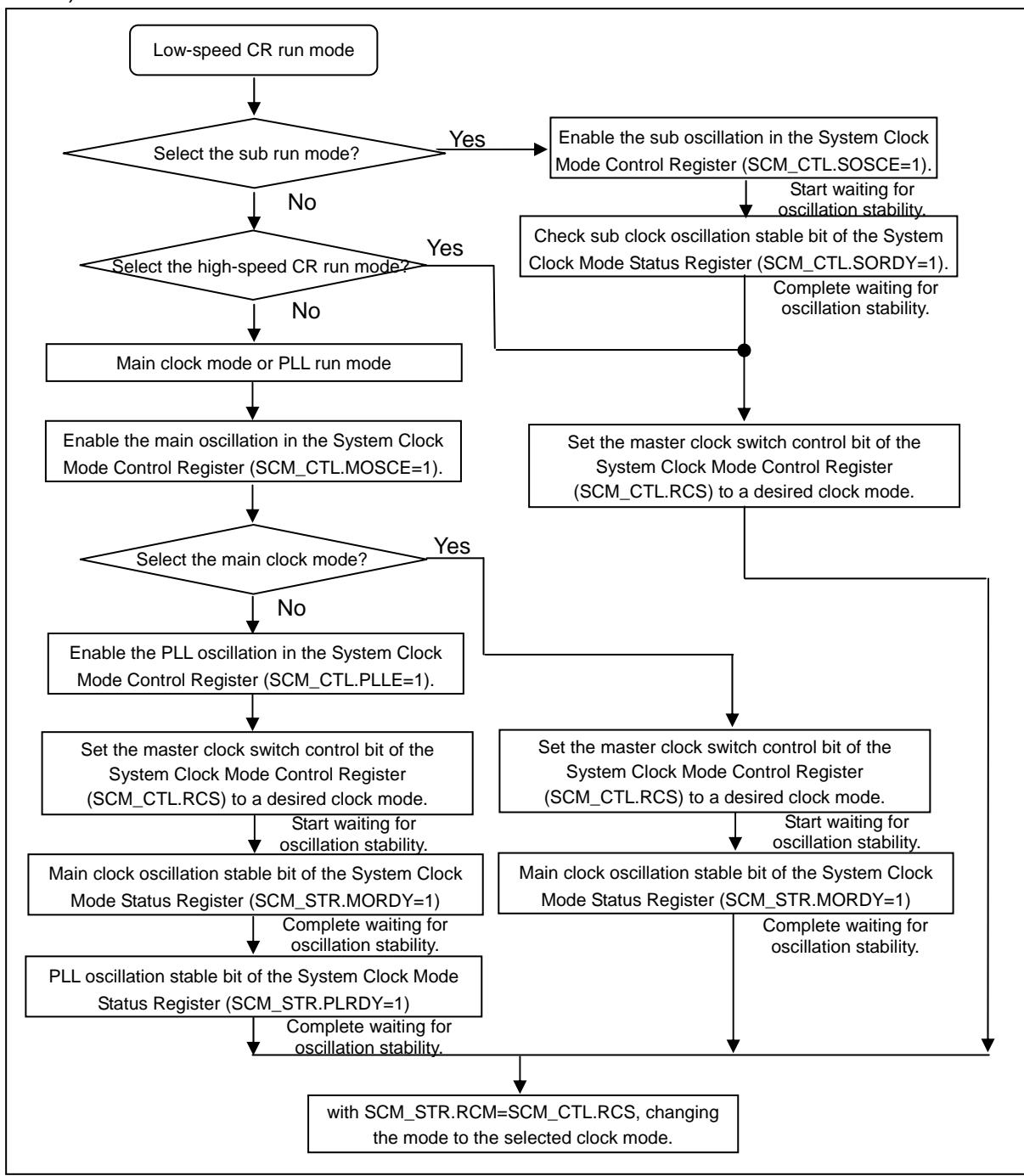


Figure 4-2 Example clock setup procedure (Low-speed CR run mode -> Desired clock run mode)



<Notes>

- Figure 4-2 assumes that settings of the oscillation stabilization wait time, interrupts, PLL multiplication ratio and bus clock frequency division for each clock have been configured previously, and they are omitted from the flowchart.
- In the sub clock mode/sub CR clock mode, the main clock(CLKMO), high-speed CR(CLKHC), PLL clock(CLKPLL) is stopped by hardware. So CLKMO/CLKHC/CLKPLL does not start oscillation only setting oscillation enable bit=1. These oscillations will start by changing the SCM_CTL.RCS bit from sub clock mode/sub CR clock mode with setting oscillation enable bit=1.
- If the main/sub oscillation stabilization wait times are short and the oscillation stabilization wait times run out before oscillators stabilize, reset may be applied by the clock supervisor function.

5. Clock Generation Unit Register List

This section provides clock generation register list.

■ Clock generation unit register list

Abbreviation	Register name	See
SCM_CTL	System Clock Mode Control Register	5.1
SCM_STR	System Clock Mode Status Register	5.2
BSC_PSR	Base Clock Prescaler Register	5.3
APBC0_PSR	APB0 Prescaler Register	5.4
APBC1_PSR	APB1 Prescaler Register	5.5
APBC2_PSR	APB2 Prescaler Register	5.6
SWC_PSR	SW-WDGT Clock Prescaler Register	5.7
TTC_PSR	Trace Clock Prescaler Register	5.8
CSW_TMR	Clock Stabilization Wait Time Register	5.9
PSW_TMR	PLL Clock Stabilization Wait Time Setup Register	5.10
PLL_CTL1	PLL Control Register 1	5.11
PLL_CTL2	PLL Control Register 2	5.12
DBWDT_CTL	Debug Break Watchdog Timer Control Register	5.13
INT_ENR	Interrupt Enable Register	5.14
INT_STR	Interrupt Status Register	5.15
INT_CLR	Interrupt Clear Register	5.16

5.1. System Clock Mode Control Register (SCM_CTL)

The SCM_CTL selects the master clock and enables/disables the clock oscillation.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	RCS[2:0]			PLLE	SOSCE	Reserved	MOSCE	Reserved
Initial value	3'b000			1'b0	1'b0	-	1'b0	-
Attribute	R/W			R/W	R/W	-	R/W	-

■ Register functions

[bit 7:5] RCS: Master clock switch control bits

Bit 7	Bit 6	Bit 5	Description
0	0	0	Internal high-speed CR oscillation clock [Initial value]
0	0	1	Main oscillation clock
0	1	0	PLL oscillation clock
0	1	1	Setting disabled
1	0	0	Internal low-speed CR oscillation clock
1	0	1	Sub oscillation clock
1	1	0	Setting disabled
1	1	1	Setting disabled

[bit 4] PLLE: PLL oscillation enable bit

Bit	Description
0	Disables PLL oscillation [Initial value]
1	Enables PLL oscillation

[bit 3] SOSCE: Sub clock oscillation enable bit

Bit	Description
0	Disables sub oscillation [Initial value]
1	Enables sub oscillation

[bit 2] res: Reserved bit

"0" is read from this bit.

Set this bit to "0" when writing.

[bit 1] MOSCE: Main clock oscillation enable bit.

Bit	Description
0	Disables main oscillation [Initial value]
1	Enables main oscillation

[bit 0] RES: Reserved bit
"0" is read from this bit.
Set this bit to "0" when writing.

<Notes>

- This register is not initialized by software reset.
- When you change the clock mode, you should set the enable bit to transition for desired clock oscillation. Then, you can change the the System Clock Mode Control Register (SCM_CTL.RCS).

5.2. System Clock Mode Status Register (SCM_STR)

The SCM_STR indicates a clock selected for the master clock and a waiting state for clock oscillation stability.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	RCM[2:0]			PLRDY	SORDY	Reserved	MORDY	Reserved
Initial value	3'b000			1'b0	1'b0	-	1'b0	-
Attribute	R			R	R	-	R	-

■ Register functions

[bit 7:5] RCM2 to RCM0: Master clock selection bits

Bit 7	Bit 6	Bit 5	Description
0	0	0	Internal high-speed CR oscillation clock [Initial value]
0	0	1	Main oscillation clock
0	1	0	PLL oscillation clock
0	1	1	Reserve
1	0	0	Internal low-speed CR oscillation clock
1	0	1	Sub oscillation clock
1	1	0	Reserve
1	1	1	Reserve

[bit 4] PLRDY: PLL oscillation stable bit

Bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit 3] SORDY: Sub clock oscillation stable bit

Bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit 2] RES: Reserved bit

"0" is read from this bit.

[bit 1] MORDY: Main clock oscillation stable bit

Bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit 0] RES: Reserved bit

"0" is read from this bit.

<Note>

This register is not initialized by software reset.

5.3. Base Clock Prescaler Register (BSC_PSR)

The BSC_PSR sets the frequency division ratio of the base clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					BSR		
Initial value	-					3'b000		
Attribute	-					R/W		

■ Register functions

[bit 7:3] RES: Reserved bits

"0b00000" is read from these bits.

Set these bits to "0b00000" when writing.

[bit 2:0] BSR: Base clock frequency division ratio setting bit

Bit 2	Bit 1	Bit 0	Description
0	0	0	1/1 [Initial value]
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	0	0	1/6
1	0	1	1/8
1	1	0	1/16
1	1	1	Reserved

<Note>

This register is not initialized by software reset.

5.4. APB0 Prescaler Register (APBC0_PSR)

The APBC0_PSR sets the APB0 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				APBC0			
Initial value	-				2'b00			
Attribute	-				R/W			

■ Register functions

[bit 7:2] RES: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1:0] APBC0: APB0 bus clock frequency division ratio setting bit

Bit 1	Bit 0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.5. APB1 Prescaler Register (APBC1_PSR)

The APBC1_PSR sets the APB1 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	APBC1EN	Reserved		APBC1RST	Reserved		APBC1	
Initial value	1'b1	-		1'b0	-		2'b00	
Attribute	R/W	-		R/W	-		R/W	

■ Register functions

[bit 7] APBC1EN: APB1 clock enable bit

Bit	Description
0	Disables PCLK1 output
1	Enables PCLK1 output [Initial value]

[bit 6:5] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 4] APBC1RST: APB1 bus reset control bit

Bit	Description
0	APB1 bus reset, inactive [Initial value]
1	APB1 bus reset, active

[bit 3:2] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 1:0] APBC1: APB1 bus clock frequency division ratio setting bit

Bit 1	Bit 0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.6. APB2 Prescaler Register (APBC2_PSR)

The APBC2_PSR sets the APB2 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	APBC2EN	Reserved		APBC2RST	Reserved		APBC2	
Initial value	1'b1	-		1'b0	-		2'b00	
Attribute	R/W	-		R/W	-		R/W	

■ Register functions

[bit 7] APBC2EN: APB2 clock enable bit

Bit	Description
0	Disables PCLK2 output
1	Enables PCLK2 output [Initial value]

[bit 6:5] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 4] APBC2RST: APB2 bus reset control bit

Bit	Description
0	APB2 bus reset, inactive [Initial value]
1	APB2 bus reset, active

[bit 3:2] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 1:0] APBC2: APB2 bus clock frequency division ratio setting bit

Bit 1	Bit 0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.7. Software Watchdog Clock Prescaler Register (SWC_PSR)

The SWC_PSR sets the frequency division and enables the output of the software watchdog clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	TESTB			Reserved			SWDS	
Initial value	1'bx			-			2'b00	
Attribute	R/W			-			R/W	

■ Register functions

[bit 7] TESTB: TEST bit

Bit	Description
0	No permitting.
1	Always written by "1"

[bit 7:2] RES: Reserved bits

"0b100000" is read from these bits.

Set these bits to "0b100000" when writing.

[bit 1:0] SWDS: Software watchdog clock frequency division ratio setting bit

Bit 1	Bit 0	Description
0	0	Sets 1/1 frequency of PCLK0. [Initial value]
0	1	Sets 1/2 frequency of PCLK0.
1	0	Sets 1/4 frequency of PCLK0.
1	1	Sets 1/8 frequency of PCLK0.

<Notes>

- This register is not initialized by software reset.
- Be sure to set the bit 7 to "1" when writing a value to this register.

5.8. Trace Clock Prescaler Register (TTC_PSR)

The TTC_PSR sets the trace clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved				TTC
Initial value				-				1'b0
Attribute				-				R/W

■ Register functions

[bit 7:1] RES: Reserved bits

"0b00000000" is read from these bits.

Set these bits to "0b00000000" when writing.

[bit 0] TTC: Trace clock frequency division ratio setting bit

Bit	Description
0	1/1 [Initial value]
1	1/2

<Note>

This register is not initialized by software reset.

5.9. Clock Stabilization Wait Time Register (CSW_TMR)

CSW_TMR sets the oscillation stabilization wait time of the main/sub clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0	
Field	Reserved	SOWT			MOWT				
Initial value	-	3b000				4b0000			
Attribute	-	R/W				R/W			

■ Register functions

[bit 7] RES: Reserved bits

"0b0" is read from this bit.

Set this bit to "0b0" when writing.

[bit 6:4] SOWT: Sub clock stabilization wait time setup bit

Bit 6	Bit 5	Bit 4	Description	
0	0	0	$2^{10}/F_{CL}$: Approx. 10.3 ms * [Initial value]
0	0	1	$2^{11}/F_{CL}$: Approx. 20.5 ms *
0	1	0	$2^{12}/F_{CL}$: Approx. 41 ms *
0	1	1	$2^{13}/F_{CL}$: Approx. 82 ms *
1	0	0	$2^{14}/F_{CL}$: Approx. 164 ms *
1	0	1	$2^{15}/F_{CL}$: Approx. 327 ms *
1	1	0	$2^{16}/F_{CL}$: Approx. 655 ms *
1	1	1	$2^{17}/F_{CL}$: Approx. 1.31 s *

*: If $F_{CL}=100\text{kHz}$

[bit 3:0] MOWT: Main clock stabilization wait time setup bit

Bit 3	Bit 2	Bit 1	Bit 0	Description	
0	0	0	0	$2^1/F_{CH}$: Approx. 500 ns * [Initial value]
0	0	0	1	$2^5/F_{CH}$: Approx. 8 μs *
0	0	1	0	$2^6/F_{CH}$: Approx. 16 μs *
0	0	1	1	$2^7/F_{CH}$: Approx. 32 μs *
0	1	0	0	$2^8/F_{CH}$: Approx. 64 μs *
0	1	0	1	$2^9/F_{CH}$: Approx. 128 μs *
0	1	1	0	$2^{10}/F_{CH}$: Approx. 256 μs *
0	1	1	1	$2^{11}/F_{CH}$: Approx. 512 μs *
1	0	0	0	$2^{12}/F_{CH}$: Approx. 1.0 ms *
1	0	0	1	$2^{13}/F_{CH}$: Approx. 2.0 ms *
1	0	1	0	$2^{14}/F_{CH}$: Approx. 4.0 ms *
1	0	1	1	$2^{15}/F_{CH}$: Approx. 8.0 ms *
1	1	0	0	$2^{17}/F_{CH}$: Approx. 33.0 ms *
1	1	0	1	$2^{19}/F_{CH}$: Approx. 131 ms *
1	1	1	0	$2^{21}/F_{CH}$: Approx. 524 ms *
1	1	1	1	$2^{23}/F_{CH}$: Approx. 2.0 s *

*: If $F_{CH}=4\text{MHz}$

<Notes>

- Set each oscillation stabilization wait time before enabling the oscillation enable bit of the SCM_CTL.
If you change MOWT or SOWT while waiting for oscillation stability of each oscillator, each oscillation stabilization wait time is not guaranteed.
- This register is not initialized by software reset.

5.10. PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)

The PSW_TMR sets the PLL clock stabilization wait time.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			PINC	Reserved	-	POWT	-
Initial value	-			1'b0	-	-	3'b000	-
Attribute	-			R/W	-	-	R/W	-

■ Register functions

[bit 7:5] RES: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit 4] PINC: PLL input clock select bit

Bit	Description
0	Selects CLKMO (main oscillation) [Initial value]
1	Setting disabled

Note: This bit must not be set to "1". Always set it to "0".

[bit 3] RES: Reserved bit

"0b0" is read from this bit.

Set this bit to "0b0" when writing.

[bit 2:0] POWT: PLL clock stabilization wait time setup bit

Bit 2	Bit 1	Bit 0	Description
0	0	0	$2^9 / F_{CH}$: Approx. 128 μ s * [Initial value]
0	0	1	$2^{10} / F_{CH}$: Approx. 256 μ s *
0	1	0	$2^{11} / F_{CH}$: Approx. 512 μ s *
0	1	1	$2^{12} / F_{CH}$: Approx. 1.02 ms *
1	0	0	$2^{13} / F_{CH}$: Approx. 2.05 ms *
1	0	1	$2^{14} / F_{CH}$: Approx. 4.10 ms *
1	1	0	$2^{15} / F_{CH}$: Approx. 8.20 ms *
1	1	1	$2^{16} / F_{CH}$: Approx. 16.40 ms *

*: $F_{CH}=4\text{MHz}$

<Notes>

- Set each oscillation stabilization wait time before enabling the oscillation enable bit of the SCM_CTL. If you change POWT while waiting for oscillation stability of the PLL oscillator, the oscillation stabilization wait time is not guaranteed.
- This register is not initialized by software reset.

5.11. PLL Control Register 1 (PLL_CTL1)

The PLL_CTL1 sets the PLL frequency division ratio.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field		PLLK				PLLM		
Initial value		4'b0000				4'b0000		
Attribute		R/W				R/W		

■ Register functions

[bit 7:4] PLLK: PLL input clock frequency division ratio setting bit

Bit 7:4	Description
0000	
0001	
•	The frequency division is 1/(PLLK+1). Example: PLLK=0000 +1 => 1/1 frequency [Initial value]
•	
1111	

[bit 3:0] PLLM: PLL VCO clock frequency division ratio setting bit

Bit 3:0	Description
0000	
0001	
•	The frequency division is 1/(PLLM+1). Example: PLLM=0000 +1 => 1/1 frequency [Initial value]
•	
1111	

<Notes>

- Set each frequency division ratio before enabling the PLL oscillation enable bit of the SCM_CTL.
- This register is not initialized by software reset.

5.12. PLL Control Register 2 (PLL_CTL2)

The PLL_CTL2 sets the PLL frequency division ratio.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				PLLN			
Initial value					5'b00000			
Attribute	-				R/W			

■ Register functions

[bit 7:5] RES: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit 4:0] PLLN: PLL feedback frequency division ratio setting bit

Bit 4:0	Description
00000	
00001	
.	The frequency division is 1/(PLLN+1). Example: PLLN=00000 +1 => 1/1 division [Initial value]
.	
11111	

<Notes>

- Set the frequency division ratio before enabling the PLL oscillation enable bit of the SCM_CTL.
- This register is not initialized by software reset.

5.13. Debug Break Watchdog Timer Control Register (DBWDT_CTL)

The DBWDT_CTL sets the watchdog timer count operation for debug mode tool break.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	DPHWBE	Reserved	DPSWBE				Reserved	
Initial value	1'b0	1'b0	1'b0					
Attribute	R/W	-	R/W					

■ Register functions

[bit 7] DPHWBE: HW-WDG debug mode break bit

Bit	Description
0	HW-WDG stops counting at the tool break [Initial value]
1	HW-WDG continues counting at the tool break

[bit 6] RES: Reserved bit

"0b0" is read from this bit.

Set this bit to "0b0" when writing.

[bit 5] DPSWBE: SW-WDG debug mode break bit

Bit	Description
0	SW-WDG stops counting at the tool break [Initial value]
1	SW-WDG continues counting at the tool break

[bit 4:0] RES: Reserved bits

"0b00000" is read from these bits.

Set these bits to "0b00000" when writing.

<Note>

This register is not initialized by software reset.

5.14. Interrupt Enable Register (INT_ENR)

The INT_ENR enables/disables interrupts.

■ Register configuration

bit	7	6	5	4	3	2	1	0			
Field	Reserved		FCSE		Reserved		PCSE		SCSE		MCSE
Initial value				1'b0			1'b0		1'b0		1'b0
Attribute				R/W			R/W		R/W		R/W

■ Register functions

[bit 7:6] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 5] FCSE: Anomalous frequency detection interrupt enable bit

Bit	Description
0	Disables FCS interrupts
1	Enables FCS interrupts

[bit 4:3] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 2] PCSE: PLL oscillation stabilization completion interrupt enable bit

Bit	Description
0	Disables PLL oscillation stabilization completion interrupts
1	Enables PLL oscillation stabilization completion interrupts

[bit 1] SCSE: Sub oscillation stabilization completion interrupt enable bit

Bit	Description
0	Disables sub oscillation stabilization completion interrupts
1	Enables sub oscillation stabilization completion interrupts

[bit 0] MCSE: Main oscillation stabilization completion interrupt enable bit

Bit	Description
0	Disables main oscillation stabilization completion interrupts
1	Enables main oscillation stabilization completion interrupts

<Note>

For "Anomalous frequency detection", see Chapter "Clock supervisor".

After each clock oscillation enable turn ON, then stabilization completion interrupt is available.

5.15. Interrupt Status Register (INT_STR)

The INT_STR indicates the status of interrupts.

■ Register configuration

bit	7	6	5	4	3	2	1	0	
Field	Reserved		FCSI		Reserved		PCSI	SCSI	MCSI
Initial value			1'b0			1'b0	1'b0	1'b0	
Attribute			R			R	R	R	

■ Register functions

[bit 7:6] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 5] FCSI: Anomalous frequency detection interrupt status bit

Bit	Description
0	No FCS interrupt has been asserted.
1	An FCS interrupt has been asserted.

[bit 4:3] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 2] PCSI: PLL oscillation stabilization completion interrupt status bit

Bit	Description
0	No PLL oscillation stabilization completion interrupt has been asserted.
1	A PLL oscillation stabilization completion interrupt has been asserted.

[bit 1] SCSI: Sub oscillation stabilization completion interrupt status bit

Bit	Description
0	No sub oscillation stabilization completion interrupt has been asserted.
1	A sub oscillation stabilization completion interrupt has been asserted.

[bit 0] MCSI: Main oscillation stabilization completion interrupt status bit

Bit	Description
0	No main oscillation stabilization completion interrupt has been asserted.
1	A main oscillation stabilization completion interrupt has been asserted.

5.16. Interrupt Clear Register (INT_CLR)

The INT_CLR clears interrupt causes.

■ Register configuration

bit	7	6	5	4	3	2	1	0			
Field	Reserved		FCSC		Reserved		PCSC		SCSC		MCSC
Initial value			1'b0				1'b0		1'b0		1'b0
Attribute			W				W		W		W

■ Register functions

[bit 7:6] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 5] FCSC: Anomalous frequency detection interrupt cause clear bit

Bit	Description
When 0 is written	The FCS interrupt cause is not affected by the written value.
When 1 is written	Clears the FCS interrupt cause.
When read	The fixed value "0" is read.

[bit 4:3] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 2] PCSC: PLL oscillation stabilization completion interrupt cause clear bit

Bit	Description
When 0 is written	The PLL oscillation stabilization completion interrupt cause is not affected by the written value.
When 1 is written	Clears the PLL oscillation stabilization completion interrupt cause.
When read	The fixed value "0" is read.

[bit 1] SCSC: Sub oscillation stabilization completion interrupt cause clear bit

Bit	Description
When 0 is written	The sub oscillation stabilization completion interrupt cause is not affected by the written value.
When 1 is written	Clears the sub oscillation stabilization completion interrupt cause.
When read	The fixed value "0" is read.

[bit 0] MCSC: Main oscillation stabilization completion interrupt cause clear bit

Bit	Description
When 0 is written	The main oscillation stabilization completion interrupt cause is not affected by the written value.
When 1 is written	Clears the main oscillation stabilization completion interrupt cause.
When read	The fixed value "0" is read.

<Note>

When this register is cleared, the interrupt status bit of the INT_STR is also cleared.

6. Clock Generation Unit Usage Precautions

This section explains the precautions for using the clock generation unit.

- The oscillation stabilization wait time of main and sub oscillators
Because the stabilization wait time of main/sub oscillator depends on the oscillator type (crystal, ceramic, etc.), the wait time suitable for the oscillator type must be selected.
- Changing the frequency division under stabilized PLL oscillation
When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then re-enable the PLL oscillation.
- Peripherals independent of clock control by the clock generation unit
The following peripherals run independently of clock control by the clock generation unit.
For information about how to handle each operating clock, see each relevant chapter.
 - USB operating clock generation unit : See Chapter "USB Clock Generation".
 - Clock supervisor : See Chapter "Clock supervisor".
 - Watchdog Timer : See Chapter "Watchdog Timer".
 - Watch counter : See Chapter "Watch Counter".
 - CAN prescaler : See Chapter "CAN Prescaler".
- Setting the oscillation stabilization wait time
Set the oscillation stabilization wait time of the main, sub, and PLL oscillators with relevant oscillation stabilization wait time setup registers, and then enable each oscillator.
Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.
- Checking main oscillation while using the PLL clock
It is prohibited to stop main oscillation while using PLL oscillation.
- Switching clock modes
Clock modes can be switched by changing the "SCM_CTL.RCS" register.
To switch clock modes, take the following steps:
 1. Set the oscillation stabilization wait time of each oscillator.
 2. Set the oscillation enable bit of the desired clock (SCM_CTL.xxxE) to 1.
 3. Check the oscillation stable bit of the desired clock (SCM_CTL.xxxRDY) to 1.
 4. Switch SCM_CTL.RCS.
 5. Wait until SCM_STR.RCM = SCM_CTL.RCS.
- Correlation between the clock mode switching and the oscillation stable bit
The timings when the oscillation stable bit (SCM_STR.xxxRDY) turns to 1 vary for the following clock mode switching.
 - When switching from the high-speed CR run, main run, or main PLL run to another clock mode:
Setting SCM_CTL.xxxE to 1 can start the oscillation stabilization wait time. You can check that SCM_STR.xxxRDY is 1 after the oscillation stabilization wait time has elapsed.
 - When switching from the low-speed CR run or sub run to the high-speed CR run, main run, or PLL run:
Setting SCM_CTL.MOSCE (or .PLLE) to 1 does not start the oscillation stabilization wait time. To start the main (or high-speed CR or PLL) oscillation stabilization wait time, SCM_CTL.RCS must be switched. After the oscillation stabilization wait time has elapsed, you can check that SCM_STR.xxRDY is 1.

- If the standby mode is released by an interrupt, the device restarts in the clock mode that indicated by the RCM bit in the SCM_CTL.
- If any reset occurs other than software resets, the high-speed CR clock (CLKHC) is set as a master clock. In addition, high-speed CR clock mode is set as clock mode.
- If any reset other than software resets is executed, the main and sub oscillators, and PLL oscillation stop. If you want to use those oscillators again after the reset, enable them using the "SCM_CTL".
- For the correlation between each clock mode and start/stop of the oscillation, see Chapter "Low Power Consumption Mode".

Chapter: High-Speed CR Trimming

This chapter explains the High-Speed CR Trimming Function.

1. High-Speed CR Trimming Function Overview
2. High-Speed CR Trimming Function Configuration and Block Diagram
3. High-Speed CR Trimming Function Operation
4. High-Speed CR Trimming Function Setup Procedure Example
5. High-Speed CR Trimming Function Register List
6. High-Speed CR Trimming Function Usage Precautions

1. High-Speed CR Trimming Function Overview

This section explains frequency trimming function of the internal high-speed CR oscillator.

The internal high-speed CR oscillators used for this device have fluctuation in frequency accuracy due to process variation. The fluctuation range of frequency accuracy can be reduced by configuring the trimming function.

The frequency trimming setup has the following functions:

- It can be configured the high-speed CR frequency trimming by writing a trimming value to the Frequency Trimming Register (MCR_FTRM).
- It can be calculated the value set to the Frequency Trimming Register from the count value within a certain period by using input capture.

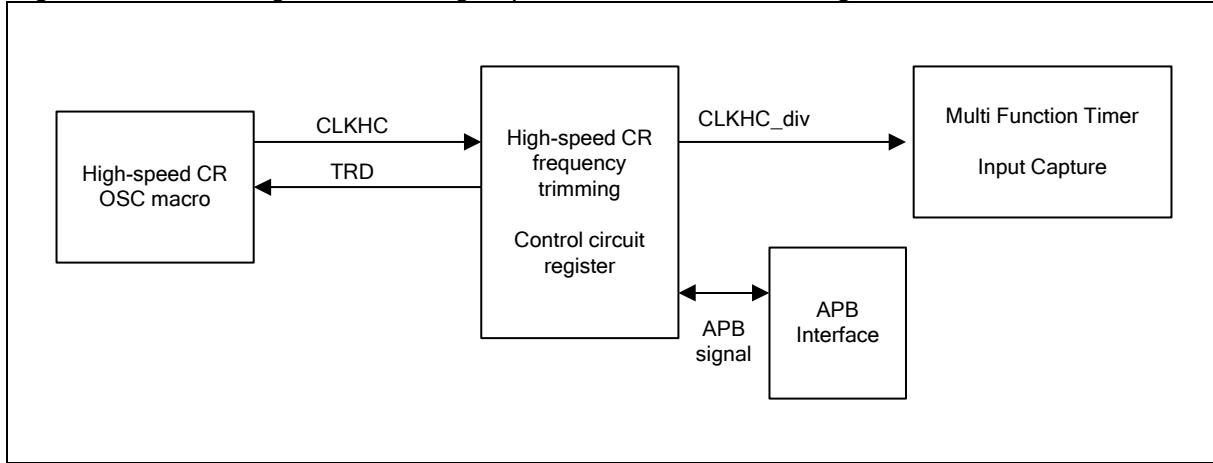
For the high-speed CR frequency accuracy, see electric characteristics described in "Data Sheet".

2. High-Speed CR Trimming Function Configuration and Block Diagram

This section explains the configuration and block diagram of internal high-speed CR oscillator frequency trimming function.

Figure 2-1 shows the block diagram of internal high-speed CR frequency trimming function.

Figure 2-1 Block diagram of the High-speed CR Oscillator Timing Circuit



■ Configuration

- **High-speed CR macro**

A macro of the high-speed CR clock outputs CLKHC (high-speed CR clock).

- **High-speed CR Trimming Control Circuit and registers**

A control circuit and registers for trimming high-speed CR.

- **Multi-function timer input capture**

This block counts frequency before setting to calculate the frequency trimming data for high-speed CR.

<Note>

For the clock definition, see Chapter "Clock".

3. High-Speed CR Trimming Function Operation

This section explains operation conducted by frequency trimming function of the internal high-speed CR oscillator.

■ Operation of high-speed CR oscillation frequency trimming function

● Frequency trimming setup

- The setup process writes a trimming data value to the Frequency Trimming Register (MCR_FTRM) to correct the misalignment of CR frequency accuracy caused by process variation.

● Register lock function

Write protect function is provided for the Frequency Trimming Register (MCR_FTRM), a function that protects the register from being rewritten without authorization when the system runs out of control.

● Trimming data acquisition

Data written to the Frequency Trimming Register (MCR_FTRM) can be acquired by one of the following three methods:

- Use the factory preset value stored in the "CR trimming" area inside flash memory.
- Calculate by yourself the value set to the Frequency Trimming Register from the count value within a certain period by using input capture.
- Output CR oscillation to an external pin, monitor the waveform to trim the frequency.

<Notes>

- Erasing flash memory also erases the "CR trimming" area inside the memory at the same time. If you use a value in the "CR trimming" area, therefore, save the data to other area (such as RAM) before erasing the flash memory,
or only erase sectors other than in the "CR trimming" area.
- For the address of the "CR trimming" area, see "Flash Programming Manual".

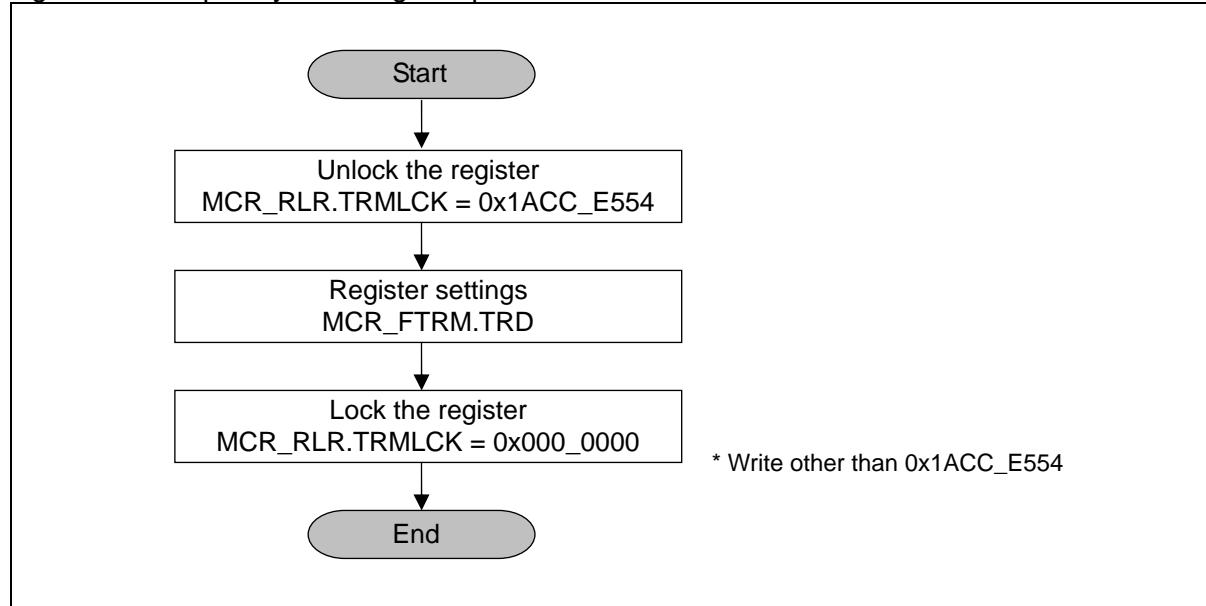
4. High-Speed CR Trimming Function Setup Procedure Example

This section provides an example of setting up frequency trimming function of the high-speed CR oscillator.

■ Frequency trimming setup

Take the steps shown in the following Figure 4-1 to set up frequency trimming.

Figure 4-1 Frequency trimming setup



1. Write "0x1ACCE554" to the TRMLCK[31:0] Register to unlock the MCR_FTRM Register.
2. Set the MCR_FTRM.
3. Write a value other than "0x1ACCE554" to the TRMLCK[31:0] Register to lock the MCR_FTRM Register.

■ Frequency trimming data acquisition example

When acquiring the data from the "CR trimming" area in flash memory,

Read the "CR trimming" area in flash memory and get the data.

Write the acquired value to the Frequency Trimming Register (MCR_FTRM).

■ How to calculate the frequency trimming data

The following explains how to calculate the trimming data of high-speed CR oscillation.

1. Let Ytgt, a target oscillation frequency be 4M[Hz]. Let Xtrm be the TRD value at the time.
2. Let Xtrrmin be the "initial value of the TRD bit - 20%" value. Let Ymin[Hz] be the frequency at this time.
3. Let Xtrrmax be the "initial value of the TRD bit + 20%" value. Let Ymax[Hz] be the frequency at this time.
4. The following expressions give TRD set value Xtrm, amounting to target oscillation frequency Ytgt.

$$(Tilt) \quad K = \frac{Y_{max} - Y_{min}}{X_{trrmax} - X_{trrmin}}$$

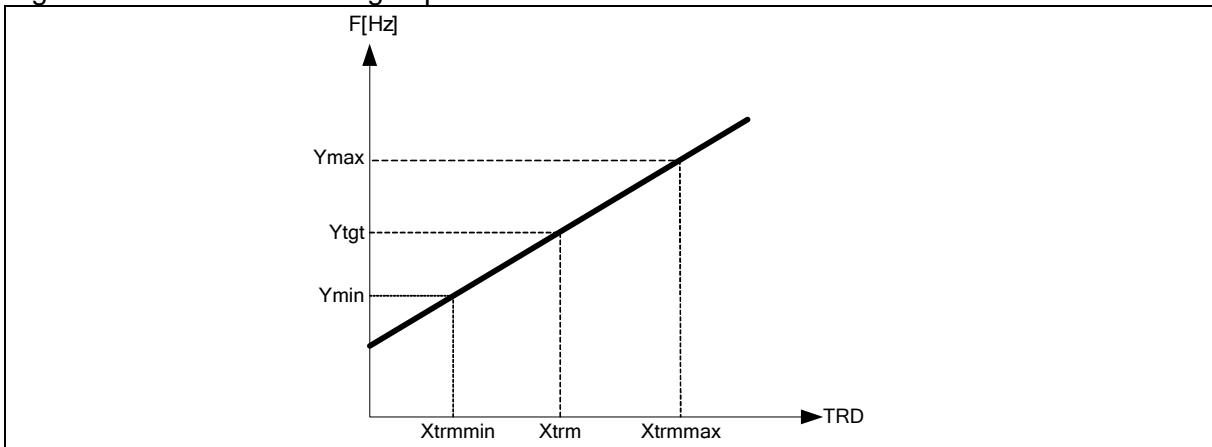
$$(TRD \text{ set value}) = \frac{Y_{tgt} - Y_{min}}{K} + X_{trrmin}$$

Example: When Ytgt = 4MHz, Ymax=4.8MHz, Ymin=3.2MHz, Xtrrmax=d'130 and Xtrrmin=d'100, the value Xtrm becomes as follows:

$$K = \frac{4.8 \text{ M} - 3.2 \text{ M}}{130 - 100} = 53333$$

$$X_{trm} = \frac{4\text{M} - 3.2\text{M}}{53333} + 100 = 115$$

Figure 4-2 Method to trim high-speed CR oscillation



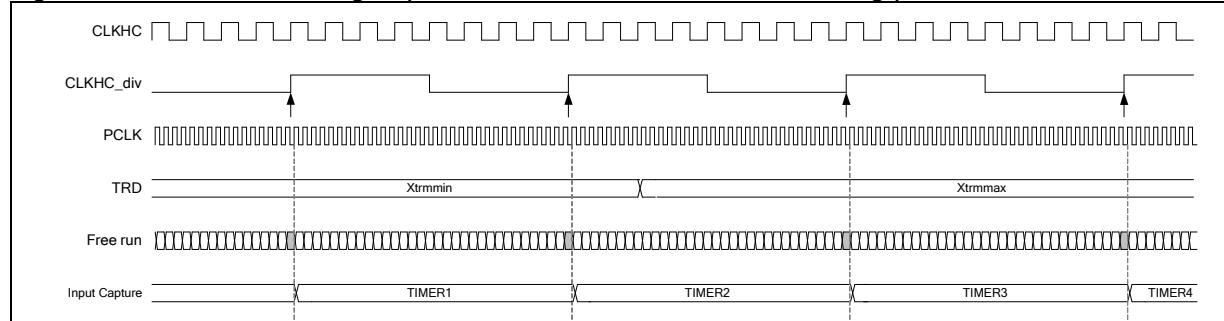
<Note>

For information about how to measure Ymin and Ymax, see "Example trimming data acquisition using input capture".

■ Example trimming data acquisition using input capture

Figure 4-3 shows the time chart of high-speed CR oscillation and the trimming process.

Figure 4-3 Time chart of high-speed CR oscillation and the trimming process



Run the free run timer by setting the main oscillation clock (CLKMC) as the master clock (measurement reference clock). Clear the free run timer once before measurement (to avoid overflow). Activate a trigger on the rising of the high-speed CR frequency division clock (CLKHC_div) when setting Xtrrmin or Xtrrmax, read the input capture timer value at that time, and perform the following calculations.

$$Y_{\min} = \frac{\text{DIV}}{(\text{TIMER2} - \text{TIMER1}) \times \text{PCLK}}$$

$$Y_{\max} = \frac{\text{DIV}}{(\text{TIMER4} - \text{TIMER3}) \times \text{PCLK}}$$

- TIMER1 to TIMER2: Input capture timer value at Y_{\min}
- TIMER3 to TIMER4: Input capture timer value at Y_{\max}
- PCLK: Cycle when main oscillation is selected for the master clock
- DIV: Frequency division ratio (CSR setting value)
- CLKHC_div : Division clock for the high-speed CR oscillation clock (CLKHC)

Example: When PCLK = 40MHz = 25ns, frequency division ratio = 1/8, and TIMER2 - TIMER1 = 100,

$$Y_{\min} = \frac{8}{(100 \times 25 \text{ sec}) \times 10^{-9}} \doteq 3.2 \text{MHz}$$

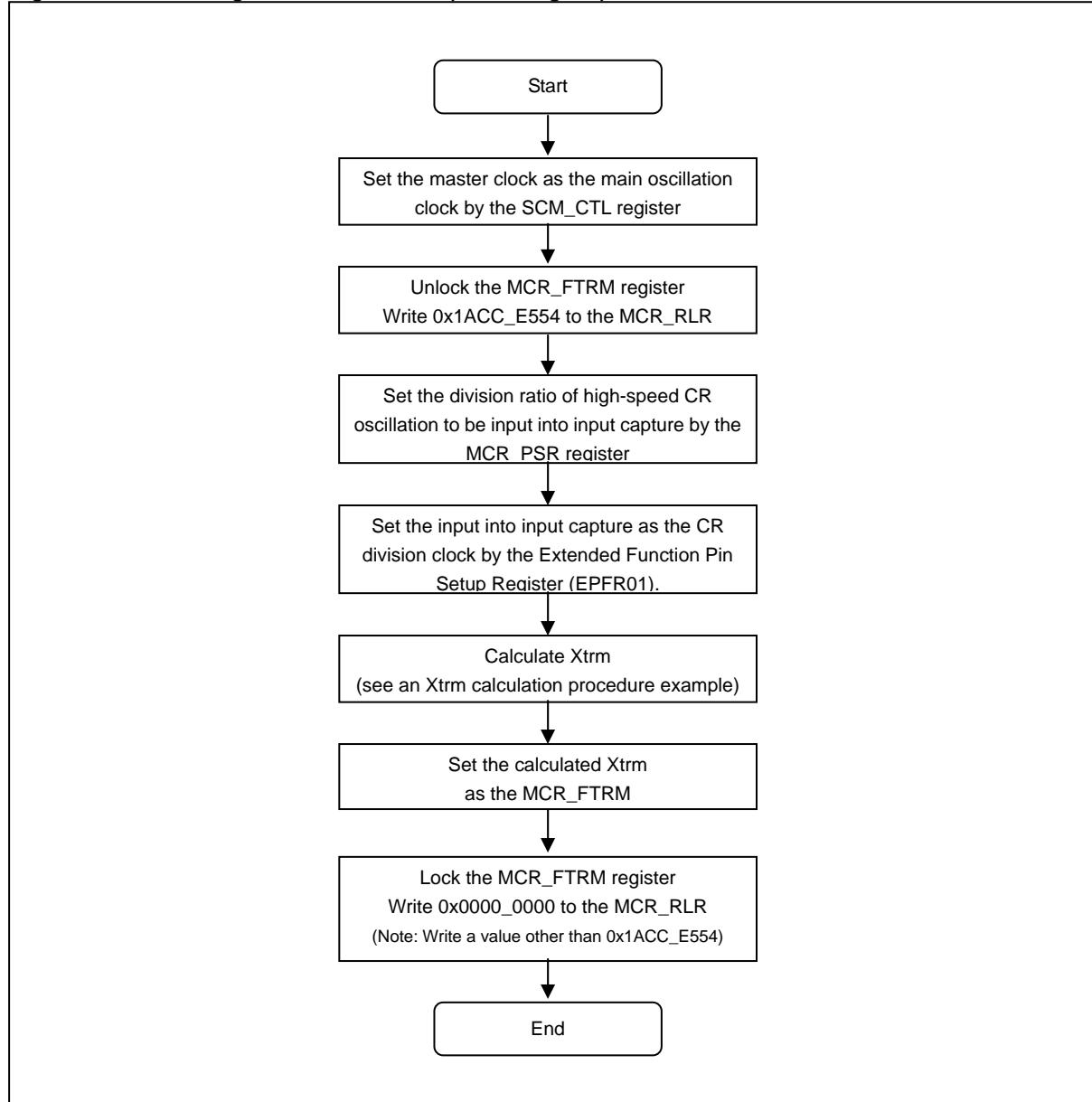
<Note>

The input capture which can be used for trimming is ICU ch.3 of the multi-function timer Unit0. PCLK in Figure 4-3 is an APB1 clock. For the PCLK at that time, main oscillation must be selected for the master clock.

■ Frequency trimming procedure example

The following shows a trimming procedure example of high-speed CR oscillation.

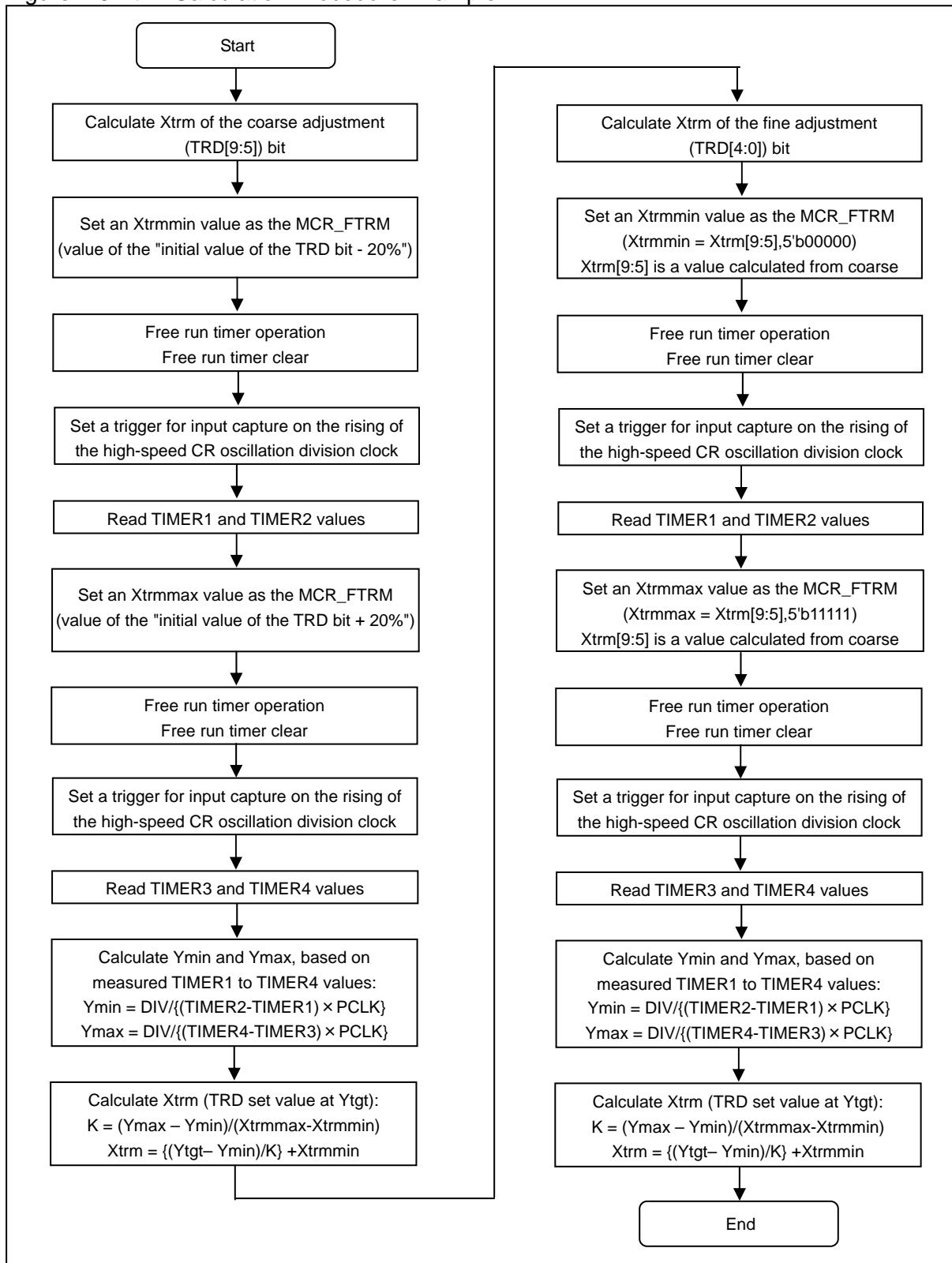
Figure 4-4 Trimming Procedure Example of High-speed CR Oscillation



■ Xtrm calculation procedure example

The following shows an Xtrm calculation procedure example. Perform frequency trimming in the two stages of coarse adjustment and fine adjustment.

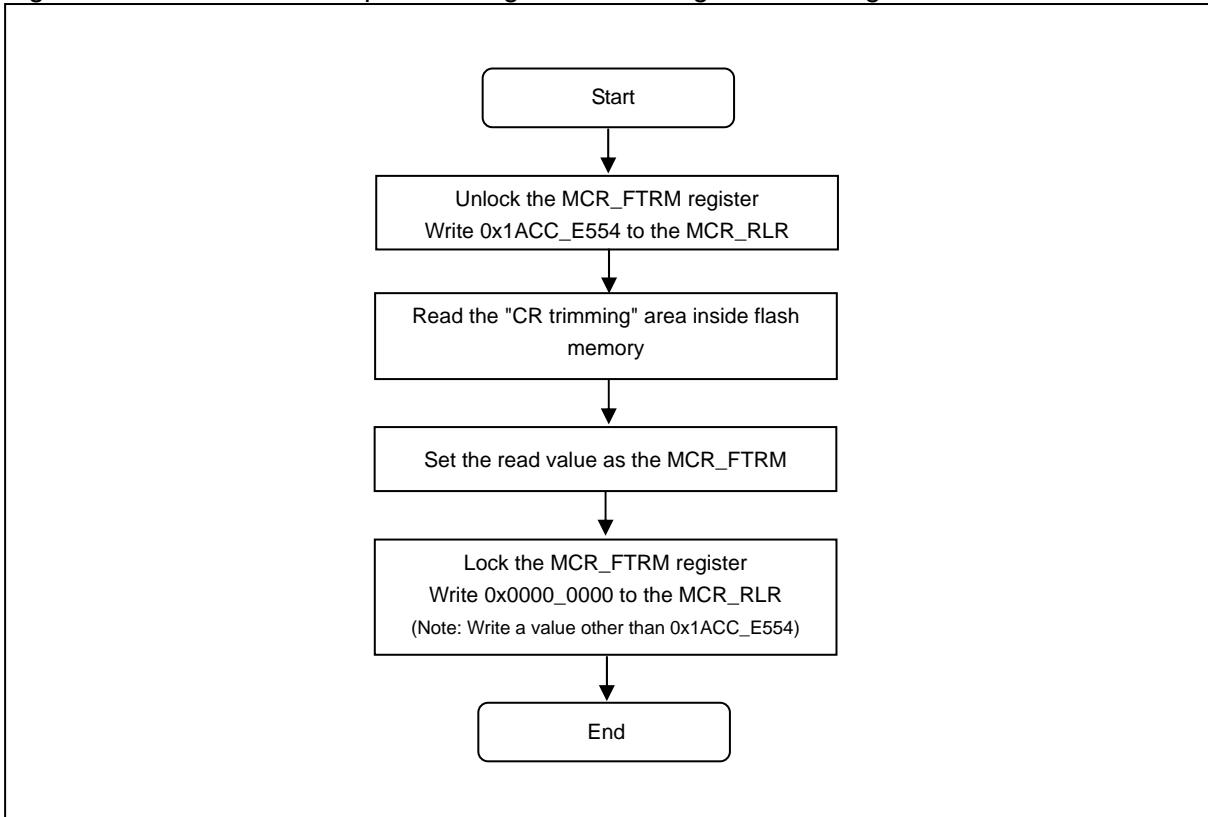
Figure 4-5 Xtrm Calculation Procedure Example



■ Procedure example of using "CR trimming" area storage data inside flash memory

The following shows a procedure example of reading trimming data stored in the "CR trimming" area inside flash memory and setting it as the frequency trimming register.

Figure 4-6 Procedure Example of Using "CR Trimming" Area Storage Data



<Note>

For the address of the CR trimming area, see "Flash Programming Manual".

5. High-Speed CR Trimming Function Register List

The following lists and explains registers used for frequency trimming function of the high-speed CR oscillator.

Table 5-1 lists the registers.

Table 5-1 Register list

Abbreviation	Register name	See
MCR_PSR	High-speed CR oscillation Frequency Division Setup Register	5.1
MCR_FTRM	High-speed CR oscillation Frequency Trimming Register	5.2
MCR_RLR	High-speed CR oscillator Register Write-Protect Register	5.3

5.1. High-speed CR oscillation Frequency Division Setup Register (MCR_PSR)

The MCR_PSR sets the frequency division ratio of high-speed CR oscillation.
A divided clock can be input into input capture.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field			Reserved				CSR	
Initial value			-				2'b01	
Attribute			-				R/W	

■ Register functions

[bit 7:2] : Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1:0] CSR: High-speed CR oscillation frequency division ratio setting bit

Bit 1	Bit 0	Description
0	0	1/4
0	1	1/8 [Initial value]
1	0	1/16
1	1	1/32

<Note>

This register is not initialized by software reset.

5.2. High-speed CR oscillation Frequency Trimming Register (MCR_FTRM)

The MCR_FTRM sets the frequency trimming value.
This section explains register configuration and register functions.

■ Register configuration

bit	31	16
Field	Reserved	
Initial value	-	
Attribute	-	
bit	15	10 9 8 7 6 5 4 3 2 1 0
Field	Reserved	TRD[9:0]
Initial value	-	10'b0110000000
Attribute	-	R/W

■ Register functions

[bit 31:10] : Reserved bits
 "0" is always read from these bits.
 Writing is ignored.

[bit 9:0] TRD: Frequency trimming setup bit

Bit 9:5	Description
When write	This bit makes coarse adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. This bit fluctuates in frequency steps of approximately 6% each time ±1 setting is made.
When read	A specified value is read. As an initial value, 5'b01100 is read.

Bit 4:0	Description
When write	This bit makes fine adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. This bit fluctuates in frequency steps of approximately 0.4% each time ±1 setting is made.
When read	A specified value is read. As an initial value, 5'b00000 is read.

<Notes>

- This register is not initialized by software reset.
- For values to be set to the TRD bit, see trimming data acquisition in the operation explanation of the frequency trimming function.

5.3. High-Speed CR Oscillator Register Write-Protect Register (MCR_RLR)

The MCR_RLR controls the write-protect state of the frequency trimming register (MCR_FTRM).

■ Register configuration

bit	31	16
Field	TRMLCK[31:16]	
Initial value	16'h0000	
Attribute	R/W	
bit	15	0
Field	TRMLCK[15:0]	
Initial value	16'h0001	
Attribute	R/W	

■ Register functions

[bit 31:0] TRMLCK: Register write-protect bits

Bit 31:0	Description
When read	When 0x00000000 is read, the MCR_FTRM register is currently unlocked. When 0x00000001 is read, the MCR_FTRM register is currently locked.
Writing other than 0x1ACCE554	Locks the MCR_FTRM register
Writing 0x1ACCE554	Unlocks the MCR_FTRM register

<Note>

This register is not initialized by software reset.

6. High-Speed CR Trimming Function Usage Precautions

This section explains the precautions for using high-speed CR trimming function.

- About low-speed CR oscillator
This trimming function is only enabled for the high-speed CR oscillator.
It cannot apply to the low-speed CR oscillator.
- About data stored in the "CR trimming" area
The "CR trimming" area stores the factory preset frequency trimming data. For the preset data, see "Data Sheet".
- How to use input capture
For information about how to use input capture, see Chapters "Multi-Function Timer" and "I/O Port".
- About FCS (Anomalous Frequency Detection)
For FCS function (anomalous frequency detection), see Chapter "Clock supervisor". Do not perform CR trimming after the FCS function is enabled.

Chapter: Resets

This chapter explains the function and operation of the resets.

1. Overview
2. Configuration
3. Explanation of Operations
4. Registers

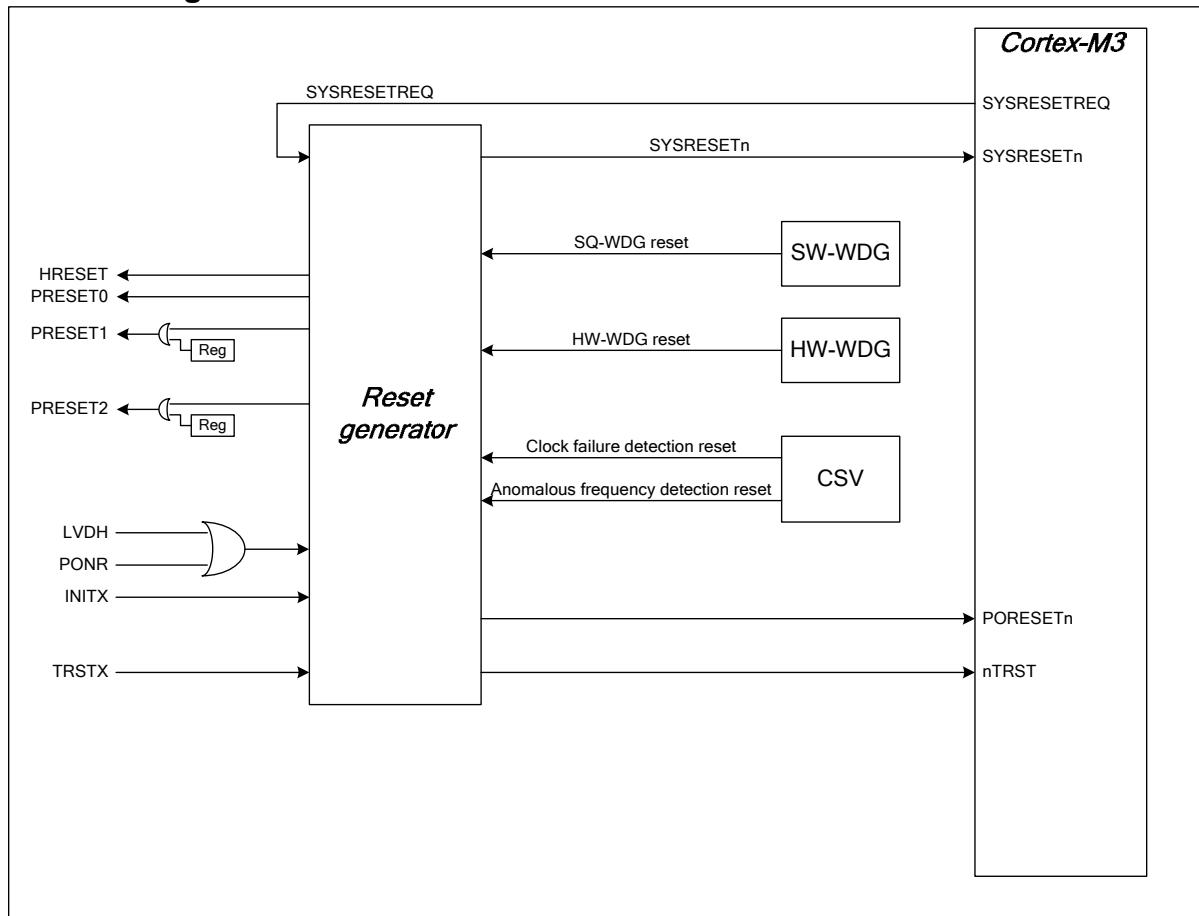
1. Overview

This product has the following reset causes and issues a reset to initialize a device upon accepting a reset cause.

- Power-on reset
- INITX pin input
- External power supply/low-voltage detection reset
- Software watchdog reset
- Hardware watchdog reset
- Clock failure detection reset
- Anomalous frequency detection reset
- Software reset
- TRSTX pin input

2. Configuration

■ Block Diagram of Resets



PONR	: Power-on reset
INITX	: INITX pin input reset
LVDH	: Low-voltage detection reset
TRSTX	: TRSTX pin input reset
HRESET	: AHB bus reset (a bus reset issued by all reset causes)
PRESET0, 1, 2	: APB0, APB1, APB2 bus resets (bus resets issued by all reset causes)
SW-WDG reset	: Software watchdog reset
HW-WDG reset	: Hardware watchdog reset
PORESETn	: Power-on reset that is input to Cortex-M3
SYSRESETn	: System reset that is input to Cortex-M3
SYSRESETREQ	: "SYSRESETREQ bit" signal of Cortex-M3 internal reset control register
nTRST	: SWJ-DP reset

3. Explanation of Operations

This section explains the operations of the resets.

- 3.1 Reset Causes
- 3.2 Resetting Inside the Device
- 3.3 Reset Sequence
- 3.4 Operations After Resets are Cleared

3.1. Reset Causes

This section explains reset causes.

■ Power-On Reset (PONR)

A reset that is generated at power-up.

Generated by	This signal is generated by detecting a rising edge of the power supply.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware.
Flag	Bit 0 (PONR) of reset cause register (RST_STR) = 1

■ INITX Pin Input Reset (INITX)

A reset that is externally input from a device.

Generated by	This signal is generated by inputting a low level to INITX pin.
Cleared by	This signal is cleared by inputting a high level to INITX pin.
Initialization target	Initializes all register settings and hardware except the debug circuit. Note: The reset cause register is not initialized.
Flag	Bit 1 (INITX) of reset cause register (RST_STR) = 1

* The content of the internal RAM is retained if a reset is asynchronously input from INITX pin.

■ Low-voltage Detection Reset, External Voltage Monitoring (LVDH)

A reset that is input from a low-voltage detection circuit when a decrease in the external voltage is detected.

Generated by	This signal is generated when an external voltage is lowered than a specified level.
Cleared by	This signal is cleared when an external voltage is more than a specified level.
Initialization target	Initializes all register settings and hardware.
Flag	Bit 0 (PONR) of reset cause register (RST_STR) = 1

■ Software Watchdog Reset (SWDGR)

A reset that is input from the software watchdog timer.

Generated by	This signal is generated when the software watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and hardware watchdog timer (including control registers). Note: The reset cause register is not initialized.
Flag	Bit 4 (SWDT) of reset cause register (RST_STR) = 1

■ Hardware Watchdog Reset (HWDGR)

A reset that is input from the hardware watchdog timer.

Generated by	This signal is generated when the hardware watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit. Note: The reset cause register is not initialized.
Flag	Bit 5 (HWDT) of reset cause register (RST_STR) = 1

■ Clock Failure Detection Reset (CSVR)

A reset that is input when the main or sub crystal oscillator being monitored fails.

Generated by	This signal is generated when a clock failure is detected in the main or sub crystal oscillator.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and clock failure detection circuit(some registers). Note: The reset cause register is not initialized.
Flag	Bit 6 (CSVR) of reset cause register (RST_STR) = 1 Bit 1 (SCMF) or bit 0 (MCMF) of CSV status register (CSV_STR) = 1 Note: For details on the CSV_STR, see Chapter "Clock Failure Detection".

■ Anomalous Frequency Detection Reset (FCSR)

A reset that is input when an anomalous frequency is detected in the main crystal oscillator.

Generated by	This signal is generated when the frequency of the main crystal oscillator is outside of any given setting.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and the anomalous frequency detection(some registers). Note: The reset cause register is not initialized.
Flag	Bit 7 (FCSR) of reset cause register (RST_STR) = 1

■ Software Reset (SRST)

A reset that is generated when an access to the reset control register occurs.

Generated by	This signal is generated by a write to the reset control register (SYSRESETREQ bit).
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the following: Registers that are not initialized by a software reset <ul style="list-style-type: none">· Debug circuit· All registers related to clock control· Part of registers that control software and hardware watchdog timers· Part of registers in the clock failure detection circuit· Part of registers that detect an anomalous frequency· Part of registers for CR trimming· Reset cause register
Flag	Bit 8 (SRST) of reset cause register (RST_STR) = 1

<Notes>

- For reset control register (SYSRESETREQ) that controls the software reset, see "Chapter 3, System Control", in "Cortex-M3 Technical Reference Manual".
 - The reset cause register that can determine the occurrence of each reset cause is initialized only by power-on reset.
-

3.2. Resetting Inside the Device

This section explains the internal reset signals of this device.

Resets that are internally connected to the device are divided into resets that are input to the Cortex-M3 core and resets that are input to peripheral circuits.

3.2.1 Resets to Cortex-M3

3.2.2 Resets to Peripheral Circuit

3.2.1. Resets to Cortex-M3

The device has three reset inputs to the Cortex-M3: PORESETn, SYSRESETn, and nTRST. The following provides reset causes for these three reset inputs.

■ Power-on reset PORESETn

Reset Causes	<ul style="list-style-type: none">· Power-on reset (PONR)· Low-voltage detection reset (LVDH)
--------------	--

■ System reset SYSRESETn

Reset Causes	<ul style="list-style-type: none">· Power-on reset (PONR)· Low-voltage detection reset (LVDH)· INITX pin input (INITX)· Software watchdog reset (SWDGR)· Hardware watchdog reset (HWDGR)· Clock Failure Detection reset (CSVR)· Anomalous frequency detection reset (FCSR)· Software reset (SRST)
--------------	--

■ SWJ-DP Reset nTRST

Reset Causes	<ul style="list-style-type: none">· Power-on reset (PONR)· Low-voltage detection reset (LVDH)· TRSTX pin input (TRSTX)
--------------	--

3.2.2. Resets to Peripheral Circuit

The bus resets (HRESET, PRESET0, PRESET1, and PRESET2) that are input to the peripheral circuit are basically generated by all reset causes. Resetting of PRESET1 and PRESET2 can be controlled by register settings.

The following provides reset causes for the bus resets.

■ Resets to Peripheral Circuit

● HRESET and PRESET0

Reset Causes	<ul style="list-style-type: none">Power-on reset (PONR)Low-voltage detection reset (LVDH)INITX pin input (INITX)Software watchdog reset (SWDGR)Hardware watchdog reset (HWDGR)Clock Failure Detection reset (CSVR)Anomalous frequency detection reset (FCSR)Software reset (SRST)
--------------	--

● PRESET1 and PRESET2

Reset Causes	<ul style="list-style-type: none">Power-on reset (PONR)Low-voltage detection reset (LVDH)INITX pin input (INITX)Software watchdog reset (SWDGR)Hardware watchdog reset (HWDGR)Clock Failure Detection reset (CSVR)Anomalous frequency detection reset (FCSR)Software reset (SRST)APB bus resets (APBC1_PSR and APBC2_PSR)
--------------	---

<Notes>

- The peripheral circuit is essentially initialized with all reset causes. Depending on the specifications of the peripheral circuit, there are registers that are initialized only with specific causes. For the initialization conditions for registers, see the initialization conditions for the registers described in the relevant chapter.
- For details on APB bus resets (APBC1_PSR and APBC2_PSR), see Chapter "Clocks".

3.3. Reset Sequence

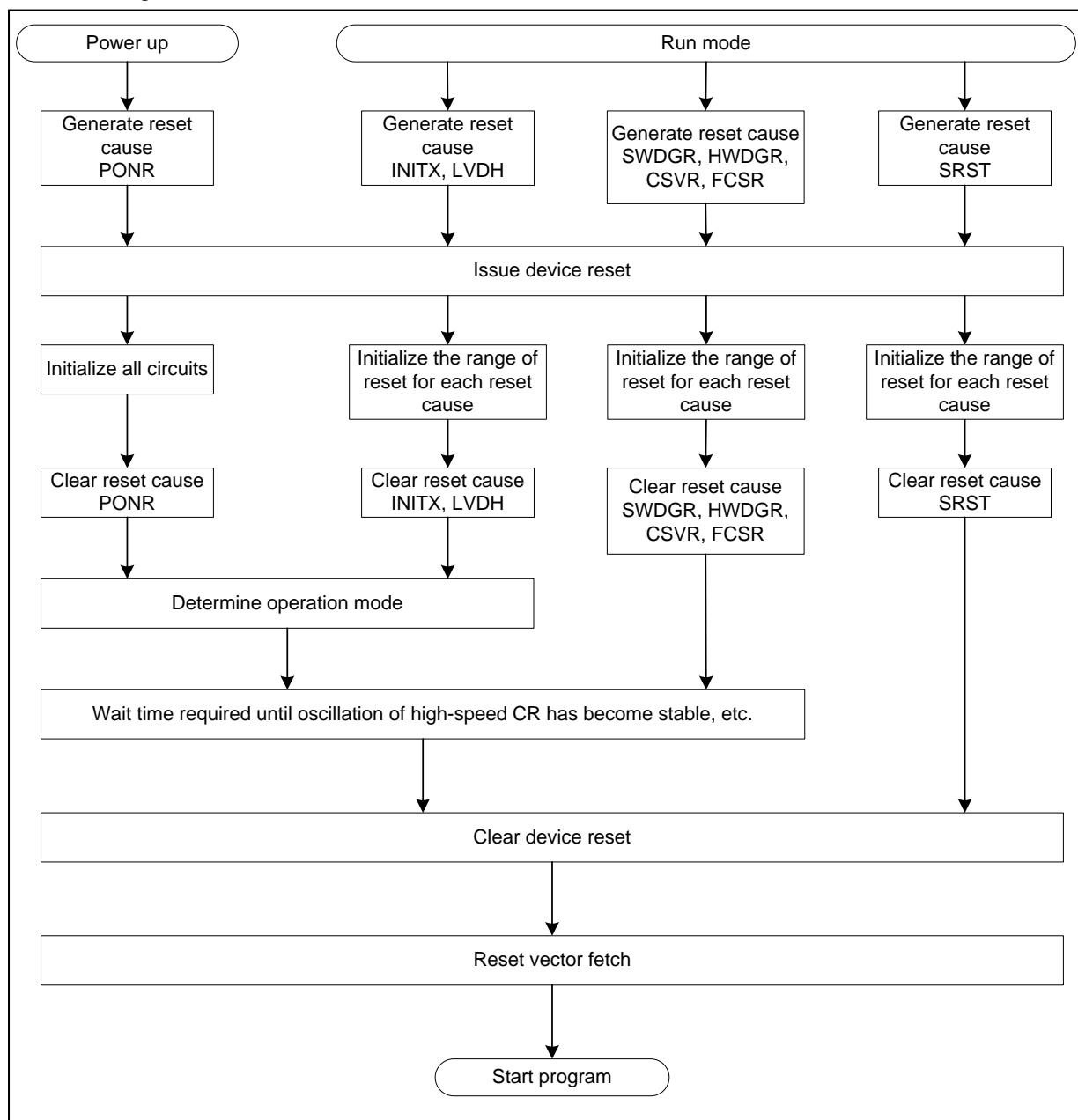
This model initiates the hardware the program and hardware operations starting with the initial state when a reset cause is cleared.

A series of operations starting with the reset and ending with the initiation of the operations is called a reset sequence.

The following explains a reset sequence.

■ State Transition Diagram for Resets

The following diagram shows a transition of reset states. The detailed operations are given in the following sections.



1. Capturing reset causes

Reset causes are captured and retained until a reset is issued to the device.

2. Issuing resets

When a reset is ready to be issued, a device internal reset is issued.

3. Clearing resets

When a reset cause is cleared, a device internal reset is extended for the amount of time required to clear the reset (for example, a wait required until oscillation of a high-speed CR has become stable). When the extended period of time has expired, the reset is cleared.

4. Determining operation mode

The operation mode is determined as PONR, LVDH or INITX is cleared and notified to each piece of the hardware. Any other reset causes do not cause the operation mode to change.

5. Reset vector fetch

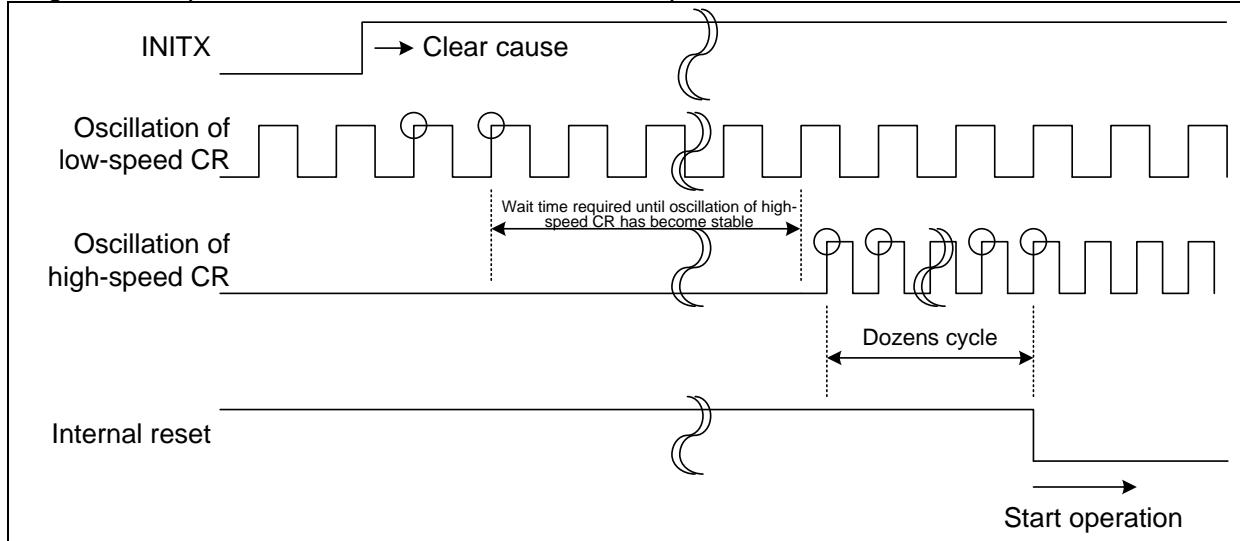
After a device internal reset is cleared, the CPU starts fetching a reset vector. The CPU fetches the obtained reset vector into the program counter and starts programmed operations.

3.4. Operations After Resets are Cleared

■ PONR, LVDH, INITX, HWDGR, SWDGR, CSVR, FCSR

Figure 3-1 provides an example of the operation waveform after a cause of INITX pin input reset has been cleared.

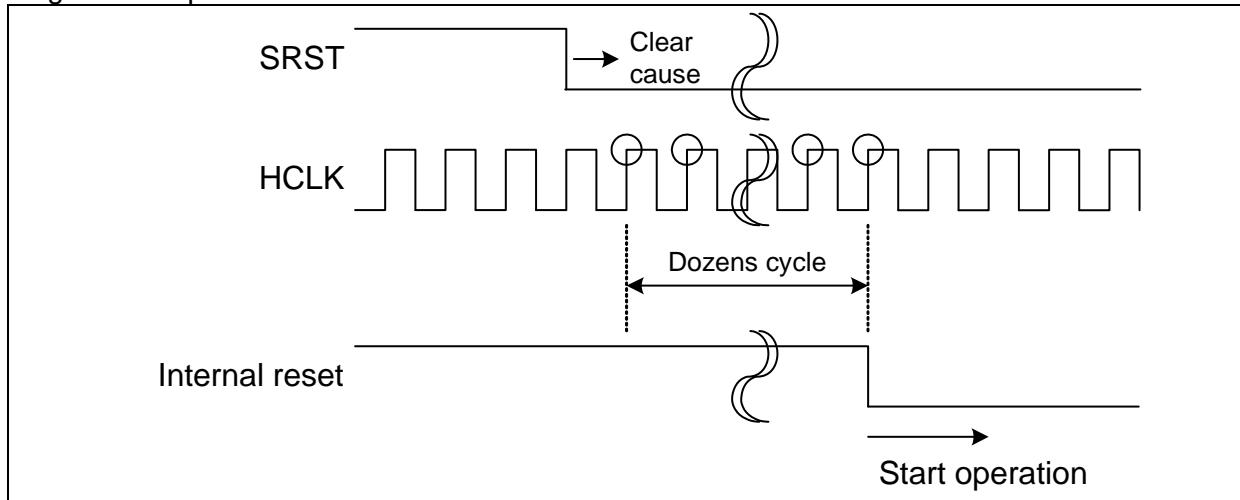
Figure 3-1 Operation Waveform After INITX Pin Input Reset has been Cleared



■ SRST

Figure 3-2 shows an operation waveform after a software reset has been cleared.

Figure 3-2 Operation Waveform After a Software Reset has been Cleared



4. Registers

This section explains the configuration and functions of the registers.

■ Register list

Abbreviation	Register name	See
RST_STR	Reset cause register	4.1

4.1. Reset Cause Register (RST_STR: ReSeT STatus Register)

The reset cause register shows causes of resets that have just occurred and initializes values upon power up.

Reading the register clears all bits.

It stores all reset causes that have been generated until after it has been read upon power up.

bit	15	14	13	12	11	10	9	8
Field	Reserved							SRST
Attribute	-	-	-	-	-	-	-	R
Initial value	-	-	-	-	-	-	-	1'b0
Bit	7	6	5	4	3	2	1	0
Field	FCSR	CSV	HWDT	SWDT	Reserved	Reserved	INITX	PONR
Attribute	R	R	R	R	-	-	R	R
Initial value	1'b0	1'b0	1'b0	1'b0	-	-	1'b0	1'b1

Note: Is the value of the initial value after power-up.

[bits 15:9] Reserved:

These are reserved bits and read value has no meaning.

[bit8] SRST: Software reset flag

Indicates a reset that is generated by one writing to the reset control register (SYSRESETREQ).

When a software reset is generated, SRST is enabled (SRST = 1).

bit	Description
0	A software reset has not been issued.
1	A software reset has been issued.

[bit 7] FCSR: Flag for anomalous frequency detection reset (FCSR)

Indicates a reset when an anomalous frequency is detected in the main crystal oscillator.

When the frequency of the main crystal oscillator is outside of a given setting, a reset is issued and FCSR is enabled (FCSR = 1).

bit	Description
0	An anomalous frequency detection reset has not been issued.
1	An anomalous frequency detection reset has been issued.

[bit 6] CSV: Clock failure detection reset flag

Indicates a reset when a failure is detected in the main or sub crystal oscillator.

If a stop is detected, a reset is issued and CSV is enabled (CSV = 1).

bit	Description
0	A clock failure detection reset has not been issued.
1	A clock failure detection reset has been issued.

Note: Please refer to another chapter 'Clock supervisor' for the method of judging whether the main oscillation or the suboscillation broke down.

[bit 5] HWDG: Hardware watchdog reset flag

Indicates a reset from the hardware watchdog timer.

If the timer underflows, a reset is issued and HWDT is enabled (HWDT = 1).

bit	Description
0	A hardware watchdog reset has not been issued.
1	A hardware watchdog reset has been issued.

[bit 4] SWDG: Software watchdog reset flag

Indicates a reset from the software watchdog timer.

If the timer overflows, a reset is issued and SWDT is enabled (SWDT = 1).

bit	Description
0	A software watchdog reset has not been issued.
1	A software watchdog reset has been issued.

[bit 3:2] Reserved:

It is a reserved bit and read value has no meaning.

[bit 1] INITX: INITX pin input reset flag

Indicates a reset that is externally input.

If a reset is externally input, INITX is enabled (INITX = 1).

bit	Description
0	An INITX pin input reset has not been issued.
1	An INITX pin input reset has been issued.

[bit 0] PONR: Power-on reset/low-voltage detection reset flag

Indicates a reset at power up and when a low-voltage is detected.

If a rising edge of power supply or a low-voltage is detected, a reset is issued and PONR is enabled (PONR = 1).

bit	Description
0	A power-on reset or low-voltage detection reset has not been issued.
1	A power-on reset or low-voltage detection reset has been issued.

<Note>

This register is initialized by a power-on reset or low-voltage detection reset. It is not initialized by any other reset causes. Reading the register clears all bits.

Chapter: Low-voltage Detection

This chapter explains the functions and operations of the Low-voltage Detection Circuit.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup Procedure Examples
5. Registers

1. Overview

The Low-voltage Detection Circuit monitors the power supply voltage, and generates reset and interrupt signals when the power supply voltage falls below the detection voltage.

■ Overview of Low-voltage Detection Circuit

● Operations of Low-voltage Reset Circuit

- This circuit monitors the power supply voltage (VCC) and generates a reset signal when the power supply voltage falls below the specified value.
- This circuit always monitors the power supply voltage.
- This circuit monitors the power supply voltage even in standby modes.
- This circuit generates a reset signal when the reduction of the power supply voltage is detected in standby modes.

● Operations of Low-voltage Interrupt Circuit

- This circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.
- This circuit allows selection of whether to enable or stop operations. The initial value is set to disable.
- This circuit allows specification of the detection voltage.
- This circuit can monitor the power supply voltage even in standby modes.
- This circuit returns from standby mode when the reduction of the power supply voltage is detected in the standby mode.

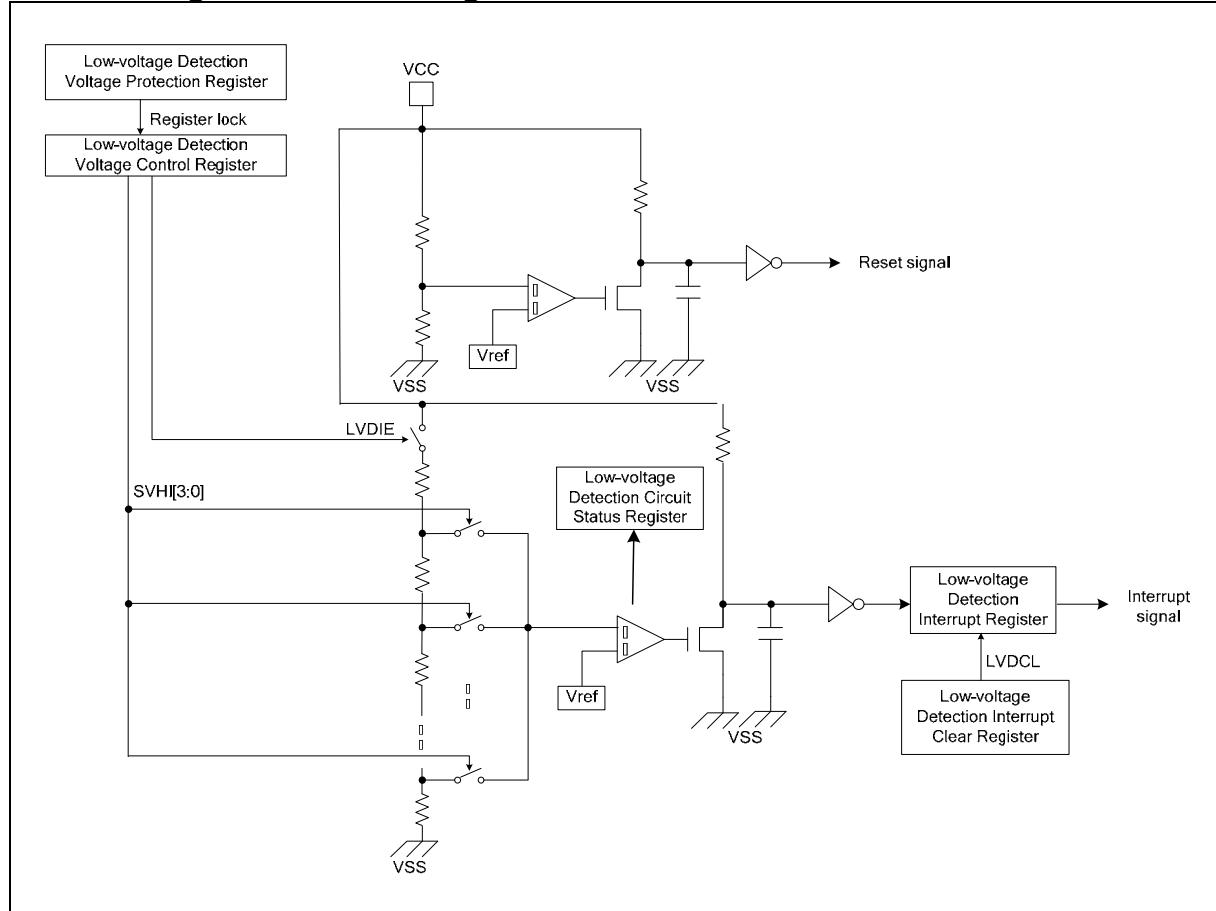
<Notes>

- If a low-voltage detection interrupt is enabled or the detection voltage is specified for a low-voltage detection interrupt, this circuit starts VCC voltage monitoring after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.
For the stabilization wait time of the Low-voltage Detection Circuit, refer to "Data Sheet".
- This circuit does not conduct monitoring the power supply voltage if PCLK2 is gated by TIMER mode, STOP mode, or PBC2_PSR Register while waiting for the stabilization of the Low-voltage Detection Circuit. After the status flag is read and the stabilization wait time has lapsed, change to the desired mode.
- The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected to prevent a writing error. To release write protection mode, write 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR).

2. Configuration

This section shows the block diagram of the Low-voltage Detection Circuit.

■ Block diagram of Low-voltage Detection Circuit



● Low-voltage Detection Voltage Control Register

This register controls whether to enable monitoring the power supply voltage for a low-voltage detection interrupt and specifies the detection voltage for a low-voltage detection interrupt.

● Low-voltage Detection Voltage Protection Register

This register write-protects the Low-voltage Detection Voltage Control Register.

● Low-voltage Detection Interrupt Register

This register holds a low-voltage detection interrupt cause.

● Low-voltage Detection Interrupt Clear Register

This register clears a low-voltage detection interrupt cause.

● Low-voltage Detection Circuit Status Register

This register checks the operation status of a low-voltage detection interrupt circuit.

■ Pins of Low-voltage Detection Circuit

The following shows the pins used in the Low-voltage Detection Circuit.

- VCC pin
The Low-voltage Detection Circuit monitors the power supply voltage of this pin.
- VSS pin
This pin is a GND pin used as a basis to detect the power supply voltage.

3. Explanation of Operations

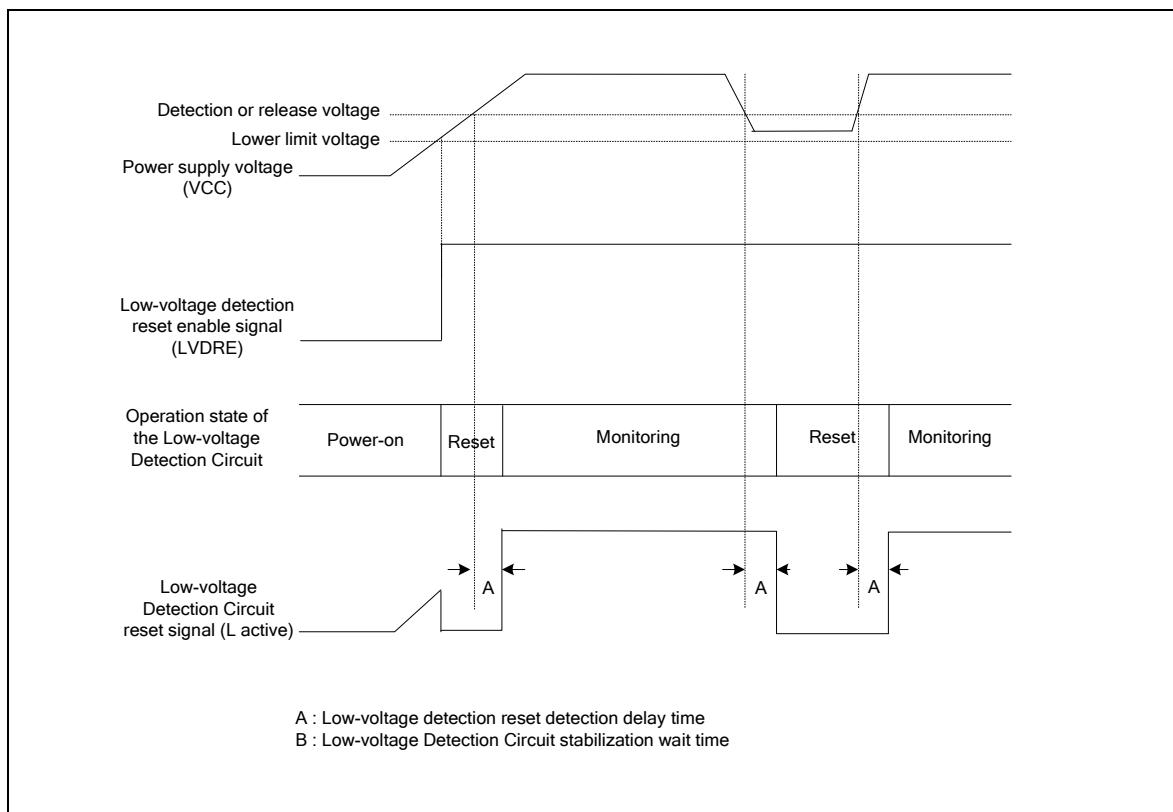
This section explains the operations of the Low-Voltage Detection Reset Circuit and the Low-voltage Detection Interrupt Circuit.

■ Operations of Low-Voltage Detection Reset Circuit

● Operations

The Low-Voltage Detection Reset Circuit always enters a monitoring state after power-on. This circuit generates a reset signal when the power supply voltage (VCC) falls below the detection voltage. A reset is released when the power supply voltage exceeds the release voltage.

This circuit is available in standby modes: SLEEP, TIMER, and STOP modes.



■ Operations of Low-voltage Detection Interrupt Circuit

● Operations

The Low-voltage Detection Interrupt Circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.

An interrupt request is enabled when "1" is set to the LVDIE bit of the Low-voltage Detection Voltage Control Register. The initial value is set to Not Enable ("0"). The interrupt detection voltage can be set by the SVHI bit of the Low-voltage Detection Voltage Control Register. When an interrupt request is enabled and the interrupt detection voltage is specified, the status flag LVDIRDY bit is set to "1" and this circuit starts monitoring the power supply voltage after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.

This circuit is available in standby modes: SLEEP, TIMER, and STOP modes. It is also applicable when the CPU returns from standby mode.

● Low-voltage detection interrupt request

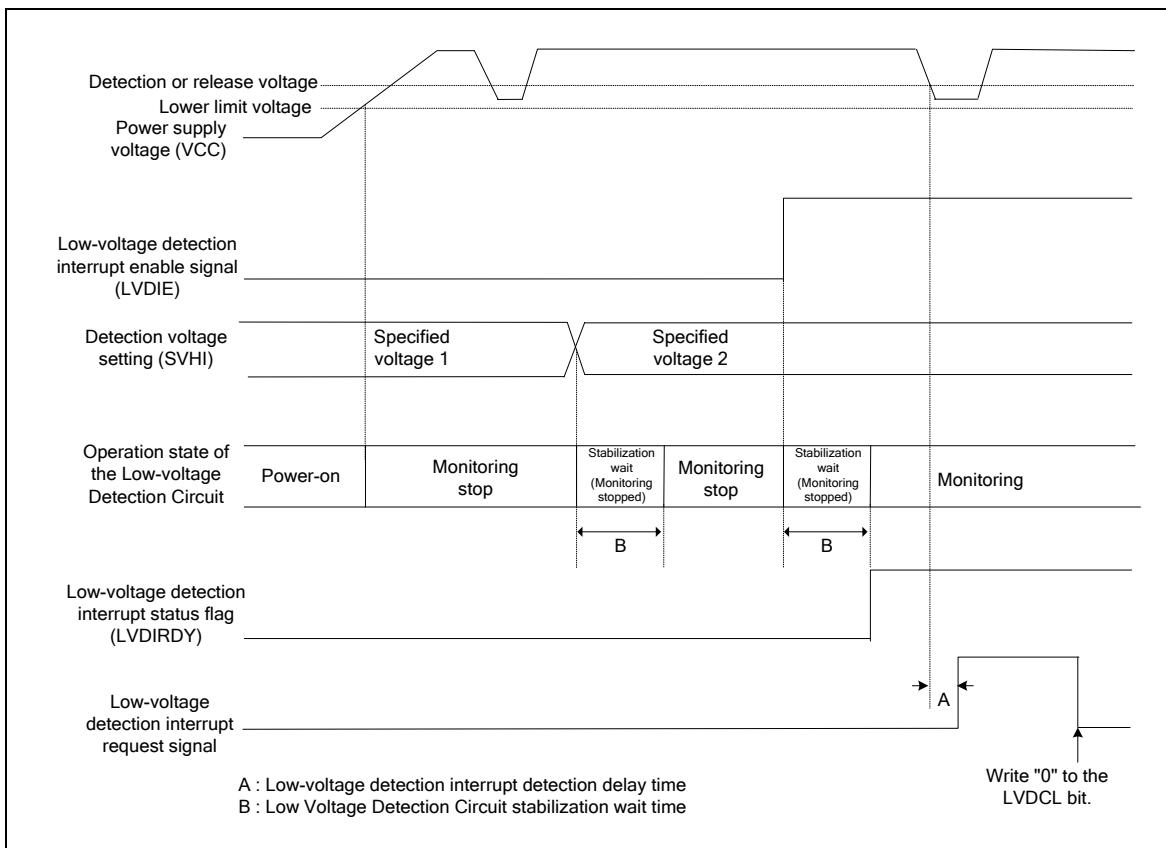
When the power supply voltage (VCC) falls below the specified voltage while a low-voltage detection interrupt is effective, "1" is set to the LVDIR bit of the Low-voltage Detection Interrupt Register to generate an interrupt request signal.

An interrupt request can be checked by reading the LVDIR bit.

● Canceling a low-voltage detection interrupt request

To cancel a low-voltage detection interrupt request, write "0" to the LVDCL bit of the Low-voltage Detection Interrupt Clear Register. This clears a low-voltage detection interrupt cause and cancels a low-voltage detection interrupt request.

Even when the power supply voltage is below the specified detection voltage, an interrupt request is canceled when "0" is written to the Low-voltage Detection Interrupt Clear Register.



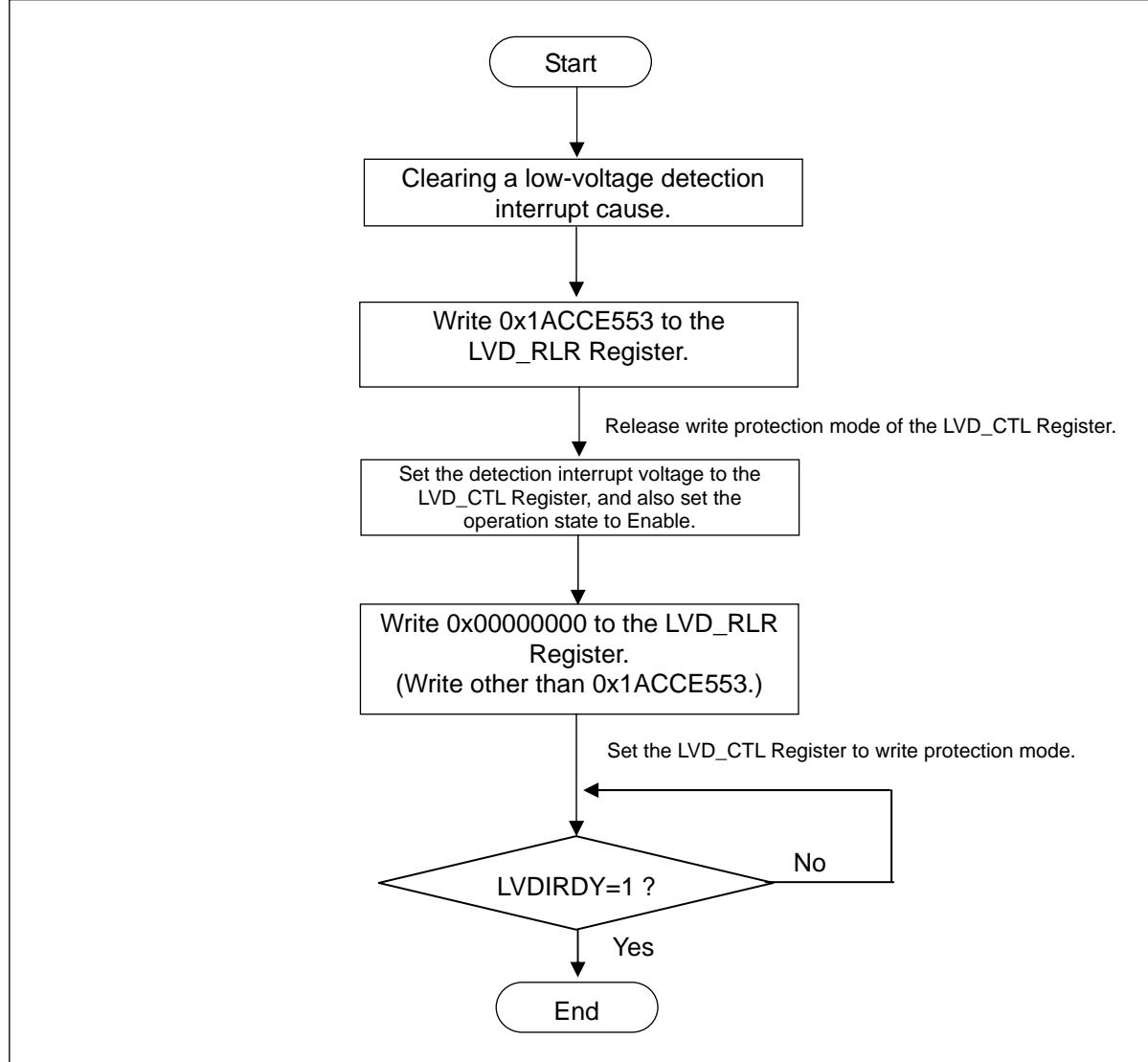
<Note>

This circuit does not conduct monitoring the power supply voltage if PCLK2 is gated by TIMER mode, STOP mode, or PBC2_PSR Register while waiting for the stabilization of the Low-voltage Detection Circuit. After checking that the status flag LVDIRDY is set to "1", change to the desired mode.

4. Setup Procedure Examples

This section explains the procedures to set up the Low-voltage Detection Circuit, giving examples.

Figure 4-1 Low-voltage detection interrupt setting



5. Registers

This section explains the configuration and functions of the registers used in the Low-voltage Detection Circuit.

■ List of Low-voltage Detection Circuit Registers

Abbreviation	Register name	See
LVD_CTL	Low-voltage Detection Voltage Control Register	5.1
LVD_STR	Low-voltage Detection Interrupt Register	5.2
LVD_CLR	Low-voltage Detection Interrupt Clear Register	5.3
LVD_RLR	Low-voltage Detection Voltage Protection Register	5.4
LVD_STR2	Low-voltage Detection Circuit Status Register	5.5

5.1. Low-voltage Detection Voltage Control Register (LVD_CTL)

The Low-voltage Detection Voltage Control Register (LVD_CTL) controls whether to enable monitoring the power supply voltage for a low-voltage detection interrupt and specifies the detection voltage for a low-voltage detection interrupt.

Bit	7	6	5	4	3	2	1	0
Field	LVDIE	Reserved	SVHI				Reserved	
Attribute	R/W	-	R/W	R/W	R/W	R/W	-	-
Initial value	0	1	0	0	0	0	0	0

[bit 7] LVDIE: Low-voltage detection interrupt enable bit

This bit is used to enable monitoring the power supply voltage of a low-voltage detection interrupt. When not enabling monitoring the power supply voltage, the Low-voltage Detection Interrupt Circuit is stopped.

Bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit 6] Reserved: Reserved bit

Read value has no meaning.
Set "1" to this bit.

[bit 5:2] SVHI: Low-voltage detection interrupt voltage setting bit

This bit is used to specify the detection voltage of a low-voltage detection interrupt.

bit5:2	Description
0000	Set the low-voltage detection interrupt voltage in the vicinity of 2.8V. [Initial value]
0001	Set the low-voltage detection interrupt voltage in the vicinity of 3.0V.
0010	Set the low-voltage detection interrupt voltage in the vicinity of 3.2V.
0011	Set the low-voltage detection interrupt voltage in the vicinity of 3.6V.
0100	Set the low-voltage detection interrupt voltage in the vicinity of 3.7V.
0101	Setting disabled
0110	Setting disabled
0111	Set the low-voltage detection interrupt voltage in the vicinity of 4.0V.
1000	Set the low-voltage detection interrupt voltage in the vicinity of 4.1V.
1001	Set the low-voltage detection interrupt voltage in the vicinity of 4.2V.
Others	Setting disabled

[bit 1:0] Reserved: Reserved bits
"0" is always set in read mode.
This bit has no effect in write mode.

<Notes>

- The low-voltage detection interrupt enable bit (LVDIE) must be enabled after "0" was written to the LVDCL bit of the Low-voltage Detection Interrupt Clear Register (LVD_CLR) to clear the interrupt request.
 - When the low-voltage detection interrupt enable bit (LVDIE) is not enabled, the Low-voltage Detection Circuit for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection interrupt flag is not set.
 - The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.
-

5.2. Low-voltage Detection Interrupt Register (LVD_STR)

The Low-voltage Detection Interrupt Register (LVD_STR) holds a low-voltage detection interrupt cause.

Bit	7	6	5	4	3	2	1	0
Field	LVDIR				Reserved			
Attribute	R	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit 7] LVDIR: Low-voltage detection interrupt bit

Bit	Description
0	A low-voltage detection interrupt request is not detected. [Initial value]
1	A low-voltage detection interrupt request has been detected.

[bit 6:0] Reserved: Reserved bits

"0" is always set in read mode.

This bit has no effect in write mode.

5.3. Low-voltage Detection Interrupt Clear Register (LVD_CLR)

The Low-voltage Detection Interrupt Clear Register (LVD_CLR) clears a low-voltage detection interrupt cause.

Bit Field	7	6	5	4	3	2	1	0
Attribute	LVDCL				Reserved			
Initial value	R/W	-	-	-	-	-	-	-

[bit 7] LVDCL: Low-voltage detection interrupt clear bit

Bit	Description
0	Clears the low-voltage detection interrupt bit of the Low-voltage Detection Interrupt Register (LVD_STR) to "0".
1	Has no effect in write mode. [Initial value]

"1" is always set in read mode.

[bit 6:0] Reserved: Reserved bits

"0" is always set in read mode.

This bit has no effect in write mode.

5.4. Low-voltage Detection Voltage Protection Register (LVD_RLR)

The Low-voltage Detection Voltage Protection Register (LVD_RLR) write-protects the Low-voltage Detection Voltage Control Register (LVD_CTL).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LVDLCK[31:16]															

Attribute R/W

Initial value 0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LVDLCK[15:0]															

Attribute R/W

Initial value 0x0001

[bit 31:0] LVDLCK[31:0]: Low-voltage Detection Voltage Control Register protection bit

- Setting 0x1ACCE553 enables writing the Low-voltage Detection Voltage Control Register (releases write protection mode).
- Setting a value other than 0x1ACCE553 disables writing the Low-voltage Detection Voltage Control Register (enables write protection mode).
- When the Low-voltage Detection Voltage Control Register is not set in write protection mode, 0x00000000 is read.
- When the Low-voltage Detection Voltage Control Register is set in write protection mode, 0x00000001 is read.

<Notes>

- The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.
- To enable write protection mode of the LVD_CTL register, set a value other than 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR).
- Once write protection mode is released for the LVD_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the LVD_RLR Register.

5.5. Low-voltage Detection Circuit Status Register (LVD_STR2)

The Low-voltage Detection Circuit Status Register (LVD_STR2) checks the operation status of a low-voltage detection interrupt.

bit	7	6	5	4	3	2	1	0
Field	LVDIRDY				Reserved			
Attribute	R	-	-	-	-	-	-	-
Initial value	0	1	0	0	0	0	0	0

[bit7] LVDIRDY : Low-voltage detection interrupt status flag

bit	Description
0	Stabilization wait state or monitoring stop state [Initial value]
1	Monitoring state

This bit has no effect in write mode.

[bit6:0] Reserved : Reserved bits

Read value has no meaning.

This bit has no effect in write mode.

Chapter: Low Power Consumption Mode

This chapter describes the functions and operations of low power consumption mode.

1. Overview of Low Power Consumption Mode
2. Configuration of CPU Operation Modes
3. Operations of Standby Modes
4. Standby Mode Setting Procedure Examples
5. List of Low Power Consumption Registers

1. Overview of Low Power Consumption Mode

To reduce the power consumption, the system provides low power consumption mode, which enables the use of three types of standby modes: SLEEP, TIMER, and STOP modes.

■ Overview of CPU operation modes

CPU operation modes are classified into the following types.

● RUN modes

- High speed CR run mode
- Main run mode
- PLL run mode
- Low speed CR run mode
- Sub run mode

● SLEEP modes

- High speed CR sleep mode
- Main sleep mode
- PLL sleep mode
- Low speed CR sleep mode
- Sub sleep mode

● TIMER modes

- High speed CR timer mode
- Main timer mode
- PLL timer mode
- Low speed CR timer mode
- Sub timer mode

● STOP mode

■ Overview of RUN mode

RUN mode is defined with a clock selected as a master clock. The base clocks, which are obtained by dividing the master clock frequency, are supplied to CPU clock, AHB bus clock, and APB bus clock to run the CPU, buses, and most peripherals.

The source clock frequency can be changed dynamically. When not using the main or sub oscillator, the source clock oscillator can be stopped.

RUN mode is divided into the following modes depending on the clock selected as a master clock.

● High speed CR run mode

In this mode, the high speed CR oscillator clock is used as a master clock. When not using the main or sub oscillator, the respective oscillators can be stopped. The low speed CR oscillator is always set to the active state. It changes to this mode after a reset has been released.

● Main run mode

In this mode, the main oscillator clock is used as a master clock. The status of the PLL Multiplier Circuit or sub oscillator varies depending on the setting of the PLLE or SOSCE bit. The high and low speed CR oscillators are always set to the active state.

● PLL run mode

In this mode, the PLL clock obtained by multiplying the main oscillator clock is used as a master clock. The main, high speed CR, and low speed CR oscillators are always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit.

● Low speed CR run mode

In this mode, the low speed CR oscillator clock is used as a master clock. The status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

● Sub run mode

In this mode, the sub oscillator clock is used as a master clock. The low speed CR oscillator is always set to the active state. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

■ Overview of SLEEP mode

SLEEP mode is classified as one of standby modes. SLEEP mode is used to stop CPU clocks. This causes the CPU to be stopped, reducing the power consumption. The resources connected to the AHB and APB bus clocks continue operations.

SLEEP mode is divided into the following modes depending on the clock selected as a master clock.

● High speed CR sleep mode

When the high speed CR oscillator clock is selected as a master clock, the system changes to high speed CR sleep mode if the transition to SLEEP mode is requested. In this mode, the status of the main or sub oscillator varies depending on the setting of the MOSCE or SOSCE bit. The low speed CR oscillator is always set to the active state.

● Main sleep mode

When the main clock is selected as a master clock, the system changes to main sleep mode if the transition to SLEEP mode is requested. In this mode, the status of the PLL Multiplier Circuit or sub oscillator varies depending on the setting of the PLLE or SOSCE bit. The high and low speed CR oscillators are always set to the active state.

● PLL sleep mode

When the PLL clock is selected as a master clock, the system changes to PLL sleep mode if the transition to SLEEP mode is requested. In this mode, the main, high speed CR, and low speed CR oscillators are always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit.

● Low speed CR sleep mode

When the low speed CR clock is selected as a master clock, the system changes to low speed CR sleep mode if the transition to SLEEP mode is requested. In this mode, the status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

● Sub sleep mode

When the sub clock is selected as a master clock, the system changes to sub sleep mode if the transition to SLEEP mode is requested. In this mode, the low speed CR oscillator is always set to the active state. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

■ Overview of TIMER mode

TIMER mode is classified as one of standby modes. TIMER mode is used to stop supplying a base clock. This causes the CPU clock, AHB bus clock, and all APB bus clocks to be stopped, reducing the power consumption. In this case, all functions are stopped, excluding the oscillators, PLL, hardware watchdog timer, watch counter, clock failure detector, and Low Voltage Detection Circuit.

TIMER mode is divided into the following modes depending on the clock selected as a master clock.

● High speed CR timer mode

When the high speed CR oscillator clock is selected as a master clock, the system changes to high speed CR timer mode if the transition to TIMER mode is requested. In this mode, the status of the main or sub oscillator varies depending on the setting of the MOSCE or SOSCE bit. The low speed CR oscillator is always set to the active state.

● Main timer mode

When the main clock is selected as a master clock, the system changes to main timer mode if the transition to TIMER mode is requested. In this mode, the status of the PLL Multiplier Circuit or sub oscillator varies depending on the setting of the PLLE or SOSCE bit. The high and low speed CR oscillators are always set to the active state.

● PLL timer mode

When the PLL clock is selected as a master clock, the system changes to PLL timer mode if the transition to TIMER mode is requested. In this mode, the main, high speed CR, and low speed CR oscillators are always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit.

● Low speed CR timer mode

When the low speed CR clock is selected as a master clock, the system changes to low speed CR timer mode if the transition to TIMER mode is requested. In this mode, the status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

● Sub timer mode

When the sub clock is selected as a master clock, the system changes to sub timer mode if the transition to TIMER mode is requested. In this mode, the sub oscillator and low speed CR oscillator are always set to the active state. The main oscillator, high speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

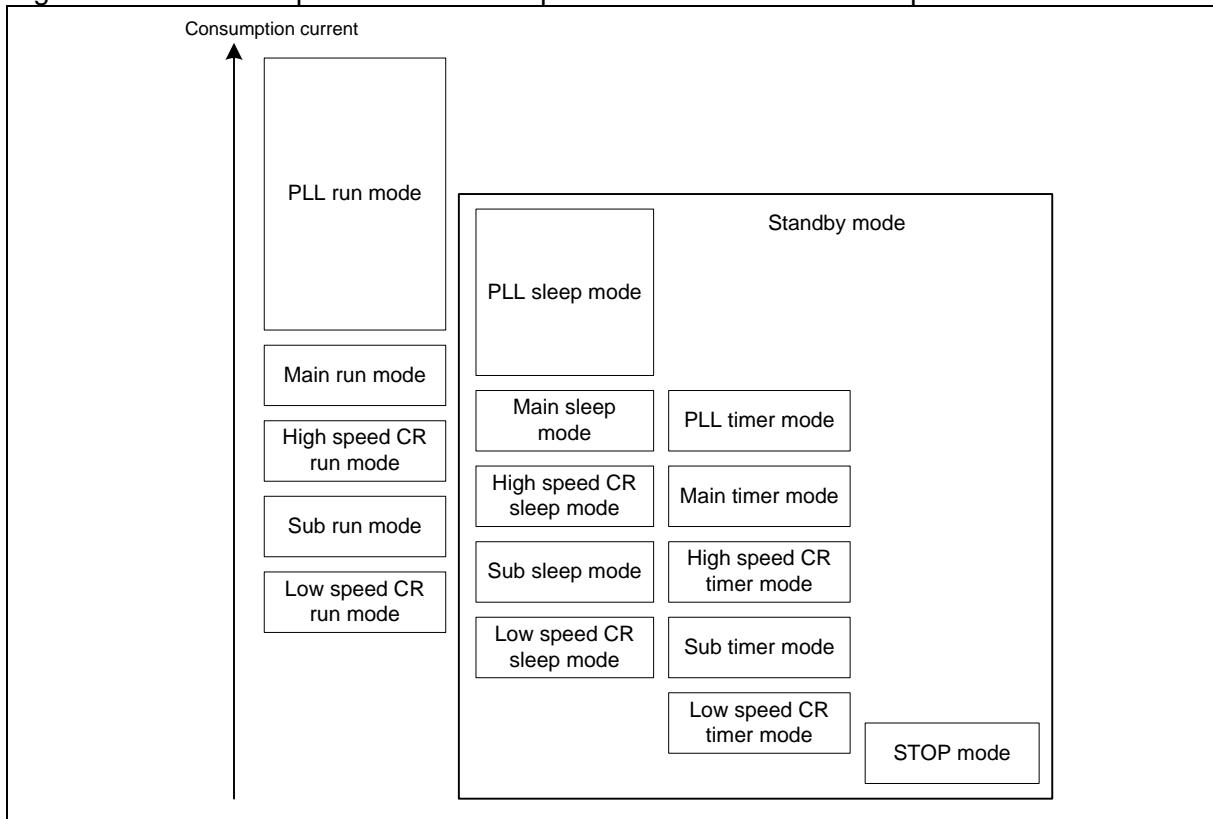
■ Overview of STOP mode

STOP mode is classified as one of standby modes. STOP mode is used to stop all oscillating operations. Enabling this mode stops all functions, excluding the Low Voltage Detection Circuit. This therefore allows data to be held with the minimum power consumption.

■ Relationships between CPU operation modes and consumption current values.

Figure 1-1 shows the relationships between CPU operation modes and consumption current values.

Figure 1-1 Relationships between CPU operation modes and consumption current values



<Note>

The figure above shows only the approximate consumption current values. The actual consumption current values vary depending on the oscillator and PLL starting conditions in each mode or the clock configuration of the selected frequency and other elements.

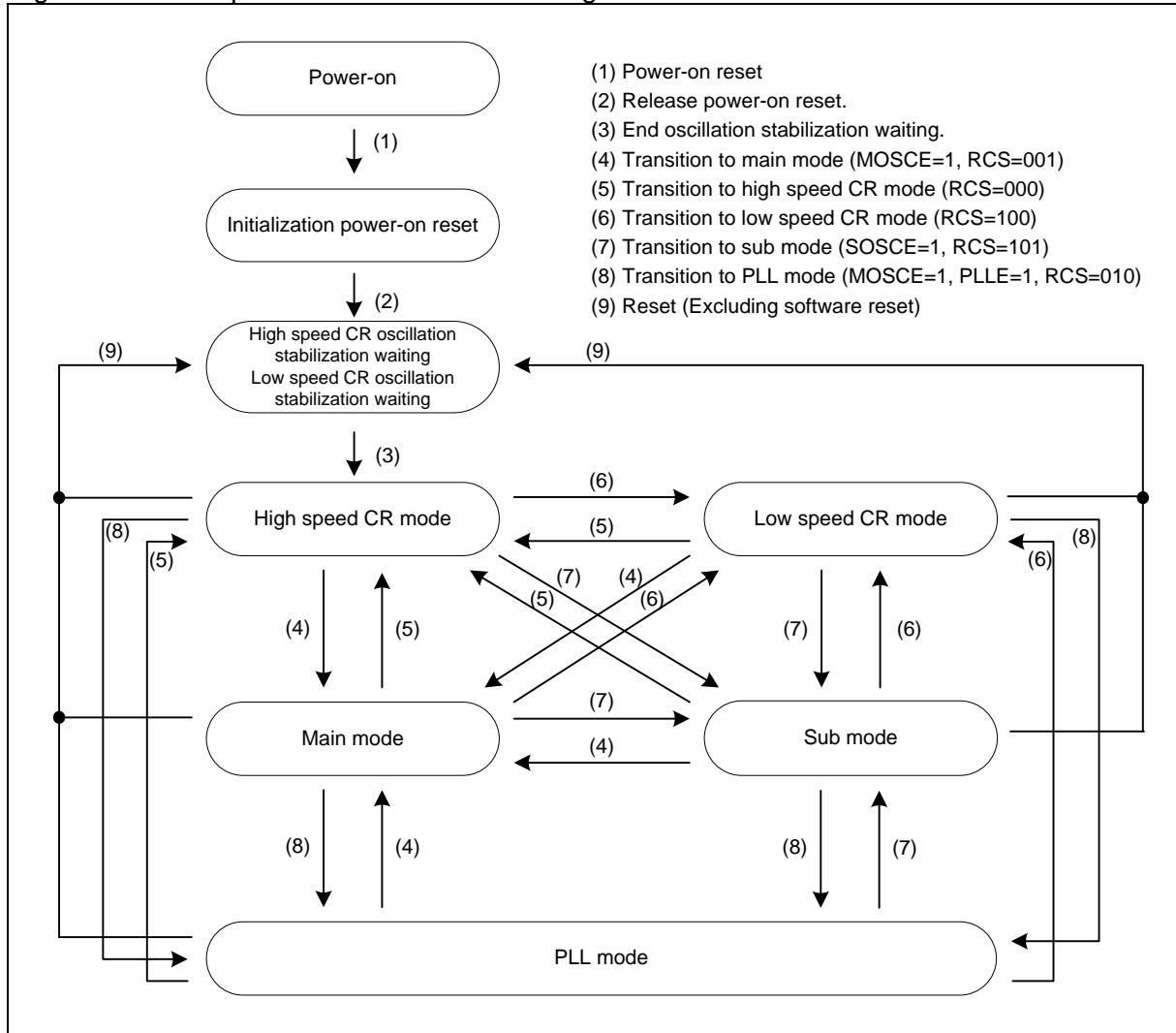
2. Configuration of CPU Operation Modes

This section explains the configuration of CPU operation modes.

■ CPU operation mode transition diagram

Figure 2-1 shows the CPU operation mode transition diagram.

Figure 2-1 CPU operation mode transition diagram



● High speed CR mode

In this mode, the high speed CR oscillator clock is used as a master clock.

● Main mode

In this mode, the main oscillator clock is used as a master clock.

● Low speed CR mode

In this mode, the low speed CR oscillator clock is used as a master clock.

● Sub mode

In this mode, the sub oscillator clock is used as a master clock.

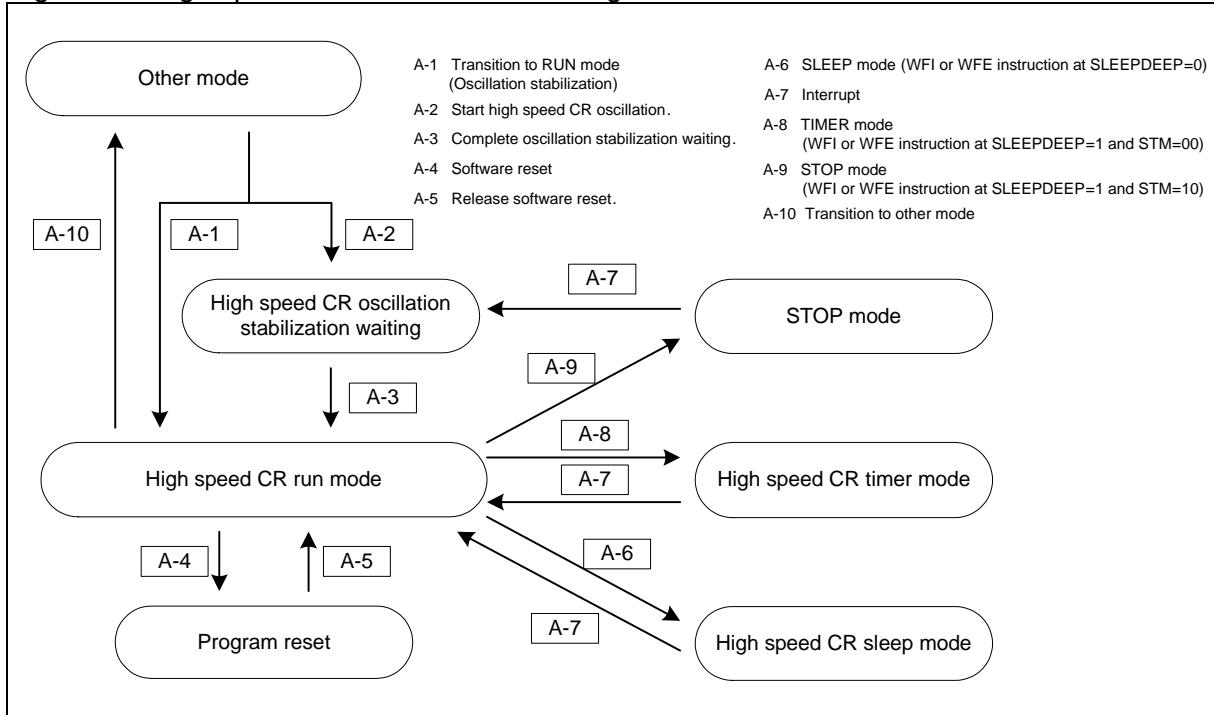
● PLL mode

In this mode, the PLL oscillator clock is used as a master clock.

■ High speed CR mode transition diagram

In high speed CR mode, the high speed CR oscillator clock is used as a master clock.

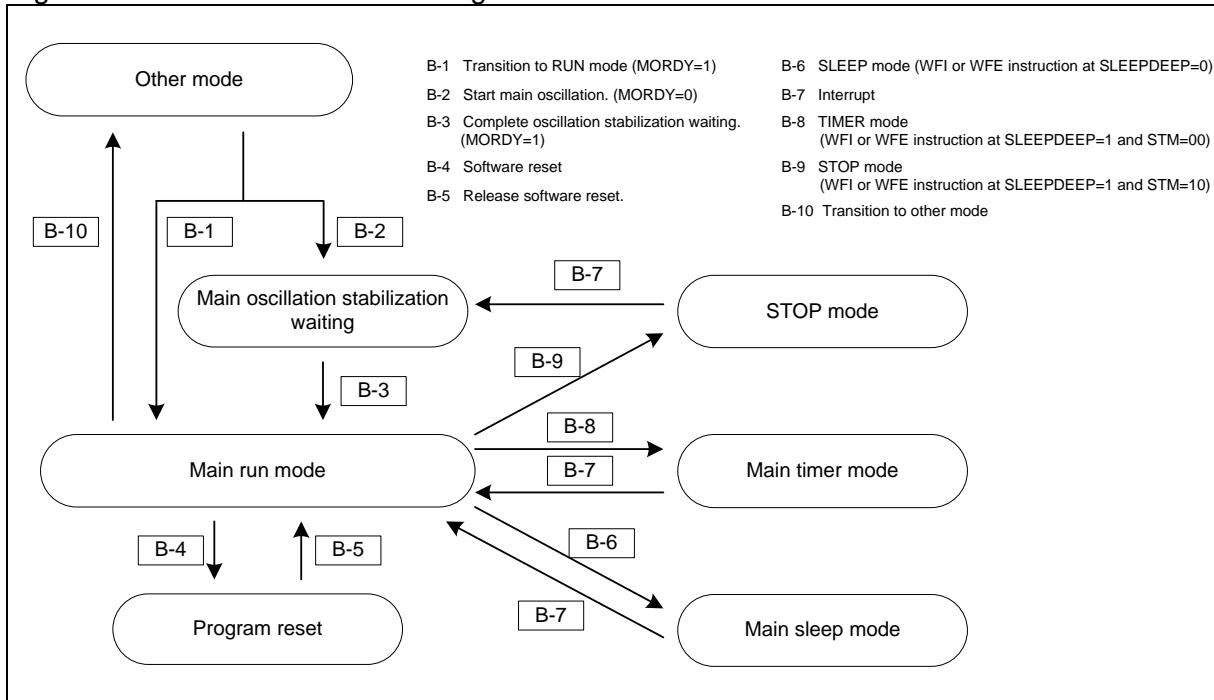
Figure 2-2 High speed CR mode transition diagram



■ Main mode transition diagram

In main mode, the main oscillator clock is used as a master clock.

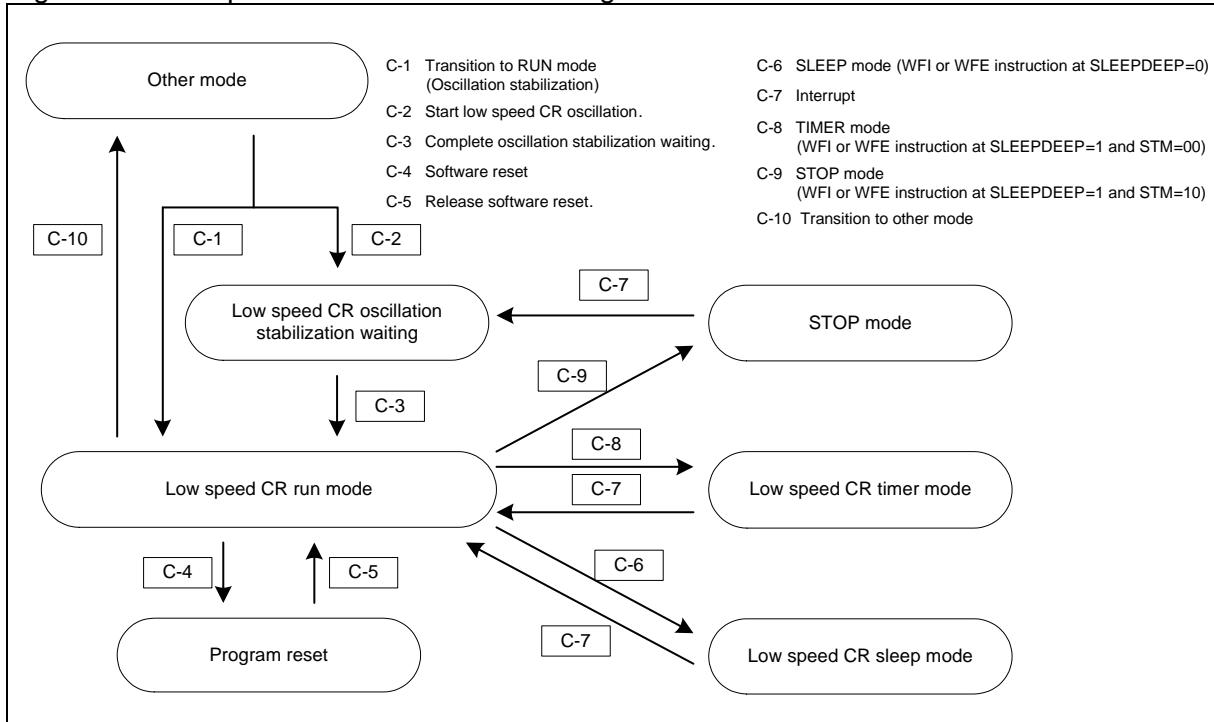
Figure 2-3 Main mode transition diagram



■ Low speed CR mode transition diagram

In low speed CR mode, the low speed CR oscillator clock is used as a master clock.

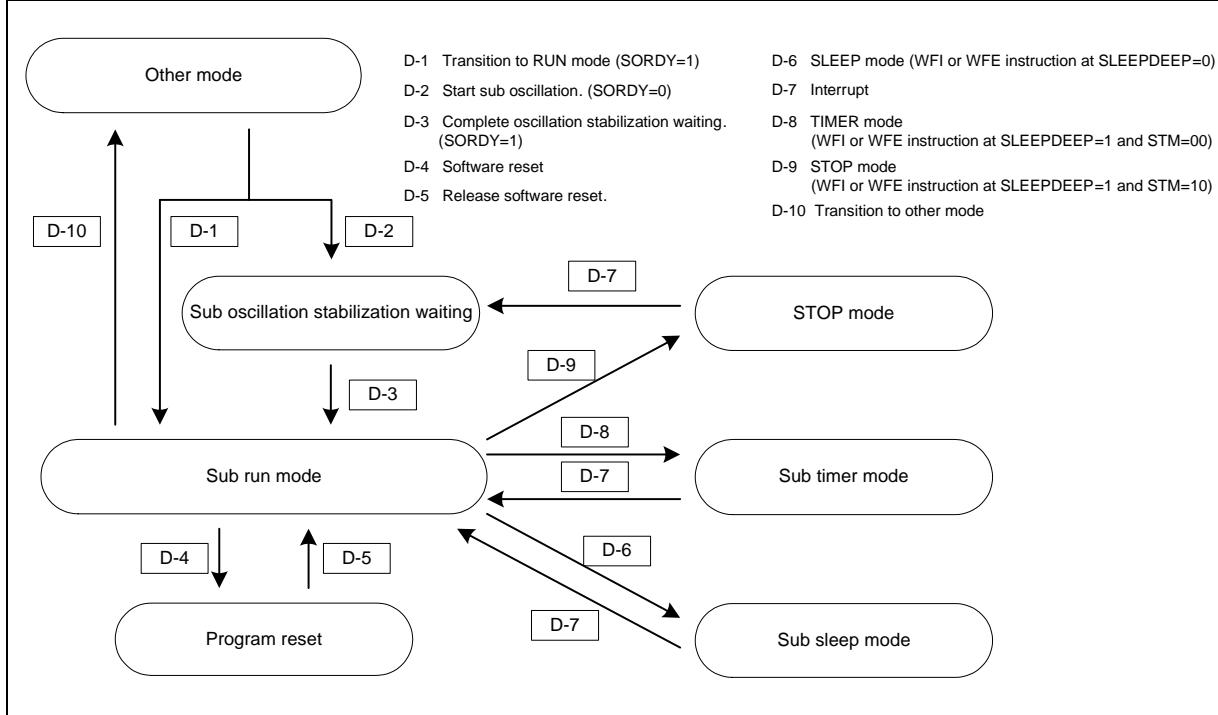
Figure 2-4 Low speed CR mode transition diagram



■ Sub mode transition diagram

In sub mode, the sub oscillator clock is used as a master clock.

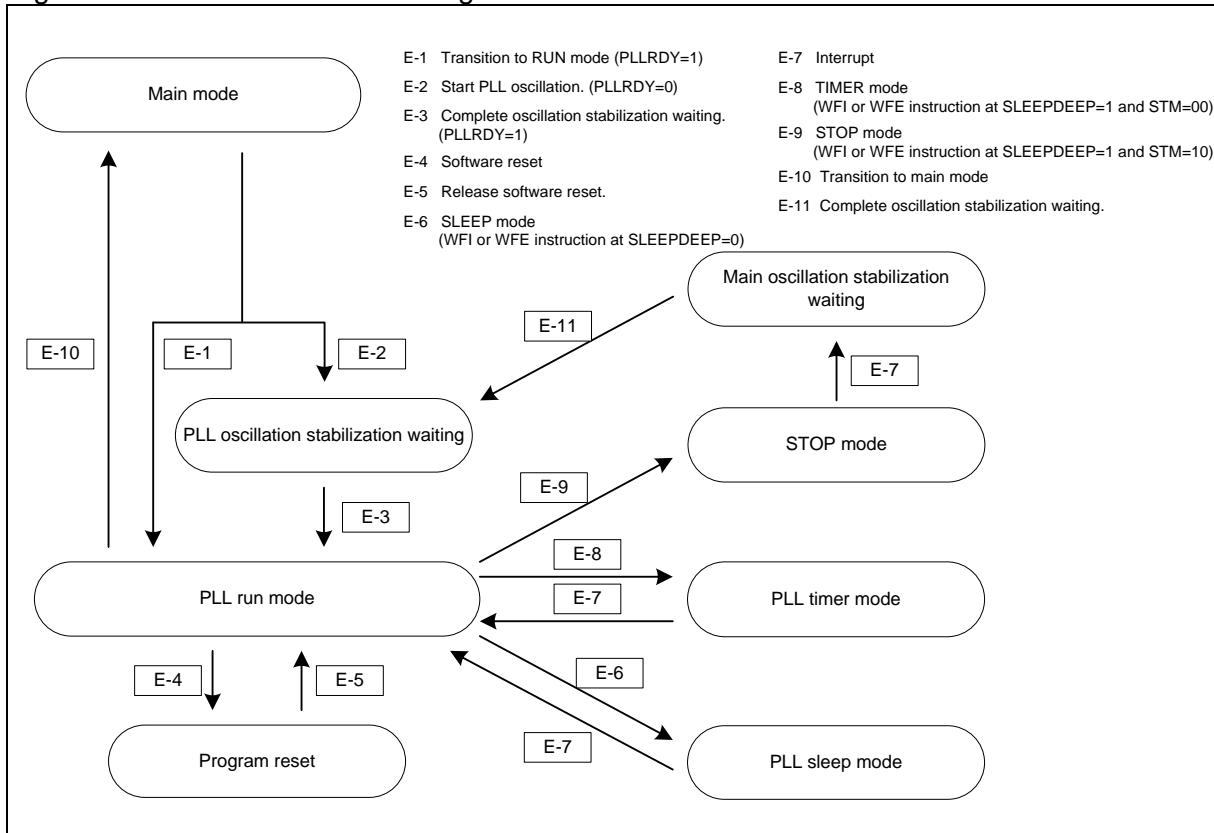
Figure 2-5 Sub mode transition diagram



■ PLL mode transition diagram

In PLL mode, the PLL clock is used as a master clock.

Figure 2-6 PLL mode transition diagram



MOSCE : MOSCE bit of System Clock Mode Control Register (SCM_CTL)

SOSCE : SOSCE bit of System Clock Mode Control Register (SCM_CTL)

PLLE : PLLE bit of System Clock Mode Control Register (SCM_CTL)

RCS : RCS bit of System Clock Mode Control Register (SCM_CTL)

MORDY : MORDY bit of System Clock Mode Status Register (SCM_STR)

SORDY : SORDY bit of System Clock Mode Status Register (SCM_STR)

PLRDY : PLRDY bit of System Clock Mode Status Register (SCM_STR)

* For the SCM_CTL and SCM_STR Registers, refer to Chapter "Clocks".

<Note>

To return from low speed CR timer mode, sub timer mode, or STOP mode, the voltage stabilization wait time (a few hundred of μ s) of the built-in regulator is ensured. After the wait time has lapsed, the system performs operations to return to each RUN mode.

3. Operations of Standby Modes

This section explains operations of standby modes.

Standby modes are classified into three types: SLEEP modes (high speed CR sleep, main sleep, PLL sleep, low speed CR sleep, and sub sleep), TIMER modes (high speed CR timer, main timer, PLL timer, low speed CR timer, and sub timer), and STOP mode.

■ Clock operation states in standby modes

The table below shows the states of the oscillator clock, CPU clock, AHB bus clock, and APB bus clock in SLEEP, TIMER, and STOP modes.

Table 3-1 Clock operation states in SLEEP modes

		SLEEP modes				
		High speed CR sleep mode	Main sleep mode	PLL sleep mode	Low speed CR sleep mode	Sub sleep mode
High speed CR clock	Operating	Operating	Operating	Operating	Stopped	Stopped
Main clock	Varies depending on the setting of the MOSCE bit.	Operating	Operating	Operating	Stopped	Stopped
PLL clock	Setting disabled	Varies depending on the setting of the PLLE bit.	Operating	Operating	Stopped	Stopped
Low speed CR clock	Operating	Operating	Operating	Operating	Operating	Operating
Sub clock	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Operating
USB PLL clock	Setting disabled	Varies depending on the setting of the UPLLEN bit.	Varies depending on the setting of the UPLLEN bit.	Varies depending on the setting of the UPLLEN bit.	Stopped	Stopped
CPU clock	Stopped					
AHB bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock	
APB0 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock	
APB1 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock	
* Whether or not operation is enabled is determined depending on the setting of the PBC1EN bit.						
APB2 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock	
* Whether or not operation is enabled is determined depending on the setting of the PBC2EN bit.						

Table 3-2 Clock operation states in TIMER modes

	TIMER modes				
	High speed CR timer mode	Main timer mode	PLL timer mode	Low speed CR timer mode	Sub timer mode
High speed CR clock	Operating	Operating	Operating	Stopped	Stopped
Main clock	Varies depending on the setting of the MOSCE bit.	Operating	Operating	Stopped	Stopped
PLL clock	Setting disabled	Varies depending on the setting of the PLLE bit.	Operating	Stopped	Stopped
Low speed CR clock	Operating	Operating	Operating	Operating	Operating
Sub clock	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Operating
USB PLL clock	Stopped	Stopped	Stopped	Stopped	Stopped
CPU clock	Stopped				
AHB bus clock	Stopped				
APB0 bus clock	Stopped				
APB1 bus clock	Stopped				
APB2 bus clock	Stopped				

Table 3-3 Clock operation states in STOP modes

	STOP mode
High speed CR clock	
Main clock	
PLL clock	
Low speed CR clock	
Sub clock	
USB PLL clock	Stopped
CPU clock	
AHB bus clock	
APB0 bus clock	
APB1 bus clock	
APB2 bus clock	

MOSCE : MOSCE bit of System Clock Mode Control Register (SCM_CTL)

SOSCE : SOSCE bit of System Clock Mode Control Register (SCM_CTL)

PLLE : PLLE bit of System Clock Mode Control Register (SCM_CTL)

UPLLEN : UPLLEN bit of USB-PLL Control Register-1 (UPCR1)

APBC1EN : APBC1EN bit of Peripheral Bus Clock Frequency Division Register (APBC1_PSR)

APBC2EN : APBC2EN bit of Peripheral Bus Clock Frequency Division Register (APBC2_PSR)

* For the SCM_CTL and SCM_STR Registers, refer to Chapter "Clocks".

■ Return factors from standby modes

The table below shows the factors by which the system returns from the SLEEP, TIMER, and STOP modes.

Table 3-4 Return factors from standby modes

	SLEEP mode	TIMER mode	STOP mode
Return factors by reset	<ul style="list-style-type: none"> · INITX pin input reset · Low-voltage detection reset · Software watchdog reset · Hardware watchdog reset · Clock failure detection reset · Anomalous frequency detection reset 	<ul style="list-style-type: none"> · INITX pin input reset · Low-voltage detection reset · Hardware watchdog reset · Clock failure detection reset 	<ul style="list-style-type: none"> · INITX pin input reset · Low-voltage detection reset
Return factors by interrupt	<ul style="list-style-type: none"> · Effective interrupt from each peripheral 	<ul style="list-style-type: none"> · NMI interrupt · External interrupt · Hardware watchdog timer interrupt · USB wake up interrupt · Watch counter interrupt · Low voltage detection interrupt 	<ul style="list-style-type: none"> · NMI interrupt · External interrupt · USB wake up interrupt · Low voltage detection interrupt

3.1. Operations of SLEEP modes (high speed CR sleep, main sleep, PLL sleep, low speed CR sleep, and sub sleep modes)

SLEEP mode is classified as one of standby modes. Enabling SLEEP mode stops CPU clocks, reducing the power consumption.

■ Functions of SLEEP mode

● CPU and internal memory

In SLEEP mode, the CPU clock is stopped. However, the AHB bus clock continues to be active. The internal memory active and data is held.

● Peripherals

The APB0 bus clock is still active in SLEEP mode. The states of the APB1 and APB2 bus clocks vary depending on the APBC1EN and APBC2EN settings. Peripherals are held in the state that is set at transition.

● Watch counter

The watch counter has no effect in SLEEP mode. It continues operations based on the configuration that is provided before the transition to SLEEP mode.

● Oscillator clocks

Table 3-1 (Clock operation states in SLEEP modes) shows the status of each oscillator clock.

● Reset and interrupt

Reset and interrupt are available to return from SLEEP mode.

● External bus

The external bus is still active in SLEEP mode.

● Status of pin

All pin settings are held in SLEEP mode.

■ SLEEP mode setting procedure

Execute the following steps to change to SLEEP mode.

1. Set "0" to the SLEEPDEEP bit of the Cortex-M3 System Control Register.
2. Execute the WFI or WFE instruction.

The system changes to the appropriate SLEEP mode according to the current clock mode indicated in the RCM bit of the System Clock Mode Control Register (SCM_CTL).

For the System Clock Mode Control Register (SCM_CTL), refer to Chapter "Clocks".

■ Return from SLEEP mode

The CPU returns from SLEEP mode in one of the following cases.

● Return by reset

If a reset (INITX pin input reset, low-voltage detection reset, software watchdog reset, hardware watchdog reset, clock failure detection reset, or anomalous frequency detection reset) occurs, the CPU changes to high speed CR run mode regardless of clock mode.

● Return by interrupt

If an effective interrupt is received from a peripheral in SLEEP mode, the CPU returns from SLEEP mode and changes to RUN mode to fit clock mode indicated in the RCM bit of SCM_CTL.

Table 3-5 Operation modes after the CPU returned from SLEEP mode by interrupt

	Status of master clock before transition to SLEEP mode				
	RCM=000 (High speed CR oscillator)	RCM=001 (Main oscillator)	RCM=010 (PLL oscillator)	RCM=100 (Low speed CR oscillator)	RCM=101 (Sub oscillator)
Operation modes after return by interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

RCM: RCM bit of System Clock Mode Status Register (SCM_STR)

* For the SCM_CTL and SCM_STR Registers, refer to Chapter "Clocks".

● Waiting for oscillation stabilization at return

When the CPU returns by reset, it waits for the stabilization of high and low speed CR clock oscillations. If the CPU returns by interrupt, it does not need to wait for oscillation to stabilize.

3.2. Operations of TIMER modes (high speed CR timer, main timer, PLL timer, low speed CR timer, and sub timer modes)

TIMER mode is used to stop supplying a base clock. This causes the CPU clock, AHB bus clock, and all APB bus clocks to be stopped, leading to the further reduction of power consumption. In this case, all functions are stopped, excluding the oscillators, PLL, hardware watchdog timer, watch counter, clock failure detector, and Low Voltage Detection Circuit.

■ Functions of TIMER mode

● CPU and internal memory

Enabling TIMER mode stops CPU clocks and AHB bus clocks supplied to the internal memory or DMA controller. The contents of the internal memory are held. The debug function is stopped.

● Peripherals

In TIMER mode, all APB clocks are stopped, and all resources, excluding the hardware watchdog timer, watch counter, clock failure detector, and Low Voltage Detection Circuit, are stopped in the last state.

● Watch counter

The watch counter has no effect in TIMER mode. It continues operations based on the configuration that is provided before the transition to TIMER mode.

● Oscillator clocks

Table 3-2 (Clock operation states in TIMER modes) shows the status of each oscillator clock.

● Reset and interrupt

Reset and interrupt are available to return from TIMER mode.

● External bus

The external bus is stopped in TIMER mode.

● Status of pin

The system can control whether to hold the state just before the external pin changes to TIMER mode or change to the high impedance state depending on the setting of the SPL bit in the Standby Mode Control Register (STB_CTL).

■ TIMER mode setting procedure

Execute the following steps to change to TIMER mode.

1. Write "0x1ACC" to the KEY bit and "0b00" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in TIMER mode.
2. Set "1" to the SLEEPDEEP bit of the Cortex-M3 System Control Register.
3. Execute the WFI or WFE instruction.

The system changes to the appropriate TIMER mode according to the current clock mode indicated in the RCM bit of the System Clock Mode Control Register (SCM_CTL).

■ Return from TIMER mode

The CPU returns from TIMER mode in one of the following cases.

● Return by reset

If a reset (INITX pin input reset, low-voltage detection reset, hardware watchdog reset, or clock failure detection reset) occurs, the CPU changes to high speed CR run mode regardless of clock mode.

Software watchdog reset and anomalous frequency detection reset are not available in this mode; therefore, the CPU cannot return by those resets.

● Return by interrupt

If an effective NMI interrupt, external interrupt, hardware watchdog timer interrupt, USB wake up interrupt, watch counter interrupt, or low voltage detection interrupt request is received in TIMER mode, the CPU returns from TIMER mode and changes to RUN mode to fit clock mode indicated in the RCM bit of SCM_CTL.

Table 3-6 Operation modes after the CPU returned from TIMER mode by interrupt

	Status of master clock before transition to TIMER mode				
	RCM=000 (High speed CR oscillator)	RCM=001 (Main oscillator)	RCM=010 (PLL oscillator)	RCM=100 (Low speed CR oscillator)	RCM=101 (Sub oscillator)
Operation modes after return by interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

● Waiting for oscillation stabilization at return

When the CPU returns by reset, it waits for the stabilization of high and low speed CR clock oscillations. If the CPU returns by interrupt, it does not need to wait for oscillation to stabilize.

● Waiting for the stabilization of the built-in regulator voltage at return

To return from low speed CR timer mode or sub timer mode by reset or interrupt, the voltage stabilization wait time (a few hundred of μ s) of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

<Notes>

- When an interrupt priority used for return is not set at a level to return the CPU, clock will be returned with the interrupt but the CPU remains stop state without returning. In order to do this, be sure to set the interrupt priority at a level which CPU is able to return.
 - If the transition to TIMER mode is made during debugging, as the clock to the CPU stops, a return to RUN mode cannot be performed by the ICE. Use a return by reset or interrupt.
 - In case of transitioning to the Low speed CR timer mode or Sub timer mode, ensure that the flash memory automatic programming algorithm is terminated before executing transition.
-

3.3. Operations of STOP mode

STOP mode is used to stop all oscillating operations. Enabling this mode stops all functions, excluding the Low Voltage Detection Circuit. This therefore allows data to be held with the minimum power consumption.

■ Functions of STOP mode

● CPU and internal memory

Enabling STOP mode stops CPU clocks and AHB bus clocks supplied to the internal memory or DMA controller. The contents of the internal memory are held. The debug function is stopped.

● Peripherals

All APB bus clocks are stopped, and all resources, excluding the Low Voltage Detection Circuit, are stopped in the last state.

● Oscillator clocks

All oscillator clocks are stopped.

● Reset and interrupt

Reset and interrupt are available to return from STOP mode.

● External bus

The external bus is stopped in STOP mode.

● Status of pin

The system controls whether to hold the state just before the external pin changes to STOP mode or change to the high impedance state depending on the setting of the SPL bit in the Standby Mode Control Register (STB_CTL).

■ STOP mode setting procedure

Execute the following steps to change to STOP mode.

1. Write "0x1ACC" to the KEY bit and "0b10" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in STOP mode.
2. Set "1" to the SLEEPDEEP bit of the Cortex-M3 System Control Register.
3. Execute the WFI or WFE instruction.

■ Return from STOP mode

The CPU returns from STOP mode in one of the following cases.

● Return by reset

If a reset (INITX pin input reset or low-voltage detection reset) occurs, the CPU changes to the high speed CR run mode regardless of clock mode.

Software watchdog reset, hardware watch dog reset, clock failure detection reset, and anomalous frequency detection reset are not available in this mode; therefore, the CPU cannot return by those resets.

● Return by interrupt

If an effective NMI interrupt, external interrupt, USB wake up interrupt, or low voltage detection interrupt request is received in STOP mode, the CPU returns from STOP mode and changes to RUN mode to fit clock mode indicated in the RCM bit of SCM_CTL.

Table 3-7 Operation modes after the CPU returned from the STOP mode by interrupt

	Status of master clock before changing to STOP mode				
	RCM=000 (High speed CR oscillator)	RCM=001 (Main oscillator)	RCM=010 (PLL oscillator)	RCM=100 (Low speed CR oscillator)	RCM=101 (Sub oscillator)
Operation modes after return by interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

● Waiting for oscillation stabilization at return

When the CPU returns by reset, it waits for the stabilization of high and low speed CR clock oscillations. If the CPU returns by interrupt, the oscillation stabilization wait state varies depending on the master clock that is output before the CPU changes to STOP mode as shown in Table 3-8.

Table 3-8 Waiting for oscillation to stabilize at return from STOP mode by interrupt

	Status of master clock before changing to STOP mode					
	RCM=000 (High speed CR oscillator)	RCM=001 (Main oscillator)	RCM=010 (PLL oscillator)	RCM=100 (Low speed CR oscillator)	RCM=101 (Sub oscillator)	
Oscillation stabilization waiting after return by interrupt	High speed CR clock	ON	ON	ON	OFF	OFF
	Main clock	MOSCE="0": OFF MOSCE="1": ON	ON	ON	OFF	OFF
	PLL clock	OFF	PLLE="0": OFF PLLE="1": ON	ON	OFF	OFF
	Low speed CR clock	ON	ON	ON	ON	ON
	Sub clock	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	ON

● Waiting for the stabilization of the built-in regulator voltage at return

When the CPU returns from STOP mode, the voltage stabilization wait time (a few hundred of μ s) of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

<Notes>

- When an interrupt priority used for return is not set at a level to return the CPU, clock will be returned with the interrupt but the CPU remains stop state without returning. In order to do this, be sure to set the interrupt priority at a level which CPU is able to return.
 - If the transition to STOP mode is made during debugging, as the clock to the CPU stops, a return to RUN mode cannot be performed by the ICE. Use a return by reset or interrupt.
 - In case of transitioning to the STOP mode, ensure that the flash memory automatic programming algorithm is terminated before executing transition.
-

4. Standby Mode Setting Procedure Examples

This section provides standby mode setting procedure examples.

Figure 4-1 Main timer mode setting procedure example

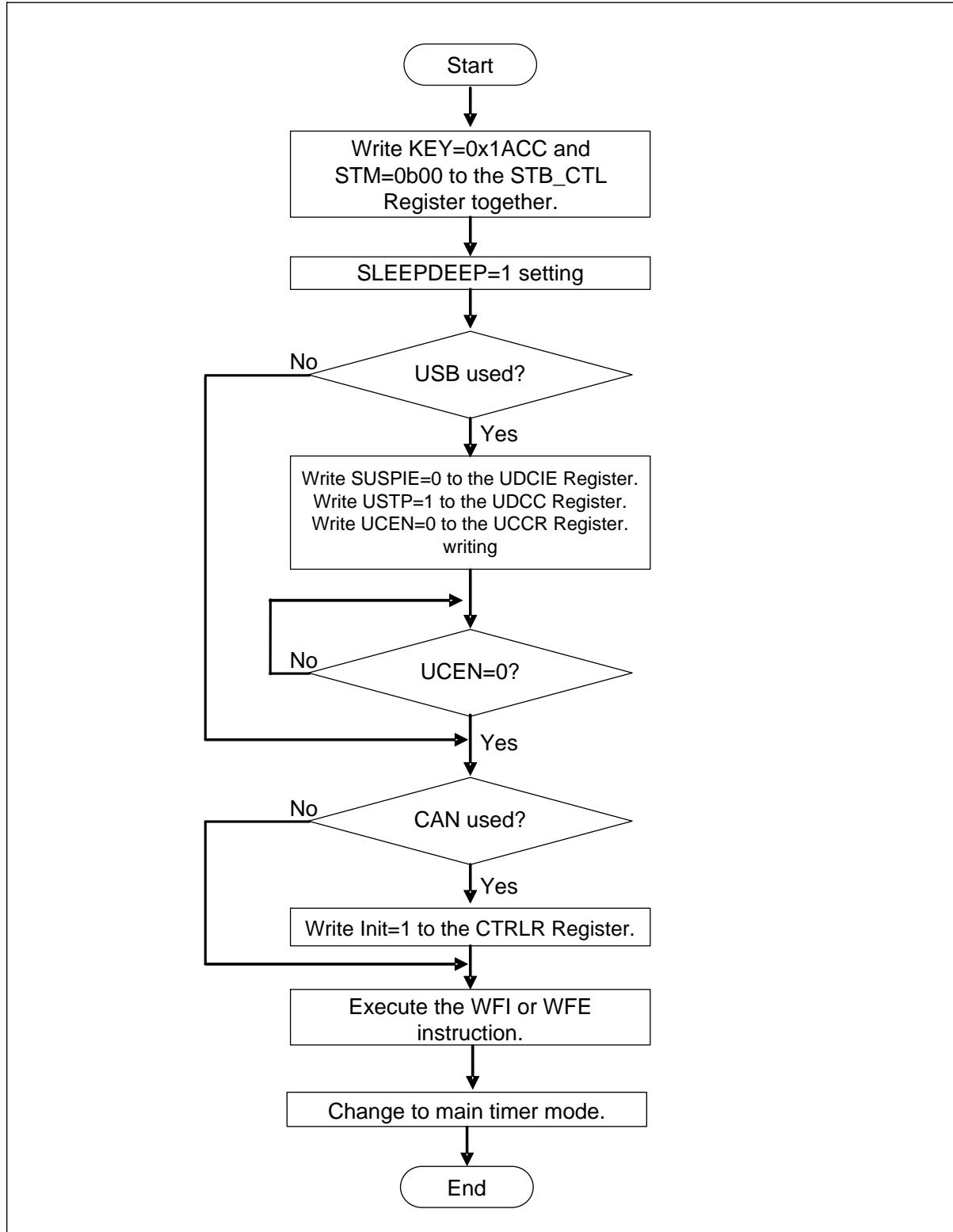
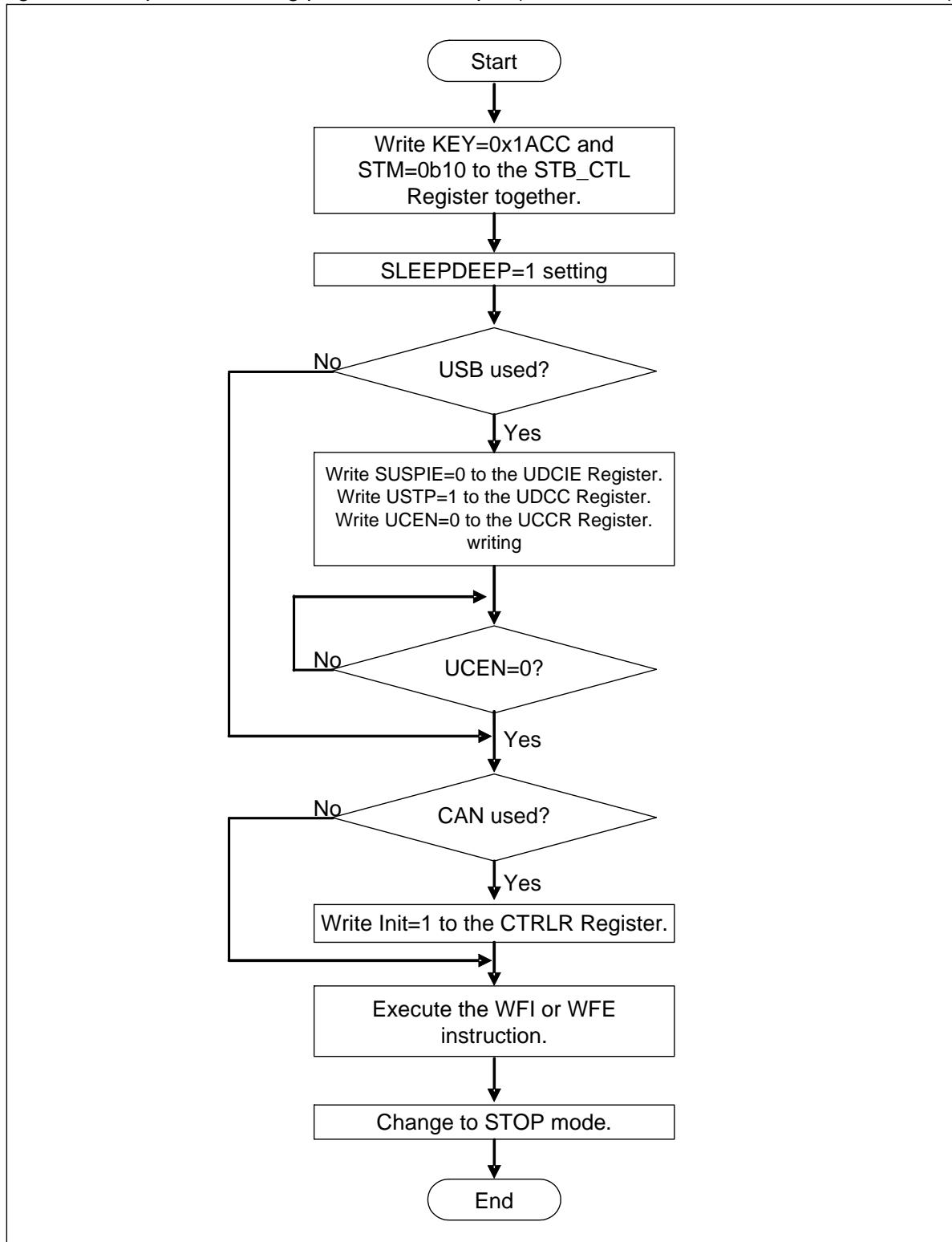


Figure 4-2 Stop mode setting procedure example (Main clock is selected as a master clock.)



<Note>

In case of transitioning to the STOP mode, ensure that the flash memory automatic programming algorithm is terminated before executing transition.

5. List of Low Power Consumption Registers

This section explains the configuration and functions of the registers used in standby mode.

■ List of Low Power Consumption Registers

Abbreviation	Register name	See
STB_CTL	Standby Mode Control Register	5.1

<Note>

For the Clock Mode Selection Register, refer to Chapter "Clocks".

5.1. Standby Mode Control Register (STB_CTL)

The Standby Mode Control Register controls TIMER or STOP mode. The value written to the SPL or STM bit is effective only when 0x1ACC is simultaneously written to the KEY bit.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	KEY															
Attribute	R/W															
Initial value	0x0000															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															
Attribute	R/W															
Initial value	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

[bit31:16] KEY: Standby mode control write control bit

This bit releases the SPL or STM bit writing control.

- The value written to the SPL or STM bit is effective only when 0x1ACC is written to the KEY bit.
- If a value other than 0x1ACC is written to the KEY bit, the value written to the SPL or STM bit is not effective.
- 0x0000 is always read in read mode.

[bit15:5] Reserved: Reserved bits

"0" is always set in read mode.

This bit has no effect in write mode.

[bit4] SPL: Standby pin level setting bit

This bit sets the status of external pin in TIMER or STOP mode.

Bit	Description
0	Holds the status of each external pin in TIMER or STOP mode. [Initial value]
1	Sets the status of each external pin to high impedance in TIMER or STOP mode.

[bit3:2] Reserved: Reserved bits

"0" is always set in read mode.

This bit has no effect in write mode.

[bit1:0] STM: Standby mode selection bit

This bit selects whether to change to TIMER or STOP mode.

Bit1	Bit0	Description
0	0	TIMER mode [Initial value]
0	1	Setting disabled
1	0	STOP mode
1	1	Setting disabled

<Note>

The value written to the SPL or STM bit in the Standby Mode Control Register (STB_CTL) is effective only when 0x1ACC is simultaneously written to the KEY bit. If a value other than 0x1ACC is written to the KEY bit, the value written to the SPL or STM bit becomes invalid.

CHAPTER: Interrupts

This chapter explains the interrupt controller and peripheral interrupt requests.

1. Overview
2. Structure
3. Exception and Interrupt Vectors
4. Registers
5. Usage Warnings

1. Overview

The interrupt controller determines the priority of interrupt requests and sends the requests to the CPU. The Cortex-M3 CPU core is equipped with the nested vectored interrupt controller (NVIC) internally within the core. Interrupt signals from several peripherals are aggregated and input to a single interrupt vector. The interrupt requests that have occurred can be checked using the interrupt request batch read register. Furthermore, for some of the interrupt sources, the interrupt requests can be configured to be converted into DMA request signals.

■ Features of the Nested Vectored Interrupt Controller (NVIC)

- 48 maskable peripheral interrupt channels (not including the 16 exception interrupts of Cortex-M3)
- 16 programmable interrupt priority levels (using 4-bit prioritized interrupts)
- Facilitates low-latency exception and interrupt handling
- Implements System Control Registers
- Supports non-maskable interrupt (NMI) input

The NVIC and the processor core interface are closely coupled, providing mechanisms that enable low-latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains the nested interrupt information to enable tail chaining of interrupts.

All interrupts are managed by the NVIC, including core exceptions. See "Chapter 5: Exceptions" and "Chapter 8: Nested Vectored Interrupt Controller" in the "Cortex-M3 Technical Reference Manual" from ARM for details on exceptions and NVIC.

<Note>

In the "Cortex-M3 Technical Reference Manual", all exception type:IRQ are defined as external interrupt inputs. In this manual, exception type:IRQ are expressed as peripheral interrupts. Peripheral interrupts include "External Interrupt and NMI Control Unit" interrupts from external pins and interrupts from peripheral resources within the LSI.

■ Interrupt Source Aggregation Function

The interrupt request signals from each peripheral resource are aggregated into 48 sources and input to the NVIC. Furthermore, the interrupt request signal from the external NMIX pin is logically OR'ed with the hardware watchdog interrupt signal and input to the NVIC.

■ Peripheral Interrupt Request Batch Read Function

The interrupt request batch read register allows the interrupt request signals from the peripheral resources aggregated into a single interrupt request signal to be read out at once. Reading this register makes it possible to check which interrupt request has occurred. However, the interrupt request flags cannot be cleared by using this function. Clear the interrupt request flags using the registers of each peripheral function.

■ Peripheral Interrupt Requests Output Selection Function

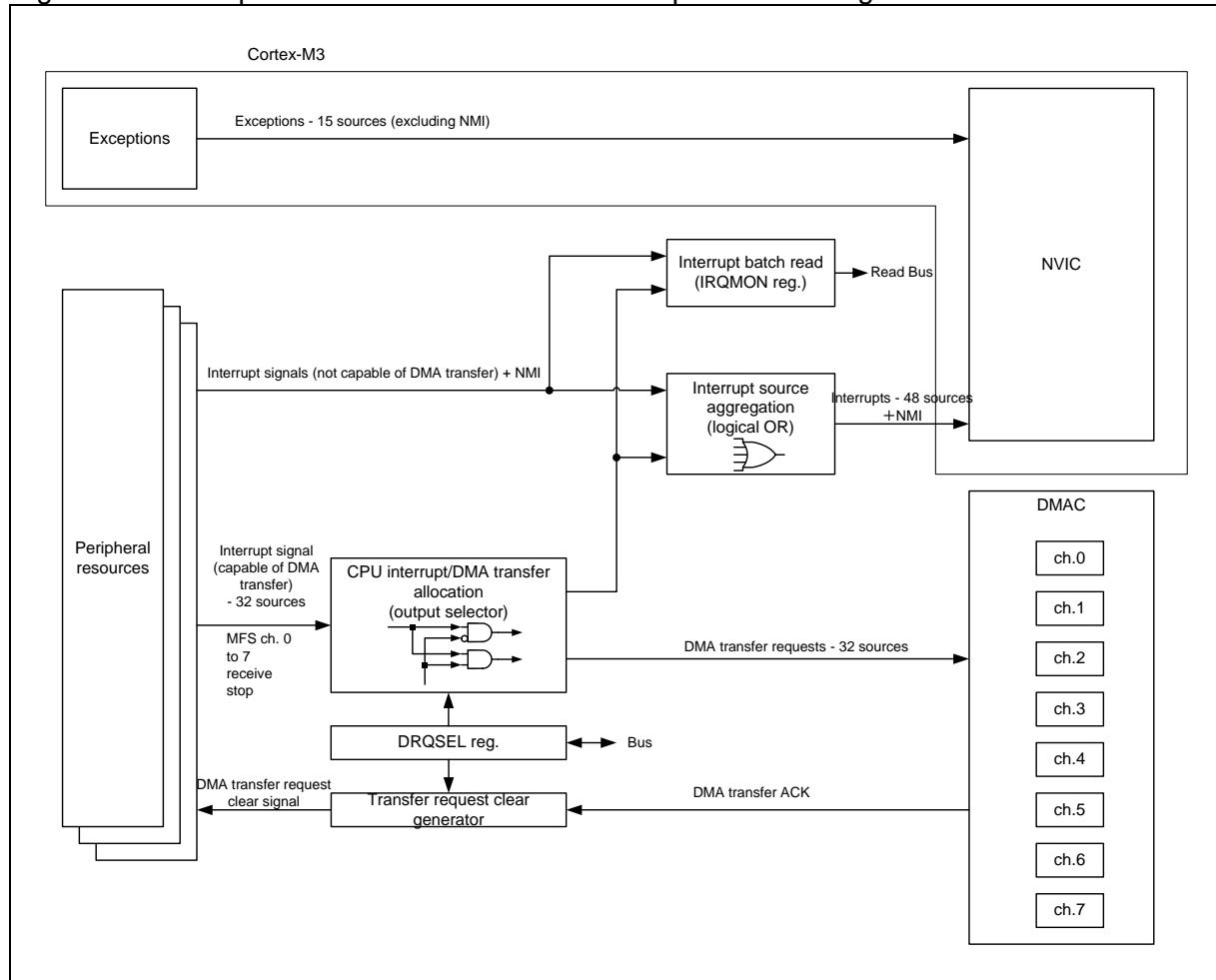
DMA transfers can be started using the 32 interrupt requests from the peripheral functions. Use the DRQSEL register to select whether the interrupt request signals from each of the peripheral resources are output to the CPU or output to the DMAC.

2. Structure

This section shows the structure of the relationship between the interrupt controller and DMA transfer requests.

■ Interrupt Controller and DMA Transfer Request Block Diagram

Figure 2-1 Interrupt controller and DMA transfer request block diagram



3. Exception and Interrupt Vectors

This section shows a vector table of the exceptions and interrupts input to the NVIC.

Table 3-1 Exception and interrupt vectors

Vector No.	IRQ No.	Exception and Interrupt Source	Vector Offset
0	-	Initial SP Value	0x00
1	-	Reset	0x04
2	-	Non-Maskable Interrupt (NMI) / Hardware Watchdog Timer	0x08
3	-	Hard Fault	0x0C
4	-	Memory Management	0x10
5	-	Bus Fault	0x14
6	-	Usage Fault	0x18
7-10	-	Reserved	0x1C - 0x2B
11	-	SVCALL (Supervisor Call)	0x2C
12	-	Debug Monitor	0x30
13	-	Reserved	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	Anomalous Frequency Detection by Clock Supervisor (FCS)	0x40
17	1	Software Watchdog Timer	0x44
18	2	Low Voltage Detector (LVD)	0x48
19	3	MFT unit0, unit1 Wave Form Generator / DTIF(Motor Emergency Stop)	0x4C
20	4	External Interrupt Request ch.0 to ch.7	0x50
21	5	External Interrupt Request ch.8 to ch.15	0x54
22	6	Dual Timer / Quadrature Position/Resolution Counter (QPRC) ch.0, ch.1	0x58
23	7	Reception Interrupt Request of Multi-Function Serial Interface ch.0	0x5C
24	8	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.0	0x60
25	9	Reception Interrupt Request of Multi-Function Serial Interface ch.1	0x64
26	10	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.1	0x68
27	11	Reception Interrupt Request of Multi-Function Serial Interface ch.2	0x6C
28	12	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.2	0x70
29	13	Reception Interrupt Request of Multi-Function Serial Interface ch.3	0x74
30	14	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.3	0x78
31	15	Reception Interrupt Request of Multi-Function Serial Interface ch.4	0x7C
32	16	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.4	0x80
33	17	Reception Interrupt Request of Multi-Function Serial Interface ch.5	0x84

Vector No.	IRQ No.	Exception and Interrupt Source	Vector Offset
34	18	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.5	0x88
35	19	Reception Interrupt Request of Multi-Function Serial Interface ch.6	0x8C
36	20	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.6	0x90
37	21	Reception Interrupt Request of Multi-Function Serial Interface ch.7	0x94
38	22	Transmission Interrupt Request and Status Interrupt Request of Multi-Function Serial Interface ch.7	0x98
39	23	PPG ch.0/2/4/8/10/12	0x9C
40	24	External Main OSC / External Sub OSC / Main PLL / PLL for USB / Watch Counter	0xA0
41	25	A/D Converter unit0	0xA4
42	26	A/D Converter unit1	0xA8
43	27	A/D Converter unit2	0xAC
44	28	MFT unit0, unit1 Free-run Timer	0xB0
45	29	MFT unit0, unit1 Input Capture	0xB4
46	30	MFT unit0, unit1 Output Compare	0xB8
47	31	Base Timer ch.0 to ch.7	0xBC
48	32	CAN ch.0	0xC0
49	33	CAN ch.1	0xC4
50	34	USB Function (DRQ of End Point 1 to 5) *	0xC8
51	35	USB Function (DRQI of End Point 0, DRQO and each status) / USB HOST (each status) *	0xCC
52	36	Reserved	0xD0
53	37	Reserved	0xD4
54	38	DMA Controller (DMAC) ch.0	0xD8
55	39	DMA Controller (DMAC) ch.1	0xDC
56	40	DMA Controller (DMAC) ch.2	0xE0
57	41	DMA Controller (DMAC) ch.3	0xE4
58	42	DMA Controller (DMAC) ch.4	0xE8
59	43	DMA Controller (DMAC) ch.5	0xEC
60	44	DMA Controller (DMAC) ch.6	0xF0
61	45	DMA Controller (DMAC) ch.7	0xF4
62	46	Reserved	0xF8
63	47	Reserved	0xFC

***: USB Interrupt Source**

Vector No.	IRQ No.	USB Interrupt Source	Flags
50	34	USB Function (DRQ of End Point 1 to 5)	DRQ (End Point 1 to 5)
51	35	USB Function (DRQI of End Point 0, DRQO and each status)	DRQI, DRQO, SPK, SUSP, SOF, BRST, CONF, WKUP
		USB HOST (each status)	DIRQ, URIRQ, RWKIRQ, CNNIRQ, SOFIRQ, CMPIRQ

The priorities of the exceptions for vectors no. 4 to 15 can be configured using the System Handler Priority Registers (address 0xE000ED18, 0xE000ED1C, 0xE000ED20) built into the NVIC. The priorities of the peripheral interrupts for vectors no. 16 and after can be configured using the Interrupt Priority Registers (address 0xE000E400 to 0xE000E42C) built into the NVIC.

The sources of the interrupts for vectors no. 2 and no. 16 to no. 63 can be checked using the batch read register. See "NVIC" in the "Technical Reference Manual" for details on the other exceptions and interrupts.

Furthermore, for the interrupts of vectors no. 2 and no. 16 to no. 63, the sources that are batch read may be a signal that multiple interrupt sources are logical OR'ed within each of the peripheral macros. See the descriptions of each of the peripheral resource interrupts for details.

4. Registers

This section explains the DMA transfer request selection register and the interrupt request batch read register.

■ DMA transfer request selection register and interrupt request batch read register list

Abbreviation	Register Name	See
DRQSEL	DMA Transfer Request Selection Register	4.1
EXC02MON	EXC02 Batch Read Register	4.2
IRQ00MON	IRQ00 Batch Read Register	4.3
IRQ01MON	IRQ01 Batch Read Register	4.4
IRQ02MON	IRQ02 Batch Read Register	4.5
IRQ03MON	IRQ03 Batch Read Register	4.6
IRQ04MON	IRQ04 Batch Read Register	4.7
IRQ05MON	IRQ05 Batch Read Register	4.7
IRQ06MON	IRQ06 Batch Read Register	4.8
IRQ07MON	IRQ07 Batch Read Register	4.9
IRQ08MON	IRQ08 Batch Read Register	4.10
IRQ09MON	IRQ09 Batch Read Register	4.9
IRQ10MON	IRQ10 Batch Read Register	4.10
IRQ11MON	IRQ11 Batch Read Register	4.9
IRQ12MON	IRQ12 Batch Read Register	4.10
IRQ13MON	IRQ13 Batch Read Register	4.9
IRQ14MON	IRQ14 Batch Read Register	4.10
IRQ15MON	IRQ15 Batch Read Register	4.9
IRQ16MON	IRQ16 Batch Read Register	4.10
IRQ17MON	IRQ17 Batch Read Register	4.9
IRQ18MON	IRQ18 Batch Read Register	4.10
IRQ19MON	IRQ19 Batch Read Register	4.9
IRQ20MON	IRQ20 Batch Read Register	4.10
IRQ21MON	IRQ21 Batch Read Register	4.9
IRQ22MON	IRQ22 Batch Read Register	4.10
IRQ23MON	IRQ23 Batch Read Register	4.11
IRQ24MON	IRQ24 Batch Read Register	4.12
IRQ25MON	IRQ25 Batch Read Register	4.13
IRQ26MON	IRQ26 Batch Read Register	4.13

Abbreviation	Register Name	See
IRQ27MON	IRQ27 Batch Read Register	4.13
IRQ28MON	IRQ28 Batch Read Register	4.14
IRQ29MON	IRQ29 Batch Read Register	4.15
IRQ30MON	IRQ30 Batch Read Register	4.16
IRQ31MON	IRQ31 Batch Read Register	4.17
IRQ32MON	IRQ32 Batch Read Register	4.18
IRQ33MON	IRQ33 Batch Read Register	4.18
IRQ34MON	IRQ34 Batch Read Register	4.19
IRQ35MON	IRQ35 Batch Read Register	4.20
IRQ36MON	IRQ36 Batch Read Register	4.21
IRQ37MON	IRQ37 Batch Read Register	4.21
IRQ38MON	IRQ38 Batch Read Register	4.22
IRQ39MON	IRQ39 Batch Read Register	4.22
IRQ40MON	IRQ40 Batch Read Register	4.22
IRQ41MON	IRQ41 Batch Read Register	4.22
IRQ42MON	IRQ42 Batch Read Register	4.22
IRQ43MON	IRQ43 Batch Read Register	4.22
IRQ44MON	IRQ44 Batch Read Register	4.22
IRQ45MON	IRQ45 Batch Read Register	4.22
IRQ46MON	IRQ46 Batch Read Register	4.23
IRQ47MON	IRQ47 Batch Read Register	4.23

See "Chapter 8: Nested Vectored Interrupt Controller" in the "Cortex-M3 Technical Reference Manual" for details on the registers in the NVIC.

4.1. DMA Request Selection Register (DRQSEL)

The DMA Request Selection Register (DRQSEL) selects whether interrupt signals that can start DMA transfers are output as interrupt requests to the CPU or output as transfer requests to the DMAC. If selected as a transfer request to the DMAC, the bit in the interrupt request batch read register (IRQxxMON, xx=00 to 47) that corresponds to the interrupt signal is "0".

bit	31	0
Field	DRQSEL[31:0]	
Attribute	R/W	
Initial value	0x00000000	

[bit31:0] DRQSEL :

bit no.	bit	Description
31	0	The interrupt of the external interrupt ch.3 is output as a request to the CPU.
	1	The interrupt of the external interrupt ch. 3 is output as a transfer request to the DMAC.
30	0	The interrupt of the external interrupt ch. 2 is output as a request to the CPU.
	1	The interrupt of the external interrupt ch. 2 is output as a transfer request to the DMAC.
29	0	The interrupt of the external interrupt ch. 1 is output as a request to the CPU.
	1	The interrupt of the external interrupt ch. 1 is output as a transfer request to the DMAC.
28	0	The interrupt of the external interrupt ch. 0 is output as a request to the CPU.
	1	The interrupt of the external interrupt ch. 0 is output as a transfer request to the DMAC.
27	0	The transmission interrupt of the MFS ch. 7 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 7 is output as a transfer request to the DMAC.
26	0	The reception interrupt of the MFS ch. 7 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 7 is output as a transfer request to the DMAC.
25	0	The transmission interrupt of the MFS ch. 6 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 6 is output as a transfer request to the DMAC.
24	0	The reception interrupt of the MFS ch. 6 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 6 is output as a transfer request to the DMAC.
23	0	The transmission interrupt of the MFS ch. 5 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 5 is output as a transfer request to the DMAC.
22	0	The reception interrupt of the MFS ch. 5 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 5 is output as a transfer request to the DMAC.
21	0	The transmission interrupt of the MFS ch. 4 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 4 is output as a transfer request to the DMAC.
20	0	The reception interrupt of the MFS ch. 4 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 4 is output as a transfer request to the DMAC.

bit no.	bit	Description
19	0	The transmission interrupt of the MFS ch. 3 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 3 is output as a transfer request to the DMAC.
18	0	The reception interrupt of the MFS ch. 3 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 3 is output as a transfer request to the DMAC.
17	0	The transmission interrupt of the MFS ch. 2 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 2 is output as a transfer request to the DMAC.
16	0	The reception interrupt of the MFS ch. 2 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 2 is output as a transfer request to the DMAC.
15	0	The transmission interrupt of the MFS ch. 1 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 1 is output as a transfer request to the DMAC.
14	0	The reception interrupt of the MFS ch. 1 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 1 is output as a transfer request to the DMAC.
13	0	The transmission interrupt of the MFS ch. 0 is output as a request to the CPU.
	1	The transmission interrupt of the MFS ch. 0 is output as a transfer request to the DMAC.
12	0	The reception interrupt of the MFS ch. 0 is output as a request to the CPU.
	1	The reception interrupt of the MFS ch. 0 is output as a transfer request to the DMAC.
11	0	The IRQ 0 interrupt of the base timer ch. 6 is output as a request to the CPU.
	1	The IRQ 0 interrupt of the base timer ch. 6 is output as a transfer request to the DMAC.
10	0	The IRQ 0 interrupt of the base timer ch. 4 is output as a request to the CPU.
	1	The IRQ 0 interrupt of the base timer ch. 4 is output as a transfer request to the DMAC.
9	0	The IRQ 0 interrupt of the base timer ch. 2 is output as a request to the CPU.
	1	The IRQ 0 interrupt of the base timer ch. 2 is output as a transfer request to the DMAC.
8	0	The IRQ 0 interrupt of the base timer ch. 0 is output as a request to the CPU.
	1	The IRQ 0 interrupt of the base timer ch. 0 is output as a transfer request to the DMAC.
7	0	The scan conversion interrupt of the A/D converter unit 2 is output as a request to the CPU.
	1	The scan conversion interrupt of the A/D converter unit 2 is output as a transfer request to the DMAC.
6	0	The scan conversion interrupt of the A/D converter unit 1 is output as a request to the CPU.
	1	The scan conversion interrupt of the A/D converter unit 1 is output as a transfer request to the DMAC.
5	0	The scan conversion interrupt of the A/D converter unit 0 is output as a request to the CPU.
	1	The scan conversion interrupt of the A/D converter unit 0 is output as a transfer request to the DMAC.
4	0	The EP5 DRQ interrupt of the USB ch. 0 is output as a request to the CPU.
	1	The EP5 DRQ interrupt of the USB ch. 0 is output as a transfer request to the DMAC.
3	0	The EP4 DRQ interrupt of the USB ch. 0 is output as a request to the CPU.
	1	The EP4 DRQ interrupt of the USB ch. 0 is output as a transfer request to the DMAC.
2	0	The EP3 DRQ interrupt of the USB ch. 0 is output as a request to the CPU.
	1	The EP3 DRQ interrupt of the USB ch. 0 is output as a transfer request to the DMAC.

bit no.	bit	Description
1	0	The EP2 DRQ interrupt of the USB ch. 0 is output as a request to the CPU.
	1	The EP2 DRQ interrupt of the USB ch. 0 is output as a transfer request to the DMAC.
0	0	The EP1 DRQ interrupt of the USB ch. 0 is output as a request to the CPU.
	1	The EP1 DRQ interrupt of the USB ch. 0 is output as a transfer request to the DMAC.

MFS: Multifunction serial interface

<Notes>

- When changing the DRQSEL settings during a DMA transfer, clear all of the interrupt request signals from the peripherals before making the change.
 - DMA transfers cannot be started from hardware for interrupt signals not specified in the DRQSEL settings. See "CHAPTER DMAC" for details on the DMAC transfer modes.
-

4.2. EXC02 Batch Read Register (EXC02MON)

EXC02MON indicates all of the interrupt requests allocated to interrupt vector no. 2.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														HWINT	NMI
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:2] Reserved: Reserved bits

Reads out "0".

[bit1] HWINT:

bit	Description
0	No hardware watchdog timer interrupt request
1	Hardware watchdog timer interrupt request

[bit0] NMI:

bit	Description
0	No external NMIX pin interrupt request
1	external NMIX pin interrupt request

4.3. IRQ00 Batch Read Register (IRQ00MON)

IRQ00MON indicates all of the interrupt requests allocated to interrupt vector no. 16.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															FCSINT
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved bits
Reads out "0".

[bit0] FCSINT:

bit	Description
0	No anomalous frequency detection by CSV interrupt request
1	Anomalous frequency detection by CSV interrupt request

4.4. IRQ01 Batch Read Register (IRQ01MON)

IRQ01MON indicates all of the interrupt requests allocated to interrupt vector no. 17.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															SWWDTINT
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved bits
Reads out "0".

[bit0] SWWDTINT:

bit	Description
0	No software watchdog timer interrupt request
1	Software watchdog timer interrupt request

4.5. IRQ02 Batch Read Register (IRQ02MON)

IRQ02MON indicates all of the interrupt requests allocated to interrupt vector no. 18.

bit	31		16													
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															LVDINT
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved bits
Reads out "0".

[bit0] LVDINT:

bit	Description
0	No low voltage detection (LVD) interrupt request
1	Low voltage detection (LVD) interrupt request

4.6. IRQ03 Batch Read Register (IRQ03MON)

IRQ03MON indicates all of the interrupt requests allocated to interrupt vector no. 19.

bit	31	Field	Reserved	16
Attribute			R	
Initial value			0x0000	
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Field	Reserved	WAVE1INT WAVE0INT
Attribute	R R R R R R R R R R R R R R R R R R R R	Initial value	0 0	

[bit31:8] Reserved: Reserved bits

Reads out "0".

[bit7:4] WAVE1INT:

bit no.	bit	Description
7	0	No WFG timer 54 interrupt request in MFT unit 1
	1	WFG timer 54 interrupt request in MFT unit 1
6	0	No WFG timer 32 interrupt request in MFT unit 1
	1	WFG timer 32 interrupt request in MFT unit 1
5	0	No WFG timer 10 interrupt request in MFT unit 1
	1	WFG timer 10 interrupt request in MFT unit 1
4	0	No DTIF (motor emergency stop) interrupt request in MFT unit 1
	1	DTIF (motor emergency stop) interrupt request in MFT unit 1

[bit3:0] WAVE0INT:

bit no.	bit	Description
3	0	No WFG timer 54 interrupt request in MFT unit 0
	1	WFG timer 54 interrupt request in MFT unit 0
2	0	No WFG timer 32 interrupt request in MFT unit 0
	1	WFG timer 32 interrupt request in MFT unit 0
1	0	No WFG timer 10 interrupt request in MFT unit 0
	1	WFG timer 10 interrupt request in MFT unit 0
0	0	No DTIF (motor emergency stop) interrupt request in MFT unit 0
	1	DTIF (motor emergency stop) interrupt request in MFT unit 0

4.7. IRQ04/05 Batch Read Register (IRQxxMON)

IRQ04MON indicates all of the interrupt requests allocated to interrupt vector no. 20.
IRQ05MON indicates all of the interrupt requests allocated to interrupt vector no. 21.

IRQ04MON shows the status of the interrupt requests on the external interrupt from ch.0 to ch.7.
IRQ05MON shows the status of the interrupt requests on the external interrupt from ch.8 to ch.15.

bit	31	16
Field	Reserved	
Attribute	R	
Initial value	0x0000	
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Field	Reserved	EXTINT
Attribute	R R R R R R R R R R R R R R R R R R R R	
Initial value	0 0	

[bit31:8] Reserved: Reserved bits

Reads out "0".

[bit7:0] EXTINT:

bit no.	bit	Description
7	0	No interrupt request on external interrupt ch. 7 (ch. 15)
	1	Interrupt request on external interrupt ch. 7 (ch. 15)
6	0	No interrupt request on external interrupt ch. 6 (ch. 14)
	1	Interrupt request on external interrupt ch. 6 (ch. 14)
5	0	No interrupt request on external interrupt ch. 5 (ch. 14)
	1	Interrupt request on external interrupt ch. 5 (ch. 14)
4	0	No interrupt request on external interrupt ch. 4 (ch. 13)
	1	Interrupt request on external interrupt ch. 4 (ch. 13)
3	0	No interrupt request on external interrupt ch. 3 (ch. 12)
	1	Interrupt request on external interrupt ch. 3 (ch. 12)
2	0	No interrupt request on external interrupt ch. 2 (ch. 11)
	1	Interrupt request on external interrupt ch. 2 (ch. 11)
1	0	No interrupt request on external interrupt ch. 1 (ch. 10)
	1	Interrupt request on external interrupt ch. 1 (ch. 10)
0	0	No interrupt request on external interrupt ch. 0 (ch. 9)
	1	Interrupt request on external interrupt ch. 0 (ch. 9)

Values in parentheses in the table show the case for IRQ05MON.

If DMA transfer requests are selected by the DRQSEL register, the corresponding EXTINT bit is "0".

4.8. IRQ06 Batch Read Register (IRQ06MON)

IRQ06MON indicates all of the interrupt requests allocated to interrupt vector no. 22.

bit	31	16		
Field	Reserved			
Attribute	R			
Initial value	0x0000			
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Field	Reserved	QUD1INT	QUD0INT	TIMINT
Attribute	R R			
Initial value	0 0			

[bit31:14] Reserved: Reserved bits

Reads out "0".

[bit13:8] QUD1INT:

bit no.	bit	Description
13	0	No PC match & RC match interrupt request on QPRC ch. 1
	1	PC match & RC match interrupt request on QPRC ch. 1
12	0	No interrupt request detected RC out of range on QPRC ch. 1
	1	Interrupt request detected RC out of range on QPRC ch. 1
11	0	No PC counter direction change interrupt request on QPRC ch. 1
	1	PC counter direction change interrupt request on QPRC ch. 1
10	0	No overflow/underflow/zero index interrupt request on QPRC ch. 1
	1	Overflow/underflow/zero index interrupt request on QPRC ch. 1
9	0	No PC&RC match interrupt request on QPRC ch. 1
	1	PC&RC match interrupt request on QPRC ch. 1
8	0	No PC match interrupt request on QPRC ch. 1
	1	PC match interrupt request on QPRC ch. 1

[bit7:2] QUD0INT:

bit no.	bit	Description
7	0	No PC match & RC match interrupt request on QPRC ch. 0
	1	PC match & RC match interrupt request on QPRC ch. 0
6	0	No interrupt request detected RC out of range on QPRC ch. 0
	1	Interrupt request detected RC out of range on QPRC ch. 0
5	0	No PC counter direction change interrupt request on QPRC ch. 0
	1	PC counter direction change interrupt request on QPRC ch. 0
4	0	No overflow/underflow/zero index interrupt request on QPRC ch. 0
	1	Overflow/underflow/zero index interrupt request on QPRC ch. 0
3	0	No PC&RC match interrupt request on QPRC ch. 0
	1	PC&RC match interrupt request on QPRC ch. 0
2	0	No PC match interrupt request on QPRC ch. 0
	1	PC match interrupt request on QPRC ch. 0

[bit1:0] TIMINT:

bit no.	bit	Description
1	0	No dual timer TIMINT2 interrupt request
	1	Dual timer TIMINT2 interrupt request
0	0	No dual timer TIMINT1 interrupt request
	1	Dual timer TIMINT1 interrupt request

4.9. IRQ07/09/11/13/15/17/19/21 Batch Read Register (IRQxxMON)

IRQ07MON indicates all of the interrupt requests allocated to interrupt vector no. 23.
 IRQ09MON indicates all of the interrupt requests allocated to interrupt vector no. 25.
 IRQ11MON indicates all of the interrupt requests allocated to interrupt vector no. 27.
 IRQ13MON indicates all of the interrupt requests allocated to interrupt vector no. 29.
 IRQ15MON indicates all of the interrupt requests allocated to interrupt vector no. 31.
 IRQ17MON indicates all of the interrupt requests allocated to interrupt vector no. 33.
 IRQ19MON indicates all of the interrupt requests allocated to interrupt vector no. 35.
 IRQ21MON indicates all of the interrupt requests allocated to interrupt vector no. 37.

IRQ07MON shows the status of the reception interrupt request on MFS ch.0.
 IRQ09MON shows the status of the reception interrupt request on MFS ch.1.
 IRQ11MON shows the status of the reception interrupt request on MFS ch.2.
 IRQ13MON shows the status of the reception interrupt request on MFS ch.3.
 IRQ15MON shows the status of the reception interrupt request on MFS ch.4.
 IRQ17MON shows the status of the reception interrupt request on MFS ch.5.
 IRQ19MON shows the status of the reception interrupt request on MFS ch.6.
 IRQ21MON shows the status of the reception interrupt request on MFS ch.7.

bit	31	16													
Field	Reserved														
Attribute	R														
Initial value	0x0000														
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
Field	Reserved														FMSINT
Attribute	R R														
Initial value	0 0														

[bit31:1] Reserved: Reserved bits
 Reads out "0".

[bit0] MFSINT:

bit	Description
0	No reception interrupt request on the corresponding MFS channel.
1	Reception interrupt request on the corresponding MFS channel.

If DMA transfer requests are selected by the DRQSEL register, the corresponding MFSINT bit is "0".

4.10. IRQ08/10/12/14/16/18/20/22 Batch Read Register (IRQxxMON)

IRQ08MON indicates all of the interrupt requests allocated to interrupt vector no. 24.
 IRQ10MON indicates all of the interrupt requests allocated to interrupt vector no. 26.
 IRQ12MON indicates all of the interrupt requests allocated to interrupt vector no. 28.
 IRQ14MON indicates all of the interrupt requests allocated to interrupt vector no. 30.
 IRQ16MON indicates all of the interrupt requests allocated to interrupt vector no. 32.
 IRQ18MON indicates all of the interrupt requests allocated to interrupt vector no. 34.
 IRQ20MON indicates all of the interrupt requests allocated to interrupt vector no. 36.
 IRQ22MON indicates all of the interrupt requests allocated to interrupt vector no. 38.

IRQ08MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.0.
 IRQ10MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.1.
 IRQ12MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.2.
 IRQ14MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.3.
 IRQ16MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.4.
 IRQ18MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.5.
 IRQ20MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.6.
 IRQ22MON shows the status of the transmission interrupt request and the status interrupt request on MFS ch.7.

bit	31	16													
Field	Reserved														
Attribute	R														
Initial value	0x0000														
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
Field	Reserved														MFSINT
Attribute	R R R R R R R R R R R R R R R R R R R R														
Initial value	0 0														

[bit31:2] Reserved: Reserved bits
 Reads out "0".

[bit1:0] MFSINT:

bit no.	bit	Description
1	0	No status interrupt request on the corresponding MFS channel.
	1	Status interrupt request on the corresponding MFS channel.
0	0	No transmission interrupt request on the corresponding MFS channel.
	1	Transmission interrupt request on the corresponding MFS channel.

If DMA transfer requests are selected by the DRQSEL register, the corresponding MFSINT bit is "0".

4.11. IRQ23 Batch Read Register (IRQ23MON)

IRQ23MON indicates all of the interrupt requests allocated to interrupt vector no. 39.

bit	31	16																
Field	Reserved																	
Attribute	R																	
Initial value	0x0000																	
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved										PPGINT							
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

[bit31:6] Reserved: Reserved bits

Reads out "0".

[bit5:0] PPGINT:

bit no.	bit	Description
5	0	No interrupt request on PPG ch. 12
	1	Interrupt request on PPG ch. 12
4	0	No interrupt request on PPG ch. 10
	1	Interrupt request on PPG ch. 10
3	0	No interrupt request on PPG ch. 8
	1	Interrupt request on PPG ch. 8
2	0	No interrupt request on PPG ch. 4
	1	Interrupt request on PPG ch. 4
1	0	No interrupt request on PPG ch. 2
	1	Interrupt request on PPG ch. 2
0	0	No interrupt request on PPG ch. 0
	1	Interrupt request on PPG ch. 0

4.12. IRQ24 Batch Read Register (IRQ24MON)

IRQ24MON indicates all of the interrupt requests allocated to interrupt vector no. 40.

bit	31	16
Field	Reserved	
Attribute	R	
Initial value	0x0000	
bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Field	Reserved	
Attribute	R R R R R R R R R R R R R R R R R R R R	
Initial value	0 0	

[bit31:5] Reserved: Reserved bits

Reads out "0".

[bit4] WCINT:

bit	Description
0	No watch counter interrupt request
1	Watch counter interrupt request

[bit3] UPLLINT:

bit	Description
0	No stabilization wait completion interrupt request for USB PLL oscillation
1	Stabilization wait completion interrupt request for USB PLL oscillation

[bit2] MPOLLINT:

bit	Description
0	No stabilization wait completion interrupt request for PLL oscillation
1	Stabilization wait completion interrupt request for PLL oscillation

[bit1] SOSCINT:

bit	Description
0	No stabilization wait completion interrupt request for sub-clock oscillation
1	Stabilization wait completion interrupt request for sub-clock oscillation

[bit0] MOSCINT:

bit	Description
0	No stabilization wait completion interrupt request for main clock oscillation
1	Stabilization wait completion interrupt request for main clock oscillation

4.13. IRQ25/26/27 Batch Read Register (IRQxxMON)

IRQ25MON indicates all of the interrupt requests allocated to interrupt vector no. 41.
 IRQ26MON indicates all of the interrupt requests allocated to interrupt vector no. 42.
 IRQ27MON indicates all of the interrupt requests allocated to interrupt vector no. 43.

IRQ25MON shows the status of the interrupt request in A/D converter unit 0.

IRQ26MON shows the status of the interrupt request in A/D converter unit 1.

IRQ27MON shows the status of the interrupt request in A/D converter unit 2.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												ADCINT			
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:4] Reserved: Reserved bits

Reads out "0".

[bit3:0] ADCINT:

bit no.	bit	Description
3	0	No conversion result comparison interrupt request in the corresponding A/D converter unit.
	1	Conversion result comparison interrupt request in the corresponding A/D converter unit.
2	0	No FIFO overrun interrupt request in the corresponding A/D converter unit.
	1	FIFO overrun interrupt request in the corresponding A/D converter unit.
1	0	No scan conversion interrupt request in the corresponding A/D converter unit.
	1	Scan conversion interrupt request in the corresponding A/D converter unit.
0	0	No priority conversion interrupt request in the corresponding A/D converter unit.
	1	Priority conversion interrupt request in the corresponding A/D converter unit.

If DMA transfer requests are selected by the DRQSEL register, the corresponding ADCINT bit is "0".

4.14. IRQ28 Batch Read Register (IRQ28MON)

IRQ28MON indicates all of the interrupt requests allocated to interrupt vector no. 44.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				FRT1INT				FRT0INT							
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved bits
Reads out "0".

[bit11:6] FRT1INT:

bit no.	bit	Description
11	0	No zero detection interrupt request on the free run timer ch. 2 in the MFT unit 1
	1	Zero detection interrupt request on the free run timer ch. 2 in the MFT unit 1
10	0	No zero detection interrupt request on the free run timer ch. 1 in the MFT unit 1
	1	Zero detection interrupt request on the free run timer ch. 1 in the MFT unit 1
9	0	No zero detection interrupt request on the free run timer ch. 0 in the MFT unit 1
	1	Zero detection interrupt request on the free run timer ch. 0 in the MFT unit 1
8	0	No peak value detection interrupt request on the free run timer ch. 2 in the MFT unit 1
	1	Peak value detection interrupt request on the free run timer ch. 2 in the MFT unit 1
7	0	No peak value detection interrupt request on the free run timer ch. 1 in the MFT unit 1
	1	Peak value detection interrupt request on the free run timer ch. 1 in the MFT unit 1
6	0	No peak value detection interrupt request on the free run timer ch. 0 in the MFT unit 1
	1	Peak value detection interrupt request on the free run timer ch. 0 in the MFT unit 1

[bit5:0] FRT0INT:

bit no.	bit	Description
5	0	No zero detection interrupt request on the free run timer ch. 2 in the MFT unit 0
	1	Zero detection interrupt request on the free run timer ch. 2 in the MFT unit 0
4	0	No zero detection interrupt request on the free run timer ch. 1 in the MFT unit 0
	1	Zero detection interrupt request on the free run timer ch. 1 in the MFT unit 0
3	0	No zero detection interrupt request on the free run timer ch. 0 in the MFT unit 0
	1	Zero detection interrupt request on the free run timer ch. 0 in the MFT unit 0
2	0	No peak value detection interrupt request on the free run timer ch. 2 in the MFT unit 0
	1	Peak value detection interrupt request on the free run timer ch. 2 in the MFT unit 0
1	0	No peak value detection interrupt request on the free run timer ch. 1 in the MFT unit 0
	1	Peak value detection interrupt request on the free run timer ch. 1 in the MFT unit 0
0	0	No peak value detection interrupt request on the free run timer ch. 0 in the MFT unit 0
	1	Peak value detection interrupt request on the free run timer ch. 0 in the MFT unit 0

4.15. IRQ29 Batch Read Register (IRQ29MON)

IRQ29MON indicates all of the interrupt requests allocated to interrupt vector no. 45.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								ICU1INT				ICU0INT			
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:8] Reserved: Reserved bits

Reads out "0".

[bit7:4] ICU1INT:

bit no.	bit	Description
7	0	No interrupt request on the input capture ch. 3 in the MFT unit 1
	1	Interrupt request on the input capture ch. 3 in the MFT unit 1
6	0	No interrupt request on the input capture ch. 2 in the MFT unit 1
	1	Interrupt request on the input capture ch. 2 in the MFT unit 1
5	0	No interrupt request on the input capture ch. 1 in the MFT unit 1
	1	Interrupt request on the input capture ch. 1 in the MFT unit 1
4	0	No interrupt request on the input capture ch. 0 in the MFT unit 1
	1	Interrupt request on the input capture ch. 0 in the MFT unit 1

[bit3:0] ICU0INT:

bit no.	bit	Description
3	0	No interrupt request on the input capture ch. 3 in the MFT unit 0
	1	Interrupt request on the input capture ch. 3 in the MFT unit 0
2	0	No interrupt request on the input capture ch. 2 in the MFT unit 0
	1	Interrupt request on the input capture ch. 2 in the MFT unit 0
1	0	No interrupt request on the input capture ch. 1 in the MFT unit 0
	1	Interrupt request on the input capture ch. 1 in the MFT unit 0
0	0	No interrupt request on the input capture ch. 0 in the MFT unit 0
	1	Interrupt request on the input capture ch. 0 in the MFT unit 0

4.16. IRQ30 Batch Read Register (IRQ30MON)

IRQ30MON indicates all of the interrupt requests allocated to interrupt vector no. 46.

bit	31	16															
Field	Reserved																
Attribute	R																
Initial value	0x0000																
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved				OCU1INT					OCU0INT							
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved bits
Reads out "0".

[bit11:6] OCU1INT:

bit no.	bit	Description
11	0	No interrupt request on the output compare ch. 5 in the MFT unit 1
	1	Interrupt request on the output compare ch. 5 in the MFT unit 1
10	0	No interrupt request on the output compare ch. 4 in the MFT unit 1
	1	Interrupt request on the output compare ch. 4 in the MFT unit 1
9	0	No interrupt request on the output compare ch. 3 in the MFT unit 1
	1	Interrupt request on the output compare ch. 3 in the MFT unit 1
8	0	No interrupt request on the output compare ch. 2 in the MFT unit 1
	1	Interrupt request on the output compare ch. 2 in the MFT unit 1
7	0	No interrupt request on the output compare ch. 1 in the MFT unit 1
	1	Interrupt request on the output compare ch. 1 in the MFT unit 1
6	0	No interrupt request on the output compare ch. 0 in the MFT unit 1
	1	Interrupt request on the output compare ch. 0 in the MFT unit 1

[bit5:0] OCU0INT:

bit no.	bit	Description
5	0	No interrupt request on the output compare ch. 5 in the MFT unit 0
	1	Interrupt request on the output compare ch. 5 in the MFT unit 0
4	0	No interrupt request on the output compare ch. 4 in the MFT unit 0
	1	Interrupt request on the output compare ch. 4 in the MFT unit 0
3	0	No interrupt request on the output compare ch. 3 in the MFT unit 0
	1	Interrupt request on the output compare ch. 3 in the MFT unit 0
2	0	No interrupt request on the output compare ch. 2 in the MFT unit 0
	1	Interrupt request on the output compare ch. 2 in the MFT unit 0
1	0	No interrupt request on the output compare ch. 1 in the MFT unit 0
	1	Interrupt request on the output compare ch. 1 in the MFT unit 0
0	0	No interrupt request on the output compare ch. 0 in the MFT unit 0
	1	Interrupt request on the output compare ch. 0 in the MFT unit 0

4.17. IRQ31 Batch Read Register (IRQ31MON)

IRQ31MON indicates all of the interrupt requests allocated to interrupt vector no. 47.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BTINT															
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved bits
Reads out "0".

[bit15:0] BTINT:

bit no.	bit	Description
15	0	No IRQ1 interrupt request on the base timer ch. 7
	1	IRQ1 interrupt request on the base timer ch. 7
14	0	No IRQ0 interrupt request on the base timer ch. 7
	1	IRQ0 interrupt request on the base timer ch. 7
13	0	No IRQ1 interrupt request on the base timer ch. 6
	1	IRQ1 interrupt request on the base timer ch. 6
12	0	No IRQ0 interrupt request on the base timer ch. 6
	1	IRQ0 interrupt request on the base timer ch. 6
11	0	No IRQ1 interrupt request on the base timer ch. 5
	1	IRQ1 interrupt request on the base timer ch. 5
10	0	No IRQ0 interrupt request on the base timer ch. 5
	1	IRQ0 interrupt request on the base timer ch. 5
9	0	No IRQ1 interrupt request on the base timer ch. 4
	1	IRQ1 interrupt request on the base timer ch. 4
8	0	No IRQ0 interrupt request on the base timer ch. 4
	1	IRQ0 interrupt request on the base timer ch. 4
7	0	No IRQ1 interrupt request on the base timer ch. 3
	1	IRQ1 interrupt request on the base timer ch. 3
6	0	No IRQ0 interrupt request on the base timer ch. 3
	1	IRQ0 interrupt request on the base timer ch. 3
5	0	No IRQ1 interrupt request on the base timer ch. 2
	1	IRQ1 interrupt request on the base timer ch. 2

bit no.	bit	Description
4	0	No IRQ0 interrupt request on the base timer ch. 2
	1	IRQ0 interrupt request on the base timer ch. 2
3	0	No IRQ1 interrupt request on the base timer ch. 1
	1	IRQ1 interrupt request on the base timer ch. 1
2	0	No IRQ0 interrupt request on the base timer ch. 1
	1	IRQ0 interrupt request on the base timer ch. 1
1	0	No IRQ1 interrupt request on the base timer ch. 0
	1	IRQ1 interrupt request on the base timer ch. 0
0	0	No IRQ0 interrupt request on the base timer ch. 0
	1	IRQ0 interrupt request on the base timer ch. 0

If DMA transfer requests are selected by the DRQSEL register, the corresponding BTINT bit is "0".

As shown in the Table 4-1, base timer interrupt sources IRQ0 and IRQ1 differ depending on the base timer function to be used.

Table 4-1 Interrupt sources for each function of the base timer

Function	Interrupt Source IRQ0	Interrupt Source IRQ1
16-bit PWM timer	Underflow detection/ duty match detection	Timer start trigger detection
16-bit PPG timer	Underflow detection	Timer start trigger detection
16/32-bit reload timer	Underflow detection	Timer start trigger detection
16/32-bit PWC timer	Overflow detection	Measurement finished detection

4.18. IRQ32/33 Batch Read Register (IRQxxMON)

IRQ32MON indicates all of the interrupt requests allocated to interrupt vector no. 48.
IRQ33MON indicates all of the interrupt requests allocated to interrupt vector no. 49.

IRQ32MON shows the status of the interrupt request on the CAN ch. 0.

IRQ33MON shows the status of the interrupt request on the CAN ch. 1.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															CANINT
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved bits

Reads out "0".

[bit0] CANINT:

bit	Description
0	No interrupt request on the corresponding CAN channel.
1	Interrupt request on the corresponding CAN channel.

4.19. IRQ34 Batch Read Register (IRQ34MON)

IRQ34MON indicates all of the interrupt requests allocated to interrupt vector no. 50.

bit	31	16															
Field	Reserved																
Attribute	R																
Initial value	0x0000																
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved										USB0INT						
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:5] Reserved: Reserved bits

Reads out "0".

[bit4:0] USB0INT:

bit no.	bit	Description
4	0	No Endpoint 5 DRQ interrupt request on the USB ch. 0
	1	Endpoint 5 DRQ interrupt request on the USB ch. 0
3	0	No Endpoint 4 DRQ interrupt request on the USB ch. 0
	1	Endpoint 4 DRQ interrupt request on the USB ch. 0
2	0	No Endpoint 3 DRQ interrupt request on the USB ch. 0
	1	Endpoint 3 DRQ interrupt request on the USB ch. 0
1	0	No Endpoint 2 DRQ interrupt request on the USB ch. 0
	1	Endpoint 2 DRQ interrupt request on the USB ch. 0
0	0	No Endpoint 1 DRQ interrupt request on the USB ch. 0
	1	Endpoint 1 DRQ interrupt request on the USB ch. 0

If DMA transfer requests are selected by the DRQSEL register, the corresponding USB0INT bit is "0".

4.20. IRQ35 Batch Read Register (IRQ35MON)

IRQ35MON indicates all of the interrupt requests allocated to interrupt vector no. 51.

bit	31	16														
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								USB0INT							
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:6] Reserved: Reserved bits

Reads out "0".

[bit5:0] USB0INT:

bit no.	bit	Description
5	0	No status (SOFIRQ, CMPIRO) interrupt request on the USB ch. 0
	1	Status (SOFIRQ, CMPIRO) interrupt request on the USB ch. 0
4	0	No status (DIRQ, URIRQ, RWKIRQ, CNNIRQ) interrupt request on the USB ch. 0
	1	Status (DIRQ, URIRQ, RWKIRQ, CNNIRQ) interrupt request on the USB ch. 0
3	0	No status (SPK) interrupt request on the USB ch. 0
	1	Status (SPK) interrupt request on the USB ch. 0
2	0	No status (SUSP, SOF, BRST, CONF, WKUP) interrupt request on the USB ch. 0
	1	Status (SUSP, SOF, BRST, CONF, WKUP) interrupt request on the USB ch. 0
1	0	No Endpoint 0 DRQO interrupt request on the USB ch. 0
	1	Endpoint 0 DRQO interrupt request on the USB ch. 0
0	0	No Endpoint 0 DRQI interrupt request on the USB ch. 0
	1	Endpoint 0 DRQI interrupt request on the USB ch. 0

4.21. IRQ36/37 Batch Read Register (IRQxxMON)

IRQ36MON indicates all of the interrupt requests allocated to interrupt vector no. 52.
IRQ37MON indicates all of the interrupt requests allocated to interrupt vector no. 53.

bit	31	16
Field		Reserved
Attribute		R
Initial value		0x0000
bit	15	0
Field		Reserved
Attribute		R
Initial value		0x0000

[bit31:0] Reserved: Reserved bits
Reads out "0".

4.22. IRQ38/39/40/41/42/43/44/45 Batch Read Register (IRQxxMON)

IRQ38MON indicates all of the interrupt requests allocated to interrupt vector no. 54.
 IRQ39MON indicates all of the interrupt requests allocated to interrupt vector no. 55.
 IRQ40MON indicates all of the interrupt requests allocated to interrupt vector no. 56.
 IRQ41MON indicates all of the interrupt requests allocated to interrupt vector no. 57.
 IRQ42MON indicates all of the interrupt requests allocated to interrupt vector no. 58.
 IRQ43MON indicates all of the interrupt requests allocated to interrupt vector no. 59.
 IRQ44MON indicates all of the interrupt requests allocated to interrupt vector no. 60.
 IRQ45MON indicates all of the interrupt requests allocated to interrupt vector no. 61.

IRQ38MON shows the status of the interrupt request on the DMA controller ch.0.
 IRQ39MON shows the status of the interrupt request on the DMA controller ch.1.
 IRQ40MON shows the status of the interrupt request on the DMA controller ch.2.
 IRQ41MON shows the status of the interrupt request on the DMA controller ch.3.
 IRQ42MON shows the status of the interrupt request on the DMA controller ch.4.
 IRQ43MON shows the status of the interrupt request on the DMA controller ch.5.
 IRQ44MON shows the status of the interrupt request on the DMA controller ch.6.
 IRQ45MON shows the status of the interrupt request on the DMA controller ch.7.

bit	31		16													
Field	Reserved															
Attribute	R															
Initial value	0x0000															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															DMAINT
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:1] Reserved: Reserved bits
 Reads out "0".

[bit0] DMAINT:

bit	Description
0	No interrupt request on the corresponding DMA controller channel.
1	Interrupt request on the corresponding DMA controller channel.

4.23. IRQ46/47 Batch Read Register (IRQxxMON)

IRQ46MON indicates all of the interrupt requests allocated to interrupt vector no. 62.
IRQ47MON indicates all of the interrupt requests allocated to interrupt vector no. 63.

bit	31	16
Field		Reserved
Attribute		R
Initial value		0x0000
bit	15	0
Field		Reserved
Attribute		R
Initial value		0x0000

[bit31:0] Reserved: Reserved bits
Reads out "0".

5. Usage Warnings

Be careful with the following points when using the interrupt controller.

- The interrupt request signals from each of the peripheral resources are notified by level. When exiting from interrupt processing, always clear the interrupt request.
- The NMIX pin is allocated shared with a general-purpose port. The initial value after reset is released is configured as a general-purpose port, and the NMI input is masked. In order to use the NMI, enable NMI by configuring the port setting. See the chapter of "External Interrupt and NMI Control Unit" for details.
- See the chapters for each of the macros for the correlation between the specific event detection registers and interrupt enable registers in each peripheral resource.

Chapter: External Interrupt and NMI Control Sections

This chapter explains the functions and operations of the external interrupt and NMI control sections.

-
- 1. Overview
 - 2. Block Diagram
 - 3. Operations and Setting Procedure Examples
 - 4. Registers

CODE: 9BFEXTINT-E02.1_FW12-J1.03

1. Overview

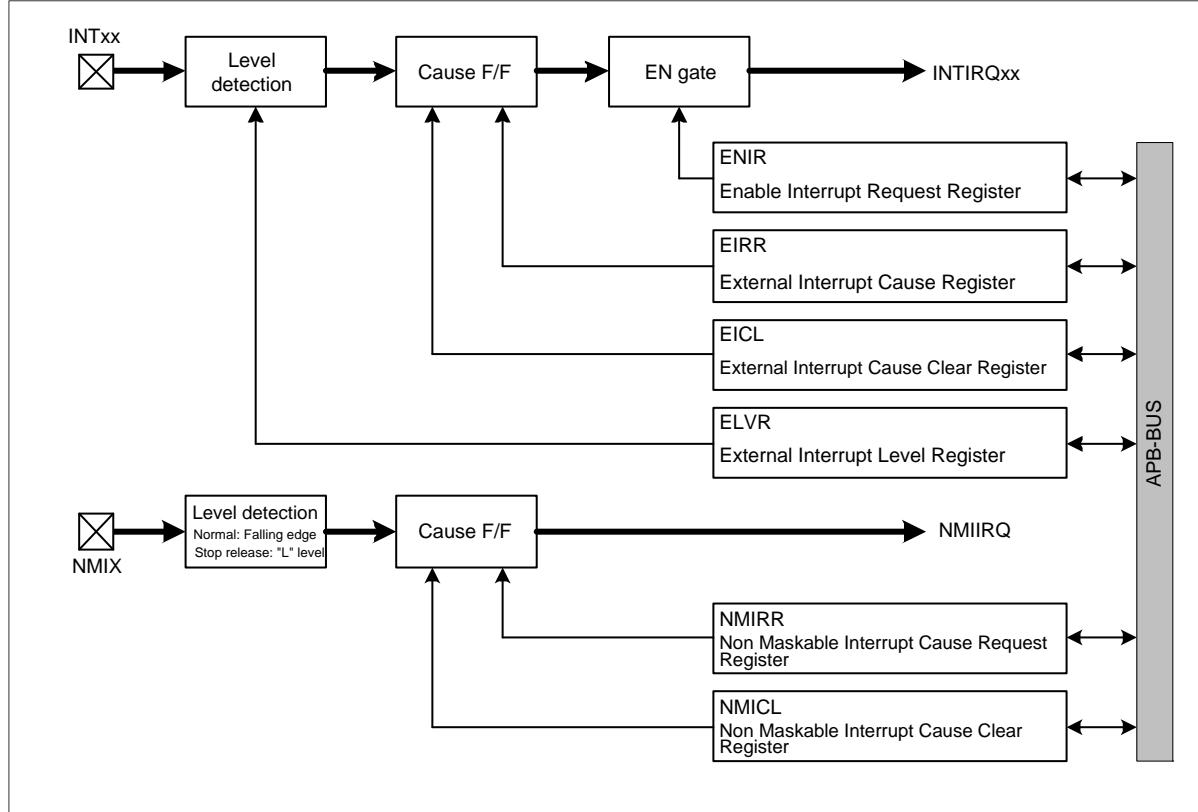
The external interrupt and NMI control sections have the following features.

- Has up to 16 external interrupt input pins and one NMI input pin mounted.
- Possible to select the "H" level, "L" level, rising edge, or falling edge to detect an external interrupt.
- Possible to use an external interrupt input or NMI input to return from standby mode.

2. Block Diagram

The following shows the block diagram of the external interrupt and NMI control sections.

Figure 2-1 Block diagram of external interrupt and NMI control sections



3. Operations and Setting Procedure Examples

This section explains operations and setting procedure examples.

- 3.1 Operations of external interrupt control section
- 3.2 Operations of NMI control section

3.1. Operations of external interrupt control section

■ Overview of operations in external interrupt control section

The external interrupt control section outputs an external interrupt request to the interrupt controller in the following procedure.

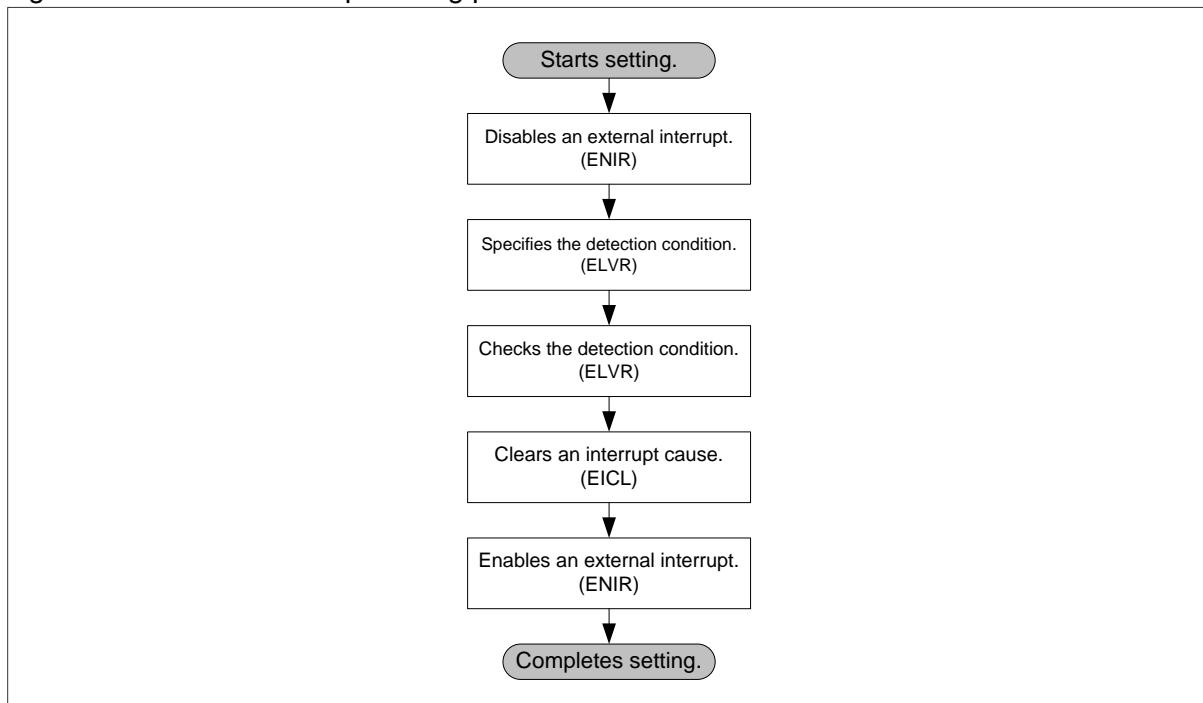
1. The signal input to pin INT_{xx} detects the edge or level specified in the External Interrupt Level Register (ELVR). The edge or level to be detected can be selected from the following four types:
"H" level, "L" level, rising edge, falling edge
2. The detected interrupt input is held in the interrupt cause F/F.
It is read with the External Interrupt Request Register (EIRR).
The held interrupt cause is cleared with the External Interrupt Clear Register (EICL).
3. If an external interrupt is enabled with the Enable Interrupt Request Register (ENIR), an external interrupt request (INTIRQ_{xx}) is output to the interrupt controller.

■ Setting procedure

Execute the following steps to configure external interrupt setting.

1. Disable an external interrupt with the Enable Interrupt Request Register (ENIR).
2. Specify the detection condition (effective edge or level) with the External Interrupt Level Register (ELVR).
3. Read the External Interrupt Level Register (ELVR).
4. Clear the external interrupt cause with the External Interrupt Clear Register (EICL).
5. Enable the external interrupt with the Enable Interrupt Request Register (ENIR).

Figure 3-1 External interrupt setting procedure



■ Canceling an external interrupt request

When the external interrupt detection condition is set to the "H" or "L" level, an interrupt cause is held in the External Interrupt Request Register (EIRR) even if an external interrupt request input (INTxx) is canceled. Therefore, an external interrupt request (INTIRQxx) remains output to the interrupt controller.

Execute the following steps to cancel an external interrupt request.

1. Read the External Interrupt Request Register (EIRR), and check the interrupt cause.
2. Write "0" to the corresponding bit in the External Interrupt Clear Register (EICL) to clear it.
3. Read the External Interrupt Request Register (EIRR), and check that the interrupt cause is cleared.

Figure 3-2 Clearing an interrupt cause

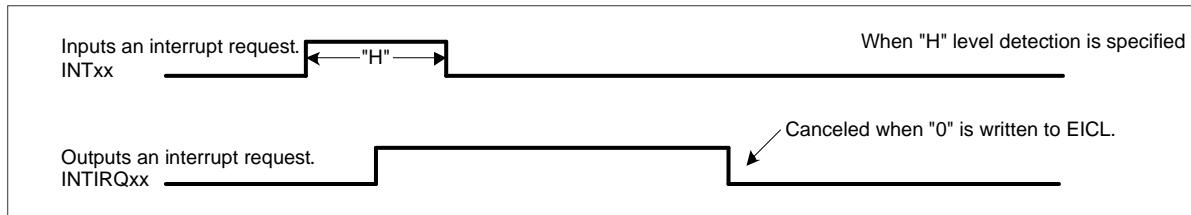
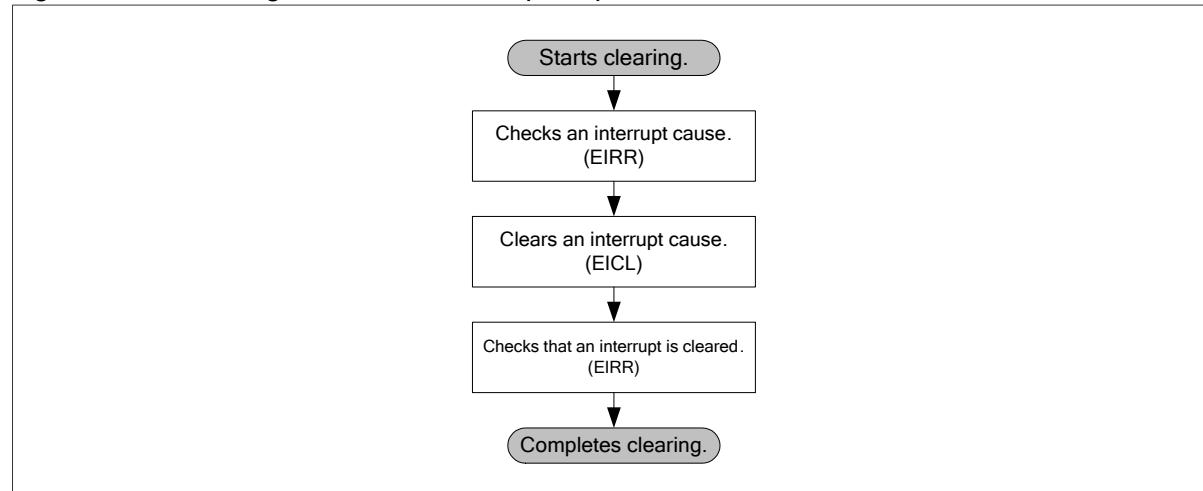


Figure 3-3 Canceling an external interrupt request



3.2. Operations of NMI control section

■ Overview of NMI control section

The NMI control section outputs an NMI interrupt request (NMIIRQ) to the CPU if the edge or level is detected from the signal input to the NMI input pin (NMIX). The following edge or level is detected.

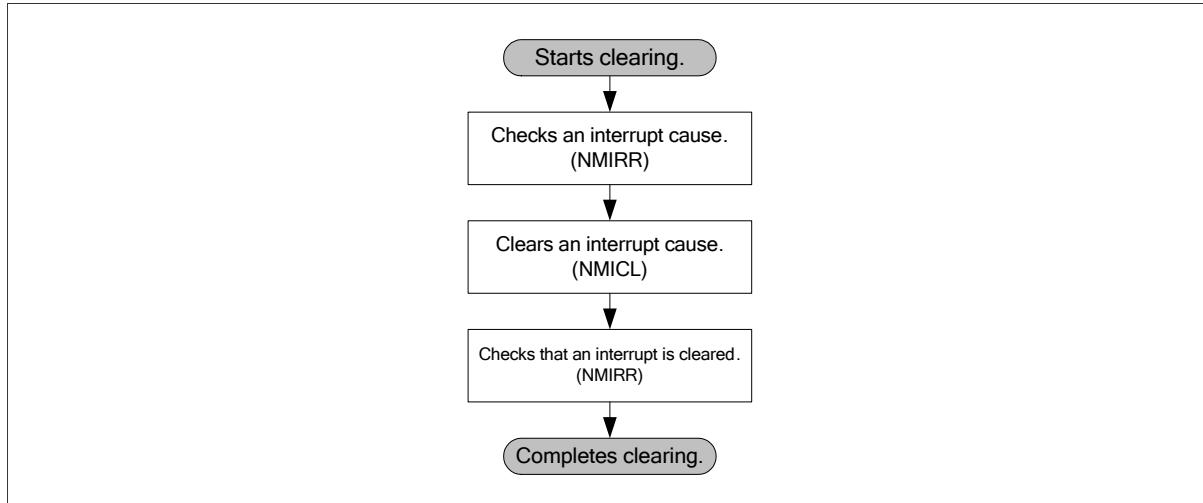
- Normal mode : Falling edge
- Stop mode : "L" level

■ Canceling an NMI request

To cancel an NMI request, clear the request register in the same way as for an external interrupt request. Execute the following steps to cancel an NMI request.

1. Read the NMI Request Register (NMIRR), and check the interrupt cause.
2. Write "0" to the corresponding bit in the NMI Clear Register (NMICL) to clear it.
3. Read the NMI Request Register (NMIRR), and check that the interrupt cause is cleared.

Figure 3-4 Canceling an NMI request



3.3. Returning from timer or stop mode

■ Overview

An external interrupt and NMI requests can be used to return from timer or stop mode.

In timer or stop mode, the signal first input to pin INTxx or NMIX is input asynchronously, and the device can return from timer or stop mode.

■ Setting before changing to stop mode

To use an external interrupt request, in the Enable Interrupt Request Register (ENIR), specify the pin used to return from stop mode and also specify the effective detection level before changing to stop mode.

- Pin used to return from stop mode. : Interrupt request output enable (ENIR = 1)
- Pin not used to return from stop mode. : Interrupt request output disable (ENIR = 0)

To use an NMI request, only the "L" level is detected, and no register setting is required.

■ Returning from stop mode

For external interrupt request, if the pre-specified effective level is detected in the pin used to return from stop mode, the device returns from stop mode.

For NMI request, if the "L" level is detected in stop mode, the device returns from stop mode.

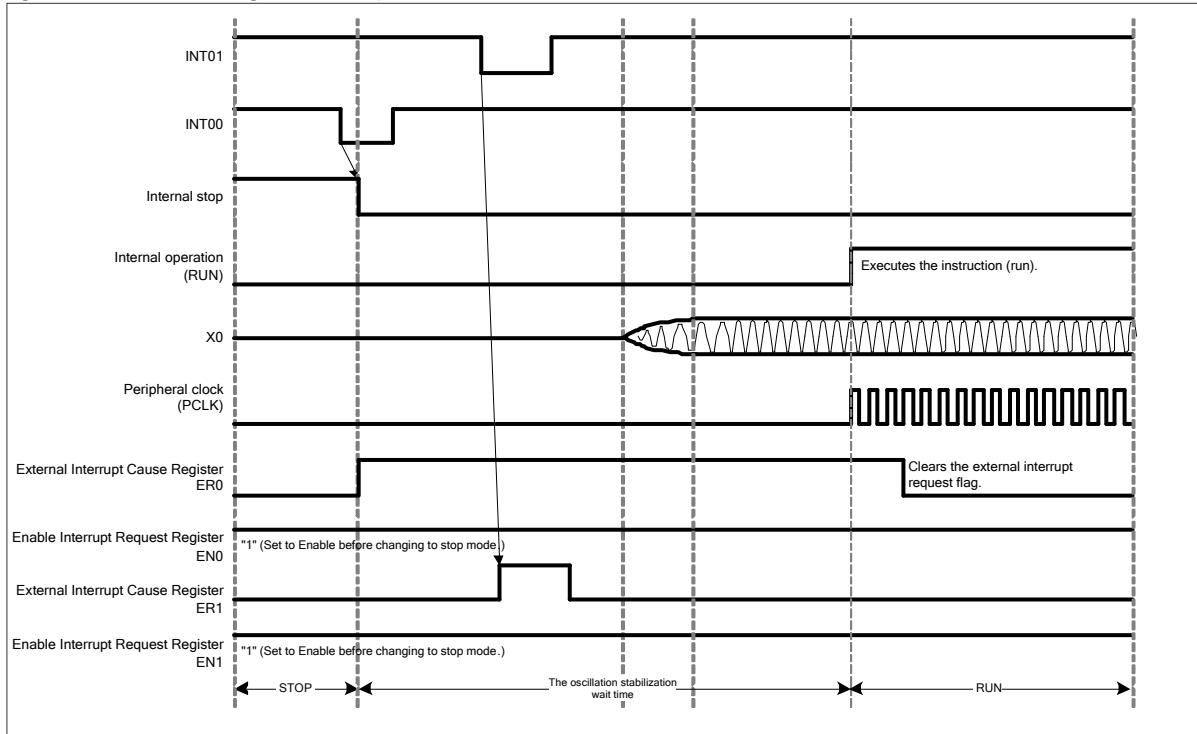
■ Notes on returning from stop mode

Any other external interrupt requests cannot be recognized until the oscillation stabilization wait time lapses after stop mode was released.

(For INT01 in Figure 3-5, any external interrupt requests cannot be recognized.)

Therefore, to input an external interrupt after stop mode was released, input an external interrupt signal after the oscillation stabilization wait time lapsed.

Figure 3-5 Returning from stop mode



4. Registers

This section provides a list of registers.

■ Register list

The following shows a list of registers in the external interrupt and NMI control sections.

Table 4-1 Registers in external interrupt and NMI control sections

Abbreviation	Register name	See
ENIR	Enable Interrupt Request Register	4.1
EIRR	External Interrupt Request Register	4.2
EICL	External Interrupt Clear Register	4.3
ELVR	External Interrupt Level Register	4.4
NMIRR	Non Maskable Interrupt Request Register	4.5
NMICL	Non Maskable Interrupt Clear Register	4.6

4.1. Enable Interrupt Request Register [ENIR]

The ENIR is used to control masking an external interrupt request output.

■ Register configuration

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Register functions

[bit15:0] EN15 to EN0: External interrupt enable bit

Bits EN15 to EN0 correspond to pins INT15 to INT00.

It is not possible to set the bit corresponding to a pin that is not defined in the model specifications.

Bit15:0	Description
0	Disables the output of an external interrupt request.
1	Enables the output of an external interrupt request.

This function enables the interrupt request output corresponding to the bit that is set to "1" in this register, and outputs a request to the interrupt controller. The pin corresponding to the bit that is set to "0" holds an interrupt cause, but outputs no request to the interrupt controller.

4.2. External Interrupt Request Register [EIRR]

The EIRR indicates that an external interrupt request is detected.

■ Register configuration

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

■ Register functions

[bit15:0] ER15 to ER0: External interrupt request detection bit

Bits ER15 to ER0 correspond to pins INT15 to INT00.

The bit corresponding to a pin that is not defined in the model specifications is indefinite.

Bit15:0	Function
0	Detects no external interrupt request.
1	Detects an external interrupt request.
During writing	No effect

<Notes>

- When level detection is set with ELVR and while valid level is input from INTxx pin, clearing applicable bit (write "0") with the External Interrupt Clear register (EICL) will reset "1" to applicable bit in the External Interrupt Request Register (EIRR).
- As the initial values of GPIO are set to general purpose ports, applicable bit in the External Interrupt Request Register (EIRR) may be set to "1". After set the GPIO to external interrupt pin, clear the External Interrupt Request Register (EIRR).

4.3. External Interrupt Clear Register [EICL]

The EICL is used to clear the held interrupt cause.

■ Register configuration

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ECL15	ECL14	ECL13	ECL12	ECL11	ECL10	ECL9	ECL8	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

■ Register functions

[bit15:0] ECL15 to ECL0: External interrupt cause clear bit

Bits ECL15 to ECL0 correspond to pins INT15 to INT00.

It is not possible to write "0" to the bit corresponding to a pin that is not defined in the model specifications.

Bit15:0	Function
When "0" is written	Clears an external interrupt cause.
When "1" is written	No effect
During reading	Always reads "1".

4.4. External Interrupt Level Register [ELVR]

The ELVR is used to select the level or edge of the signal detected as an external interrupt request.

■ Register configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8
Attribute	R/W	R/W	R/W	R/W	R/W											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
Attribute	R/W															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Register functions

- [bit31:0] LA15 to LA0 or LB15 to LB0: External interrupt request detection level selection bit
- Bits LA15 to LA0 or LB15 to LB0 correspond to pins INT15 to INT00 on a 2-bit (LA and LB) basis.
- It is not possible to set the bit corresponding to a pin that is not defined in the model specifications.
- If the edge or level selected with this bit is detected, it is recognized as an external interrupt request.

LBx	LAx	Description
0	0	Detects the "L" level.
0	1	Detects the "H" level.
1	0	Detects the rising edge.
1	1	Detects the falling edge.

4.5. Non Maskable Interrupt Request Register [NMIRR]

The NMIRR Register indicates that a non maskable interrupt (NMI) request is detected.

■ Register configuration

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field								Reserved								NR0
Attribute									-							R
Initial value									-							0

■ Register functions

[bit15:1] Reserved: Reserved bits

Bits 7 to 1 are indefinite in read mode.

They have no effect in write mode.

[bit0] NR: NMI request detection bit

The NR bit corresponds to pin NMIX.

Bit	Function
0	Detects no NMI request.
1	Detects an NMI request.
During writing	No effect

4.6. Non Maskable Interrupt Clear Register [NMICL]

The NMICL Register is used to clear the held interrupt cause.

■ Register configuration

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field								Reserved								NCL0
Attribute									-							R/W
Initial value									-							1

■ Register functions

[bit15:1] Reserved: Reserved bits

Bits 7 to 1 are indefinite in read mode.

They have no effect in write mode.

[bit0] NCL: NMI interrupt cause clear bit

The NCL bit corresponds to pin NMIX.

Bit	Function
When "0" is written	Clears an NMI interrupt cause.
When "1" is written	No effect
During reading	Always reads "1".

<Notes>

- If ELVR is rewritten to change the detection condition, an invalid interrupt cause may occur.
To avoid an invalid interrupt cause from occurring, keep the procedure shown in Figure 3-1 when changing the detection condition.
- To detect the edge or level specified in ELVR, at least 3T (T: PCLK cycle) is required as the pulse width.
If a signal that does not satisfy the pulse width is input, it is not guaranteed that correct operations will be carried out.
- When level detection is specified in ELVR, the corresponding bit in the External Interrupt Request Register (EIRR) is set to "1" again while the effective level is input from pin INTxx even if the corresponding bit is cleared (set to "0") with the External Interrupt Clear Register (EICL).
- The NMI detection level setting register is not provided. In normal mode, the falling edge is detected.
This register is used to return from stop mode when the "L" level is detected.
- NMI is targeted for non maskable interrupt, so an NMI Enable Interrupt Request Register is not provided.

CHAPTER: DMAC

This chapter describes DMAC.

1. Overview of DMAC
2. Configuration of DMAC
3. Functions and Operations of DMAC
4. DMAC Control
5. Registers of DMAC

1. Overview of DMAC

DMAC (Direct Memory Access Controller) is a function block that transfers data at high speed without CPU. Using DMAC improves the system performance.

■ Overview of DMAC

- DMAC has its own bus which is independent from the CPU bus; therefore, it allows for transfer operation even when the CPU bus is accessed.
- It consists of 8 channels enabled to execute 8 types of different DMA transfers independently from one another.
- It can set the address of the transfer destination, the address of the transfer source, the size of transfer data, the source of transfer request, and control the start of transfer operation, the forced termination of transfer and the pause of transfer for each channel.
- It can control the batch start of transfers, the forced batch termination of transfers and the batch pause of transfers for all of the channels.
- When multiple channels are operating simultaneously, it can select the priority of such channel operations from the fixed method or the rotated method.
- It supports hardware DMA transfer using an interrupt signal from Peripherals.
- It complies with the system bus (AHB), supporting 32-bit address space (4Gbyte).

■ Overview of Functions of Each Channel

- The addresses of the transfer source and transfer destination can be incremented or fixed.
- Reload function for the addresses of the transfer source and transfer destination (i.e. function to return the values to the original settings upon completion of the transfer) is available.
- The size of data to be transferred can be selected from the following three specifications:
 - Transfer data width : (Select from byte/half-word/word)
 - Setting the number of blocks : (Select from 1 to 16)
 - Setting the number of transfers: (Select from 1 to 65536)
(For information about the difference between the number of blocks and the number of transfers, see "3 Functions and Operations of DMAC".)
- Whether or not to give notification of the successful completion of transfer and unsuccessful completion of transfer can be specified.
- Transfer mode can be selected from the following five types:
 - Software-Block transfer
 - Software-Burst transfer
 - Hardware-Demand transfer
 - Hardware-Block transfer
 - Hardware-Burst transfer

■ Transfer Modes

Software transfer is a method used to start DMAC by direct instruction from CPU.

Hardware transfer is a method using an interrupt signal from a Peripheral as the DMAC transfer request signal to start DMAC directly when the Peripheral issues a transfer request.

Multifunction serial unit, USB unit and ADC unit directly instruct DMAC to start data transfer, when sending/receiving data or AD conversion data needs to be transferred. External interrupt unit and Base timer unit directly instruct DMAC to start data transfer at a transfer timing. In either of the cases, data can be transferred without CPU by making such setting beforehand.

■ Abbreviations

This chapter contains the following terms: DE, DS, DH, PR, EB, PB, ST, IS, BC, TC, MS, TW, FS, FD, RC, RS, RD, EI, CI, SS, EM. All of these terms refer to each bit of DMAC control registers (DMACR, DMACSA, DMACDA, DMACA, DMACB). See "5 Registers of DMAC".

2. Configuration of DMAC

This chapter describes the system configuration of DMAC and the I/O pins of DMAC.

2.1 DMAC and System Configuration

2.2 I/O Signals of DMAC

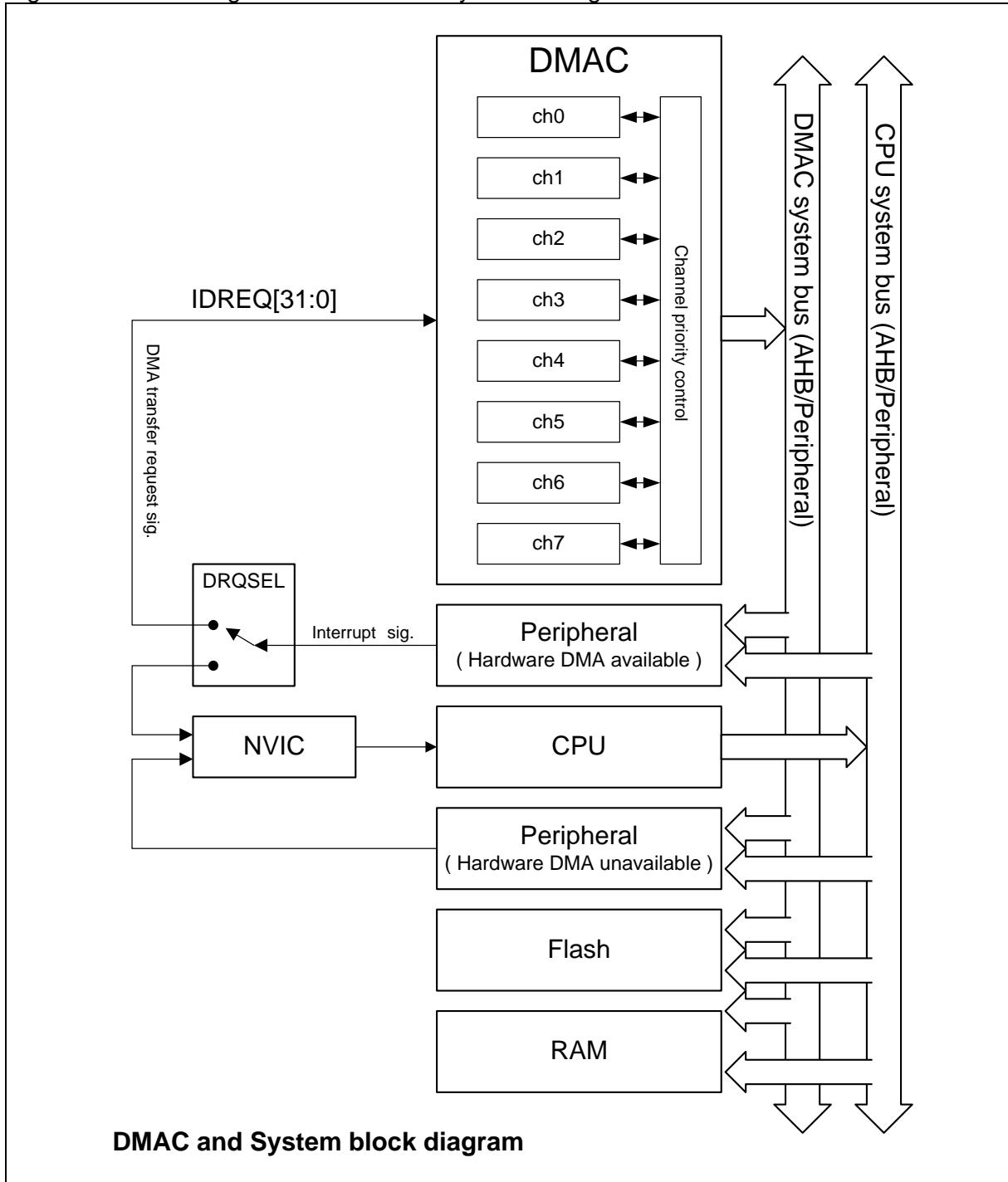
2.1. DMAC and System Configuration

This section describes DMAC and its system configuration.

■ Block Diagram

Figure 2-1 shows a diagram of DMAC and its system configuration.

Figure 2-1 Block Diagram of DMAC and System Configuration



■ Explanation of Block Diagram

● DMAC

DMAC is in 8-ch configuration. Each channel performs independent transfer. The priority controller controls the transfer operations of these channels, when there is a conflict among them.

● Connection to the system

The diagram of the system configuration in the figure has been simplified for explanation purposes. For more details, see the chapter "Overview of System". DMAC is connected to CPU, Flash, RAM and Peripherals via the system bus. It has its own bus that is independent from the CPU bus, allowing for transfer operation at CPU bus access. It accesses any address area in the system by specifying the address of transfer destination and transfer source for each channel in order to transfer data between the memory and Peripheral. Since some areas cannot be accessed from DMAC, check the memory map.

● Connection of the hardware transfer request signal

The interrupt signal from the Peripheral supporting hardware transfer is selected in the interrupt controller block (indicated as DRQSEL in the figure) either to be used as the interrupt signal to CPU or the DMA transfer request signal to DMAC.

When performing DMA transfer by hardware request, connect the interrupt signal from each Peripheral as the transfer request signal to DMAC. The interrupt signal from the Peripheral that does not support hardware transfer cannot be used as the DMA transfer request signal. When the interrupt signal is used as the transfer request to DMAC, it cannot be used as the interrupt to CPU. See the chapter "Interrupts".

There are 32 DMA transfer request signals to be input to DMAC. For the correspondence between each signal and Peripheral, see Table 2-1 in the next chapter.

(This may vary depending on the model. It should be noted that for a Peripheral with multiple channels and multiple interrupt factors, some interrupts support DMA transfer, while others don't.

In the case of hardware transfer, each channel of DMAC selects one transfer request signal out of the above 32 transfer request signals in its operation.

● Connection of the hardware transfer request clear signal

Some of the Peripherals that support hardware transfer are required to clear the transfer request signal (interrupt signal) after the completion of the transfer. Although it is not illustrated in the figure, the transfer request signal is cleared for such Peripherals via DMAC by selecting it by DRQSEL.

● Connection of the hardware transfer stop request signal

The multifunction serial unit (hereinafter abbreviated as "MFS") outputs the DMA transfer stop request signal. Although it is not illustrated in the figure, MFS's transfer stop request signal is connected to DMAC, when MFS is selected by DRQSEL. When the transfer stop request signal is asserted, DMAC stops the transfer operation. It is configured to mask the succeeding transfer request signals.

● Interrupt signal from DMAC

Although it is not illustrated in the figure, an interrupt signal used to give notification of transfer completion is connected to NVIC. Each channel has 8 interrupt outputs.

2.2. I/O Signals of DMAC

This section describes the I/O signals of DMAC.

■ Transfer Request Signals to be Input to DMAC

Table 2-1 shows a list of the transfer request signals to be input to DMAC and the interrupt signals from the corresponding Peripherals.

Table 2-1 List of Transfer Request Signals and Interrupt Signals from Corresponding Peripherals

IDREQ No.	Interrupt Signal of Corresponding Peripheral
0	Interrupt signal from EP1 DRQ of USB ch.0
1	Interrupt signal from EP2 DRQ of USB ch.0
2	Interrupt signal from EP3 DRQ of USB ch.0
3	Interrupt signal from EP4 DRQ of USB ch.0
4	Interrupt signal from EP5 DRQ of USB ch.0
5	Scan conversion interrupt signal from A/D converter unit0
6	Scan conversion interrupt signal from A/D converter unit1
7	Scan conversion interrupt signal from A/D converter unit2
8	Interrupt signal from IRQ0 of base timer ch.0
9	Interrupt signal from IRQ0 of base timer ch.2
10	Interrupt signal from IRQ0 of base timer ch.4
11	Interrupt signal from IRQ0 of base timer ch.6
12	Receiving interrupt signal from MFS ch.0
13	Sending interrupt signal from MFS ch.0
14	Receiving interrupt signal from MFS ch.1
15	Sending interrupt signal from MFS ch.1
16	Receiving interrupt signal from MFS ch.2
17	Sending interrupt signal from MFS ch2
18	Receiving interrupt signal from MFS ch.3
19	Sending interrupt signal from MFS ch.3
20	Receiving interrupt signal from MFS ch.4
21	Sending interrupt signal from MFS ch.4
22	Receiving interrupt signal from MFS ch.5
23	Sending interrupt signal from MFS ch.5
24	Receiving interrupt signal from MFS ch.6
25	Sending interrupt signal from MFS ch.6
26	Receiving interrupt signal from MFS ch.7
27	Sending interrupt signal from MFS ch.7
28	Interrupt signal from external interrupt unit ch.0
29	Interrupt signal from external interrupt unit ch.1
30	Interrupt signal from external interrupt unit ch.2
31	Interrupt signal from external interrupt unit ch.3

■ Interrupt Signals Output from DMAC

Table 2-2 shows a list of the interrupt signals output from DMAC.

Table 2-2 List of Interrupt Signals from DMAC

Name of Interrupt Signal	Interrupt Factor Register	Interrupt Enable Register	Interrupt Type
DIRQ0	DMACB0.SS[2:0]	DMACB0.CI	ch.0 successful transfer completion interrupt
		DMACB0.EI	ch.0 unsuccessful transfer completion interrupt
DIRQ1	DMACB1.SS[2:0]	DMACB1.CI	ch.1 successful transfer completion interrupt
		DMACB1.EI	ch.1 unsuccessful transfer completion interrupt
DIRQ2	DMACB2.SS[2:0]	DMACB2.CI	ch.2 successful transfer completion interrupt
		DMACB2.EI	ch.2 unsuccessful transfer completion interrupt
DIRQ3	DMACB3.SS[2:0]	DMACB3.CI	ch.3 successful transfer completion interrupt
		DMACB3.EI	ch.3 unsuccessful transfer completion interrupt
DIRQ4	DMACB4.SS[2:0]	DMACB4.CI	ch.4 successful transfer completion interrupt
		DMACB4.EI	ch.4 unsuccessful transfer completion interrupt
DIRQ5	DMACB5.SS[2:0]	DMACB5.CI	ch.5 successful transfer completion interrupt
		DMACB5.EI	ch.5 unsuccessful transfer completion interrupt
DIRQ6	DMACB6.SS[2:0]	DMACB6.CI	ch.6 successful transfer completion interrupt
		DMACB6.EI	ch.6 unsuccessful transfer completion interrupt
DIRQ7	DMACB7.SS[2:0]	DMACB7.CI	ch.7 successful transfer completion interrupt
		DMACB7.EI	ch.7 unsuccessful transfer completion interrupt

Reference: Interrupt Generation Factors and Clearing (For details, see "4 DMAC Control".)

Interrupt from each channel is generated by the following factors:

- Upon the successful completion of channel transfer, "101" is set to SS of each channel. If the above value is set to SS with CI=1 (successful transfer completion interrupt enabled), a successful transfer completion interrupt occurs.
- Upon the unsuccessful completion of channel transfer, "001", "010", "011" and "100" are set to SS of each channel. If the above value is set to SS with EI=1 (unsuccessful transfer completion interrupt enabled), an unsuccessful transfer completion interrupt occurs.
- The successful transfer completion interrupt and the unsuccessful transfer completion interrupt undergo logic OR; therefore, if either of the interrupts occurs, an interrupt occurs from the channel.

Interrupt from each channel can be cleared by writing "000" to SS.

3. Functions and Operations of DMAC

This chapter describes the operations of DMAC in each transfer mode.

- 3.1 Software-Block Transfer
- 3.2 Software-Burst Transfer
- 3.3 Hardware-Demand Transfer
- 3.4 Hardware-Block Transfer & Burst Transfer
- 3.5 Channel Priority Control

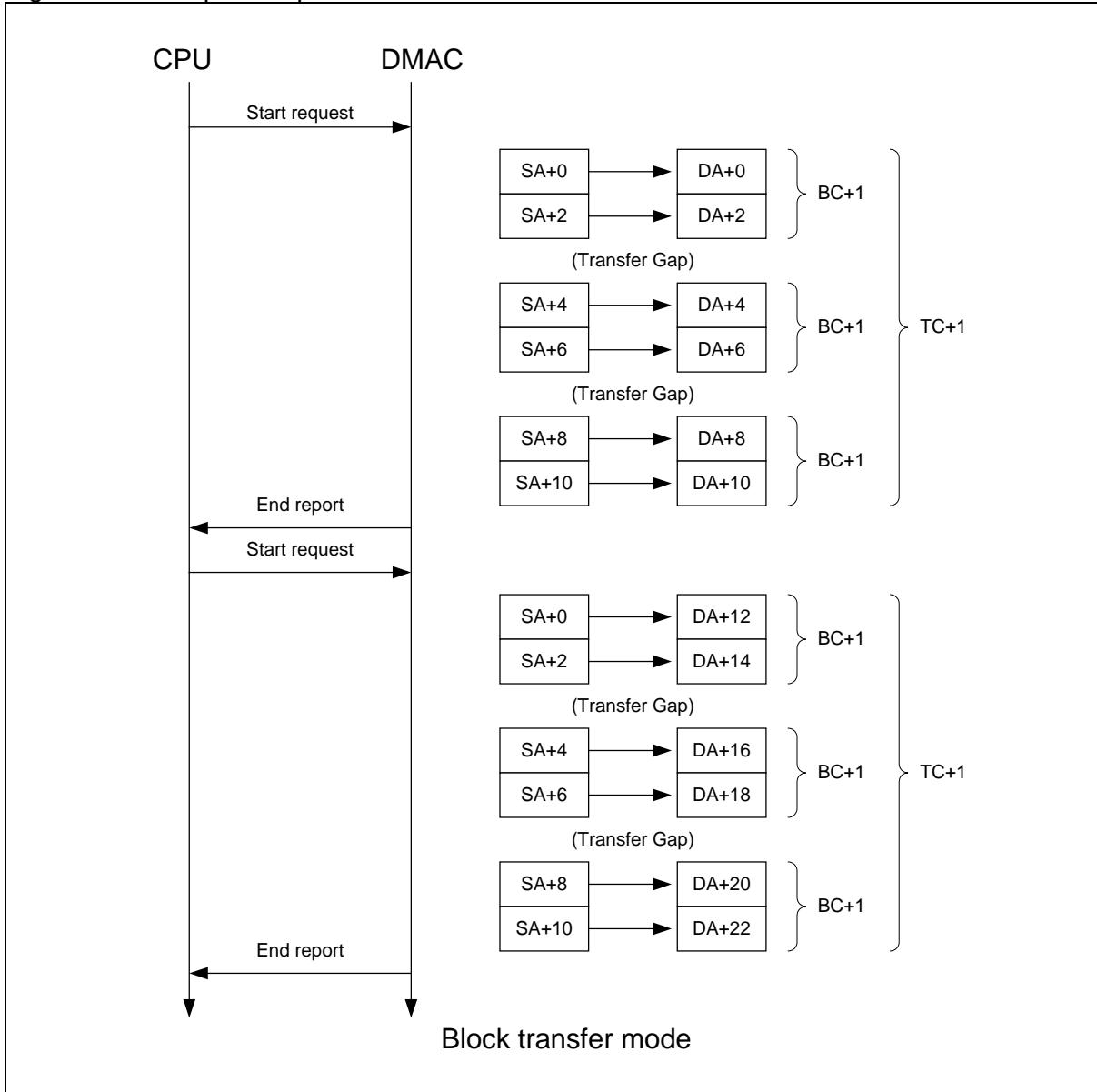
3.1. Software-Block Transfer

This section describes Software-Block transfer.

Figure 3-1 shows an example of the operation of Software-Block transfer. In this example, the following settings apply.

- Transfer mode : Software request Block transfer (ST=1, IS=000000, MS=00)
- Transfer source start address : SA(DMACSA=SA)
- Transfer source address control : Increment and reload available (FS=0, RS=1)
- Transfer destination start address : DA(DMACDA=DA)
- Transfer destination address control : Increment and reload not available (FD=0, RD=0)
- Transfer data size : Half-word (16bit), the number of blocks = 2, the number of transfers = 3 (TW=01, BC=1, TC=2)
- BC/TC reload : reload available (RC=1)

Figure 3-1 Example of Operation of Software-Block Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

- Due to the specification of the transfer data width, each transfer is performed by half-word (16bit).
- According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1).
- In the case of Block transfer, a Transfer Gap occurs every time transfer of one block is completed.
- DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) x Number of blocks (BC+1) x Number of transfers (TC+1)".
- Once the transfer is completed, DMAC notifies CPU of the completion.
- If the start of transfer is instructed again after the completion of the transfer, the transfer is restarted from the previous transfer start address (SA+0), because the transfer source address has been set to be reloaded (RS=1). As the transfer destination address has not been specified to be reloaded (RD=0), the transfer is started from the next address (DA+12) after the previous transfer end address. Also, as the reload of BC/TC has been specified, the same values as for the previous transfer are reloaded for the number of blocks and the number of transfers for the next transfer.

Transfer Gap is a time period during which no transfer is performed, and it is inserted to prevent one of the DMAC channels from taking the possession of the system access right. If multiple channels have transfer requests, DMAC switches the channels that will perform the transfer operation at the timing of the Transfer Gap. The frequency of Transfer Gap generation can be controlled by adjusting the settings of BC and TC.

Moreover, the bus access right is also passed on to CPU at the Transfer Gap timing. System buses in this model are in Multi-layered configuration with a special system bus dedicated to DMA. For this reason, if there is no conflict between CPU and the destination of access, transfer can be performed at the same time as the CPU operation. Even if there is a conflict between CPU and the destination of access, the CPU operation is little affected, as long as the DMAC transfer is in a different address area group (RAM and Peripheral, or Flash and RAM, etc). However, if the transfer is in the same address area group (RAM and RAM, etc.), the CPU operation and/or system performance may be affected, depending on the number of blocks used; therefore, attention must be paid.

("Address area group" mentioned above refers to a group of address areas that are connected on the AHB system bus with the same bus bridge.)

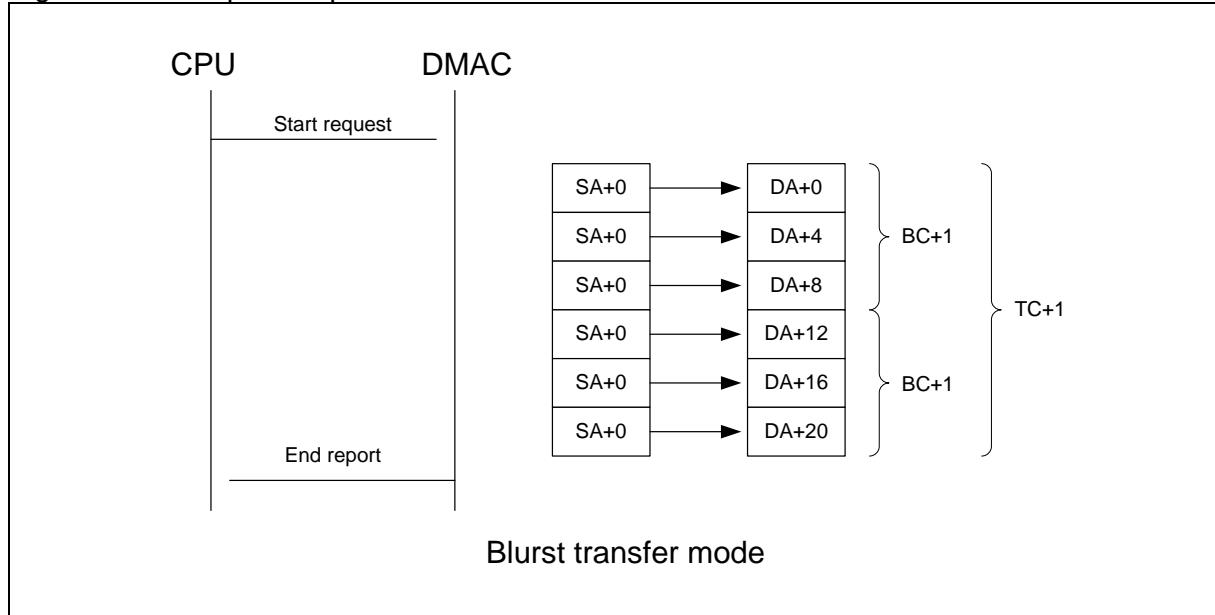
3.2. Software-Burst Transfer

This section describes Software-Burst transfer.

Figure 3-2 shows an example of the operation of Software-Burst transfer. In this example, the following settings apply.

- Transfer mode : Software request Burst transfer (ST=1, IS=000000, MS=01)
- Transfer source start address : SA(DMACSA=SA)
- Transfer source address : Fixed, reload available (FS=1, RS=1)
- Transfer destination start address : DA(DMACDA=DA)
- Transfer destination address : Increment and reload not available (FD=0, RD=0)
- Transfer data size : Word (32bit), the number of blocks = 3, the number of transfers = 2 (TW=10, BC=2, TC=1)
- Reload of the number of transfers : Number of transfers to be reloaded (RC=1)

Figure 3-2 Example of Operation of Software-Burst Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

- Due to the specification of the transfer data width, each transfer is performed by word (32bit).
- According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1). As the transfer source address is specified to be fixed, it is the same as the transfer source start address (SA+0).
- In the case of Burst transfer, the transfer is executed continuously without generating Transfer Gaps.
- DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) x Number of blocks (BC+1) x Number of transfers (TC+1)".
- When the transfer is completed, DMAC notifies CPU of the completion.

In the case of Burst transfer, no Transfer Gap is generated, unlike the Block transfer. As the channel to be controlled takes the possession of the system bus access right, it can be used to put the priority on that particular channel.

3.3. Hardware-Demand Transfer

This section describes Hardware-Demand transfer.

Hardware-Demand transfer is used when performing DMA transfer by the transfer request signal from the Peripherals of USB, MSF and ADC.

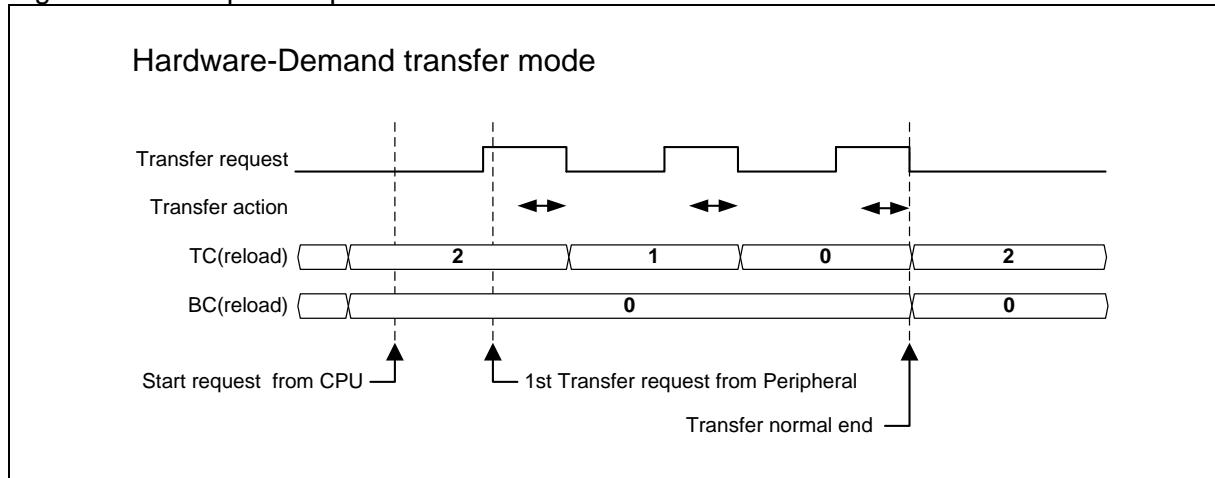
Hardware-Demand transfer is a method used to receive the transfer request signal from Peripherals on a signal level. If the transfer request signal is on High level, transfer is executed. If the transfer request signal is on Low level, no transfer is executed. Transfer is executed by setting the output of the interrupt signal from each Peripheral to High level (with interrupt request) when transfer data exists, or to Low level (without transfer request) when no transfer data exists.

In the case of Hardware-Demand transfer, always specify "1" (BC=0) as the number of blocks.

Figure 3-3 shows an example of the operation of Hardware-Demand transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode : Hardware-Demand transfer
(ST=0, IS= Peripheral at the transfer request source, MS=10)
- Transfer data size : Number of blocks = 1, Number of transfers = 3 (BC=0, TC=2)

Figure 3-3 Example of Operation of Hardware-Demand Transfer



The operation of Hardware-Demand transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs one transfer and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is (TC+1). Add the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.4. Hardware-Block Transfer & Burst Transfer

This section describes Hardware-Block transfer and Burst transfer.

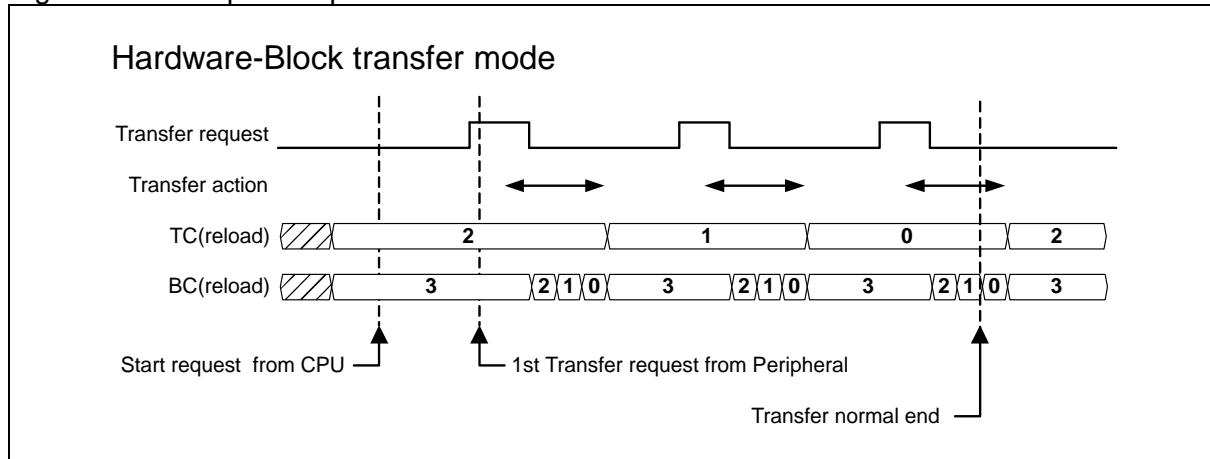
Hardware-Block transfer or Hardware-Burst transfer is used when performing DMA transfer by the transfer request signal from the Peripheral of the base timer or external interrupt.

Hardware-Block transfer and Hardware-Burst transfer are methods used to receive the transfer request signal at the rising edge of the signal. Transfer is executed, when the rising edge of the transfer request signal is detected. DMAC's transfer start timing can be specified by the output of the interrupt signal from each Peripheral.

Figure 3-4 shows an example of the operation of Hardware-Block transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode : Hardware-Block transfer
(ST=0, IS= Peripheral at the transfer request source, MS=00)
- Transfer data size : Number of blocks = 4, Number of transfers = 3 (BC=3, TC=2)

Figure 3-4 Example of Operation of Hardware-Block Transfer



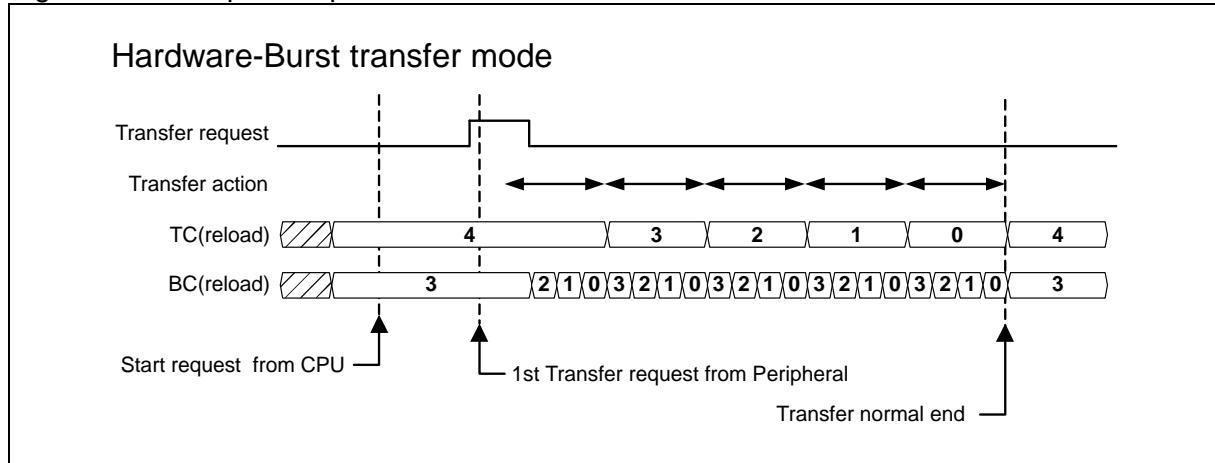
The operation of Hardware-Block transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs transfers for the number of blocks (=BC+1) and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is $(BC+1) \times (TC+1)$. Add the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

Figure 3-5 shows an example of the operation of Hardware-Burst transfer.. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode : Hardware-Burst transfer
(ST=0, IS= Peripheral at the transfer request source, MS=01)
- Transfer data size : Number of blocks =4, Number of transfers = 5 (BC=3, TC=4)

Figure 3-5 Example of Operation of Hardware-Burst Transfer



The operation of Hardware-Burst transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the first transfer request, it performs all of the transfers for the number of times calculated by $(BC+1) \times (TC+1)$. During the Hardware-Burst transfer, no Transfer Gap is generated. Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.5. Channel Priority Control

This section describes the channel priority control.

■ Channel Priority Control

If multiple channels have transfer requests, DMAC switches the channel subject to the transfer among them at the timing of the Transfer Gap of each channel. At this point, the next channel to which the transfer will be performed is determined according to the priority control. The priority control can be selected from either fixed priority or rotated priority. Figure 3-6 shows an explanatory diagram. In this figure, the right axis indicates the time axis. The arrows indicate transfer timings of each channel to perform its transfer operation when all of the channels issue transfer requests simultaneously.

■ Operation in Fixed Priority Mode (PR=0)

In fixed priority mode, the channel with the smallest channel number among all the channels with a transfer request has the priority to perform transfer operation.

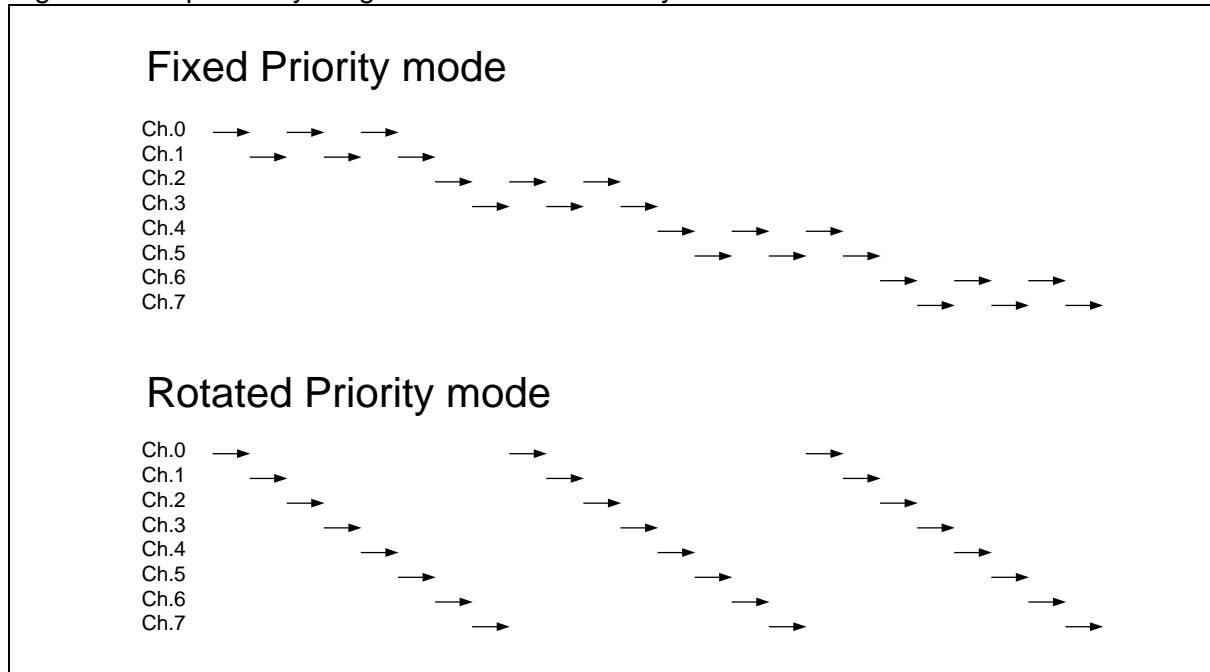
(Priority order: ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7)

First, the channel with the highest priority performs its transfer (ch.0 in the figure). As the channel with the highest priority halts the transfer operation at the timing of a Transfer Gap, then, the channel with the second highest priority performs its transfer operation (ch.1 in the figure). For this reason, the channels with the highest and the second highest priority perform the transfer operations alternately. After that, when the channel with higher priority completes its transfer, the channel with lower priority starts its transfer operation (ch.3 in the figure).

■ Operation in Rotated Priority Mode (PR=1)

In rotate priority mode, all channels perform their transfer operations equally.

Figure 3-6 Explanatory Diagram of Channel Priority Control



4. DMAC Control

This chapter describes DMAC control methods in details.

- 4.1 Overview of DMAC Control
- 4.2 DMAC Operation and Control Procedure for Software Transfer
- 4.3 DMAC Operation and Control Procedure for Hardware (EM=0) Transfer
- 4.4 DMAC Operation and Control Procedure for Hardware (EM=1) Transfer

4.1. Overview of DMAC Control

This section provides an overview of DMAC control.

The control register of each channel of DMAC has EB (individual-channel operation enable bit) and PB (individual-channel pause bit). By manipulating these bits, the start of DMA transfer operation (operation enabled), the forced termination of transfer operation (operation disabled) and the pause of transfer operation can be controlled by channel. The control register also has DE (all-channel operation enable bit) and DH (all-channel pause bit), which allow the transfer operations of all channels to be controlled at once.

Each channel is originally in the operation-prohibited state (Disable state) in which the transfer content (the address of the transfer source, the address of the transfer destination, the transfer data width, the number of transfers, the transfer mode, etc.) are specified for each channel to its configuration register. Then, the transfer operations are controlled by writing to EB, PB, DE and DH to instruct the transfer operations to be started or paused.

Once each channel completes its transfer, it sets the end code to SS (Stop Status) to give the notification of its stop state. An interrupt can be generated upon the completion of transfer. After the transfer ends, each channel clears EB and PB and returns to the operation-prohibited state (Disable state).

The following sections describe the operations of and control procedures for DMA transfer by software request and hardware DMA transfer by transfer request from Peripherals.

The following terms are used in the explanations as instructions from CPU, which refer to writing the following values to the EB, PB, DE and DH bits.

- Instruction to enable individual-channel operation (write EB=1, PB=0)
- Instruction to disable individual-channel operation (write EB=0)
- Instruction to pause individual-channel operation (write EB=1, PB=1)
- Instruction to enable all-channel operation (write DE=1, DH=0000)
- Instruction to disable all-channel operation (write DE=0)
- Instruction to pause all-channel operation (write DE=1, DH!=0000)

4.2. DMAC Operation and Control Procedure for Software Transfer

This section describes DMAC operation and control procedure for software transfer.

Figure 4-1 Transitional Diagram of Software Transfer State

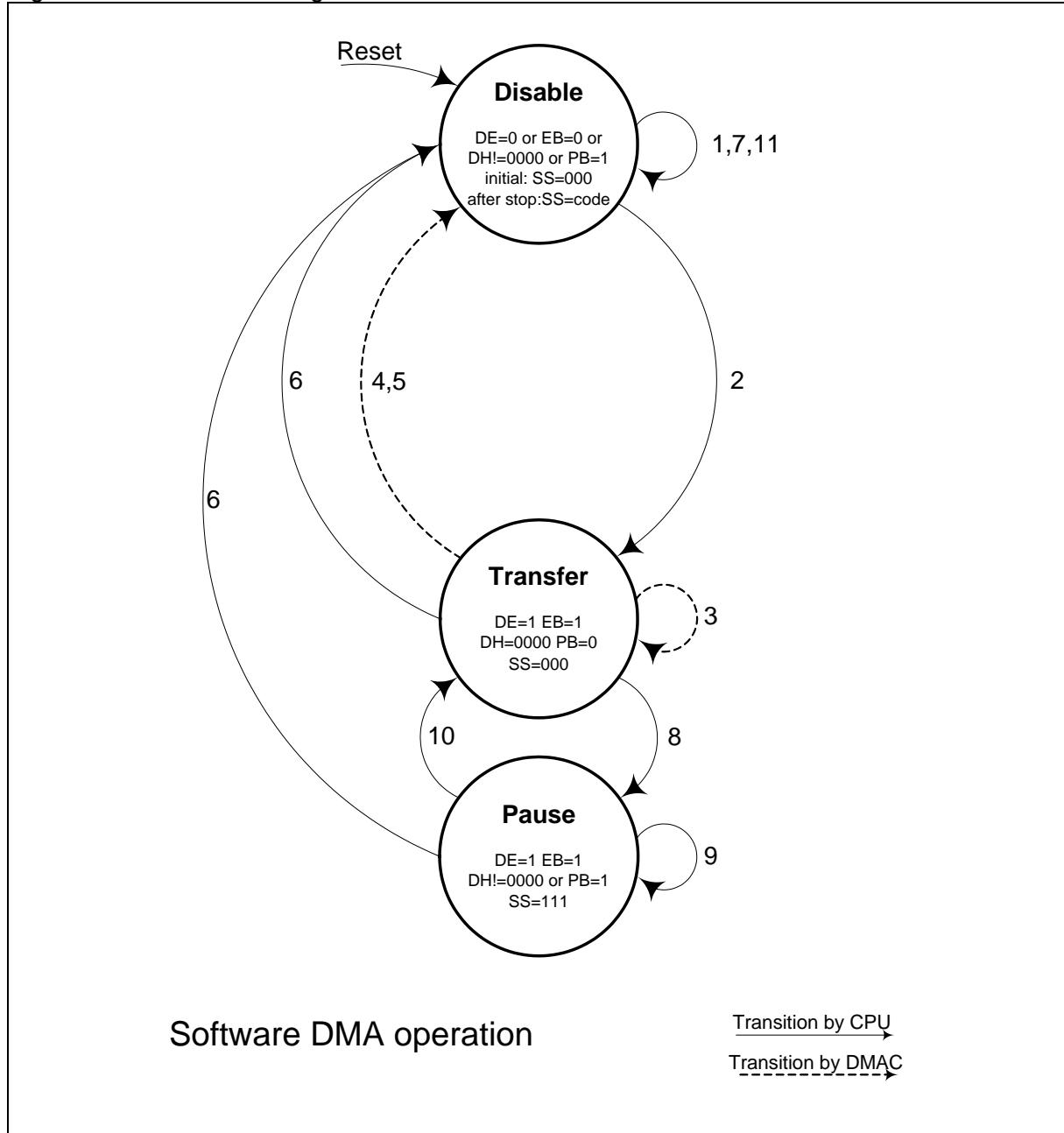


Figure 4-1 shows a transitional diagram of the states of the channel to be controlled for software transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC operation.

■ Description of Each State

Disable state

In this state, the transfer of the channel to be controlled is prohibited. Channels in this state do nothing and wait for instruction from CPU. At the system reset, DE=0, EB=0, DH=0000 and PB=0 apply to this Disable state.

Transfer state

In this state, the transfer of the channel to be controlled is enabled. Channels in this state perform transfer operation as specified. Once all of the transfer operations are completed, they return to the Disable state. The state is also changed as instructed by CPU.

Pause state

In this state, the channel to be controlled has its transfer operation on pause due to an instruction to pause, issued by CPU, and is waiting for another instruction from CPU.

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

Specify via CPU the transfer content for the channel to be controlled (writing to DMACSA, DMACDA, DMACA and DMACB). For details of transfer content to be specified, see the section describing register functions. When generating an interrupt from DMAC upon the completion of transfer, set EI and CI.

The following restrictions apply to software transfer. Specify ST=1 and IS=000000. Demand transfer mode cannot be specified to MS. Always set "0" to EM.

Give an instruction to enable all-channel operation and set PR. Data can also be written to DMACA at the same time in Step 2.

2. Disable state => Transfer state / Start of transfer

Give an instruction to enable individual-channel operation from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Transfer state.

3. Transfer state

When the channel in Transfer state becomes enabled to access the system bus, it performs a transfer according to the transfer content (it may take time to start the transfer, depending on the status of other channels). In the case of Block transfer, a Transfer Gap is generated every time TC is updated. In the case of Burst transfer, no Transfer Gap is generated. During the transfer operation, BC, TC, DMACSA and DMACDA indicate the remaining number of transfers and the transfer address at that time point. The transfer status can be checked by reading from CPU.

The specified transfer content cannot be changed via CPU to the channel in Transfer state (rewriting to DMACSA, DMACDA, DMACA[29:0], DMACB[31:1]). (However, EB, PB and EM can be rewritten.)

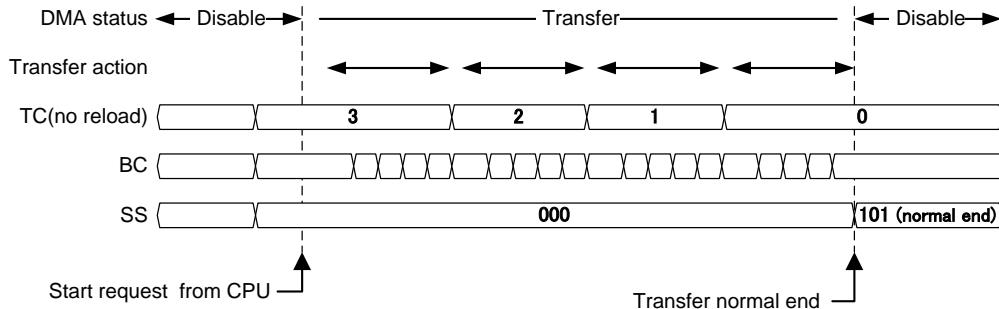
4. Transfer state => Disable state / Successful completion of transfer

When transfers are successfully completed for the number of times calculated by $(BC+1) \times (TC+1)$, the channel in Transfer state clears EB, PB and ST and moves to Disable state. It sets SS=101 to provide the notification of the successful completion. See Example 1 in Figure 4-2. If successful transfer completion interrupt has been enabled, an interrupt occurs. If reload has been specified to BC, TC, DMACSA and DMACDA, such reload is executed according to the specified transfer content.

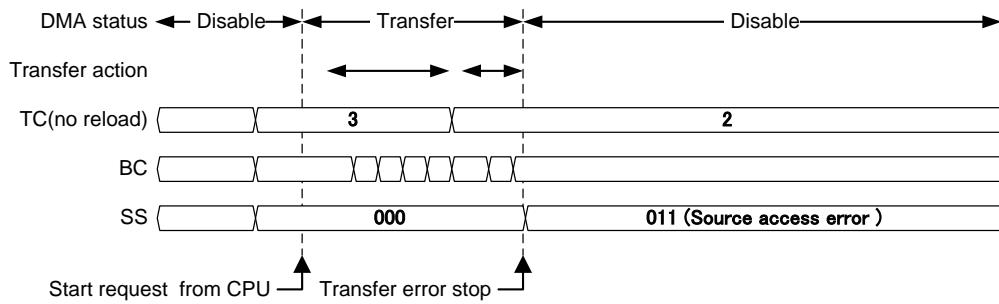
Figure 4-2 Example of Operation of Software-Block Transfer

Example of Block transfer mode (software DMA operation)
start / normal end / error stop / force stop

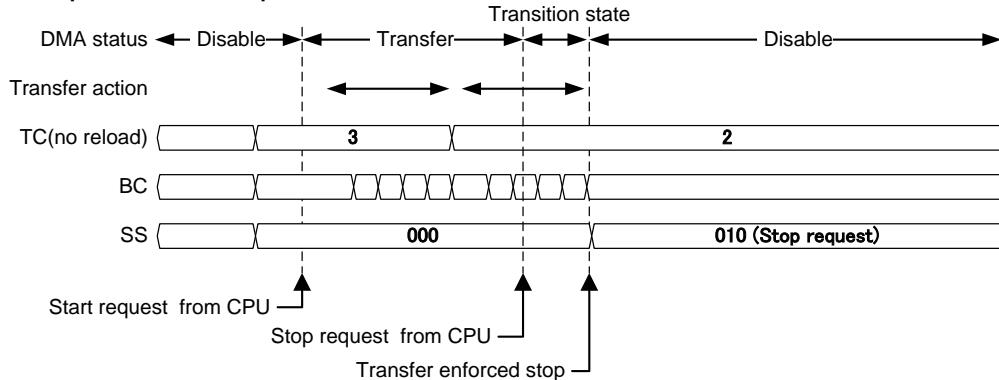
Example 1 :normal end



Example 2 : error stop



Example 3 : enforced stop from CPU



5. Transfer state => Disable state / Transfer error stop

The channel in Transfer state suspends the transfer process, if an address overflow, transfer source access error or transfer destination access error occurs. It clears EB, PB and ST and moves to Disable state. It sets the value that indicates the error content to SS to give the notification of the error stop. See Example 2 in Figure 4-2. If unsuccessful transfer completion interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.

Normally, a transfer error occurs, when an attempt is made to access an address area that does not exist in the system bus or an address area that prohibits access from DMAC. No such error occurs in general applications.

6. Transfer state, Pause state => Disable state / Forced transfer stop

If an instruction to disable individual-channel operation or an instruction to disable all-channel operation is issued from CPU to a channel in Transfer state or Pause state, the transfer operation of that channel can be forced to stop (for the operation when an instruction to disable operation is issued to a channel in Disable state, see Step 11 in the software procedure).

If an instruction is given from CPU, the relevant channel suspends its transfer process. It clears EB, PB and ST and moves to Disable state. It sets SS=010 and gives the notification that the transfer of that channel has been forced to stop. If unsuccessful transfer completion interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.

After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the transfer starts), as shown in the Example 3 in Figure 4-2. In the case of a channel in Pause state, the transfer stops immediately. There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. As a new transfer cannot be set or started during this period, always make sure that the operation has stopped before setting the next transfer.

In the case of an instruction to disable all-channel operation, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped.

Even if instructed from CPU, the transfer may not be forced to stop, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to disable the operation). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

7. Disable state / Post-transfer process

SS is read from CPU to check the state of completion of the transfer. CPU clears SS to prepare for the next transfer. If interrupts have been enabled, the interrupt signal from DMAC is deasserted by clearing SS.

In the case of successful completion, CPU resets the transfer content, as required. If each reload has been specified, the values set before the start of the transfer are reloaded to BC, TC, DMACSA and DMACDA. If each reload has not been specified, BC and TC are initialized to "0". DMACSA and DMACDA show the address for the next transfer.

In the cases of error stop and forced stop, BC, TC, DMACSA and DMACDA must always be reset, because they may have the values set at the time of the suspension.

If the transfer is stopped due to an instruction to disable all-channel operation, DE is set to "0"; therefore, the next transfer will require an instruction to enable all-channel operation and an instruction to enable individual-channel operation.

8. Transfer state / Transfer pause

If an instruction to put individual-channel operation on pause or an instruction to put all-channel operation on pause is issued from CPU to a channel in Transfer state, the transfer operation of the relevant channel(s) can be put on pause (for the operation when an instruction to put the operation on pause is issued to a channel in Disable state, see Step 11 in the software procedure).

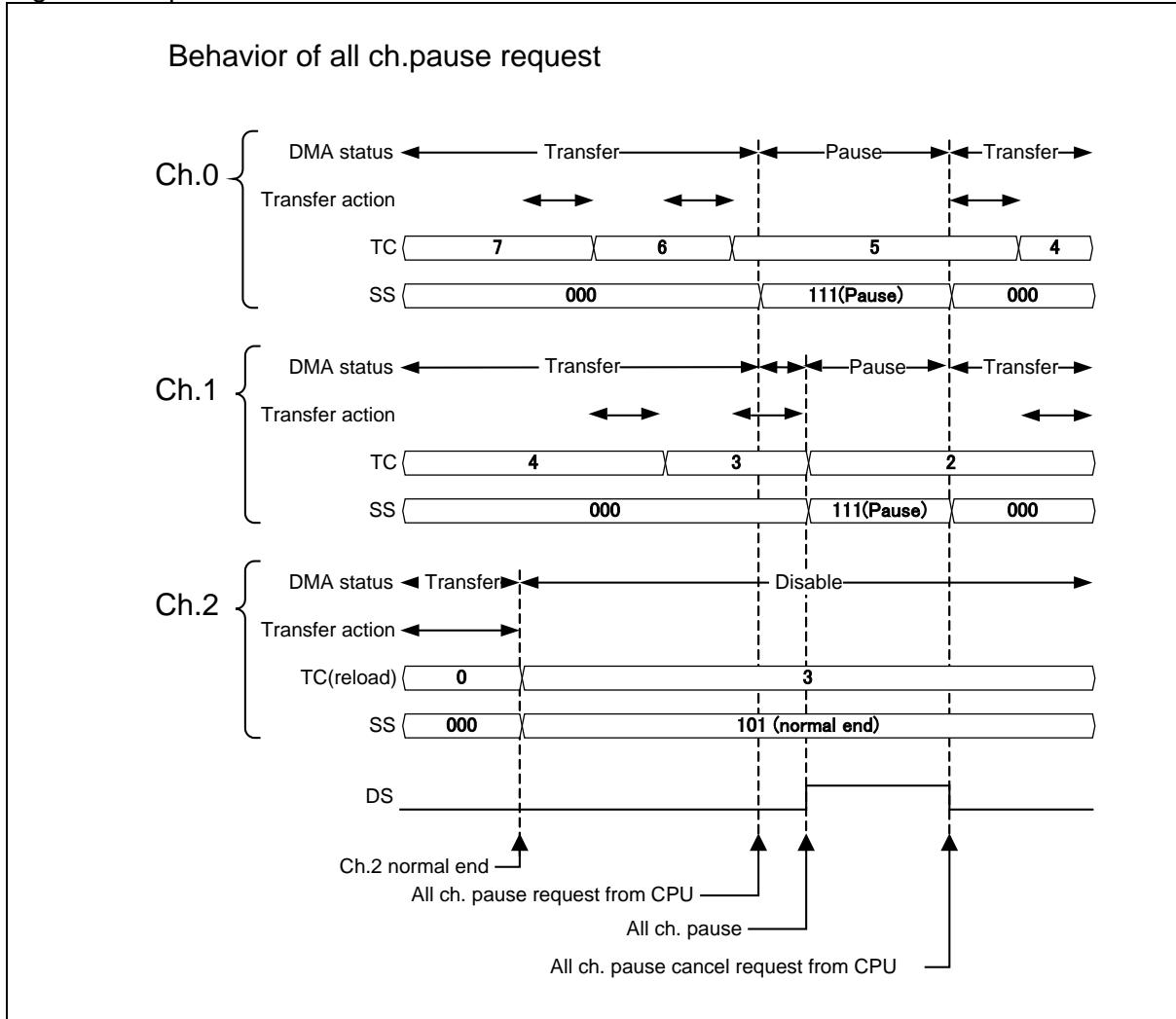
If an instruction is given from CPU, the relevant channel(s) temporarily suspends the transfer process. It sets SS=111 and gives the notification that it is in Pause state. In this case, no interrupt can be generated.

After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the start of the transfer). There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. See Figure 4-3.

In the case of an instruction to put all-channel operation on pause, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped. See Figure 4-3.

Even if instructed from CPU, the transfer may not be put on pause, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to put the operation on pause). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

Figure 4-3 Operation when All-channel Pause is Instructed



9. Pause state

SS is read from CPU to confirm the pause of the transfer. The SS of a channel in Pause state is "111". While in this state, it cannot be cleared from CPU.

Even during the pause, the transfer content cannot be specified or changed (writing DMACSA, DMACDA, DMACA[29:0] or DMACB[31:1]). Also, when a channel in Pause state is instructed to pause, it continues to remain in the Pause state.

10. Pause state / Cancellation of transfer pause

If an instruction to enable individual-channel operation is issued to a channel that has been in Pause state due to an instruction to put individual-channel operation on pause, that channel returns to Transfer state. If an instruction to enable all-channel operation is issued to channels that have been in Pause state due to an instruction to put all-channel operation on pause, those channels return to Transfer state. If both of the pause instructions have been given, issue an instruction to cancel both of them.

After the instruction, SS is cleared to "000" via DMAC.

If an instruction to enable individual-channel operation and an instruction to enable all-channel operation are issued in Pause state, they instruct the pause to be cancelled. If they are issued in Disable state, attention must be paid, as they may instruct a new transfer to be started. See Step 11 in the software procedure.

Figure 4-3 shows an example of the case where an instruction to put all-channel operation on pause. The explanation of the figure is as follows.

At the beginning, three channels, namely ch.0, ch.1 and ch.2, perform their transfer operations in Block transfer mode. ch.2 successfully completes its transfer, moves to Disable state and sets SS=101. Then, ch.0 and ch.1 perform transfers alternately.

If an instruction to put all-channel operation on pause is issued from CPU at this point, the following operation applies. As ch.0 is subject to the Transfer Gap timing, it immediately moves to Pause state and sets SS=111. As ch.1 is in the middle of transfer operation, it performs the transfer until the timing of the next Transfer Gap, and then moves to Pause state and sets SS=111. As ch.2 is in Disable state, it remains in the Disable state without changing SS. DS is set, when all of the channels stop their operations.

Next, if an instruction to enable all-channel operation (instruction to cancel the pause) is issued from CPU, the following operation applies. ch.0 and ch.1 return to Transfer state and clear SS to "000". As ch.2 is in Disable state (DE=1, EB=0), it remains in that state without starting the operation. Because the pause of all of the channels has been cancelled now, DS is reset.

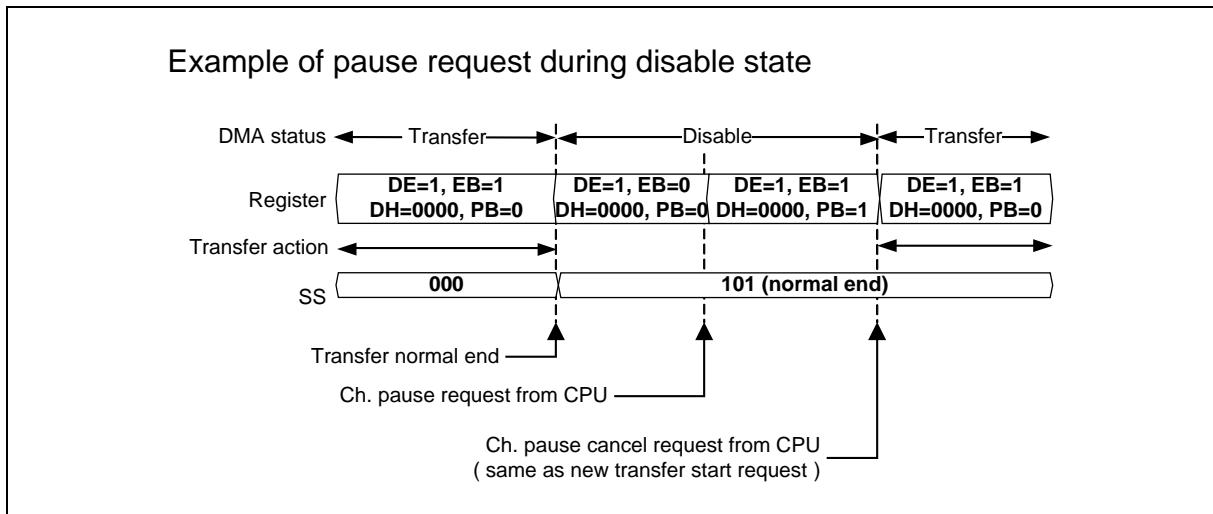
11. Operation in Disable state

A channel in Disable state remains in the Disable state, unless the conditions such as DE=1, DH=0000, EB=1, and PB=0 are established. Although in 1-2 of the software procedure, DE is set from the conditions of DE=0 and EB=0, and then, EB is set, there is no problem to set EB before DE. DE can be set last after all of the transfer settings of multiple channels subject to transfer are completed. In this case, an instruction can be issued to allow the multiple channels subject to transfer to start their transfer operations simultaneously. If such instruction for simultaneous start of transfers is issued, DMAC selects the channels to which transfers are to be started, according to the PR setting (PR can be set or changed, only when all-channel operation is disabled).

If an instruction to disable individual-channel operation, an instruction to put individual-channel operation on pause, an instruction to disable all-channel operation or an instruction to put all-channel operation on pause is issued, only the settings of DE, DH, EB and PB are changed, but the conditions of DE=1, DH=0000, EB=1 and PB=0 are not established. Therefore, the relevant channels do nothing and do not change SS. If an instruction to put all-channel operation on pause is issued from CPU to a channel in Disable state, as shown in the example of ch.2 operation in Figure 4-3, that channel does not change its state with SS indicating the completion of the previous transfer.

If an instruction to put individual- or all-channel operation on pause is issued to a channel in Disable state, it may be put in Disable state with DE=1, EB=1, (DH!=0000 or PB=1). Although the bit values in this state are the same as DE, EB, DH and PB, they can be distinguished because SS has a different value. Figure 4-4 shows such an example.

Figure 4-4 Example of Operation when Instruction to Put Individual-channel Operation on Pause is Issued in Disabled State



A certain channel is performing transfer operation. CPU issues an instruction to put individual-channel operation on pause to that channel. The instruction is issued after the transfer is completed and it moves to Disable state (DE=1, DH=0000, EB=0, PB=0). This phenomenon can occur, because the channel currently performing transfer operation changes its state outside CPU's intention. In this case, the bit values of the relevant channel change to (DE=1, DH=0000, EB=1, PB=1) due to instruction from CPU, but SS remains "101", the value set upon the completion. If the operation is stopped by a pause instruction, SS will be "111"; therefore, it will be possible to distinguish between the pause state and the state in which the transfer has been completed. It should be noted that if an instruction to cancel the pause is issued without checking the state of the channel by SS, a new transfer will accidentally start, as shown in Figure 4-4.

Additional Matter 1

As ST is cleared upon the completion of a transfer, the read value of ST is "0" after the completion of the transfer. In the case of software transfer, it should be noted that "1" must always be written to ST, regardless of its read value.

Additional Matter 2

An instruction to enable individual-channel operation cannot be issued during the period after the previous instruction to enable individual-channel operation instructs the start of transfer and before the completion of the transfer is confirmed. This is because the channel to be controlled may change its state outside CPU's intention and an instruction to start a new transfer may be issued when DMAC has moved to Disable state (EB=0). Even if the SS value confirms that the channel to be controlled is in Transfer state, the channel to be controlled may move to Disable state during the period between that point and the write operation.

Additional Matter 3

The DE and DH values can only be rewritten from CPU and these registers are never cleared from DMAC. Therefore, there is no problem to write DE=1 and DH=00000 during the transfer operation.

DH is not cleared, if an instruction to disable individual-channel operation is issued to a channel in all-channel Pause state (DE=1, DH!=0000, EB=1, PB=0). After the instruction, the relevant channel moves to Disable state (DE=1, DH!=0000, EB=0, PB=0). To start a new transfer of the relevant channel, write DE=1 and DH=0000. This indicates that the cancellation of the pause of all-channel operation is required in order to start a new transfer of the individual channel.

Additional Matter 4

The SS value is set from DMAC upon the completion of a transfer and it is never rewritten from DMAC as long as it is in Disable state. Even if the SS value is not cleared, the next transfer can be started. However, if it moves to Transfer state, the SS value may be cleared from DMAC (or may not be cleared). When an interrupt from DMAC is used, it should be noted that the interrupt signal is deasserted at a timing which is not intended by CPU, if it moves to Transfer state without clearing SS.

4.3. DMAC Operation and Control Procedure for Hardware (EM=0) Transfer

This section describes DMAC operation and control procedure for hardware (EM=0) transfer.

Figure 4-5 Transitional Diagram of Hardware (EM=0) Transfer State

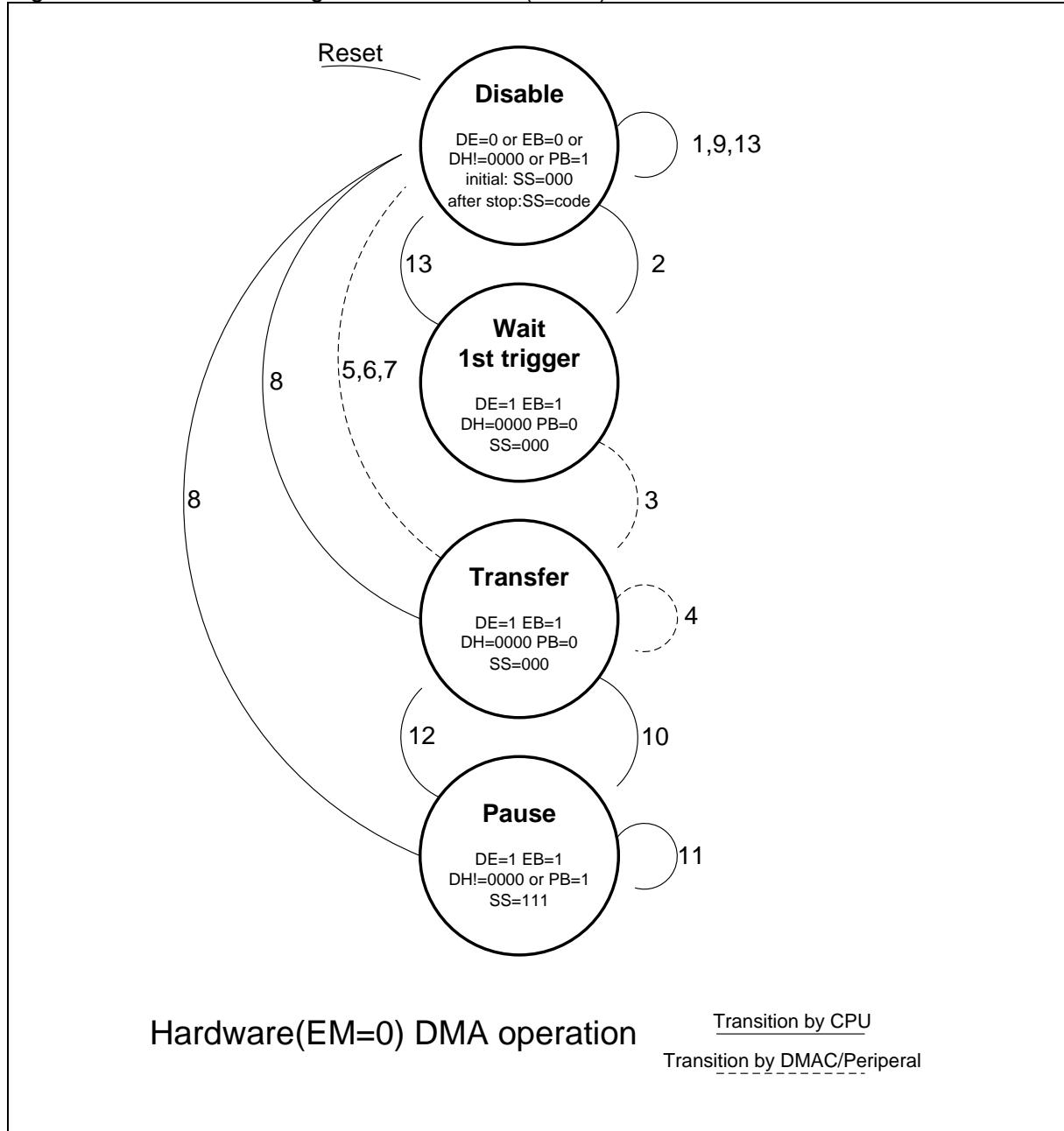


Figure 4-5 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=0) transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

Some parts of the explanation below state "See the software transfer procedure". This means that where the same control as in the software transfer procedure applies, no special mentioning is required; therefore, such redundant explanation has been omitted. In this example, the explanation assumes that EM=0 is set.

■ Description of Each State

Disable state

See the software transfer procedure.

Wait-1st-trigger state

In this state, the channel to be controlled is enabled to perform transfer. A channel in this state waits for the first transfer request from a Peripheral to be asserted. It also changes its state upon instruction from CPU.

Transfer state

In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. Once all the transfer operation is completed, it returns to Disable state. It also changes its state upon instruction from CPU.

Pause state

See the software transfer procedure.

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

See Step 1 in the software transfer procedure.

The following restrictions apply to hardware transfer. Decide in advance on which Peripheral's interrupt signal to be used as the transfer request signal to DMAC using the interrupt controller block (see the section on the functional explanation). Set ST=0 and specify which Peripheral's transfer request to be processed at the channel that will perform the transfer, by IS at the same time. Multiple channels cannot process transfer request of the same Peripheral. In the case of Demand transfer mode, set BC=0. This section explains the operation when EM=0 is set.

2. Disable state => Wait-1st-trigger state / Transfer enabled

An instruction to enable individual-channel operation is issued from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Wait-1st-trigger state.

3. Wait-1st-trigger state / Start of transfer

The channel in Wait-1st-trigger state is waiting for the transfer request signal to be asserted from the Peripheral or for an instruction from CPU. When the first transfer request signal is asserted, it moves to Transfer state.

4. Transfer state

See Step 3 in the software transfer procedure.

In the case of hardware transfer, a channel in Transfer state performs transfer operation by the transfer request signal from a Peripheral, as described in Sections 3.3 and 3.4. In each mode, match the number of transfer requests from the Peripheral with the number of transfer requests required by DMAC. Below is the explanation for the operation when the number of transfer requests goes over or below the requirement in each operation mode.

Figure 4-6 shows a case of Demand transfer. In the case of Demand transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-6).

If the number of transfer requests generated from the Peripheral exceeds the DMAC's setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues (Example 2 in Figure 4-6).

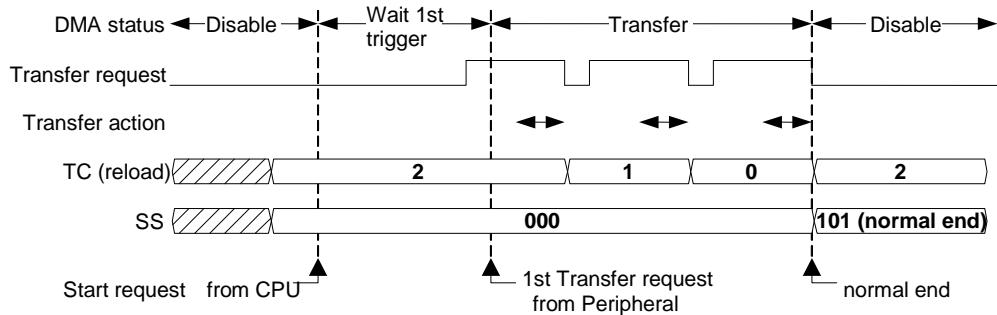
If the number of transfer requests generated from the Peripheral is smaller than DMAC's setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-6).

It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Demand transfer, the transfer request signal remains asserted; therefore, as many as TC+1 of transfers can be performed (Example 4 in Figure 4-6).

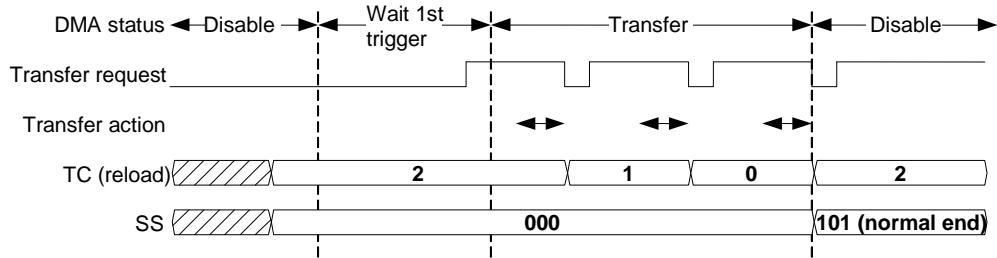
Figure 4-6 Operation of Hardware-Demand Transfer

Demand transfer mode (hardware DMA operation)

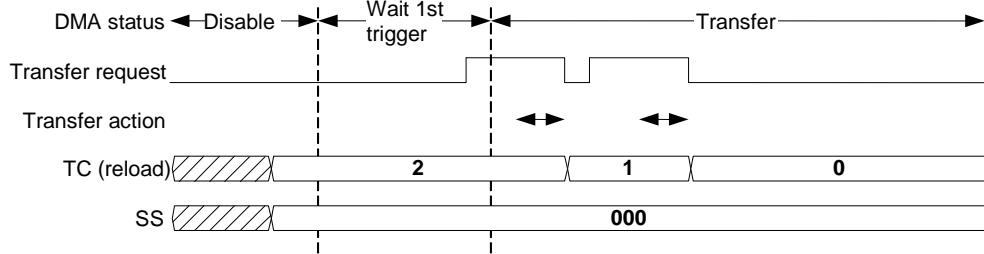
Example 1: (TC+1)== Transfer request from Periperal



Example 2: (TC+1) < Transfer request from Periperal



Example 3: (TC+1) > Transfer request from Periperal



Example 4: DMA transfer be delayed

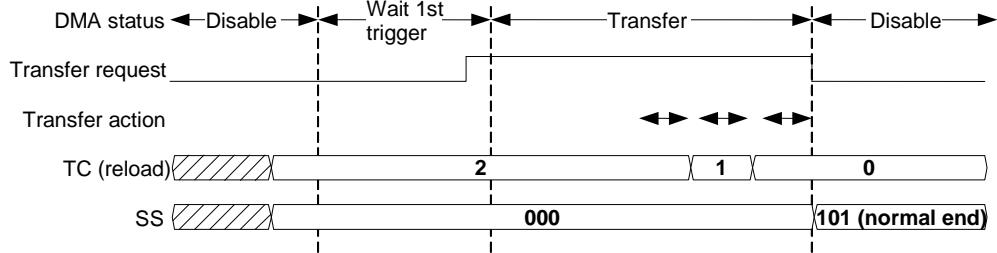
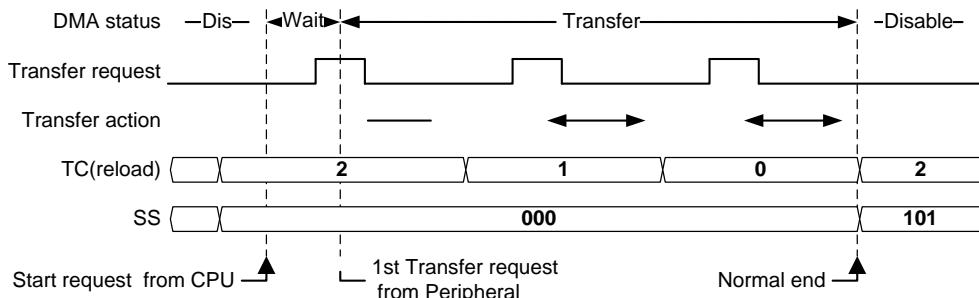


Figure 4-7 shows a case of Block transfer. In the case of Block transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-7).

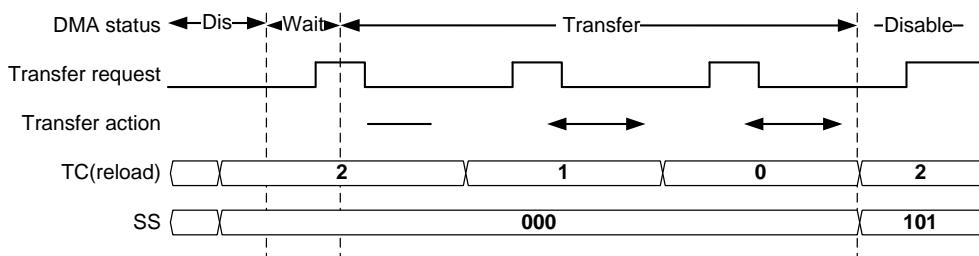
Figure 4-7 Operation of Hardware-Block Transfer

Block transfer mode (hardware DMA operation)

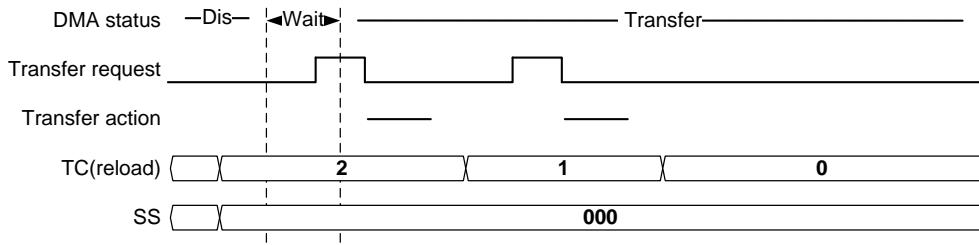
Example1: (TC+1)== Transfer request from Periperal



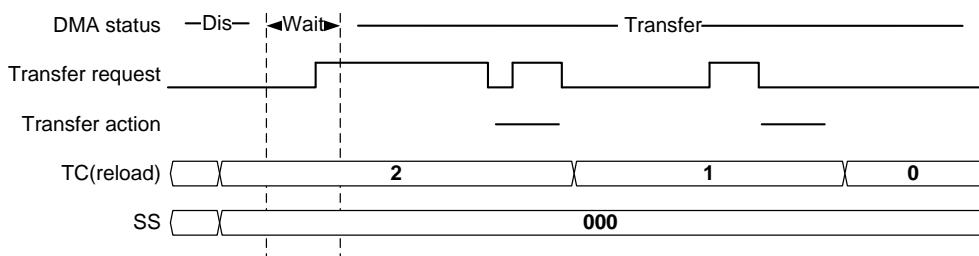
Example2: (TC+1) < Transfer request from Periperal



Example3: (TC+1) > Transfer request from Periperal



Example4: DMA transfer be delayed



If the number of transfer requests generated from the Peripheral exceeds the DMAC's setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues. In this case, deassert the transfer request signal from CPU (Example 2 in Figure 4-7).

If the number of transfer requests generated from the Peripheral is smaller than DMAC's setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-7).

It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Block transfer, if DMAC's transfer processing is delayed from the transfer request from the Peripheral, the rising edge of the next transfer request signal during the transfer operation is ignored. Also, the transfer request signal asserted during the transfer operation is cleared from DMAC. Then, DMAC waits for the remaining transfer requests in Transfer state (Example 4 in Figure 4-7).

In the case of Burst transfer, all of the $(BC+1) \times (TC+1)$ of transfers are performed when it becomes accessible to the system bus after the first transfer request is received. The required number of transfer requests from the Peripheral is only the first one. If the number of transfer request signals generated exceeds the requirement, it is ignored in Disable state, just like Block transfer.

5. Transfer state => Disable state / Successful completion of transfer

See Step 4 in the software transfer procedure.

6. Transfer state => Disable state / Transfer error stop

See Step 5 in the software transfer procedure.

7. Transfer state => Disable state / End of Peripheral stop request

The channel in Transfer state suspends its transfer processing, if the transfer stop request signal is asserted from the Peripheral. It clears EB, PB and ST and moves to Disable state. It sets "010" to SS and gives the notification of the error stop. If interrupts have been enabled by EI, an unsuccessful transfer completion interrupt occurs. BC, TC, DMACSA and DMACDA hold the values set during the suspension of the transfer. Attention must be paid to the SS value, which is the same as the stop request from software.

8. Transfer state, Pause state => Disable state / Forced termination of transfer

See Step 6 in the software transfer procedure.

9. Disable state / Post-transfer processing

See Step 7 in the software transfer procedure.

Normally, in the cases of stop request from Peripherals, forced termination from software and transfer error stop, the transfer request signal remains asserted, because the number of transfers processed is smaller than the number of transfer requests from the Peripheral. Instruct from CPU the Peripheral to deassert the transfer request signal. In the case of stop request from Peripherals, the transfer request signal is masked as long as the stop request signal is asserted. Also deassert the transfer stop request signal.

Even if DMAC has successfully completed the specified number of transfers, the transfer request signal may remain asserted or may be reasserted, depending on Peripheral's settings. Attention must be paid to the possibility that this may affect the next transfer.

10. Transfer state, Pause state / Transfer pause

See Step 8 in the software transfer procedure.

11. Pause state

See Step 9 in the software transfer procedure.

The channel in Pause state does not execute transfer, even if the transfer request signal from the Peripheral is asserted. It does not clear the transfer request signal either.

12. Pause state / Cancellation of transfer pause

See Step 10 in the software transfer procedure.

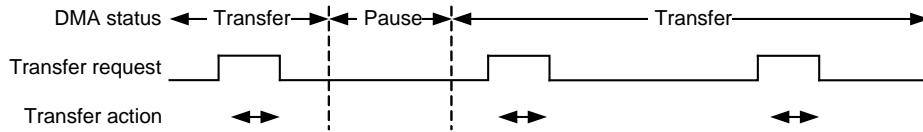
When an instruction to cancel the pause is issued while it is in Pause state, it returns to Transfer state. If the transfer request signal was asserted in the previous Pause state, the operation to follow varies as shown below, depending on the transfer mode.

In the case of Demand transfer mode, the transfer request signal remains asserted from the Pause state. Therefore, the transfer is resumed when DMAC returns to Transfer state, and the transfer request signal is cleared as normal. See Figure 4-8.

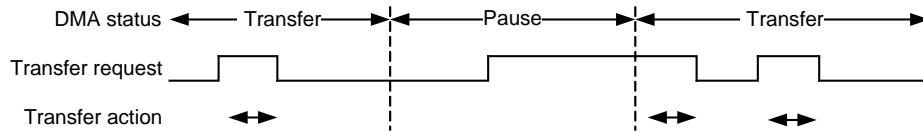
Figure 4-8 Operation of Demand Transfer in Pause State

Demand transfer mode behavior during pause state

Case of no transfer request be asserted during pause state



Case of transfer request be asserted during pause state

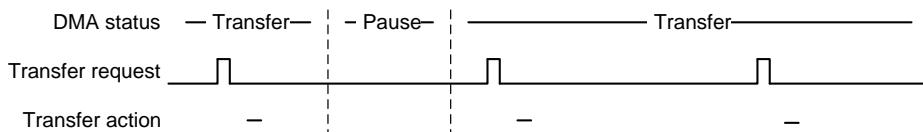


In the case of Block transfer mode, the transfer request signal remains asserted. Even when it returns to Transfer state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Pause state. Also, the transfer request signal is not cleared from DMAC. To resume the transfer which has been put on pause, instruct from CPU the Peripheral to deassert the transfer request signal after an instruction to cancel the pause is issued to DMAC. After that, the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-9.

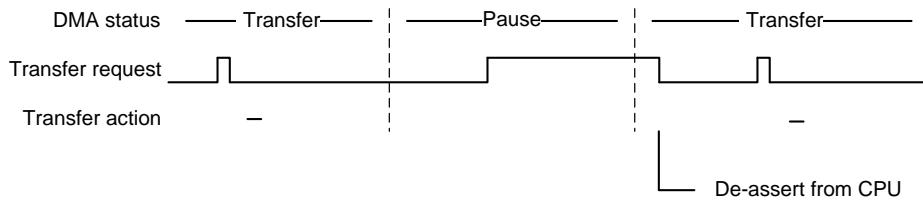
Figure 4-9 Operation of Block Transfer in Pause State

Block transfer mode behavior during pause state

Case of no transfer request be asserted during pause state



Case of transfer request be asserted during pause state



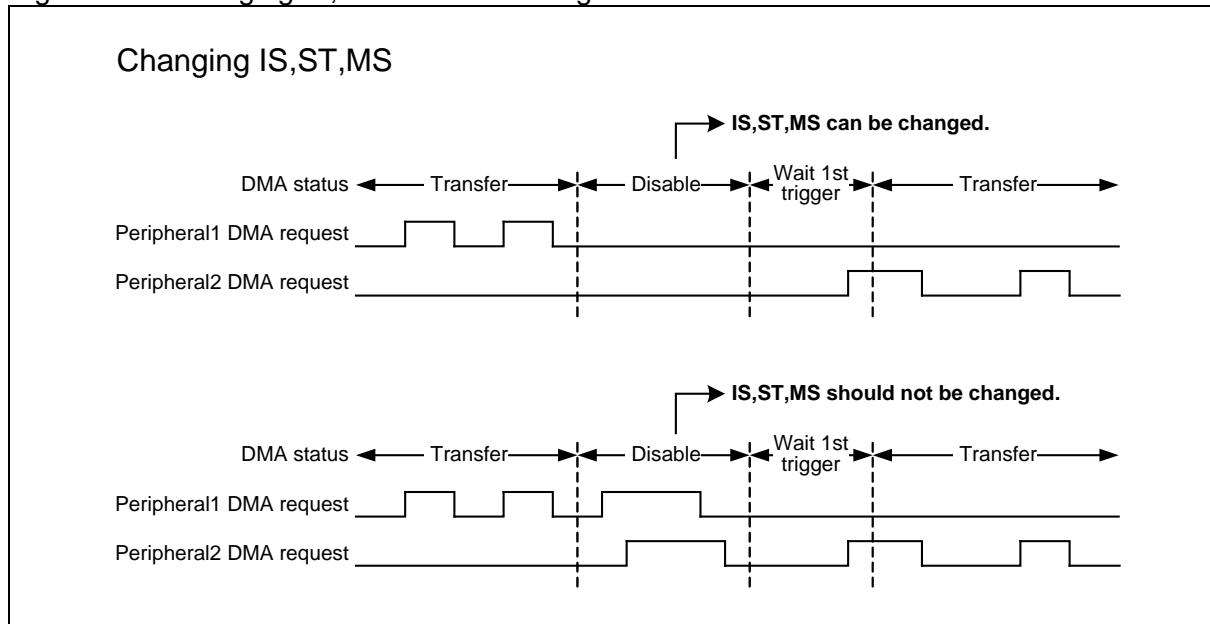
13. Operation in Disable state and Wait-1st-trigger state

See Step 11 in the software transfer procedure.

If the transfer request signal is not asserted to the channel in Disable state, the specifications of the transfer content can be changed freely (rewriting to registers DMACSA, DMACDA, DMACA[29:0], and DMACB).

If the transfer request signal is asserted or may be asserted to the channel in Disable state, the specifications of IS, ST and MS in the transfer content cannot be changed. If an attempt is made to change these settings, DMAC may perform unexpected behaviors. To change the settings of IS, ST and MS, first clear the transfer request signal to both of the Peripherals (used before and after the change) from CPU, and then always change the settings while the transfer request signal is deasserted. See Figure 4-10.

Figure 4-10 Changing IS, ST and MS Settings



The specifications of the transfer content cannot be changed to the channel in Wait-1st-trigger state from CPU

If the transfer request signal is not asserted to the channel in Wait-1st-trigger state, it moves to Disable state when CPU issues an instruction to disable individual- or all-channel operation or an instruction to put individual- or all-channel operation on pause. In this case, it is considered that the enabled transfer has been cancelled. In any case, SS does not change.

If the transfer request signal may possibly be asserted to the channel in Wait-1st-trigger state, it should be noted that DMAC has already started or completed the transfer before the attempted cancellation of the enabled transfer from CPU.

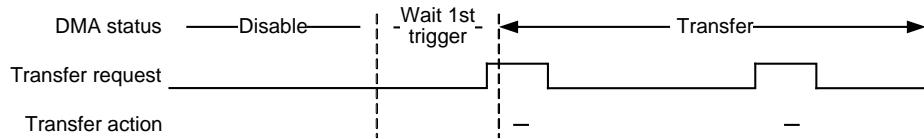
In Disable state, DMAC does not start the transfer or clear the transfer request, even if the transfer request signal is asserted. If it moves to Wait-1st-trigger state by instruction from CPU while the transfer request signal is asserted, the following operation applies (only when the settings of IS, ST and MS are not intended to be changed, as explained earlier).

In the case of Demand transfer mode, DMAC immediately moves to Transfer state and starts the transfer, because the transfer request signal remains asserted. The transfer request signal is cleared from DMAC as normal. See Figure 4-11.

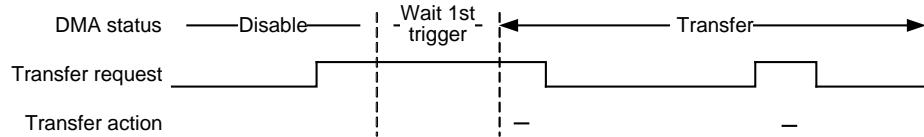
Figure 4-11 Operation of Demand Transfer in Disable State

Demand transfer mode behavior during disable state

Case of no transfer request be asserted during disable state



Case of transfer request be asserted during disable state

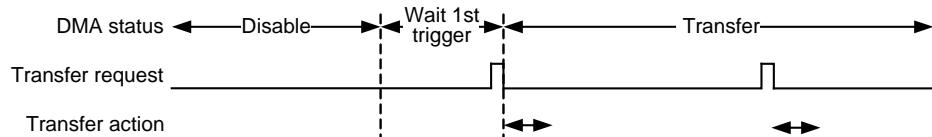


In the case of Block transfer mode, the transfer request signal remains asserted. Even when it moves to Wait-1st-trigger state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Disable state. To resume the transfer, instruct DMAC to move to Wait-1st-trigger state, and then instruct from CPU the Peripheral to deassert the transfer request signal. After that, it will move to Transfer state and the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-12.

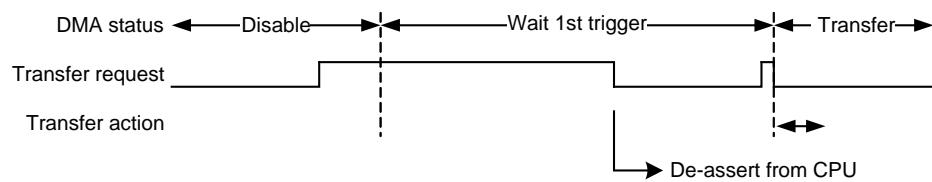
Figure 4-12 Operation of Block Transfer in Disable State

Block transfer mode behavior during disable state

Case of no transfer request be asserted during disable state



Case of transfer request be asserted during disable state



Additional Matter 1

See Additional Matter 1 in the software transfer procedure.

In the case of hardware transfer, always write "0" to ST.

Additional Matter 2

See Additional Matter 2 in the software transfer procedure.

Additional Matter 3

See Additional Matter 3 in the software transfer procedure.

Additional Matter 4

See Additional Matter 4 in the software transfer procedure.

Additional Matter 5

If the transfer request signal (interrupt signal) from the Peripheral needs to be deasserted, the following method is available. Normally, the interrupt signal from the Peripheral is the interrupt factor flag masked (logic AND) by the interrupt enable flag. The interrupt signal can be deasserted by resetting either of the flags. When the interrupt enable flag is reset and then set, the rising edge occurs to the interrupt signal. Following this procedure can notify DMAC of the transfer request for Block transfer again. For details, check the manual for each Peripheral.

4.4. DMAC Operation and Control Procedure for Hardware (EM=1) Transfer

This section describes DMAC operation and control procedure for hardware (EM=1) transfer.

Figure 4-13 Transitional Diagram of Hardware (EM=1) Transfer State

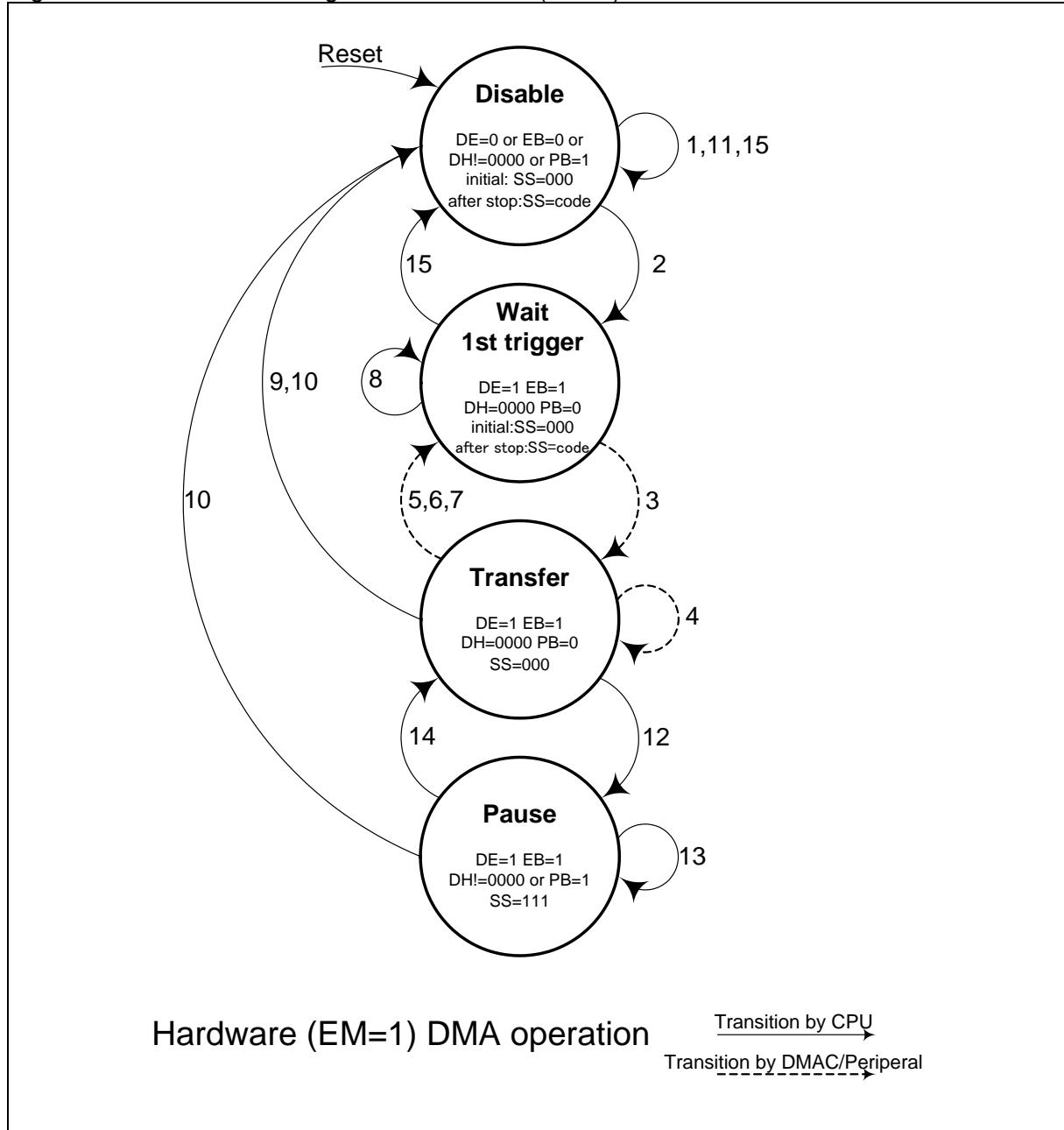


Figure 4-13 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=0) transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

EM (Enable bit clear mask) is a bit that masks EB clear upon the completion of transfer of the channel to be controlled. EM=1 enables the same transfer process to be repeated without giving instructions from CPU.

■ Description of Each State

Disable state

See the hardware transfer (EM=0) procedure.

Wait-1st-trigger state

See the hardware transfer (EM=0) procedure.

Transfer state

In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. In the case of EM=1, it moves to Wait-1st-trigger state, once all the transfer operation is completed. It also changes its state upon instruction from CPU.

Pause state

See the hardware transfer (EM=0) procedure.

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

See Step 1 in the hardware transfer (EM=0) procedure.

To set EM=1, set all of the reload specifications for the transfer content (RC, RS, RD) in order to prevent data transfer in an unintended address area. Also, CI is not set, because it is meaningless to generate a successful transfer completion interrupt from DMAC. EI is set to generate an unsuccessful transfer completion interrupt from DMAC.

2. Disable state => Wait-1st-trigger state / Enabling transfer

See Step 2 in the hardware transfer (EM=0) procedure.

3. Wait-1st-trigger state / Start of transfer

See Step 3 in the hardware transfer (EM=0) procedure.

4. Transfer state

See Step 4 in the hardware transfer (EM=0) procedure.

5. Transfer state => Wait-1st-trigger state / Successful completion of transfer

When transfers are successfully completed for the number of times calculated by $(BC+1) \times (TC+1)$, the channel in Transfer state does not clear EB but does clear PB and ST and moves to Wait-1st-trigger. It sets SS=101 to provide the notification of the successful completion. As CI is not set, no successful transfer completion interrupt is generated. Since RC, RS and RD are set, the specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded.

6. Transfer state => Wait-1st-trigger state / Transfer error end

See Step 6 in the hardware transfer (EM=0) procedure.

In the case of EM=1, EB is not cleared even if the transfer ends due to an error. It clears PB and ST, moves to Wait-1st-trigger state and waits for the next transfer request. Therefore, it is recommended not to use DMA transfer with EM=1 in an address area where a transfer error may occur.

7. Transfer state => Wait-1st-trigger state / End of Peripheral stop request

See Step 7 in the hardware transfer (EM=0) procedure.

In the case of EM=1, EB is not cleared even if a stop request is issued from the Peripheral. It clears PB and ST and moves to Wait-1st-trigger state. Since RC, RS and RD are set, the specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded. As EI is set, an unsuccessful transfer completion interrupt is generated.

8. Wait-1st-trigger state / Post-transfer process

In the case of EM=1, EB is not cleared upon the completion of the transfer. (DE=1, EB=1, DH=0000, PB=0) is set and it moves to Wait-1st-trigger state. When the next transfer request is generated from the Peripheral, therefore, the next transfer starts without an instruction from CPU.

If it moves to Wait-1st-trigger state due to a stop request from the Peripheral, an unsuccessful completion interrupt occurs and that state can be confirmed. Also, the transfer request signal is masked while the stop request signal is asserted. Even if the next transfer request signal is asserted from the Peripheral, it will not be recognized and the channel to be controlled will remain in Wait-1st-trigger state, waiting for an instruction from CPU.

In the above case, SS is read from CPU to check the state of the transfer completion. The interrupt signal is deasserted by clearing SS from CPU. CPU clears EB and it returns to Disable state (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure). The transfer request signal and the stop request signal from the Peripheral are deasserted, as shown in Step 7 of the hardware transfer (EM=0) procedure

9. Transfer state => Disable state / Completion of transfer by EM=0

The operation can exit from the loop of Wait-1st-trigger state and Transfer state by writing EM=0 from CPU. At the timing when the transfer stops after the instruction, EB, ST and PB are cleared and the Transfer state changes to Disable state (DE=1, EB=0, DH=0000, PB=0) to successfully complete the transfer. In this case, no successful transfer completion interrupt is generated, as CI is not set.

10. Transfer state, Pause state => Disable state / Forced termination of transfer

See Step 8 in the hardware transfer (EM=0) procedure.

The operation can exit from the loop of Wait-1st-trigger state and Transfer state by an operation disable instruction. When an instruction to disable individual-channel operation is issued, the relevant channel moves to Disable state (DE=1, EB=0, DH=0000, PB=0) and stops the operation. When an instruction to enable all-channel operation is issued, it moves to Disable state (DE=0, EB=1, DH=0000, PB=0) and stops the operation. In the case of an instruction to disable all-channel operation, EB is not cleared either; therefore, attention must be paid.

When the operation exits from Transfer state, an unsuccessful transfer completion interrupt occurs because it is unsuccessful completion due to the forced stop. When it exits from Wait-1st-trigger state, the enabled transfer is cancelled (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure).

11. Disable state / Post-transfer processing

See Step 9 in the hardware transfer (EM=0) procedure.

12. Transfer state, Pause state / Transfer pause

See Step 10 in the hardware transfer (EM=0) procedure.

13. Pause state

See Step 11 in the hardware transfer (EM=0) procedure.

14. Pause state / Cancellation of transfer pause

See Step 12 in the hardware transfer (EM=0) procedure.

15. Operation in Disable state and Wait-1st-trigger state

See Step 13 in the hardware transfer (EM=0) procedure.

In the case of EM=1, the Transfer state changes directly to Wait-1st-trigger state. Therefore, the specifications of the transfer content cannot be rewritten during the repeated transfer operation (rewriting the registers DMACSA, DMACDA, DMACB[31:1] and DMACA[28:0]).

Additional Matter 1

See Additional Matter 1 in the hardware transfer (EM=0) procedure.

Additional Matter 2

See Additional Matter 2 in the hardware transfer (EM=0) procedure.

In the case of EM=1, Additional Matter 2 does not apply, because EB is not cleared during the transfer operation.

Additional Matter 3

See Additional Matter 3 in the hardware transfer (EM=0) procedure.

Additional Matter 4

See Additional Matter 4 in the hardware transfer (EM=0) procedure.

The following explains what must be noted when setting interrupts from DMAC with EM=1. As the target channel does not change from Wait-1st-trigger state due to an unsuccessful completion interrupt by a stop request from the Peripheral, the interrupt signal is not deasserted until it is cleared from CPU. Similarly, as the target channel moves to Disable state due to an unsuccessful transfer completion interrupt by a stop request from software, the interrupt signal is not deasserted until it is cleared from CPU. Other successful transfer completion interrupts and unsuccessful transfer completion interrupts may be deasserted at a timing that is not intended by CPU, if the relevant channel moves to Transfer state. Therefore, attention must be paid.

Additional Matter 5

See Additional Matter 5 in the hardware transfer (EM=0) procedure.

5. Registers of DMAC

This chapter describes each register function of DMAC.

- 5.1 List of Registers
- 5.2 Entire DMAC Configuration Register (DMACR)
- 5.3 Configuration A Register (DMACA)
- 5.4 Configuration B Register (DMACB)
- 5.5 Transfer Source Address Register (DMACSA)
- 5.6 Transfer Destination Address Register (DMACDA)
- 5.7 Notes on Register Setting

5.1. List of Registers

Table 5-1 shows a list of DMAC control registers.

Table 5-1 List of DMAC Control Registers

Abbreviation	Ch. Controlled	Register name	See
DMACR	All	Entire DMAC configuration register	5.2
DMACA0	ch.0	Configuration A register	5.3
DMACB0		Configuration B register	5.4
DMACSA0		Transfer source address register	5.5
DMACDA0		Transfer destination address register	5.6
DMACA1	ch.1	Configuration A register	5.3
DMACB1		Configuration B register	5.4
DMACSA1		Transfer source address register	5.5
DMACDA1		Transfer destination address register	5.6
DMACA2	ch.2	Configuration A register	5.3
DMACB2		Configuration B register	5.4
DMACSA2		Transfer source address register	5.5
DMACDA2		Transfer destination address register	5.6
DMACA3	ch.3	Configuration A register	5.3
DMACB3		Configuration B register	5.4
DMACSA3		Transfer source address register	5.5
DMACDA3		Transfer destination address register	5.6
DMACA4	ch.4	Configuration A register	5.3
DMACB4		Configuration B register	5.4
DMACSA4		Transfer source address register	5.5
DMACDA4		Transfer destination address register	5.6
DMACA5	ch.5	Configuration A register	5.3
DMACB5		Configuration B register	5.4
DMACSA5		Transfer source address register	5.5
DMACDA5		Transfer destination address register	5.6
DMACA6	ch.6	Configuration A register	5.3
DMACB6		Configuration B register	5.4
DMACSA6		Transfer source address register	5.5
DMACDA6		Transfer destination address register	5.6
DMACA7	ch.7	Configuration A register	5.3
DMACB7		Configuration B register	5.4
DMACSA7		Transfer source address register	5.5
DMACDA7		Transfer destination address register	5.6

5.2. Entire DMAC Configuration Register (DMACR)

This section describes entire DMAC configuration register (DMACR).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DE	DS	-	PR		DH[3:0]		-	-	-	-	-	-	-	-	-
Attribute	R/W	R/W	R/W	R/W		R/W		R/W								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31] DE : DMA Enable (all-channel operation enable bit)

This bit controls the enabling and disabling of transfer operations for all of the channels.

When "1" is set to this bit, the operations of all of the channels are enabled and each channel operates according to its settings.

When "0" is set to this bit, the operations of all of the channels are disabled, and no transfer is performed until "1" is set to the bit. Also, a channel in the middle of its transfer operation is forced to stop the transfer.

This bit can be used to force all of the channels that are currently performing a transfer to stop it and reset the configuration register.

bit31	Function
0	Disables the operations of all of the channels. (Initial value)
1	Enables the operations of all of the channels.

[bit30] DS : DMA Stop

This bit indicates the transfer state of all of the channels.

If either of the following conditions is established during transfer operation, the bit is set to "1" by DMAC.

- When "0" is written to the DMACR/DE bit and then the transfers of all of the channels are completed.
- When other than "0000" is written to the DMACR/DH bit and then the transfers of all of the channels pause.

When DMACR/DE=1 and DMACR/DH=0000 are set and all of the channels become enabled to operate, this bit is set to "0" by DMAC.

Although the attribute of this bit is R/W, writing to it by CPU does not affect DMAC's operation. If, however, the DMACR register needs to be updated without affecting the state of this bit, first read from this bit and then rewrite the same value.

bit30	Function
0	Clears the disabling of all-channel operation or the setting of all-channel pause. (Initial value)
1	The transfers of all of the channels have stopped due to the disabling of all-channel operation or the setting of all-channel pause.

[bit29] Reserved

[bit28] PR : Priority Rotation

This bit controls the order of transfer priority among channels.

When this bit is set to "0", the priority order is fixed for all of the channels.

When this bit is set to "1", the priority order is determined in a rotation method for all of the channels.

bit28	Function
0	Fixes the priority order. (ch.0>ch.1>ch.2>ch.3>ch.4>ch.5>ch.6>ch.7) (Initial value)
1	Applies the rotation method to the priority order.

For selection of the transfer priority order, see Section 3.5.

[bit27:24] DH : DMA Halt (All-channel pause bit)

This bit controls the pause/cancellation of transfer operations for all of the channels.

When this bit is set to a value other than "0000", all of the channels that are currently performing a transfer are put on pause. When it is set to "0000", the transfers are resumed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration registers of all of the channels.

bit27:24	Function
0000	Cancels the pause of transfers for all of the channels. (Initial value)
Other than 0000	Puts the transfers of all of the channels on pause.

5.3. Configuration A Register (DMACA)

This section describes configuration A register (DMACA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	EB	PB	ST				IS[5:0]			-	-	-		BC[3:0]		
Attribute	R/W	R/W	R/W				R/W			R/W	R/W	R/W		R/W		R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field								TC[15:0]								
Attribute									R/W							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31] EB : Enable Bit (individual-channel operation enable bit)

This bit controls the enabling and disabling of the transfer operation of an individual channel.

When this bit is set to "1", the relevant channel is enabled to operate and waits for a trigger to start its transfer operation (the DMACR/DE must be set to "1").

If the EM bit (DMACB[0]) is not set to "1", DMAC clears this bit to "0" upon the completion of the transfer.

When this bit is set to "0", the relevant channel is disabled to operate and does not perform transfer operation until it is set to "1". Also, if it is in the middle of transfer operation, it is forced to stop the transfer. This bit can be used to force the relevant channel that is currently in transfer operation to stop it and reset the configuration register.

bit31	Function
0	The operation of the relevant channel is disabled. (Initial value)
1	The operation of the relevant channel is enabled.

[bit30] PB : Pause Bit (individual-channel pause bit)

This bit controls the pause/cancellation of the transfer operation of an individual channel.

When this bit is set to "1" and the relevant channel is currently in transfer operation, it puts the transfer on pause. When this bit is set to "0", it resumes the transfer.

This bit is cleared to "0", when the transfer operation of the channel is completed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration register of the relevant channel.

bit30	Function
0	Cancels the pause of the transfer of the relevant channel.
1	Puts the transfer of the relevant channel on pause.

[bit29] ST : Software Trigger

This bit is used to generate a software transfer request for an individual channel.

When this bit is set to "1", a trigger is generated by the software transfer request and the relevant channel starts its transfer. After the completion of the transfer, DMAC clears this bit to "0".

When this bit is set to "0" during the transfer, the transfer stops.

bit29	Function
0	No software transfer request (Initial value)
1	Software transfer request available

[bit28:23] IS : Input Select

These bits select the trigger for transfer requests.

When the transfer trigger is set to software request (ST=1), set the IS bit to "000000".

When the transfer trigger is set to hardware request, specify which Peripheral's interrupt signal to be used to start transfer. Any Peripheral can be selected for all of the channels.

The hardware transfer request signal to be connected to DMAC varies depending on the model. Check the transfer request signal to be connected in "2.2 I/O Signals of DMAC" before setting the selection.

bit28:23	Function
000000	Software (Initial value)
100000	IDREQ[0]
100001	IDREQ[1]
100010	IDREQ[2]
100011	IDREQ[3]
100100	IDREQ[4]
100101	IDREQ[5]
100110	IDREQ[6]
100111	IDREQ[7]
101000	IDREQ[8]
101001	IDREQ[9]
101010	IDREQ[10]
101011	IDREQ[11]
101100	IDREQ[12]
101101	IDREQ[13]
101110	IDREQ[14]
101111	IDREQ[15]
110000	IDREQ[16]
110001	IDREQ[17]
110010	IDREQ[18]
110011	IDREQ[19]

bit28:23	Function
110100	IDREQ[20]
110101	IDREQ[21]
110110	IDREQ[22]
110111	IDREQ[23]
111000	IDREQ[24]
111001	IDREQ[25]
111010	IDREQ[26]
111011	IDREQ[27]
111100	IDREQ[28]
111101	IDREQ[29]
111110	IDREQ[30]
111111	IDREQ[31]
Setting other than above	Setting prohibited

[bit22:20] : Reserved

[bit19:16] BC : Block Count

These bits specify the number of blocks for Block/Burst transfer.

When the transfer mode is set to Demand transfer, set BC to "0000".

Set the value "BC = Number of blocks - 1". The maximum allowed number of blocks is 16.

The value of these bits can be read during a transfer. Normally, as one transfer source access or one transfer destination access is completed successfully, BC is decreased by 1.

In the case of RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.

In the case of RC=0, the value is set to "0" upon successful completion of the transfer, while the value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit19:16	Function
xxxx	Number of transfer blocks (Initial value : 4'b0000)

[bit15:0] TC : Transfer Count

These bits specify the number of transfers for Block/Burst/Demand transfer.

Set the value "TC = Number of transfers - 1". The maximum allowed number of transfers is 65536.

The value of these bits can be read during a transfer. Normally, as the transfer of one block is completed, TC is decreased by 1.

In the case of RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.

In the case of RC=0, the value is set to "0" upon successful completion of the transfer, while the value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit15:0	Function
16'hxxxx	Number of transfers (Initial value : 16'h0000)

5.4. Configuration B Register (DMACB)

This section describes configuration B register (DMACB).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	-	MS[1:0]	TW[1:0]	FS	FD	RC	RS	RD	EI	CI			SS[2:0]			
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W0		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EM
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:30] Reserved

[bit29:28] MS : Mode Select

These bits select the transfer mode.

bit29:28	Function
00	Block transfer mode (Initial value)
01	Burst transfer mode
10	Demand transfer mode
11	Reserved

[bit27:26] TW : Transfer Width

These bits specify the bit width of transfer data.

bit27:26	Function
00	Byte (8bit) (Initial value)
01	Half-word (16bit)
10	Word (32bit)
11	Reserved

[bit25] FS : Fixed Source

This bit specifies whether to increment or fix the transfer source address.

bit25	Function
0	Increments the transfer source address according to TW. (Initial value)
1	Fixes the transfer source address.

[bit24] FD : Fixed Destination

This bit specifies whether to increment or fix the transfer destination address.

bit24	Function
0	Increments the transfer destination address according to TW. (Initial value)
1	Fixes the transfer destination address.

[bit23] RC : Reload Count (BC/TC reload)

This bit controls the reload function of BC and TC.

When this bit is set to "1", the value set when the transfer started is reloaded to BC and TC upon completion of the transfer.

bit23	Function
0	Disables the reload function of BC/TC. (Initial value)
1	Enables the reload function of BC/TC.

[bit22] RS : Reload Source

This bit controls the reload function of the transfer source address.

When this bit is set to "1", the value set when the transfer started is reloaded to DMACSA upon completion of the transfer.

bit22	Function
0	Disables the reload function of the transfer source address. (Initial value)
1	Enables the reload function of the transfer source address.

[bit21] RD : Reload Destination

This bit controls the reload function of the transfer destination address (DMACDA).

When this bit is set to "1", the value set when the transfer started is reloaded to DMACDA upon completion of the transfer.

bit21	Function
0	Disables the reload function of the transfer destination address. (Initial value)
1	Enables the reload function of the transfer destination address.

[bit20] EI :Error Interrupt (unsuccessful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been unsuccessfully completed.

When this bit is set to "1", an interrupt is issued if SS is in the following status upon completion of the transfer.

- Address overflow
- Stop by transfer stop request from a Peripheral, or the disabling of transfer by the EB/DE bit
- Transfer source access error
- Transfer destination access error

bit20	Function
0	Disables an interrupt to be issued upon unsuccessful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon unsuccessful completion of transfer.

[bit19] Completion Interrupt : (successful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been successfully completed.

When this bit is set to "1", an interrupt is generated, if SS is set to successful completion upon completion of the transfer.

bit19	Function
0	Disables an interrupt to be issued upon successful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon successful completion of transfer.

[bit18:16] SS : Stop Status (stop status notification)

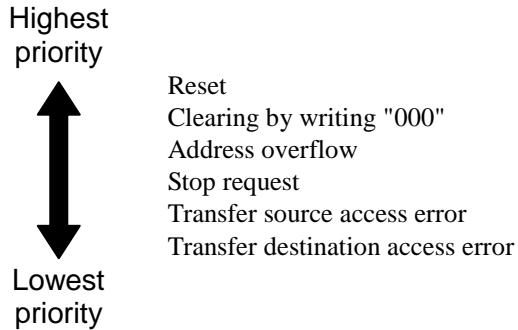
These bits represent a code that indicates the stop status or completion status of a transfer.

The following table shows the available codes.

If a successful transfer completion interrupt or unsuccessful transfer completion interrupt is issued, the interrupt signal is deasserted by writing "000" to these bits.

bit18:16	Description
000	Initial value
001	Termination by transfer error (address overflow)
010	Termination by transfer stop request (stop by transfer stop request for Peripheral or the disabling of transfer by the EB/DE bit)
011	Termination by transfer error (transfer source access error)
100	Termination by transfer error (transfer destination access error)
101	Successful transfer completion
110	Reserved
111	Transfer on pause

If various errors occur simultaneously, the termination code is indicated according to the following priority.



[bit15:8] Reserved

When writing, always write "0". "0" is always read.

[bit7:1] Reserved

[bit0] EM : Enable bit Mask (EB bit clear mask)

This bit is used to mask the clear of the EB bit (DMACA[31]) from DMAC upon completion of the transfer.

In the case of EM=0, DMAC clears the EB bit (DMACA[31]) to "0" upon completion of the transfer.

In the case of EM=1, it does not clear the EB bit upon completion of the transfer. This function allows transfers to be repeated without instruction from CPU.

This function can only be used for hardware transfer. To use the function, enable the reload function of RC, RS and RD.

bit0	Function
0	Clears EB upon completion of the transfer. (Initial value)
1	Does not clear EB upon completion of the transfer.

5.5. Transfer Source Address Register (DMACSA)

This section describes transfer source address register (DMACSA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACSA[31:16]															
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACSA[15:0]															
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:0] DMACSA : DMAC Source Address

These bits specify the transfer start address of the transfer source.

It is not possible to set unaligned address to transfer data width (TW). The value of these bits can be read during the transfer.

In the case of FS=1, the transfer source address is set to a fixed value and no change occurs.

In the cases of FS=0 and RS=0, the value is incremented according to TW. Upon successful transfer completion, it is the next address after the transfer completion address. Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of FS=0 and RS=1, it is incremented according to TW during the transfer. Upon completion of the transfer, the value set when the transfer started is reloaded.

bit31:0	Function
32'hxxxxxxxxx	Specifies the transfer source address from which the transfer starts. (Initial value: 32'h00000000)

5.6. Transfer Destination Address Register (DMACDA)

This section describes transfer destination address register (DMACDA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACDA[31:16]															
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACDA[15:0]															
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:0] DMACDA : DMAC Destination Address

These bits specify the transfer start address of the transfer destination.

It is not possible to set unaligned address to transfer data width (TW). The value of these bits can be read during the transfer.

In the case of FD=1, the transfer destination address is set to a fixed value and no change occurs.

In the cases of FD=0 and RD=0, the value is incremented according to TW. Upon successful transfer completion, it is the next address after the transfer completion address. Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of FD=0 and RD=1, it is incremented according to TW during the transfer. Upon completion of the transfer, the value set when the transfer started is reloaded.

bit31:0	Function
32'hxxxxxxxxx	Transfer destination address from which DMA transfer starts (Initial value: 32'h00000000)

5.7. Notes on Register Setting

Care must be taken on the following matters when setting the DMAC registers.

- The DMACR, DMACA, DMACB, DMACSA and DMACDA registers can be accessed by byte, half-word and word.
- The register address in DMAC cannot be set to the DMACSA or DMACDA register.
- Channel setting registers cannot be changed during DMA transfer, except the DE/DH bits of DMACR, the EB/PB bits of DMACA and the EM bit of DMACB.

CHAPTER: I/O PORT

This chapter explains the I/O port.

1. Overview
2. Configuration, Block Diagram, and Operation
3. Setup Procedure Example
4. Register List
5. Usage Precautions

1. Overview

This section provides an overview of the I/O port.

The I/O port of this series provides the following features.

- The I/O port of this series shares the following functions.
 - GPIO
General-purpose I/O ports, which can read an input level and set an output level from the CPU.
 - Peripheral input/output
Digital input/output signal ports of peripheral functions.
 - Special I/O ports
 - Analog input port
An analog input port of an A/D converter.
 - USB port
 - Oscillation port
- The followings settings can be made for each pin.
 - You can set whether the I/O port will be used as a GPIO, a digital pin of peripheral functions, or a special pin.
 - You can set whether the I/O port will be used as an input port or an output port.
 - You can enable or disable pull-up.
 - Peripheral functions are assigned to two or more I/O ports with input/output of the same function. You can set to which I/O port the function can be allocated (relocation function).
 - By setting registers, you can set the I/O port to Hi-Z status while the CPU is in standby mode.

2. Configuration, Block Diagram, and Operation

This section explains the configuration, block diagram, and operation of the I/O port.

■ Configuration of the I/O Port

By setting registers of the I/O port, select Input/Output direction and select GPIO/peripheral. Figure 2-1 shows the details of the I/O port.

Figure 2-1 Block Diagram of the I/O Port

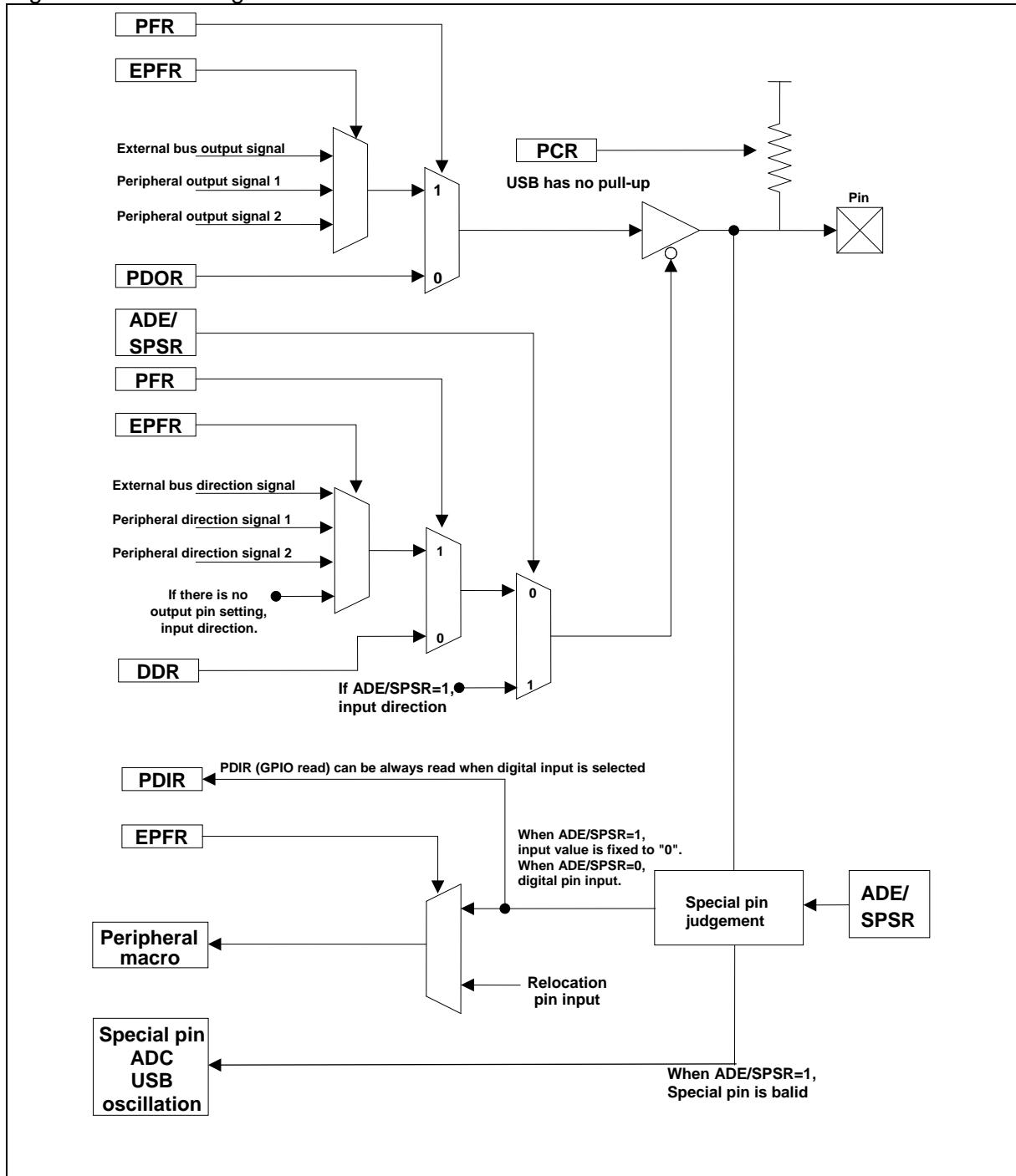


Table 2-1 describes register function.

- The PFR, DDR, PDIR, PDOR, and PCR register have 1-bit control register for each I/O port and select a function for the I/O port.
- The ADE register has 1-bit control register for each I/O port which doubles as an analog input pin and selects a function for the I/O port.
- The SPSR register selects a function for the I/O port which doubles as a USB pin or an oscillation pin.
- The EPFR register has control register for each I/O pin of peripheral functions and selects to which I/O port an I/O pin of peripheral functions will be relocated.

Table 2-1 Register Function Descriptions

Register name	Function description
ADE	A register to set whether the I/O port will be used as a special pin (an analog input pin) or a digital input/output pin.
SPSR	A register to set whether the I/O port will be used as a special pin (USB or oscillation) or a digital input/output pin.
PFR	A register to set whether the I/O port will be used as a GPIO function or an input/output pin of peripheral functions.
PCR	A register to set whether a pull-up resistor of the I/O port will be connected or disconnected if the I/O port is used as a digital input pin or a digital bidirectional pin.
DDR	A register to set whether the I/O port will be used as an input pin or an output pin if the I/O port is used as a GPIO function pin. Note: If a pin is selected as an I/O pin of peripheral functions, a setting value is invalid.
PDIR	A register to read the level status of the I/O port. <ul style="list-style-type: none"> • If the I/O port is used as a digital input pin, it reads input level. • If the I/O port is used as a digital output pin, it reads output level. • If the I/O port is used as an analog input pin, it always reads "0".
PDOR	A register to set output level if the I/O port is used as an output pin of GPIO function. <ul style="list-style-type: none"> • When "0" is set, it outputs Low level. • When "1" is set, it outputs High level. Note: If a pin is selected as GPIO input or intput/output of peripheral functions, a setting value is invalid.
EPFR	A register to select a function for an input/output of peripheral functions and set relocation function. <ul style="list-style-type: none"> • Setting a peripheral output pin It sets whether to produce output for the I/O port or not. In addition, it can also set to which I/O port a pin of peripheral functions will be relocated for each pin. • Setting a peripheral input pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin. • Setting a peripheral bidirectional pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin

Table 2-2 lists pin functions which availability depends on selected I/O port functions and register setting values.

Table 2-2 I/O Port Functions and Register Setting Values

I/O Port Function		ADE/ SPSR	PFR	DDR	PCR	EPFR
Available main function	Available sub function					
Special pin Analog input USB Oscillation	N/A	1	-	-	Disconnect	*0
GPIO function input pin	Peripheral function input pin	0	0	0	Valid	*1
GPIO function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)			1	Disconnect	
Peripheral function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)		1	-	Disconnect	*2
Peripheral function bidirectional pin	GPIO function input pin (FB) Peripheral function input pin (FB)	1	-	Valid	Valid	*3
Peripheral function input pin	GPIO function input pin			Valid	Valid	*4

Legends

- : Indicates that a register setting value does not affect pin functions.
 - Valid : Indicates that a pull-up resistor is disconnected if PCR register value is 0.
Indicates that a pull-up resistor is connected if PCR register value is 1.
 - Disconnect : Indicates that a pull-up resistor is disconnected regardless of PCR register value.
 - (FB) : Indicates that an output signal of the I/O port provides feedback and the level of the I/O port can be read from PDIR. The signal can be also used as input for peripheral functions.
- *0 : If the input pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.
- *1 : If the input pin of peripheral functions is selected for the I/O port, the setting is valid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.
- *2 : Indicates that the output pin of peripheral functions is selected for the I/O port.
- *3 : Indicates that the bidirectional pin of peripheral functions is selected for the I/O port.
- *4 : Indicates that neither the output pin nor the bidirectional pin of peripheral functions is selected for the I/O port.

■ Initially Selected Functions for the I/O Port

Table 2-3 describes initially selected functions for each I/O port after reset is released.

Table 2-3 Initially Selected Functions for Each I/O Port after Reset Is Released

No	Pin	Initially selected function
1	TRSTX, TCK, TDI, TMS, TDO	JTAG pin is selected. Pull-up is enabled.
2	ANxx	Can be used as an analog input pin. Digital input is cut off and "0" is input.
3	X0A, X1A	Can be used as an oscillation pin. Digital input is cut off and "0" is input.
4	All GPIO pins other than the above pins	Digital input. Output is Hi-Z.

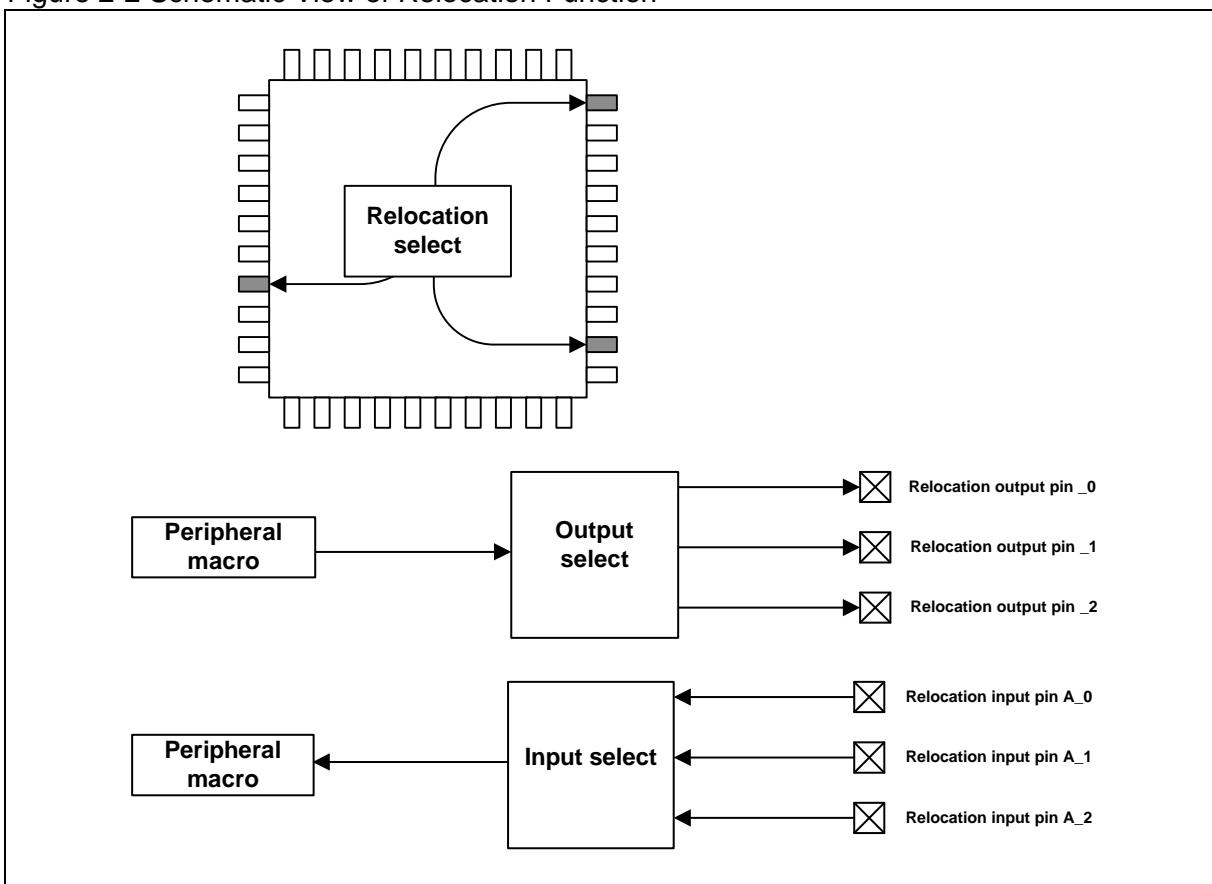
Note: For the status of pins other than GPIO (a MD pin, a reset pin), see "Data Sheet".

All the output selection values of EPFR during reset are "no output".

■ About Relocation Function

- Some input/output of peripheral functions have more than one pin (relocation pin). One of the pins can be selected by setting EPFR. Figure 2-2 show the schematic view of relocation function.

Figure 2-2 Schematic View of Relocation Function

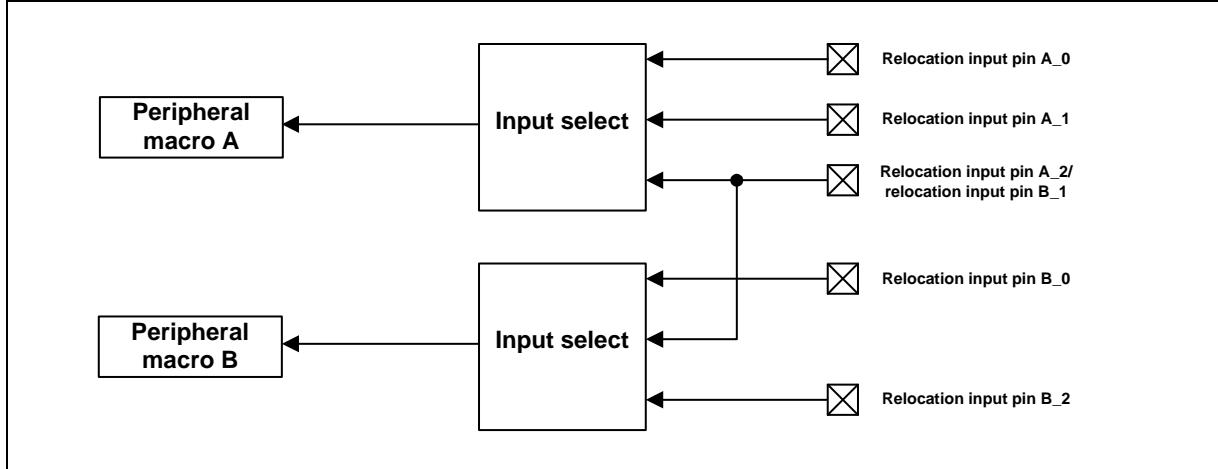


Note: Which peripheral function is allocated to which pin depends on products.

See the pin function list of "Data Sheet".

- Even if the input of one I/O port is connected to two or more peripheral functions, all peripheral inputs can be used by setting EPFR. For example, in Figure 2-3, by selecting input for both "Relocation input pin A_2" and "Relocation input pin B_1", simultaneous usage is possible. In this way, it is possible to use external interrupt and a multi-function serial input pin shared by one I/O port simultaneously.

Figure 2-3 Multiple Peripheral Inputs



- Even if an I/O cell pin is set as output, it can work as an input pin because input is not masked. For example, timer output can be used as external interrupt input which shared.

■ About Fixed Priority of EPFR Outputs

- Only one output pin function among two or more outputs is allocated to one I/O port.
- By setting the EPFR register, if more than one output is set, fixed priority is applied and output pins are selected. Figure 2-4 shows output pins and fixed priority.

Figure 2-4 Output Pins and Fixed Priority

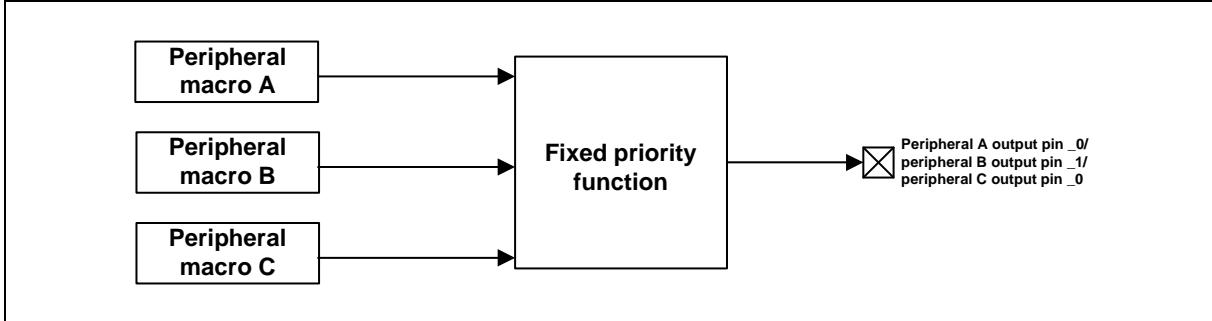


Table 2-4 describes fixed priority of EPFR.

Table 2-4 Fixed Priority of EPFR

Higher	Peripheral function	Applied pin
↓	JTAG, trace	Output pin, I/O pin
↓	USB	I/O pin
↓	CAN	Output pin
↓	Multi-function serial	Output pin, I/O pin
↓	Base timer output	I/O pin
↓	Multi-function timer	Output pin
↓	External bus	Output pin, I/O pin
↓	Internal CR waveform output	Output pin
Lower		

Note: Fixed priority is only applicable when function is set to "output". In case of "input", there is no fixed priority.

- Due to output setting on the lower part of the priority, the EPFR register always includes "no output" setting.
- If you are going to use a pin as an input pin of peripheral functions, disable all shared output settings. If every output of a pin is not selected, the pin works as an external input pin.

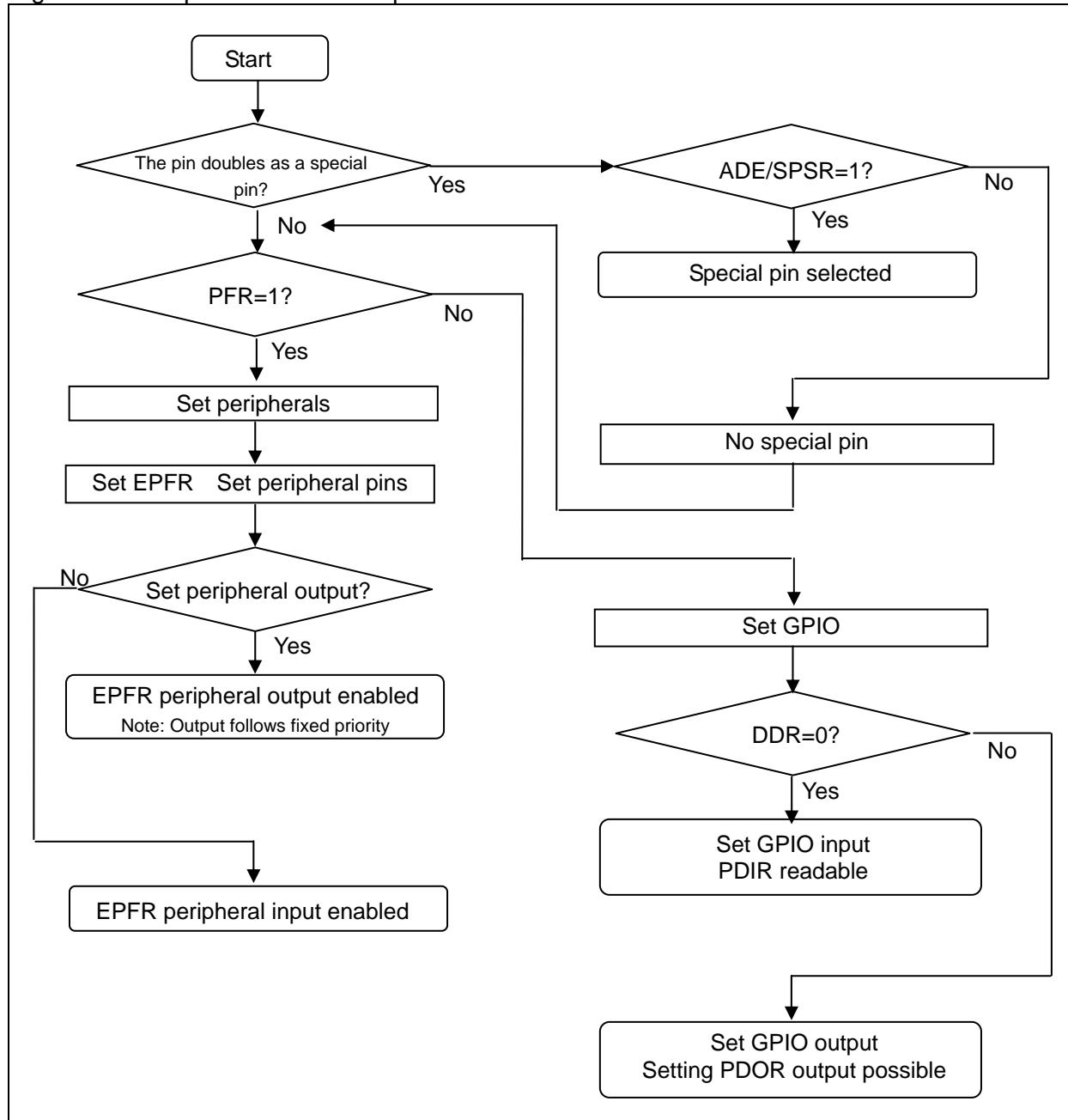
3. Setup Procedure Example

This section explains a procedure example of setting up the I/O port.

■ Setup of the I/O Port

By setting registers of the I/O port, select I/O direction and select GPIO/peripheral.
Figure 3-1 shows a setup procedure example.

Figure 3-1 Setup Procedure Example of the I/O Port



4. Register List

This section provides the register list of the I/O port.

Table 4-1 provides the register list.

Table 4-1 Register List of the I/O Port

Abbreviation	Register name	See
PFR0	Port function setting register 0	4.1
PFR1	Port function setting register 1	
PFR2	Port function setting register 2	
PFR3	Port function setting register 3	
PFR4	Port function setting register 4	
PFR5	Port function setting register 5	
PFR6	Port function setting register 6	
PFR7	Port function setting register 7	
PFR8	Port function setting register 8	
PCR0	Pull-up setting register 0	4.2
PCR1	Pull-up setting register 1	
PCR2	Pull-up setting register 2	
PCR3	Pull-up setting register 3	
PCR4	Pull-up setting register 4	
PCR5	Pull-up setting register 5	
PCR6	Pull-up setting register 6	
PCR7	Pull-up setting register 7	
DDR0	Port input/output direction setting register 0	4.3
DDR1	Port input/output direction setting register 1	
DDR2	Port input/output direction setting register 2	
DDR3	Port input/output direction setting register 3	
DDR4	Port input/output direction setting register 4	
DDR5	Port input/output direction setting register 5	
DDR6	Port input/output direction setting register 6	
DDR7	Port input/output direction setting register 7	
DDR8	Port input/output direction setting register 8	
PDIR0	Port input data register 0	4.4
PDIR1	Port input data register 1	
PDIR2	Port input data register 2	

Abbreviation	Register name	See
PDIR3	Port input data register 3	4.4
PDIR4	Port input data register 4	
PDIR5	Port input data register 5	
PDIR6	Port input data register 6	
PDIR7	Port input data register 7	
PDIR8	Port input data register 8	
PDOR0	Port output data register 0	4.5
PDOR1	Port output data register 1	
PDOR2	Port output data register 2	
PDOR3	Port output data register 3	
PDOR4	Port output data register 4	
PDOR5	Port output data register 5	
PDOR6	Port output data register 6	
PDOR7	Port output data register 7	
PDOR8	Port output data register 8	
ADE	Analog input setting register	4.6
SPSR	Special Port Setting Register	4.18
EPFR00	Extended pin function setting register 00	4.8
EPFR01	Extended pin function setting register 01	4.9
EPFR02	Extended pin function setting register 02	4.10
-	Reserved	-
EPFR04	Extended pin function setting register 04	4.11
EPFR05	Extended pin function setting register 05	4.12
EPFR06	Extended pin function setting register 06	4.13
EPFR07	Extended pin function setting register 07	4.14
EPFR08	Extended pin function setting register 08	4.15
EPFR09	Extended pin function setting register 09	4.16
EPFR10	Extended pin function setting register 10	4.17

4.1. Port Function Setting Register (PFRx)

The PFRx register selects usage of a pin.

■ List of PFR Register Configuration

31	16	15	0	Initial value	Attribute	Support
reserved		PFR0		0x001F	R/W	P0F to P00
reserved		PFR1		0x0000	R/W	P1F to P10
reserved		PFR2		0x0000	R/W	P2F to P20
reserved		PFR3		0x0000	R/W	P3F to P30
reserved		PFR4		0x0000	R/W	P4F to P40
reserved		PFR5		0x0000	R/W	P5F to P50
reserved		PFR6		0x0000	R/W	P6F to P60
reserved		PFR7		0x0000	R/W	P7F to P70
reserved		PFR8		0x0000	R/W	P8F to P80

■ Detailed Register Configuration

bit	31	16	15	0
Field	reserved			PFRx

■ Register Function

[bit31:16] res : Reserved Bit

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PFRx : Port Function Setting Register x

Selects usage of a pin.

bit15:0	Description
Reading	Can read out the setting value of the register.
Writing 0	Uses a pin as a GPIO pin.
Writing 1	Uses a pin as an input/output pin of peripheral functions.

<Notes>

- The "x" of PFRx is a wildcard. PFRx indicates PFR0, PFR1, PFR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- Functions can be set for 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PFR0 sets P0F, the 14th bit of PFR0 sets POE, and the 0th bit of PFR0 sets P00.
- As a JTAG pin is selected for P04 to P00, the initial value is "1".
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.

4.2. Pull-up Setting Register (PCR_x)

The PCR_x register sets pull-up of a pin.

■ List of PCR Register Configuration

31	16	15	0	Initial value	Attribute	Support
Reserved		PCR0		0x001F	R/W	P0F to P00
Reserved		PCR1		0x0000	R/W	P1F to P10
Reserved		PCR2		0x0000	R/W	P2F to P20
Reserved		PCR3		0x0000	R/W	P3F to P30
Reserved		PCR4		0x0000	R/W	P4F to P40
Reserved		PCR5		0x0000	R/W	P5F to P50
Reserved		PCR6		0x0000	R/W	P6F to P60
Reserved		PCR7		0x0000	R/W	P7F to P70

■ Detailed Register Configuration

bit	31	16	15	0
Field	reserved			PCR _x

■ Register Function

[bit31:16] res : Register Bit

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PCR_x : Pull-up Setting Register x

Sets pull-up of a pin

bit15:0	Description
Reading	Can read out the setting value of the register.
Writing 0	Disconnects the pull-up resistor of a pin.
Writing 1	When a pin is in input status (for both GPIO and peripheral functions), the pull-up resistor is connected. When a pin is in output status, the pull-up resistor is disconnected.

<Notes>

- The "x" of PCR_x is a wildcard. PCR_x indicates PCR0, PCR1, PCR2, etc.
- The "x" of Px0 and Px_F is a wildcard. Px0 indicates P00, P10, P20, etc. Px_F indicates P0F, P1F, P2F, etc.
- One register allows setting 16 pull-ups from Px_F to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PCR0 sets P0F, the 14th bit of PCR0 sets P0E, and the 0th bit of PCR0 sets P00.
- As a JTAG pin is selected for P00 to P04, the initial value is "1".
- When using I²C function, use external pull-up by setting PCR_x=0.
- PCR8 is not available.
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.

4.3. Port input/output Direction Setting Register (DDRx)

The DDRx register sets input/output direction of a pin.

■ List of DDR Register Configuration

31	16	15	0	Initial value	Attribute	Support
reserved		DDR0	0	0x0000	R/W	P0F to P00
reserved		DDR1	1	0x0000	R/W	P1F to P10
reserved		DDR2	2	0x0000	R/W	P2F to P20
reserved		DDR3	3	0x0000	R/W	P3F to P30
reserved		DDR4	4	0x0000	R/W	P4F to P40
reserved		DDR5	5	0x0000	R/W	P5F to P50
reserved		DDR6	6	0x0000	R/W	P6F to P60
reserved		DDR7	7	0x0000	R/W	P7F to P70
reserved		DDR8	8	0x0000	R/W	P8F to P80

■ Detailed Register Configuration

bit	31	16	15	0
Field	reserved			DDRx

■ Register Function

[bit31:16] res : Reserved Bit

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] DDRx : Port input/output Direction Setting Register x

Sets input/output direction of a pin.

bit15:0	Description
Reading	Can read out the setting value of the register.
Writing 0	Uses GPIO in input direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.
Writing 1	Uses GPIO in output direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.

<Notes>

- The "x" of DDRx is a wildcard. DDRx indicates DDR0, DDR1, DDR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows setting the input/output direction of 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of DDR0 sets P0F, the 14th bit of DDR0 sets P0E, and the 0th bit of DDR0 sets P00.
- If the output RTO of a multifunction timer is selected, in an emergency stop due to DTTIX signal, a DDR controls pin status. For more information, see the chapter "MULTIFUNCTION TIMER".
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.

4.4. Port Input Data Register (PDIRx)

The PDIRx register indicates input data of a pin.

■ List of PDIR Register Configuration

31	16	15	0	Initial value	Attribute	Support
reserved		PDIR0	0x0000	R	P0F to P00	
reserved		PDIR1	0x0000	R	P1F to P10	
reserved		PDIR2	0x0000	R	P2F to P20	
reserved		PDIR3	0x0000	R	P3F to P30	
reserved		PDIR4	0x0000	R	P4F to P40	
reserved		PDIR5	0x0000	R	P5F to P50	
reserved		PDIR6	0x0000	R	P6F to P60	
reserved		PDIR7	0x0000	R	P7F to P70	
reserved		PDIR8	0x0000	R	P8F to P80	

■ Detailed Register Configuration

bit	31	16	15	0
Field	reserved			PDIRx

■ Register Function

[bit31:16] res : Reserved Bit

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PDIRx : Port Input Data Register x

Reads out input data of a pin.

bit15:0	Description
Reading 0	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "L" level input or "L" level output. When a special pin is selected by ADE/SPSR, as input is cut off, 0 is always read out.
Reading 1	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "H" level input or "H" level output.
Writing	Writing does not affect anything.

<Notes>

- The "x" of PDIRx is a wildcard. PDIRx indicates PDIR0, PDIR1, PDIR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows reading input data of 16 ports from PxF to Px0.
- Each bit in the register indicates the status of each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PDIR0 indicates P0F, the 14th bit of PDIR0 indicates P0E, and the 0th bit of PDIR0 indicates P00.
- "0" is always read for a bit value of the pin which is not available in your product.

4.5. Port Output Data Register x (PDORx)

The PDORx register sets output data to a pin.

■ List of PDOR Register Configuration

31	16	15	0	Initial value	Attribute	Support
reserved		PDOR0		0x0000	R/W	P0F to P00
reserved		PDOR1		0x0000	R/W	P1F to P10
reserved		PDOR2		0x0000	R/W	P2F to P20
reserved		PDOR3		0x0000	R/W	P3F to P30
reserved		PDOR4		0x0000	R/W	P4F to P40
reserved		PDOR5		0x0000	R/W	P5F to P50
reserved		PDOR6		0x0000	R/W	P6F to P60
reserved		PDOR7		0x0000	R/W	P7F to P70
reserved		PDOR8		0x0000	R/W	P8F to P80

■ Detailed Register Configuration

Bit	31	16	15	0
Field	reserved			PDORx

■ Register Function

[bit31:16] res : Reserved Bit

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PDORx : Port Output Data Register x

Sets output data of a pin.

bit15:0	Description
Reading	Reads out the register value.
Writing 0	Outputs "L" level to GPIO. If a pin is selected as GPIO input or peripheral functions input/output, a setting value is invalid.
Writing 1	Outputs "H" level to GPIO. If a pin is selected as GPIO input or peripheral functions input/output, a setting value is invalid.

<Notes>

- The "x" of PDORx is a wildcard. PDORx indicates PDOR0, PDOR1, PDOR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows setting output data of 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PDOR0 sets P0F, the 14th bit of PDOR0 sets P0E, and the 0th bit of PDOR0 sets P00.
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.

4.6. Analog Input Setting Register (ADE)

The ADE register sets a pin as an analog signal input pin of ADC.

■ Register Configuration

bit	31	16	15	0
Field	reserved		ADE	
Attribute	-		R/W	
Initial value	-		0xFFFF	

■ Register Function

[bit31:16] res : Reserved Bit

"0xFFFF" is read out from these bits.

When writing these bits, set "0xFFFF" to them.

[bit15:0] ADE : Analog Input Setting Register

Sets as an analog signal input pin.

bit15:0	Description
Reading	Reads out the register value.
Writing 0	Uses a pin not as an analog input but digital input/output.
Writing 1	Uses a pin as analog input. (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)

<Notes>

- This register sets ports from P1F to P10 as analog input pins.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of ADE sets P1F (AN15), the 14th bit of ADE sets P1E (AN14), and the 0th bit of ADE sets P10 (AN00).

4.7. Extended Pin Function Setting Register (EPFRx)

The EPFRx register assigns functions to a pin if there is more than one function.

■ List of EPFRx Register Configuration

31	0	Initial value	Attribute	Support
	EPFR00	0x00030000	R/W	System function
	EPFR01	0x00000000	R/W	Multi-function timer
	EPFR02	0x00000000	R/W	
	Reserved	-	-	-
	EPFR04	0x00000000	R/W	Base timer
	EPFR05	0x00000000	R/W	
	EPFR06	0x00000000	R/W	External interrupt
	EPFR07	0x00000000	R/W	Multi-function serial
	EPFR08	0x00000000	R/W	
	EPFR09	0x00000000	R/W	CAN/ADC trigger/QPRC
	EPFR10	0x00000000	R/W	External bus

4.8. Extended Pin Function Setting Register 00 (EPFR00)

The EPFR00 register assigns functions to a pin if there is more than one function.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field				Reserved			TRC1E	TRC0E
Attribute				-			R/W	R/W
Initial value							1'b0	1'b0
bit	23	22	21	20	19	18	17	16
Field				Reserved			JTAGEN1S	JTAGEN0B
Attribute				-			R/W	R/W
Initial value							1'b1	1'b1
bit	15	14	13	12	11	10	9	8
Field				Reserved			USB0PE	-
Attribute				-			R/W	-
Initial value							1'b0	-
bit	7	6	5	4	3	2	1	0
Field				Reserved			CROUTE	NMIS
Attribute				-			R/W	R/W
Initial value							1'b0	1'b0

■ Register Function

[bit31:26] res : Reserved Bit

"0b000000" is read out from these bits.

When writing these bits, set them to "0b000000".

[bit25] TRC1E : TRACED Function Select Bit 1

Selects a function for TRACED2 and TRACED3.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use two pins of TRACED2 and TRACED3. [Initial value] (A shared pin is available)
Writing 1	Uses two pins of TRACED2 and TRACED3.

[bit24] TRC0E : TRACED Function Select Bit 0

Selects a function for TRACECLK, TRACED0, and TRACED1 pins.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use three pins of TRACECLK, TRACED0, and TRACED1. [Initial value] (A shared pin is available)
Writing 1	Uses three pins of TRACECLK, TRACED0, and TRACED1.

[bit23:18] res : Reserved Bit

"0b000000" is read out from these bits.

When writing these bits, set them to "0b000000".

[bit17] JTAGEN1S : JTAG Function Select Bit 1
Selects a function for TRSTX and TDI.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use two pins of TRSTX and TDI. (A shared pin is available.)
Writing 1	Uses two pins of TRSTX and TDI. [Initial value]

[bit16] JTAGEN0B : JTAG Function Select Bit 0
Selects a function for TCK, TMS, and TDO pins.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use three pins of TCK, TMS, and TDO. (A shared pin is available.)
Writing 1	Uses three pins of TCK, TMS, and TDO. [Initial value]

[bit15:10] res : Reserved Bit

"0b0000000" is read out from these bits.

When writing these bits, set them to "0b0000000".

[bit9] USBP0E : USBch.0 Function Select Bit 1
Selects a function for USBch.0.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output D+ resistor control signal (HCONTX) for USBch.0. [Initial value] (A shared pin is available.)
Writing 1	Produces output D+ resistor control signal (HCONTX) for USBch.0.

[bit8:2] res : Reserved Bit

"0b00000000" is read out from these bits.

When writing these bits, set them to "0b00000000".

[bit1] CROUTE : Internal high-speed CR Oscillation Output Function Select Bit
Selects internal high-speed CR oscillation output.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce internal high-speed CR oscillation output. [Initial value]
Writing 1	Produces internal high-speed CR oscillation output.

[bit0] NMIS : NMIX Function Select Bit
Selects a function for the NMIX pin.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use the NMIX pin. [Initial value]
Writing 1	Uses the NMIX pin.

4.9. Extended Pin Function Setting Register 01 (EPFR01)

The EPFR01 register assigns functions to a pin of the multifunction timer Unit0.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field		IC03S			IC02S		IC01S	
Attribute		R/W			R/W		R/W	
Initial value		3'b000			3'b000		3'b000	
bit	23	22	21	20	19	18	17	16
Field	IC01S		IC00S		FRCK0S		DTTI0S	
Attribute		R/W			R/W		R/W	
Initial value		3'b000			2'b00		2'b00	
bit	15	14	13	12	11	10	9	8
Field		Reserved		DTTI0C		RTO05E		RTO04E
Attribute		-		R/W		R/W		R/W
Initial value		-		1'b0		2'b00		2'b00
bit	7	6	5	4	3	2	1	0
Field	RTO03E		RTO02E		RTO01E		RTO00E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

■ Register Function

[bit31:29] IC03S : IC03 Input Select Bit

Selects input for IC03.

bit31:29	Description
Reading	Reads out the register value.
Writing 000	Uses IC03_0 at the input pin of the input capture IC03. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC03_1 at the input pin of the input capture IC03.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.3LSYN for input of the input capture IC03.
Writing 101	Uses internal macro MFSch.7LSYN for input of the input capture IC03.
Writing 110	Setting is prohibited.
Writing 111	Uses the internal macro pin CRTRIM for input of the input capture IC03.

[bit28:26] IS02S : IC02 Input Select Bit
Selects input for IC02.

bit28:26	Description
Reading	Reads out the register value.
Writing 000	Uses IC02_0 at the input pin of the input capture IC02. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC02_1 at the input pin of the input capture IC02.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.2LSYN for input of the input capture IC02.
Writing 101	Uses internal macro MFSch.6LSYN for input of the input capture IC02.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit25:23] IC01S : IC01 Input Select Bit
Selects input for IC01.

bit25:23	Description
Reading	Reads out the register value.
Writing 000	Uses IC01_0 at the input pin of the input capture IC01. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC01_1 at the input pin of the input capture IC01.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.1LSYN for input of the input capture IC01.
Writing 101	Uses internal macro MFSch.5LSYN for input of the input capture IC01.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit22:20] IC00S : IC00 Input Select Bit
Selects input for IC00.

bit22:20	Description
Reading	Reads out the register value.
Writing 000	Uses IC00_0 at the input pin of the input capture IC00. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC00_1 at the input pin of the input capture IC00.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.0LSYN for input of the input capture IC00.
Writing 101	Uses internal macro MFSch.4LSYN for input of the input capture IC00.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit19:18] FRCK0S : FRCK0 Input Select Bit
Selects input for FRCK0.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Uses FRCK0_0 at the input pin of the free-run timer FRCK0. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses FRCK0_1 at the input pin of the free-run timer FRCK0.
Writing 11	Setting is prohibited.

[bit17:16] DTTI0S : DTTIX0 Input Select Bit
Selects input for DTTIX0.

bit17:16	Description
Reading	Reads out the register value.
Writing 00	Uses DTTIX0_0 at the input pin of the waveform generator DTTIX0. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses DTTIX0_1 at the input pin of the waveform generator DTTIX0.
Writing 11	Setting is prohibited.

[bit15:13] res : Reserved Bit
"0b000" is read out from these bits.
When writing these bits, set them to "0b000".

[bit12] DTTI0C : DTTIX0 Function Select Bit
Selects a function for DTTIX0.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not switch GPIO by DTTIF0 for output of pins RTO00 to RTO05. [Initial value]
Writing 1	Switches GPIO by DTTIF0 for output of pins RTO00 to RTO05.

[bit11:10] RTO05E : RTO05E Output Select Bit
Selects output for RTO05.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO05. [Initial value]
Writing 01	Uses RTO05_0 at the output pin of the waveform generator RTO05.
Writing 10	Uses RTO05_1 at the output pin of the waveform generator RTO05.
Writing 11	Setting is prohibited.

[bit9:8] RTO04E : RTO04E Output Select Bit
Selects output for RTO04.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO04. [Initial value]
Writing 01	Uses RTO04_0 at the output pin of the waveform generator RTO04.
Writing 10	Uses RTO04_1 at the output pin of the waveform generator RTO04.
Writing 11	Setting is prohibited.

[bit7:6] RTO03E : RTO03E Output Select Bit
Selects output for RTO03.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO03. [Initial value]
Writing 01	Uses RTO03_0 at the output pin of the waveform generator RTO03.
Writing 10	Uses RTO03_1 at the output pin of the waveform generator RTO03.
Writing 11	Setting is prohibited.

[bit5:4] RTO02E : RTO02E Output Select Bit
Selects output for RTO02.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO02. [Initial value]
Writing 01	Uses RTO02_0 at the output pin of the waveform generator RTO02.
Writing 10	Uses RTO02_1 at the output pin of the waveform generator RTO02.
Writing 11	Setting is prohibited.

[bit3:2] RTO01E : RTO01E Output Select Bit
Selects output for RTO01.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO01. [Initial value]
Writing 01	Uses RTO01_0 at the output pin of the waveform generator RTO01.
Writing 10	Uses RTO01_1 at the output pin of the waveform generator RTO01.
Writing 11	Setting is prohibited.

[bit1:0] RTO00E : RTO00E Output Select Bit
Selects output for RTO00.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO00. [Initial value]
Writing 01	Uses RTO00_0 at the output pin of the waveform generator RTO00.
Writing 10	Uses RTO00_1 at the output pin of the waveform generator RTO00.
Writing 11	Setting is prohibited.

4.10. Extended Pin Function Setting Register 02 (EPFR02)

The EPFR02 register assigns functions to a pin of the multifunction timer Unit1.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field		IC13S			IC12S		IC11S	
Attribute		R/W			R/W		R/W	
Initial value		3'b000			3'b000		3'b000	
bit	23	22	21	20	19	18	17	16
Field	IC11S		IC10S		FRCK1S		DTTI1S	
Attribute		R/W			R/W		R/W	
Initial value		3'b000			2'b00		2'b00	
bit	15	14	13	12	11	10	9	8
Field		reserved		DTTI1C		RTO15E		RTO14E
Attribute		-		R/W		R/W		R/W
Initial value		-		1'b0		2'b00		2'b00
Bit	7	6	5	4	3	2	1	0
Field	RTO13		RTO12E		RTO11E		RTO10E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

■ Register Function

[bit31:29] IC13S : IC13 Input Select Bit

Selects input for IC13.

bit31:29	Description
Reading	Reads out the register value.
Writing 000	Uses IC13_0 at the input pin of the input capture IC13. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC13_1 at the input pin of the input capture IC13.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.3LSYN for input of the input capture IC13.
Writing 101	Uses internal macro MFSch.7LSYN for input of the input capture IC13.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit28:26] IC12S : IC12 Input Select Bit
Selects input for IC12.

bit28:26	Description
Reading	Reads out the register value.
Writing 000	Uses IC12_0 at the input pin of the input capture IC12. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC12_1 at the input pin of the input capture IC12.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.2LSYN for input of the input capture IC12.
Writing 101	Uses internal macro MFSch.6LSYN for input of the input capture IC12.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit25:23] IC11S : IC11 Input Select Bit
Selects input for IC11.

bit25:23	Description
Reading	Reads out the register value.
Writing 000	Uses IC11_0 at the input pin of the input capture IC11. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC11_1 at the input pin of the input capture IC11.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.1LSYN for input of the input capture IC11.
Writing 101	Uses internal macro MFSch.5LSYN for input of the input capture IC11.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit22:20] IC10S : IC10 Input Select Bit
Selects input for IC10.

bit22:20	Description
Reading	Reads out the register value.
Writing 000	Uses IC10_0 at the input pin of the input capture IC10. [Initial value]
Writing 001	Same as Writing 000.
Writing 010	Uses IC10_1 at the input pin of the input capture IC10.
Writing 011	Setting is prohibited.
Writing 100	Uses internal macro MFSch.0LSYN for input of the input capture IC10.
Writing 101	Uses internal macro MFSch.4LSYN for input of the input capture IC10.
Writing 110	Setting is prohibited.
Writing 111	Setting is prohibited.

[bit19:18] FRCK1S : FRCK1 Input Select Bit
Selects input for FRCK1.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Uses FRCK1_0 at the input pin of the free-run timer FRCK1. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses FRCK1_1 at the input pin of the free-run timer FRCK1.
Writing 11	Setting is prohibited.

[bit17:16] DTTI1S : DTTIX1 Input Select Bit
Select input for DTTIX1.

bit17:16	Description
Reading	Reads out the register value.
Writing 00	Uses DTTIX1_0 at the input pin of the waveform generator DTTIX1. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses DTTIX1_1 at the input pin of the waveform generator DTTIX1.
Writing 11	Setting is prohibited.

[bit15:13] res : Reserved Bit
"0b000" is read out from these bits.
When writing these bits, set them to "0b000".

[bit12] DTTI1C : DTTIX1 Function Select Bit
Selects a function for DTTIX1.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not switch GPIO by DTTIF1 for output of pins RTO10 to RTO15. [Initial value]
Writing 1	Switches GPIO by DTTIF1 for output of pins RTO10 to RTO15.

[bit11:10] RTO15E : RTO15E Output Select Bit
Selects output for RTO15.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO15. [Initial value]
Writing 01	Uses RTO15_0 at the output pin of the waveform generator RTO15.
Writing 10	Uses RTO15_1 at the output pin of the waveform generator RTO15.
Writing 11	Setting is prohibited.

[bit9:8] RTO14E : RTO14E Output Select Bit

Selects output for RTO14.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO14. [Initial value]
Writing 01	Uses RTO14_0 at the output pin of the waveform generator RTO14.
Writing 10	Uses RTO14_1 at the output pin of the waveform generator RTO14.
Writing 11	Setting is prohibited.

[bit7:6] RTO13E : RTO13E Output Select Bit

Selects output for RTO13.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO13. [Initial value]
Writing 01	Uses RTO13_0 at the output pin of the waveform generator RTO13.
Writing 10	Uses RTO13_1 at the output pin of the waveform generator RTO13.
Writing 11	Setting is prohibited.

[bit5:4] RTO12E : RTO12E Output Select Bit

Selects output for RTO12.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO12. [Initial value]
Writing 01	Uses RTO12_0 at the output pin of the waveform generator RTO12.
Writing 10	Uses RTO12_1 at the output pin of the waveform generator RTO12.
Writing 11	Setting is prohibited.

[bit3:2] RTO11E : RTO11E Output Select Bit

Selects output for RTO11.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO11. [Initial value]
Writing 01	Uses RTO11_0 at the output pin of the waveform generator RTO11.
Writing 10	Uses RTO11_1 at the output pin of the waveform generator RTO11.
Writing 11	Setting is prohibited.

[bit1:0] RTO10E : RTO10E Output Select Bit

Selects output for RTO10.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for the waveform generator RTO10. [Initial value]
Writing 01	Uses RTO10_0 at the output pin of the waveform generator RTO10.
Writing 10	Uses RTO10_1 at the output pin of the waveform generator RTO10.
Writing 11	Setting is prohibited.

4.11. Extended Pin Function Setting Register 04 (EPFR04)

The EPFR04 register assigns functions to pins of ch.0, ch.1, ch.2 and ch.3 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	reserved		TIOB3S		TIOA3E		TIOA3S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		2'b00		2'b00		2'b00	
bit	23	22	21	20	19	18	17	16
Field	reserved		TIOB2S		TIOA2E		reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		2'b00		2'b00		-	
bit	15	14	13	12	11	10	9	8
Field	reserved		TIOB1S		TIOA1E		TIOA1S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		2'b00		2'b00		2'b00	
bit	7	6	5	4	3	2	1	0
Field	reserved		TIOB0S		TIOA0E		reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		2'b00		2'b00		-	

■ Register Function

[bit31:30] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB3S : TIOB3 Input Select Bit

Selects input for TIOB3.

bit29:28	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB3_0 at the input pin of BT-ch3-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB3_1 at the input pin of BT-ch3-TIOB.
Writing 11	Uses TIOB3_2 at the input pin of BT-ch3-TIOB.

[bit27:26] TIOA3E : TIOA3E Output Select Bit

Selects output for TIOA3.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch3-TIOA. [Initial value]
Writing 01	Uses TIOA3_0 at the output pin of BT-ch3-TIOA.
Writing 10	Uses TIOA3_1 at the output pin of BT-ch3-TIOA.
Writing 11	Uses TIOA3_2 at the output pin of BT-ch3-TIOA.

[bit25:24] TIOA3S : TIOA3 Input Select Bit
Selects input for TIOA3.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Uses TIOA3_0 at the input pin of BT-ch3-TIOA. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOA3_1 at the input pin of BT-ch3-TIOA.
Writing 11	Uses TIOA3_2 at the input pin of BT-ch3-TIOA.

[bit23:22] res : Reserved Bit
"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB2S : TIOB2 Input Select Bit
Selects input for TIOB2.

bit21:20	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB2_0 at the input pin of BT-ch2-TIOB. [Initial value]
Writing 01	Same as Writing 00
Writing 10	Uses TIOB2_1 at the input pin of BT-ch2-TIOB.
Writing 11	Uses TIOB2_2 at the input pin of BT-ch2-TIOB.

[bit19:18] TIOA2E : TIOA2 Output Select Bit
Selects output for TIOA2.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch2-TIOA. [Initial value]
Writing 01	Uses TIOA2_0 at the output pin of BT-ch2-TIOA.
Writing 10	Uses TIOA2_1 at the output pin of BT-ch2-TIOA.
Writing 11	Uses TIOA2_2 at the output pin of BT-ch2-TIOA.

[bit17:14] res : Reserved Bit
"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit13:12] TIOB1S : TIOB1 Input Select Bit
Selects input for TIOB1.

bit13:12	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB1_0 at the input pin of BT-ch1-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB1_1 at the input pin of BT-ch1-TIOB.
Writing 11	Uses TIOB1_2 at the input pin of BT-ch1-TIOB.

[bit11:10] TIOA1E : TIOA1E Output Select Bit

Selects output for TIOA1.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch1-TIOA. [Initial value]
Writing 01	Uses TIOA1_0 at the output pin of BT-ch1-TIOA.
Writing 10	Uses TIOA1_1 at the output pin of BT-ch1-TIOA.
Writing 11	Uses TIOA1_2 at the output pin of BT-ch1-TIOA.

[bit9:8] TIOA1S : TIOA1 Input Select Bit

Selects input for TIOA1.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses TIOA1_0 at the input pin of BT-ch1-TIOA. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOA1_1 at the input pin of BT-ch1-TIOA.
Writing 11	Uses TIOA1_2 at the input pin of BT-ch1-TIOA.

[bit7:6] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit5:4] TIOB0S : TIOB0 Input Select Bit

Selects input for TIOB0.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB0_0 at the input pin of BT-ch0-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB0_1 at the input pin of BT-ch0-TIOB.
Writing 11	Uses TIOB0_2 at the input pin of BT-ch0-TIOB.

[bit3:2] TIOA0E : TIOA0 Output Select Bit

Selects output for TIOA0.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch0-TIOA. [Initial value]
Writing 01	Uses TIOA0_0 at the output pin of BT-ch0-TIOA.
Writing 10	Uses TIOA0_1 at the output pin of BT-ch0-TIOA.
Writing 11	Uses TIOA0_2 at the output pin of BT-ch0-TIOA.

[bit1:0] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

<Notes>

- TIOA
Even channels are for output only.
Odd channels are for both input and output.
- TIOB
Input only.
- TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them. When output is chosen for odd TIOA channel, input setting will be ignored.

Example1 : Use TIOA1 as an output pin:

When TIOA1 is output to TIOA1_0, select EPFR04.TIOA1E = 01.
When TIOA1 is output to TIOA1_1, select EPFR04.TIOA1E = 10.
When TIOA1 is output to TIOA1_2, select EPFR04.TIOA1E = 11.
Settings for EPFR04.TIOA1S will be ignored.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

Example2 : When TIOA1 is used as an input pin:

Select EPFR04.TIOA1E = 00.
When TIOA1 is input from TIOA1_0, select EPFR04.TIOA1S = 00 or 01.
When TIOA1 is input from TIOA1_1, select EPFR04.TIOA1S = 10.
When TIOA1 is input from TIOA1_2, select EPFR04.TIOA1S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

* When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

4.12. Extended Pin Function Setting Register 05 (EPFR05)

The EPFR05 register assigns functions to pins of ch.4, ch.5, ch.6, and ch.7 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	reserved		TIOB7S		TIOA7E		TIOA7S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		2'b00		2'b00		2'b00	

bit	23	22	21	20	19	18	17	16
Field	reserved		TIOB6S		TIOA6E		reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		2'b00		2'b00		-	

bit	15	14	13	12	11	10	9	8
Field	reserved		TIOB5S		TIOA5E		TIOA5S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		2'b00		2'b00		2'b00	

bit	7	6	5	4	3	2	1	0
Field	reserved		TIOB4S		TIOA4E		reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		2'b00		2'b00		-	

■ Register Function

[bit31:30] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB7S : TIOB7 Input Select Bit

Selects input for TIOB7.

bit29:28	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB7_0 at the input pin of BT-ch7-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB7_1 at the input pin of BT-ch7-TIOB.
Writing 11	Uses TIOB7_2 at the input pin of BT-ch7-TIOB.

[bit27:26] TIOA7E : TIOA7E Output Select Bit

Selects output for TIOA7.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch7-TIOA. [Initial value]
Writing 01	Uses TIOA7_0 at the output pin of BT-ch7-TIOA.
Writing 10	Uses TIOA7_1 at the output pin of BT-ch7-TIOA.
Writing 11	Uses TIOA7_2 at the output pin of BT-ch7-TIOA.

[bit25:24] TIOA7S : TIOA7 Input Select Bit
Selects input for TIOA7.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Uses TIOA7_0 at the input pin of BT-ch7-TIOA. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOA7_1 at the input pin of BT-ch7-TIOA.
Writing 11	Uses TIOA7_2 at the input pin of BT-ch7-TIOA.

[bit23:22] res : Reserved Bit
"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB6S : TIOB6 Input Select Bit
Selects input for TIOB6.

bit21:20	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB6_0 at the input pin of BT-ch6-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB6_1 at the input pin of BT-ch6-TIOB.
Writing 11	Uses TIOB6_2 at the input pin of BT-ch6-TIOB.

[bit19:18] TIOA6E : TIOA6 Output Select Bit
Selects output for TIOA6.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch6-TIOA. [Initial value]
Writing 01	Uses TIOA6_0 at the output pin of BT-ch6-TIOA.
Writing 10	Uses TIOA6_1 at the output pin of BT-ch6-TIOA.
Writing 11	Uses TIOA6_2 at the output pin of BT-ch6-TIOA.

[bit17:14] res : Reserved Bit
"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit13:12] TIOB5S : TIOB5 Input Select Bit
Selects input for TIOB5.

bit13:12	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB5_0 at the input pin of BT-ch5-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB5_1 at the input pin of BT-ch5-TIOB.
Writing 11	Uses TIOB5_2 at the input pin of BT-ch5-TIOB.

[bit11:10] TIOA5E : TIOA5E Output Select Bit

Selects output for TIOA5.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for BT-ch5-TIOA. [Initial value]
Writing 01	Uses TIOA5_0 at the output pin of BT-ch5-TIOA.
Writing 10	Uses TIOA5_1 at the output pin of BT-ch5-TIOA.
Writing 11	Uses TIOA5_2 at the output pin of BT-ch5-TIOA.

[bit9:8] TIOA5S : TIOA5 Input Select Bit

Selects input for TIOA5.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses TIOA5_0 at the input pin of BT-ch5-TIOA. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOA5_1 at the input pin of BT-ch5-TIOA.
Writing 11	Uses TIOA5_2 at the input pin of BT-ch5-TIOA.

[bit7:6] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit5:4] TIOB4S : TIOB4 Input Select Bit

Selects input for TIOB4.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses TIOB4_0 at the input pin of BT-ch4-TIOB. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses TIOB4_1 at the input pin of BT-ch4-TIOB.
Writing 11	Uses TIOB4_2 at the input pin of BT-ch4-TIOB.

[bit3:2] TIOA4E : TIOA4 Output Select Bit

Selects output for TIOA4.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Produces output for BT-ch4-TIOA. [Initial value]
Writing 01	Uses TIOA4_0 at the output pin of BT-ch4-TIOA.
Writing 10	Uses TIOA4_1 at the output pin of BT-ch4-TIOA.
Writing 11	Uses TIOA4_2 at the output pin of BT-ch4-TIOA.

[bit1:0] res : Reserved Bit

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

<Notes>

- TIOA
Even channels are for output only.
Odd channels are for both input and output.
- TIOB
Input only.
- TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them. When output is chosen for odd TIOA channel, input setting will be ignored.

Example1 : Use TIOA1 as an output pin:

When TIOA1 is output to TIOA1_0, select EPFR04.TIOA1E = 01.
When TIOA1 is output to TIOA1_1, select EPFR04.TIOA1E = 10.
When TIOA1 is output to TIOA1_2, select EPFR04.TIOA1E = 11.
Settings for EPFR04.TIOA1S will be ignored.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

Example2 : When TIOA1 is used as an input pin:

Select EPFR04.TIOA1E = 00.
When TIOA1 is input from TIOA1_0, select EPFR04.TIOA1S = 00 or 01.
When TIOA1 is input from TIOA1_1, select EPFR04.TIOA1S = 10.
When TIOA1 is input from TIOA1_2, select EPFR04.TIOA1S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

* When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

4.13. Extended Pin Function Setting Register 06 (EPFR06)

The EPFR06 register assigns functions to external interrupt pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	EINT15S		EINT14S		EINT13S		EINT12S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

bit	23	22	21	20	19	18	17	16
Field	EINT11S		EINT10S		EINT09S		EINT08S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

bit	15	14	13	12	11	10	9	8
Field	EINT07S		EINT06S		EINT05S		EINT04S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

bit	7	6	5	4	3	2	1	0
Field	EINT03S		EINT02S		EINT01S		EINT00S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

■ Register Function

[bit31:30] EINT15S : External Interrupt Input Select Bit

Selects input for EINT15.

bit31:30	Description
Reading	Reads out the register value.
Writing 00	Uses INT15_0 at the input pin of EINT-ch15. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT15_1 at the input pin of EINT-ch15.
Writing 11	Uses INT15_2 at the input pin of EINT-ch15.

[bit29:28] EINT14S : External Interrupt Input Select Bit

Selects input for EINT14.

bit29:28	Description
Reading	Reads out the register value.
Writing 00	Uses INT14_0 at the input pin of EINT-ch14. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT14_1 at the input pin of EINT-ch14.
Writing 11	Uses INT14_2 at the input pin of EINT-ch14.

[bit27:26] EINT13S : External Interrupt Input Select Bit
Selects input for EINT13.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Uses INT13_0 at the input pin of EINT-ch13. [Initial value]
Writing 01	Same as Writing 00
Writing 10	Uses INT13_1 at the input pin of EINT-ch13.
Writing 11	Uses INT13_2 at the input pin of EINT-ch13.

[bit25:24] EINT12S : External Interrupt Input Select Bit
Selects input for EINT12.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Uses INT12_0 at the input pin of EINT-ch12. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT12_1 at the input pin of EINT-ch12.
Writing 11	Uses INT12_2 at the input pin of EINT-ch12.

[bit23:22] EINT11S : External Interrupt Input Select Bit
Selects input for EINT11.

bit23:22	Description
Reading	Reads out the register value.
Writing 00	Uses INT11_0 at the input pin of EINT-ch11. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT11_1 at the input pin of EINT-ch11.
Writing 11	Uses INT11_2 at the input pin of EINT-ch11.

[bit21:20] EINT10S : External Interrupt Input Select Bit
Selects input for EINT10.

bit21:20	Description
Reading	Reads out the register value.
Writing 00	Uses INT10_0 at the input pin of EINT-ch10. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT10_1 at the input pin of EINT-ch10.
Writing 11	Uses INT10_2 at the input pin of EINT-ch10.

[bit19:18] EINT09S : External Interrupt Input Select Bit
Selects input for EINT09.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Uses INT09_0 at the input pin of EINT-ch9. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT09_1 at the input pin of EINT-ch9.
Writing 11	Uses INT09_2 at the input pin of EINT-ch9.

[bit17:16] EINT08S : External Interrupt Input Select Bit

Selects input for EINT08.

bit17:16	Description
Reading	Reads out the register value.
Writing 00	Uses INT08_0 at the input pin of EINT-ch8. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT08_1 at the input pin of EINT-ch8.
Writing 11	Uses INT08_2 at the input pin of EINT-ch8.

[bit15:14] EINT07S : External Interrupt Input Select Bit

Selects input for EINT07.

bit15:14	Description
Reading	Reads out the register value.
Writing 00	Uses INT07_0 at the input pin of EINT-ch7. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT07_1 at the input pin of EINT-ch7.
Writing 11	Uses INT07_2 at the input pin of EINT-ch7.

[bit13:12] EINT06S : External Interrupt Input Select Bit

Selects input for EINT06.

bit13:12	Description
Reading	Reads out the register value.
Writing 00	Uses INT06_0 at the input pin of EINT-ch6. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT06_1 at the input pin of EINT-ch6.
Writing 11	Uses INT06_2 at the input pin of EINT-ch6.

[bit11:10] EINT05S : External Interrupt Input Select Bit

Selects input for EINT05.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Uses INT05_0 at the input pin of EINT-ch5. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT05_1 at the input pin of EINT-ch5.
Writing 11	Uses INT05_2 at the input pin of EINT-ch5.

[bit9:8] EINT04S : External Interrupt Input Select Bit

Selects input for EINT04.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses INT04_0 at the input pin of EINT-ch4. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT04_1 at the input pin of EINT-ch4.
Writing 11	Uses INT04_2 at the input pin of EINT-ch4.

[bit7:6] EINT03S : External Interrupt Input Select Bit

Selects input for EINT03.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Uses INT03_0 at the input pin of EINT-ch3. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT03_1 at the input pin of EINT-ch3.
Writing 11	Uses INT03_2 at the input pin of EINT-ch3.

[bit5:4] EINT02S : External Interrupt Input Select Bit

Selects input for EINT02.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses INT02_0 at the input pin of EINT-ch2. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT02_1 at the input pin of EINT-ch2.
Writing 11	Uses INT02_2 at the input pin of EINT-ch2.

[bit3:2] EINT01S : External Interrupt Input Select Bit

Selects input for EINT01.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Uses INT01_0 at the input pin of EINT-ch1. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT01_1 at the input pin of EINT-ch1.
Writing 11	Uses INT01_2 at the input pin of EINT-ch1.

[bit1:0] EINT00S : External Interrupt Input Select Bit

Selects input for EINT00.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	Uses INT00_0 at the input pin of EINT-ch0. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses INT00_1 at the input pin of EINT-ch0.
Writing 11	Uses INT00_2 at the input pin of EINT-ch0.

4.14. Extended Pin Function Setting Register 07 (EPFR07)

The EPFR07 register assigns functions of multi-function serial channel 0, channel 1, and channel3.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field		reserved			SCK3B		SOT3B	
Attribute		-			R/W		R/W	
Initial value		-			2'b00		2'b00	
bit	23	22	21	20	19	18	17	16
Field	SIN3S		SCK2B		SOT2B		SIN2S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	
bit	15	14	13	12	11	10	9	8
Field	SCK1B		SOT1B		SIN1S		SCK0B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	
bit	7	6	5	4	3	2	1	0
Field	SOT0B		SIN0S		reserved		-	-
Attribute	R/W		R/W		-		-	-
Initial value	2'b00		2'b00		-		-	-

■ Register Function

[bit31:28] res : Reserved Bit

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK3B : SCK3 Input/Output Select Bit

Selects input/output for SCK3.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Uses SCK3_0 at the input pin of MFS-ch3-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK3_0 at the input pin of MFS-ch3-SCK. Uses SCK3_0 at the output pin.
Writing 10	Uses SCK3_1 at the input pin of MFS-ch3-SCK. Uses SCK3_1 at the output pin.
Writing 11	Uses SCK3_2 at the input pin of MFS-ch3-SCK. Uses SCK3_2 at the output pin.

[bit25:24] SOT3B : SOT3B Input/Output Select Bit

Selects input/output for SOT3.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Uses SOT3_0 at the input pin of MFS-ch3-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT3_0 at the input pin of MFS-ch3-SOT. Uses SOT3_0 at the output pin.
Writing 10	Uses SOT3_1 at the input pin of MFS-ch3-SOT. Uses SOT3_1 at the output pin.
Writing 11	Uses SOT3_2 at the input pin of MFS-ch3-SOT. Uses SOT3_2 at the output pin.

[bit23:22] SIN3S : SIN3S Input Select Bit

Selects input for SIN3.

bit23:22	Description
Reading	Reads out the register value.
Writing 00	Uses SIN3_0 at the input pin of MFS-ch3-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN3_1 at the input pin of MFS-ch3-SIN.
Writing 11	Uses SIN3_2 at the input pin of MFS-ch3-SIN.

[bit21:20] SCK2B : SCK2 Input/Output Select Bit

Selects input/output for SCK2.

bit21:20	Description
Reading	Reads out the register value.
Writing 00	Uses SCK2_0 at the input pin of MFS-ch2-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK2_0 at the input pin of MFS-ch2-SCK. Uses SCK2_0 at the output pin.
Writing 10	Uses SCK2_1 at the input pin of MFS-ch2-SCK. Uses SCK2_1 at the output pin.
Writing 11	Uses SCK2_2 at the input pin of MFS-ch2-SCK. Uses SCK2_2 at the output pin.

[bit19:18] SOT2B : SOT2B Input/Output Select Bit

Selects input/output for SOT2.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Uses SOT2_0 at the input pin of MFS-ch2-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT2_0 at the input pin of MFS-ch2-SOT. Uses SOT2_0 at the output pin.
Writing 10	Uses SOT2_1 at the input pin of MFS-ch2-SOT. Uses SOT2_1 at the output pin.
Writing 11	Uses SOT2_2 at the input pin of MFS-ch2-SOT. Uses SOT2_2 at the output pin.

[bit17:16] SIN2S : SIN2S Input Select Bit

Selects input for SIN2.

bit17:16	Description
Reading	Reads out the register value.
Writing 00	Uses SIN2_0 at the input pin of MFS-ch2-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN2_1 at the input pin of MFS-ch2-SIN.
Writing 11	Uses SIN2_2 at the input pin of MFS-ch2-SIN.

[bit15:14] SCK1B : SCK1 Input/Output Select Bit

Selects input/output for SCK1.

bit15:14	Description
Reading	Reads out the register value.
Writing 00	Uses SCK1_0 at the input pin of MFS-ch1-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK1_0 at the input pin of MFS-ch1-SCK. Uses SCK1_0 at the output pin.
Writing 10	Uses SCK1_1 at the input pin of MFS-ch1-SCK. Uses SCK1_1 at the output pin.
Writing 11	Uses SCK1_2 at the input pin of MFS-ch1-SCK. Uses SCK1_2 at the output pin.

[bit13:12] SOT1B : SOT1B Input/Output Select Bit

Selects input/output for SOT1.

bit13:12	Description
Reading	Reads out the register value.
Writing 00	Uses SOT1_0 at the input pin of MFS-ch1-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT1_0 at the input pin of MFS-ch1-SOT. Uses SOT1_0 at the output pin.
Writing 10	Uses SOT1_1 at the input pin of MFS-ch1-SOT. Uses SOT1_1 at the output pin.
Writing 11	Uses SOT1_2 at the input pin of MFS-ch1-SOT. Uses SOT1_2 at the output pin.

[bit11:10] SIN1S : SIN1S Input Select Bit

Selects input for SIN1.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Uses SIN1_0 at the input pin of MFS-ch1-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN1_1 at the input pin of MFS-ch1-SIN.
Writing 11	Uses SIN1_2 at the input pin of MFS-ch1-SIN.

[bit9:8] SCK0B : SCK0 Input/Output Select Bit

Selects input/output for SCK0.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses SCK0_0 at the input pin of MFS-ch0-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK0_0 at the input pin of MFS-ch0-SCK. Uses SCK0_0 at the output pin.
Writing 10	Uses SCK0_1 at the input pin of MFS-ch0-SCK. Uses SCK0_1 at the output pin.
Writing 11	Uses SCK0_2 at the input pin of MFS-ch0-SCK. Uses SCK0_2 at the output pin.

[bit7:6] SOT0B : SOT0B Input/Output Select Bit

Selects input/output for SOT0.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Uses SOT0_0 at the input pin of MFS-ch0-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT0_0 at the input pin of MFS-ch0-SOT. Uses SOT0_0 at the output pin.
Writing 10	Uses SOT0_1 at the input pin of MFS-ch0-SOT. Uses SOT0_1 at the output pin.
Writing 11	Uses SOT0_2 at the input pin of MFS-ch0-SOT. Uses SOT0_2 at the output pin.

[bit5:4] SIN0S : SIN0S Input Select Bit

Selects input for SIN0.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses SIN0_0 at the input pin of MFS-ch0-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN0_1 at the input pin of MFS-ch0-SIN.
Writing 11	Uses SIN0_2 at the input pin of MFS-ch0-SIN.

[bit3:0] res : Reserved Bit

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

4.15. Extended Pin Function Setting Register 08 (EPFR08)

The EPFR08 register assigns functions of multi-function serial channel 4, channel 5, channel 6, and channel 7.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field		reserved			SCK7B		SOT7B	
Attribute		-			R/W		R/W	
Initial value		-			2'b00		2'b00	
bit	23	22	21	20	19	18	17	16
Field	SIN7S		SCK6B		SOT6B		SIN6S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	
bit	15	14	13	12	11	10	9	8
Field	SCK5B		SOT5B		SIN5S		SCK4B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	
bit	7	6	5	4	3	2	1	0
Field	SOT4B		SIN4S		CTS4S		RTS4E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

■ Register Function

[bit31:28] res : Reserved Bit

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK7B : SCK7 Input/Output Select Bit

Selects input/output for SCK7.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Uses SCK7_0 at the input pin of MFS-ch7-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK7_0 at the input pin of MFS-ch7-SCK. Uses SCK7_0 at the output pin.
Writing 10	Uses SCK7_1 at the input pin of MFS-ch7-SCK. Uses SCK7_1 at the output pin.
Writing 11	Uses SCK7_2 at the input pin of MFS-ch7-SCK. Uses SCK7_2 at the output pin.

[bit25:24] SOT7B : SOT7B Input/Output Select Bit

Selects input/output for SOT7.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Uses SOT7_0 at the input pin of MFS-ch7-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT7_0 at the input pin of MFS-ch7-SOT. Uses SOT7_0 at the output pin.
Writing 10	Uses SOT7_1 at the input pin of MFS-ch7-SOT. Uses SOT7_1 at the output pin.
Writing 11	Uses SOT7_2 at the input pin of MFS-ch7-SOT. Uses SOT7_2 at the output pin.

[bit23:22] SIN7S : SIN7S Input Select Bit

Selects input for SIN7.

bit23:22	Description
Reading	Reads out the register value.
Writing 00	Uses SIN7_0 at the input pin of MFS-ch7-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN7_1 at the input pin of MFS-ch7-SIN.
Writing 11	Uses SIN7_2 at the input pin of MFS-ch7-SIN.

[bit21:20] SCK6B : SCK6 Input/Output Select Bit

Selects input/output for SCK6.

bit21:20	Description
Reading	Reads out the register value.
Writing 00	Uses SCK6_0 at the input pin of MFS-ch6-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK6_0 at the input pin of MFS-ch6-SCK. Uses SCK6_0 at the output pin.
Writing 10	Uses SCK6_1 at the input pin of MFS-ch6-SCK. Uses SCK6_1 at the output pin.
Writing 11	Uses SCK6_2 at the input pin of MFS-ch6-SCK. Uses SCK6_2 at the output pin.

[bit19:18] SOT6B : SOT6B Input/Output Select Bit

Selects input/output for SOT6.

bit19:18	Description
Reading	Reads out the register value.
Writing 00	Uses SOT6_0 at the input pin of MFS-ch6-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT6_0 at the input pin of MFS-ch6-SOT. Uses SOT6_0 at the output pin.
Writing 10	Uses SOT6_1 at the input pin of MFS-ch6-SOT. Uses SOT6_1 at the output pin.
Writing 11	Uses SOT6_2 at the input pin of MFS-ch6-SOT. Uses SOT6_2 at the output pin.

[bit17:16] SIN6S : SIN6S Input Select Bit

Selects input for SIN6.

bit17:16	Description
Reading	Reads out the register value.
Writing 00	Uses SIN6_0 at the input pin of MFS-ch6-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN6_1 at the input pin of MFS-ch6-SIN.
Writing 11	Uses SIN6_2 at the input pin of MFS-ch6-SIN.

[bit15:14] SCK5B : SCK5 Input/Output Select Bit

Selects input/output for SCK5.

bit15:14	Description
Reading	Reads out the register value.
Writing 00	Uses SCK5_0 at the input pin of MFS-ch5-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK5_0 at the input pin of MFS-ch5-SCK. Uses SCK5_0 at the output pin.
Writing 10	Uses SCK5_1 at the input pin of MFS-ch5-SCK. Uses SCK5_1 at the output pin.
Writing 11	Uses SCK5_2 at the input pin of MFS-ch5-SCK. Uses SCK5_2 at the output pin.

[bit13:12] SOT5B : SOT5B Input/Output Select Bit

Selects input/output for SOT5.

bit13:12	Description
Reading	Reads out the register value.
Writing 00	Uses SOT5_0 at the input pin of MFS-ch5-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT5_0 at the input pin of MFS-ch5-SOT. Uses SOT5_0 at the output pin.
Writing 10	Uses SOT5_1 at the input pin of MFS-ch5-SOT. Uses SOT5_1 at the output pin.
Writing 11	Uses SOT5_2 at the input pin of MFS-ch5-SOT. Uses SOT5_2 at the output pin.

[bit11:10] SIN5S : SIN5S Input Select Bit

Selects input for SIN5.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Uses SIN5_0 at the input pin of MFS-ch5-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN5_1 at the input pin of MFS-ch5-SIN.
Writing 11	Uses SIN5_2 at the input pin of MFS-ch5-SIN.

[bit9:8] SCK4B : SCK4 Input/Output Select Bit

Selects input/output for SCK4.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses SCK4_0 at the input pin of MFS-ch4-SCK. Does not produce output. [Initial value]
Writing 01	Uses SCK4_0 at the input pin of MFS-ch4-SCK. Uses SCK4_0 at the output pin.
Writing 10	Uses SCK4_1 at the input pin of MFS-ch4-SCK. Uses SCK4_1 at the output pin.
Writing 11	Uses SCK4_2 at the input pin of MFS-ch4-SCK. Uses SCK4_2 at the output pin.

[bit7:6] SOT4B : SOT4B Input/Output Select Bit

Selects input/output for SOT4.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Uses SOT4_0 at the input pin of MFS-ch4-SOT. Does not produce output. [Initial value]
Writing 01	Uses SOT4_0 at the input pin of MFS-ch4-SOT. Uses SOT4_0 at the output pin.
Writing 10	Uses SOT4_1 at the input pin of MFS-ch4-SOT. Uses SOT4_1 at the output pin.
Writing 11	Uses SOT4_2 at the input pin of MFS-ch4-SOT. Uses SOT4_2 at the output pin.

[bit5:4] SIN4S : SIN4S Input Select Bit

Selects input for SIN4.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses SIN4_0 at the input pin of MFS-ch4-SIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses SIN4_1 at the input pin of MFS-ch4-SIN.
Writing 11	Uses SIN4_2 at the input pin of MFS-ch4-SIN.

[bit3:2] CTS4S : CTS4S Input Select Bit

Selects input for CTS4.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Uses CTS4_0 at the input pin of MFS-ch4-CTS. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses CTS4_1 at the input pin of MFS-ch4-CTS.
Writing 11	Uses CTS4_2 at the input pin of MFS-ch4-CTS.

[bit1:0] RTS4E : RTS4E Output Select Bit

Selects output for RTS4.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for MFS-ch4-RTS. [Initial value]
Writing 01	Uses RTS4_0 at the output pin of MFS-ch4-RTS.
Writing 10	Uses RTS4_1 at the output pin of MFS-ch4-RTS.
Writing 11	Uses RTS4_2 at the output pin of MFS-ch4-RTS.

4.16. Extended Pin Function Setting Register 09 (EPFR09)

The EPFR09 register assigns functions to CAN, ADC trigger, and QPRC peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	CTX1E		CRX1S		CTX0E		CRX0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	
bit	23	22	21	20	19	18	17	16
Field	ADTRG2S				ADTRG1S			
Attribute	R/W				R/W			
Initial value	4'b0000				4'b0000			
bit	15	14	13	12	11	10	9	8
Field	ADTRG0S				QZIN1S		QBIN1S	
Attribute	R/W				R/W		R/W	
Initial value	4'b0000				2'b00		2'b00	
bit	7	6	5	4	3	2	1	0
Field	QAIN1S		QZIN0S		QBIN0S		QAIN0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	2'b00		2'b00		2'b00		2'b00	

■ Register Function

[bit31:30] CTX1E : CTX1E Output Select Bit

Selects output for CAN TX1.

bit31:30	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for CAN-ch1-TX. [Initial value]
Writing 01	Sets the output pin of CAN-ch1-TX to TX1_0.
Writing 10	Sets the output pin of CAN-ch1-TX to TX1_1.
Writing 11	Sets the output pin of CAN-ch1-TX to TX1_2.

[bit29:28] CRX1S : CRX1S Input Select Bit

Selects input for CAN RX1.

bit29:28	Description
Reading	Reads out the register value.
Writing 00	Sets the input pin of CAN-ch1-RX to RX1_0. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Sets the input pin of CAN-ch1-RX to RX1_1.
Writing 11	Sets the input pin of CAN-ch1-RX to RX1_2.

[bit27:26] CTX0E : CTX0E Output Select Bit

Selects output for CAN TX0.

bit27:26	Description
Reading	Reads out the register value.
Writing 00	Does not produce output for CAN-ch0-TX. [Initial value]
Writing 01	Sets the output pin of CAN-ch0-TX to TX0_0.
Writing 10	Sets the output pin of CAN-ch0-TX to TX0_1.
Writing 11	Sets the output pin of CAN-ch0-TX to TX0_2.

[bit25:24] CRX0S : CRX0S Input Select Bit

Selects input for CAN RX0.

bit25:24	Description
Reading	Reads out the register value.
Writing 00	Sets the input pin of CAN-ch0-RX to RX0_0. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Sets the input pin of CAN-ch0-RX to RX0_1.
Writing 11	Sets the input pin of CAN-ch0-RX to RX0_2.

[bit23:20] ADTRG2S : ADTRG2 Input Select Bit

Selects input for ADTRG2.

bit23:20	Description
Reading	Reads out the register value.
Writing 0000	Uses ADTG_0 at the input pin of ADC unit 2's startup trigger. [Initial value]
Writing 0001	Same as Writing 0000.
Writing 0010	Uses ADTG_1 at the input pin of ADC unit 2's startup trigger.
Writing 0011	Uses ADTG_2 at the input pin of ADC unit 2's startup trigger.
Writing 0100	Uses ADTG_3 at the input pin of ADC unit 2's startup trigger.
Writing 0101	Uses ADTG_4 at the input pin of ADC unit 2's startup trigger.
Writing 0110	Uses ADTG_5 at the input pin of ADC unit 2's startup trigger.
Writing 0111	Uses ADTG_6 at the input pin of ADC unit 2's startup trigger.
Writing 1000	Uses ADTG_7 at the input pin of ADC unit 2's startup trigger.
Writing 1001	Uses ADTG_8 at the input pin of ADC unit 2's startup trigger.
Writing other data	Setting disabled.

[bit19:16] ADTRG1S : ADTRG1 Input Select Bit

Selects input for ADTRG1.

bit19:16	Description
Reading	Reads out the register value.
Writing 0000	Uses ADTG_0 at the input pin of ADC unit 1's startup trigger. [Initial value]
Writing 0001	Same as Writing 0000.
Writing 0010	Uses ADTG_1 at the input pin of ADC unit 1's startup trigger.
Writing 0011	Uses ADTG_2 at the input pin of ADC unit 1's startup trigger.
Writing 0100	Uses ADTG_3 at the input pin of ADC unit 1's startup trigger.
Writing 0101	Uses ADTG_4 at the input pin of ADC unit 1's startup trigger.
Writing 0110	Uses ADTG_5 at the input pin of ADC unit 1's startup trigger.
Writing 0111	Uses ADTG_6 at the input pin of ADC unit 1's startup trigger.
Writing 1000	Uses ADTG_7 at the input pin of ADC unit 1's startup trigger.
Writing 1001	Uses ADTG_8 at the input pin of ADC unit 1's startup trigger.
Writing other data	Setting disabled.

[bit15:12] ADTRG0S : ADTRG0 Input Select Bit

Selects input for ADTRG0.

bit15:12	Description
Reading	Reads out the register value.
Writing 0000	Uses ADTG_0 at the input pin of ADC unit 0's startup trigger. [Initial value]
Writing 0001	Same as Writing 0000.
Writing 0010	Uses ADTG_1 at the input pin of ADC unit 0's startup trigger.
Writing 0011	Uses ADTG_2 at the input pin of ADC unit 0's startup trigger.
Writing 0100	Uses ADTG_3 at the input pin of ADC unit 0's startup trigger.
Writing 0101	Uses ADTG_4 at the input pin of ADC unit 0's startup trigger.
Writing 0110	Uses ADTG_5 at the input pin of ADC unit 0's startup trigger.
Writing 0111	Uses ADTG_6 at the input pin of ADC unit 0's startup trigger.
Writing 1000	Uses ADTG_7 at the input pin of ADC unit 0's startup trigger.
Writing 1001	Uses ADTG_8 at the input pin of ADC unit 0's startup trigger.
Writing other data	Setting disabled.

[bit11:10] QZIN1S : QZIN1S Input Select Bit

Selects input for QPRC ZIN1.

bit11:10	Description
Reading	Reads out the register value.
Writing 00	Uses ZIN1_0 at the input pin of QPRC-ch1's ZIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses ZIN1_1 at the input pin of QPRC-ch1's ZIN.
Writing 11	Uses ZIN1_2 at the input pin of QPRC-ch1's ZIN.

[bit9:8] QBIN1S : QBIN1S Input Select Bit

Selects input for QPRC BIN1.

bit9:8	Description
Reading	Reads out the register value.
Writing 00	Uses BIN1_0 at the input pin of QPRC-ch1's BIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses BIN1_1 at the input pin of QPRC-ch1's BIN.
Writing 11	Uses BIN1_2 at the input pin of QPRC-ch1's BIN.

[bit7:6] QAIN1S : QAIN1S Input Select Bit

Selects input for QPRC AIN1.

bit7:6	Description
Reading	Reads out the register value.
Writing 00	Uses AIN1_0 at the input pin of QPRC-ch1's AIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses AIN1_1 at the input pin of QPRC-ch1's AIN.
Writing 11	Uses AIN1_2 at the input pin of QPRC-ch1's AIN.

[bit5:4] QZIN0S : QZIN0S Input Select Bit

Selects input for QPRC ZIN0.

bit5:4	Description
Reading	Reads out the register value.
Writing 00	Uses ZIN0_0 at the input pin of QPRC-ch0's ZIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses ZIN0_1 at the input pin of QPRC-ch0's ZIN.
Writing 11	Uses ZIN0_2 at the input pin of QPRC-ch0's ZIN.

[bit3:2] QBIN0S : QBIN0S Input Select Bit

Selects input for QPRC BIN0.

bit3:2	Description
Reading	Reads out the register value.
Writing 00	Uses BIN0_0 at the input pin of QPRC-ch0's BIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses BIN0_1 at the input pin of QPRC-ch0's BIN.
Writing 11	Uses BIN0_2 at the input pin of QPRC-ch0's BIN.

[bit1:0] QAIN0S : QAIN0S Input Select Bit

Selects input for QPRC AIN0.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	Uses AIN0_0 at the input pin of QPRC-ch0's AIN. [Initial value]
Writing 01	Same as Writing 00.
Writing 10	Uses AIN0_1 at the input pin of QPRC-ch0's AIN.
Writing 11	Uses AIN0_2 at the input pin of QPRC-ch0's AIN.

4.17. Extended Pin Function Setting Register 10 (EPFR10)

The EPFR10 register assigns functions to external bus peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	UEA24E	UEA23E	UEA22E	UEA21E	UEA20E	UEA19E	UEA18E	UEA17E
Attribute	R/W							
Initial value	1'b0							

bit	23	22	21	20	19	18	17	16
Field	UEA16E	UEA15E	UEA14E	UEA13E	UEA12E	UEA11E	UEA10E	UEA09E
Attribute	R/W							
Initial value	1'b0							

bit	15	14	13	12	11	10	9	8
Field	UEA08E	UEAOOE	UECS7E	UECS6E	UECS5E	UECS4E	UECS3E	UECS2E
Attribute	R/W							
Initial value	1'b0							

bit	7	6	5	4	3	2	1	0
Field	UECS1E	UEFLSE	UEOEXE	UEDQME	UEWEXE	TESTB	UEDTHB	UEDEFB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

■ Register Function

[bit31] UEA24E : UEA24E Output Select Bit

Selects output for external bus Adress24.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD24. [Initial value]
Writing 1	Produces output for user external bus MAD24.

[bit30] UEA23E : UEA23E Output Select Bit

Selects output for external bus Adress23.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD23. [Initial value]
Writing 1	Produces output for user external bus MAD23.

[bit29] UEA22E : UEA22E Output Select Bit

Selects output for external bus Adress22.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD22. [Initial value]
Writing 1	Produces output for user external bus MAD22.

[bit28] UEA21E : UEA21E Output Select Bit

Selects output for external bus Adress21.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD21. [Initial value]
Writing 1	Produces output for user external bus MAD21.

[bit27] UEA20E : UEA20E Output Select Bit

Selects output for external bus Adress20.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD20. [Initial value]
Writing 1	Produces output for user external bus MAD20.

[bit26] UEA19E : UEA19E Output Select Bit

Selects output for external bus Adress19.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD19. [Initial value]
Writing 1	Produce output for user external bus MAD19.

[bit25] UEA18E : UEA18E Output Select Bit

Selects output for external bus Adress18.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD18. [Initial value]
Writing 1	Produces output for user external bus MAD18.

[bit24] UEA17E : UEA17E Output Select Bit

Selects output for external bus Adress17.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD17. [Initial value]
Writing 1	Produces output for user external bus MAD17.

[bit23] UEA16E : UEA16E Output Select Bit

Selects output for external bus Adress16.

bit	Description
Reading	Reads out the register value.
Writing 0	Produces output for user external bus MAD16. [Initial value]
Writing 1	Produces output for user external bus MAD16.

[bit22] UEA15E : UEA15E Output Select Bit
Selects output for external bus Adress15.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD15. [Initial value]
Writing 1	Produces output for user external bus MAD15.

[bit21] UEA14E : UEA14E Output Select Bit
Selects output for external bus Adress14.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD14. [Initial value]
Writing 1	Produces output for user external bus MAD14.

[bit20] UEA13E : UEA13E Output Select Bit
Selects output for external bus Adress13.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD13. [Initial value]
Writing 1	Produces output for user external bus MAD13.

[bit19] UEA12E : UEA12E Output Select Bit
Selects output for external bus Adress12.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD12. [Initial value]
Writing 1	Produces output for user external bus MAD12.

[bit18] UEA11E : UEA11E Output Select Bit
Selects output for external bus Adress11.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD11. [Initial value]
Writing 1	Produces output for user external bus MAD11.

[bit17] UEA10E : UEA10E Output Select Bit
Selects output for external bus Adress10.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD10. [Initial value]
Writing 1	Produces output for user external bus MAD10.

[bit16] UEA09E : UEA09E Output Select Bit
Selects output for external bus Adress09.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD09. [Initial value]
Writing 1	Produces output for user external bus MAD09.

[bit15] UEA08E : UEA08E Output Select Bit
Selects output for external bus Adress08.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD08. [Initial value]
Writing 1	Produces output for user external bus MAD08.

[bit14] UEAOOE : UEAOOE Output Select Bit
Selects output for external bus Adress00.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD00. [Initial value]
Writing 1	Produces output for user external bus MAD00.

[bit13] UECS7E : UECS7E Output Select Bit
Selects output for external bus CS7.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX7. [Initial value]
Writing 1	Produces output for user external bus MCSX7.

[bit12] UECS6E : UECS6E Output Select Bit
Selects output for external bus CS6.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX6. [Initial value]
Writing 1	Selects output for user external bus MCSX6.

[bit11] UECS5E : UECS5E Output Select Bit
Selects output for external bus CS5.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX5. [Initial value]
Writing 1	Produces output for user external bus MCSX5.

[bit10] UECS4E : UECS4E Output Select Bit

Selects output for external bus CS4.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX4. [Initial value]
Writing 1	Produces output for user external bus MCSX4.

[bit9] UECS3E : UECS3E Output Select Bit

Selects output for external bus CS3.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX3. [Initial value]
Writing 1	Produces output for user external bus MCSX3.

[bit8] UECS2E : UECS2E Output Select Bit

Selects output for external bus CS2.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX2. [Initial value]
Writing 1	Produces output for user external bus MCSX2.

[bit7] UECS1E : UECS1E Output Select Bit

Selects output for external bus CS1.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MCSX1. [Initial value]
Writing 1	Produces output for user external bus MCSX1.

[bit6] UEFLSE : UEFLSE Output Select Bit

Selects output for external bus NAND-Flash control signal.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MNALE, MNCLE, MNWEX, or MNREX. [Initial value]
Writing 1	Produces output for user external bus MNALE, MNCLE, MNWEX, and MNREX.

[bit5] UEOEXE : UEOEXE Output Select Bit

Selects output for external bus OEX.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MOEX. [Initial value]
Writing 1	Produces output for user external bus MOEX.

[bit4] UEDQME : UEDQME Output Select Bit
Selects output for external bus DQM.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MDQM1 or MDQM0. [Initial value]
Writing 1	Produces output for user external bus MDQM1 and MDQM0.

[bit3] UEWEXE : UEWEXE Output Select Bit
Selects output for external bus WEX.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MWEX. [Initial value]
Writing 1	Produces output for user external bus MWEX.

[bit2] TESTB : TESTB Test Bit
This bit is used for test.

bit	Description
Reading	Reads out the register value.
Writing 0	Always write "0". [Initial value]
Writing 1	Setting disabled.

[bit1] UEDTHB : UEDTHB Input/Output Select Bit
Selects input/output for external bus data.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus WD15-WD08. [Initial value] Input of user external bus RD15-RD08 is connected to the pin.
Writing 1	Produces output for user external bus WD15-WD08. Input of user external bus RD15-RD08 is connected to the pin.

[bit0] UEDEFB : UEDEFB Input/Output Select Bit
Selects input/output for external bus signal.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not produce output for user external bus MAD07-MAD01. Does not produce output for user external bus MCSX0. Does not produce output for user external bus WD07-WD00. Input of user external bus RD07-RD00 is connected to the pin. [Initial value]
Writing 1	Produces output for user external bus MAD07-MAD01. Produces output for user external bus MCSX0. Produces output for user external bus WD07-WD00. Input of user external bus RD07-RD00 is connected to the pin.

4.18. Special Port Setting Register (SPSR)

The SPSR register sets a pin as a signal pin of special functions.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field					Reserved			
Attribute					-			
Initial value					-			
bit	23	22	21	20	19	18	17	16
Field					Reserved			
Attribute					-			
Initial value					-			
bit	15	14	13	12	11	10	9	8
Field					Reserved			
Attribute					-			
Initial value					-			
bit	7	6	5	4	3	2	1	0
Field		Reserved		USB0C		Reserved		SUBXC
Attribute		-		R/W		-		R/W
Initial value				1'b0				1'b1

■ Register Function

[bit31:8] res : Reserved Bit

"0x000000" is read from these bits.

When writing these bits, set them to "0x000000".

[bit7:5] res : Reserved Bit

"0b000" is read from these bits.

When writing these bits, set them to "0b000".

[bit4] USB0C : USB (ch.0) Pin Setting Register

This bit sets a pin as a USB pin.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use two pins of UDM0 and UDP0 as USB pins but as digital input/output pins. [Initial value]
Writing 1	Uses two pins of UDM0 and UDP0 as USB pins. (An I/O cell will be in a state of input direction and input cut-off.)

[bit3:1] res : Reserved Bit

"0b000" is read from these bits.

When writing these bits, set them to "0b000".

[bit0] SUBXC : Sub Clock (Oscillation) Pin Setting Register

This bit sets a pin as a sub clock (oscillation) pin.

bit	Description
Reading	Reads out the register value.
Writing 0	Does not use two pins of X0A and X1A as sub clock (oscillation) pins but as digital input/output pins.
Writing 1	Uses two pins of X0A and X1A as sub clock (oscillation) pins. [Initial value] (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)

<Note>

Writing "1" to the SUBXC bit does not make a sub clock start oscillation.

To make it start oscillation, enable oscillation by the SOSCE bit of the System Clock Mode Control Register (SCM_CTL) described in the chapter "CLOCK".

5. Usage Precautions

This section describes precautions for using this I/O board.

● ON/OFF of the Pull-up Resistance When SPL=1

SPL is a signal for turning a pin into Hi-Z state during standby mode.

- When SPL=0 Normal operations
- When SPL=1 Pin Hi-Z, input cut-off, pull-up disconnection
However, the SPL bit cannot be used for setting external interrupts, NMIX, JTAG, or TRACE pins.

For details of the SPL bit, see Chapter "Low Power Consumption Mode".

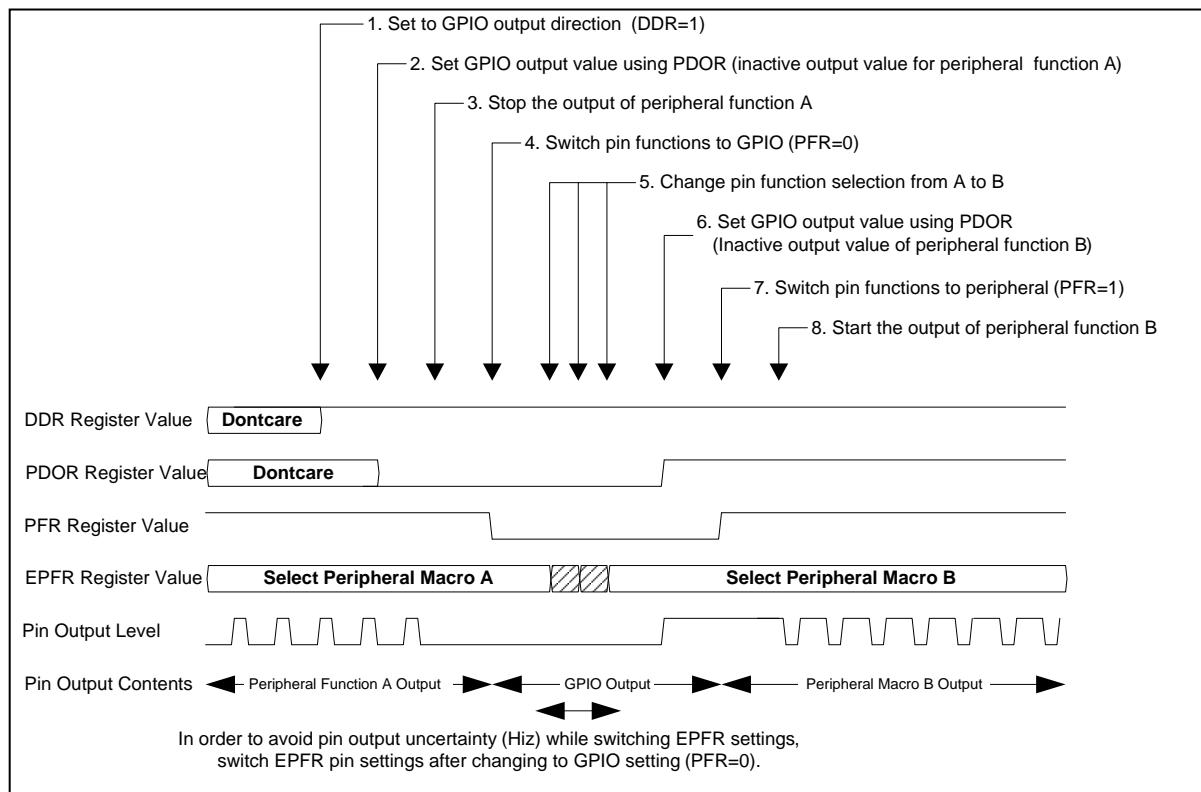
● DTTIX Input

DTTI input is an input signal for switching the dual-purpose motor control PWM output (RTO) setting output pin to its other GPIO pin setting to address a motor stop demand in an emergency.
To use this function, enable switching by EPFR.

● Procedures for Switching Pin Functions

When switching the outputs for peripheral functions using the EPFR register, to prevent pin uncertain output (Hi-Z), switch settings according to the procedures similar to the switching example shown in the following Figure 5-1.

Figure 5-1 Procedures for Switching Pin Functions



● **Reserved Bit**

This bit is read out as "0" except for the ADE reserved bit. When writing, always write "0". The ADE reserved bit is read out as "1". When writing, always write "1".

● **Connecting External Bus Pin and SRAM**

When accessing SRAM via external bus, either perform pull-up setting for the pin or connect it to external pull-up pin.

● **Multi-function Serial Pin Group**

When there are some multi-function serial inputs/outputs, set each input/output to the port of the same group. "The port of the same group" means that relocate function numbers attached to the pin name are the same, just like "xxx_0" or "yyy_1".

The following Table 5-1 shows example setting.

Table 5-1 Multi-function Serial Interface example setting

Serial Data Output	Serial Clock Input/Output	Serial Data Input	Effective Port
Pin SOUT1_0 (Port 0)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	Port 0
		Pin SIN1_1 (Port 1)	Setting Disabled
	Pin SCK1_1 (Port 1)	Pin SIN1_0 (Port 0)	Setting Disabled
		Pin SIN1_1 (Port 1)	
Pin SOUT1_1 (Port 1)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	
		Pin SIN1_1 (Port 1)	
	Pin SCK1_1 (Port 1)	Pin SIN1 (Port 0)	
		Pin SIN1_1 (Port 1)	Port 1

● **Peripheral Function Output**

As output pins for peripheral functions are uniquely determined by EPFR settings, Output for peripheral functions cannot be assigned to separate pins.

(Disabled example) Assign multifunction serial output SOUT1_0 and SOUT1_1 to the same output.

● **Pin Settings and Operation Mode**

For JTAG settings, see Chapter "Debug".

For state of each pin during standby mode or reset, see "Data Sheet".

● **Product Specifications and Peripheral Function Assignment**

Functions which are assigned to pins (GPO, peripheral output and I/O) vary in different products.

Please see the pin function table of the data sheet to confirm the pin function of each product.

Do not select a function for a pin which is not available in your product by using the EPFR register setting.

Chapter: Clock supervisor

This chapter explains clock supervisor functions.

1. Overview
2. Configurations and Block Diagrams
3. Explanation of Operations
4. Setup Procedure Examples
5. Operation Examples
6. Register list
7. Usage Precautions

1. Overview

This section provides an overview of clock supervisor functions.

The clock supervisor includes the following two types of functions.

- **Clock failure detection (CSV: Clock failure detection by clock Super Visor)**

The clock failure detection monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

- **Anomalous frequency detection (FCS: anomalous Frequency detection by Clock Super visor)**

The anomalous frequency detection monitors frequency of the main clock. Within the specified period between an edge and the next edge of the divided clock of high-speed CR, this function counts up the internal counter value using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request or a system reset request to the CPU.

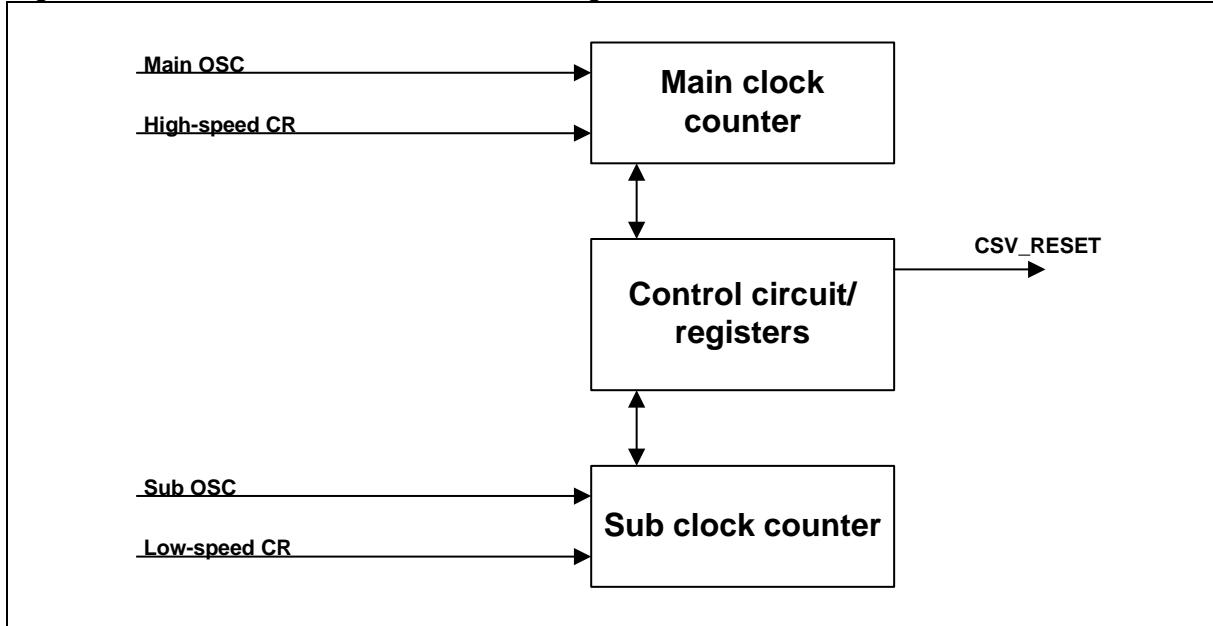
2. Configurations and Block Diagrams

This section explains the block diagrams of clock supervisor functions.

■ Clock failure detection

Figure 2-1 shows the block diagram of the clock failure detection.

Figure 2-1 Clock Failure Detection Block Diagram



The clock failure detection consists of the following three types of blocks.

● Control circuit/registers

- This block includes a circuit controlling clock failure detection,
- Also includes setup registers enabling/disabling the clock failure detection.

● Main clock counter

A counter that monitors the main clock with the high-speed CR clock.

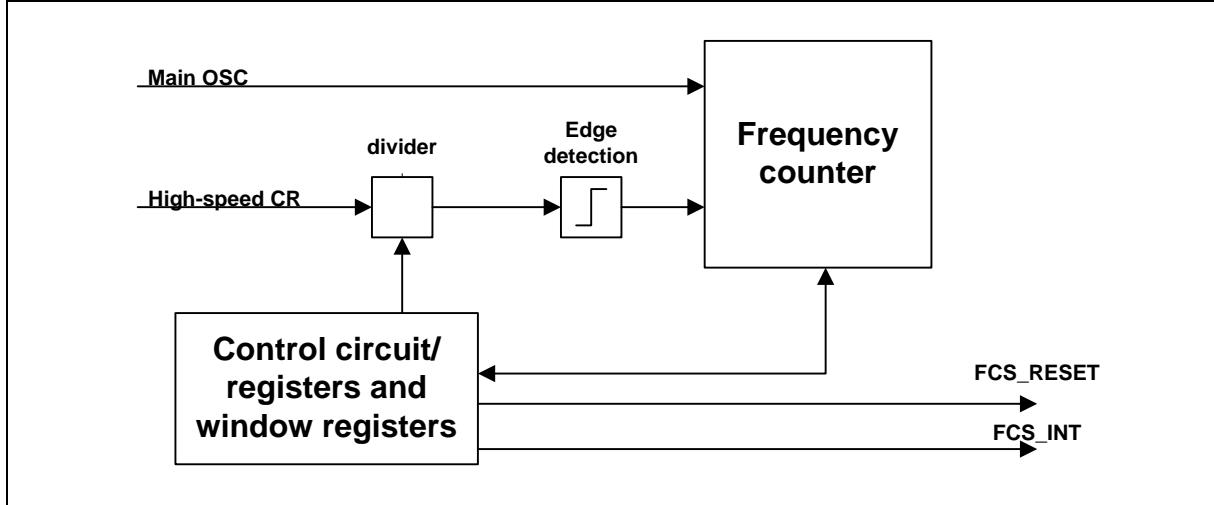
● Sub clock counter

A counter that monitors the sub clock with the low-speed CR clock.

■ Anomalous frequency detection

Figure 2-2 shows the block diagram of the anomalous frequency detection.

Figure 2-2 Anomalous Frequency Detection Block Diagram



The anomalous frequency detection consists of the following three types of blocks.

- **Control circuit/registers and window registers**

- This block includes a circuit controlling the anomalous frequency detection.
- Also includes setup registers enabling/disabling the anomalous frequency detection.
- Also includes window registers defining the frequency range for measurements.

- **Frequency counter**

A counter based on the main clock.

- **Divider/edge detection**

- This block divides the high-speed CR.
- Also detects rising edges of the divided clock of high-speed CR.

3. Explanation of Operations

This section explains the operations of clock supervisor functions.

■ Clock failure detection

The clock failure detection monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

- This reset request is referred to as the CSV reset request.
- CSV function monitors each of the main and sub clocks independently.
- It stops monitoring when the main and sub oscillators stop oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- When the oscillation stabilization wait time of main and sub oscillators ends, CSV function is automatically enabled.

<Notes>

- Each of the main and sub clocks can be enabled/disabled independently using the CSV_CTL register.
- The main clock is monitored with the high-speed CR clock, and the sub clock is monitored with the low-speed CR clock. When a rising edge is not detected within 32 clocks of high-speed CR for the main clock, or within 32 clocks of low-speed CR for the sub clock, this function determines that the oscillator has failed.

■ Anomalous frequency detection

The anomalous frequency detection monitors the main clock.

Within the specified period between an edge and the next edge of the divided clock of high-speed CR, this function counts up the internal counter using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request or a system reset request to the CPU.

- This interrupt request is referred to as the FCS interrupt request, and reset request as the FCS reset request.
- The FCS function only monitors frequency of the main clock.
- It stops monitoring when the main oscillator stops oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- The FCS function is started with software, a user program.

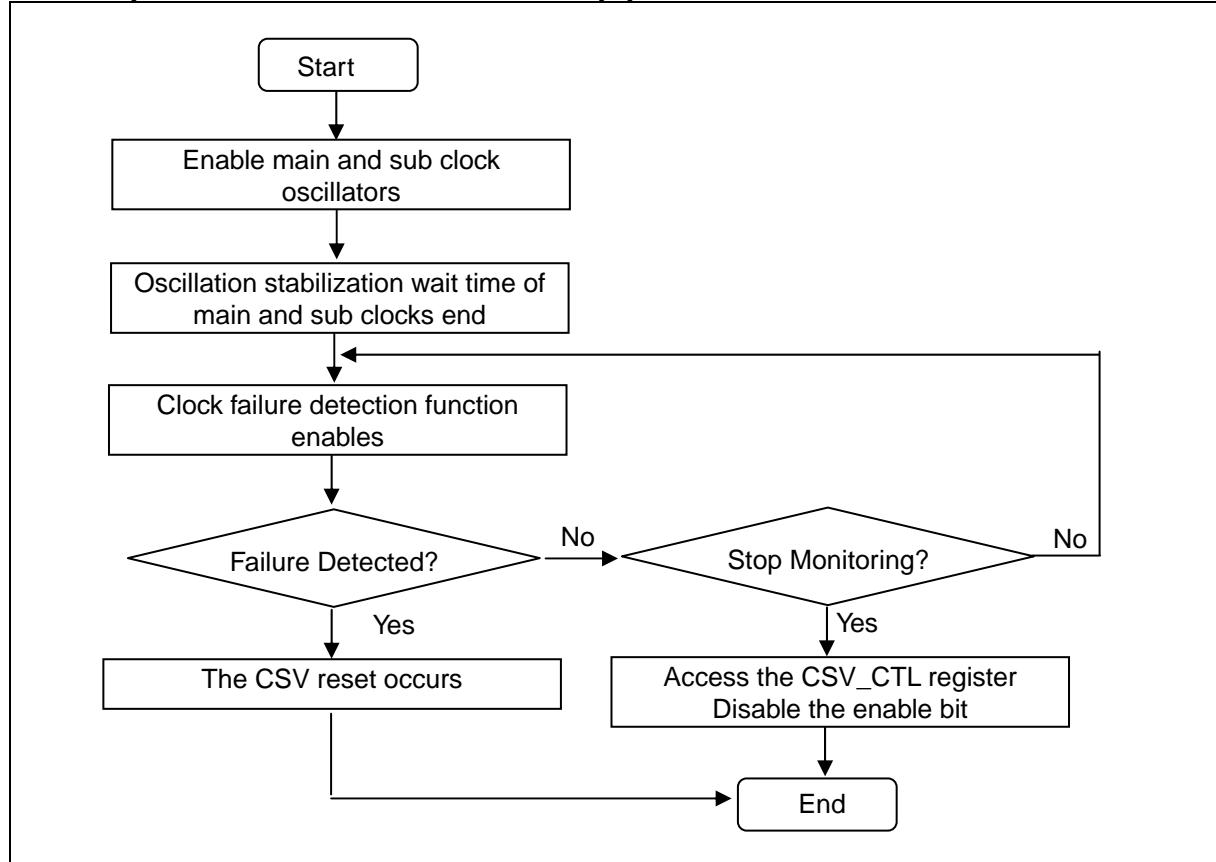
<Notes>

- If the FCS reset is enabled:
An interrupt request occurs the first time a counter value deviates from the set window. If the interrupt request has not been cleared, and the counter value falls out of the specified window, a system reset request is output.
If the FCS reset is not enabled, the reset request is masked.
- The counter value, if it goes out of the specified window, is stored in the FCSWD_CTL register.

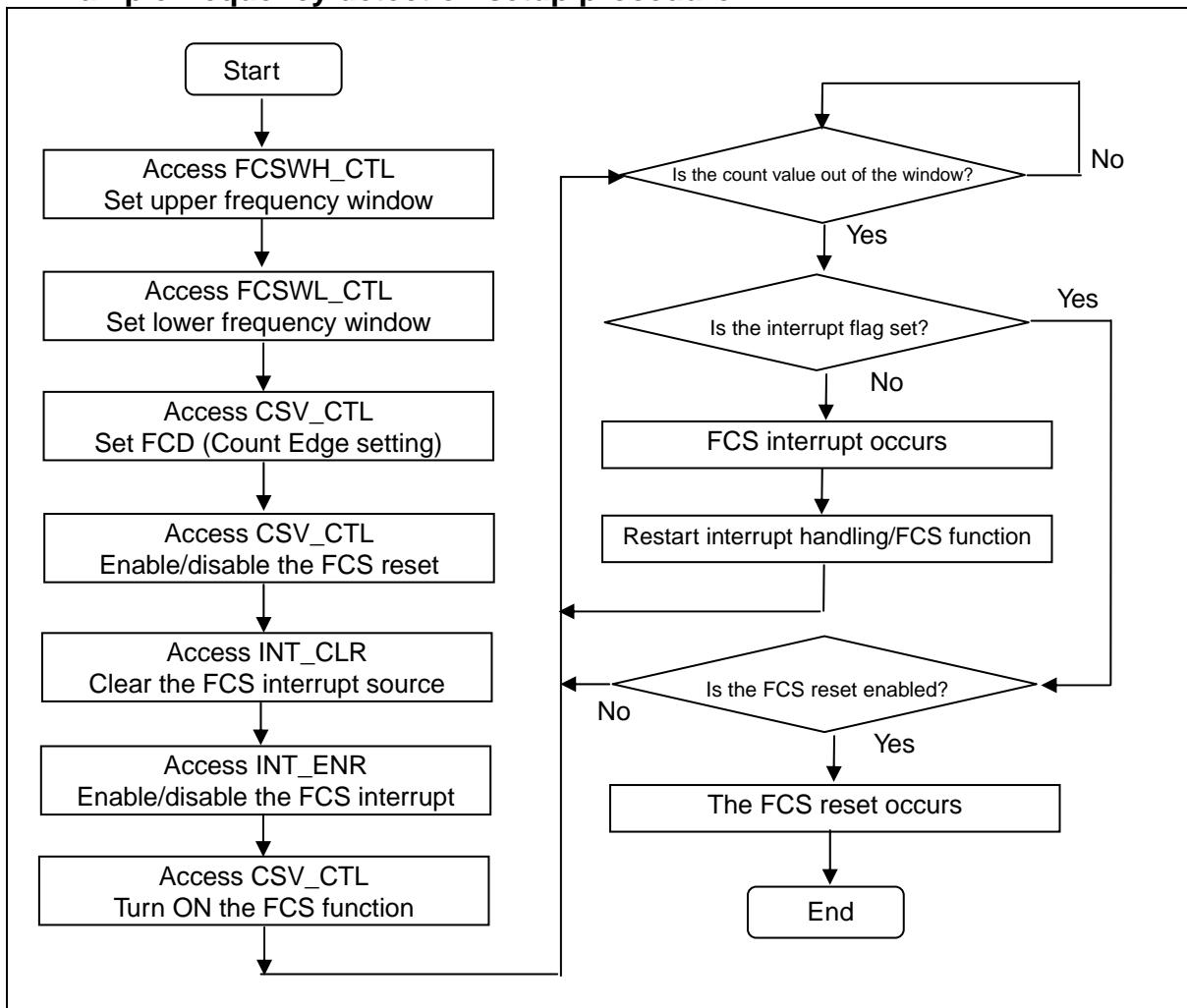
4. Setup Procedure Examples

This section explains examples of setting up clock supervisor functions.

■ Example clock failure detection setup procedure



■ Example frequency detection setup procedure



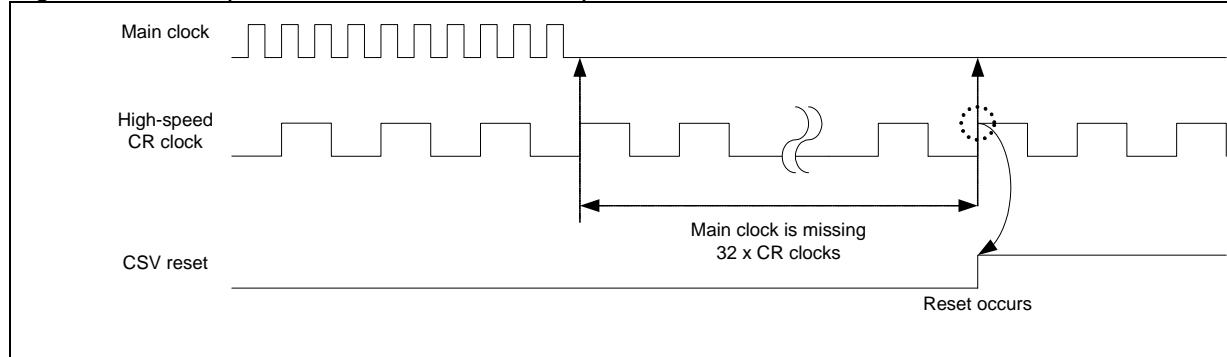
5. Operation Examples

This section explains examples of clock supervisor operations.

■ Clock failure detection

Figure 5-1 provides an example of clock failure detection operation.

Figure 5-1 Example clock failure detection operation

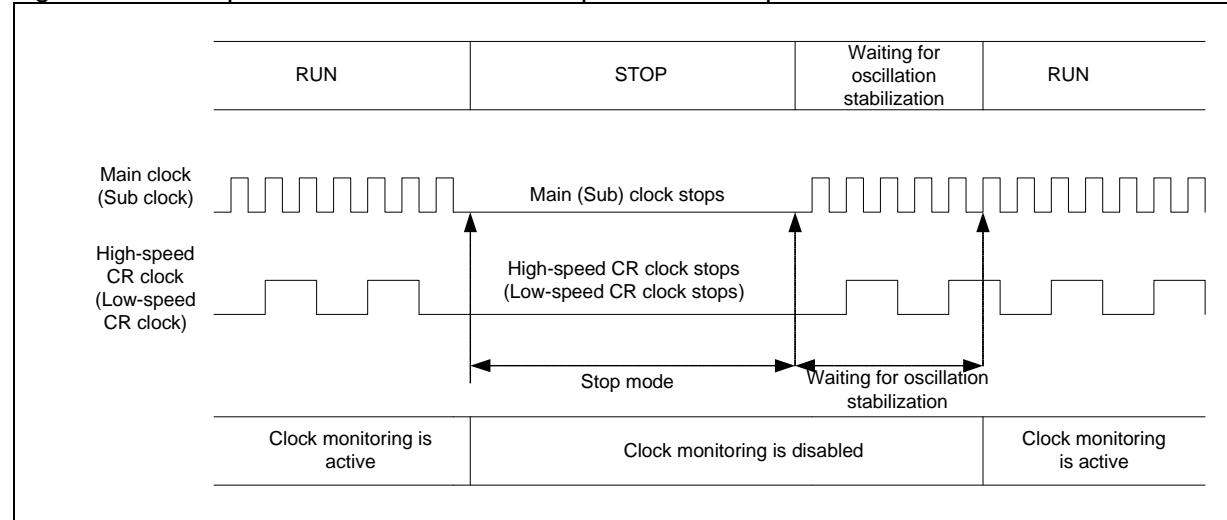


1. The main clock stops due to failure.
2. The function counts up clocks using the high-speed CR clock.
3. If the main clock keeps stopping during 32 clocks of high-speed CR, the function determines that the clock has failed and issues the CSV reset.

Note: In case of the sub clock, the function determines that the sub clock has failed if it keeps stopping during 32 clocks of low-speed CR.

Figure 5-2 provides an example of the clock failure detection operation in stop mode.

Figure 5-2 Example clock failure detection operation in stop mode

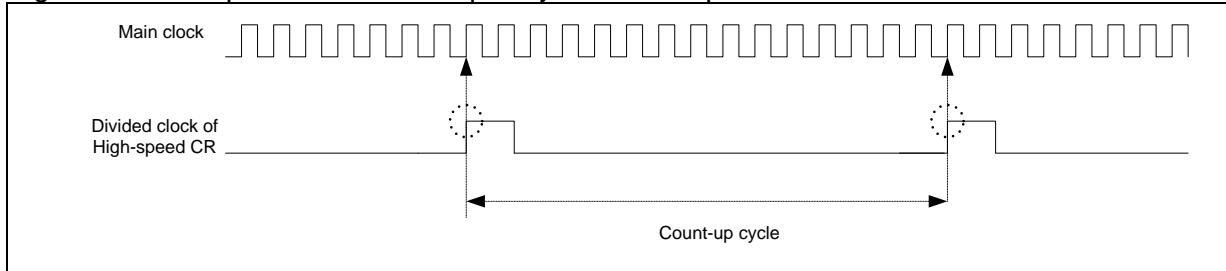


1. In stop mode, the main clock and high-speed CR clock stop. Meanwhile, the clock monitoring function also stops.
2. Upon the release of stop mode, oscillation of main clock and CR clock restart, waiting for oscillation stabilization. Meanwhile, the clock monitoring function keeps stopping.
3. When the oscillation stabilization wait time ends, the clock monitoring restarts.

■ Anomalous frequency detection

Figure 5-3 provides an example of anomalous frequency detection operation.

Figure 5-3 Example anomalous frequency detection operation



1. This function detects rising edges of the divided clock of high-speed CR.
 2. After detecting edges, it counts up clocks using the main clock.
 3. It keeps counting up until it detects the next rising edge of the divided clock of high-speed CR.
 4. Let " α " be the count value with the main clock.
- Also let B denote the lower window value, and A the upper window value. Compare the count value α with those window values and if expression
- $$B \leq \alpha \leq A$$
- holds true, then the frequency is considered to be normal.
- If the count value α is out of the range, i.e., either
- $$\alpha < B \text{, or } A < \alpha$$
- is true, then the frequency is considered to be anomalous, and an interrupt occurs.
- If the interrupt flag has not been cleared after the interrupt and an anomalous frequency is detected again, then the function issues a reset depending on the setting.

■ Example anomalous frequency detection window setting

The anomalous frequency detection counts up between edges of the divided clock of high-speed CR. The measurement interval is also affected by the accuracy of CR. When you configure the window register value, therefore, the CR accuracy must be considered for the value.

For frequency accuracy of the CR oscillator, check the relevant data sheet.

● Calculation method

The count value range of anomalous frequency detection must be added the internal CR accuracy, then, the window register value is set. The count range expression must be used as follows.

$$\text{Count value} = \left\{ \frac{1}{\text{Frequency of divided clock of CR} \times \left(1 \pm \frac{\text{CR accuracy}}{100} \right)} \right\} \times \text{Frequency of main clock}$$

The count value by main clock of frequency L [Hz] can be calculated using the divide-by-Y high-speed CR oscillator clock of $\pm Z\%$ accuracy with frequency K [Hz].

$$\text{Count value A (positive CR frequency accuracy)} = 1 / [(K/Y) \times (1 + Z/100)] \times L$$

$$\text{Count value B (negative CR frequency accuracy)} = 1 / [(K/Y) \times (1 - Z/100)] \times L$$

Those expressions lead the count value within the range A to B added internal CR accuracy.

Set the value smaller than count value A for the lower limit of the window, and larger than count value B for the upper limit.

The window setting is determined by the value allowed for frequency fluctuation of main oscillation defined by the user.

● Example calculation

The count value by main clock of frequency 4M [Hz] is calculated using the divide-by-1024 high-speed CR oscillator clock of $\pm 5\%$ accuracy with frequency 4M [Hz].

Count value A (positive CR frequency accuracy)

$$\text{Count value A} = \left\{ \frac{1}{\frac{4 \times 10^6}{1024} \times \left(1 + \frac{5}{100} \right)} \right\} \times 4 \times 10^6 \approx 975$$

Count value B (negative CR frequency accuracy)

$$\text{Count value B} = \left\{ \frac{1}{\frac{4 \times 10^6}{1024} \times \left(1 - \frac{5}{100} \right)} \right\} \times 4 \times 10^6 \approx 1078$$

Those expressions yield the count value within the range 975 to 1078. If the window setting value is 5%, window value is as follows.

$$\text{Window lower limit} = 975 \times 0.95(-5\%) = 926.25 \approx 3.43\text{MHz}$$

$$\text{Window upper limit} = 1078 \times 1.05(+5\%) = 1131.9 \approx 4.64\text{MHz}$$

Thus, you can recognize that a main clock frequency out of the 3.4MHz to 4.6MHz range is anomalous. Table 5-1 provides an example of the window settings.

Table 5-1 Example window settings

Divided clock of High-speed CR	Main crystal oscillation	High-speed CR error	Count value including high-speed CR error	Lower limit of window set value	Upper limit of window set value
Divide-by-1024 clocks of CR:4MHz	4MHz	±5%	975 (\approx 3.61MHz) - 1078 (\approx 4.42MHz)	926 (\approx 3.43MHz)	1131 (\approx 4.64MHz)

6. Register list

This section explains the register list of clock supervisor functions.

■ Register list

Table 6-1 shows the register list.

Table 6-1 Register list

Abbreviation	Register name	See
CVS_CTL	CSV control register	6.1
CVS_STR	CSV status register	6.2
FCSWH_CTL	Frequency detection window setting register (Upper)	6.3
FCSWL_CTL	Frequency detection window setting register (Lower)	6.4
FCSWD_CTL	Frequency detection counter register	6.5

6.1. CSV control register (CSV_CTL)

The CSV_CTL register configures the control of CSV function.

■ Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved		FCD		Reserved		FCSRE	FCSDE
Attribute	-		R/W	-		R/W	R/W	
Initial value	1'b0		3'b111		2'b00		1'b0	1'b0
bit	7	6	5	4	3	2	1	0
Field			Reserved				SCSVE	MCSVE
Attribute			-				R/W	R/W
Initial value			6'b000000				1'b1	1'b1

■ Register functions

[bit 15] Reserved bit

"0b0" is read from these bits.

Set these bits to "0b0" when writing.

[bits 14:12] FCD: FCS count cycle setting bits

bits 14:12	Description
When 000 is written	Setting disabled
When 001 is written	Setting disabled
When 010 is written	Setting disabled
When 011 is written	Setting disabled
When 100 is written	Setting disabled
When 101 is written	1/256 frequency of high-speed CR oscillation
When 110 is written	1/512 frequency of high-speed CR oscillation
When 111 is written	1/1024 frequency of high-speed CR oscillation [Initial value]
When read	The register value is read.

[bits 11:10] Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[bit 9] FCSRE: FCS reset output enable bit

bit	Description
When 0 is written	The FCS reset is disabled [Initial value]
When 1 is written	The FCS reset is enabled
When read	The register value is read.

[bit 8] FCSDE: FCS function enable bit

bit	Description
When 0 is written	The FCS function is disabled [Initial value]
When 1 is written	The FCS function is enabled.
When read	The register value is read.

[bits 7:2] Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1] SCSVE: Sub CSV function enable bit

bit	Description
When 0 is written	The sub CSV function is disabled
When 1 is written	The sub CSV function is enabled. [Initial value]
When read	The register value is read.

[bit 0] MCSVE: Main CSV function enable bit

bit	Description
When 0 is written	The main CSV function is disabled
When 1 is written	The main CSV function is enabled. [Initial value]
When read	The register value is read.

<Note>

This register is not initialized by software reset.

6.2. CSV status register (CSV_STR)

The CSV_STR register indicates the status of CSV function.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved			SCMF	MCMF
Attribute			-				R	R
Initial value				6'b000000			1'b0	1'b0

■ Register functions

[bits 7:2] Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1] SCMF : Sub clock failure detection flag

bit	Description
When written	No effect
When 0 is read	No sub clock failure has been detected. [Initial value]
When 1 is read	A sub clock failure has been detected.

[bit 0] MCMF : Main clock failure detection flag

bit	Description
When written	No effect
When 0 is read	No main clock failure has been detected. [Initial value]
When 1 is read	A main clock failure has been detected.

<Note>

This register is cleared when being read.

6.3. Frequency detection window setting register (Upper) (FCSWH_CTL)

The FCSWH_CTL register configures the frequency detection window setting register (Upper).

■ Register configuration

bit	15	0
Field	FWH	
Attribute	R/W	
Initial value	16'0xFFFF	

■ Register functions

[bits 15:0] FWH: Frequency detection window setting bits (Upper)

bits 15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

<Notes>

- Set a value larger than the value set in FCSWL_CTL (Frequency detection window setting register (Lower)).
- This register is not initialized by software reset.

6.4. Frequency detection window setting register (Lower) (FCSWL_CTL)

The FCSWL_CTL register configures the frequency detection window setting register (Lower).

■ Register configuration

bit	15	0
Field		FWL
Attribute		R/W
Initial value		16'0x0000

■ Register functions

[bits 15:0] FWL: Frequency detection window setting bits (Lower)

bits 15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

<Notes>

- Set a value smaller than the value set in FCSWH_CTL (Frequency detection window setting register (Upper)).
- This register is not initialized by software reset.

6.5. Frequency detection counter register (FCSWD_CTL)

The FCSWD_CTL register indicates the counter value of frequency detection using the main clock.

■ Register configuration

bit	15	0
Field	FWD	
Attribute	R	
Initial value	16'0x0000	

■ Register functions

[bits 15:0] FWD: Frequency detection count data

bits 15:0	Description
When written	No effect
When read	The count value is read.

<Notes>

- This register retains the count value when detecting an error.
- This register is not initialized by software reset.

7. Usage Precautions

This section explains the precautions for using clock supervisor functions.

- For details on enabling and clearing the frequency detection interrupt sources, see Chapter "Clocks".
- For details on clock failure detection and anomalous frequency detection reset sources, see Chapter "Resets".
- Operation after the occurrence of a reset.
After the occurrence of a reset triggered by clock failure detection, clock mode returns to high-speed CR. Do not select the faulty clock again.
- The internal high-speed CR clock for use of the frequency detection
The frequency failure detection is affected by the frequency accuracy of internal high-speed CR itself. When you configure frequency window, therefore, the accuracy of internal high-speed CR must be considered for the window value.
Do not trim the high-speed CR clock after the anomalous frequency detection has been enabled.
- The order of the frequency detection settings before using
Before enabling FCS (FCSDE=1), specify the count cycle (FCD), reset enable (FCSRE), and frequency window (FWH/FWL) settings.
If you want to change any of FCD/FCSRE/FWH/FWL after FCS has been enabled, stop the FCS function before changing the setting. Do not change the setting while FCS is enabled.
- The enable settings of the anomalous frequency detection before using
Depending on the setting of the FCSRE bit in the CSV control register (CSV_CTL), operation during anomalous frequency detection varies. Table 7-1 shows the setting list.

Table 7-1 List of the FCS function and FCSRE bit settings

	FCSRE=0	FCSRE=1
FCSDE=0	Stops FCS function	Stops FCS function
FCSDE=1	Enables FCS function Generates an interrupt upon error detection	Enables FCS function An interrupt occurs upon the first error detection A reset occurs upon the second error detection

- Interrupt settings for the frequency detection and main timer mode
The internal bus clock stops while the clock is in main timer mode. In this mode, an interrupt does not occur even if an error is detected while FCSRE is set to 0.
In main timer mode, therefore, do not set FCSRE to 0. If FCSRE is set to 1, a reset occurs.
- The settings for CSV OFF and external reset.
When CSV function is set to OFF, the CSV reset is not generated, and moreover, the external reset (INITX) is not accepted. So, it is recommended not to turn OFF the CSV function, if you don't have special reason.

CHAPTER: Watchdog timer

This chapter describes the watchdog timer.

1. Overview
2. Configuration and Block Diagram
3. Operations
4. Setting Procedure Example
5. Operation Example
6. Registers
7. Notes

1. Overview

This section describes the overview of the watchdog timer.

The watchdog timer is a function to detect runaway of user program.

If the watchdog timer is not cleared within the specified interval time, it judges that a user program is out of control, and outputs a system reset request or an interrupt request to CPU.

This interrupt is called a watchdog interrupt request, and a reset request is called a watchdog reset request.

During watchdog timer operation, it is required to clear continually and periodically before the specified interval time has elapsed. If an abnormal operation of user program such as hanging up prevents it from being periodically cleared, it underflows and outputs a watchdog interrupt request or a watchdog reset request.

This MCU has two kinds of watchdog timers as follows.

● Software watchdog timer

- The software watchdog timer is activated by user program.
- A divided clock of APB bus clock is used for a count clock.
- It counts cycles while CPU program is operating, and it stops counting while APB clock of the standby mode (timer mode, stop mode, and during oscillation stabilization wait time of the source clock). The count value is retained so that it continues counting after returning from the standby mode.
- The software watchdog timer is stopped by all the resets.

● Hardware watchdog timer

- The hardware watchdog timer is activated by tuning on the device, and after releasing all the resets except software resets without an intervention of software.
- The hardware watchdog timer can be stopped by accessing a register by software.
- Low-speed CR clock (CLKLC) is used for a count clock.
- It counts cycles while CLKLC is operated, and it stops counting while CLKLC is stopped in the standby mode (stop mode). The count value is retained so that it continues counting after returning from the standby mode.

● Software/hardware watchdog timer

- Each watchdog timer has a lock register, accessing to all the registers of watchdog timers can not be done unless accessing and releasing a lock with a certain procedure.
- The watchdog timers can be reloaded by accessing to the watchdog clear register.
- When the first underflow of the watchdog counter is generated, an interrupt request is generated. When the second underflow is generated without clearing the interrupt request, a reset request is generated. This function can be set by the register.

2. Configuration and Block Diagram

This section shows the configuration and block diagram of the watchdog timer.

Figure 2-1 Block Diagram of Software Watchdog Timer

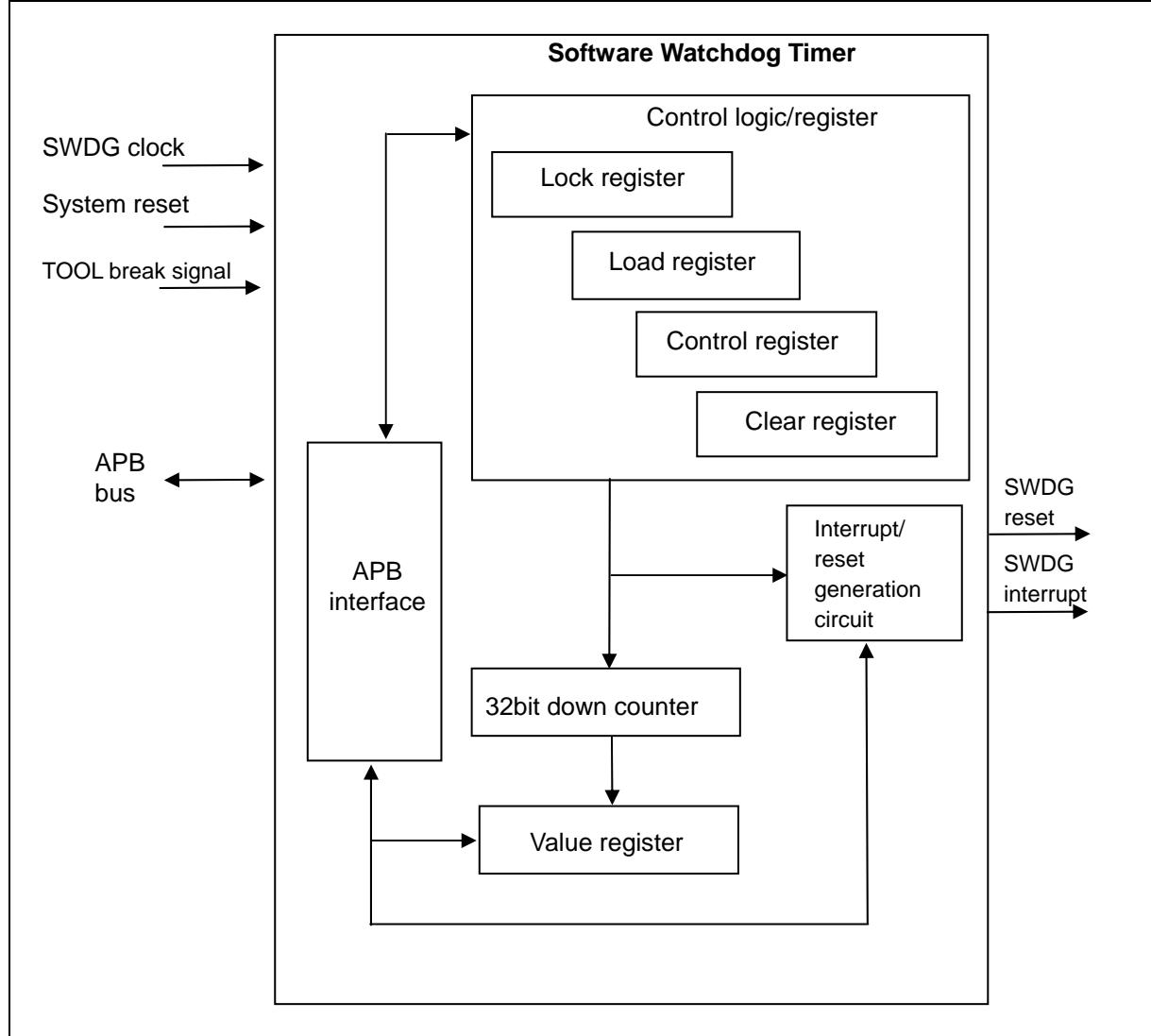
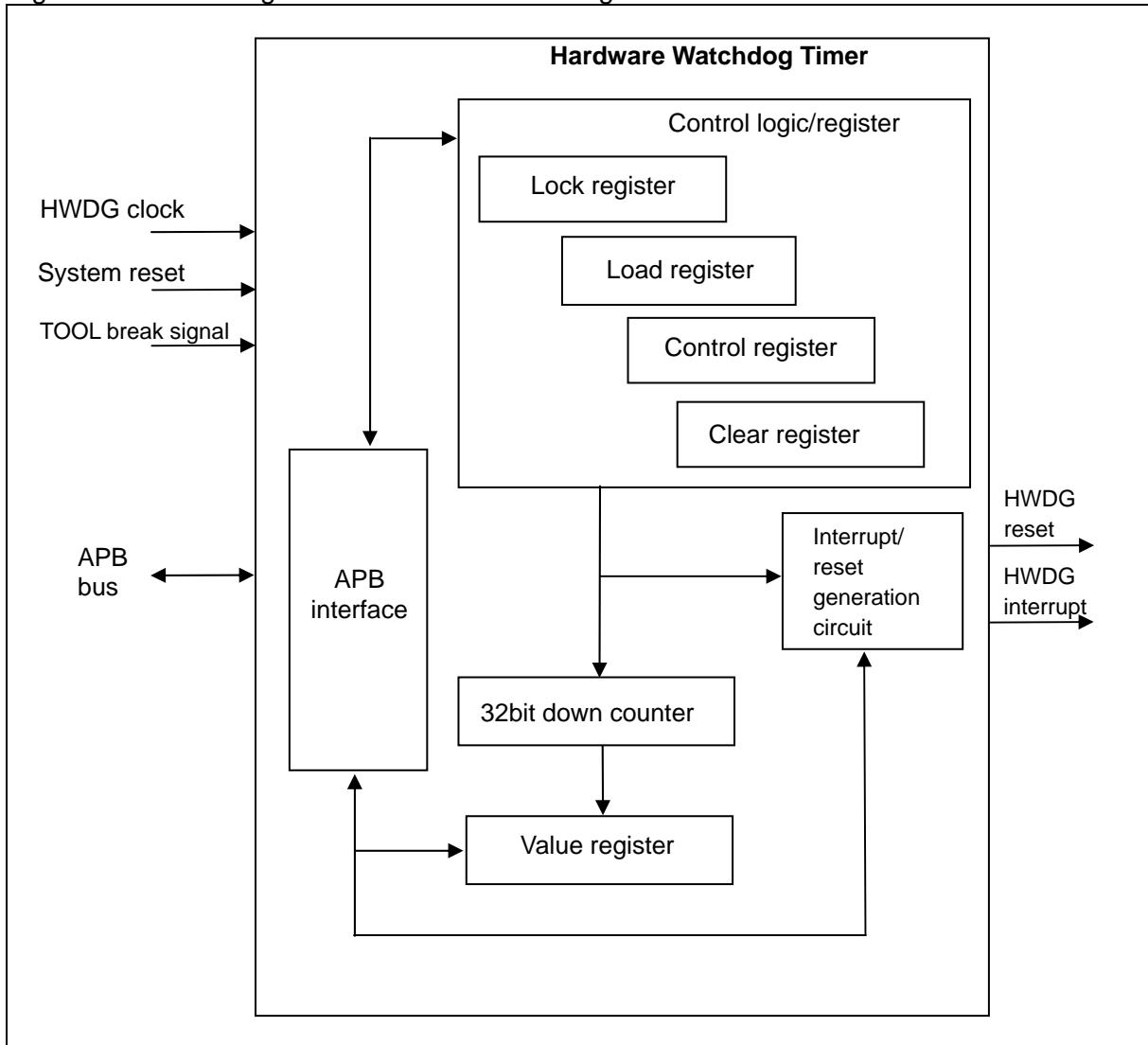


Figure 2-2 Block Diagram of Hardware Watchdog Timer



3. Operations

This section shows the operation of watchdog timer.

The watchdog timer consists of the following blocks.

■ Software Watchdog Timer

● Control register / logic

This circuit controls the software watchdog timer.

It consists of the load register, the lock register, the control register, and the clear register.

- Load register (WdogLoad)

This register is a 32-bit register used to set count interval cycles of the software watchdog timer. The initial value is "0xFFFFFFFF". Table 3-1 shows the examples of interval time setting.

Table 3-1 Examples of Interval Time Setting of Software Watchdog Timer

Count Frequency	Interval Set Value	Interval Time
40MHz	"0xFFFFFFFF" [initial value]	Approx. 107s
20MHz	"0xFFFFFFFF" [initial value]	Approx. 214s
40MHz	"0x0000FFFF"	Approx. 1.6ms
20MHz	"0x0000FFFF"	Approx. 3.2ms

- Lock register (WdogLock)

This register controls accesses of all the registers of the software watchdog timer.

Writing a value of "0x1ACCE551" enables write access to all the other registers of the software watchdog timer.

- Control register (WdogControl)

This register sets an interrupt enable of the software watchdog and a reset enable of the software watchdog.

- Clear register (WdogIntClr)

This is a clear register of the software watchdog timer. Writing any 32-bit value reloads the timer counter from the set value set to the load register, and continues counting.

● Watchdog Timer Counter (32-bit Down Counter)

This is a 32-bit down counter. The count value is reloaded from the set value of the load register (WdogLoad) by accessing to the clear register (WdogIntClr) before the counter value becomes "0" by decrementing.

Table 3-2 shows the down counter reload condition.

Table 3-2 Down Counter Reload Condition of Software Watchdog Timer

Reload Conditions
Accessing to the clear register (WdogIntClr)
When the down counter reaches "0"
When the load register (WdogLoad) is modified
When the watchdog is stopped by writing INTEN=0 to the control register (WdogControl), and reactivated by writing INTEN=1

● Value register (WdogValue)

This register can read the counter value of the watchdog timer.

● Interrupt and reset generation circuit

When an underflow of the watchdog timer counter is detected, a watchdog interrupt and a watchdog reset are generated due to the register setting.

- Interrupt status register (WdogRIS)
This register shows the status of a software watchdog interrupt.

● Activation of software watchdog timer

- Access to the control register (WdogControl), and enable the watchdog reset.
- Table 3-3 shows the combination of watchdog interrupt and watchdog reset settings.

Table 3-3 Combination of software watchdog interrupt and reset

Interrupt	Reset	Operation
Disable	Disable	The watchdog timer is not operated
Enable	Disable	An interrupt is generated underflow
Disable	Enable	The watchdog timer is not operated
Enable	Enable	An interrupt is generated at the first underflow A reset is generated at the second underflow [Initial setting]

Enabling an interrupt of the control register (WdogControl) becomes an activation trigger of the watchdog timer.

● Reload and lock of the register of the software watchdog timer

- The register has not been locked with initial condition after reset. When you wish to enable locking, write any values other than "0x1ACCE551" to the WdogLock register with software.
- When you access the clear register, write "0x1ACCE551" to the WdogLock register to release the lock.
- The value set to the load register (WdogLoad) is reloaded by writing an arbitrary value to the clear register (WdogIntClr).
- After accessed the clear register, it will not be automatically locked. Lock it again with software.

● Halting the software watchdog timer

- The software watchdog timer is stopped by accessing to the control register (WdogControl), and writing "0" to the watchdog interrupt enable bit.
- The software watchdog timer is stopped by generating a reset.

■ Hardware Watchdog Timer

● Control register / logic

This is a circuit to control the hardware watchdog timer.

It consists of the load register, the lock register, the control register, and the clear register.

- Load register (WDG_LDR)

This register is a 32-bit register used to set count interval cycles of the hardware watchdog timer. The initial value is "0x0000FFFF" (down counter for 16 bits=> approx. 655ms @ 100kHz (TYP)). For the frequency of CLKLC which is a count clock, see "data sheet".

- Lock register (WDG_LCK)

This register controls the accesses of all the registers of the hardware watchdog timer. Writing a value of "0x1ACCE551" enables write access to all the registers except the control register (WDG_CTL).

- Control register (WDG_CTL)

This register sets watchdog interrupt enable and watchdog reset enable. To access this register, it is required to write "0x1ACCE551" to the lock register, and also write "0xE5331AAE" to the lock register. In case of not writing the correct value after writing "0x1ACCE551", it is necessary to repeat the process from the beginning.

- Clear register (WDG_ICL)

This is a clear register of the hardware watchdog timer.

Writing any 8-bit value, and also write an arbitrary reversal value reloads the timer counter from the set value set to the load register, and continues counting.

● Watchdog Timer Counter (32-bit down counter)

This is a 32-bit down counter. The count value is reloaded from the set value of the load register (WDG_LDR) by accessing to the clear register (WDG_ICL) before the counter value becomes "0" by decrementing.

Table 3-4 shows the down counter reload condition.

Table 3-4 Down Counter Reload Condition of Hardware Watchdog Timer

Reload Conditions
Accessing to the clear register (WDG_ICL)
When the down counter reaches "0"
When the load register (WDG_LDR)
When the watchdog is stopped by writing INTEN=0 to the control register (WDG_CTL), and reactivated by writing INTEN=1

● Value register (WDG_VLR)

This register can read the counter value of the current watchdog timer. However, during tool break, a correct value can be read when the watchdog timer is stopped. Except during tool break, an inaccurate value may be read due to asynchronous reading. In this case, a countermeasure is necessary such as comparing read values after reading it twice.

● Interrupt and reset generation circuit

When an underflow of the watchdog timer counter is detected, a watchdog interrupt and a watchdog reset are generated due to the register setting.

- Interrupt status register (WDG_RIS)
This register shows the status of a hardware watchdog interrupt.

● Activation of hardware watchdog timer

- Writing "0x1ACCE551" to the lock register (WDG_LCK) and then writing a reversal value "0xE5331AAE" to it enables to access to the control register (WDG_CTL) also.
- Access to the control register (WDG_CTL), and enable the watchdog interrupt and the watchdog reset. Table 3-5 shows the combination of watchdog interrupt and watchdog reset settings.

Table 3-5 Combination of hardware watchdog interrupt and reset

Interrupt	Reset	Operation
Disable	Disable	The watchdog timer is not operated
Enable	Disable	An interrupt is generated underflow
Disable	Enable	The watchdog timer is not operated
Enable	Enable	An interrupt is generated at the first underflow A reset is generated at the second underflow [Initial setting]

Enabling an interrupt of the control register (WDG_CTL) becomes an activation trigger of the hardware watchdog timer.

● Reload and lock of the register of the hardware watchdog timer

The value set to the load register is reloaded by writing an arbitrary value to the clear register (WDG_ICL). After reloading, the register is locked again.
Unlock is required each time accessing to the clear register hereafter.

● Stopping the hardware watchdog timer

- Writing "0x1ACCE551" to the lock register (WDG_LCK) and then writing a reversal value "0xE5331AAE" to it enables to access to the control register (WDG_CTL).
- The hardware watchdog timer is stopped by accessing to the control register (WDG_CTL), and writing "0" to the watchdog interrupt enable bit.

■ Differences between software watchdog timer and hardware watchdog timer

Table 3-6 shows the major differences between software watchdog timer and hardware watchdog timer.

Table 3-6 Differences between software watchdog timer and hardware watchdog timer

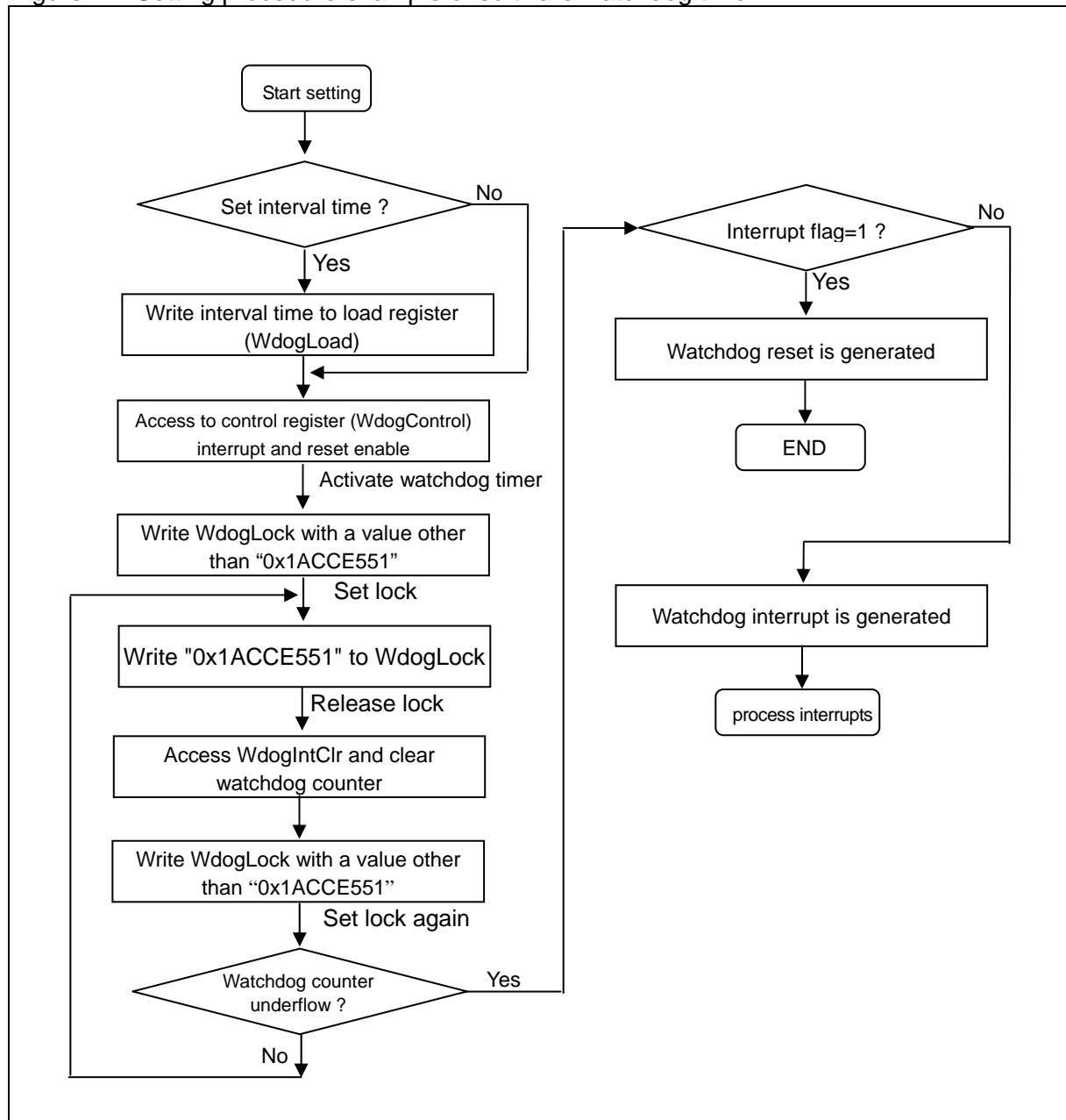
	Software Watchdog	Hardware Watchdog
Count clock	Divided clock of APB	CLKLC
Read value of the value register	Synchronous reading Reading possible	Asynchronous reading Only during tool break, a correct value can be read. Except during tool break, an inaccurate value may be read.
Initial value of watchdog interrupt setting and reset setting	Disable (No watchdog operation)	Enable (With a watchdog operation)
Register lock function initial state	No lock (Software locks after activation)	Lock (Hardware locks from the activation)
Releasing lock	Writing "0x1ACCE551" to lock register to release all lock for the registers	Writing "0x1ACCE551" to lock register to release all lock for the registers except WDG_CTL
Wdog_Control/ WDG_CTL register Releasing separate lock	None	Writing "0xE5331AAE" to lock register to release lock of WDG_CTL register
Relock conditions	Writing a value other than "0x1ACCE551" to the lock register locks all the registers again.	<p>After releasing lock for the registers except WDG_CTL, the lock is resumed under any of the following conditions:</p> <ul style="list-style-type: none"> · Writing a value other than "0x1ACCE551" or "0xE5331AAE" to WDG_LCK · Writing to WDG_LDR · Writing to WDG_CTL · Writing to WDG_ICL again <p>After releasing lock for the registers including WDG_CTL, lock is resumed under any of the following conditions:</p> <ul style="list-style-type: none"> · Writing a value other than "0x1ACCE551" to WDG_LCK · Writing to WDG_LDR · Writing to WDG_ICL · Writing to WDG_CTL
Initial value of load register	0xFFFFFFFF	0x0000FFFF
Bit number of clear register	32 bits	8 bits
Clear register access	Clear by writing an arbitrary value	Clear by writing an arbitrary value, and then writing a reversal value of the arbitrary value

4. Setting Procedure Example

This section explains a setting procedure example of watchdog timer.

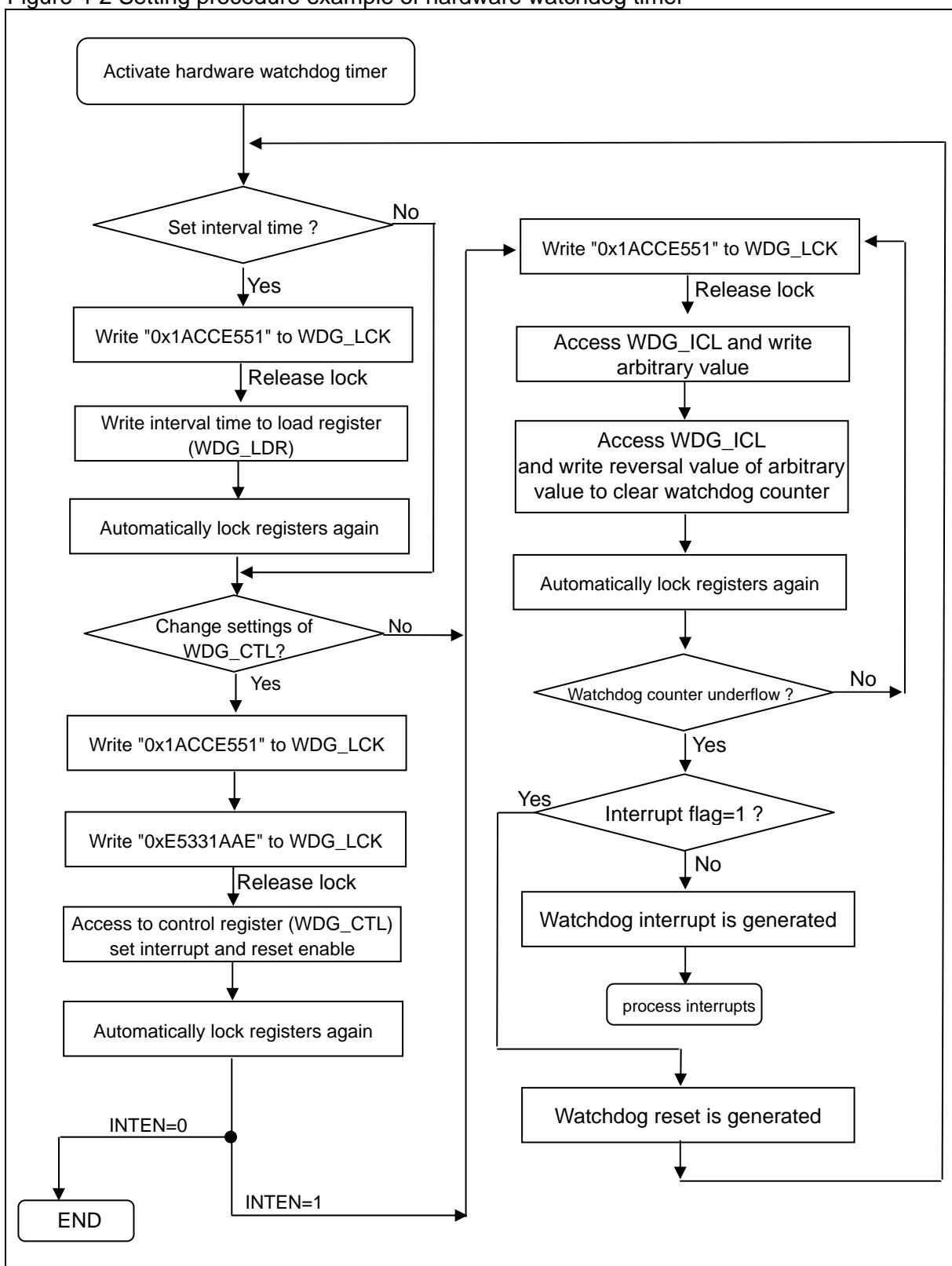
■ Software watchdog timer

Figure 4-1 Setting procedure example of software watchdog timer



■ Hardware watchdog timer

Figure 4-2 Setting procedure example of hardware watchdog timer

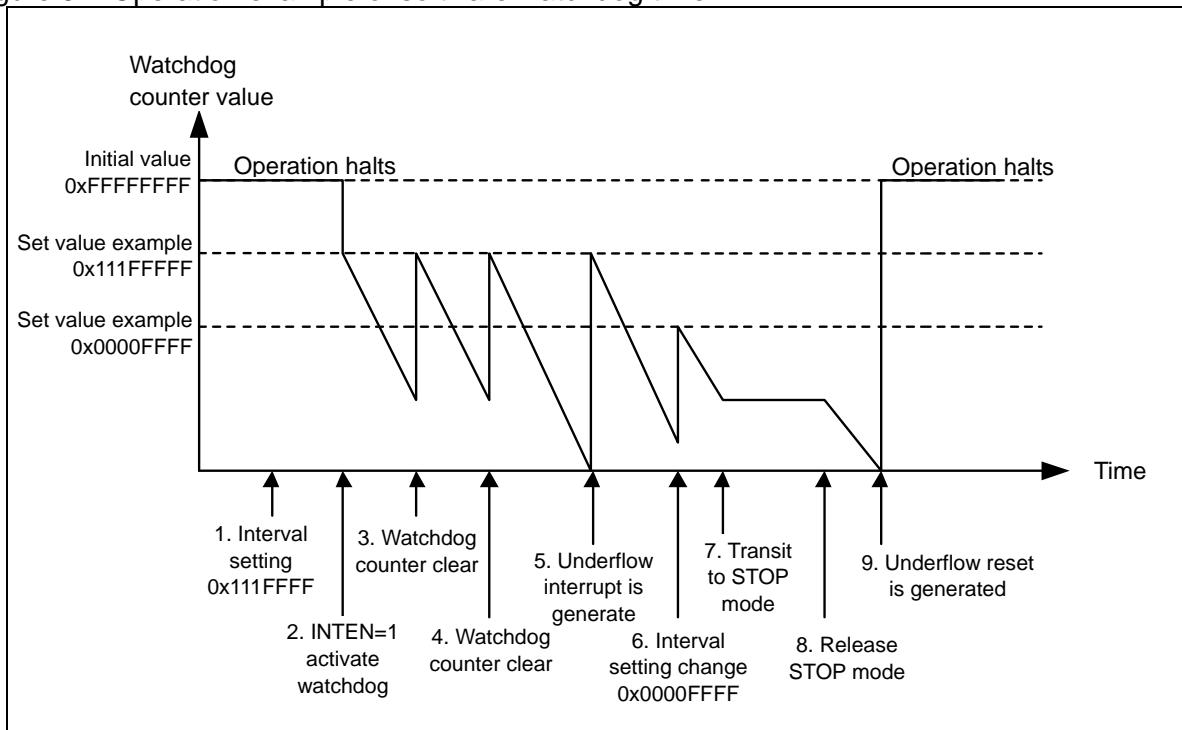


5. Operation Example

This section shows an operation example of the watchdog timers.

■ Software watchdog timer

Figure 5-1 Operation example of software watchdog timer



1. Set SWC_PSR, WDGT_CTL before activation.
Write to WdogLoad register to set interval time.
The interval time is not reflected because the watchdog is not activated. The count value is the initial value.
2. Access to WdogControl register, and write INTEN=1 to activate the watchdog.
At this time, the interval time is reflected and decrementing will be started from the value set in 1.
3. Access to WdogIntClr register, and write an arbitrary value to clear the watchdog counter.
At this time, the set value will be the value set in 2.
4. Access to WdogIntClr register, and write an arbitrary value to clear the watchdog counter.
At this time, the set value will be the value set in 2.
5. Without clearing the counter, an interrupt will be generated underflow.
At this time, the set value will be the value set in 2.
6. Access to WdogLoad register to change interval time.
At this time, the down count value will be cleared to the set value.
7. Transit to STOP mode. The software watchdog will be stopped by this.
8. Release STOP mode. The down counter is restarted. The count value is not cleared.
(Note) Decrementing will be restarted after oscillation wait stabilization is completed, and the base clock starts its operation.

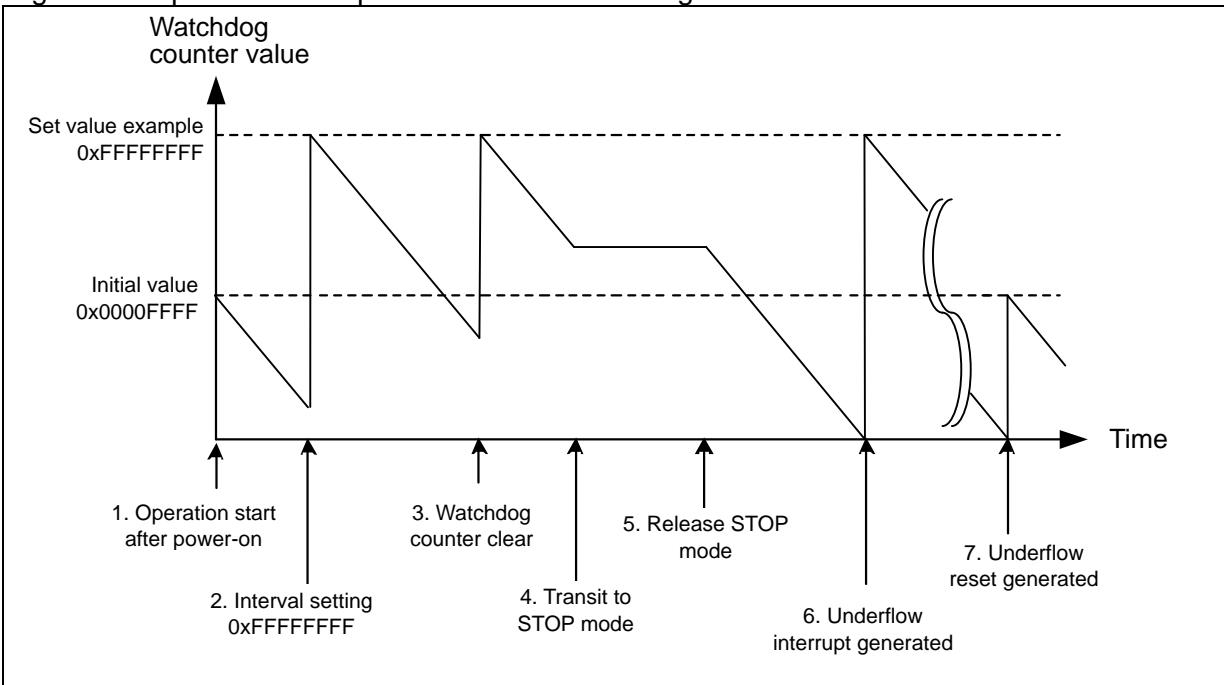
9. A software watchdog reset will be generated when the second underflow is generated without clearing the interrupt WdogIntClr register.
The software watchdog timer stops its operation by generating a reset.

<Note>

Release of the lock register is required to access each register. It is omitted in the operation example.

■ Hardware watchdog timer

Figure 5-2 Operation example of hardware watchdog timer



1. Hardware watchdog timer starts operation after turning on the power.
The initial value of the count value is "0x0000FFFF".
2. Access to WDG_LDR register to change interval time.
At this time, the decremented value will be cleared to the set value.
3. Access to WDG_ICL register, and write an arbitrary value and then write a reversal value of arbitrary value to clear the watchdog counter.
At this time, the set value will be the value set in 2.
4. Transit to STOP mode. The hardware watchdog will be stopped by this.
5. Release STOP mode. The down counter is restarted. The count value is not cleared.
(Note) Decrementing will be restarted after oscillation wait stabilization is completed, and the base clock starts its operation.
6. Without clearing the counter, an interrupt will be generated underflow.
At this time, the set value will be the value set in 2.
7. A hardware watchdog reset will be generated when the second underflow is generated without clearing the interrupt WDG_IC register.
The count value returns to the initial value and decrementing is started.

<Note>

Release of the lock register is required to access each register. It is not mentioned in the operation example.

6. Registers

This section explains the registers of clock generation.

Table 6-1 List of registers for the watchdog timer

Register Name	Explanation	Reference
WdogLoad	Software watchdog timer load register	6.1
WdogValue	Software watchdog timer value register	6.2
WdogControl	Software watchdog timer control register	6.3
WdogIntClr	Software watchdog timer clear register	6.4
WdogRIS	Software watchdog timer interrupt status register	6.5
WdogLock	Software watchdog timer lock register	6.6
WDG_LDR	Hardware watchdog timer load register	6.7
WDG_VLR	Hardware watchdog timer value register	6.8
WDG_CTL	Hardware watchdog timer control register	6.9
WDG_ICL	Hardware watchdog timer clear register	6.10
WDG_RIS	Hardware watchdog timer interrupt status register	6.11
WDG_LCK	Hardware watchdog timer lock register	6.12

6.1. Software Watchdog Timer Load Register (WdogLoad)

WdogLoad register sets the cycle of the software watchdog timer.

■ Register configuration

bit	31	0
Field	WdogLoad	
Attribute	R/W	

Initial value 0xFFFFFFFF

■ Register function

[bit31:0] WdogLoad : Interval cycle setting bit

bit31:0	Explanation
In case of writing	Sets the cycle of the software watchdog. The initial value is "0xFFFFFFFF". The minimum value for writing is "1". When "0" is written, an interrupt will be generated. (A reset may be generated by setting.)
In case of reading	A set value can be read. The initial value "0xFFFFFFFF" is read.

<Notes>

- During watchdog timer operation, if the value of WdogLoad is modified, the value of WdogLoad will be reflected to the timer counter, and counting is continued.
- During the watchdog timer is halting, if the value of WdogLoad is modified, the value of WdogLoad will be reflected to the timer counter at activation.

6.2. Software Watchdog Timer Value Register (WdogValue)

WdogValue register can read the current value of the software watchdog timer.

■ Register configuration

bit	31	0
Field	WdogValue	
Attribute	R	
Initial value	0xFFFFFFFF	

■ Register function

[bit31:0] WdogValue : Counter value bit

bit31:0	Explanation
In case of writing	No effect.
In case of reading	The count value of the current watchdog counter is read. The initial value "0xFFFFFFFF" is read if reading before activation.

<Note>

See "Debug Break Watchdog Timer Control Register" in the chapter of "Clock" for the setting of watchdog timer at tool break.

6.3. Software Watchdog Timer Control Register (WdogControl)

WdogControl register sets enable/disable of the software watchdog timer.

■ Register configuration

bit	7	2	1	0
Field	Reserved		RESEN	INTEN
Attribute	-		R/W	R/W
Initial value	-		1'b0	1'b0

■ Register function

[bit7:2] res : Reserved bits

"0b000000" is read from these bits.

In case of writing, set "0b000000".

[bit1] RESEN : Reset enable bit of the software watchdog

bit	Explanation
In case of reading	Register value is read.
In case of writing 0	A watchdog reset is disabled.
In case of writing 1	A watchdog reset is enabled.

[bit0] INTEN : Interrupt and counter enable bit of the software watchdog

bit	Explanation
In case of reading	Register value is read.
In case of writing 0	A watchdog interrupt is disabled. A watchdog counter is disabled.
In case of writing 1	A watchdog interrupt is enabled. A watchdog counter is enabled.

<Notes>

- Writing "1" to the watchdog counter loads an interval cycle value from WdgLoad and the software watchdog timer is activated.
- Writing "0" to INTEN stops the watchdog counter. The watchdog counter reloads the cycle value from WdgLoad when "1" is written again and reactivated.
- The watchdog timer can be activated by enabling INTEN only. The watchdog timer is not activated by enabling RESEN only. To activate the watchdog timer, INTEN should be enabled. See "3. Operations" for more details.
- Writing "0" to INTEN clears the interrupt flag in software watchdog interrupt status register (WdogRIS).

6.4. Software Watchdog Timer Clear Register (WdogIntClr)

WdogIntClr register clears the software watchdog timer.

■ Register configuration

bit	31	0
Field		WdogIntClr
Attribute		R/W
Initial value		0xxxxxxxxx

■ Register function

[bit31:0] WdogIntClr : clear bit

bit31:0	Explanation
In case of reading	An undefined bit is read.
In case of writing	<p>Writing an arbitrary value</p> <ul style="list-style-type: none">Clears an interrupt of the watchdog timer, if an interrupt of the watchdog timer is generated.Reloads the set value from WdogLoad register to the watchdog timer counter.

6.5. Software Watchdog Timer Interrupt Status Register (WdogRIS)

WdogRIS register shows the status of the software watchdog timer.

■ Register configuration

bit	7	1	0
Field		Reserved	RIS
Attribute	-	-	R
Initial value	-	-	1'b0

■ Register function

[bit7:1] res : Reserved bits

"0b0000000" is read from these registers.

In case of writing, set "0b0000000".

[bit0] RIS : Software watchdog interrupt status bit

bit	Explanation
In case of writing	No effect.
In case of reading 0	A watchdog interrupt is not generated.
In case of reading 1	A watchdog interrupt is generated.

6.6. Software Watchdog Timer Lock Register (WdogLock)

WdogLock register controls accesses of all the registers of software watchdog timer.

■ Register configuration

bit	31	0
Field	WdogLock	
Attribute	R/W	
Initial value	0x00000000	

■ Register function

[bit31:0] WdogLock : Software watchdog lock register

bit31:0	Explanation
In case of writing	Writing "0x1ACCE551": Releases locks of all the registers of software watchdog timer. Writing other than "0x1ACCE551": Lock function for all of the software watchdog timer registers will be enabled.
In case of reading	"0x00000000" : The locks are released. "0x00000001" : The locks are not released.

<Notes>

- Lock for initial values are not enabled. Enable lock function after the software watchdog timer is started.
- After lock is released, the clear register (WdogIntClr) will become accessible.
- After accessed the clear register (WdogIntClr), lock will not be automatically enabled. Incorporate "lock release -> clear -> lock enable" for any clear sequence.
- If locks are not released, reading is enabled and the values in each register can be read by accessing each register of the software watchdog. Writing is disabled.

6.7. Hardware Watchdog Timer Load Register (WDG_LDR)

WDG_LDR register sets the cycle of hardware watchdog timer.

■ Register configuration

bit	31	0
Field	WDG_LDR	
Attribute	R/W	
Initial value	0x0000FFFF	

■ Register function

[bit31:0] WDG_LDR : Interval cycle setting bit

bit31:0	Explanation
In case of writing	Sets cycle of the hardware watchdog. The initial value is "0x0000FFFF". The minimum value of writing is "1". An interrupt is generated after "0" is written.
In case of reading	A set value can be read. The initial value "0x0000FFFF" is read.

<Notes>

- During watchdog timer operation, if the value of WDG_LDR is modified, the value of WDG_LDR will be reflected to the timer counter and counting is continued.
- During the watchdog timer is halting, if the value of WDG_LDR is modified, the value of WDG_LDR will be reflected to the timer counter at activation of the watchdog timer.
- The case of modifying the WDG_LDR register when the watchdog timer interrupt was occurred, the watchdog timer interrupt is cleared.
- This register can not be cleared by a software reset or a software watchdog reset.

6.8. Hardware Watchdog Timer Value Register (WDG_VLR)

WDG_VLR register can read the current counter value of the hardware watchdog timer.

■ Register configuration

bit	31	0
Field		WDG_VLR
Attribute		R
Initial value		0xxxxxxxx

■ Register function

[bit31:0] WDG_VLR : Counter value bit

bit31:0	Explanation
In case of reading	The current count value of the watchdog counter can be read. By turning on the power, the hardware watchdog automatically activates, therefore decrementing is already started at the time of reading. The value after power on or the value decremented from the initial value "0x0000FFFF" is read.
In case of writing	No effect.

<Notes>

- This register can not be cleared by software reset or software watchdog reset.
- Reading a correct value of this register is possible only if the watchdog timer stops at tool break. See "Debug Break Watchdog Timer Control Register" in the chapter of "Clock" for the setting of watchdog timer at tool break. Except during tool break, an inaccurate value may be read due to asynchronous reading for the bus clock. In this case, a countermeasure is necessary such as comparing read values after reading it twice.

6.9. Hardware Watchdog Timer Control Register (WDG_CTL)

WDG_CTL register sets enable/disable of the hardware watchdog timer.

■ Register configuration

bit	7	2	1	0
Field	Reserved		RESEN	INTEN
Attribute	-		R/W	R/W
Initial value	-		1'b1	1'b1

■ Register function

[bit7:2] res : Reserved bits

"0b000000" is read from these bits.

In case of writing, set these bits to "0b000000".

[bit1] RESEN : Hardware watchdog timer reset enable bit

bit	Explanation
In case of reading	A value of register is read.
In case of writing 0	A watchdog reset is disabled.
In case of writing 1	A watchdog reset is enabled.

[bit0] INTEN : Hardware watchdog interrupt and counter enable bit

bit	Explanation
In case of reading	The value of the register is read.
In case of writing 0	A watchdog interrupt is disabled. A watchdog counter is disabled.
In case of writing 1	A watchdog interrupt is enabled. A watchdog counter is enabled.

<Notes>

- Writing "0" to INTEN stops the watchdog counter. When writing "1" again, the watchdog counter reloads the cycle value from WDG_LDR, and continues counting.
- The watchdog timer can be activated by enabling INTEN only. The watchdog timer is not activated by enabling RESEN only. To activate the watchdog timer, INTEN should be enabled.
- To access this register, it is required to write "0x1ACCE51" to the lock register, and also write the reversal value "0xE531AAE" to release lock.
- This register cannot be cleared by a software reset or a software watchdog reset
- Writing "0" to INTEN clears the interrupt flag in hardware watchdog interrupt status register (WDG_RIS).

6.10. Hardware Watchdog Timer Clear Register (WDG_ICL)

WDG_ICL register clears the hardware watchdog timer.

■ Register configuration

bit	7	0
Field	WDG_ICL	
Attribute	R/W	
Initial value	0xxx	

■ Register function

[bit7:0] WDG_ICL : clear bit

bit7:0	Explanation
In case of reading	Undefined value is read.
In case of writing	Writing an arbitrary 8-bit value, and then write a reversal value of the arbitrary value, · Clears an interrupt, if an interrupt of watchdog timer is generated. · Reloads the set value from WDG_LDR register to the watchdog timer counter.

<Note>

This register cannot be cleared by a software reset or a software watchdog reset.

6.11. Hardware Watchdog Timer Interrupt Status Register (WDG_RIS)

WDG_RIS register shows the status of the hardware watchdog timer.

■ Register configuration

bit	7	1	0
Field		Reserved	RIS
Attribute	-	-	R
Initial value	-	-	1'b0

■ Register function

[bit7:1] res : Reserved bits

"0b0000000" is read from these bits.

In case of writing, set "0b0000000".

[bit0] RIS : Hardware watchdog interrupt status bit

bit	Explanation
In case of writing	No effect
In case of reading 0	Hardware watchdog interrupt is not generated.
In case of reading 1	Hardware watchdog interrupt is generated.

<Note>

This register cannot be cleared by a software reset or a software watchdog reset.

6.12. Hardware Watchdog Timer Lock Register (WDG_LCK)

WDG_LCK register controls all the registers of the hardware watchdog timer.

■ Register configuration

bit	31	0
Field	WDG_LCK	
Attribute	R/W	
Initial value	0x00000001	

■ Register function

[bit31:0] WDG_LCK : Hardware watchdog lock register

bit31:0	Explanation
In case of writing	In case of writing "0x1ACCE551": The locks of all the registers other than the control register are released. Later, in case of writing the reversal value, "0xE5331AAE": The locks of all the registers are released. In case of other procedure is performed or writing any value other than above: No effect.
In case of reading	"0x00000000" : The locks are released. "0x00000001" : The locks are not released.

<Notes>

- This register cannot be cleared by a software reset or a software watchdog reset.
- In case of accessing to each register of the hardware watchdog when the locks are not released, reading is enabled and the values of each register can be read. Writing is ignored.

7. Notes

The section explains the notes when using the watchdog timer.

- Hardware watchdog timer clear register
To clear the hardware watchdog, write an arbitrary 8-bit value, and then write a reversal value of the arbitrary value. Clearing cannot be performed unless the correct reversal value is written. Even if clearing is not performed, the register is locked again.
- Cooperation with a debug tool
When a tool break is applied by a debug tool, to continue or stop of the counter of the watchdog timer can be set by setting of the register. See the chapter, "Clock" for details about the behavior of the watchdog timers during debugging.
- Operation at standby mode
Writing to a key register is required at setting of the standby mode to not to stop the watchdog timer for the case of the mode is transited to the standby mode because of an unintended program operation. See the chapter, "Low-power Consumption Mode" for more details.
- Generation of a watchdog reset can be confirmed by the reset source register. See the section for Reset Source Register in the chapter, "Reset" for more details.
- See the section for Interrupt Source Register in the chapter "Interrupt" for an interrupt source.
- Use a divided clock of APB clock for the count clock of the software watchdog.
See the chapter, "Clock" for divided clock setting of the count clock.
- Hardware watchdog and interrupt handler
Before releasing the Lock for WDG_CTL (after releasing the Lock for the register other than WDG_CTL), if an another interrupt becomes effective by the hardware watchdog and the interrupt handler begins its processing, the Lock releasing count could not be detected by hardware. So, at the beginning of the interrupt handler, write values in the WDG_LCK register to lock the register.

CHAPTER: Dual Timer

This chapter introduces the Dual Timer functions and operations.

1. Overview
2. Architecture
3. Operation Description
4. Setting Procedure Example
5. Register

1. Overview

Dual Timer consists of two programmable 32/16-bit down counters. An interrupt is generated when the count reaches zero.

■ Dual Timer Overview

Dual Timer consists of two programmable Free Running Counters. Each timer block operates identically. The Free Running Counters can be programmed for 32-bit or 16-bit counter size by Control Register. Also, any one of the following three timer modes can be selected:

- Free-running mode
The counter operates continuously and wraps around to its maximum value each time that it reaches zero.
- Periodic mode
The counter is reloaded from Load Register and operates continuously each time that it reaches zero.
- One-shot mode
Writing to the Load Register loads the counter with a new value. The counter halts until it is reprogrammed when the counter reaches zero.

Two Free Running Counters operate in common timer clock (TIMCLK). APB bus clock (PCLK) is used as the timer clock. Also, each Free Running Counter has a prescaler that can divide by 1, 16, or 256. Therefore, the count rate of each Free Running Counter can be controlled by each prescaler.

Writing to the Load Register loads the counter with the timer count value. If the timer counter is enabled, the timer decrements at the rate determined in the timer clock and in the prescaler setting. When the timer counter has been running, writing to the Load Register restarts the counter immediately with a new value.

An alternative way of loading the timer count is to write to Background Load Register. In this way, the current count value is not affected immediately after the writing, and the counter continues to decrement. Then, in the case where the counter reaches zero, the timer counter is reloaded with a new load value if it is in Periodic Mode.

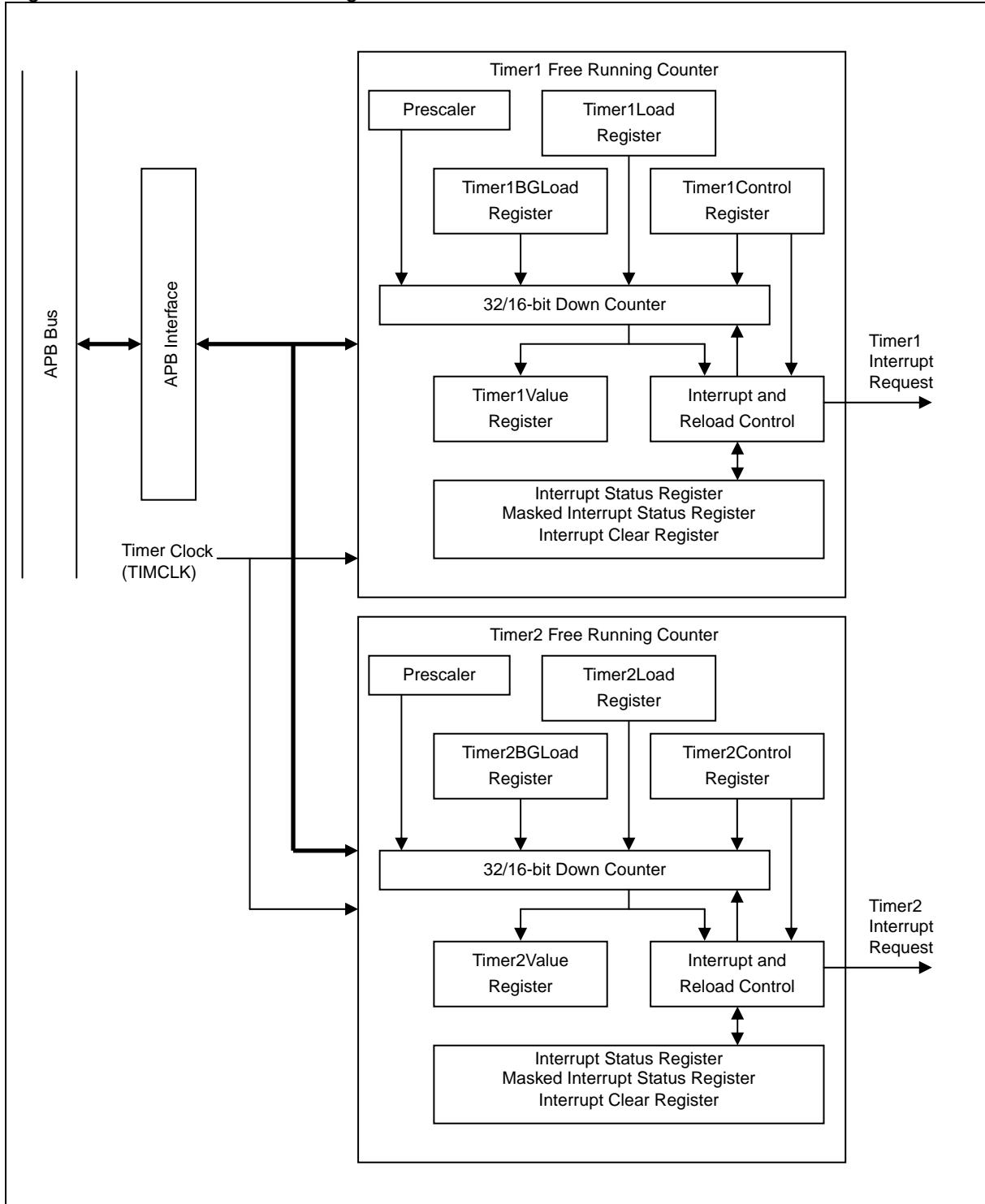
When the timer count reaches zero, an interrupt is generated. Writing to Interrupt Clear Register clears the interrupt. Also, the interrupt output signal can be masked by Interrupt Mask Register.

The current counter value can be read from Value Register at any time.

2. Architecture

This chapter illustrates the Dual Timer architecture.

Figure 2-1 Dual Timer Block Diagram



3. Operation Description

This chapter describes Dual Timer operations.

3.1 Timer Operating Mode

3.2 Default

3.3 Interrupt Behavior

3.1. Timer Operating Mode

Operating modes are selected from three timer modes based on the settings of the Control Register (TimerXControl)'s mode bit (TimerMode) and one-shot mode bit (OneShot).

Table 3-1 Mode Selection Table

TimerMode	OneShot	Selective Mode
0	0	Free-running Mode
1	0	Periodic Mode
-	1	One-shot Mode

Timer size bit (TimerSize) of the Control Register is used to appropriately configure 32-bit or 16-bit counter operation.

<Note>

The character "X" in a register name in this chapter indicates either register of Free Running Counter 1 or 2.

■ Free-running Mode

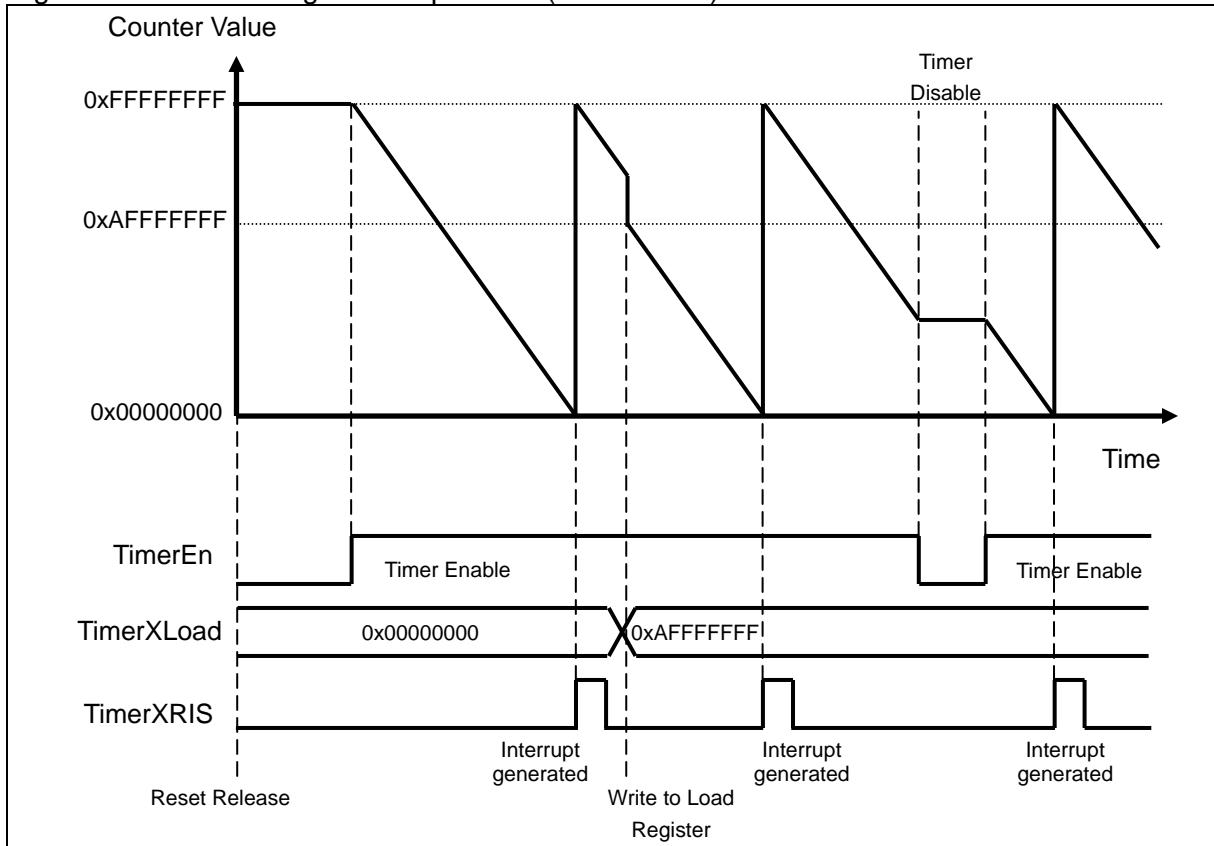
When a reset is performed, the timer value is initialized to 0xFFFFFFFF. Then, if the counter is enabled, the count decrements by one at the timer clock (TIMCLK) rising edge. Alternatively, writing to the Load Register (TimerXLoad) loads a new initial counter value. Then, if the counter is enabled, the counter starts to decrement from this loaded value.

In 32-bit mode, when the count reaches zero (0x00000000), an interrupt is generated. Then, regardless of the Load Register's value, the counter wraps around to 0xFFFFFFFF. The counter starts to decrement again, and as long as the counter is enabled, this whole cycle is repeated.

In 16-bit mode, only the least significant 16 bits of the counter are decremented. When the count reaches 0x0000, an interrupt is generated. Then, regardless of the Load Register's value, the counter wraps around to 0xFFFF. If the counter is enabled again, the counter continues to decrement from the current value.

The counter value can be read from the Value Register (TimerXValue) at any time.

Figure 3-1 Free-running Mode Operation (32-bit Mode)



■ Periodic Mode

Writing to the Load Register (TimerXLoad) loads an initial counter value. Then, the counter starts to decrement from this value if the counter is enabled.

In 32-bit mode, all 32 bits of the counter are decremented. Then, when the count reaches zero (0x00000000), an interrupt is generated. The counter reloads the Load Register value. The counter starts to decrement again. As long as the counter is enabled, this whole cycle is repeated.

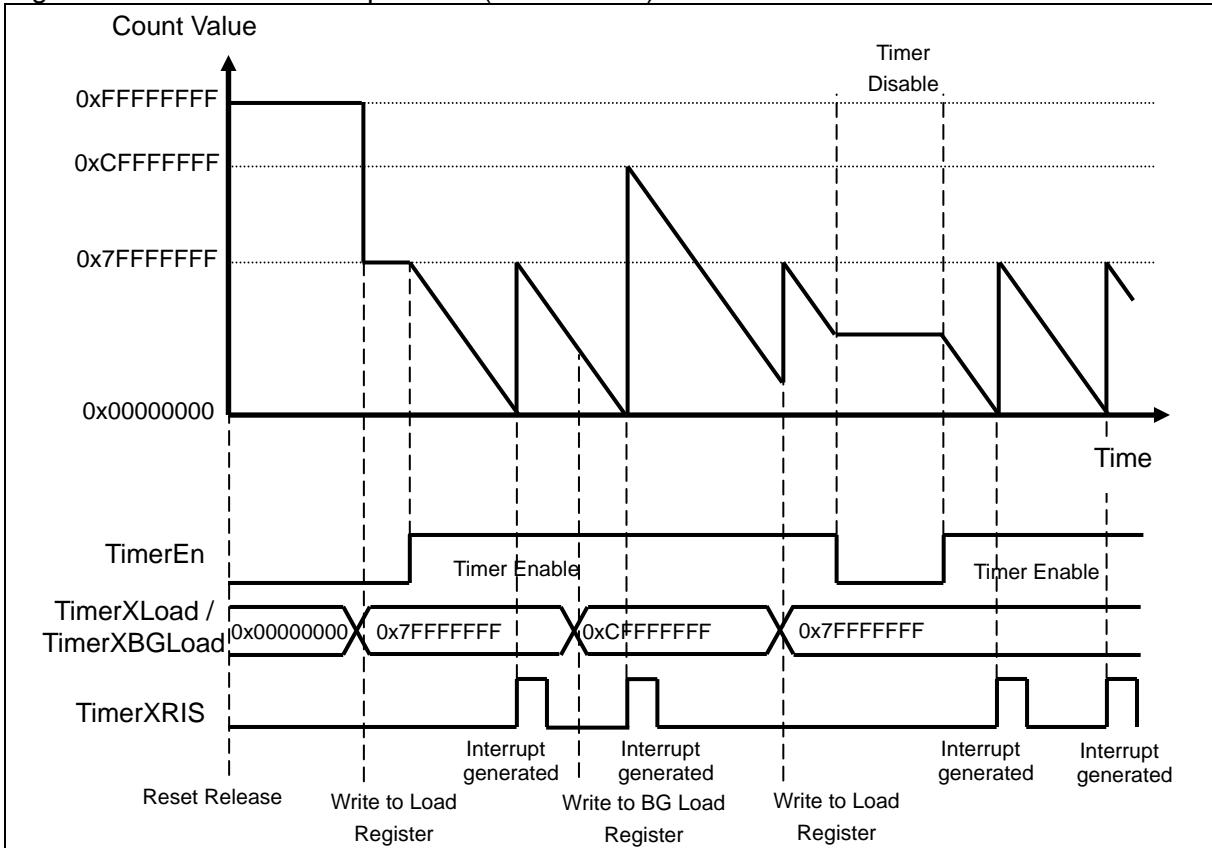
In 16-bit mode, only the least significant 16 bits of the counter are decremented. When the count reaches 0x0000, an interrupt is generated. Then, the counter reloads the Load Register value. The counter starts to decrement again. As long as the counter is enabled, this whole cycle is repeated.

When a new value is written to the Background Load Register (TimerXBGLoad) while the counter is running, the value of the Load Register is also updated to the same value. However, the counter continues to decrement to zero. When the counter reaches zero, it reloads the new value. As long as the Timer is set to Periodic Mode, this new load value is used for each subsequent reload.

When a new value is written to the Load Register for loading the value to the counter while the counter is running, the counter value is changed to the new load value at the next timer clock.

If the Enable bit (TimerEn) of the Control Register (TimerXControl) is cleared and that the counter is disabled, the counter halts and holds the current value. If the counter is enabled again, the counter continues to decrement from the current value.

Figure 3-2 Periodic Mode Operation (32-bit Mode)



■ One-shot Mode

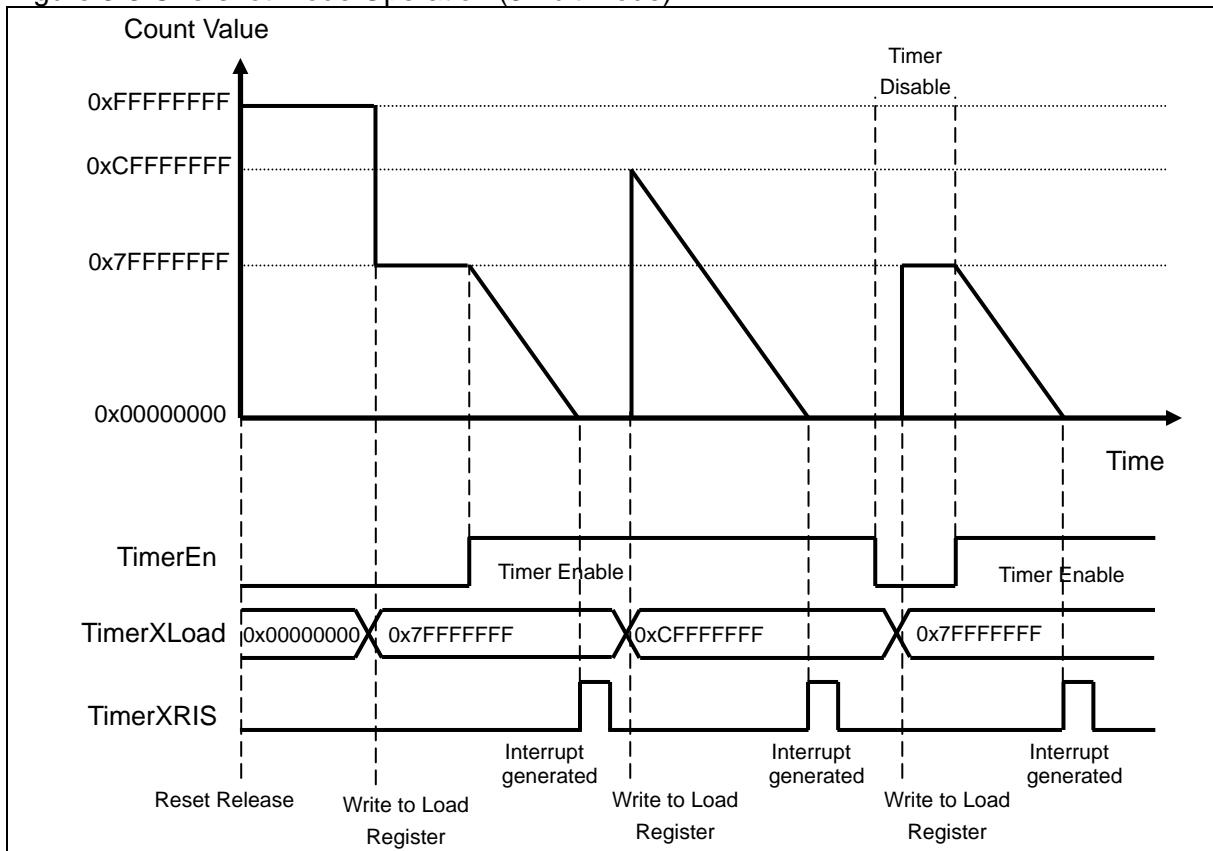
To start the count down sequence in One-shot Mode, a new load value is written to the Load Register (TimerXLoad). If the counter is enabled, it starts to decrement from this value.

In 32-bit mode, all 32 bits of the counter are decremented. Then, when the count reaches zero (0x00000000), an interrupt is generated. Then, the counter halts.

In 16-bit mode, only the least significant 16 bits of the counter are decremented. When the count reaches 0x0000, an interrupt is generated. Then, the counter halts.

In One-shot Mode, writing a new value to the Load Register starts the counter again. Then, the counter value is changed to the new load value at the next timer clock.

Figure 3-3 One-shot Mode Operation (32-bit Mode)



3.2. Default

After the reset, the timer is initialized as shown below:

- Timer counter disabled
- Free-running mode selected
- 16-bit counter mode selected
- Prescaler in the setting of dividing by 1
- Interrupt clear and interrupt enable states
- Load Register set to zero
- Counter value set to 0xFFFFFFFF

3.3. Interrupt Behavior

This section describes interrupt behaviors.

An interrupt is generated when the counter reaches 0x00000000 (in 32-bit mode) or 0xFFFF0000 (in 16-bit mode) in the setting of interrupt enable (IntEnable=1). In 16-bit mode, the most significant 16 bits of the counter are ignored.

Writing to Interrupt Clear Register (TimerXIntClr) clears an interrupt.

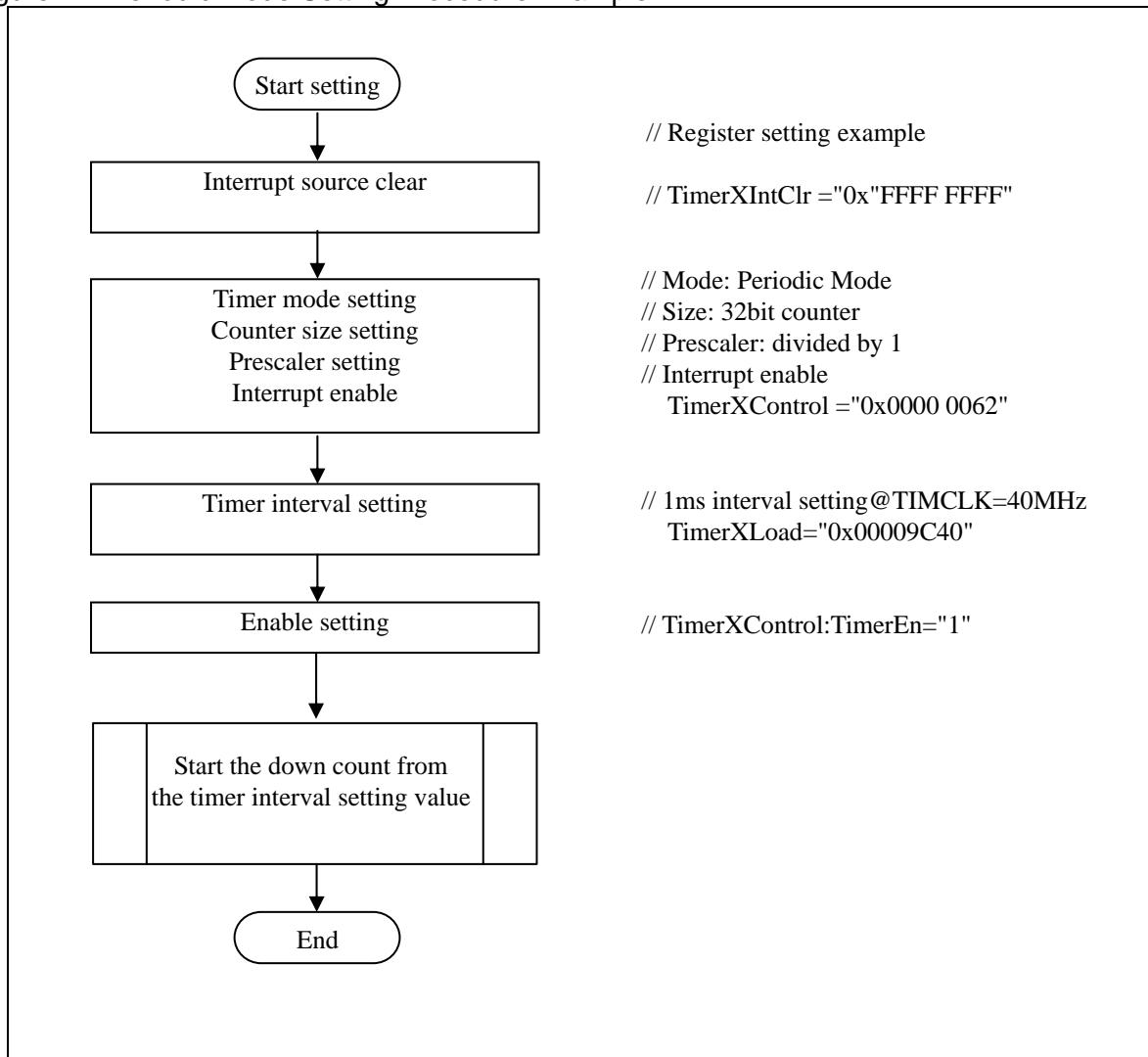
The interrupt signals generated in the Timer module can be masked when Interrupt Enable bit (IntEnable) of the Control Register is set to 0. The raw interrupt state before being masked can be read from Interrupt Status Register (TimerXRIS). Also, the masked interrupt state can be read from Masked Interrupt Status Register (TimerXMIS).

4. Setting Procedure Example

This chapter describes an example of the Dual Timer setting procedure.

■ Dual Timer Setting Procedure Flow

Figure 4-1 Periodic Mode Setting Procedure Example



■ Timer Interval Setting

Expressions of the timer interval calculations in respective modes are shown in Table 4-1:

Table 4-1 Expression for Timer Interval Calculation

Mode	Timer Interval
32-bit Free-running	$(\text{PREScale}_{\text{DIV}} / \text{TIMCLK}_{\text{FREQ}}) \times 2^{32}$
16-bit Free-running	$(\text{PREScale}_{\text{DIV}} / \text{TIMCLK}_{\text{FREQ}}) \times 2^{16}$
Periodic & One-shot	$(\text{PREScale}_{\text{DIV}} / \text{TIMCLK}_{\text{FREQ}}) \times \text{TimerXLoad}$

- $\text{TIMCLK}_{\text{FREQ}}$ is the timer clock (TIMCLK) frequency.
- $\text{PREScale}_{\text{DIV}}$ is the prescaler division factor of 1, 16, or 256 configured by bit [3:2] of the Control Register (TimerXControl).
- TimerXLoad is the value of the Load Register (TimerXLoad).

For example, in the case of $\text{TIMCLK}=40\text{MHz}$ and $\text{PREScale}_{\text{DIV}}=1$, the value of the Load Register (TimerXLoad) to configure 1ms timer interval can be calculated as follows:

$$\begin{aligned}\text{TimerXLoad} &= \text{Timer interval} \times \text{TIMCLK}_{\text{FREQ}} / \text{PREScale}_{\text{DIV}} \\ &= 1\text{ms} \times 40\text{MHz} / 1 = 4 \times 10^4 = 0x00009C40\end{aligned}$$

<Note>

The minimum valid value of the Load Register (TimerXLoad) is "1". If the Load Register is set to "0", an interrupt will be immediately generated.

5. Register

This chapter describes the structures and functions of the registers used in Dual Timer.

■ Dual Timer Register List

Abbreviation	Register Name	See
Timer1Load	Timer1 Load Register	5.1
Timer1Value	Timer1 Value Register	5.2
Timer1Control	Timer1 Control Register	5.3
Timer1IntClr	Timer1 Interrupt Clear Register	5.4
Timer1RIS	Timer1 Interrupt Status Register	5.5
Timer1MIS	Timer1 Masked Interrupt Status Register	5.6
Timer1BGLoad	Timer1 Background Load Register	5.7
Timer2Load	Timer2 Load Register	5.1
Timer2Value	Timer2 Value Register	5.2
Timer2Control	Timer2 Control Register	5.3
Timer2IntClr	Timer2 Interrupt Clear Register	5.4
Timer2RIS	Timer2 Interrupt Status Register	5.5
Timer2MIS	Timer2 Masked Interrupt Status Register	5.6
Timer2BGLoad	Timer2 Background Load Register	5.7

5.1. Load Register (TimerXLoad) X=1 or 2

Load Register (TimerXLoad) has a start value to decrement the counter in 32-bit Register.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TimerXLoad[31:16]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TimerXLoad[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:0] TimerXLoad : Timer X Load bit

When a value is directly written to this register, the current count is immediately set to a new value at the next timer clock. Also, in Periodic Mode setting, this value is used for reloading the counter when the current count reaches zero.

In addition, the value in this register is also overwritten when the Background Register (TimerXBGLoad) is written. However, in this case, the current count is not immediately affected.

After either the Load Register or the Background Register is written, the register value written last is returned at any reading. In other words, the same value is read from both of the Load Register and the Background Register, and the value is always reloaded after the counter reaches zero in Periodic Mode.

<Note>

The minimum valid value of the Load Register (TimerXLoad) is "1". If the Load Register is set to "0", an interrupt will be immediately generated.

5.2. Value Register (TimerXValue) X=1 or 2

Value Register (TimerXValue) indicates the current value of the decrement counter in 32-bit Read Only Register.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TimerXValue[31:16]															
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TimerXValue[15:0]															
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[bit31:0] TimerXValue : Timer X Value bit

After a load operation which a new load value is written to the Load Register (TimerXLoad), the new load value is reflected immediately to this Value Register (TimerXValue).

<Note>

In 16-bit timer mode, the most significant 16 bits of 32-bit Value Register (TimerXValue) are not automatically set to "0". For example, when no writing to the Load Register (TimerXLoad) has occurred yet since the change in the Timer from 32-bit mode to 16-bit mode, the most significant 16 bits of the Value Register have non-zero values.

5.3. Control Register (TimerXControl) X=1 or 2

Control Register (TimerXControl) controls the Timer.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								Timer En	Timer Mode	Int Enable	Reserved	TimerPre	Timer Size	One Shot	R/W
Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X	0	0	1	0	0	0	0	0

[bit31:8] Reserved : Reserved bit

Writing	No effect on operation
Reading	Value not determined

[bit7] TimerEn : Enable bit

bit	Description
0	Timer disabled [Initial value]
1	Timer enabled

[bit6] TimerMode : Mode bit

bit	Description
0	Free-running Mode [Initial value]
1	Periodic Mode

[bit5] IntEnable : Interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled [Initial value]

[bit4] Reserved : Reserved bit

Writing	No effect on operation
Reading	Value not determined

[bit3:2] TimerPre : Prescale bit

bit3	bit2	Description
0	0	Clock divided by 1 [Initial value]
0	1	Clock divided by 16
1	0	Clock divided by 256
1	1	Undefined, do not use

[bit1] TimerSize : Counter size bit

Select 16/32-bit counter operation.

bit	Description
0	16-bit counter [Initial value]
1	32-bit counter

[bit0] OneShot : One-shot mode bit

Select One-shot Mode or Counter Wrapping Mode (Free-running Mode/Periodic Mode). Based on Mode bit (TimerMode) settings, Free-running Mode or Periodic Mode is selected.

bit	Description
0	Wrapping Mode (Free-running Mode/Periodic Mode) [Initial value]
1	One-shot Mode

<Note>

The counter mode, size, or prescale settings must not be changed while the Timer is running. To configure a new setting, the Timer needs to be disabled first and that a new setting value needs to be written to respective registers. Then, after the setting is changed, the Timer needs to be enabled again. Failure to follow this setting procedure can result in unpredictable behaviors of the device.

5.4. Interrupt Clear Register (TimerXIntClr) X=1 or 2

Interrupt Clear Register (TimerXIntClr) clears an interrupt.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TimerXIntClr[31:16]															
Attribute	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TimerXIntClr[15:0]															
Attribute	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit31:0] TimerXIntClr : Interrupt clear bit

Writing any value to this register clears an interrupt output from the counter.

5.5. Interrupt Status Register (TimerXRIS) X=1 or 2

Interrupt Status Register (TimerXRIS) indicates an unmasked and raw interrupt status.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															

Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															

Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

[bit31:1] Reserved : Reserved bit

Writing	No effect on operation
Reading	Value not determined

[bit0] TimerXRIS :Interrupt Status Register bit

bit	Description
0	No interrupt generated from the counter [Initial value]
1	Interrupt generated from the counter

5.6. Masked Interrupt Status Register (TimerXMIS) X=1 or 2

Masked Interrupt Status Register (TimerXMIS) indicates the masked interrupt status.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															

Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															

Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

[bit31:1] Reserved : Reserved bit

Writing	No effect on operation
Reading	Value not determined

[bit0] TimerXMIS : Masked Interrupt Status bit

This bit is a logical AND value of the Raw Interrupt Status and the Timer Interrupt Enable bit of the Control Register (TimerXControl). The same value as this bit is connected to the interrupt output signal.

bit	Description
0	No interrupt generated from the counter
1	Interrupt generated from the counter [Initial value]

5.7. Background Load Register (TimerXBGLoad)

X=1 or 2

Background Load Register (TimerXBGLoad) is a 32-bit register having a value which the counter starts to decrement.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TimerXBGLoad[31:16]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TimerXBGLoad[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:0] TimerXBGLoad : Background Load bit

This register is used to reload the counter when the current count reaches zero in Periodic Mode setting.
This is not used in Free-running Mode or One-shot Mode.

Writing to this register reloads the counter differently from the writing to the Load Register (TimerXLoad). The difference is as follows. Writing to the Load Register immediately starts the counter with the new value; however, writing to this register does not immediately restart the counter with the new value.

After a value is written to either of the Load Register or the Background Register, the register value written last is returned at any reading. In other words, the same value is read from the Load Register and the Background register, and the value is always reloaded after the counter reaches zero in Periodic Mode.

CHAPTER: Watch Counter Prescaler

This chapter explains the functions and operations of the watch counter prescaler.

1. Overview of the Watch Counter Prescaler
2. Configuration of the Watch Counter Prescaler
3. Explanation of Operations and Setting Procedure Examples of the Watch Counter Prescaler
4. Registers of the Watch Counter Prescaler

1. Overview of the Watch Counter Prescaler

The watch counter prescaler is a prescaler which generates a counter clock used for a watch counter.

■ Watch Counter Prescaler

This is a prescaler which generates a count clock of the watch counter.

The watch counter prescaler can select a main clock or a sub clock as an input clock (F_{CL}). The watch counter prescaler outputs the division clocks (WCCK0 to 3) shown in Table 1-1 by setting SEL_OUT bit of the clock selection register (CLK_SEL).

Table 1-1 Division clocks generated by the watch counter prescaler

SEL_OUT	WCCK3	WCCK2	WCCK1	WCCK0
0	$2^{15}/F_{CL}$	$2^{14}/F_{CL}$	$2^{13}/F_{CL}$	$2^{12}/F_{CL}$
1	$2^{25}/F_{CL}$	$2^{24}/F_{CL}$	$2^{23}/F_{CL}$	$2^{22}/F_{CL}$

SEL_OUT : Output clock selection bit of clock selection register (CLK_SEL)

F_{CL} : Frequency of input clock

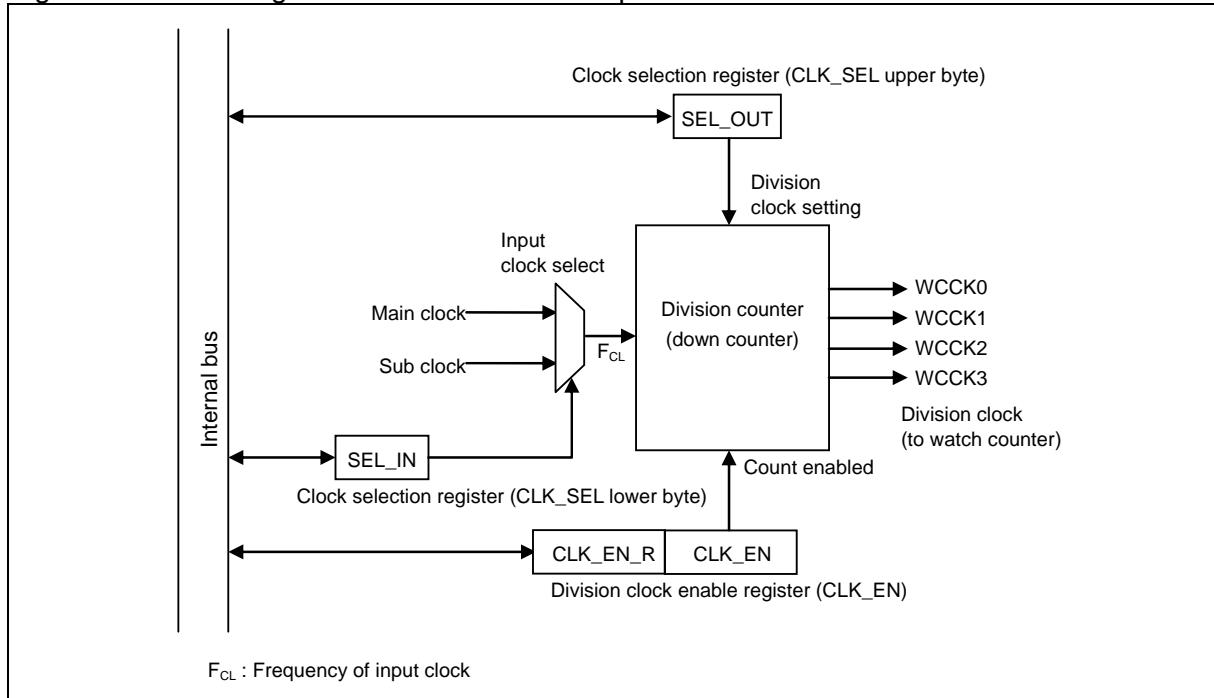
2. Configuration of the Watch Counter Prescaler

This section shows the block diagram of the watch counter prescaler.

■ Block diagram of the watch counter prescaler

Figure 2-1 shows the block diagram of the watch counter prescaler.

Figure 2-1 Block diagram of the watch counter prescaler



● Clock selection register (CLK_SEL)

This register selects the input clock (F_{CL}) which inputs the division counter, and sets the division clocks (WCCK0 to 3) that output.

● Division clock enable register (CLK_EN)

This register enables counting down of the division counter.

There is a delay for 2 cycles of the clock selected by SEL_IN bit of the clock selection register (CLK_SEL) during a period of time from a value is written to this register until the division counter starts to operate.

● Division counter

This is a down counter which generates the division clocks (WCCK0 to 3) of the input clock (F_{CL}).

3. Explanation of Operations and Setting Procedure Examples of the Watch Counter Prescaler

This section explains the operations of the watch counter prescaler. Also, procedures for setting the operating state are shown.

■ Procedures for setting the watch counter prescaler

The procedures for setting the watch counter prescaler are shown below.

● To start output of the division clock

1. Select the input clock (F_{CL}) of the division counter with SEL_IN bit of the clock selection register (CLK_SEL). Also, set the division clock that outputs with SEL_OUT bit of the clock selection register (CLK_SEL).
At this time, the division clock to be output is fixed to "L" since the division counter is not operated.
2. Set "1" to CLK_EN bit of the division clock enable register (CLK_EN) to enable output of the division clock.

● To stop output of the division clock

1. Set "1" to CLK_EN bit of the division clock enable register (CLK_EN) to disable output of the division clock.

● To restart after stopping output of the division clock

1. Set "1" to CLK_EN bit of the division clock enable register (CLK_EN) to enable output of the division clock.
2. Write "0" to WCEN bit of the watch counter control register (WCCR) of the watch counter, and clear the value of the 6-bit down counter in the watch counter to "0b000000".
3. Write "0" to WCEN bit of the watch counter control register (WCCR) of the watch counter to restart the operation of the watch counter.

● To switch while the division clock is operating

1. Set "0" to CLK_EN bit of the division clock enable register (CLK_EN) to disable output of the division clock.
2. Read CLK_EN_R bit of the division clock enable register (CLK_EN), and confirm whether output of the division clock is stopped (CLK_EN_R=0).
3. Select the input clock (F_{CL}) of the division counter by SEL_IN bit of the clock selection register (CLK_SEL). Also, set the division clock to be output with SEL_OUT bit of the clock selection register (CLK_SEL).
4. Set "1" to CLK_EN bit of the division clock enable register (CLK_EN) to enable output of the division clock.

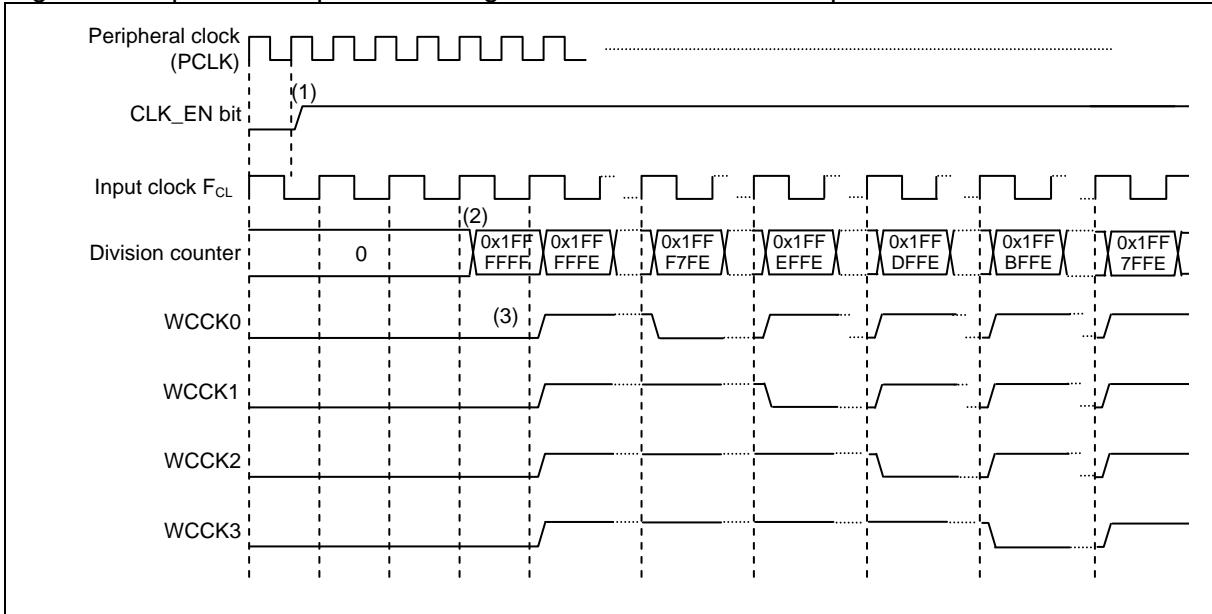
<Notes>

- The peripheral clock (PCLK) is used to set each register of the watch counter prescaler. The input clock (F_{CL}) of the division clock and the peripheral clock (PCLK) are not synchronized. Since the input clock (F_{CL}) of the division counter and peripheral clock (PCLK) are not synchronized, a delay for 3 clocks of the input clock (F_{CL}) is occurred to WCCK0 to 3 after a value is set to each register.
- Regarding 2. of "●To switch while the division clock is operating", a glitch may be occurred while the division clock is operating. Confirm whether output of the division counter is stopped.
- The watch counter uses output of the watch counter prescaler as a count clock. Therefore, the settings of the watch counter prescaler should not be changed while the watch counter is operating.

■ Operation of the watch counter prescaler

Figure 3-1 shows an operation of the watch counter prescaler when SEL_OUT is set to "0" as an example.

Figure 3-1 Operation explanation diagram of the watch counter prescaler



(1) Set CLK_EN bit at rising of the peripheral clock (PCLK).

(2) The division counter is operated synchronizing with the input clock (F_{CL}).

(3) The clocks are output to WCCK0 to 3 from the counter according to the settings of SEL_OUT bit.

<Note>

The peripheral clock (PCLK) is used for the settings of each register of the watch counter prescaler. Since the input clock (F_{CL}) of the division counter and peripheral clock (PCLK) are not synchronized, a delay for 4 clocks of the input clock (F_{CL}) is occurred to WCCK0 to 3 after a value is set to each register.

■ Relationship between the frequency of the input clock (F_{CL}) and the cycle of the division clock

Table 3-1 shows the setting example of the frequency of the input clock (F_{CL}) and the cycle of the division clock.

Table 3-1 Setting example of the watch counter prescaler

SEL_IN	SEL_OUT	Input clock frequency (F_{CL})	Cycle of division clock			
			WCCK3	WCCK2	WCCK1	WCCK0
0 (sub clock)	0	32.768 kHz	1s	500 ms	250 ms	125 ms
1 (main clock)	1	33.554 MHz	1s	500 ms	250 ms	125 ms

4. Registers of the Watch Counter Prescaler

This section explains the registers for the watch counter prescaler.

■ List of registers for the watch counter prescaler

Table 4-1 List of registers for the watch counter prescaler

Abbreviated Register Name	Register Name	Reference
CLK_SEL	Clock selection register	4.1
CLK_EN	Division clock enable register	4.2

4.1. Clock Selection Register (CLK_SEL)

The clock selection register (CLK_SEL) selects the input clock (F_{CL}) and sets the division clocks (WCCK0 to 3) to be output.

bit	15	to	11	10	9	8
Field		res		res		SEL_OUT
Attribute		R/W		R/W		R/W
Initial value		0b000000		0b00		0

bit	7	to	1	0
Field		res		SEL_IN
Attribute		R/W		R/W
Initial value		0b0000000		0

[bit15:11, bit7:1] res : Reserved bits

"0" is always read.

Writing is ignored.

[bit10: 9] res : Reserved bits

Always write "0" to these bits.

[bit8] SEL_OUT : Output clock selection bit

This bit selects the division clocks (WCCK0 to 3) to be output from the division counter.

bit	Explanation			
	WCCK3	WCCK2	WCCK1	WCCK0
0	$2^{15}/F_{CL}$	$2^{14}/F_{CL}$	$2^{13}/F_{CL}$	$2^{12}/F_{CL}$
1	$2^{25}/F_{CL}$	$2^{24}/F_{CL}$	$2^{23}/F_{CL}$	$2^{22}/F_{CL}$

[bit0] SEL_IN : Input clock selection bit

This bit selects the input clock (F_{CL}) to be used.

bit	Explanation
0	Generates a division clock using the sub clock.
1	Generates a division clock using the main clock.

4.2. Division Clock Enable Register (CLK_EN)

The division clock enable register (CLK_EN) is a register to enable a count down of the division counter.

bit	7	to	2	1	0
Field		res		CLK_EN_R	CLK_EN
Attribute		R/W		R/W	R/W
Initial value		0b000000		0	0

[bit7:2] res : Reserved bits

"0" is always read.

Writing is ignored.

[bit1] CLK_EN_R : Division clock enable read bit

This bit can read the value of CLK_EN used for controlling the division. Writing to this bit does not affect the operations and the reading value.

bit	Explanation
0	The counter for the clock division stops counting, and oscillation of the division clock is not performed.
1	The counter for the clock division starts counting, and oscillation of the division clock is not performed.

[bit0] CLK_EN : Division clock enable bit

There is a delay for 2 cycles in the clock selected by CLK_SEL register during a period of time from a value is written to CLK_EN bit until the value is reflected.

bit	Explanation
0	The division counter stops counting, and disables oscillation of the division clock. Clears the value of the division counter to "0".
1	The division counter starts counting, and enables oscillation of the division clock.

CHAPTER: Watch Counter

This chapter explains the functions and operations of the watch counter.

1. Overview of the Watch Counter
2. Configuration of the Watch Counter
3. Interrupts of the Watch Counter
4. Explanation of Operations and Setting Procedure Examples of the Watch Counter
5. Registers of the Watch Counter

1. Overview of the Watch Counter

The watch counter is a timer that counts down starting from the specified value, and it generates an interrupt request at the time that the 6-bit down counter enters an underflow condition.

■ Watch counter

- For the watch counter, one of the four types of clock (WCCK0, WCCK1, WCCK2, and WCCK3) selected by the CS1 and CS0 bits of the watch counter control register (WCCR) is used as a count clock of the 6-bit down counter.
- A number between 0 and 63 can be set as the value used for counting by the 6-bit down counter. If "60" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 1 minute. If "0" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 64 seconds
- An interrupt request can be generated at the time that the 6-bit down counter enters an underflow condition.

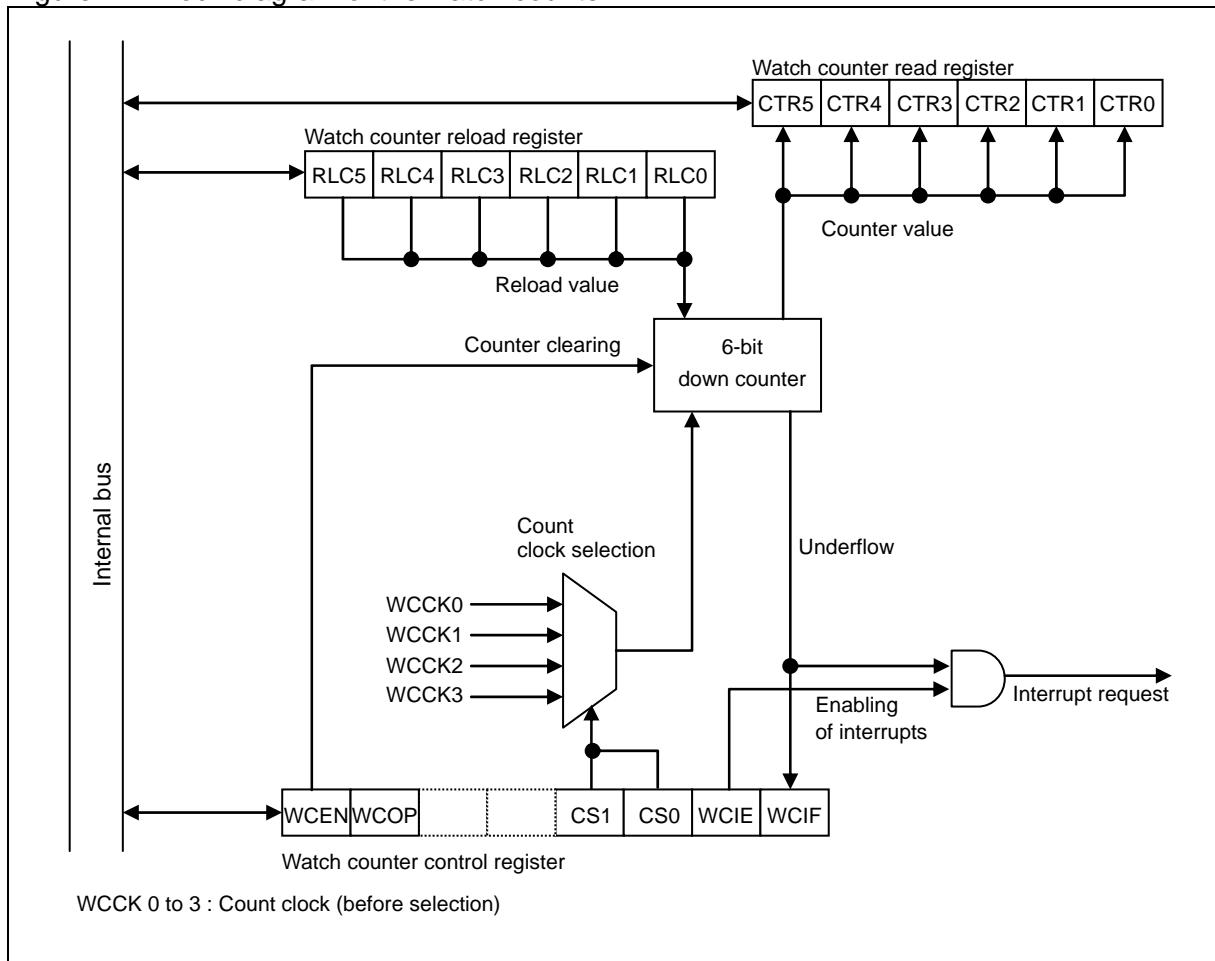
2. Configuration of the Watch Counter

This section shows the watch counter block diagram.

■ Block diagram of the watch counter

Figure 2-1 shows a block diagram of the watch counter.

Figure 2-1 Block diagram of the watch counter



● 6-bit down counter

This is the 6-bit down counter of the watch counter. It reloads the value set in the watch counter reload register (WCRL) and starts counting down.

● Watch counter reload register (WCRL)

This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in this register.

● Watch counter read register (WCRD)

This register reads the value in the 6-bit down counter. Also, the register can be read to check the count value.

● Watch counter control register (WCCR)

This register controls the operation of the watch counter.

3. Interrupts of the Watch Counter

The 6-bit down counter enters an underflow condition when the value in the 6-bit down counter becomes "0b000001", and an underflow interrupt request is then generated.

■ Interrupts of the watch counter

Table 3-1 outlines the interrupts that can be used with the watch counter.

Table 3-1 Interrupts of the watch counter

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Underflow interrupt request	WCIF=1 for WCCR	WCIE=1 for WCCR	Write "0" to the WCIF bit for WCCR

WCCR : Watch counter control register

<Note>

If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time. To enable generation of the interrupt request, do either of the following.

- Clear interrupt requests before enabling the generation of interrupt requests.
- Clear interrupt requests simultaneously with interrupts enabled.

4. Explanation of Operations and Setting Procedure Examples of the Watch Counter

This section explains operations of the watch counter. Also, examples of procedures for setting the operating state are shown.

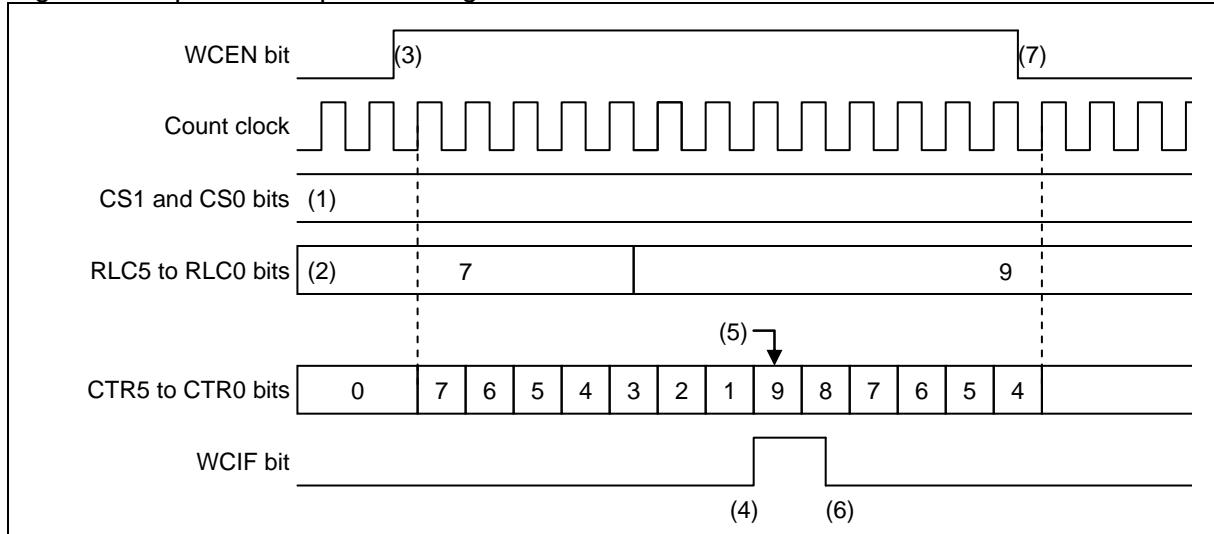
■ Setting procedure examples of the watch counter

To operate the watch counter, follow the procedure below.

- (1) Select a count clock by using the CS1 and CS0 bits of the watch counter control register (WCCR).
- (2) Set a count value to the RLC5 to RLC0 bits in the watch counter reload register (WCRL).
- (3) Enable the operation of the watch counter by using the WCEN bit (WCEN = 1) of the watch counter control register (WCCR).
Start a countdown. Counting is performed at the rising edge of the count clock.
- (4) If the 6-bit down counter enters an underflow condition, the value of the WCIF bit in the watch counter control register (WCCR) is changed to "1".
At this time, if generation of underflow interrupt requests has been enabled by the WCIE bit in the watch counter control register (WCCR), an underflow interrupt request is generated.
Also, the value that is set in the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is reloaded in the 6-bit down counter and the countdown is restarted.
- (5) If the value of the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is changed to another value while the watch counter is active, the watch counter is updated with the new value at the next reload time.
- (6) The underflow interrupt request is cleared when "0" is written to the WCIF bit in the watch counter control register (WCCR).
- (7) The 6-bit down counter is cleared to "0b000000" and the counting operation is stopped when "0" is written to the WCEN bit in the watch counter control register (WCCR).

Figure 4-1 shows the operation of the watch counter.

Figure 4-1 Operation explanation figure of the watch counter



<Notes>

- The peripheral clock (PCLK) is used for the settings of each register of the watch counter. Since the count clock and peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: Count clock period) may occur at the count start time, depending on the time at which "1" is written to the WCEN bit in the watch counter control register (WCCR).
- Even at transition of the timer mode, the watch counter continues operating as long as the main clock or sub clock is operating. The timer mode can be canceled with the watch counter interrupt processing routine.
- Under the following condition, verify that the watch counter is stopped by checking the WCOP bit (WCOP=0) in the watch counter control register (WCCR) before reactivating the watch counter.
Condition: In case of activating the watch counter after the watch counter is stopped by writing "0" to the WCEN in the watch counter control register (WCCR) by using the WCEN bit (WCEN = 1).

5. Registers of the Watch Counter

This section explains the registers for the watch counter.

■ List of the registers for the watch counter

Table 5-1 List of registers for the watch counter

Abbreviated Register Name	Register Name	See
WCRD	Watch counter read register	5.1
WCRL	Watch counter reload register	5.2
WCCR	Watch counter control register	5.3

5.1. Watch Counter Read Register (WCRD)

This register reads the value in the 6-bit down counter.

bit	7	6	5	4	3	2	1	0
Field	res		CTR5	CTR4	CTR3	CTR2	CTR1	CTR0
Attribute	R		R	R	R	R	R	R
Initial value	0b00		0	0	0	0	0	0

[bit7:6] res : Reserved bits

"0" is always read.

Writing is ignored.

[bit5:0] CTR5 to CTR0 : Counter read bits

These bits can read the counter value.

Writing is ignored.

<Note>

If the 6-bit down counter is operating when its value is read, the register value must be read twice and verified to be the same value.

5.2. Watch Counter Reload Register (WCRL)

This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in the register.

The register specifies the reload value for the 6-bit down counter. If the 6-bit down counter enters an underflow condition, the value in this register is reloaded in the 6-bit down counter, and the countdown is restarted.

bit	15	14	13	12	11	10	9	8
Field	res		RLC5	RLC4	RLC3	RLC2	RLC1	RLC0
Attribute	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0b00		0	0	0	0	0	0

[bit15:14] res : Reserved bits

"0" is always read.

Writing is ignored.

[bit13:8] RLC5 to RLC0 : Counter reload value setting bit

These bits set the reload value for the 6-bit down counter.

The 6-bit counter counts downwards from the reload value and enters an underflow condition when its value reaches "1". If "0" is set in these bits, it performs 64 countdowns from "63" to "0".

If this bit is modified during counting, the modified value is valid at reloading after underflow.

<Notes>

- If the value of these bits is changed to another value while the 6-bit down counter is active, an underflow occurs and the new value is then reloaded.
- If the value of RLC bit is changed to another value at the same time that an underflow interrupt request is generated, the correct value is not reloaded. Be sure to rewrite the value of RLC bit either when the watch counter is stopped or in the interrupt processing routine before an interrupt request is generated.
- To verify whether the reload value is correctly set, read this register.

5.3. Watch Counter Control Register (WCCR)

This register selects a count clock for the watch counter or enables/disables generation of interrupt requests. The register also enables/disables the operation of the watch counter.

bit	23	22	21	20	19	18	17	16
Field	WCEN	WCOP	res		CS1	CS0	WCIE	WCIF
Attribute	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0b00	0	0	0	0	0

[bit23] WCEN : Watch counter operation enable bit

This bit enables the operation of the watch counter.

- The peripheral clock (PCLK) is used for the settings of each register of the watch counter. Since the count clock and the peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: count clock period) may occur at the count start time, depending on the time at which "1" is written to WCEN bit of watch counter control register (WCCR).
- Before writing "1" to this bit to start the operation of the watch counter, verify that the watch counter is stopped by checking the WCOP bit (WCOP=0).

bit	Explanation
0	The watch counter is disabled/stopped. The value in the 6-bit down counter is cleared to "0b000000".
1	The watch counter is enabled/started.

[bit22] WCOP : Watch counter operating state flag

This bit indicates the operating state of the watch counter.

bit	Explanation
0	The watch counter is stopped.
1	The watch counter is active.

[bit21:20] res : Reserved bits

"0" is always read.

Writing is ignored.

[bit19:18] CS1 to CS0 : Count clock select bits

These bits select a clock for the watch counter.

Change these bits when WCCR : WCEN=0 (watch counter operation disabled) and WCOP=0 (watch counter stopped).

bit19	bit18	Explanation
0	0	Selects WCCK0 as a count clock.
0	1	Selects WCCK1 as a count clock.
1	0	Selects WCCK2 as a count clock.
1	1	Selects WCCK3 as a count clock.

[bit17] WCIE : Interrupt request enable bit

This bit specifies whether to generate an underflow interrupt request when the 6-bit down counter underflows (WCIF=1).

bit	Explanation
0	Disables generation of underflow interrupt requests.
1	Enables generation of underflow interrupt requests.

[bit16] WCIF : Interrupt request flag bit

This bit becomes "1" when the counter underflows.

- When this bit and WCIE bit are "1", a watch counter interrupt is generated.
- When a read-modify-write instruction is used, "1" is read.

bit	Explanation
0	This bit is cleared.
1	Ignored

Chapter: Base Timer I/O Select Function

This chapter explains about base timer I/O function.

1. Overview
2. Configuration
3. I/O Mode
4. Registers

1. Overview

The base timer I/O select function sets the I/O mode, and thereby determines the method to input and output signals (external clock, external start trigger, and waveform) to/from the base timer.

By switching timer function, each channel of the base timer can be also used as one of the following timers:

- 16-bit PWM timer
- 16-bit PPG timer
- 16/32-bit reload timer
- 16/32-bit PWC timer

■ Overview

One of the following 9 types of I/O modes can be selected for each 2 channels.

Software-based simultaneous startup function is provided for multiple channels, enabling up to 16 channels to be started up via software.

- I/O mode 0: Standard 16-bit timer mode
This mode operates each channel of the base timer individually.
- I/O mode 1: Timer full mode
This mode assigns each even channel signal of the base timer with an external pin individually to operate the channel.
- I/O mode 2: Shared external trigger mode
This mode can input an external trigger to two channels of the base timer simultaneously. Using this mode, the base timer of two channels can be started up simultaneously.
- I/O mode 3: Shared channel signal trigger mode
This mode uses a signal from another channel as an external startup trigger. This mode cannot be selected for channel 0 or 1.
- I/O mode 4: Timer start/stop mode
This mode controls the start/stop of the odd channel using the even channel. The odd channel starts on the rising edge of output signal from the even channel, and stops on the falling edge.
- I/O mode 5: Software-based simultaneous startup mode
This mode starts up multiple channels simultaneously via software.
- I/O mode 6: Software-based startup and timer start/stop mode
This mode controls the start/stop of the odd channel using the even channel. An even channel is started up via software. The odd channel starts on the rising edge of output signal from the even channel, and stops on the falling edge.
- I/O mode 7: Timer start mode
This mode controls the start of the odd channel using the even channel. The odd channel starts on the rising edge of output signal from the even channel.
- I/O mode 8: Shared channel signal trigger and timer start/stop mode
This mode uses a signal from another channel as an external startup trigger. This mode cannot be selected for channel 0 or 1.

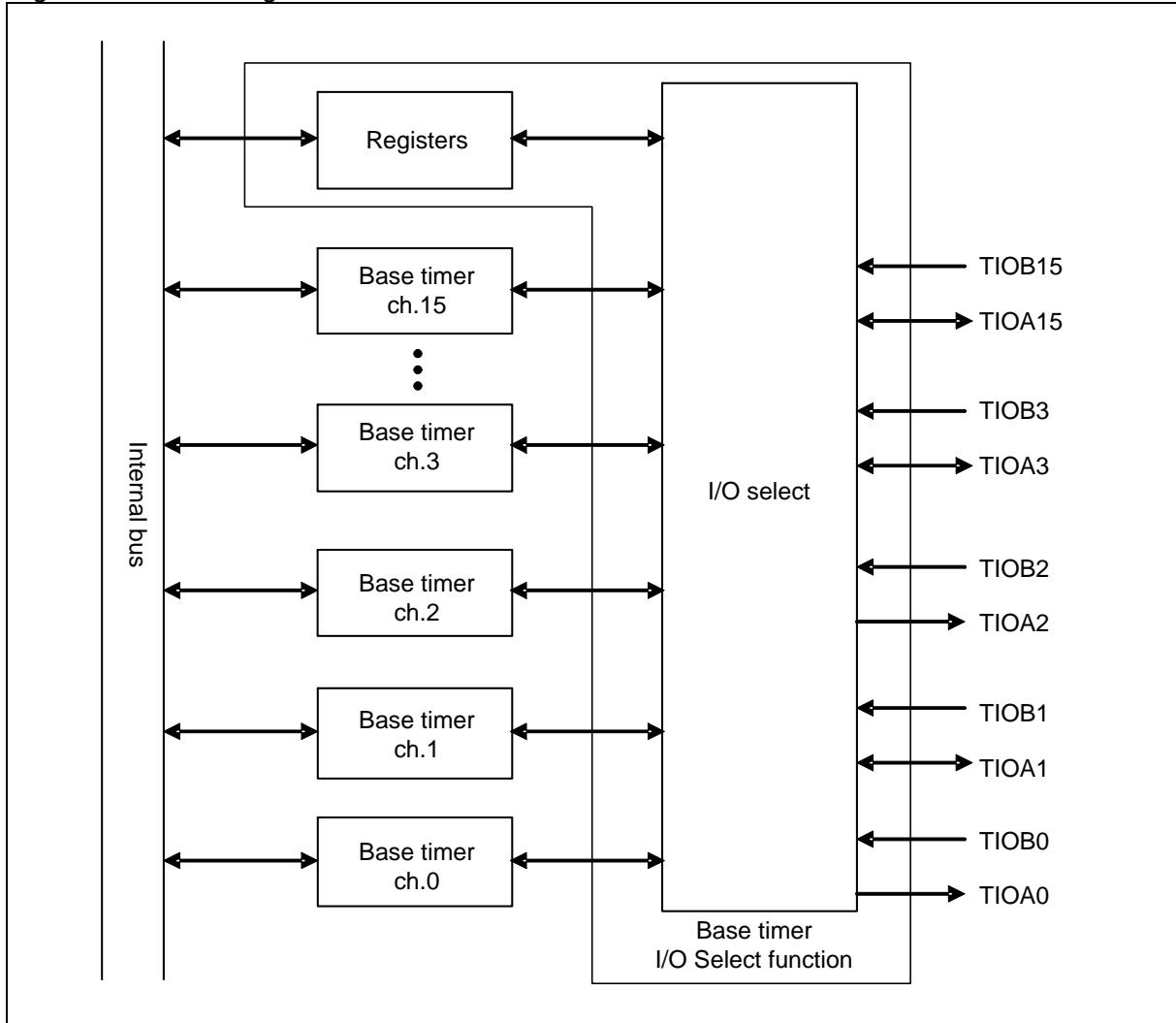
2. Configuration

The base timer I/O select function consists of the following blocks.

■ Block diagram

Figure 2-1 shows the block diagram of the base timer I/O select function.

Figure 2-1 Block diagram of base timer I/O select function



- I/O select
A circuit that selects the I/O mode of the base timer for each channel.
- Base timer (Channels 0 to 15)
Base timer channels 0 to 15 (up to 16 channels).
- Registers
Registers of base timer I/O select function.

3. I/O Mode

This section explains pins used by the base timer I/O select function to set the I/O mode, and also explains each I/O mode.

3.1 Pins

3.2 I/O mode

3.1. Pins

This section explains pins used by the base timer I/O select function to set the I/O mode.

Each channel of the base timer has 2 types of external pins and 5 types of internal signals. Also base timer I/O select function has 2 types of internal signals. By connecting an internal signal with an external pin, the signal corresponding to the connected (external clock (ECK signal)/external startup trigger (TGIN signal)/waveform (TIN signal)) is input or output to/from the base timer. The external pin and internal signal can be connected by setting the I/O mode of the base timer. The pin used and the signal input or output differ depending on the I/O mode.

■ External pins

- TIOA pin
This pin is used to output the base timer waveform (TOUT signal), or input an external startup trigger (TGIN signal).
- TIOB pin
This pin is used to input external startup trigger (TGIN signal)/external clock (ECK signal)/another channel waveform (TIN signal).

■ Internal signals

A signal is input or output to/from the base timer by being connected with an above external pin, or by inputting an output signal from another channel.

- TOUT signal
This signal is the output waveform of the base timer. (Not used by the 16/32-bit PWC timer.)
- ECK signal
This signal is an external clock of the base timer. (Not used by the 16/32-bit PWC timer.)
It is input when the external clock is selected as a counting clock.
- TGIN signal
This signal is the external startup trigger of the base timer. (Not used by the 16/32-bit PWC timer.)
When the valid edge of external startup trigger is selected, the base timer detects the edge of this signal to start up.
- TIN signal
This signal is the input waveform of the base timer. This signal is the waveform to be measured. (Used only by the 16/32-bit PWC timer.)
- DTRG signal
This signal is the trigger input to the base timer. The base timer stops operating on the falling edge of this signal.
- COUT signal
This signal is the trigger output of the base timer I/O select function. This signal is output to another channel.
- CIN signal
This signal is the trigger input to the base timer I/O select function. This signal is input from another channel.

■ Connecting the external pin to the internal signal

The external pin and internal signal can be connected by setting the I/O mode of the base timer.

Table 3-1 shows the correspondence between I/O modes and pin connections.

Table 3-1 Correspondence between I/O modes and pin connections

I/O mode	TIOAn (Even channel)		TIOBn (Even channel)		TIOAn+1 (Odd channel)		TIOBn+1 (Odd channel)	
	Connected to	I/O	Connected to	I/O	Connected to	I/O	Connected to	I/O
0	Ch.n TOUT	Output	Ch.n ECK/TGIN/ TIN	Input	Ch.n+1 TOUT	Output	Ch.n+1 ECK/TGIN/ TIN	Input
1	Ch.n TOUT	Output	Ch.n ECK	Input	Ch.n TGIN	Input	Ch.n TIN	Input
2	Ch.n TOUT	Output	Ch.n/Ch.n+1 ECK/TGIN/ TIN *1	Input	Ch.n+1 TOUT	Output	Not used	
3	Ch.n TOUT	Output	Not used		Ch.n+1 TOUT	Output		
4	Ch.n TOUT	Output	Ch.n ECK/TGIN/ TIN	Input	Ch.n+1 TOUT	Output		
5	Ch.n TOUT	Output	Not used		Ch.n+1 TOUT	Output		
6	Ch.n TOUT	Output			Ch.n+1 TOUT	Output		
7	Ch.n TOUT	Output	Ch.n ECK/TGIN/ TIN	Input	Ch.n+1 TOUT	Output		
8	Ch.n TOUT	Output	Not used		Ch.n+1 TOUT	Output		

n : Even

Ch.n : Even channel

Ch.n+1 : Odd channel

*1 : Synchronized by the peripheral clock (PCLK)

3.2. I/O mode

I/O mode selected by the I/O Select Register (BTSEL) determines the functions of external pins and the start/stop timing of the base timer.

■ I/O mode 0 (Standard 16-bit timer mode)

This mode uses each channel of the base timer individually.

Table 3-2 shows the external pins used when this mode is selected.

Table 3-2 External pins used when I/O mode 0 is selected.

	Even channel	Odd channel
Number of input pins	1	1
Number of output pins	1	1

Table 3-3 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-3 External pin connections and input/output signals when I/O mode 0 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOA	Output	TOUT	Outputs the base timer waveform
TIOB	Input	ECK/TGIN/TIN*	Uses the input signal as one of the following signals: <ul style="list-style-type: none"> · External clock (ECK signal) · External startup trigger (TGIN signal) · Waveform to be measured (TIN signal)

* : The usage of input signals (ECK/TGIN/TIN) differs depending on the Timer Control Register (TMCR) setting of the base timer.

Figure 3-1 provides the block diagram of I/O mode 0 (Standard 16-bit timer mode).

Figure 3-1 I/O mode 0 (Standard 16-bit timer mode) block diagram

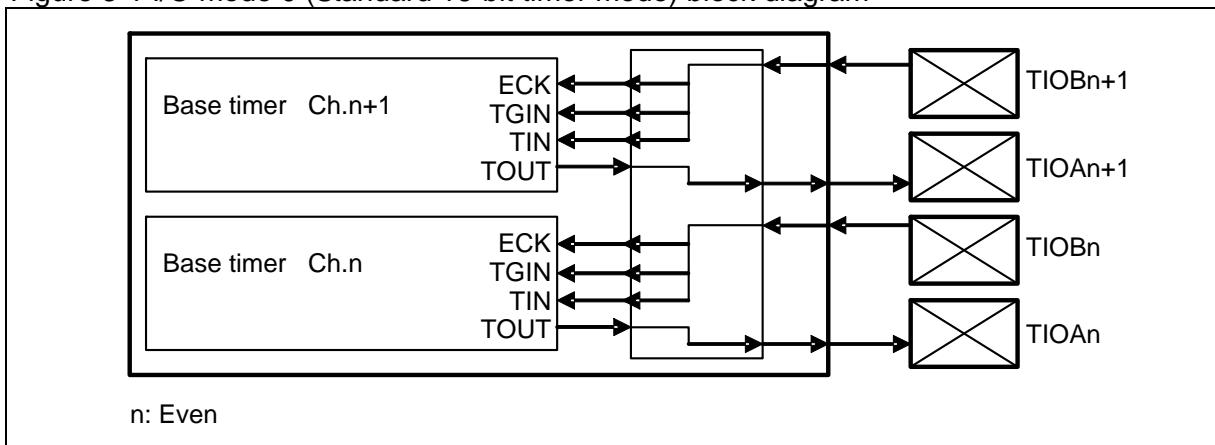


Table 3-4 shows signal connections in I/O mode 0.

Table 3-4 I/O mode 0 signal connections

Signal	Connected to
Ch.n TOUT signal	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to Ch.n as ECK/TGIN/TIN signals
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin
Input signal from the TIOBn+1 pin	Input to Ch.n+1 as ECK/TGIN/TIN signals

n : Even

■ **I/O mode 1 (timer full mode)**

This mode assigns every even channel signal with an external pin individually.

Table 3-5 shows the external pins used when this mode is selected.

Table 3-5 External pins used when I/O mode 1 is selected.

		Even channel
Number of input pins		3
Number of output pins		1

Table 3-6 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-6 External pin connections and input/output signals when I/O mode 1 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOBn	Input	Even channel ECK	Inputs an external clock (ECK signal) to the even channel.
TIOAn+1	Input	Even channel TGIN	Inputs an external startup trigger (TGIN signal) to the even channel.
TIOBn+1	Input	Even channel TIN	Inputs the waveform to be measured (TIN signal) to the even channel.

n : Even

Figure 3-2 shows the block diagram of I/O mode 1 (timer full mode).

Figure 3-2 I/O mode 1 (timer full mode) block diagram

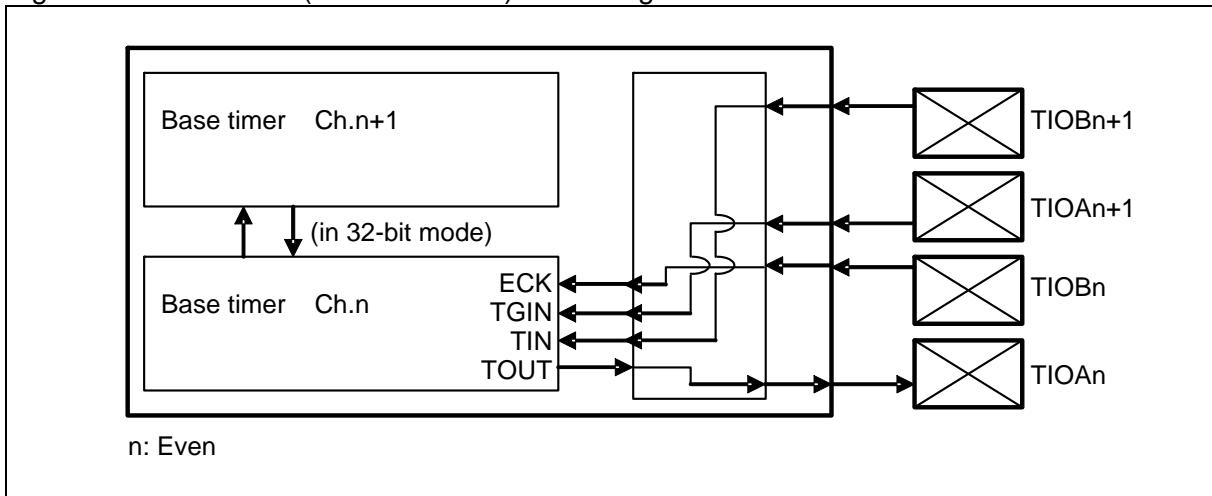


Table 3-7 shows signal connections in I/O mode 1.

Table 3-7 I/O mode 1 signal connections

Signal	Connected to
Ch.n TOUT signal	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to Ch.n as a TIN signal
Ch.n+1 TOUT signal	Input to Ch.n as a TGIN signal
TIOBn+1 pin	Input to Ch.n as an ECK signal

n : Even

<Note>

When this mode is selected, the TIOA pins (TIOA1, TIOA3, etc.) corresponding to the even channel must be set to port input mode with the Port Function Register (PFR) of GPIO.

■ I/O mode 2 (Shared external trigger mode)

This mode shares the input signals (ECK/TGIN/TIN) of the base timer between two channels.

Table 3-8 shows the external pins used when this mode is selected.

Table 3-8 External pins used when I/O mode 2 is selected.

	Even channel	Odd channel
Number of input pins	1 (shared by two channels)	
Number of output pins	1	1

Table 3-9 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-9 External pin connections and input/output signals when I/O mode 2 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	Input	ECK/TGIN/TIN of even and odd channels *	Input to both the even and odd channels (synchronized by the peripheral clock (PCLK)) and used as one of the following signals: · External clock (ECK signal) · External startup trigger (TGIN signal) · Waveform to be measured (TIN signal)
TIOBn+1	-	-	Not used

n : Even

* : The usage of input signals (ECK/TGIN/TIN) differs depending on the Timer Control Register (TMCR) setting of the base timer.

Figure 3-3 shows the block diagram of I/O mode 2 (Shared external trigger mode).

Figure 3-3 I/O mode 2 (Shared external trigger mode) block diagram

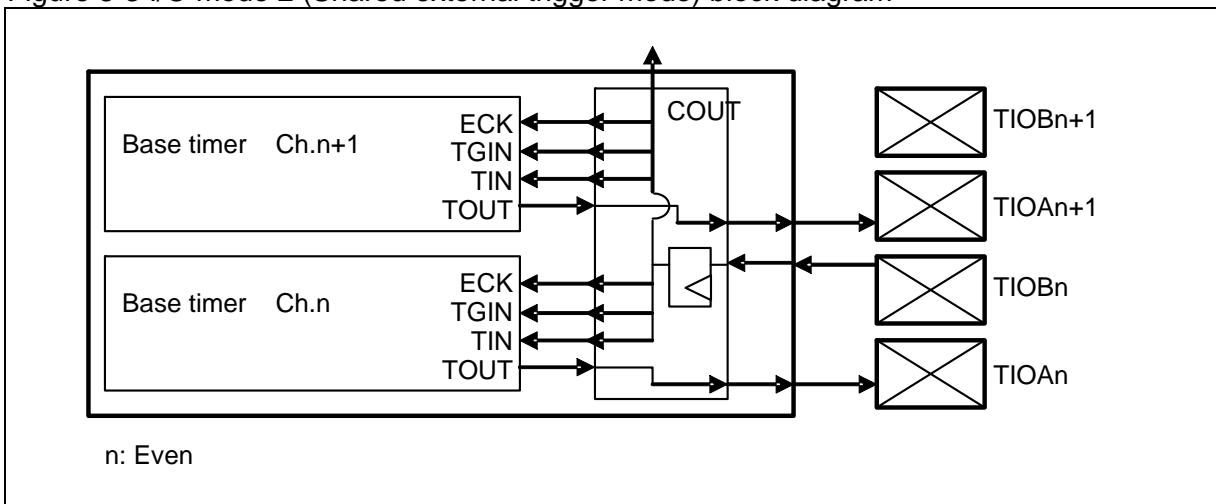


Table 3-10 shows signal connections in I/O mode 2.

Table 3-10 I/O mode 2 signal connections

Signal	Connected to	Remarks
Ch.n TOUT signal	Output from the TIOAn pin	
Input signal from the TIOBn pin	<ul style="list-style-type: none"> · Input to Ch.n and Ch.n+1 as ECK/TGIN/TIN signals · Output to another channel as a COUT signal 	Synchronized by the peripheral clock (PCLK)
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin	

n : Even

<Note>

If two channels above the channels set to this mode ($n+2, n+3$) are set to I/O mode 3 (Shared channel signal trigger mode), the input signals (ECK/TGIN/TIN) can be input to the 4 channels simultaneously.

(Example: If channels 0 and 1 are set to this mode, and channels 2 and 3 are set to I/O mode 3, input signals (ECK/TGIN/TIN) can be input to four channels of 0 to 3 simultaneously.)

■ I/O mode 3 (Shared channel signal trigger mode)

This mode inputs the COUT signal from channels below the two channels as a CIN signal, and uses it as ECK/TGIN/TIN signals.

Table 3-11 shows the external pins used when this mode is selected.

Table 3-11 External pins used when I/O mode 3 is selected.

	Even channel	Odd channel
Number of input pins	Not used	
Number of output pins	1	1

Table 3-12 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-12 External pin connections and input/output signals when I/O mode 3 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	-	-	Not used
TIOBn+1	-	-	Not used

n : Even

Figure 3-4 shows the block diagram of I/O mode 3 (Shared channel signal trigger mode).

Figure 3-4 I/O mode 3 (Shared channel signal trigger mode) block diagram

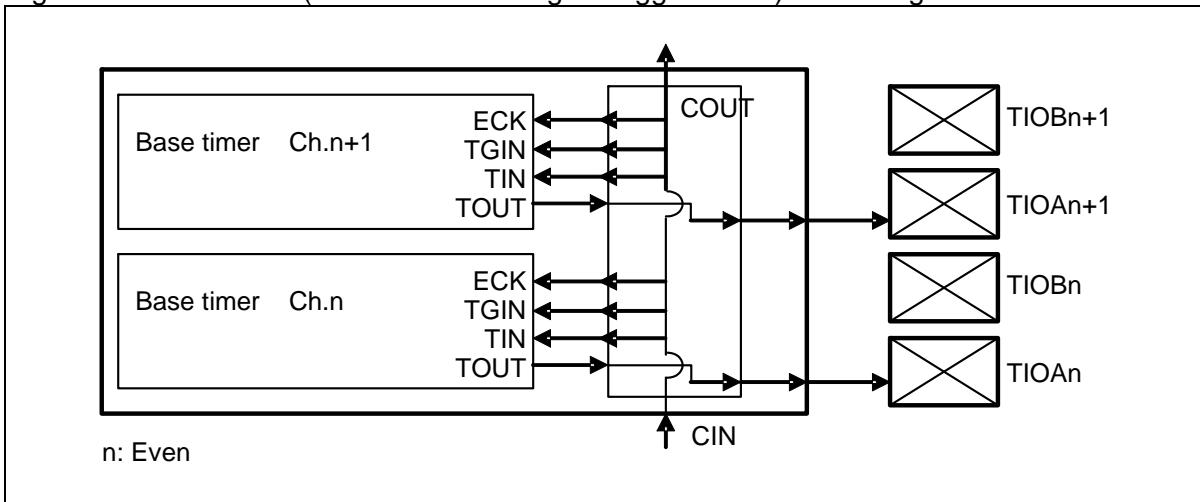


Table 3-13 shows signal connections in I/O mode 3.

Table 3-13 I/O mode 3 signal connections

Signal	Connected to
Ch.n TOUT signal	Output from the TIOAn pin
CIN signal *	<ul style="list-style-type: none"> · Input to Ch.n and Ch.n+1 as ECK/TGIN/TIN signals · Output to another channel as a COUT signal
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin

n : Even

* : The COUT signal from another channel is input as a CIN signal.

The following shows Ch.n-2/n-1 signals that can be input to ECK/TGIN/TIN of Ch.n/n+1.

- Signal that the peripheral clock generates by synchronizing TIOBn-2 input in I/O mode 2.
- Trigger signal input from Ch.n-4/n-3 in I/O mode 3.
- TIOAn-2 output in I/O mode 4.
- TIOAn-2 output in I/O mode 6.
- TIOAn-2 output in I/O mode 7.
- Trigger signal input from Ch.n-4/n-3 in I/O mode 8.

<Notes>

- Select the rising edge as a trigger input edge using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)
 - The channels set to this mode use the COUT signal from lower two channels (n-2 and n-1) as a CIN signal. (Example: If channels 2 and 3 are set to this mode, they use the COUT signal from channels 0 and 1.) Therefore, channels 0 and 1 cannot be set to this mode.
-

■ I/O mode 4 (Timer start/stop mode)

This mode can control the start/stop of the odd channel using the even channel.

The odd channel starts on the rising edge of output waveform (TOUT signal) of the even channel, and stops on the falling edge.

Table 3-14 shows the external pins used when this mode is selected.

Table 3-14 External pins used when I/O mode 4 is selected.

	Even channel	Odd channel
Number of input pins	1	Not used
Number of output pins	1	1

Table 3-15 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-15 External pin connections and input/output signals when I/O mode 4 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	Input	ECK/TGIN/TIN of even channel *	Input to the even channel and used as one of the following signals: <ul style="list-style-type: none"> · External clock (ECK signal) · External startup trigger (TGIN signal) · Waveform to be measured (TIN signal)
TIOBn+1	-	-	Not used

n : Even

* : The usage of input signals (ECK/TGIN/TIN) differs depending on the Timer Control Register (TMCR) setting of the base timer.

Figure 3-5 shows the block diagram of I/O mode 4 (Timer start/stop mode).

Figure 3-5 I/O mode 4 (Timer start/stop mode) block diagram

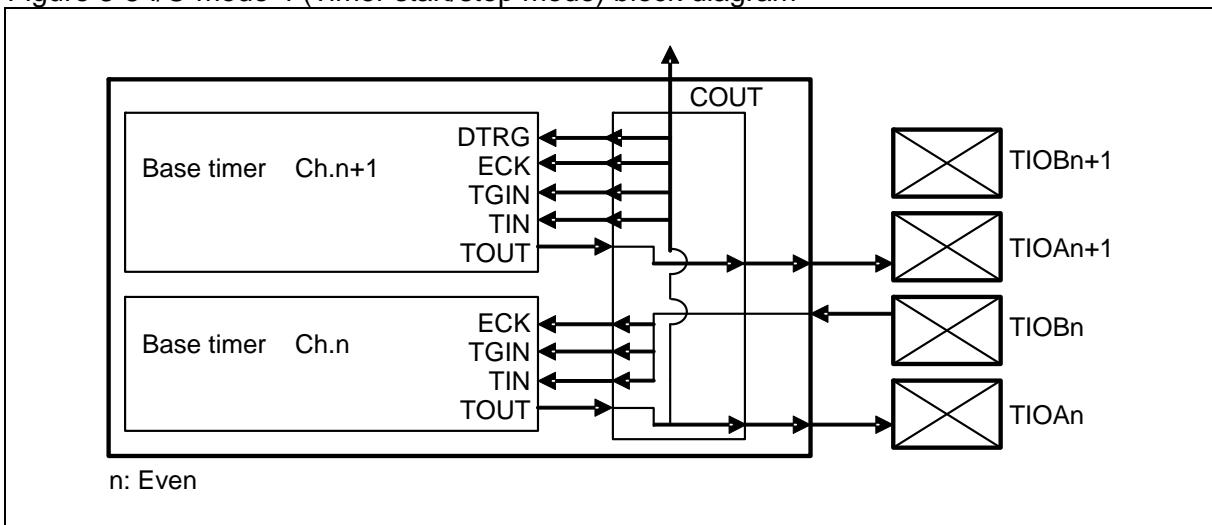


Table 3-16 shows signal connections in I/O mode 4.

Table 3-16 I/O mode 4 signal connections

Signal	Connected to
Ch.n TOUT signal	<ul style="list-style-type: none"> · Output from the TIOAn pin · Input to Ch.n+1 as ECK/TGIN/TIN and DTRG signals · Output to another channel as a COUT signal
Input signal from the TIOBn pin	Input to Ch.n as ECK/TGIN/TIN signals
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin

n : Even

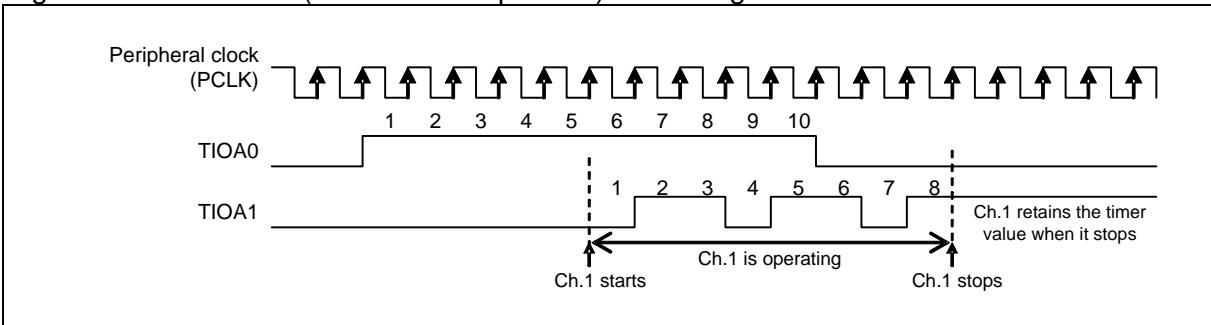
<Notes>

- Select the rising edge as a trigger input edge of the odd channel using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)
- The odd channel stops operating when a falling edge is detected in the DTRG signal.

Figure 3-6 shows example operation when I/O mode 4 (Timer start/stop mode) is selected, and when channels 0 and 1 are used as PWM timer.

Base timer Ch.0	Set value	Base timer Ch.1	Set value
Cycle Setup Register (PCSR)	0x0010	Cycle Setup Register (PCSR)	0x0002
Duty Setup Register (PDUT)	0x0009	Duty Setup Register (PDUT)	0x0001
Timer Control Register (TMCR)	0x0013	Timer Control Register (TMCR)	0x0112

Figure 3-6 I/O mode 4 (Timer start/stop mode) block diagram



■ I/O mode 5 (Software-based simultaneous startup mode)

This mode starts up multiple channels simultaneously using the Software-based Simultaneous Startup Register (BTSSSR).

All the channels corresponding to the Software-based Simultaneous Startup Register (BTSSSR) bits that have been set to "1" start up simultaneously.

Table 3-17 shows the external pins used when this mode is selected.

Table 3-17 External pins used when I/O mode 5 is selected.

	Even channel	Odd channel
Number of input pins	Not used	
Number of output pins	1	1

Table 3-18 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-18 External pin connections and input/output signals when I/O mode 5 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	-	-	Not used
TIOBn+1	-	-	

n : Even

Figure 3-7 shows the block diagram of I/O mode 5 (Software-based simultaneous startup mode).

Figure 3-7 I/O mode 5 (Software-based simultaneous startup mode) block diagram

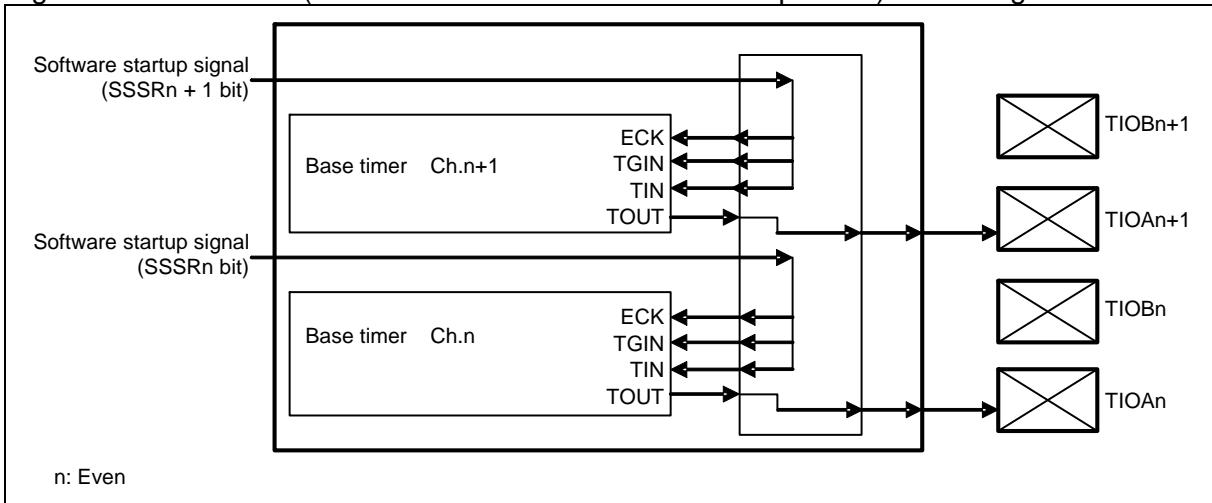


Table 3-19 shows signal connections in I/O mode 5.

Table 3-19 I/O mode 5 signal connections

Signal	Connected to
Ch.n TOUT signal	Output from the TIOAn pin
Software startup signal (Write "1" to the SSSRn bit in the BTSSSR)	Input to Ch.n as ECK/TGIN/TIN signals
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin
Software startup signal (Write "1" to the SSSRn+1 bit in the BTSSSR)	Input to Ch.n+1 as ECK/TGIN/TIN signals

n : Even

BTSSSR : Software-based Simultaneous Startup Register

When "1" is written to a Software-based Simultaneous Startup Register (BTSSSR) bit, a rising edge is input (ECK/TGIN/TIN signals) to the channel corresponding to the bit.

<Note>

Select the rising edge as a trigger input edge using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)

■ I/O mode 6 (Software-based startup and timer start/stop mode)

This mode can control the start/stop of the odd channel using the even channel.

The even channel can be started by writing "1" to the Software-based Simultaneous Startup Register (BTSSSR).

The odd channel starts when the rising edge is detected in output waveform (TOUT signal) of the even channel, and stops when the falling edge is detected.

Table 3-20 shows the external pins used when this mode is selected.

Table 3-20 External pins used when I/O mode 6 is selected.

	Even channel	Odd channel
Number of input pins	Not used	
Number of output pins	1	1

Table 3-21 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-21 External pin connections and input/output signals when I/O mode 6 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn TIOBn+1	-	-	Not used

n : Even

Figure 3-8 shows the block diagram of I/O mode 6 (Software-based startup and timer start/stop mode).

Figure 3-8 I/O mode 6 (Software-based startup and timer start/stop mode) block diagram

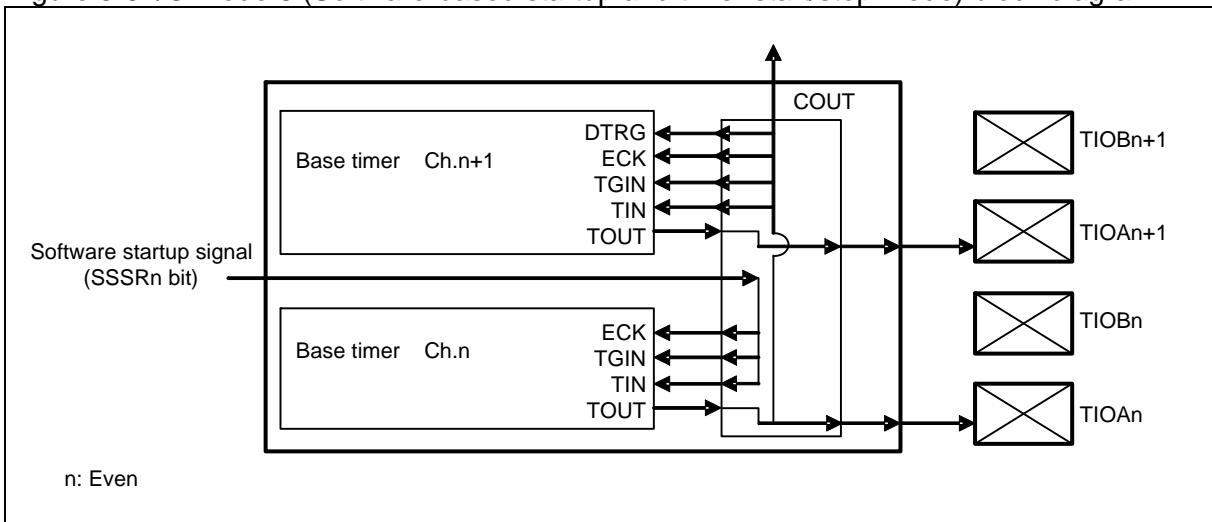


Table 3-22 shows signal connections in I/O mode 6.

Table 3-22 I/O mode 6 signal connections

Signal	Connected to
Ch.n TOUT signal	<ul style="list-style-type: none">Output from the TIOAn pinInput to Ch.n+1 as ECK/TGIN/TIN/DTRG signalsOutput to another channel as a COUT signal
Software startup signal (Write "1" to the SSSRn bit in the BTSSSR)	Input to Ch.n as ECK/TGIN/TIN signals
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin

n : Even

BTSSSR : Software-based Simultaneous Startup Register

When "1" is written to the Software-based Simultaneous Startup Register (BTSSSR) bit corresponding to the even channel you want to start up, a rising edge is input (ECK/TGIN/TIN signals) to the channel.

The start/stop timing of Ch.n is the same as that for I/O mode 4.

<Notes>

- Select the rising edge as a trigger input edge using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)
 - The odd channel stops operating when a falling edge is detected in the DTRG signal.
-

■ I/O mode 7 (Timer start mode)

This mode uses the output waveform from the even channel (TOUT signal) as input signals (ECK/TGIN/TIN signals) of the odd channel.

Table 3-23 shows the external pins used when this mode is selected.

Table 3-23 External pins used when I/O mode 7 is selected.

	Even channel	Odd channel
Number of input pins	1	Not used
Number of output pins	1	1

Table 3-24 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-24 External pin connections and input/output signals when I/O mode 7 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	Input	Even channel ECK/ TGIN/TIN *	Input to the even channel and used as one of the following signals: · External clock (ECK signal) · External startup trigger (TGIN signal) · Waveform to be measured (TIN signal)
TIOBn+1	-	-	Not used

n : Even

* : The usage of input waveforms (ECK/TGIN/TIN signals) differs depending on the Timer Control Register (TMCR) setting.

Figure 3-9 shows the block diagram of I/O mode 7 (Timer start mode).

Figure 3-9 I/O mode 7 (Timer start mode) block diagram

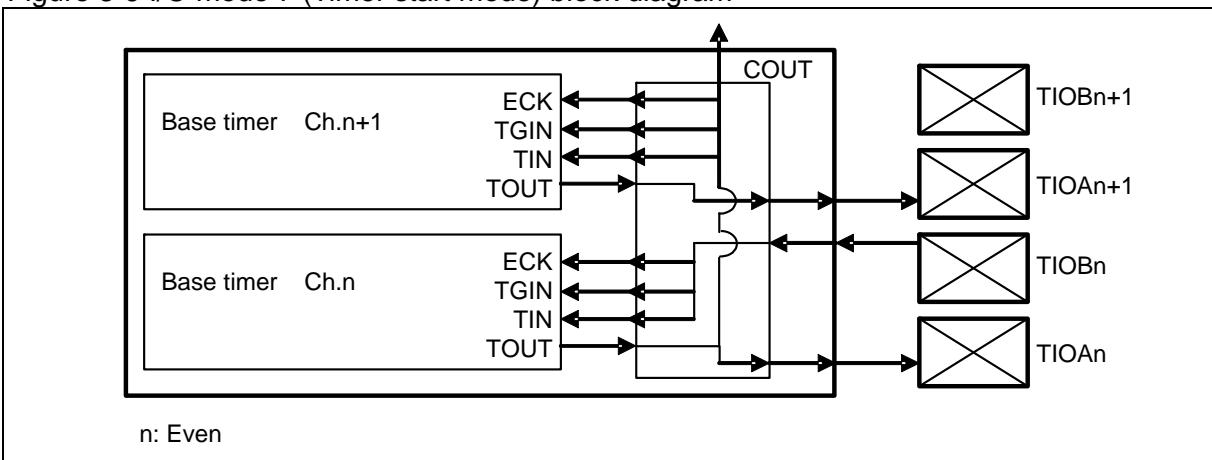


Table 3-25 shows signal connections in I/O mode 7.

Table 3-25 I/O mode 7 signal connections

Signal	Connected to
Ch.n TOUT signal	<ul style="list-style-type: none"> Output from the TIOAn pin Input to Ch.n+1 as ECK/TGIN/TIN signals Output to another channel as a COUT signal
Input signal from the TIOBn pin	Input to Ch.n as ECK/TGIN/TIN signals
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin

n : Even

The start timing of Ch.n is the same as that for I/O mode 4.

■ I/O mode 8 (Shared channel signal trigger and timer start/stop mode)

This mode inputs the COUT signal from channels below two channels as a CIN signal, and uses it as an external startup trigger (TGIN signal).

Table 3-26 shows the external pins used when this mode is selected.

Table 3-26 External pins used when I/O mode 8 is selected.

	Even channel	Odd channel
Number of input pins	Not used	
Number of output pins	1	1

Table 3-27 shows the internal signals to which the external pins connect, and signals input or output.

Table 3-27 External pin connections and input/output signals when I/O mode 8 is selected.

External pin	I/O	Connected to (internal signal)	Signal input/output
TIOAn	Output	Even channel TOUT	Outputs the even channel waveform
TIOAn+1	Output	Odd channel TOUT	Outputs the odd channel waveform
TIOBn	-	-	Not used
TIOBn+1	-	-	Not used

n : Even

Figure 3-10 shows the block diagram of I/O mode 8 (Shared channel signal trigger and timer start/stop mode).

Figure 3-10 I/O mode 8 (Shared channel signal trigger and timer start/stop mode) block diagram

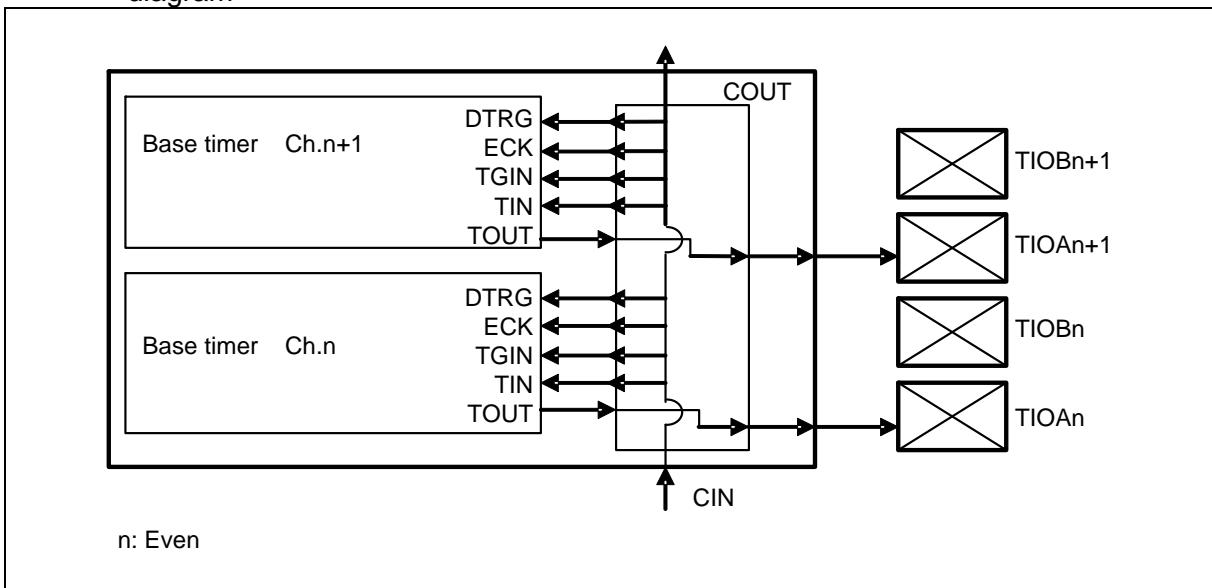


Table 3-28 shows signal connections in I/O mode 8.

Table 3-28 I/O mode 8 signal connections

Signal	Connected to
Ch.n TOUT signal	Output from the TIOAn pin
Ch.n+1 TOUT signal	Output from the TIOAn+1 pin
CIN signal *	<ul style="list-style-type: none"> · Input to Ch.n and Ch.n+1 as ECK/TGIN/TIN and DTRG signals · Output to another channel as a COUT signal

n : Even

* : The COUT signal from another channel is input as a CIN signal.

The following shows Ch.n-2/n-1 signals that can be input to ECK/TGIN/TIN of Ch.n/n+1.

- Signal that the peripheral clock generates by synchronizing TIOBn-2 input in I/O mode 2.
- Trigger signal input from Ch.n-4/n-3 in I/O mode 3.
- TIOAn-2 output in I/O mode 4.
- TIOAn-2 output in I/O mode 6.
- TIOAn-2 output in I/O mode 7.
- Trigger signal input from Ch.n-4/n-3 in I/O mode 8.

<Notes>

- The channels set to this mode use the COUT signal from lower 2 channels (n-2 and n-1) as a CIN signal.
(Example: If channels 2 and 3 are set to this mode, they use the COUT signal from channels 0 and 1.)
Therefore, channels 0 and 1 cannot be set to this mode.
 - Select the rising edge as a trigger input edge, for the channel set in this mode, using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)
However, do not enable this setting if the timer function is set to the 16/32-bit PWC timer using FMD2 to FMD0 bits in the Timer Control Register (TMCR) of the base timer (FMD2 to FMD0 are set to 0b100).
 - Base timer stops operating when a falling edge is detected in the DTRG signal.
-

4. Registers

This section provides the register list of the base timer I/O select function.

■ Base Timer I/O Select Function Registers

Table 4-1 Base timer I/O select function register list

Abbreviation	Register name	See
BTSEL0123	I/O Select Register	4.1
BTSEL4567	I/O Select Register	4.2
BTSSSR	Software-based Simultaneous Startup Register	4.3

4.1. I/O Select Register (BTSEL0123)

This register selects the I/O mode for channels 0 to 3 of the base timer.

■ Register configuration

bit	15	14	13	12	11	10	9	8
Field	SEL23_3	SEL23_2	SEL23_1	SEL23_0	SEL01_3	SEL01_2	SEL01_1	SEL01_0
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15:12] SEL23_3 to SEL23_0: I/O select bits for Ch.2/Ch.3

Bit 15	Bit 14	Bit 13	Bit 12	I/O select bit
0	0	0	0	I/O mode 0 (Standard 16-bit timer mode)
0	0	0	1	I/O mode 1 (Timer full mode)
0	0	1	0	I/O mode 2 (Shared external trigger mode)
0	0	1	1	I/O mode 3 (Shared channel signal trigger mode)
0	1	0	0	I/O mode 4 (Timer start/stop mode)
0	1	0	1	I/O mode 5 (Software-based simultaneous startup mode)
0	1	1	0	I/O mode 6 (Software-based startup and timer start/stop mode)
0	1	1	1	I/O mode 7 (Timer start mode)
1	0	0	0	I/O mode 8 (Shared channel signal trigger and timer start/stop mode)
Others				Setting disabled

[bit 11:8] SEL01_3 to SEL01_0: I/O select bits for Ch.0/Ch.1

Bit 11	Bit 10	Bit 9	Bit 8	I/O select bit
0	0	0	0	I/O mode 0 (Standard 16-bit timer mode)
0	0	0	1	I/O mode 1 (Timer full mode)
0	0	1	0	I/O mode 2 (Shared external trigger mode)
0	0	1	1	I/O mode 3 (Shared channel signal trigger mode)
0	1	0	0	I/O mode 4 (Timer start/stop mode)
0	1	0	1	I/O mode 5 (Software-based simultaneous startup mode)
0	1	1	0	I/O mode 6 (Software-based startup and timer start/stop mode)
0	1	1	1	I/O mode 7 (Timer start mode)
1	0	0	0	I/O mode 8 (Shared channel signal trigger and timer start/stop mode)
Others				Setting disabled

<Notes>

- Channels 0 and 1 are the lowest channels of the base timer, and cannot use the modes that use signal from lower channels. Therefore, the following modes cannot be selected for the channels:
 - I/O mode 3 (Shared channel signal trigger mode)
 - I/O mode 8 (Shared channel signal trigger and timer start/stop mode)
- Before rewriting this register, set the base timer to reset mode using the FMD2 to FMD0 bits in the Timer Control Register (TMCR) of the base timer. (Set FMD2 to FMD0 to 0b000.)

4.2. I/O Select Register (BTSEL4567)

This register selects the I/O mode for channels 4 to 7 of the base timer.

■ Register configuration

bit	15	14	13	12	11	10	9	8
Field	SEL67_3	SEL67_2	SEL67_1	SEL67_0	SEL45_3	SEL45_2	SEL45_1	SEL45_0
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15:12] SEL67_3 to SEL67_0: I/O select bits for Ch.6/Ch.7

Bit 15	Bit 14	Bit 13	Bit 12	I/O select bit
0	0	0	0	I/O mode 0 (Standard 16-bit timer mode)
0	0	0	1	I/O mode 1 (Timer full mode)
0	0	1	0	I/O mode 2 (Shared external trigger mode)
0	0	1	1	I/O mode 3 (Shared channel signal trigger mode)
0	1	0	0	I/O mode 4 (Timer start/stop mode)
0	1	0	1	I/O mode 5 (Software-based simultaneous startup mode)
0	1	1	0	I/O mode 6 (Software-based startup and timer start/stop mode)
0	1	1	1	I/O mode 7 (Timer start mode)
1	0	0	0	I/O mode 8 (Shared channel signal trigger and timer start/stop mode)
Others				Setting disabled

[bit 11:8] SEL45_3 to SEL45_0: I/O select bits for Ch.4/Ch.5

Bit 11	Bit 10	Bit 9	Bit 8	I/O select bit
0	0	0	0	I/O mode 0 (Standard 16-bit timer mode)
0	0	0	1	I/O mode 1 (Timer full mode)
0	0	1	0	I/O mode 2 (Shared external trigger mode)
0	0	1	1	I/O mode 3 (Shared channel signal trigger mode)
0	1	0	0	I/O mode 4 (Timer start/stop mode)
0	1	0	1	I/O mode 5 (Software-based simultaneous startup mode)
0	1	1	0	I/O mode 6 (Software-based startup and timer start/stop mode)
0	1	1	1	I/O mode 7 (Timer start mode)
1	0	0	0	I/O mode 8 (Shared channel signal trigger and timer start/stop mode)
Others				Setting disabled

<Note>

Before rewriting this register, set the base timer to reset mode using the FMD2 to FMD0 bits in the Timer Control Register (TMCR) of the base timer. (Set FMD2 to FMD0 to 0b000.)

4.3. Software-based Simultaneous Startup Register (BTSSSR)

This register starts up multiple base timer channels using software.

Up to 16 channels can be started simultaneously if the bits corresponding to the channel are set to "1".

■ Register configuration

bit	15	-	0
Field		SSSR15 to 0	
Attribute		W	
Initial value		X	

■ Register functions

[bit 15:0] SSSR15 to SSSR0: Software-based simultaneous startup bit

SSSRx	Software-based simultaneous startup bit
0	Writing "0" to this bit is invalid
1	Starts Ch.x of the base timer

x : 15 to 0

<Notes>

- Do not write to this register unless set to either of the following modes:
 - I/O mode 5 (Software-based simultaneous startup mode)
 - I/O mode 6 (Software-based startup and timer start/stop mode)(Even channels only)
- For the channel started up by using this register, select the rising edge as a trigger input edge using the EGS1 and EGS0 bits in the Timer Control Register (TMCR) of the base timer. (Set EGS1 and EGS0 to 0b01.)

Chapter: Base Timer

This chapter explains the functions and operations of the base timer.

1. Overview of Base Timer
2. Block Diagram Of Base Timer
3. Operations of the Base Timer
4. 32-bit mode operations
5. Base Timer Interrupt
6. Starting the DMA Controller (DMAC)
7. Base Timer Registers
8. Notes on using the base timer
9. Descriptions of base timer functions

1. Overview of Base Timer

The function of the base timer can be set to either the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, or 16/32-bit PWC timer using the FMD2, 1, and 0 bits in the Timer Control Register. The following provides an overview of each selectable timer function.

■ Relationship between mode settings and timer functions

Settings of FMD2, FMD1, and FMD0 bits	Function
0b000	Reset mode
0b001	16-bit PWM timer
0b010	16-bit PPG timer
0b011	16/32-bit reload timer
0b100	16/32-bit PWC timer

■ Reset mode

The reset mode is a status where the base timer macros are reset (with each register set to the initial value). Be sure to set this mode before switching to a different timer function or T32 bit setting. However, it is not necessary to set this mode before setting the timer function or T32 bit immediately after the macros are reset.

■ 16-bit PWM timer

This timer consists of a 16-bit down counter, a 16-bit data register with a cycle set buffer, a 16-bit compare register with a duty set buffer, and a pin controller.

The cycle and duty data is stored in a buffered register and thus can be rewritten while the timer is in operation.

The counter clock of the 16-bit down counter can be selected from eight internal clocks (1, 4, 16, 128, 256, 512, 1024, and 2048 frequency divisions of the machine clock) and three external events (detection of a rising edge, a falling edge, or both).

The one-shot mode where the counting stops at an underflow or the continuous mode where the counting is repeated after reloading can be selected.

The start event of the 16-bit PWM timer can be selected from a software trigger and three external events (detection of a rising edge, a falling edge, or both).

■ 16-bit PPG timer

This timer consists of a 16-bit down counter, a 16-bit data register for setting the HIGH width, a 16-bit data register for setting the LOW width, and a pin controller.

The count clock of the 16-bit down counter can be selected from eight internal clocks (1, 4, 16, 128, 256, 512, 1024, and 2048 frequency divisions of the machine clock) and three external events (detection of a rising edge, a falling edge, or both).

The one-shot mode where the counting stops at an underflow or the continuous mode where the counting is repeated after reloading can be selected.

The start event of the 16-bit PPG timer can be selected from a software trigger and three external events (detection of a rising edge, a falling edge, or both).

■ 16/32-bit reload timer

This timer consists of a 16-bit down counter, a 16-bit reload register, and a pin controller.

The count clock of the 16-bit down counter can be selected from eight internal clocks (1, 4, 16, 128, 256, 512, 1024, and 2048 frequency divisions of the machine clock) and three external events (detection of a rising edge, a falling edge, or both).

The one-shot mode where the counting stops at an underflow or the continuous mode where the counting is repeated after reloading can be selected.

The start event of the 16/32-bit reload timer can be selected from a software trigger and three external events (detection of a rising edge, a falling edge, or both).

■ 16/32-bit PWC timer

This timer consists of a 16-bit up counter, a measurement input pin, and a control register.

This timer measures the time between any events using an external pulse input.

The reference count clock can be selected from eight internal clocks (1, 4, 16, 128, 256, 512, 1024, and 2048 frequency divisions).

Measurement modes	HIGH pulse width (\uparrow to \downarrow) / LOW pulse width (\downarrow to \uparrow) Rising cycle (\uparrow to \uparrow) / Falling cycle (\downarrow to \downarrow) Edge interval measurement (\uparrow or \downarrow to \downarrow or \uparrow)
-------------------	--

An interrupt request can be generated when the measurement is completed.

One-time or continuous measurement can be selected.

2. Block Diagram Of Base Timer

Figure 2-1 to Figure 2-4 show block diagrams of the base timer in each mode.

Figure 2-1 Block diagram of 16-bit PWM timer

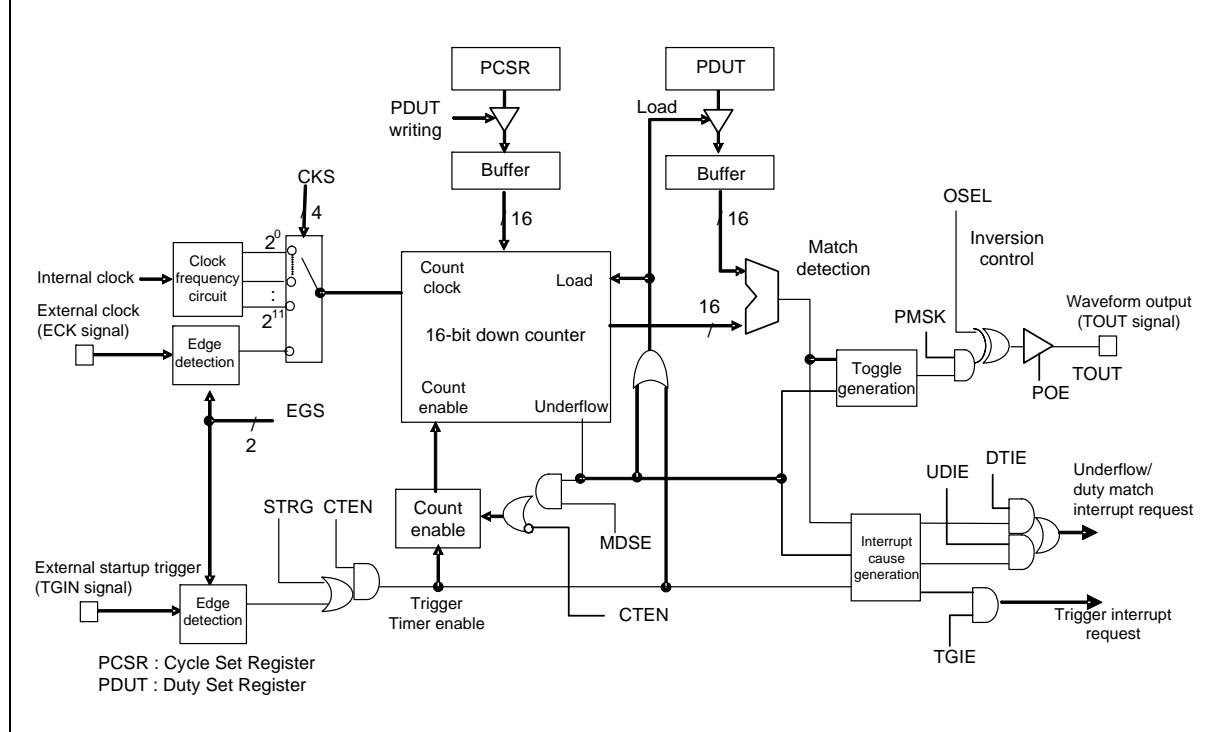


Figure 2-2 Block diagram of 16-bit PPG timer

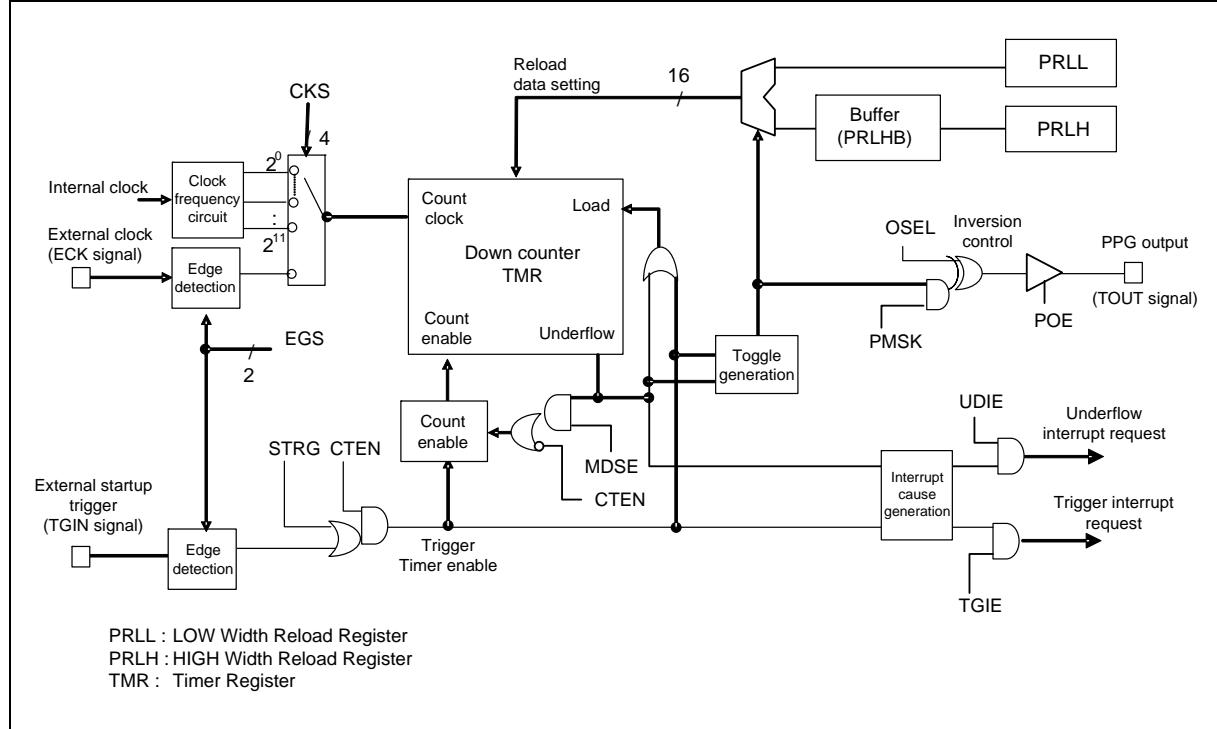


Figure 2-3 Block diagram of 16/32-bit reload timer (ch1 and ch0)

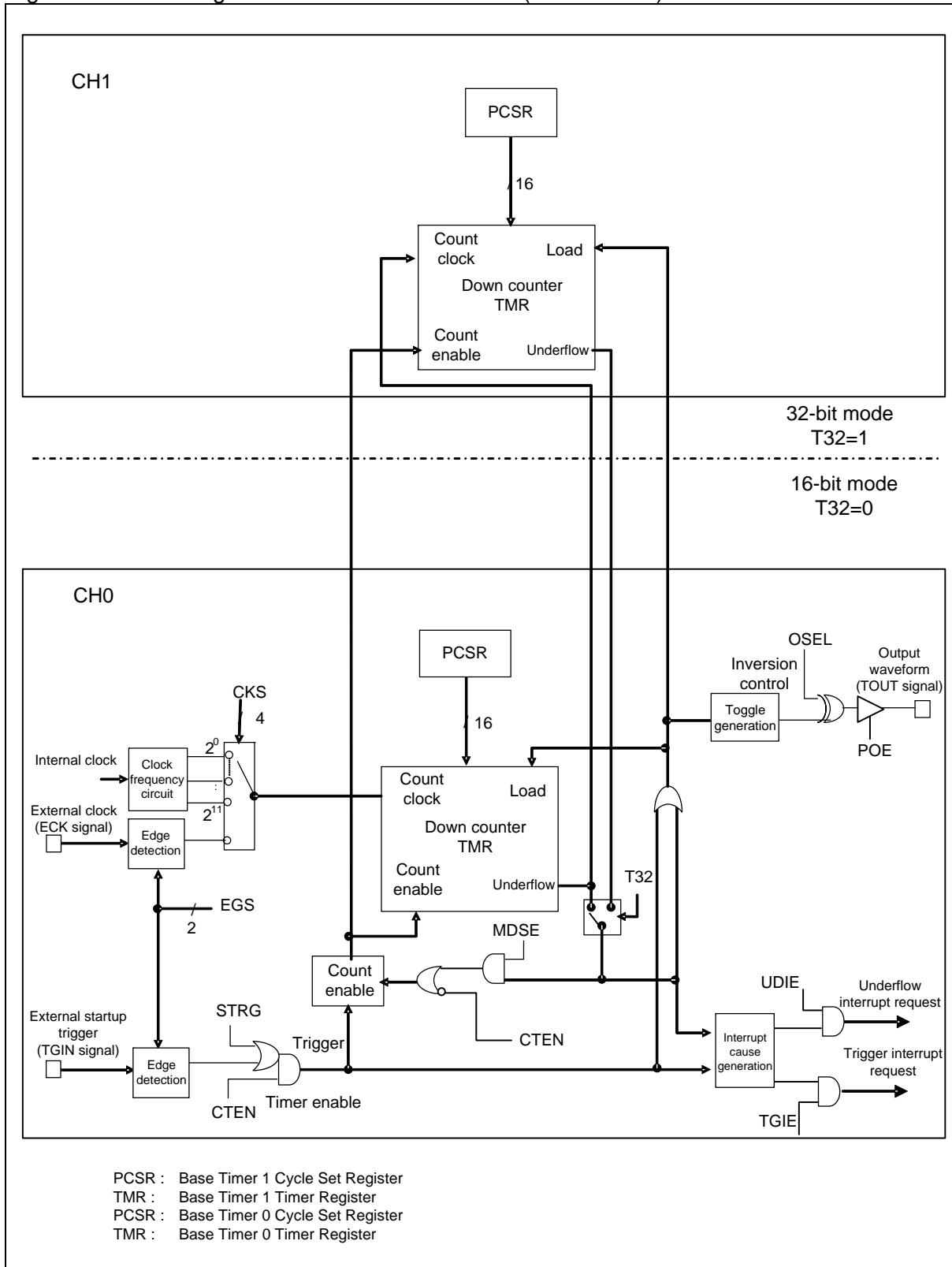
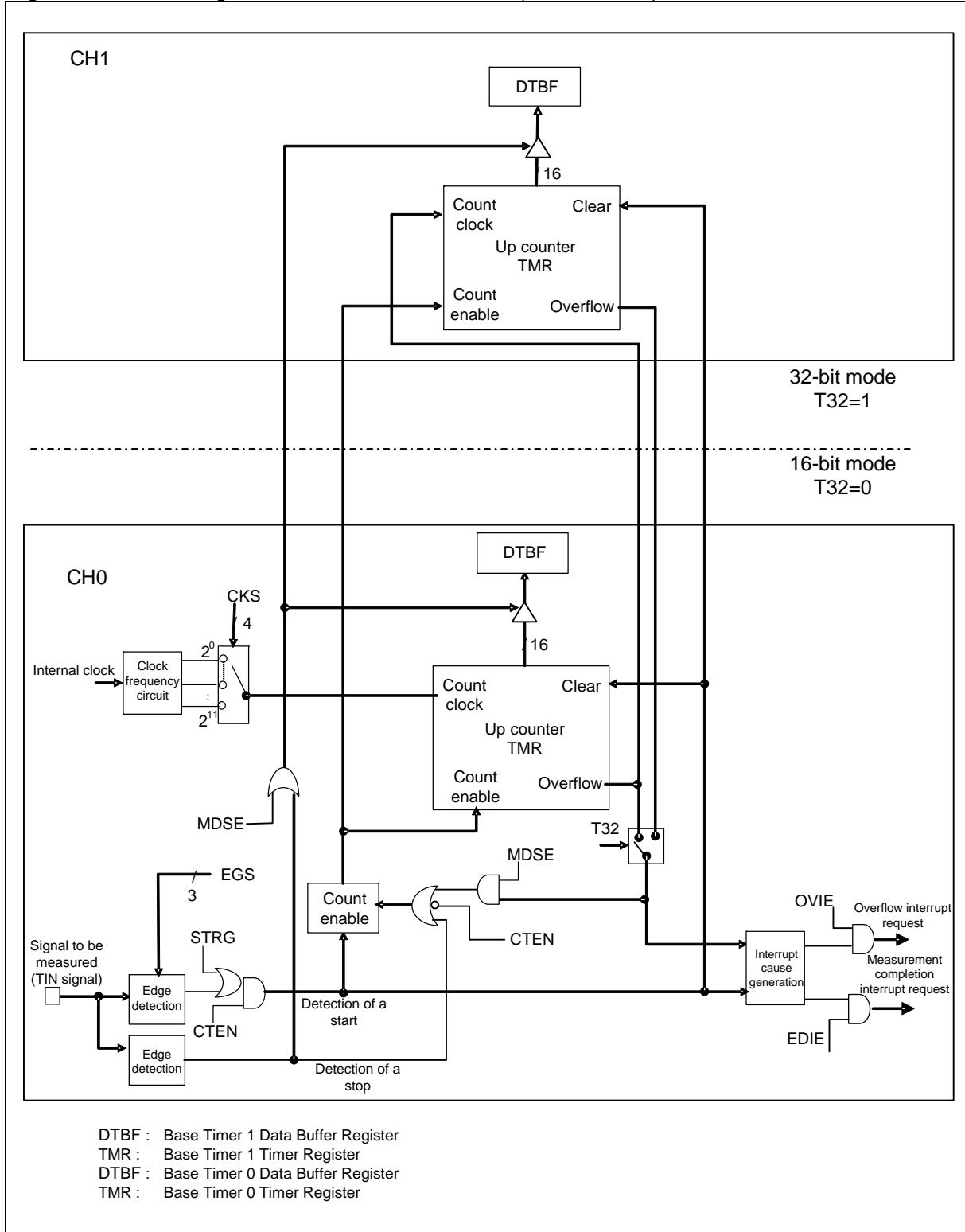


Figure 2-4 Block diagram of 16/32-bit PWC timer (ch1 and ch0)



3. Operations of the Base Timer

This section explains operations of the base timer.

■ Operations of the base timer

● Reset mode

The reset mode is a status where the base timer macros are reset (with each register set to the initial value). Be sure to set this mode before switching to a different timer function or T32 bit setting. However, it is not necessary to set this mode before setting the timer function or T32 bit immediately after the macros are reset. In a 32-bit mode, setting this mode for the even channel also resets the odd channel. It is not necessary to set the reset mode for the odd channel.

● 16-bit PWM timer

When triggered, the 16-bit PWM timer starts decrementing from the cycle set value. First, it outputs a LOW level pulse. When the 16-bit down counter matches the value set in the PWM Duty Set Register, the output inverts to the HIGH level. Then, the output inverts again to the LOW level when a counter underflow occurs. This can generate waveforms with any cycle and duty.

● 16-bit PPG timer

When triggered, the 16-bit PPG timer starts decrementing from the value set in the LOW Width Reload Register. First, it outputs a LOW level pulse. The output inverts to the HIGH level upon an underflow. Then, it starts decrementing from the value set in the HIGH Width Reload Register. The output inverts to the LOW level when an underflow occurs. This can generate waveforms having any LOW and HIGH widths.

● 16-bit reload timer

When triggered, the 16-bit reload timer starts decrementing from the cycle set value. When an underflow occurs on the 16-bit down counter, an interrupt flag is set. The output is either the toggle output where the level inverts according to the MDSE bit setting as an underflow occurs or the pulse output where the level is HIGH at the start of counting and LOW at the occurrence of an underflow.

● 32-bit reload timer

This timer has the same basic operations as the 16-bit reload timer. However, it uses two channels, even and odd, to operate as a 32-bit reload timer. The even channel operates as a lower 16-bit timer and the odd channel as an upper 16-bit timer. The interrupt control and output waveform control are defined by the settings for the even channel only. When setting the cycle, first write it in the upper register (odd channel) and then in the lower register (even channel).

When reading the timer value, read it from the lower register (even channel) and then from the upper channel (odd channel).

● **16-bit PWC timer**

The PWC timer starts the 16-bit up counter with input of the specified measurement start edge and stops the counter with detection of a measurement end edge. The value counted in between is stored as a pulse width in the data buffer register.

● **32-bit PWC timer**

This timer has the same basic operations as the 16-bit PWC timer. However, it uses two channels, even and odd, to operate as a 32-bit PWC timer. The even channel operates as a lower 16-bit counter and the odd channel as an upper 16-bit counter. The interrupt control is defined by the settings for the even channel only. When reading the measured or count value, read it from the lower register (even channel) and then from the upper channel (odd channel).

4. 32-bit mode operations

Using two channels, the reload timer and PWC provide 32-bit mode operations. This section explains the basic functions and operations of the 32-bit mode functions.

■ 32-bit mode functions

This function enables the operations of a 32-bit data reload timer or 32-bit data PWC timer by combining two channels of base timers. Since the upper 16-bit timer counter value in the odd channel is read together with the lower 16-bit timer counter value in the even channel, the timer counter value can be read during operation.

■ 32-bit mode settings

First, set "0b000" to set the reset mode to reset the status of the FMD2, FMD1, and FMD0 bits in the TMCR register of the even channel. Then, as carried out for in 16-bit mode, select the reload or PWC timer and set the operation. By writing "1" also to the T32 bit in the TMCR register, the 32-bit operation mode is set. Do not change "0" for the T32 bit in the odd channel. It is also not necessary to set it to reset mode. For the reload timer, set the reload value of the upper 16 of 32 bits in the PWM Cycle Set Register of the odd channel. Then, set the reload value of the lower 16 bits in the PWM Cycle Set Register of the even channel.

Because transition to 32-bit operation mode is reflected immediately after the T32 bit is written, stop the counting before changing the settings in each channel.

To change from 32-bit mode to 16-bit mode, set it to reset mode by setting "0b000" for the FMD2, FMD1, and FMD0 bits in the TMCR register of the even channel. This resets the status of both even and odd channels, enabling settings in 16-bit mode in each channel.

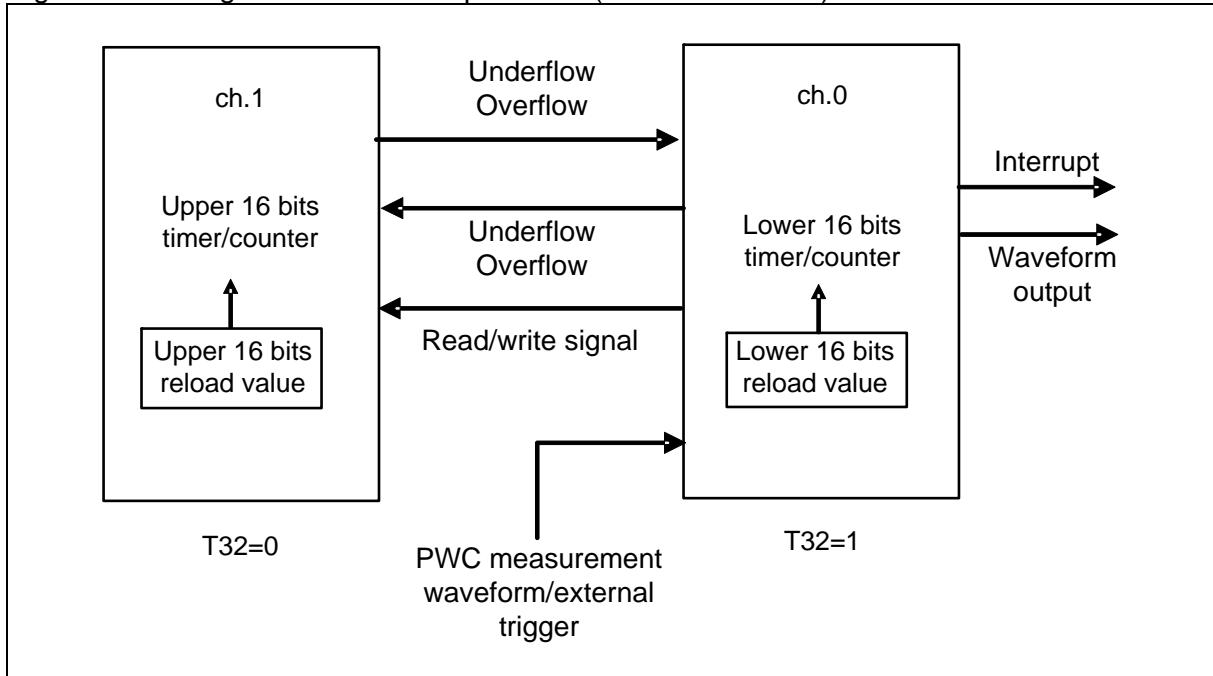
■ 32-bit mode operations

After transition to 32-bit mode, if the reload or PWC timer is started by control of the even channel, the timer/counter in the even channel operates with the lower 16 bits. Also, the time/counter in the odd channel operates with the upper 16 bits.

The operations in 32-bit mode are defined by the settings for the even channel. For this reason, the settings for the odd channel (except the Cycle Set Register for the reload timer) are ignored. For the timer start, waveform output, and interrupt signal functions, settings for the even channel are also applied (the even channel is masked and fixed to "LOW").

Figure 4-1 shows the configuration of ch.0 and ch.1.

Figure 4-1 Configuration of 32-bit operations (for ch.0 and ch.1)



5. Base Timer Interrupt

This section provides a list of interrupt request flags, interrupt enable bits, and interrupt causes for each function of the base timer.

■ Interrupt control bits and interrupt causes for each function

Table 5-1 shows the interrupt control bits and interrupt causes for each function.

Table 5-1 Interrupt control bits and interrupt causes in each mode

	Status Control Register (STC)			
	Interrupt request flag bit	Interrupt request enable bit	Interrupt causes	Interrupt cause output signal
PWM timer function	UDIR: bit 0	UDIE: bit 4	Detection of an underflow	IRQ0
	DTIR: bit 1	DTIE: bit 5	Detection of a match in duty	
	TGIR: bit 2	TGIE: bit 6	Detection of a timer start trigger	IRQ1
PPG timer function	UDIR: bit 0	UDIE: bit 4	Detection of an underflow	IRQ0
	TGIR: bit 2	TGIE: bit 6	Detection of a timer start trigger	IRQ1
Reload timer function	UDIR: bit 0	UDIE: bit 4	Detection of an underflow	IRQ0
	TGIR: bit 2	TGIE: bit 6	Detection of a timer start trigger	IRQ1
PWC timer function	OVIR: bit 0	OVIE: bit 4	Detection of an overflow	IRQ0
	EDIR: bit 2	EDIE: bit 6	Detection of the completion of measurement	IRQ1

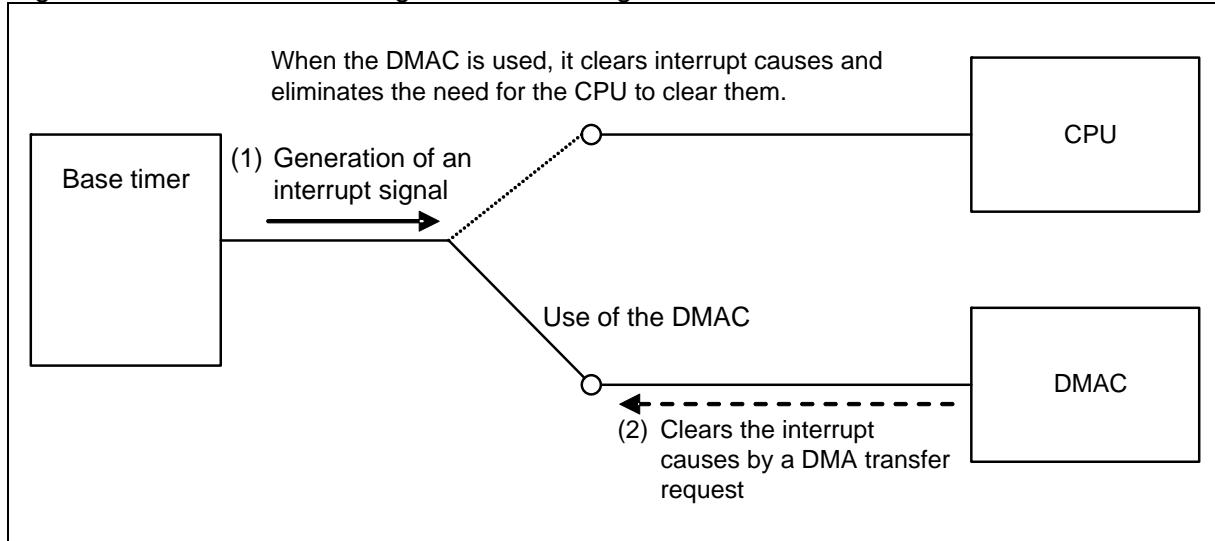
6. Starting the DMA Controller (DMAC)

The DMAC can be started using the generation of an interrupt request by the base timer.

■ DMA transfer operation using interrupt causes of the base timer

The DMAC can be started using the generation of an interrupt cause by the base timer. Figure 6-1 gives an overview of starting the DMAC using the base timer.

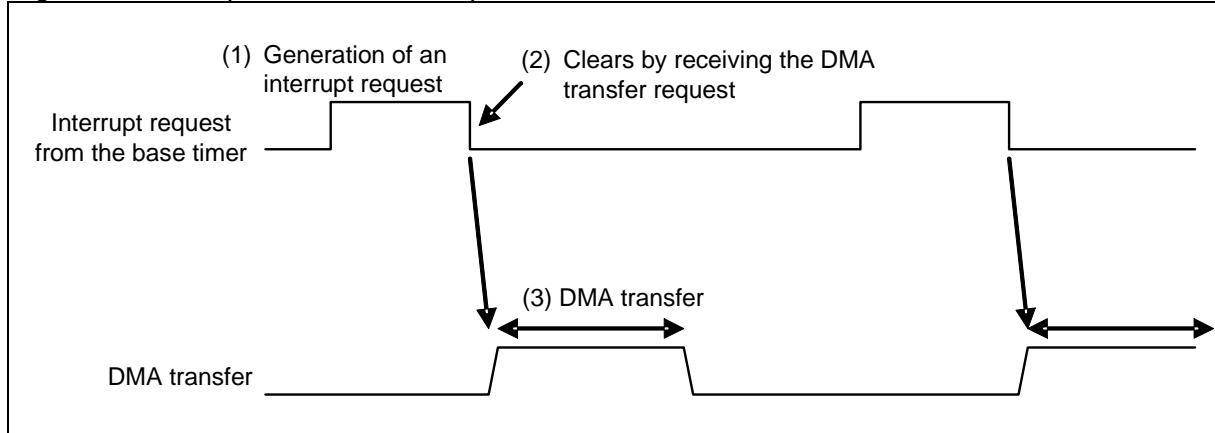
Figure 6-1 Overview of starting the DMAC using the base timer



Before starting the DMAC using the base timer, configure the DMAC. For settings and details on the DMAC, see Chapters "DMAC" and "Interrupt".

Figure 6-2 gives an example of a DMA transfer operation using an interrupt request by the base timer.

Figure 6-2 Example DMA transfer operation



7. Base Timer Registers

This section provides register lists of the base timer in each mode.

■ List of registers used when the 16-bit PWM timer is selected

Table 7-1 List of registers used when the 16-bit PWM timer is selected

Abbreviation	Register name	See
TMCR	Timer Control Register	9.1.6
TMCR2	Timer Control Register 2	9.1.6
STC	Status Control Register	9.1.6
PCSR	PWM Cycle Set Register	9.1.7
PDUT	PWM Duty Set Register	9.1.8
TMR	Timer Register	9.1.9

■ List of registers used when the 16-bit PPG timer is selected

Table 7-2 List of registers used when the 16-bit PPG timer is selected

Abbreviation	Register name	See
TMCR	Timer Control Register	9.2.6
TMCR2	Timer Control Register 2	9.2.6
STC	Status Control Register	9.2.6
PRLL	LOW Width Reload Register	9.2.7
PRLH	HIGH Width Reload Register	9.2.8
TMR	Timer Register	9.2.9

■ List of registers used when the reload timer is selected

Table 7-3 List of registers used when the reload timer is selected

Abbreviation	Register name	See
TMCR	Timer Control Register	9.3.3
TMCR2	Timer Control Register 2	9.3.3
STC	Status Control Register	9.3.3
PCSR	PWM Cycle Set Register	9.3.4
TMR	Timer Register	9.3.5

■ List of registers used when the PWC timer is selected

Table 7-4 List of registers used when the PWC timer is selected

Abbreviation	Register name	See
TMCR	Timer Control Register	9.4.2
TMCR2	Timer Control Register 2	9.4.2
STC	Status Control Register	9.4.2
DTBF	Data Buffer Register	9.4.3

8. Notes on using the base timer

This section provides notes on using the base timer.

■ Notes on setting the program common to each timer

- It is prohibited to rewrite the following bits in the TMCR2 and TMCR registers during operation.
Rewriting of the bits must be performed before starting or after stopping the operation.
 - [TMCR2 bit 8], [TMCR bit 14, 13, 12] CKS3 to CKS0 : Clock selection bits
 - [bit 10, 9, 8] EGS2, EGS1, EGS0 : Measurement edge selection bits
 - [bit 7] T32 : 32-bit timer selection bit
 - (When the reload timer PWC function is selected)
 - [bit 6, 5, 4] FMD2 to FMD0 : Timer function selection bits
 - [bit 2] MDSE : Measurement mode (one-shot/continuous) selection bit
- When the FMD2 to FMD0 bits in the TMCR register are set to reset mode with "0b000", all registers of the base timer are initialized. Therefore, all registers must be set again.
- When the FMD2 to FMD0 bits in the TMCR register are set to reset mode with "0b000", settings for the bits other than the FMD2 to FMD0 bits in the TMCR register are ignored and initialized.

■ Notes on using the 16-bit PWM/PPG/reload timer

- If the interrupt request flag set timing coincides the clear timing, the flag set operation takes precedence and the clear operation is not performed.
- If the load timing and count timing of the down counter coincide, the load operation takes precedence.
- Set the timer function with the FMD2, FMD1, and FMD0 bits in the TMCR register, and then set the cycle, duty, HIGH width, and LOW width.
- In one-shot mode, if a restart is detected at the end of counting, the count value is reloaded and the restart operation is started.

■ Notes on using the PWC timer

- If the count start enable bit (CTEN) is set to "1", the counter is cleared. As the result, the data existed in the counter before the start is enabled becomes invalid.
- If the setting for PWC mode (FMD = 0b100) and the setting for starting measurement (CTEN = 1) are performed simultaneously in system reset/reset mode, the resultant operation may depend on the status of the last measurement signal.
- In continuous measurement mode, if a measurement start edge is detected at the same time a restart is set, the counting is started immediately from "0x0001".
- If a restart is performed after the count operation has been started, the following operations may occur depending on the timing.
 - If it coincides with a measurement end edge in pulse width one-shot measurement mode:
The timer is restarted and waits for detection of a measurement start edge. However, a measurement end flag (EDIR) is set.
 - If it coincides with a measurement end edge in pulse width continuous measurement mode:
The timer is restarted and waits for detection of a measurement start edge. However, a measurement end flag (EDIR) is set and the measurement result at the time is transferred to the DTBF.

When restarting the timer during operation, pay attention to flag operations as described above and use the interrupt control.

9. Descriptions of base timer functions

This section explains each function of the base timer.

■ Base timer functions

1. PWM timer function
2. PPG timer function
3. Reload timer function
4. PWC timer function

9.1. PWM timer function

The function of the base timer can be set to either the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, or 16/32-bit PWC timer using the FMD2, 1, and 0 bits in the Timer Control Register. This section explains the timer functions available when PWM is set.

1. 16-bit PWM timer operations
2. One-shot operation
3. Interrupt causes and timing chart
4. Output waveforms
5. PWM timer operation flowchart
6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWM timer is selected
7. PWM Cycle Set Register (PCSR)
8. PWM Duty Set Register (PDUT)
9. Timer Register (TMR)

9.1.1.16-bit PWM timer operations

In PWM timer operations, waveforms in the specified cycle from the detection of a trigger can be output in one-shot or continuously.

The cycle of the output pulse can be controlled by changing the PCSR value.

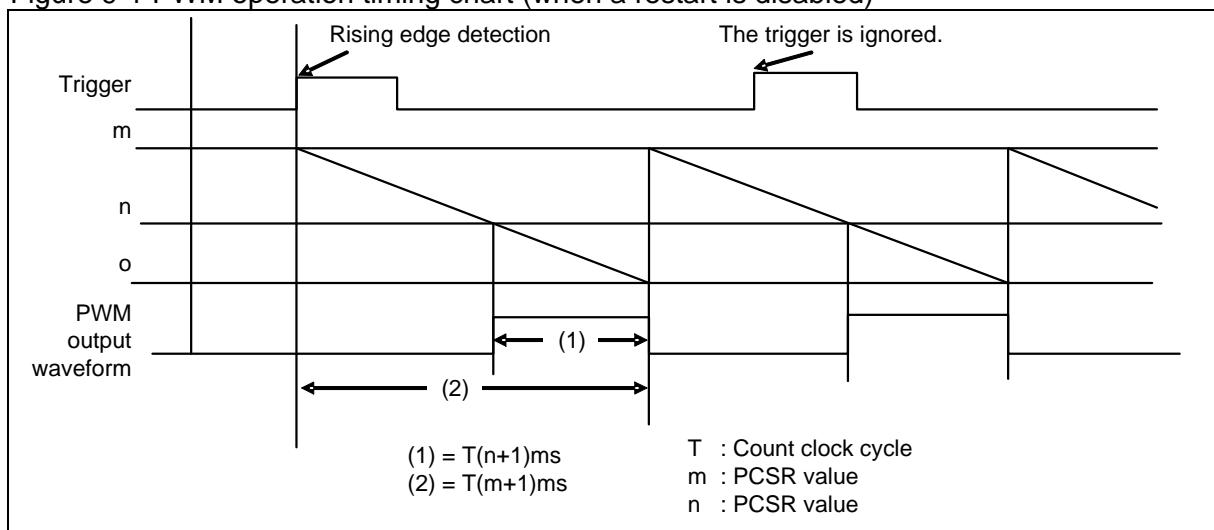
The duty ration can be controlled by changing the PDUT value.

After writing data to the PCSR, be sure to write it to the PDUT.

■ Continuous operation

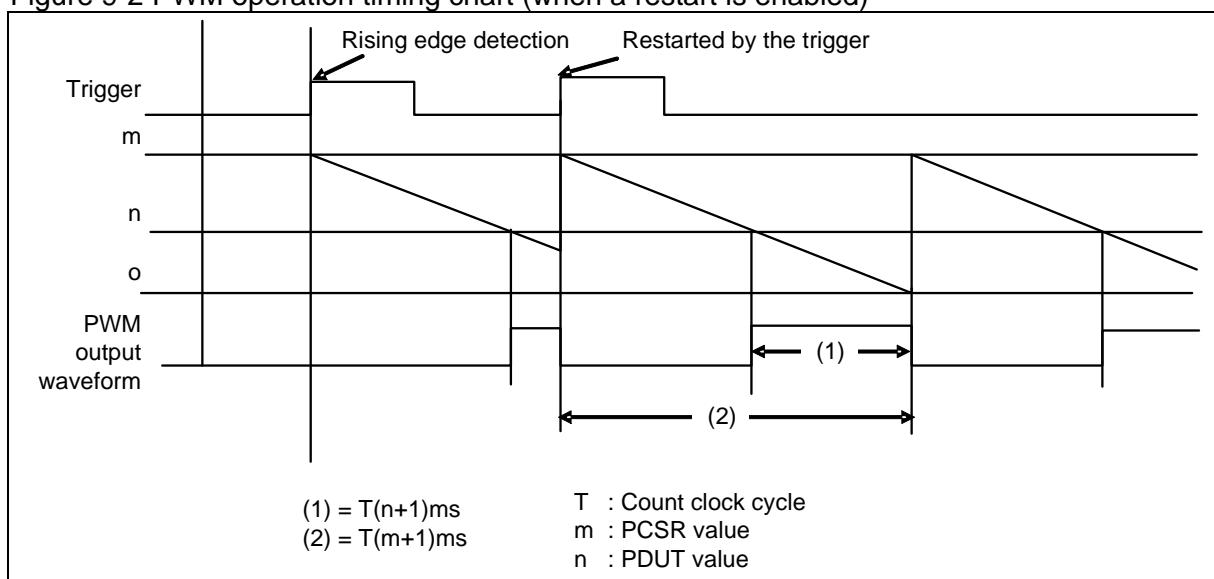
● When a restart is disabled (RTGEN = 0)

Figure 9-1 PWM operation timing chart (when a restart is disabled)



● When a restart is enabled (RTGEN = 1)

Figure 9-2 PWM operation timing chart (when a restart is enabled)



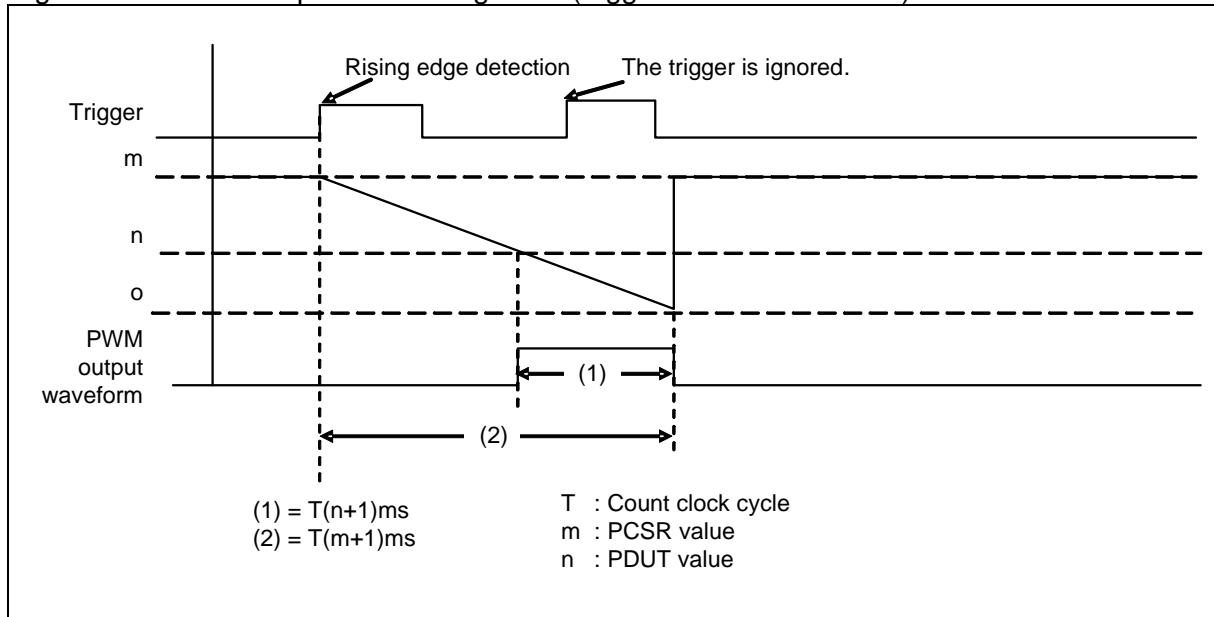
9.1.2. One-shot operation

In one-shot operation, a single pulse of any width can be output using a trigger. When a restart is enabled, the counter is reloaded when an edge is detected during operation.

■ One-shot operation

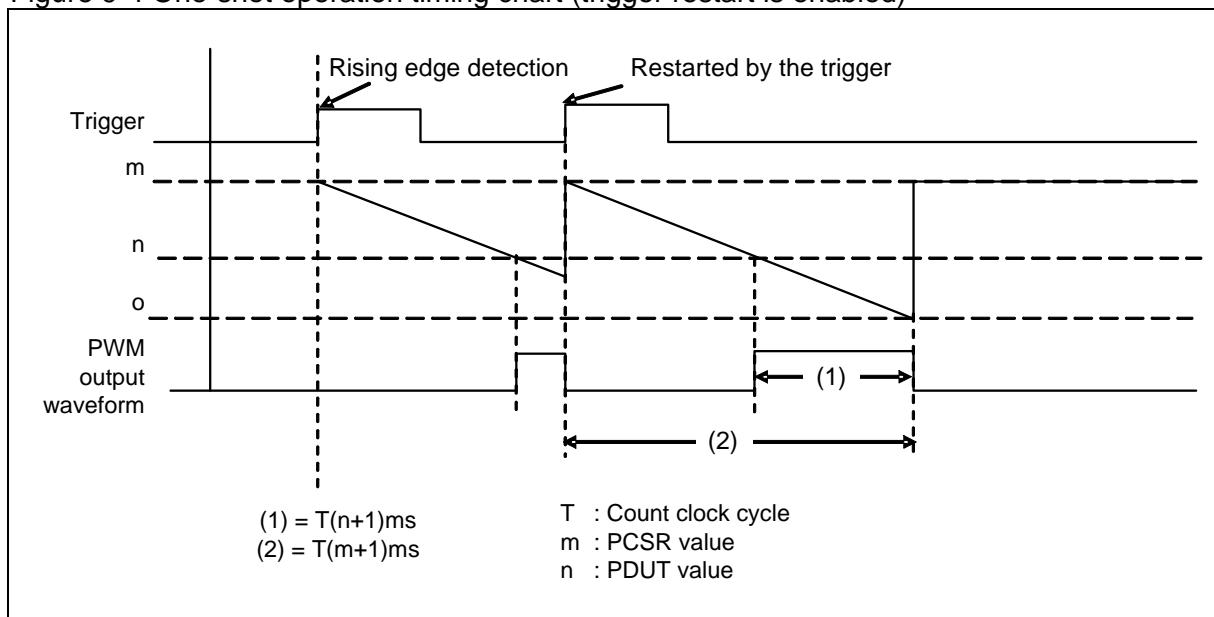
- When a restart is disabled (RTGEN = 0)

Figure 9-3 One-shot operation timing chart (trigger restart is disabled)



- When a restart is enabled (RTGEN = 1)

Figure 9-4 One-shot operation timing chart (trigger restart is enabled)



9.1.3. Interrupt causes and timing chart

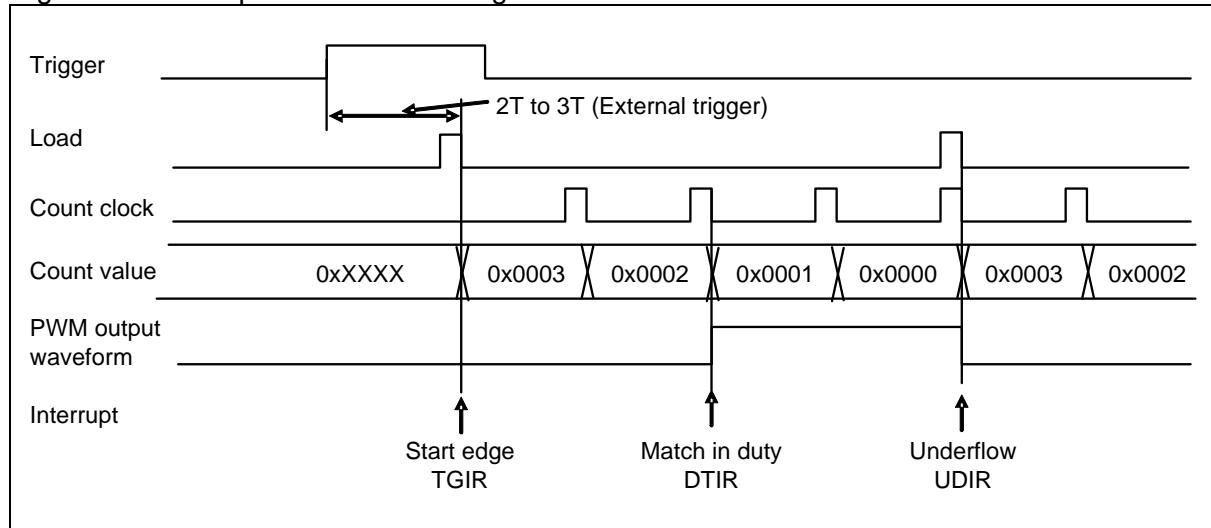
This section explains interrupt causes and a timing chart.

■ Interrupt causes and timing chart (PWM output: Normal polarity)

As a time from trigger input to loading of the counter value, T is required for software triggering or 2T to 3T (T: machine cycle) for external triggering.

Figure 9-5 shows the interrupt causes and a timing chart where the cycle set value = 3 and duty value = 1.

Figure 9-5 Interrupt causes and timing chart of the PWM timer



9.1.4. Output waveforms

This section explains the PWM output.

■ How to make an all-LOW or all-HIGH PWM output

Figure 9-6 shows how to make all-LOW PWM output and Figure 9-7 shows how to make all-HIGH output.

Figure 9-6 Example of outputting all-LOW level waveforms as PWM output

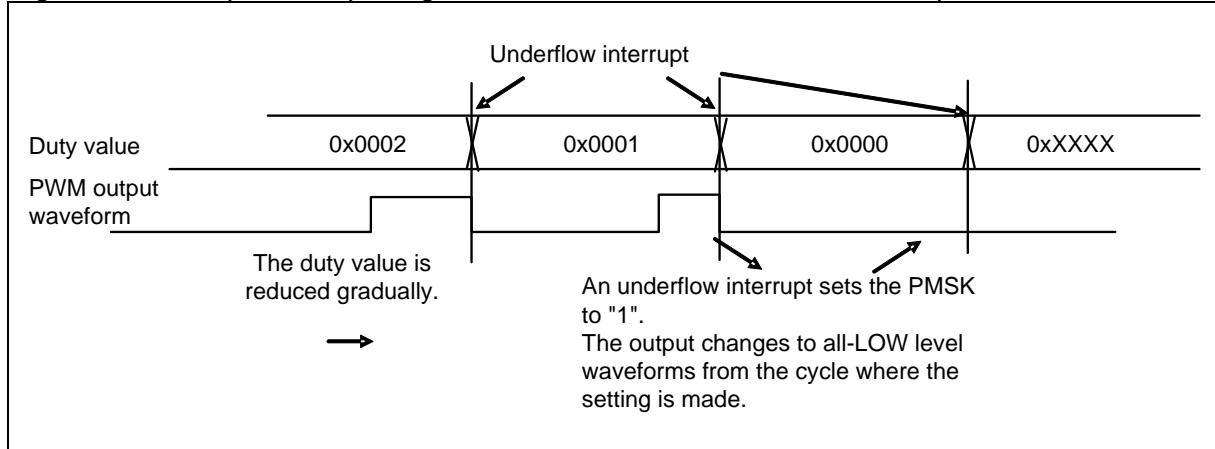
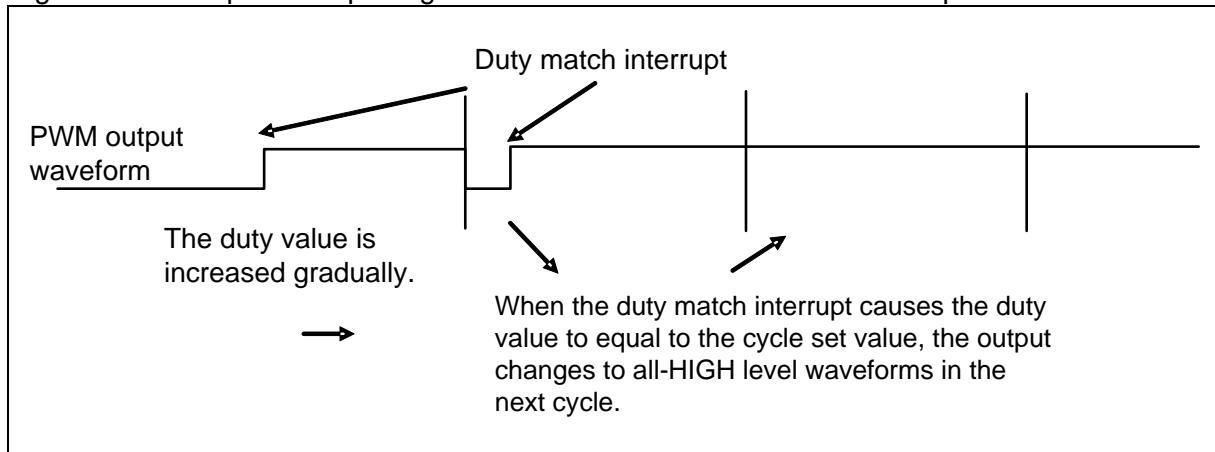


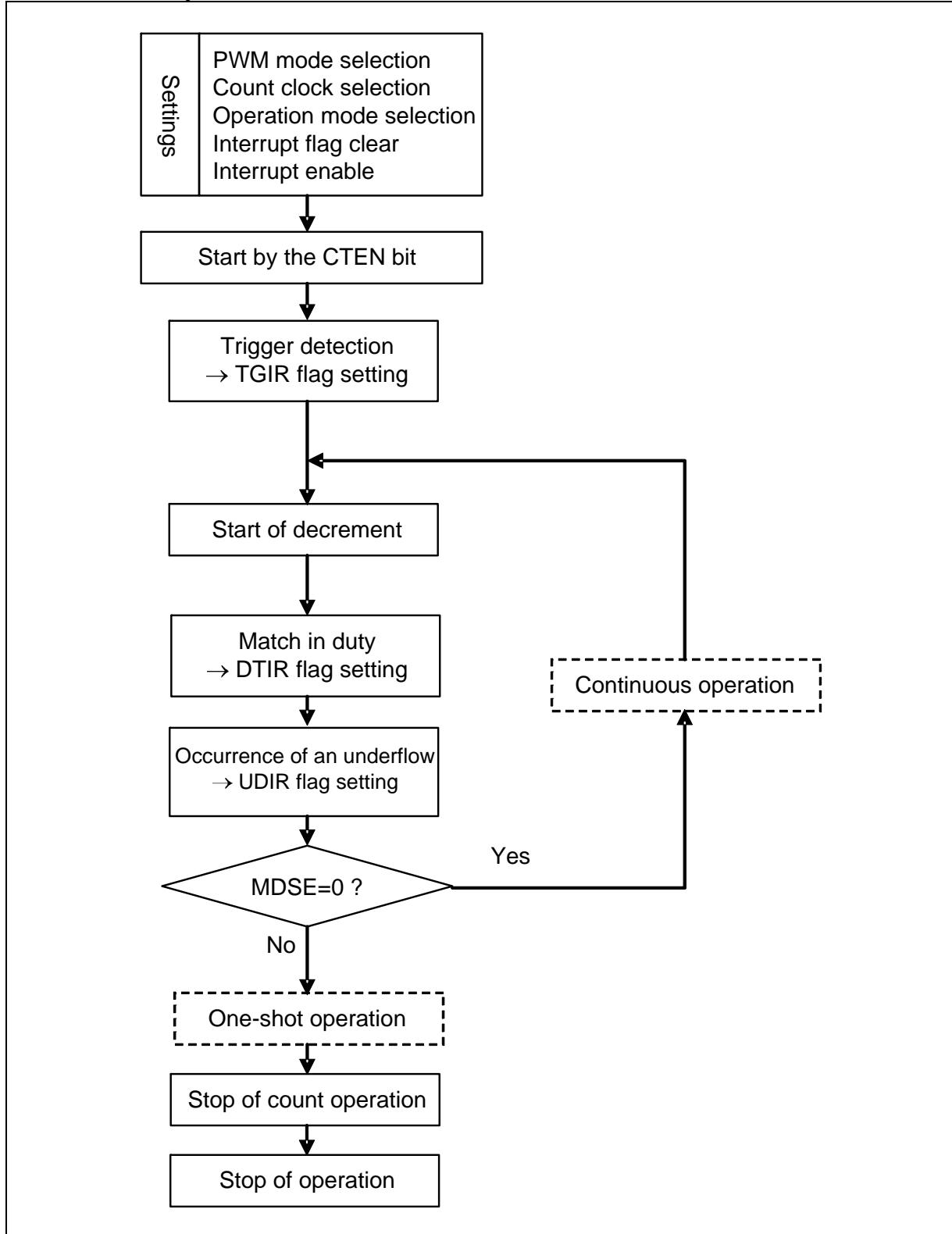
Figure 9-7 Example of outputting all-HIGH level waveforms as PWM output



9.1.5. PWM timer operation flowchart

This section provides an operation flowchart of the PWM timer.

■ PWM timer operation flowchart



9.1.6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWM timer is selected

The Timer Control Register (TMCR) controls the PWM timer. Note that some bits cannot be rewritten while the PWM timer is in operation.

■ Timer Control Register (High-order bytes of TMCR)

bit	15	14	13	12	11	10	9	8
Field	res	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 15] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 14:12, TMCR2: bit 8] CKS3 to CKS0: Count clock selection bit

- Select the count clock for the 16-bit down counter.
- Changes to the count clock setting are applied immediately. For this reason, changes to CKS3 through CKS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	ϕ
0	0	0	1	$\phi / 4$
0	0	1	0	$\phi / 16$
0	0	1	1	$\phi / 128$
0	1	0	0	$\phi / 256$
0	1	0	1	External clock (rising edge event)
0	1	1	0	External clock (falling edge event)
0	1	1	1	External clock (both edge event)
1	0	0	0	$\phi / 512$
1	0	0	1	$\phi / 1024$
1	0	1	0	$\phi / 2048$
Others				Setting disabled

[bit 11] RTGEN: Restart enable bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Restart disabled
1	Restart enabled

[bit 10] PMSK: Pulse output mask bit

- This bit controls the output level of PWM output waveforms.
- When this bit is set to "0", PWM waveforms are output as they are.
- When this bit is set to "1", the PWM output is masked with LOW output regardless of the cycle and duty set values.

<Note>

When OSEL in bit 3 is set to inverted output, setting PMSK to "1" masks the output with HIGH.

Bit	Description
0	Normal output
1	Fixed to LOW output

[bit 9:8] EGS1, EGS0: Trigger input edge selection bits

- These bits select a valid edge for input waveforms as an external start cause and set the trigger condition.
- When the initial value or "0b00" is set, the timer is not started by external waveforms because the setting means that no valid edge is selected for input waveforms.

<Note>

If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

- Changes to EGS1 or EGS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 9	Bit 8	Description
0	0	Trigger input disabled
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

■ Timer Control Register (Low-order bytes of TMCR)

bit	7	6	5	4	3	2	1	0
Field	res	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 6:4] FMD2 to FMD0: Timer function selection bits

- These bits select the timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b001", the PWM function is selected.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 6	Bit 5	Bit 4	Description
0	0	0	Reset mode
0	0	1	Selection of the PWM function
0	1	0	Selection of the PPG function
0	1	1	Selection of the reload timer function
1	0	0	Selection of the PWC function
1	0	1	Setting disabled
1	1	0	
1	1	1	

[bit 3] OSEL: Output polarity specification bit

- This bit sets the polarity of the PWM output.

Polarity	After reset	Match in duty	Underflow
Normal	LOW output		
Inverted	HIGH output		

Bit	Description
0	Normal polarity
1	Inverted polarity

[bit 2] MDSE: Mode selection bit

- This bit selects continuous pulse output or one-shot pulse output.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit 1] CTEN: Count operation enable bit

- This bit enables the operation of the down counter.
- When the counter is in operation enabled status (the CTEN bit is "1"), writing "0" to this bit stops the counter.

Bit	Description
0	Stop
1	Operation enabled

[bit 0] STRG: Software trigger bit

- When the CTEN bit is "1", writing "1" to the STRG bit enables software triggering.
- The read value of the STRG bit is always "0".

<Notes>

- Software triggering is also enabled when "1" is written to the CTEN and STRG bits simultaneously.
- If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

Bit	Description
0	Invalid
1	Start triggered by software

■ Timer Control Register 2 (TMCR2)

bit	15	14	13	12	11	10	9	8
Field				res				CKS3
Attribute				R/W				R/W
Initial value				0b0000000				0

Note: This register is placed above the STC register.

[bit 15:9] res: Reserved bits

The read value is "0".

Set "0" to this bit.

[bit 8] CKS3: Count clock selection bit

See "Count clock selection bit" in "9.1.6 Timer Control Register (TMCR)".

■ Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	DTIE	UDIE	res	TGIR	DTIR	UDIR
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is placed in the upper bytes of this register.

[bit 7] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 6] TGIE: Trigger interrupt request enable bit

- This bit controls interrupt requests of bit 2 TGIR.
- When the TGIE bit is enabled, setting bit 2 TGIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 5] DTIE: Duty match interrupt request enable bit

- This bit controls interrupt requests of bit 1 DTIR.
- When the DTIE bit is enabled, setting bit 1 DTIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 4] UDIE: Underflow interrupt request enable bit

- This bit controls interrupt requests of bit 0 UDIR.
- When the UDIE bit is enabled, setting bit 0 UDIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 3] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 2] TGIR: Trigger interrupt request bit

- When a software trigger or trigger input is detected, the TGIR bit is set to "1".
- The TGIR bit is cleared by writing "0".
- Even if "1" is written to the TGIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

[bit 1] DTIR: Duty match interrupt request bit

- When the count value matches the duty set value, the DTIR bit is set to "1".
- The DTIR bit is cleared by writing "0".
- Even if "1" is written to the DTIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

[bit 0] UDIR: Underflow interrupt request bit

- When a count value underflow from 0x0000 to 0xFFFF occurs, the UDIR bit is set to "1".
- The UDIR bit is cleared by writing "0".
- Even if "1" is written to the UDIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

9.1.7. PWM Cycle Set Register (PCSR)

The PWM Cycle Set Register (PCSR) is a buffered register for setting the cycle. Transfer to the Timer Register is performed at startup and underflow.

bit	15	0
Field	PCSR [15:0]	
Attribute	R/W	
Initial value	0xFFFF	

This is a buffered register for setting the cycle. Transfer to the Timer Register is performed at startup and underflow.

When initializing or rewriting the PWM Cycle Set Register, be sure to perform writing to the PWM Duty Set Register after performing writing to the PWM Cycle Set Register.

- Access the PCSR register with 16-bit data.
- Set the cycle for the PCSR register after setting the PWM function using the FMD2, FMD1, and FMD0 bits in the TMCR register.

9.1.8. PWM Duty Set Register (PDUT)

The PWM Duty Set Register (PDUT) is a buffered register for setting the duty. Transfer from the buffer is performed at an underflow.

bit	15	0
Field	PDUT [15:0]	
Attribute	R/W	
Initial value	0xFFFF	

This is a buffered register for setting the duty. Transfer from the buffer is performed at an underflow.

When the cycle set register value is set equal to the duty set register value, an all-HIGH pulse is output under normal polarity and an all-LOW pulse is output under inverted polarity.

Do not set a value that makes BTPSCR < PDUT. The PWM output becomes undefined.

- Access the PDUT register with 16-bit data.
- Set the duty for the PDUT register after setting the PWM function using the FMD2, FMD1, and FMD0 bits in the TMCR register.

9.1.9. Timer Register (TMR)

The Timer Register (TMR) reads the value of the 16-bit down counter.

bit	15	0
Field	TMR [15:0]	
Attribute	R	
Initial value	0x0000	

The value of the 16-bit down counter is read.

- Access the TMR register with 16-bit data.

9.2. PPG timer function

The function of the base timer can be set to either the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, or 16/32-bit PWC timer using the FMD2, 1, and 0 bits in the Timer Control Register. This section explains the timer functions available when PPG is set.

1. 16-bit PPG timer operations
2. Continuous operation
3. One-shot operation
4. Interrupt causes and timing chart
5. PPG timer operation flowchart
6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PPG timer is selected
7. LOW Width Reload Register (PRLL)
8. HIGH Width Reload Register (PRLH)
9. Timer Register (TMR)

9.2.1.16-bit PPG timer operations

In PPG timer operations, any output pulse can be controlled by setting the LOW and HIGH widths of the pulse in respective reload registers.

■ Overview of operations

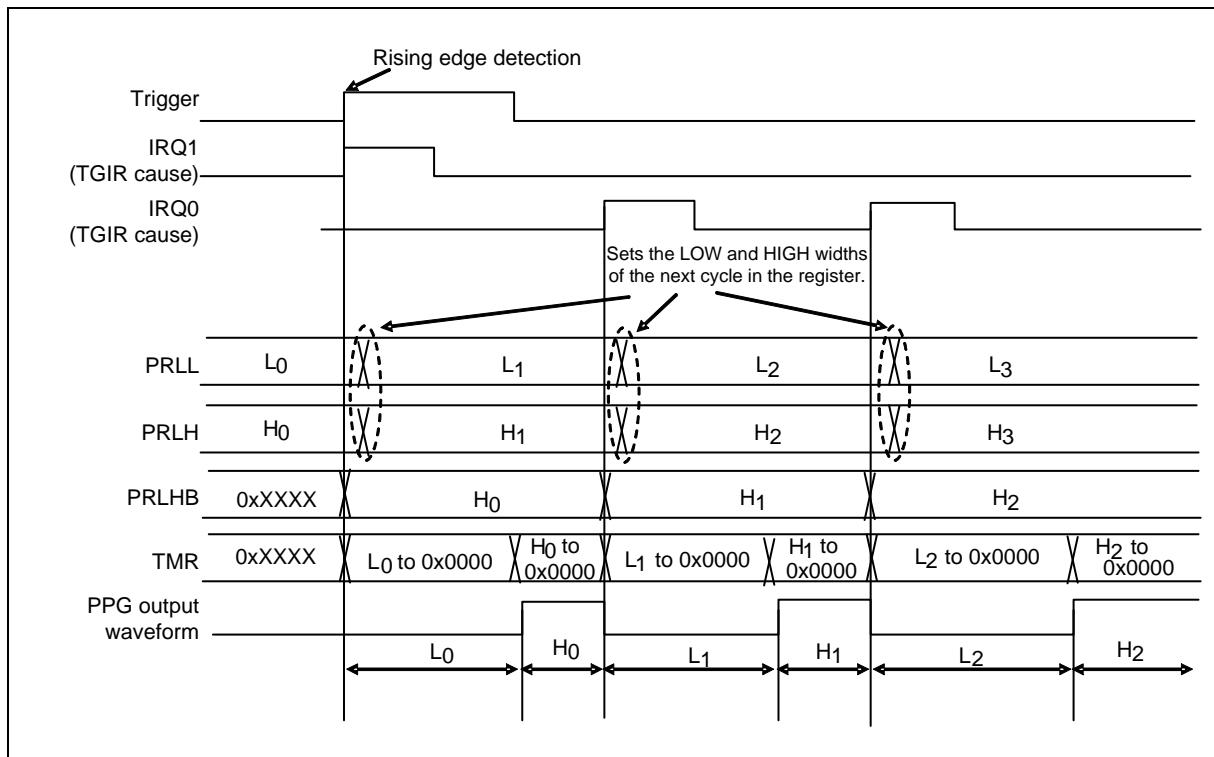
Two 16-bit long reload registers for setting the LOW and HIGH widths, respectively, and one buffer for setting the HIGH width are used (PRLL, PRLH, and PRLHB).

A start trigger initially causes the PRLL set value to be loaded to the 16-bit down counter and, at the same time, the PRLH set value to be transferred to the PRLHB. The PPG timer changes the output level to LOW and counts down for every count clock. Upon detection of an underflow, the PPG timer reloads the PRLHB value to the counter, inverts the PPG output waveforms, and continues to count down. At the next detection of an underflow, it inverts the PPG output waveforms, reloads the PRLL set value to the counter, and transfers the PRLH set value to the PRLHB.

This operation causes the output waveform to be pulse output having LOW and HIGH widths corresponding to the values in the respective reload registers.

■ Timing of writing to the reload registers

Data writing to the PRLL and PRLH reload registers occurs upon detection of a start trigger and during the period from when an underflow interrupt cause (UDIR) is set to when the next cycle starts. The data set here is used as the setting for the next cycle. The items of data set in the PRLL and PRLH are automatically transferred to the TMR and PRLHB, respectively, when a start trigger is detected and when an underflow occurs at the completion of HIGH width counting. The data transferred to the PRLHB is automatically reloaded to the TMR when an underflow occurs at the completion of LOW width counting.



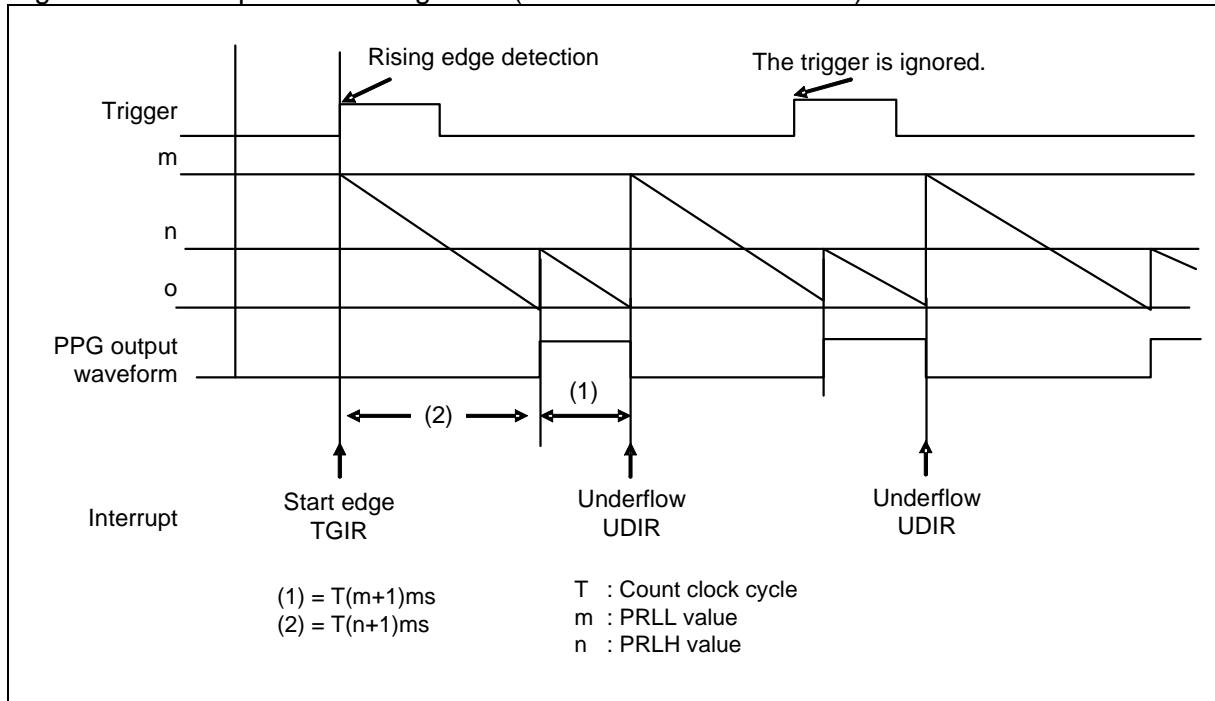
9.2.2. Continuous operation

In continuous operations, any pulse can be output continuously by updating the LOW and HIGH widths at the set timing of each interrupt cause. When a restart is enabled, the counter is reloaded when an edge is detected during operation.

■ Continuous operation

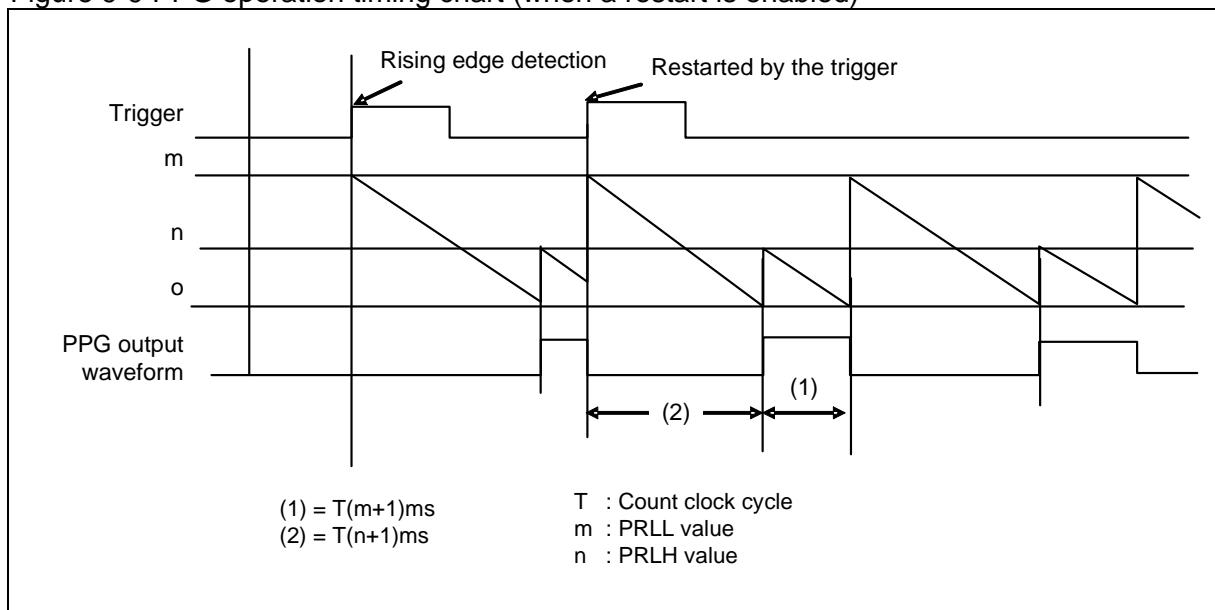
● When a restart is disabled (RTGEN = 0)

Figure 9-8 PPG operation timing chart (when a restart is disabled)



● When a restart is enabled (RTGEN = 1)

Figure 9-9 PPG operation timing chart (when a restart is enabled)



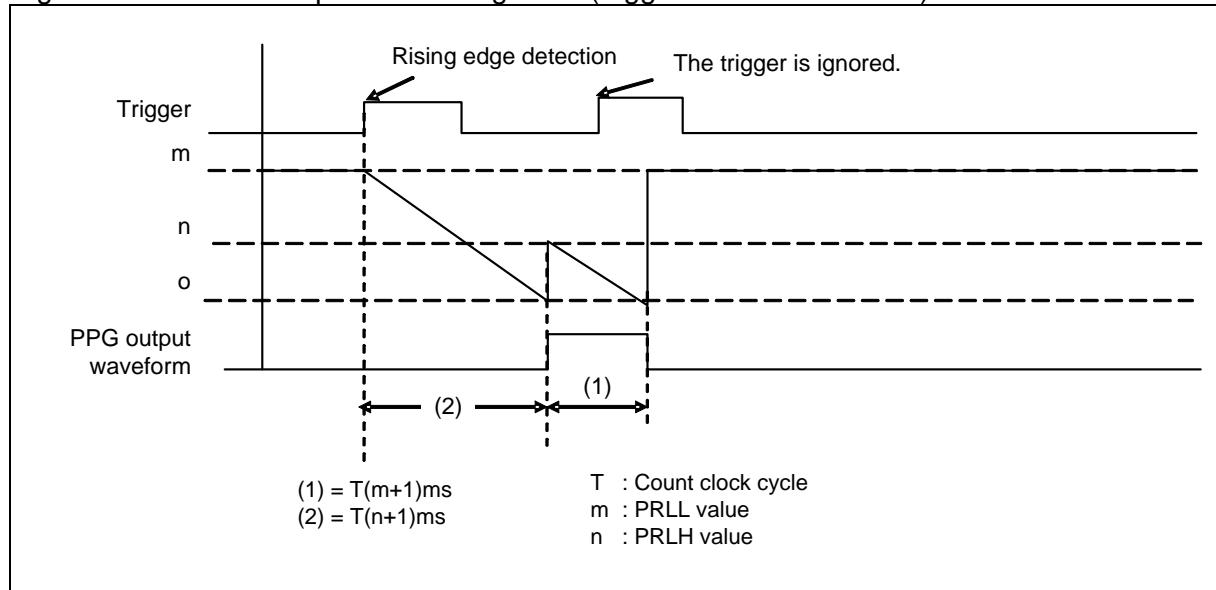
9.2.3. One-shot operation

In one-shot operation, a single pulse of any width can be output using a trigger. When a restart is enabled, the counter is reloaded when an edge is detected during operation.

■ One-shot operation

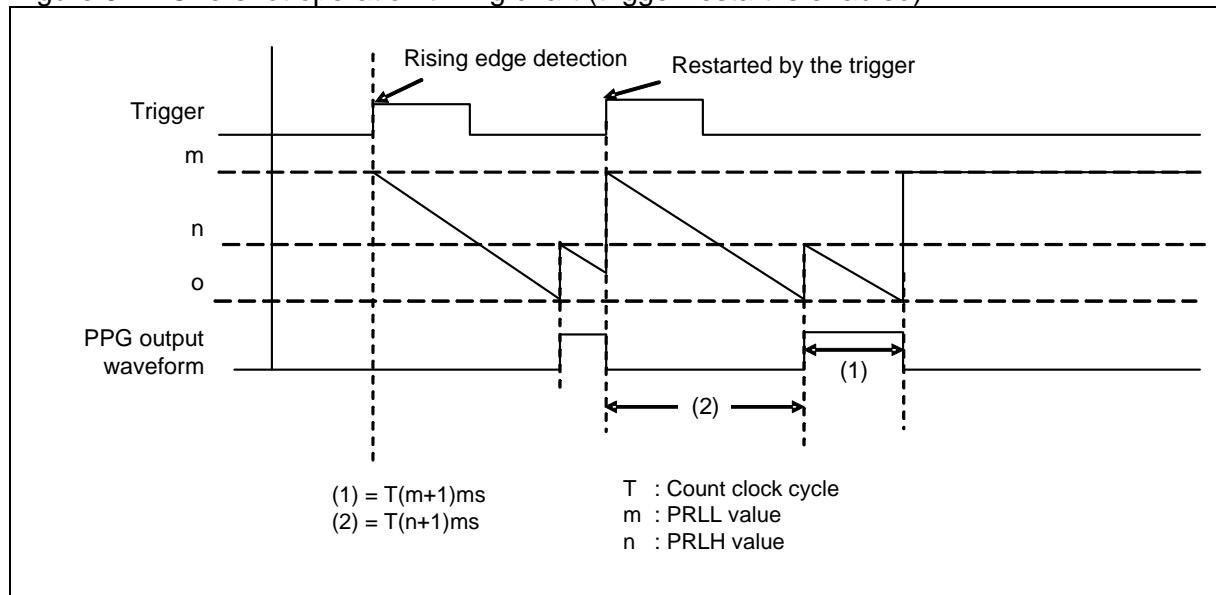
● When a restart is disabled (RTGEN = 0)

Figure 9-10 One-shot operation timing chart (trigger restart is disabled)



● When a restart is enabled (RTGEN = 1)

Figure 9-11 One-shot operation timing chart (trigger restart is enabled)



■ Relation between reload value and pulse width

The output pulse width is equal to the 16-bit long reload register value added by 1, and which is multiplied by the count clock cycle. Therefore, when the reload register value is "0x0000", the pulse width is equal to one count clock cycle. When the reload register value is "0xFFFF", the pulse width is equal to 65536 count clock cycle. The pulse width calculation formulas are as follows:

$$PL = T \times (L + 1)$$

$$PH = T \times (H + 1)$$

PL : Width of LOW pulse

PH : Width of HIGH pulse

T : Count clock cycle

L : PRLL value

H : PRLH value

9.2.4. Interrupt causes and timing chart

This section explains interrupt causes and a timing chart.

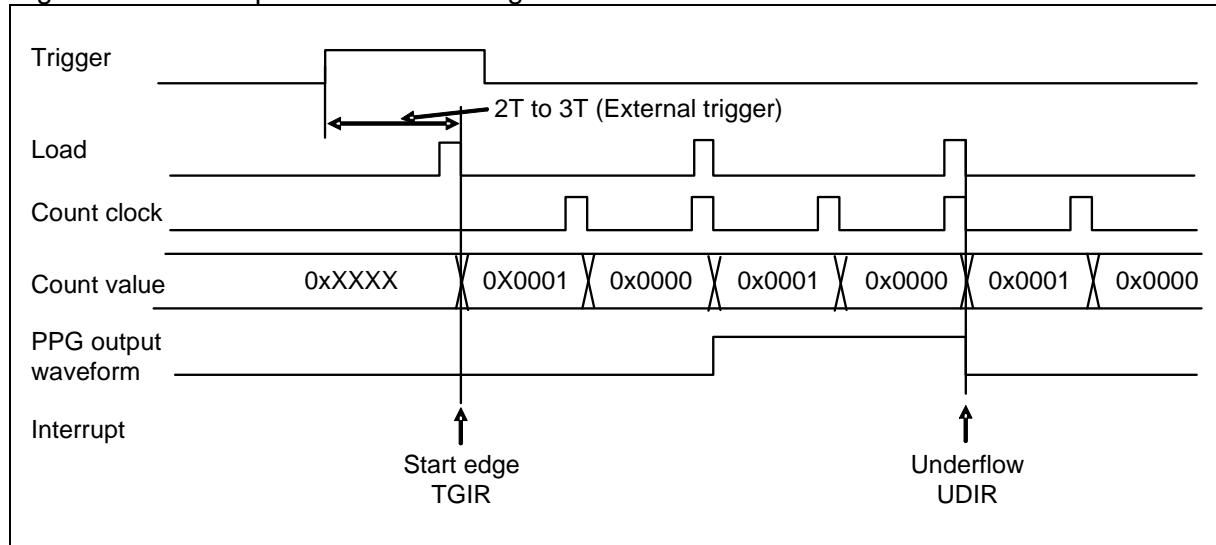
■ Interrupt causes and timing chart (PPG output: Normal polarity)

As a time from trigger input to loading of the counter value, T is required for software triggering or 2T to 3T (T: machine cycle) for external triggering.

Interrupt causes are set to detection of a PPG start trigger and an underflow in HIGH level output.

Figure 9-12 shows the interrupt cause and a timing chart where LOW width set value = 1 and HIGH width set value = 1.

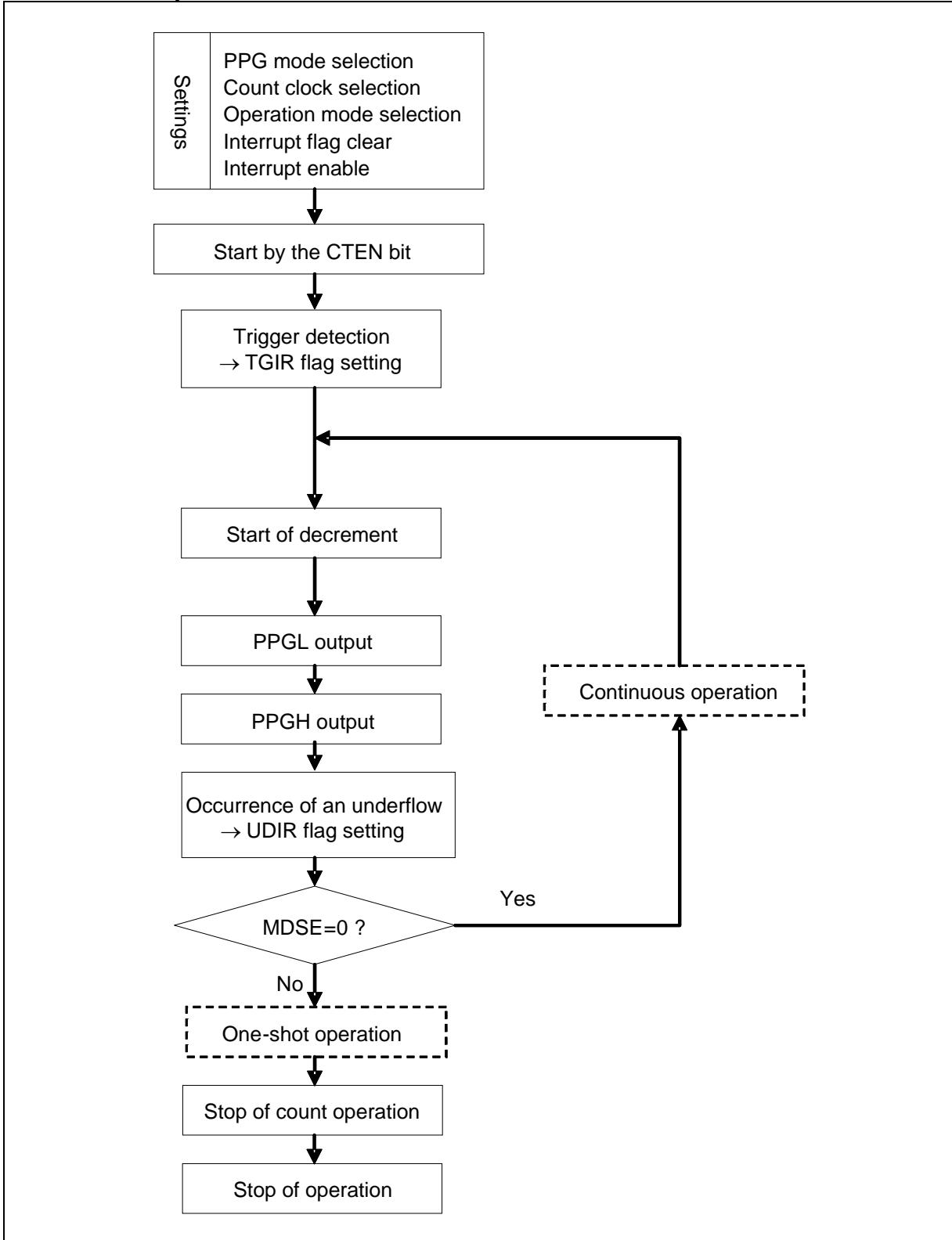
Figure 9-12 Interrupt causes and timing chart of the PPG timer



9.2.5. PPG timer operation flowchart

This section provides an operation flowchart of the PPG timer.

■ PPG timer operation flowchart



9.2.6. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PPG timer is selected

The Timer Control Register (TMCR) controls the PPG timer. Note that some bits cannot be rewritten while the PPG timer is in operation.

■ Timer Control Register (High-order bytes of TMCR)

bit	15	14	13	12	11	10	9	8
Field	res	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 15] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 14:12, TMCR2: bit 8] CKS3 to CKS0: Count clock selection bit

- Select the count clock for the 16-bit down counter.
- Changes to the count clock setting are applied immediately. For this reason, changes to CKS3 through CKS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	ϕ
0	0	0	1	$\phi /4$
0	0	1	0	$\phi /16$
0	0	1	1	$\phi /128$
0	1	0	0	$\phi /256$
0	1	0	1	External clock (rising edge event)
0	1	1	0	External clock (falling edge event)
0	1	1	1	External clock (both edge event)
1	0	0	0	$\phi /512$
1	0	0	1	$\phi /1024$
1	0	1	0	$\phi /2048$
Others				Setting disabled

[bit 11] RTGEN: Restart enable bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Restart disabled
1	Restart enabled

[bit 10] PMSK: Pulse output mask bit

- This bit controls the output level of PPG output waveforms.
- When this bit is set to "0", PPG waveforms are output as they are.
- When this bit is set to "1", the PPG output is masked with LOW output regardless of the cycle and duty set values.

<Note>

When OSEL in bit 3 is set to inverted output, setting PMSK to "1" masks the output with HIGH.

Bit	Description
0	Normal output
1	Fixed to LOW output

[bit 9:8] EGS1, EGS0: Trigger input edge selection bits

- These bits select a valid edge for input waveforms as an external start cause and set the trigger condition.
- When the initial value or "0b00" is set, the timer is not started by external waveforms because the setting means that no valid edge is selected for input waveforms.

<Note>

If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

- Changes to EGS1 or EGS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 9	Bit 8	Description
0	0	Trigger input disabled
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

■ Timer Control Register (Low-order bytes of TMCR)

bit	7	6	5	4	3	2	1	0
Field	res	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] res: Reserved bit

The read value is "0".

Set "0" to this bit.

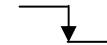
[bit 6:4] FMD2 to FMD0: Timer function selection bits

- These bits select the timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b010", the PPG function is selected.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 6	Bit 5	Bit 4	Description
0	0	0	Reset mode
0	0	1	Selection of the PWM function
0	1	0	Selection of the PPG function
0	1	1	Selection of the reload timer function
1	0	0	Selection of the PWC function
1	0	1	Setting disabled
1	1	0	
1	1	1	

[bit 3] OSEL: Output polarity specification bit

- This bit sets the polarity of the PPG output.

Polarity	After reset	Completion of LOW width counting	Completion of HIGH width counting
Normal	LOW output		
Inverted	HIGH output		

Bit	Description
0	Normal polarity
1	Inverted polarity

[bit 2] MDSE: Mode selection bit

- This bit selects continuous pulse output or one-shot pulse output.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit 1] CTEN: Count operation enable bit

- This bit enables the operation of the down counter.
- When the counter is in operation enabled status (the CTEN bit is "1"), writing "0" to this bit stops the counter.

Bit	Description
0	Stop
1	Operation enabled

[bit 0] STRG: Software trigger bit

- When the CTEN bit is "1", writing "1" to the STRG bit enables software triggering.
- The read value of the STRG bit is always "0".

<Notes>

- Software triggering is also enabled when "1" is written to the CTEN and STRG bits simultaneously.
- If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

Bit	Description
0	Invalid
1	Start triggered by software

■ Timer Control Register 2 (High-order bytes of TMCR2)

bit	15	14	13	12	11	10	9	8
Field				res				CKS3
Attribute				R/W				R/W
Initial value				0b0000000				0

Note: This register is placed above the STC register.

[bit 15:9] res: Reserved bits

The read value is "0".

Set "0" to this bit.

[bit 8] CKS3: Count clock selection bit

See "Count clock selection bit" in "9.2.6 Timer Control Register (High-order bytes of TMCR)".

■ Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	res	UDIE	res	TGIR	res	UDIR
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is placed in the upper bytes of this register.

[bit 7] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 6] TGIE: Trigger interrupt request enable bit

- This bit controls interrupt requests of bit 2 TGIR.
- When the TGIE bit is enabled, setting bit 2 TGIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 5] res : Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 4] UDIE: Underflow interrupt request enable bit

- This bit controls interrupt requests of bit 0 UDIR.
- When the UDIE bit is enabled, setting bit 0 UDIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 3] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 2] TGIR: Trigger interrupt request bit

- When a software trigger or trigger input is detected, the TGIR bit is set to "1".
- The TGIR bit is cleared by writing "0".
- Even if "1" is written to the TGIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

[bit 1] res: Reserved bit

The read value is "0".
Set "0" to this bit.

[bit 0] UDIR: Underflow interrupt request bit

- When a count value underflow from 0x0000 to 0xFFFF occurs during counting from the value for which the HIGH width is set, the UDIR bit is set to "1".
- The UDIR bit is cleared by writing "0".
- Even if "1" is written to the UDIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

9.2.7. LOW Width Reload Register (PRLL)

The LOW Width Reload Register (PRLL) is a register used to set the LOW width of PPG output waveforms. Transfer to the Timer Register is performed at detection of a start trigger or at an underflow after the completion of HIGH width counting.

bit	15	0
Field	PRLL [15:0]	
Attribute	R/W	
Initial value	0xFFFF	

This register is used to set the LOW width of PPG output waveforms. Transfer to the Timer Register is performed at detection of a start trigger and at an underflow at the completion of HIGH width counting.

- Access the PRLL register with 16-bit data.
- Set the LOW width for the PRLL register after setting the PPG function using the FMD2, FMD1, and FMD0 bits in the TMCR register.

9.2.8. HIGH Width Reload Register (PRLH)

The HIGH Width Reload Register (PRLH) is a buffered register used to set the HIGH width of PPG output waveforms. Transfer from the PRLH to the buffer register is performed at detection of a start trigger and at an underflow after the completion of HIGH width counting. Transfer from the buffer register to the Timer Register is performed at an underflow at the completion of LOW width counting.

bit	15	0
Field	PRLH [15:0]	
Attribute	R/W	
Initial value	0xFFFF	

This register is used to set the HIGH width of PPG output waveforms. Transfer from the PRLH to the buffer register is performed at detection of a start trigger and at an underflow at the completion of HIGH width counting. Transfer from the buffer register to the Timer Register is performed at an underflow at the completion of LOW width counting.

- Access the PRLH register with 16-bit data.
- Set the HIGH width for the PRLH register after setting the PPG function using the FMD2, FMD1, and FMD0 bits in the TMCR register.

9.2.9. Timer Register (TMR)

The Timer Register (TMR) reads the value of the 16-bit down counter.

bit	15	0
Field	TMR [15:0]	
Attribute	R	
Initial value	0x0000	

The value of the 16-bit down counter is read.

- Access the TMR register with 16-bit data.

9.3. Reload timer function

The function of the base timer can be set to either the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, or 16/32-bit PWC timer using the FMD2, 1, and 0 bits in the Timer Control Register. This section explains the timer functions available when the reload timer is set.

1. Operations of the 16-bit reload timer
2. Reload timer operation flowchart
3. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the reload timer is selected
4. PWM Cycle Set Register (PCSR)
5. Timer Register (TMR)

9.3.1. Operations of the 16-bit reload timer

In reload timer operations, countdown is performed from the value set in the PWM Cycle Set Register in synchronization with the count clock. This operation continues until the count value reaches 0 or the cycle setting is loaded automatically to stop the countdown.

■ Count operation performed when the internal clock is selected

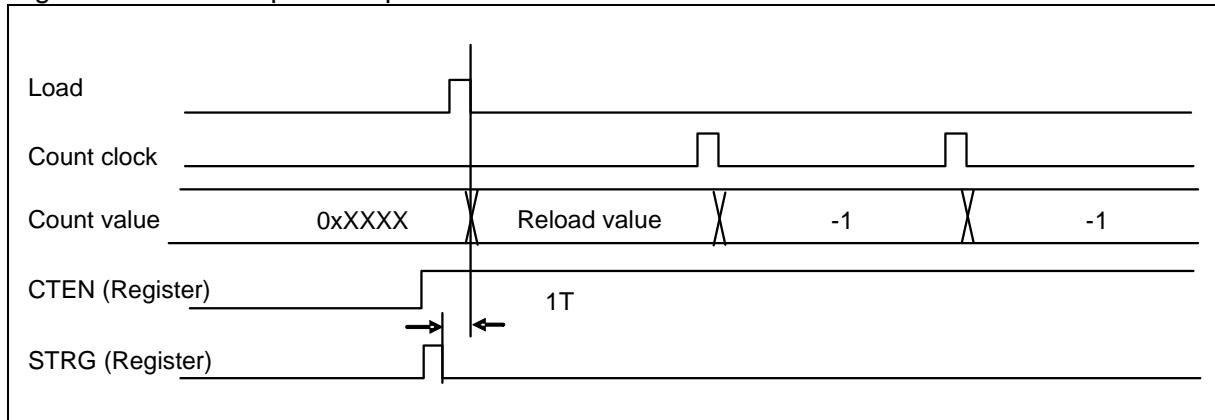
To start the count operation at the same time counting is enabled, write "1" to both CTEN and STRG bits in the Timer Control Register. When the timer is started ($CTEN = 1$), trigger input with the STRG bit is valid regardless of the operation mode.

When the count operation is enabled and the timer is started with a software trigger or an external trigger, the value in the PWM Cycle Set Register is loaded to the counter and countdown is started.

It takes a time of $1T$ (T : machine cycle) from setting of a counter start trigger to loading of the PWM Cycle Set Register data to the counter.

Figure 9-13 shows the start of the counter by a software trigger and counter operation.

Figure 9-13 Count operation performed when the internal clock is selected



■ Underflow operation

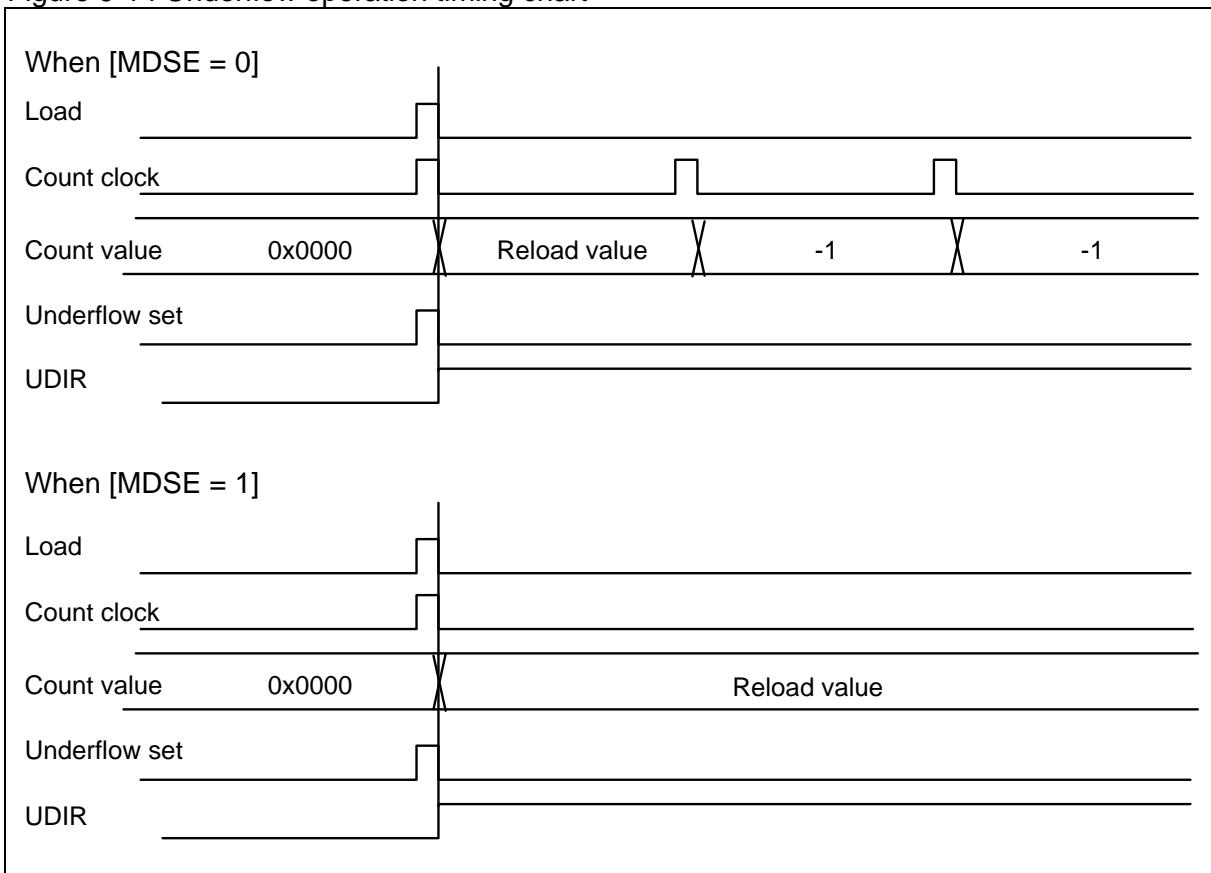
An underflow occurs when the counter value changes from 0x0000 to 0xFFFF. Therefore, an underflow occurs at a count of [Set value in the PWM Cycle Set Register + 1].

When an underflow occurs, the contents of the PWM Cycle Set Register (PCSR) are loaded to the counter. When the MDSE bit in the Timer Control Register (TMCR) is "0", the count operation continues. When the MDSE bit is "1", the counter stops while keeping the loaded counter value.

An underflow sets the UDIR bit in the Status Control Register (STC). In this case, an interrupt request occurs when the UDIE bit is "1".

Figure 9-14 shows a timing chart of underflow operations.

Figure 9-14 Underflow operation timing chart

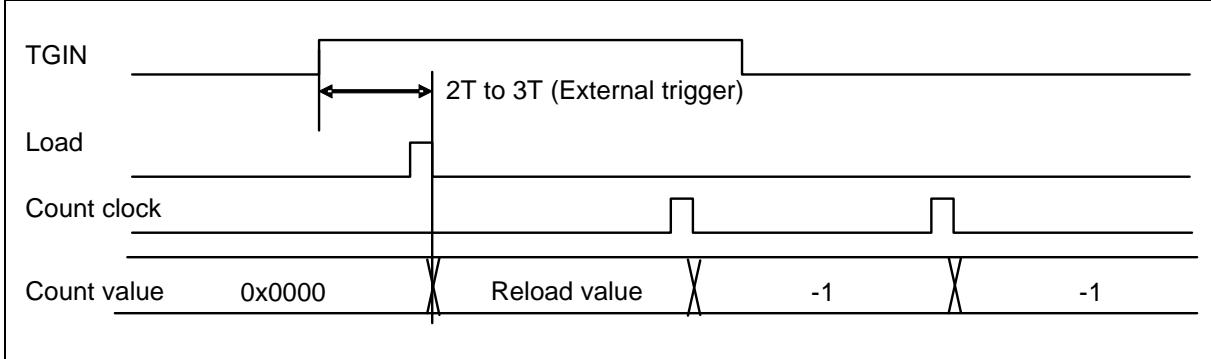


■ Operation of the input pin function

The TGIN pin can be used for trigger input. When a valid edge is input to the TGIN pin, the contents of the PWM Cycle Set Register are loaded to the counter and the count operation is started. As a time from trigger input to loading of the counter value, 2T to 3T (T: machine cycle) is required.

Figure 9-15 shows a trigger input operation performed when a rising edge is specified as a valid edge.

Figure 9-15 Operation caused by a trigger input

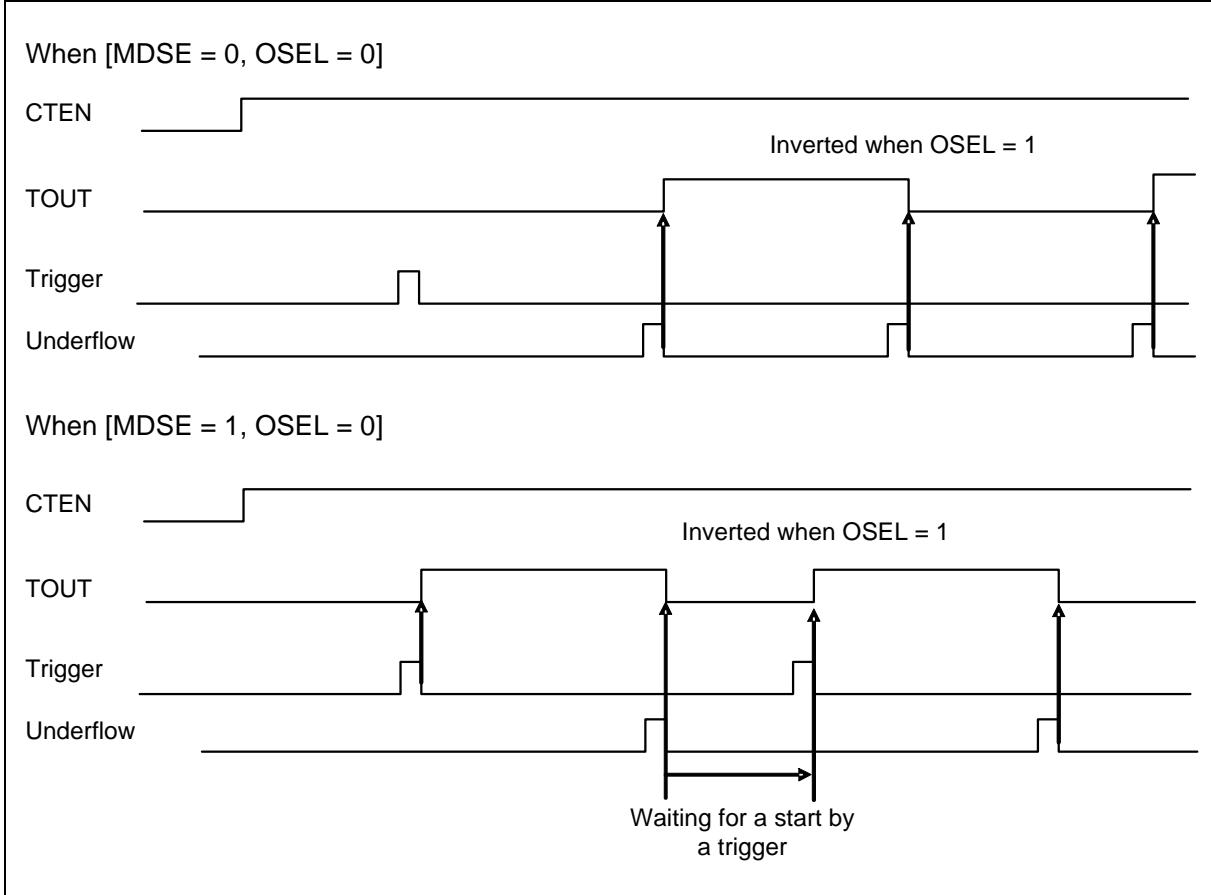


■ Operation of the output pin function

The TOUT output pin functions as, reload mode, toggle output inverted by an underflow and, in one-shot mode, pulse output indicating that counting is in progress. The output polarity can be set with the OSEL bit in the Timer Control Register (TMCR). If OSEL = 0, toggle output has an initial value of "0", and one-shot pulse output is "1" during counting. When OSEL is set to 1, the output waveform is inverted.

Figure 9-16 shows a timing chart of output pin function operations.

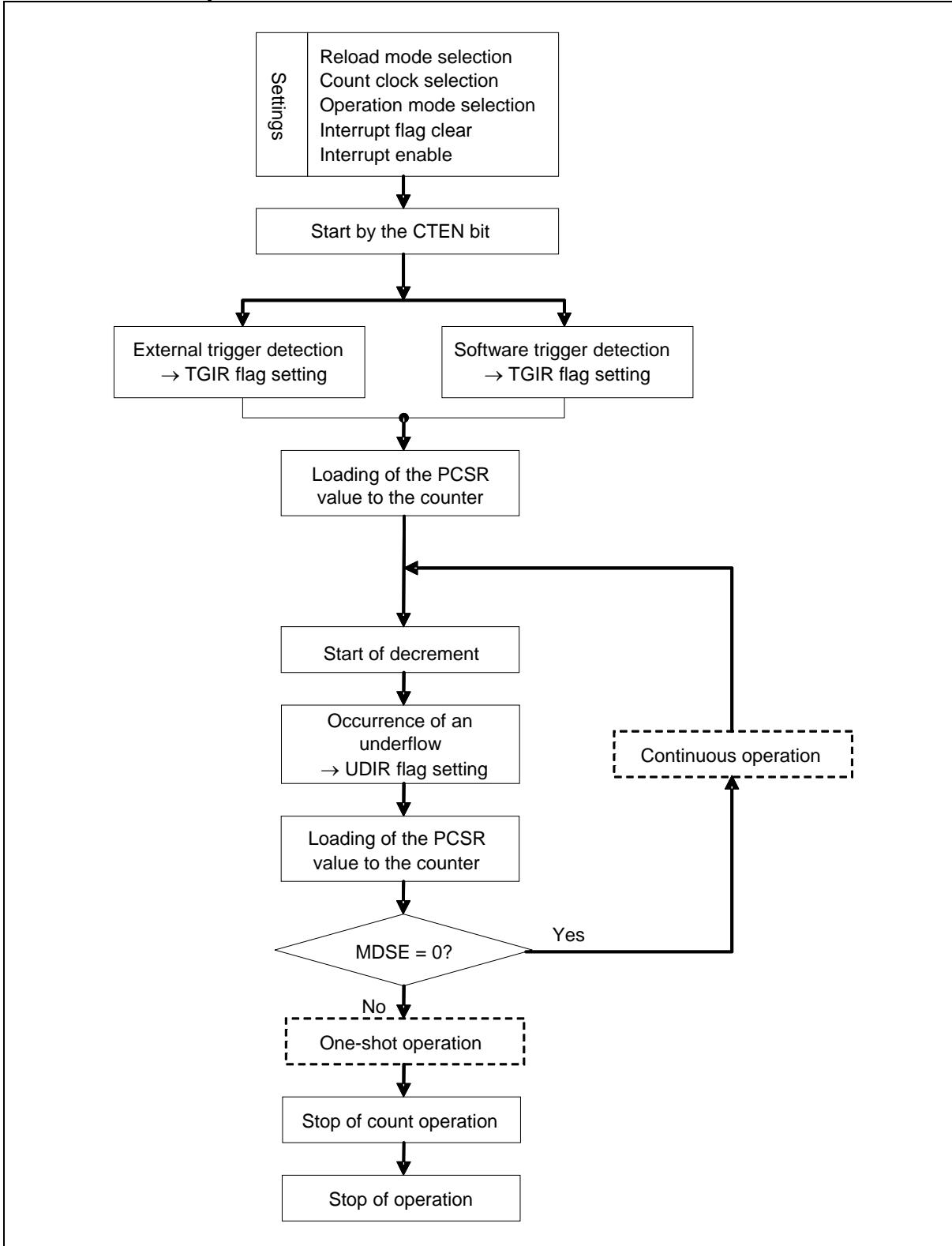
Figure 9-16 Output pin function operation timing chart



9.3.2. Reload timer operation flowchart

This section provides an operation flowchart of the reload timer.

■ Reload timer operation flowchart



9.3.3. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the reload timer is selected

The Timer Control Register (TMCR) controls timer operations.

■ Timer Control Register (High-order bytes of TMCR)

bit	15	14	13	12	11	10	9	8
Field	res	CKS2	CKS1	CKS0	res	EGS1	EGS0	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0b00	0	0	0

[bit 15] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 14:12, TMCR2: bit 8] CKS3 to CKS0: Count clock selection bit

- Select the count clock for the 16-bit down counter.
- Changes to the count clock setting are applied immediately. For this reason, changes to CKS3 through CKS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	ϕ
0	0	0	1	$\phi/4$
0	0	1	0	$\phi/16$
0	0	1	1	$\phi/128$
0	1	0	0	$\phi/256$
0	1	0	1	External clock (rising edge event)
0	1	1	0	External clock (falling edge event)
0	1	1	1	External clock (both edge event)
1	0	0	0	$\phi/512$
1	0	0	1	$\phi/1024$
1	0	1	0	$\phi/2048$
Others				Setting disabled

[bit 11:10] res : Reserved bits

The read value is "0".

Set "0" to this bit.

[bit 9:8] EGS1, EGS0: Trigger input edge selection bits

- These bits select a valid edge for input waveforms as an external start cause and set the trigger condition.
- When the initial value or "0b00" is set, the timer is not started by external waveforms because the setting means that no valid edge is selected for input waveforms.

<Note>

If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

- Changes to EGS1 or EGS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 9	Bit 8	Description
0	0	Trigger input disabled
0	1	External trigger (rising edge)
1	0	External trigger (falling edge)
1	1	External trigger (both edges)

■ Timer Control Register 2 (Low-order bytes of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] T32: 32-bit timer selection bit

- This bit selects the 32-bit timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b011" to select the reload timer function, setting the T32 bit to "1" selects 32-bit timer mode.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit (see 32-bit mode operations).

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit 6:4] FMD2 to FMD0: Timer function selection bits

- These bits select the timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b011", the reload timer function is selected.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 6	Bit 5	Bit 4	Description
0	0	0	Reset mode
0	0	1	Selection of the PWM function
0	1	0	Selection of the PPG function
0	1	1	Selection of the reload timer function
1	0	0	Selection of the PWC function
Others		Setting disabled	

[bit 3] OSEL: Output polarity specification bit

- This bit selects whether to invert the timer output level.
- Used in combination with bit 2 MDSE, this bit generates the following output waveforms.

MDSE	OSEL	Output waveforms
0	0	Toggle output at the LOW level at the start of counting
0	1	Toggle output at the HIGH level at the start of counting
1	0	Rectangular waves at the HIGH level during counting
1	1	Rectangular waves at the LOW level during counting

Bit	Description
0	Normal polarity
1	Inverted polarity

[bit 2] MDSE: Mode selection bit

- When the MDSE bit is set to "0", reload mode is selected. When a value underflow from 0x0000 to 0xFFFF occurs, the reload register value is loaded to the counter at the same time, and the count operation is continued.
- When the MDSE bit is set to "1", one-shot mode is selected. A count value underflow from 0x0000 to 0xFFFF stops the operation.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit	Description
0	Reload mode
1	One-shot mode

[bit 1] CTEN: Timer enable bit

- This bit enables the operation of the down counter.
- When the counter is in operation enabled status (the CTEN bit is "1"), writing "0" to this bit stops the counter.

Bit	Description
0	Stop
1	Operation enabled

[bit 0] STRG: Software trigger bit

- When the CTEN bit is "1", writing "1" to the STRG bit enables software triggering.
- The read value of the STRG bit is always "0".

<Notes>

- Software triggering is also enabled when "1" is written to the CTEN and STRG bits simultaneously.
- If the STRG bit is set to "1", software triggering is enabled regardless of the EGS1 and EGS0 settings.

Bit	Description
0	Invalid
1	Start triggered by software

■ Timer Control Register 2 (High-order bytes of TMCR2)

bit	15	14	13	12	11	10	9	8
Field				res				CKS3
Attribute				R/W				R/W
Initial value				0b0000000				0

Note: This register is placed above the STC register.

[bit 15:9] res: Reserved bits

The read value is "0".

Set "0" to this bit.

[bit 8] CKS3: Count clock selection bit

See "Count clock selection bit" in "9.3.3 Timer Control Register (High-order bytes of TMCR)".

■ Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	res	UDIE	res	TGIR	res	UDIR
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is placed in the upper bytes of this register.

[bit 7] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 6] TGIE: Trigger interrupt request enable bit

- This bit controls interrupt requests of bit 2 TGIR.
- When the TGIE bit is enabled, setting bit 2 TGIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 5] res : Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 4] UDIE: Underflow interrupt request enable bit

- This bit controls interrupt requests of bit 0 UDIR.
- When the UDIE bit is enabled, setting bit0 UDIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 3] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 2] TGIR: Trigger interrupt request bit

- When a software trigger or trigger input is detected, the TGIR bit is set to "1".
- The TGIR bit is cleared by writing "0".
- Even if "1" is written to the TGIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

[bit 1] res: Reserved bit

The read value is "0".
Set "0" to this bit.

[bit 0] UDIR: Underflow interrupt request bit

- When a count value underflow from 0x0000 to 0xFFFF occurs during counting from the value for which the HIGH width is set, the UDIR bit is set to "1".
- The UDIR bit is cleared by writing "0".
- Even if "1" is written to the UDIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

9.3.4. PWM Cycle Set Register (PCSR)

The PWM Cycle Set Register (PCSR) is a register for storing the initial counter value. In 32-bit mode and for the even channel, the initial count value of the lower 16 bits is stored. For the odd channel, the initial count value of the upper 16 bits is stored. The initial value after a reset is undefined. Be sure to use the 16-bit data transfer instruction to access this register.

bit	15	0
Field	PCSR [15:0]	
Attribute	R/W	
Initial value	0xXXXX	

This is a register for setting the cycle. Transfer to the Timer Register is performed at an underflow.

- Access the PCSR register with 16-bit data.
- Set the cycle for the PCSR register after setting the reload timer function using the FMD2, FMD1, and FMD0 bits in the TMCR register.
- When writing data in the PCSR register in 32-bit mode, access the upper 16-bit data (odd channel data) first, and then access the lower 16-bit data (even channel data).

9.3.5. Timer Register (TMR)

The Timer Register (TMR) is a register that reads the count value of a timer. In 32-bit mode and for the even channel, the count value of the lower 16 bits is read. For the odd channel, the count value of the upper 16 bits is read. The initial value is undefined.

Be sure to use the 16-bit data transfer instruction to read this register.

bit	15	0
Field	TMR [15:0]	
Attribute	R	
Initial value	0xFFFF	

The value of the 16-bit down counter is read.

- Access the TMR register with 16-bit data.
- When reading the TMR register in 32-bit mode, read the lower 16-bit data (even channel data) first, and then read the upper 16-bit data (odd channel data).

9.4. PWC timer function

The function of the base timer can be set to either the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, or 16/32-bit PWC timer using the FMD2, 1, and 0 bits in the Timer Control Register. This section explains the timer functions available when PWC is set.

1. Operations of the PWC timer
2. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWC timer is selected
3. Data Buffer Register (DTBF)

9.4.1. Operations of the PWC timer

The PWC timer has the pulse width measurement function. Five types of count clock are available for measuring the time and cycle between any input pulse events. This section explains the basic functions and operations of the pulse width measurement function.

■ Pulse width measurement function

Count operation is not performed until the counter is started and cleared to "0x0000" and the specified measurement start edge is input. Upon detecting a measurement start edge, the counter starts count-up from "0x0001" and stops counting upon detecting a measurement end edge. The value counted in between is stored as a pulse width in the register.

An interrupt request can be generated when the measurement is completed or an overflow occurs.

After the completion of measurement, it operates as follows depending on the measurement mode:

- In one-shot measurement mode: Stops the operation.
- In continuous measurement mode: Transfers the counter value to the buffer register and stops counting until the measurement start edge is input again.

Figure 9-17 Pulse width measurement operation (one-shot measurement mode/HIGH width measurement)

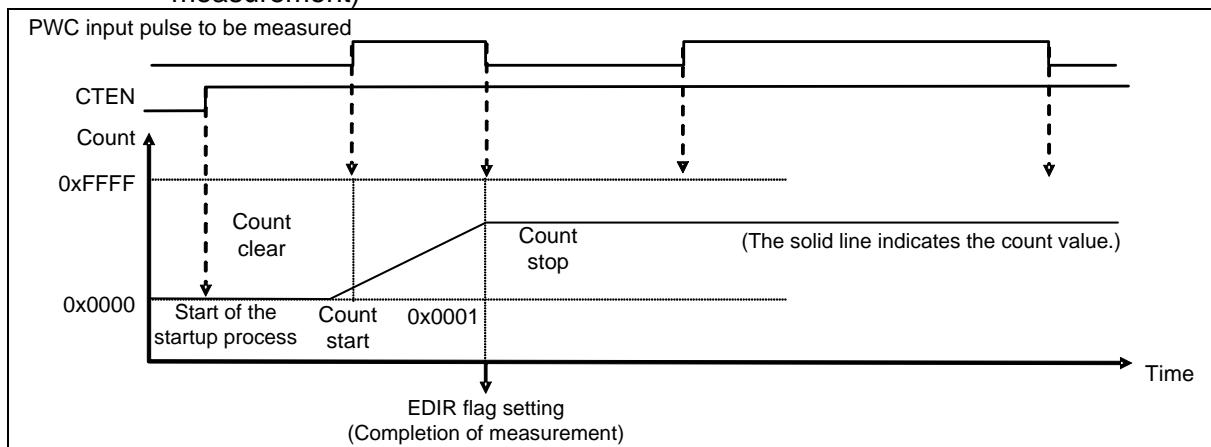
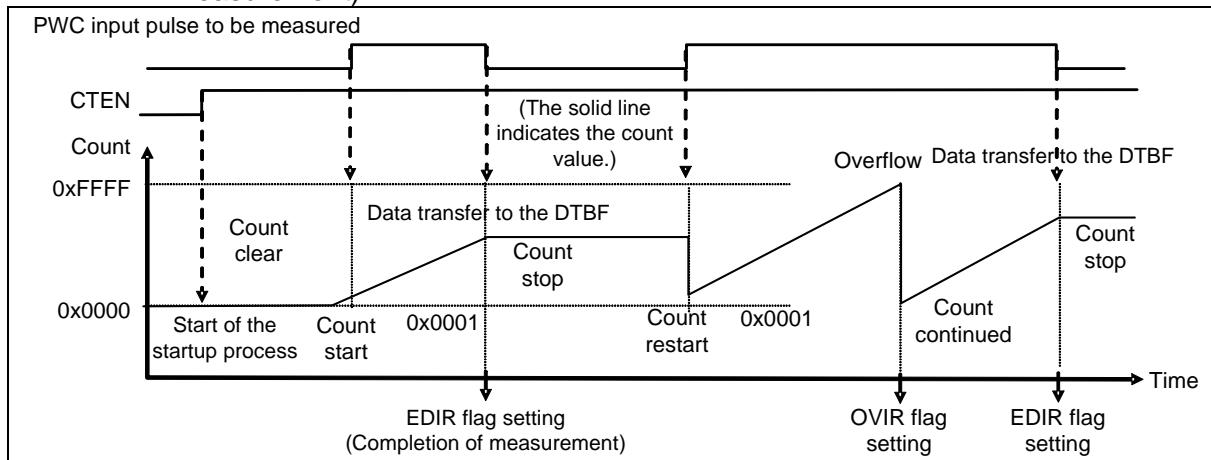


Figure 9-18 Pulse width measurement operation (continuous measurement mode/HIGH width measurement)



■ Selection of count clock

The count clock of the counter can be selected from eight types by setting bit 8: CKS3 in the TMCR2 register and bit 14 to 12: CKS2, CKS1, and CKS0 in the TMCR register.

The selectable count clocks are as follows:

TMCR2 and TMCR registers		Internal count clock to be selected
CKS3, CKS2, CKS1, and CKS0 bits		
0b0000		Machine clock [Initial value]
0b0001		1/4 frequency of the machine clock
0b0010		1/16 frequency of the machine clock
0b0011		1/128 frequency of the machine clock
0b0100		1/256 frequency of the machine clock
0b0101		Setting disabled
0b0110		
0b0111		
0b1000		1/512 frequency of the machine clock
0b1001		1/1024 frequency of the machine clock
0b1010		1/2048 frequency of the machine clock
Others		Setting disabled

The machine clock is selected as the initial value after a reset.

Be sure to select the counter clock before starting the counter.

■ Selection of operation mode

Set the TMCR to select the operation/measurement mode.

Operation mode setting ... TMCR bit 10 to 8: EGS2, EGS1, and EGS0 (Selection of measurement edge)
Measurement mode setting ... TMCR bit 2: MDSE (Selection of one-shot/continuous measurement)

The following provides a list of operation mode settings.

Operation mode		MDSE	EGS2	EGS1	EGS0
↑ to ↓ HIGH pulse width measurement	Continuous measurement: Buffer enabled	0	0	0	0
	One-shot measurement: Buffer disabled	1	0	0	0
↑ to ↑ Cycle measurement between rising edges	Continuous measurement: Buffer enabled	0	0	0	1
	One-shot measurement: Buffer disabled	1	0	0	1
↓ to ↓ Cycle measurement between falling edges	Continuous measurement: Buffer enabled	0	0	1	0
	One-shot measurement: Buffer disabled	1	0	1	0
↑ or ↓ to ↑ or ↓ Interval measurement between all edges	Continuous measurement: Buffer enabled	0	1	1	1
	One-shot measurement: Buffer disabled	1	1	1	1
↓ to ↑ LOW pulse width measurement	Continuous measurement: Buffer enabled	0	1	0	0
	One-shot measurement: Buffer disabled	1	1	0	0
Setting disabled		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

HIGH pulse width measurement in one-shot measurement mode is selected as the initial value after a reset.
Be sure to select an operation mode before starting the counter.

■ Starting and stopping pulse width measurement

Set bit 1: CTEN bit in the TMCR to start, restart, or stop forcibly each operation.

The pulse width measurement is started or restarted by writing "1" to the CTEN bit, and it is stopped forcibly by writing "0" to the CTEN bit.

CTEN	Function
1	Starts or restarts the pulse width measurement.
0	Forcibly stops the pulse width measurement.

■ Operation after a restart

After the counter is restarted in pulse measurement mode, counting is not performed until a measurement start edge is input. After a measurement start edge is detected, the 16-bit up counter starts counting from "0x0001".

■ Restart

An operation to start the counter again after it has been started and while it is in operation (writing "1" again while the CTEN bit is "1") is referred to as a restart. When restarted, the counter performs the following operation:

- When waiting for a measurement start edge:
Has no effect on operation.
- When performing measurement:
Clears the count to "0x0000" and waits for a measurement start edge again. When detection of a measurement end edge and the restart operation occur simultaneously, a measurement end flag (EDIR) is set and, when in continuous measurement mode, the measurement result is transferred to the DTBF.

■ Stop

In one-shot measurement mode, since the count operation is stopped automatically by a counter overflow or completion of measurement, you do not have to be aware of the stop. In continuous measurement mode or when you want to stop the operation before it stops automatically, you have to stop it forcibly.

■ Counter clear and initial value

The 16-bit up counter is cleared to "0x0000" in the following cases:

- When a reset is performed
- When "1" is written to bit 1: CTEN bit in the TMCR (including the cases for restarting)

The 16-bit up counter is initialized to "0x0001" in the following case:

- When a measurement start edge is detected

■ Details on pulse width measurement operations

● One-shot measurement and continuous measurement

Pulse width measurement can be performed in two modes: one for performing measurement only one time and the other for performing it continuously. Each mode is selected with the MDSE bit in the TMCR (see "Selection of operation mode"). Differences between these modes are as follows:

One-shot measurement mode:

When the first measurement end edge is input, the counter stops counting, a measurement end flag (EDIR) in the STC is set, and no further measurement is performed.

However, when restarted at the same time, it waits to start measurement.

Continuous measurement mode:

When a measurement end edge is input, the counter stops counting, a measurement end flag (EDIR) in the STC is set, and the counter stops until the measurement start edge is input again. When the measurement start edge is input again, the counter is initialized to "0x0001" and measurement is started. After the measurement is completed, the result in the counter is transferred to the DTBF.

Be sure to select/change measurement modes while the counter is stopped.

● Measurement result data

There are differences between one-shot and continuous modes in handling of measurement results and counter values and in DTBF functions. Differences between these modes in handling of measurement results are as follows:

One-shot measurement mode:

Reading the DTBF during operation obtains the count value being measured.

Reading the DTBF after the completion of measurement obtains the measurement result data.

Continuous measurement mode:

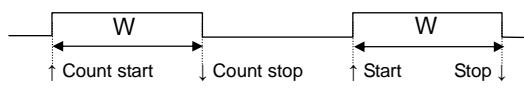
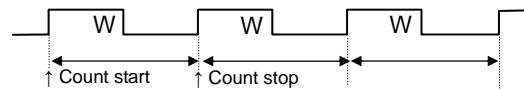
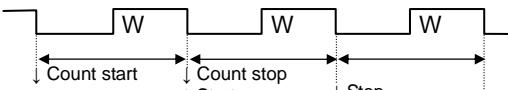
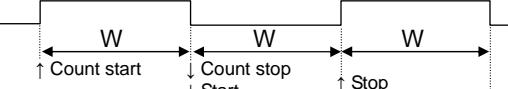
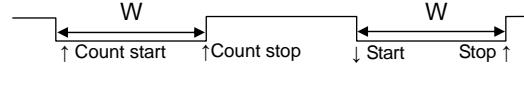
After the measurement is completed, the result in the counter is transferred to the DTBF.

Reading the DTBF obtains the last measurement result. The previous measurement result is retained during the measurement operation. It is not possible to read the count value being measured.

In continuous measurement mode, if the next measurement is completed before the measurement result is read, the previous result is overwritten by the new result. In this case, an error flag (ERR) in the STC is set. The error flag is cleared automatically when the DTBF is read.

● Measurement modes and count operations

The measurement mode can be selected from five types, differing in which part of the input pulse is measured. The following are explanations:

Measurement mode	EGS2, 1, and 0	Item to be measured (W: Pulse width to be measured)
HIGH pulse width measurement	0b000	 <p>The width of the HIGH period is measured. Count (measurement) start: At detection of a rising edge Count (measurement) end: At detection of a falling edge</p>
Cycle measurement between rising edges	0b001	 <p>The cycle between rising edges is measured. Count (measurement) start: At detection of a rising edge Count (measurement) end: At detection of a rising edge</p>
Cycle measurement between falling edges	0b010	 <p>The cycle between falling edges is measured. Count (measurement) start: At detection of a falling edge Count (measurement) end: At detection of a falling edge</p>
Interval measurement between all edges	0b011	 <p>The width between continuously input edges is measured. Count (measurement) start: At detection of an edge Count (measurement) end: At detection of an edge</p>
LOW pulse width measurement	0b100	 <p>The width of the LOW period is measured. Count (measurement) start: At detection of a falling edge Count (measurement) end: At detection of a rising edge</p>

In any measurement mode, the counter is cleared to "0x0000" when started, and it does not perform the count operation until a measurement start edge is input. Once a measurement start edge is input, the counter continues incrementing for every count clock until a measurement end edge is input.

In pulse width measurement between all edges or cycle measurement in continuous measurement mode, an end edge is also a start edge for the next measurement.

● Pulse width/cycle calculation method

After completion of measurement, the measured pulse width/cycle can be calculated as follows from the measurement result data stored in the DTBF:

$$T_W = n \times t$$

T_W : Measured pulse width/cycle

n : Measurement result data stored in the DTBF

t : Count clock cycle

● Generation of interrupt requests

Two interrupt requests can be generated.

- Interrupt request due to a counter overflow

When count-up causes an overflow during measurement, an overflow flag (OVIR) is set and an interrupt request is generated if overflow interrupt requests are enabled.

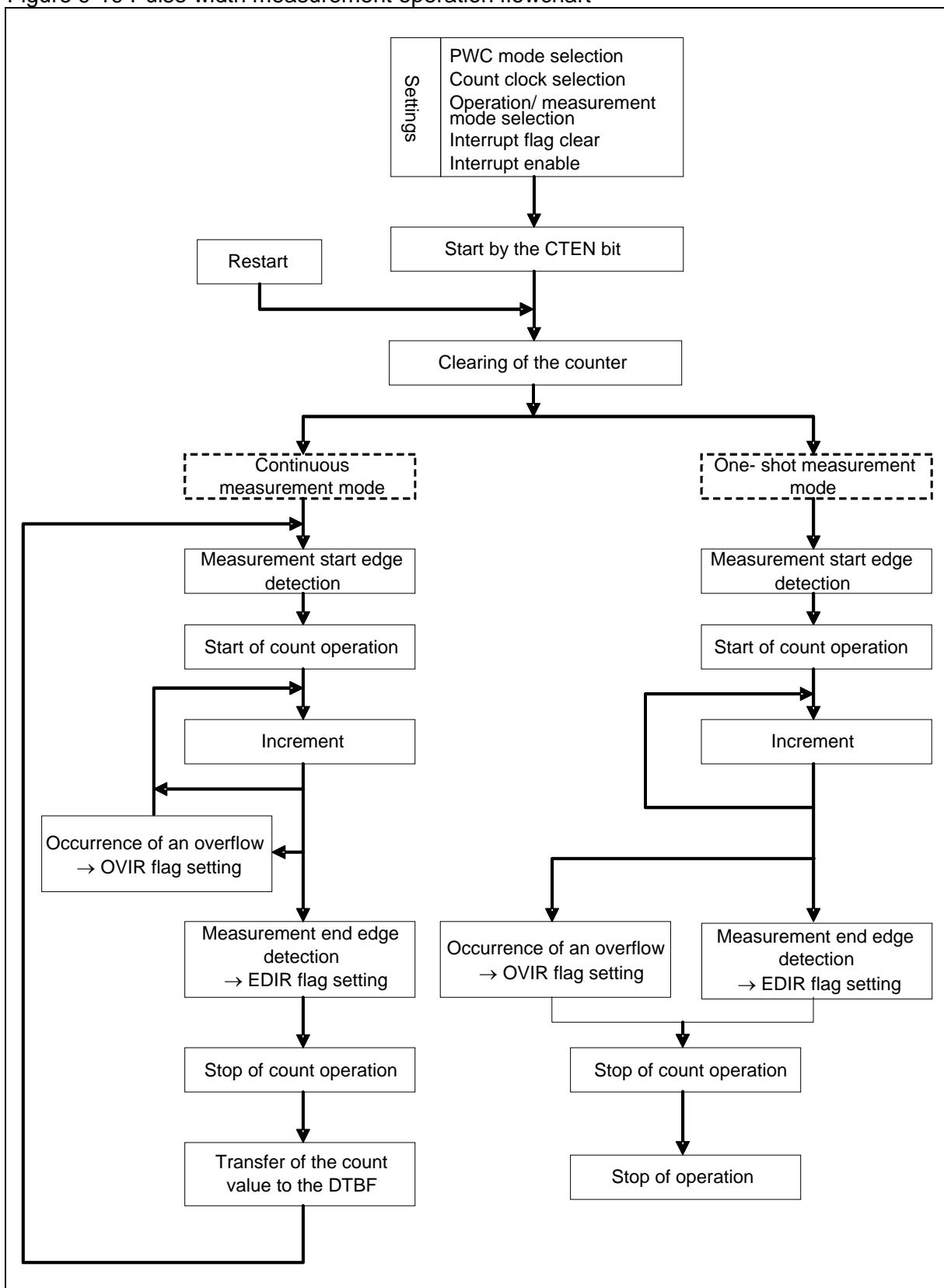
- Interrupt request due to completion of measurement

When a measurement end edge is detected, a measurement end flag (EDIR) in the STC is set and an interrupt request is generated if measurement end interrupt requests are enabled.

The measurement end flag (EDIR) is cleared automatically when the measurement result DTBF is read.

● Pulse width measurement operation flowchart

Figure 9-19 Pulse width measurement operation flowchart



9.4.2. Timer Control Registers (TMCR and TMCR2) and Status Control Register (STC) used when the PWC timer is selected

The Timer Control Register (TMCR) controls timer operations.

■ Timer Control Register (High-order bytes of TMCR)

bit	15	14	13	12	11	10	9	8
Field	res	CKS2	CKS1	CKS0	res	EGS2	EGS1	EGS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 15] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 14:12, TMCR2: bit 8] CKS3 to CKS0: Count clock selection bit

- Select the count clock for the 16-bit down counter.
- Changes to the count clock setting are applied immediately. For this reason, changes to CKS3 through CKS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	ϕ
0	0	0	1	$\phi/4$
0	0	1	0	$\phi/16$
0	0	1	1	$\phi/128$
0	1	0	0	$\phi/256$
0	1	0	1	Setting disabled
0	1	1	0	
0	1	1	1	
1	0	0	0	$\phi/512$
1	0	0	1	$\phi/1024$
1	0	1	0	$\phi/2048$
Others				Setting disabled

[bit 11] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 10:8] EGS2 to EGS0: Measurement edge selection bits

- These bit set measurement edge conditions.
- Changes to EGS2, EGS1, or EGS0 must be made when the counting is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 10	Bit 9	Bit 8	Description
0	0	0	HIGH pulse width measurement (\uparrow to \downarrow)
0	0	1	Cycle measurement between rising edges (\uparrow to \uparrow)
0	1	0	Cycle measurement between falling edges (\downarrow to \downarrow)
0	1	1	Pulse width measurement between all edges (\uparrow or \downarrow to \downarrow or \uparrow)
1	0	0	LOW pulse width measurement (\downarrow to \uparrow)
1	0	1	Setting disabled
1	1	0	
1	1	1	

■ Timer Control Register (Low-order bytes of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	res	MDSE	CTEN	res
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] T32: 32-bit timer selection bit

- This bit selects the 32-bit timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b100" to select the PWC function, setting the T32 bit to "1" selects 32-bit PWC mode.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit (see 32-bit mode operations).

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit 6:4] FMD2 to FMD0: Timer function selection bits

- These bits select the timer function.
- When the FMD2, FMD1, and FMD0 bits are set to "0b100", the PWC timer function is selected.
- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit 6	Bit 5	Bit 4	Description
0	0	0	Reset mode
0	0	1	Selection of the PWM function
0	1	0	Selection of the PPG function
0	1	1	Selection of the reload timer function
1	0	0	Selection of the PWC function
1	0	1	Setting disabled
1	1	0	
1	1	1	

[bit 3] res: Reserved bit

The read value is "0".
Set "0" to this bit.

[bit 2] MDSE: Mode selection bit

- Changes must be made while the timer is stopped (CTEN = "0"). However, it is possible to make changes at the same time you set "1" to the CTEN bit.

Bit	Description
0	Continuous measurement mode (Buffer register enabled)
1	One-shot measurement mode (Stops after one measurement)

[bit 1] CTEN: Timer enable bit

- This bit enables the start or restart of the up counter.
- When the counter is in operation enabled status (the CTEN bit is "1"), writing "1" restarts the counter. The counter is cleared and waits for a measurement start edge.
- When the counter is in operation enabled status (the CTEN bit is "1"), writing "0" to this bit stops the counter.

Bit	Description
0	Stop
1	Operation enabled

[bit 0] res: Reserved bit

The read value is "0".

Set "0" to this bit.

■ Timer Control Register 2 (High-order bytes of TMCR2)

bit	15	14	13	12	11	10	9	8
Field				res				CKS3
Attribute				R/W				R/W
Initial value				0b0000000				0

Note: This register is placed above the STC register.

[bit 15:9] res: Reserved bits

The read value is "0".

Set "0" to this bit.

[bit 8] CKS3: Count clock selection bit

See "Count clock selection bit" in "9.4.2 Timer Control Register (High-order bytes of TMCR)".

■ Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	ERR	EDIE	res	OVIE	res	EDIR	res	OVIR
Attribute	R	R/W	R/W	R/W	R/W	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Note: The TMCR2 register is placed in the upper bytes of this register.

[bit 7] ERR: Error flag bit

- This flag indicates that the next measurement has been completed in continuous measurement mode before the measurement result is read from the DTBF register. In this case, the result of the previous measurement in the DTBF register is replaced by that of the next measurement.
- The measurement is continued regardless of the ERR bit value.
- The ERR is read-only. Writing a value does not affect the bit value.
- The ERR bit is cleared by reading the measurement result (DTBF).

Bit	Description
0	Normal status
1	A measurement result not yet read was overwritten by the next measurement result.

[bit 6] EDIE: Measurement completion interrupt request enable bit

- This bit controls interrupt requests of bit 2 EDIR.
- When the EDIE bit is enabled, setting bit 2 EDIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 5] res : Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 4] OVIE: Overflow interrupt request enable bit

- This bit controls interrupt requests of bit 0 OVIR.
- When the OVIE bit is enabled, setting bit 0 OVIR generates an interrupt request to the CPU.

Bit	Description
0	Disables interrupt requests.
1	Enables interrupt requests.

[bit 3] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 2] EDIR: Measurement completion interrupt request bit

- This bit indicates that the completion of measurement. The flag is set to "1" when the measurement is completed.
- The EDIR bit is cleared by reading the measurement result (DTBF).
- The EDIR is read-only. Writing a value does not affect the bit value.

Bit	Description
0	Reads the measurement result (DTBF).
1	Detects an interrupt cause.

[bit 1] res: Reserved bit

The read value is "0".

Set "0" to this bit.

[bit 0] OVIR: Overflow interrupt request bit

- When a counter value overflow from 0xFFFF to 0x0000 occurs, the flag is set to "1".
- The OVIR bit is cleared by writing "0".
- Even if "1" is written to the OVIR bit, the bit value is not affected.
- The read value of read-modify-write instructions is "1" regardless of the bit value.

Bit	Description
0	Clears an interrupt cause.
1	Detects an interrupt cause.

9.4.3. Data Buffer Register (DTBF)

The Data Buffer Register (DTBF) is a register that reads the measured or count value of the PWC timer. In 32-bit mode, the value of the lower 16 bits is read for the even channel and that of the upper 16 bits for the odd channel.

Be sure to use the 16-bit data transfer instruction to read this register.

bit	15	0
Field	DTBF [15:0]	
Attribute	R	
Initial value	0x0000	

- The DTBF register is read-only in both continuous and one-shot measurement modes. Writing a value does not change the register value.
- In continuous measurement mode (TMCR bit 3 MDSE = 1), this register works as a buffer register that stores the previous measurement result.
- In one-shot measurement mode (TMCR bit 3 MDSE = 0), the DTBF register accesses the up counter directly. The count value can be read during counting. The measurement value is retained after the completion of measurement.
- Access the DTBF register with 16-bit data.

CHAPTER: Multifunction Timer

This chapter describes the multifunction timer unit.

1. Overview of Multifunction Timer
2. Configuration of Multifunction Timer
3. Operations of Multifunction Timer
4. Registers of Multifunction Timer
5. Other Matters

1. Overview of Multifunction Timer

The multifunction timer is a function block that enables three-phase motor control. In conjunction with PPG and ADC, it can provide a variety of motor controls. An overview of the multifunction timer is provided below.

■ Functions

The multifunction timer has the following functions.

- It can output PWM signals with any cycle/pulse length (PWM signal output function).
- It can start PPG in synchronization with PWM signal output. It can superimpose PPG's output signal on the PWM signal and output it (DC chopper waveform output function).
- It can generate a non-overlap signal that maintains the response time of the power transistor (dead time) from PWM signal output (dead timer function).
- It can capture the timing of changing the input signal and its pulse width in synchronization with PWM signal output (Input capture function).
- It can start ADC at any timing, in synchronization with PWM signal output (ADC start function).
- It performs noise canceling to the emergency motor shutdown interrupt signal (DTTIX input signal). It can set freely the pin state at the time of motor shutdown, when a valid signal input is detected (DTIF interrupt function).

■ Block Configuration

The multifunction timer (1 unit) consists of the following function blocks.

- Free-run Timer Unit : 3ch.
- Output Compare Unit : 6ch.(2ch.×3unit)
- Waveform Generator Unit : 3ch.
- Noise Canceler Unit : 1ch.
- Input Capture Unit : 4ch.(2ch.×2unit)
- ADC Start Compare Unit : 3ch
- ADC Start Trigger Selector Unit : 3ch

The multifunction timer is configured to enable one three-phase motor control, when 1 unit is used. Some models in this series contain multiple units of the multifunction timer, which are configured to support multiple three-phase motor controls.

■ Abbreviations

In this chapter, the following abbreviations are used in explanations.

- MFT : Multifunction Timer Unit
- PPG : Programmable Pulse Generator Unit
- FRT : Free-run Timer Unit
- FRTS : Free-run Timer Selector
- OCU : Output Compare Unit
- WFG : Waveform Generator Unit
- NZCL : Noise Canceler Unit
- ICU : Input Capture Unit
- ADCMP : ADC Start Compare Unit
- ATSA : ADC Start Trigger Selector Unit

2. Configuration of Multifunction Timer

This chapter describes the configuration of the multifunction timer and the functions of each function block and I/O pin.

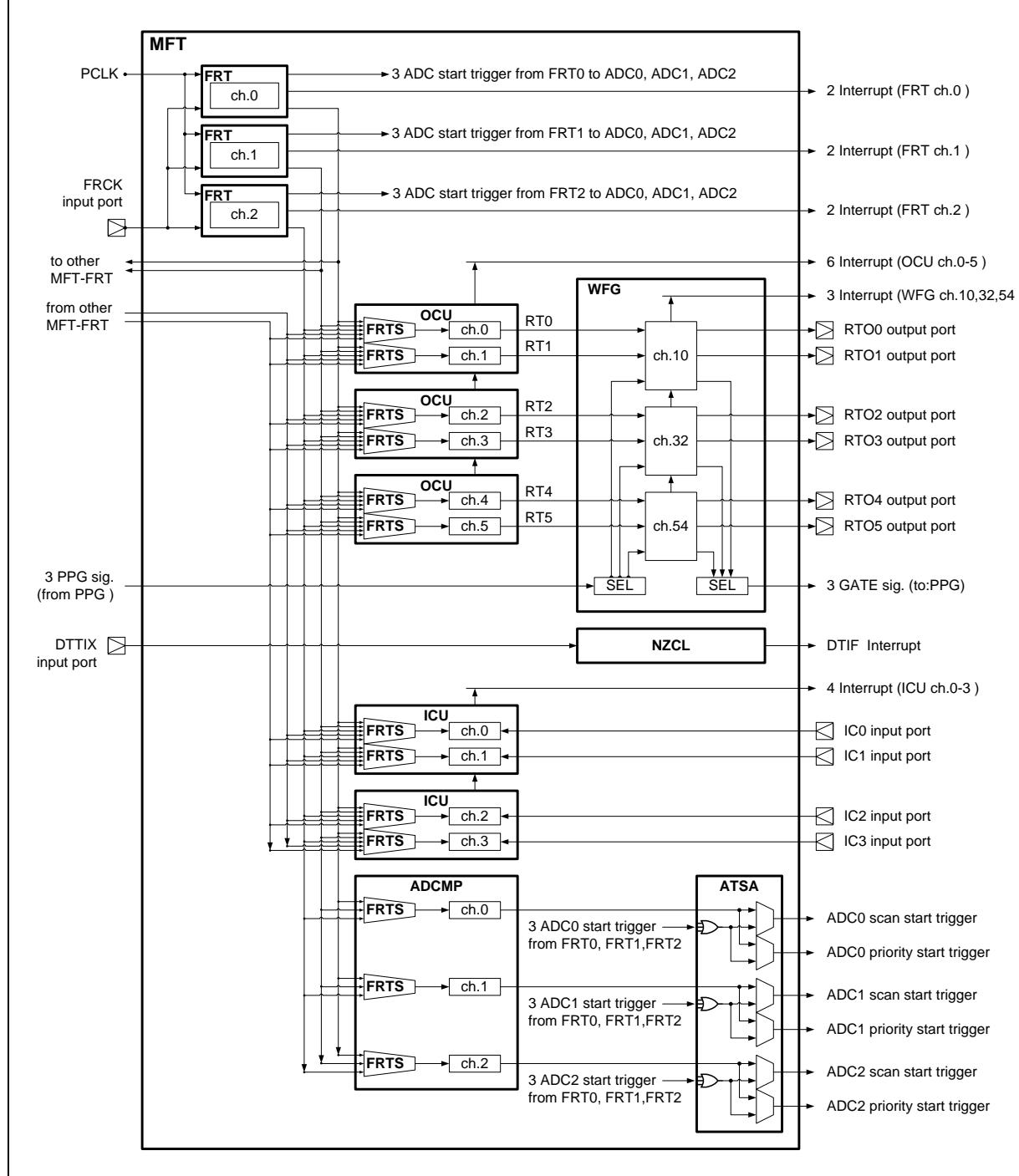
- 2.1 Block Diagram of Multifunction Timer
- 2.2 Description of Each Function Block
- 2.3 I/O Pins of Multifunction Timer Unit

2.1. Block Diagram of Multifunction Timer

■ Block Diagram

Figure 2-1 shows the block diagram of the entire function timer.

Figure 2-1 Block Diagram of Multifunction Timer

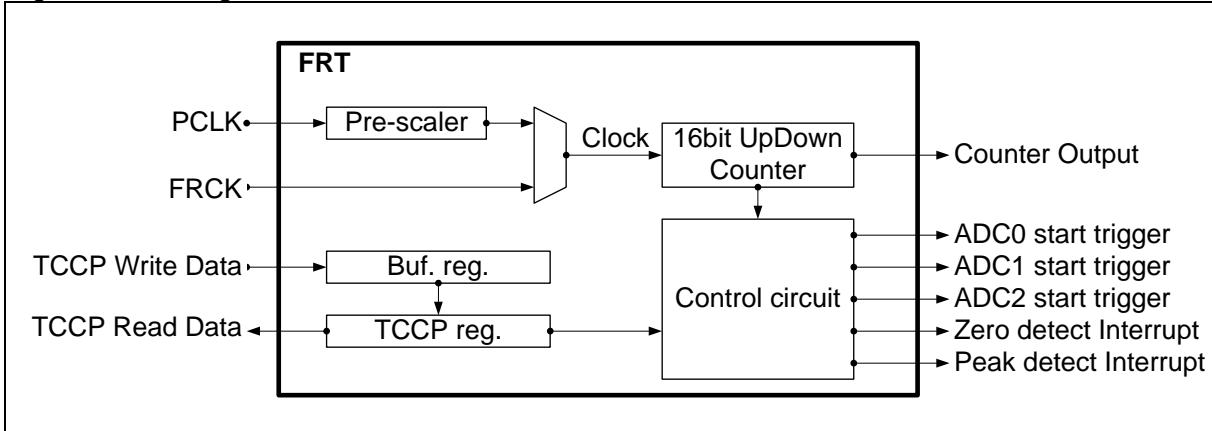


2.2. Description of Each Function Block

■ FRT: 3ch.

- FRT is a timer function block that outputs the reference counter value for the operation of each function block in MFT.
- FRT consists of a clock pre-scaler., 16-bit Up/Down counter, cycle setting register (TCCP register) and controller. Figure 2-2 shows the configuration of FRT.
 - The clock pre-scaler divides the peripheral clock (PCLK) signal in LSI to generate the operating clock for the 16-bit Up/Down counter.
 - The TCCP register sets the count cycle for the 16-bit Up/Down counter. It has a buffer register to change a cycle during count operation.
 - The 16-bit Up/Down counter performs Up-count or Down-count operation in the count cycle specified by the TCCP register in order to output a counter value.
 - The following processing can be achieved by instructing the controller via CPU.
 - The division ratio of the clock pre-scaler can be selected.
 - The use of PCLK (internal clock) and FRCK (external clock) can be selected.
 - The count mode for the 16-bit Up/Down-counter can be selected to specify whether to start or stop the count operation.
 - The buffer register function of the TCCP register can be enabled or disabled.
 - Interrupt can be generated to CPU by detecting a case where the count value is set to "0x0000" or the peak value (= TCCP value) (two interrupt signals output per FRT1 channel).
 - AD conversion start signals can be generated to each ADC by detecting a case where the count value is set to "0x0000" (three AD conversion start signals output per FRT1 channel).
 - Each MFT unit is in a 3-channel configuration with three FRT's, which can operate independently from one another.
 - Inside MFT, the output of the FRT counter value is connected to OCU, ICU and ADCMP. These units have a circuit (FRTS) that selects FRT to be connected. Interlocked operation can be performed based on the output of the counter value of the selected FRT. All of the units can be interlocked by a single FRT, or 2 or 3 groups can be formed as interlocked operational groups.

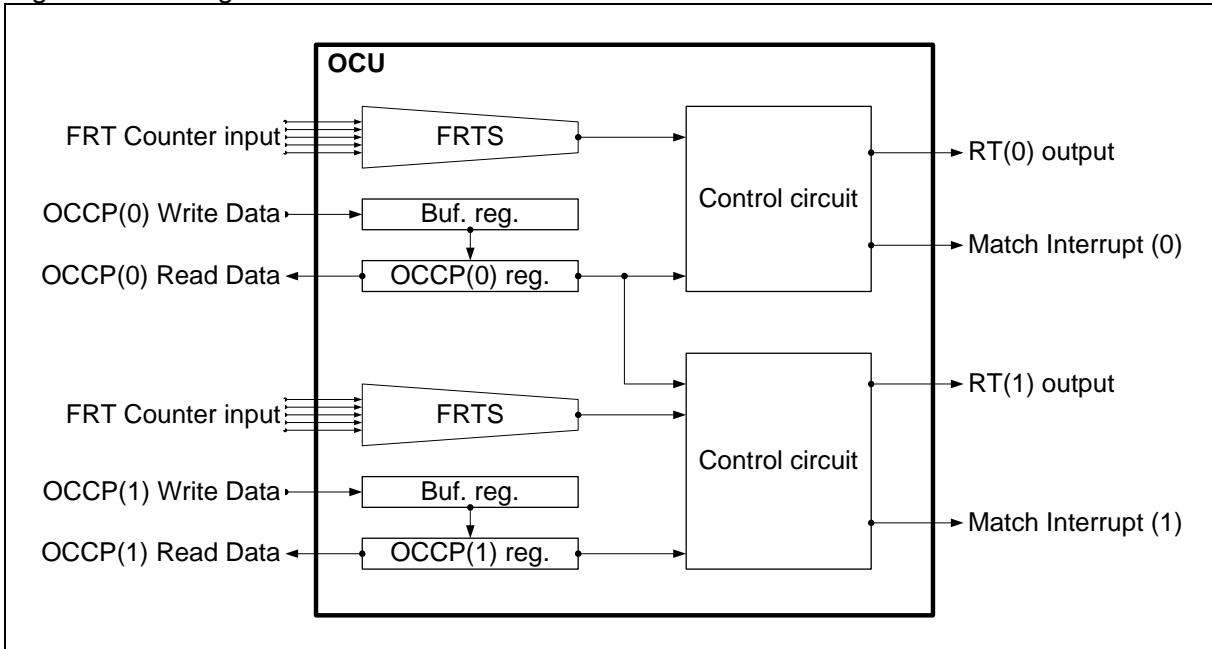
Figure 2-2 Configuration of FRT



■ OCU: 6ch. (2ch. ×3)

- OCU is a function block that generates and outputs PWM signals based on the counter value of FRT. The signal names of PWM signals output by OCU are RT0 to RT5. These signals are output to LSI's external output pins via WFG.
- OCU consists of FRTS, compare value store register (OCCP register) and controller. The basic unit is in a 2-channel configuration with two sets of each circuit. Figure 2-3 shows the configuration of OCU.
- FRTS is a circuit that selects the counter value of FRT to be connected to OCU for use.
- The OCCP register specifies the timing of changing the PWM signal as the compare value for the FRT counter value. It has a buffer register to enable data to be written to the OCCP register asynchronously from FRT's count operation.
- The following processing can be achieved by instructing the controller from CPU.
 - FRT to be connected to OCU can be selected.
 - Whether to enable or disable OCU's operation can be specified.
 - The output level of the RT0 to RT5 signal can be specified directly when OCU's operation is disabled.
 - The output level of RT0 to RT5 signal changes, if OCU's operation is enabled, the FRT counter value is compared with the value of the compare value store register and then it is detected that these values match. Signals with any cycle or pulse length can be output by setting the value of the OCCP register beforehand.
 - The following modes are available for the change conditions of the RT0 to RT5 signal and can be selected:
 - Up-count, 1-change mode
 - Up-count, 2-change mode
 - Up/Down-count, Active High mode
 - Up/Down-count, Active Low mode
 - Interrupts can be generated to CPU, when it is detected that the value of the OCCP register matches the FRT counter value.
 - Whether or not to use the buffer register of the OCCP register can be specified and the timing of transfer from the buffer register can be selected.
 - Each MFT contains three of these OCU's and consists of a total of 6 compare registers, 6 output signal pins and 6 interrupt outputs (2-channel × 3-unit configuration).

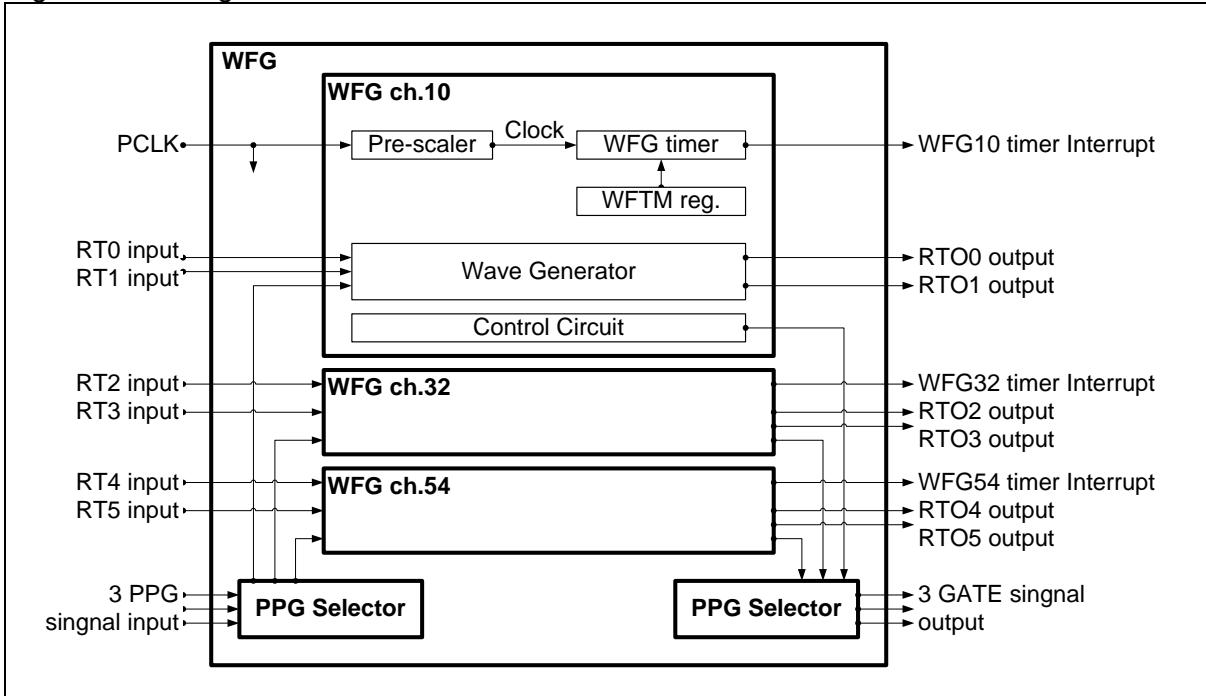
Figure 2-3 Configuration of OCU



■ WFG: 3ch

- WFG is a function block that is located at the back of OCU and generates signal waveforms for motor control from the RT0 to RT5 and PPG signals (PPG is located outside the multifunction timer).
- The signal outputs to LSI external pins from WFG are called "RTO0" to "RTO5". They are divided into blocks: the block that outputs RTO0 and RTO1 from RT0 and RT1; the block that outputs RTO2 and RTO3 from RT2 and RT3; and the block that outputs RTO4 and RTO5 from RT4 and RT5. They are called "WFG ch.10", "WFG ch.32" and "WFG ch.54", respectively.
- WFG consists of a clock pre-scaler, 16-bit timer (WFG timer), WFG timer initial value register (WFTM register), waveform generator, PPG timer unit selectors and controller. Figure 2-4 shows the configuration of WFG.
- The clock pre-scaler divides the peripheral clock signal (PCLK) in LSI to generate the operating clock for the WFG timer.
- The WFG timer is a timer circuit that counts the time set by the WFTM register and generates signal waveforms. In an operation mode that does not use a timer to generate a waveform, it can be used as a single reload timer, allowing interrupts to be generated regularly to CPU. Each WFG timer has one WFG timer interrupt output.
- The WFTM register can be used to set any count time to the WFG timer.
- The waveform generator is a block that generates LSI external output signals through waveform generation processing based on the RT0 to RT5 signals from OCU, signals from PPG and the count state of the WFG timer.
- The PPG timer unit selector is a circuit that selects the PPG timer unit to be used by WFG. It selects the output destination of the instruction signal (GATE signal) for PPG activation and the PPG output signal.
- The following processing can be achieved by instructing the controller from CPU.
 - The division ratio of the clock pre-scaler can be selected.
 - The following modes are available and can be selected for wave generation.
 - Through mode
 - RT-PPG mode
 - Timer-PPG mode
 - RT dead timer mode
 - PPG dead timer mode
 - GATE signal can be output to instruct PPG to start up.
 - Output polarity can be reversed in RT dead timer mode and PPG timer mode.

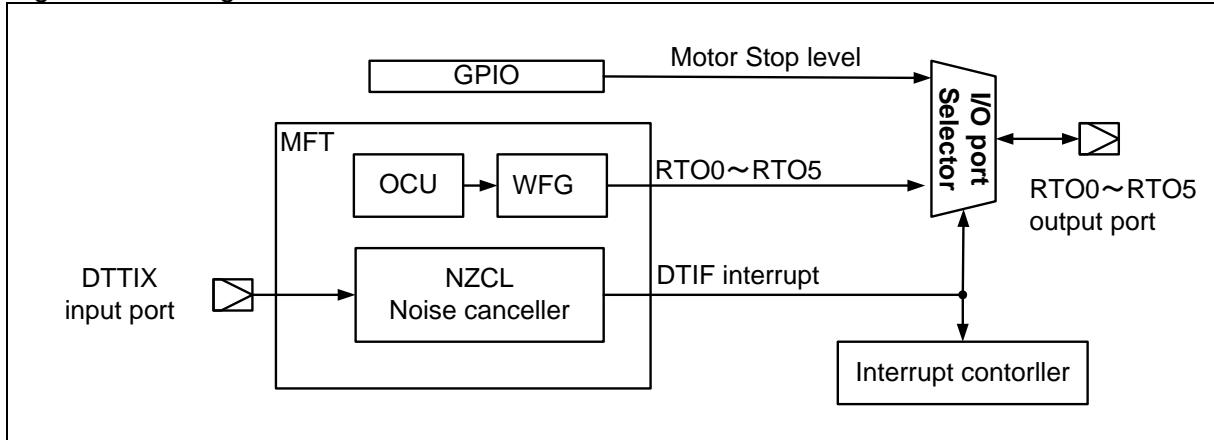
Figure 2-4 Configuration of WFG



■ NZCL

- NZCL is a function block that performs noise cancellation to the external interrupt input signal (DTTIX signal) for emergency shutdown of the motor and generates DTIF interrupts to CPU.
- NZCL consists of a noise canceller and controller.
- It can be switched to the state of the GPIO port which is also used for WFG's external output signals (RTO0 to RTO5) using the selection function of the I/O port block while DTIF interrupt is being generated. Emergency shutdown of the motor can be performed by setting the I/O state of the GPIO port to the Motor Stop level.
- Figure 2-5 shows the configuration of NZCL and I/O port selector.

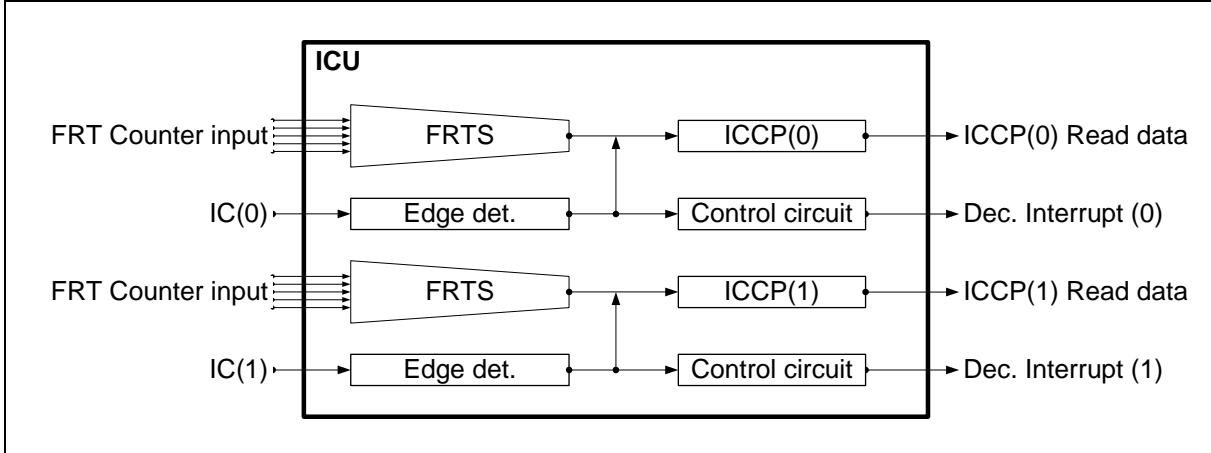
Figure 2-5 Configuration of NZCL and I/O Port Selector



■ ICU: 4ch. (2ch. ×2)

- ICU is a function block that captures the FRT count value and generates an interrupt to CPU when a valid edge is detected at an external input pin signal.
- ICU consists of FRTS, edge detector, 16-bit capture register and control register. Its basic unit is in a 2-channel configuration with two sets of each circuit. Figure 2-6 shows the configuration of ICU.
- FRTS is a circuit that selects the counter value of the FRT to be connected to ICU for use.
- The edge detector is a circuit that detects the valid edge of the input signal.
- The ICCP register captures the timing of changing the input signal as FRT's count value.
- The following processing can be achieved by instructing the controller from CPU.
 - FRT to be connected to ICU can be selected.
 - The valid edge of the input signal can be selected from rising edge, falling edge and both edges.
 - Whether to enable or disable ICU's operation can be specified.
 - An interrupt can be generated to CPU when the valid edge is detected and the capture operation is performed.
- Each MFT contains 2 ICU's and consists of a total of 4 external input pins and 4 capture registers (2-channel × 2-unit configuration). LSI external input signals to ICU are called "IC0" to "IC3".
- Some ICU input signals can be switched to LSI's internal signals for use, other than LSI external pins, using the selection function of the I/O port block (see "2.3 I/O Pins of Multifunction Timer Unit" for more details).

Figure 2-6 Configuration of ICU



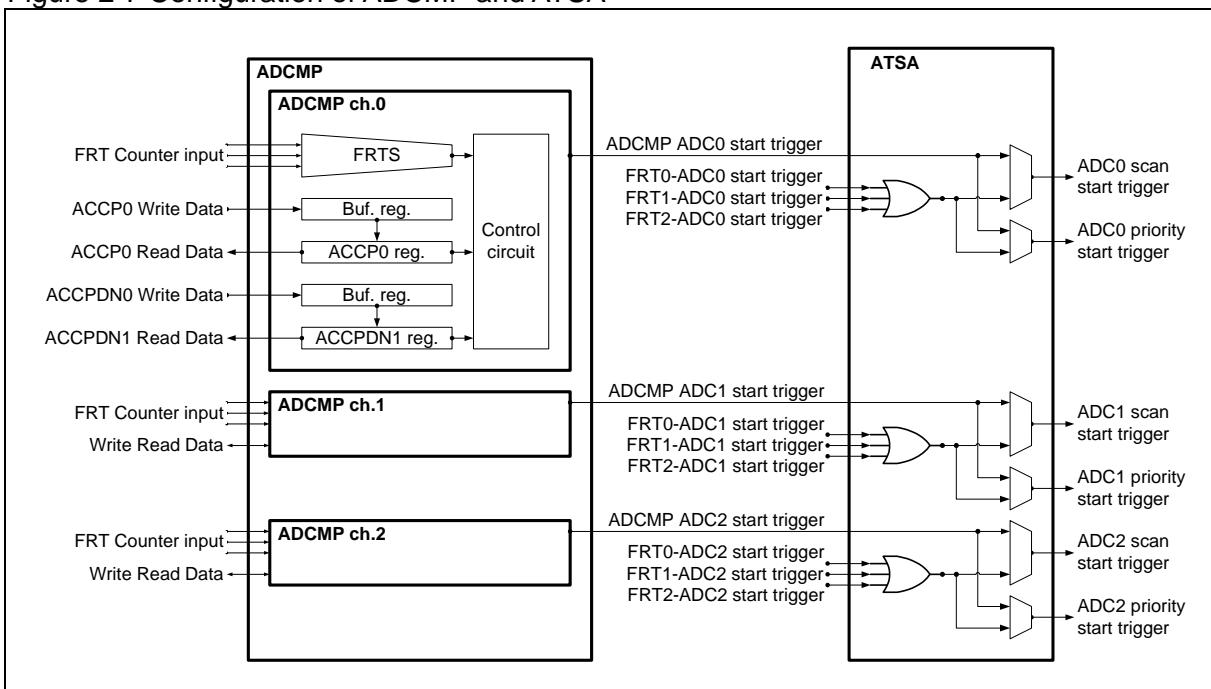
■ ADCMP: 3ch.

- ADCMP is a function block that generates AD conversion start signals at any timing of FRT cycle.
- ADCMP is in a 3-channel configuration with each corresponding to one of the 3 units of ADC mounted.
- ADCMP consists of FRTS, two 16-bit compare registers (ACCP register and ACCPDN register) and control register. Figure 2-7 shows the configuration of ADCMP and ATSA.
- FRTS is a circuit that selects the counter value of the FRT to be connected to ADCMP for use.
- The ACCP register and ACCPDN register specify the timing of starting AD conversion as the compare value of FRT's counter value. Each has a buffer register so that writing to the ACCP register and ACCPDN register can be done asynchronously from FRT's count operation.
- The following processing can be achieved by instructing the controller from CPU.
 - FRT to be connected to ADCMP can be selected.
 - Whether to enable or disable ADCMP operation can be specified.
 - The timing of starting AD conversion can be set with a specified FRT count direction.
 - Whether or not to use the ACCP register and ACCPDN's buffer register can be specified and the timing of transferring from the buffer register can be selected.

■ ATSA: 3ch.

- ATSA is a function block that selects and outputs ADC's start signals based on the value of the control register.
- ATSA consists of a logic OR circuit for AD start signals and a circuit that selects the AD start signals from ADCMP.
- The following is processed.
 - Start signals from FRT to each ADC undergo logic OR operation for each corresponding ADC unit.
 - The above ADC start signal and other ADC start signal from ADCMP are selected by register setting.
 - There are two types of ADC start triggers: scan start triggers and priority start triggers. The configuration allows each to be selected and output.
- ATSA is configured to support each start trigger of three mounted ADC units. It outputs 6 ADC start signals.

Figure 2-7 Configuration of ADCMP and ATSA



2.3. I/O Pins of Multifunction Timer Unit

■ Correspondence with LSI External I/O Pins

Of all the I/O signals illustrated in the block diagrams, Table 2-1 shows a list of correspondence between I/O pins of MFT unit and LSI external I/O pins. In this series, some models have more than one MFT unit. Therefore, LSI pin names are composed of I/O pin names shown in the block diagrams plus MFT's unit number (0, 1, 2). It should be noted that descriptions in this chapter are based on the pin names shown in the block diagrams.

Table 2-1 Correspondence Table for I/O Pins of MFT Unit and LSI External I/O Pins

Name of MFT Unit Pin (Pin Name in Block Diagram)	Function	Name of LSI Pin	
		MFT-unit0	MFT-unit1
FRCK	FRT external input clock	FRCK0	FRCK1
DTTIX	Motor emergency shutdown interrupt input	DTTI0X	DTTI1X
RTO0	WFG-PWM output ch.0	RTO00	RTO10
RTO1	WFG-PWM output ch.1	RTO01	RTO11
RTO2	WFG-PWM output ch.2	RTO02	RTO12
RTO3	WFG-PWM output ch.3	RTO03	RTO13
RTO4	WFG-PWM output ch.4	RTO04	RTO14
RTO5	WFG-PWM output ch.5	RTO05	RTO15
IC0	ICU input ch.0	IC00	IC10
IC1	ICU input ch.1	IC01	IC11
IC2	ICU input ch.2	IC02	IC12
IC3	ICU input ch.3	IC03	IC13

ICU's input pins can be switched with the following LSI internal signals, in addition to the external pin inputs, using the selector function of the I/O port block.

- SYNC signal when the LYN function of the multifunction serial block is used
- Internal CR oscillator/oscillation frequency trimming input signal

For details, see the section regarding the I/O port block.

■ Interrupt Signal Outputs

Of all the I/O signals illustrated in the block diagrams, Table 2-2 shows a list of interrupt signals generated from the MFT unit. Any model that contains more than one MFT unit has interrupt outputs equivalent to the number of mounted MFT units.

Table 2-2 List of Interrupt Signals Generated from MFT Unit

Generation Block	Interrupt Type
FRT ch.0	Zero value detection interrupt
FRT ch.1	Zero value detection interrupt
FRT ch.2	Zero value detection interrupt
FRT ch.0	Peak value detection interrupt
FRT ch.1	Peak value detection interrupt
FRT ch.2	Peak value detection interrupt
OCU ch.0	Match detection interrupt
OCU ch.1	Match detection interrupt
OCU ch.2	Match detection interrupt
OCU ch.3	Match detection interrupt
OCU ch.4	Match detection interrupt
OCU ch.5	Match detection interrupt
ICU ch.0	Input signal edge detection interrupt
ICU ch.1	Input signal edge detection interrupt
ICU ch.2	Input signal edge detection interrupt
ICU ch.3	Input signal edge detection interrupt
NZCL	DTIF interrupt (emergency motor shutdown interrupt)
WFG ch.10	WFG timer 10 interrupt
WFG ch.32	WFG timer 32 interrupt
WFG ch.54	WFG timer 54 interrupt

■ Other I/O Signals

Of all the I/O signals illustrated in the block diagrams, the following section describes the other signals.

● PCLK

This is an LSI internal peripheral clock signal used in the MFT unit. It uses the clock signal of the APB bus to be connected. FRT (when the LSI internal peripheral clock is selected) and the WFG timer operate based on the count clock divided from PCLK.

● FRT input and FRT output of external MFT

A model containing more than one MFT unit can use FRT count output for the other MFT. This connection configuration allows OCU and ICU mounted separately on multiple MFT units to be interlocked by a single FRT.

(A model containing 2 MFT units can output 12 channels of PWM simultaneously. A model containing 3 MFT units can output 18 channels of PWM simultaneously.

For details, see "5.1 Connection of Model Containing Multiple MFT's".

● GATE signal / PPG signal

GATE signal is PPG's start signal that is output from MFT and input to PPG. PPG signal is output from PPG and input to MFT. PPG units to be connected for these signals vary depending on the mounted MFT unit. For details of their connection, see "5.1 Connection of Model Containing Multiple MFT's".

● AD conversion start signal

A total of 6 AD conversion start signals are output: scan start signals and priority start signals for each of the three units of ADC.

In models containing more than one MFT unit, start signals undergo logic OR for each ADC unit and are used in each ADC unit. For details, see the chapter "A/D Converter".

3. Operations of Multifunction Timer

This chapter provides examples of operations of the multifunction timer and explains its setting procedures.

- 3.1 Example of Operation of Multifunction Timer - 1
- 3.2 Example of Operation of Multifunction Timer - 2

3.1. Example of Operation of Multifunction Timer - 1

"Example of Operation of Multifunction Timer - 1" explains the cases where each function block is operated in the following modes:

- FRT : Up-count mode, without interrupt
- OCU : Up-count mode (1-change), with interrupt
- WFG : RT-PPG mode, generation of GATE signal, superimposition of PPG signal
- ICU : Rising edge detection mode, with interrupt

■ Time Chart

Figure 3-1 Time Chart of Main Registers and I/O Signals of Each Block

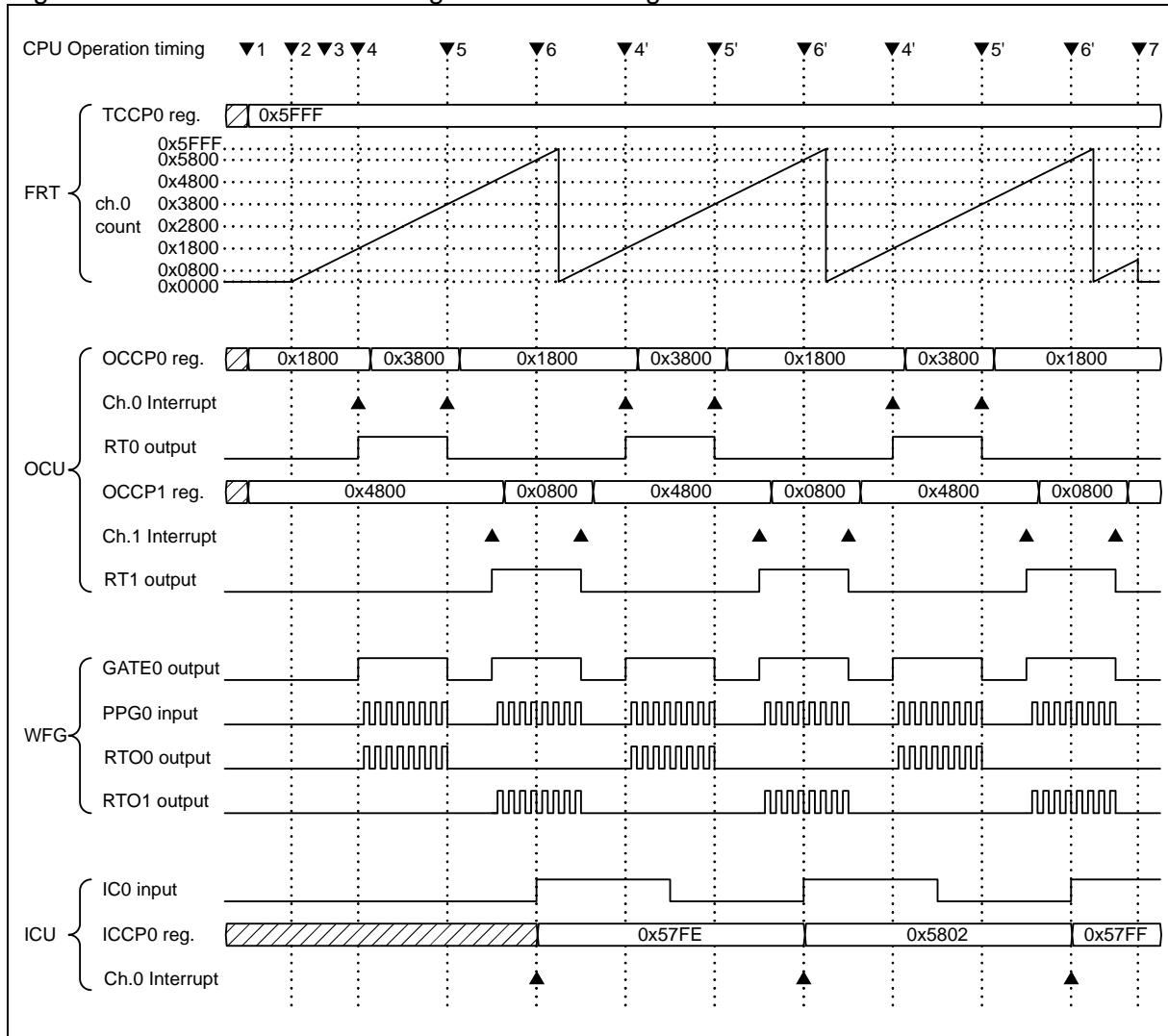


Figure 3-1 shows the time chart of main registers and I/O signals for each block. From top to bottom, the figure indicates CPU operation, FRT operation, OCU operation, WFG operation and ICU operation. The following section describes the operation of each function block and what is controlled from CPU at the timing 1 to 7. It also shows specific examples of CPU register settings at each of the timings. For details of register settings, see "4 Registers of Multifunction Timer". It should be noted that in addition to the above, LSI I/O port block, interrupt control block and PPG must be set separately.

■ Operation of FRT, OCU

Timing 1

- Set Up-count operation to FRT-ch.0 (TCSA0 register write).
- Set an operating cycle to FRT-ch.0 (TCCP0 register write). In this example, "0x5FFF" is set. When the pre-scaler is set to 1/128 and PCLK to 40MHz, the count cycle of FRT is 78.6432ms.
- Connect and set FRT-ch.0 to OCU-ch.0/ch.1 (OCFS10 register write).
- Set OCU-ch.0/ch.1 to Up-count mode (1-change) operation. Also specify the initial output level of output signals (RT0, RT1) (OCSA10, OCSB10 and OCSC register write).
- Set the timing of changing the output signal (RT0) for OCU-ch.0 (OCCP0 register write). In this example, "0x1800" is set. The value is written to the buffer register, and then transferred to the OCCP0 register.
- Set the timing of changing the output signal (RT1) for OCU-ch.1 (OCCP1 register write). In this example, "0x4800" is set. The value is written to the buffer register, and then transferred to the OCCP1 register.

Timing 2

- Instruct FRT-ch.0 to start count operation (TCSA0 register write).
- In Up-count mode, FRT-ch.0 starts counting from "0x0000" and continues the Up-count operation until the TCCP value is reached (=0x5FFF), as shown in the figure. Then, it returns to "0x0000" and continues counting.

Timing 3

- Instruct OCU-ch.0/ch.1 to enable the operation (OCSA10 register write).

Timing 4

- When OCU-ch.0 detects that the value of the FRT counter has reached "0x1800" and matched the setting value of OCCP0, it changes the output signal (RT0) from the Low to High level. It also generates an interrupt to CPU.
- CPU determines that an interrupt has been generated from OCU-ch.0, because "1" is set to the match detection flag of OCU-ch.0 (OCSA10 register read).
- Update the timing of changing the output signal (RT0) for OCU-ch.0 to "0x3800" (OCCP0 register write).
- CPU clears the match detection flag and returns from the interrupt (OCSA10 register write).

Timing 5

- When OCU-ch.0 detects that the value of the FRT counter matches the value of OCCP0, it changes the output signal (RT0) from the High to Low level. It also generates an interrupt to CPU.
- CPU identifies the interrupt from OCU-ch.0 (OCSA10 register read).
- Update the OCCP0 register of OCU-ch.0 to "0x1800" (OCCP0 register write).
- CPU clears the match detection flag and returns from the interrupt (OCSA10 register write).

After that, repeat Operations 4 and 5 so that PWM waveform in the FRT cycle can be achieved for the RT0 output signal, as shown in the figure. Similarly, PWM output waveform can also be achieved for the RT1 by updating the value of the OCCP1 register every time an interrupt occurs.

■ Operation of WFG

Timing 1

- Set RT-PPG mode operation to WFG-ch.10 (WFSA10 register write).

Timing 4

- When the RT0 signal from OCU-ch.0 is changed to the High level, WFG asserts the GATE signal (GATE0) and instructs PPG-ch.0 to start.
- When the GATE signal is asserted, PPG-ch.0 starts the output of the PPG signal (PPGO).
- WFG superimposes and outputs the PPG signal to RTO0 while the RT0 signal remains at the High level.

Timing 5

- When the RT0 signal is changed to the Low level, WFG deasserts the GATE signal and gives a stop instruction.
- PPG-ch.0 changes the PPG signal to the Low level and stops the output.
- WFG changes the RTO0 signal to the Low level and stops the output.

WFG performs the same operation to the RT1 signal from OCU-ch.1, and superimposes and outputs the PPG signal to RTO1. DC chopper control waveform, as shown in the figure, can be output to RTO0/RTO1 by using the WFG function.

■ Operation of ICU

Timing 1

- Connect and set FRT-ch.0 to ICU-ch.0/ch.1 (ICFS10 register write).
- Set rising edge detection operation to ICU-ch.0 (ICSA10 register write).

Timing 6

- When a rising edge is detected at the input signal (IC0), ICU-ch.0 stores FRT's count value to the ICCP0 register. It also generates an interrupt to CPU.
- CPU determines that an interrupt has been generated from ICU-ch.0, because "1" is set to the valid edge detection flag of ICU-ch.0 (ICSA10 register read).
- CPU captures the position of the rising edge of the signal (ICCP0 register read).
- CPU clears the valid edge detection flag and returns from the interrupt (ICSA10 register write).

■ Completion of Processing

Timing 7

The processing at Timing 7 indicates the procedure for completing the output of the PWM signal.

- Disable the operation of OCU-ch.0 and ch.1 (OCSA10 register write).
- Set the level of the output signals (RT0, RT1) for OCU-ch.0 and ch.1 (OCSB10 register write).
- Disable the operation of ICU-ch.0 (ICSA10 register write).
- When the output of OCU stops, WFG does not perform its operation.
- Instruct FRT-ch.0 to stop the count operation (TCSA0 register write).
- Set "0x0000" to FRT's count value (TCDT0 register write).

■ Processing of Other Channels

The above example explained the operation with 2 channels of OCU, one channel of WFG and one channel of ICU. If OCU-6ch, WFG-3ch, and ICU-3ch are connected to the same FRT to perform interlocking control, three-phase motor control can be achieved.

■ Details of Register Settings

Register settings in Example of Operation of Multifunction Timer - 1 are detailed below.

The meanings of the symbols in the table are as follows:

Operation	HW	Half-word write access
	BW	Byte write access
	HR	Half-word read access
	BR	Byte read access
Value	NM	Indicates either writing the same value as the register value that has already been set or reading from the register to write the original value (No Modify).
	1(RMW)	Indicates writing "1", if register clear is not intended. In the case of update by RMW access (see "5.2 Treatment of Event Detect Register and Interrupt"), it indicates that the read value can be written back.
	Other	Indicates setting bits of other channels and no relation to this explanatory example.
	DC	Indicates no relation to the read value (Don't Care).

Table 3-1 Example of Operation 1 – Register Settings 1

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
1	FRT	TCSA0	HW	CLK[3:0]	0111	Clock division pre-scaler setting: 1/128
				SCLR	0	Soft clear: Do nothing
				MODE	0	Count mode setting: Up-count mode
				STOP	1	FRT count operation: Stop counting
				BFE	1	TCCP buffer function: Enable
				ICRE	0	Peak value detection interrupt: Disable
				ICLR	0	Peak value detection: Clear
				Reserved	000	-
				IRQZE	0	Zero value detection interrupt: Disable
				IROZF	0	Zero value detection: Clear
				ECKE	0	Selection of clock used: Internal clock
	TCCP0	HW	TCCP	0x5FFF		Set FRT cycle
	OCU	OCFS10	BW	FS00[3:0]	0000	FRT connected to ch.0: FRT ch.0
				FS01[3:0]	0000	FRT connected to ch.1: FRT ch.0
		OCSA10	BW	CST0	0	ch.0 operation state: Operation disable
				CST1	0	ch.1 operation state: Operation disable
				BDIS0	1	ch.0 OCCP buffer function: Disable
				BDIS1	1	ch.1 OCCP buffer function: Disable
				IOE0	1	ch.0 interrupt: Enable
				IOE1	1	ch.1 interrupt: Enable
				IOP0	0	ch.0 match detection: Clear
				IOP1	0	ch.1 match detection: Clear
	OCSB10	OCSC	BW	OTD0	0	RT0 output level initial setting: Low
				OTD1	0	RT1 output level initial setting: Low
				Reserved	00	-
				CMOD	0	ch.0/ch.1 operation mode: Up-count (1-change)
				BTS0	0	ch.0 buffer transfer: don't care
				BTS1	0	ch.1 buffer transfer: don't care
				Reserved	0	-
				MOD0	0	ch.0 operation mode: Up-count (1-change)
				MOD1	0	ch.1 operation mode: Up-count (1-change)
				MOD2	Other	ch.2 operation mode:
				MOD3	Other	ch.3 operation mode:
				MOD4	Other	ch.4 operation mode:
				MOD5	Other	ch.5 operation mode:
				Reserved	00	-
	OCCP0	HW	OCCP	0x1800		Specify ch.0 change timing
	OCCP1	HW	OCCP	0x4800		Specify ch.1 change timing

Table 3-2 Example of Operation 1 – Register Settings 2

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
1	WFG	WFSA10	HW	DCK[2:0]	000	Clock division pre-scaler setting: 1/1 (don't care)
				TMD[2:0]	001	Operation mode: Select RT-PPG mode
				GTEN[1:0]	11	Gate signal generation: RT0, RT1 signal logic OR
				PSEL[1:0]	00	Connecting PPG : PPG0
				PGEN[1:0]	11	PPG reflection: Logic AND of PPG signal to RTO0/RTO1 signals
				DMOD	0	Output polarity: don't care
				Reserved	000	-
2	ICU	ICFS10	BW	FSI0[3:0]	0000	FRT connected to ch.0: FRT ch.0
				FSI1[3:0]	Other	FRT connected to ch.1:
		ICSA10	BW	EG0[1:0]	01	ch.0 operation state: Operation enabled, rising edge
				EG1[1:0]	Other	ch.1 operation state:
				ICE0	1	ch.0 interrupt: Enable
				ICE1	Other	ch.1 interrupt:
				ICP0	0	ch.0 edge detection: Clear
				ICP1	Other	ch.1 edge detection:
3	FRT	TCSA0	HW	CLK[3:0]	NM	Clock division pre-scaler setting:
				SCLR	NM	Soft clear:
				MODE	NM	Count mode setting:
				STOP	0	FRT count operation: Start counting
				BFE	NM	TCCP buffer function:
				ICRE	NM	Peak value detection interrupt:
				ICLR	1(RMW)	Peak value detection: Do nothing
				Reserved	NM	-
				IRQZE	NM	Zero value detection interrupt:
				IRQZF	1(RMW)	Zero value detection: Do nothing
3	OCU	OCSA10	BW	ECKE	NM	Selection of clock used: Internal clock
				CST0	1	ch.0 operation state: Operation enabled
				CST1	1	ch.1 operation state: Operation enabled
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Do nothing
				IOP1	1	ch.1 match detection: Do nothing

Table 3-3 Example of Operation 1 – Register Settings 3

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
4	OCU	OCSA10	BR	CST0	DC	ch.0 operation state:
				CST1	DC	ch.1 operation state:
				BDIS0	DC	ch.0 OCCP buffer function:
				BDIS1	DC	ch.1 OCCP buffer function:
				IOE0	DC	ch.0 interrupt:
				IOE1	DC	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Match detected
				IOP1	0	ch.1 match detection: Match not detected
		OCCP0	HW	OCCP0	0x3800	Specify ch.0 change timing
		OCSA10	BW	CST0	NM	ch.0 operation state:
				CST1	NM	ch.1 operation state:
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	0	ch.0 match detection: Flag cleared
				IOP1	1(RMW)	ch.1 match detection: Do nothing
5	OCU	OCSA10	BR	CST0	DC	ch.0 operation state:
				CST1	DC	ch.1 operation state:
				BDIS0	DC	ch.0 OCCP buffer function:
				BDIS1	DC	ch.1 OCCP buffer function:
				IOE0	DC	ch.0 interrupt:
				IOE1	DC	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Match detected
				IOP1	0	ch.1 match detection: Match not detected
		OCCP0	HW	OCCP0	0x1800	Specify ch.0 change timing
		OCSA10	BW	CST0	NM	ch.0 operation state:
				CST1	NM	ch.1 operation state:
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	0	ch.0 match detection: Flag cleared
				IOP1	1(RMW)	ch.1 match detection: Do nothing

Table 3-4 Example of Operation 1 – Register Settings 4

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
6	ICU	ICSA10	BR	EG0[1:0]	DC	ch.0 operation state:
				EG1[1:0]	DC	ch.1 operation state:
				ICE0	DC	ch.0 interrupt:
				ICE1	DC	ch.1 interrupt:
				ICP0	1	ch.0 edge detection: Edge detected
				ICP1	0	ch.1 edge detection: Edge not detected
		ICCP0	HW	ICCP0	0x57FE	Capture ch.0 capture value
		ICSA10	BW	EG0[1:0]	NM	ch.0 operation state:
				EG1[1:0]	NM	ch.1 operation state:
				ICE0	NM	ch.0 interrupt:
				ICE1	NM	ch.1 interrupt:
				ICP0	0	ch.0 edge detection: Clear
				ICP1	1(RMW)	ch.1 edge detection: Do nothing
7	OCU	OCSA10	BW	CST0	0	ch.0 operation state: Disabled
				CST1	0	ch.1 operation state: Disabled
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Do nothing
				IOP1	1	ch.1 match detection: Do nothing
		OCSB10	BW	OTD0	0	RT0 output level: Low
				OTD1	0	RT1 output level: Low
				Reserved	NM	-
				CMOD	NM	ch.0/ch.1 operation mode:
				BTS0	NM	ch.0 buffer transfer:
				BTS1	NM	ch.1 buffer transfer:
				Reserved	NM	-
		ICSA10	BW	EG0[1:0]	00	ch.0 operation state: Operation disabled
				EG1[1:0]	00	ch.1 operation state: Operation disabled
				ICE0	NM	ch.0 interrupt:
				ICE1	NM	ch.1 interrupt:
				ICP0	1	ch.0 edge detection: Do nothing
				ICP1	1	ch.1 edge detection: Do nothing
7	FRT	TCSA0	HW	CLK[3:0]	NM	Clock division pre-scaler setting:
				SCLR	1	Soft clear: Initialize FRT
				MODE	NM	Count mode setting:
				STOP	1	FRT count operation: Stop counting
				BFE	NM	TCCP buffer function:
				ICRE	NM	Peak value detection interrupt:
				ICLR	1	Peak value detection: Do nothing
				Reserved	NM	-
				IRQZE	NM	Zero value detection interrupt:
				IRQZF	1	Zero value detection: Do nothing
				ECKE	NM	Selection of clock used:
		TCDT0	HW	TCDT	0x0000	Initialize FRT count value

3.2. Example of Operation of Multifunction Timer - 2

"Example of Operation of Multifunction Timer - 2" explains the cases where each function block is operated in the following modes:

- FRT : Up/Down-count mode, with interrupt
- OCU : Up/Down-count mode (Active High), without interrupt
- WFG : RT-dead timer mode (Active High)
- ADCMP/ATSA : Instruct ADCunit0 to start scan conversion under the match condition for Up-counting

■ Time Chart

Figure 3-2 Time Chart of Main Registers and I/O Signals of Each Block

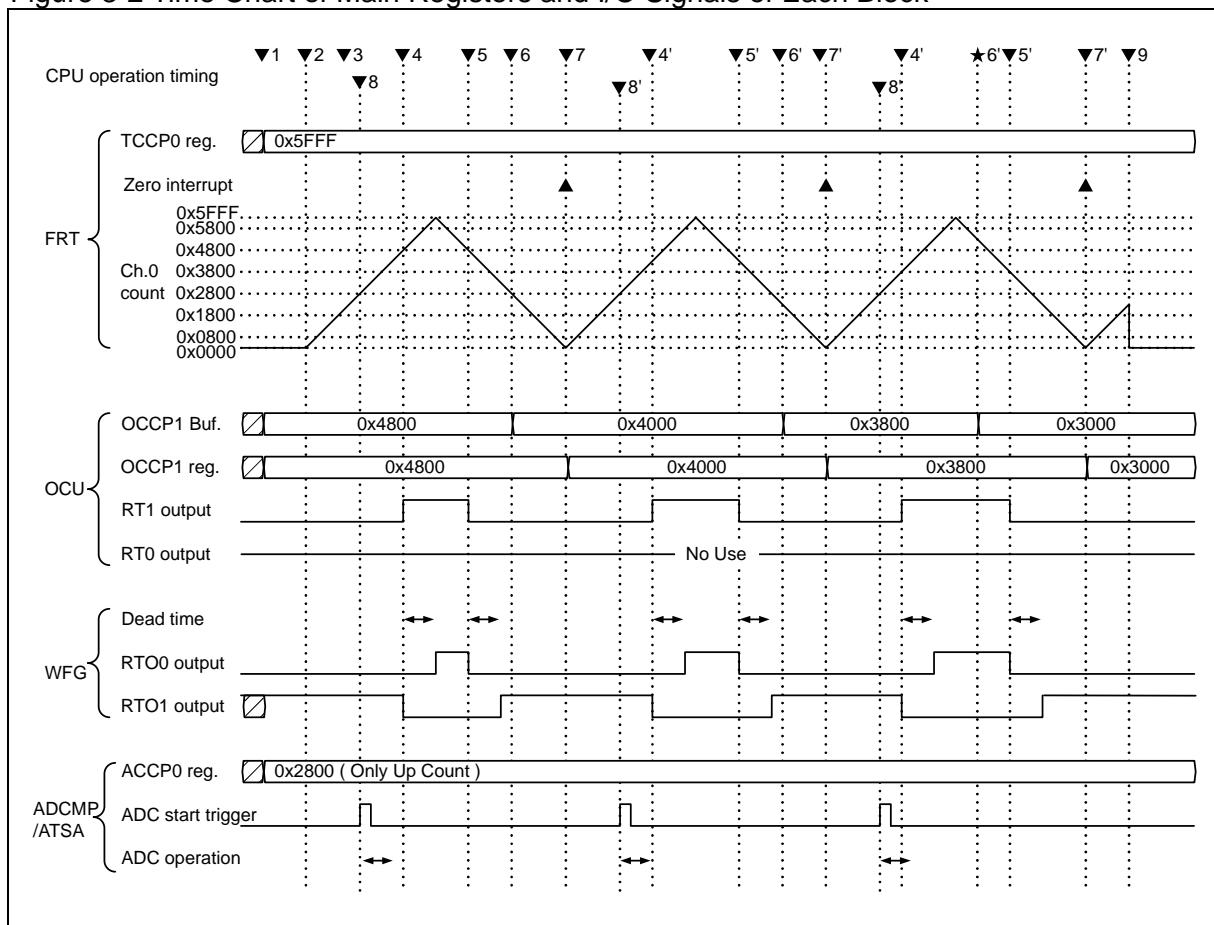


Figure 3-2 shows the time chart of main registers and I/O signals for each block. From top to bottom, the figure indicates CPU operation, FRT operation, OCU operation, WFG operation and ADCMP operation.

The following section describes the operation of each function block and what is controlled from CPU at the timing 1 to 9. It also shows specific examples of CPU register settings at each of the timings. For details of register settings, see "4 Registers of Multifunction Timer". It should be noted that in addition to the above, LSI I/O port block, interrupt control block and ADC must be set separately.

■ Operation of FRT, OCU

Timing 1

- Set Up-count operation to FRT-ch.0 (TCSA0 register write).
- Set an operating cycle to FRT-ch.0 (TCCP0 register write). In this example, "0x5FFF" is set. When the FRT pre-scaler is set to 1/4 and PCLK to 40MHz, the count cycle of FRT is 4.915μs.
- Connect and set FRT-ch.0 to OCU-ch.1 (OCFS10 register write).
- Set OCU-ch.1 to Up/Down-count mode (Active High) operation. Also specify the initial output level of the output signal (RT1) (OCSA10, OCSB10 and OCSC register write).
- Set the timing of changing the output signal (RT1) for OCU-ch.1 (OCCP1 register write). In this example, "0x4800" is set. The value is written to the buffer register, and then transferred to the OCCP1 register.

Timing 2

- Instruct FRT-ch.0 to start count operation (TCSA0 register write).
- In Up/Down-count mode, FRT-ch.0 starts counting from "0x0000" and continues the Up-count operation until the TCCP value is reached (=0x5FFF), as shown in the figure. After that, switch over the count direction and perform the Down-count operation until the TCCP value (=0x5FFF) is reached. Then, continue counting.

Timing 3

- Instruct OCU-ch.0/ch.1 to enable the operation (OCSA10 register write).

Timing 4

- When OCU-ch.1 detects that the value of the FRT counter has reached "0x4800" and matched the setting value of OCCP1 during the Up-count operation, it changes the output signal (RT1) from the Low to High level.

Timing 5

- When OCU-ch.1 detects that the value of the FRT counter has reached "0x4800" and matched the setting value of OCCP1 during the Down-count operation, it changes the output signal (RT1) from the High to Low level.

Timing 6

- CPU sets the timing of changing the output signal (RT1) for OCU-ch.1 in the next FRT cycle (OCCP1 register write). As OCCP buffer function is valid and Zero value detection transfer mode is selected, the written value is first stored in the buffer register. Then, when the value of the FRT counter reaches the Zero value (Timing 7), the written value is transferred to the OCCP1 register and reflected upon OCU output. For this reason, even if writing takes place before Timing 5, as indicated by ★ in the figure, it does not affect the timing of changing the output signal (RT1).

Timing 7

- When the count value reaches "0x0000", FRT-ch.0 generates Zero value detection interrupt to CPU (Zero value detection interrupt is not generated at Timing 3).
- CPU determines that an interrupt has been generated from FRT-ch.0, because "1" is set to the Zero value detection flag of FRT-ch.0 (TCSA0 register read).
- CPU clears the Zero value detection flag and returns from the interrupt (TCSA0 register write).

After that, repeat Operations 4 to 7 so that PWM waveform with the peak value of the FRT counter symmetrical about the RT1 output can be achieved.

■ WFG's Operation

Timing 1

- Initialize WFG-ch.10 to RT-dead timer mode (Active High) (WFSA10 register write).
- Set the dead time to WFG-ch10 (WFTM register write). In this example, "0x0010" is set. When the WFG pre-scaler is set to 1/2 and PCLK to 40MHz, the dead time to be inserted is 0.8μs.
- In this mode of WFG, signals at the same level as RT1 and the opposite level are output to the output signals of WFG (RTO0 and RTO1) respectively.

Timing 4, 5

- When the RT1 signal changes from the Low to High level (or from the High to Low level), a specified dead time (transistor response time at the destination of the output) is inserted into the RTO0 and RTO1 signals, as shown in the figure, and the output level is changed.
- Non-overlap signal containing a dead time shown in the figure can be output to RTO0 and RTO1 by using the WFG function.

■ Operation of ADCMP/ATSA

Timing 1

- Set ADCMP-ch.0 to instruct ADC-unit0 to start AD conversion with the match condition for FRT's Up-count operation (ACSA and ACSB register write).
- Initialize ATSA to select ADC-unit0's conversion start signal from ADCMP as the scan conversion start signal (ATSA register write).
- Set the timing of starting AD conversion (ACCP0 register write). In this example, "0x2800" is set.

Timing 3

- Instruct ADCMP-ch.0 to enable its operation (ACSA register write).

Timing 8

- When ADCMP-ch.0 and ATSA detect that the value of the FRT counter has reached "0x2800" during the Up-count operation, they output ADC-unit0's scan conversion start signal.

■ Completion of Processing

Timing 9

The processing at Timing 9 indicates the procedure for completing the output of the PWM signal.

- Disable the operation of OCU- ch.1 (OCSA10 register write).
- Set the level of the output signals (RT0, RT1) for OCU-ch.1 (OCSB10 register write).
- Disable the operation of ADCMP-ch.0 (ACSA register write).
- When the output of OCU stops, WFG does not perform its operation.
- Instruct FRT-ch.0 to stop the count operation (TCSA0 register write).
- Set "0x0000" to FRT's count value (TCDT0 register write).

The above example explained the operation with 1 channel of OCU, 1 channel of WFG and 1 channel of ADCMP. If OCU-3ch, WFG-3ch and ADCMP-3ch are connected to the same FRT to perform interlocking control, three-phase motor control can be achieved.

■ Details of Register Settings

The procedure for register settings in Example of Operation – 2 is as follows.

The meanings of the symbols in the table are the same as those in Example of Operation – 1.

Table 3-5 Example of Operation 2 - Register Settings 1

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
1	FRT	TCSA0	HW	CLK[3:0]	0010	Clock division pre-scaler setting: 1/4
				SCLR	0	Soft clear: Do nothing
				MODE	1	Count mode setting: Up/Down-count mode
				STOP	1	FRT count operation: Stop counting
				BFE	1	TCCP buffer function: Enable
				ICRE	0	Peak value detection interrupt: Disable
				ICLR	0	Peak value detection: Clear
				Reserved	000	-
				IRQZE	1	Zero value detection interrupt: Enable
				IRQZF	0	Zero value detection: Clear
				ECKE	0	Selection of clock to be used: Internal clock
	TCCP0	HW	TCCP	0x5FFF	Set FRT cycle	
1	OCU	OCFS10	BW	FSO0[3:0]	Other	FRT connected to ch.0:
				FSO1[3:0]	0000	FRT connected to ch.1: FRT ch.0
		OCSA10	BW	CST0	Other	ch.0 operation state:
				CST1	0	ch.1 operation state: Operation disabled
				BDIS0	Other	ch.0 OCCP buffer function:
				BDIS1	0	ch.1 OCCP buffer function: Enable
				IOE0	Other	ch.0 interrupt:
				IOE1	0	ch.1 interrupt: Disable
				IOP0	Other	ch.0 match detection:
				IOP1	0	ch.1 match detection: Clear
		OCSB10	BW	OTD0	Other	RT0 output level:
				OTD1	0	RT1 output level: Low
				Reserved	00	-
				CMOD	0	ch.0/ch.1 operation mode: Up/Down (Active High)
				BTS0	Other	ch.0 buffer transfer:
				BTS1	0	ch.1 buffer transfer: Zero value detection transfer
				Reserved	0	-
		OCSC	BW	MOD0	Other	ch.0 operation mode:
				MOD1	1	ch.1 operation mode: Up/Down (Active High)
				MOD2	Other	ch.2 operation mode:
				MOD3	Other	ch.3 operation mode:
				MOD4	Other	ch.4 operation mode:
				MOD5	Other	ch.5 operation mode:
				Reserved	00	-
	OCCP1	HW	OCCP	0x4800	Specify ch.1 change timing	

Table 3-6 Example of Operation 2 - Register Settings 2

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
1	WFG	WFSA10	HW	DCK[2:0]	001	Clock division pre-scaler setting: 1/2
				TMD[2:0]	100	Operation mode: Select RT-dead timer mode
				GTEN[1:0]	00	Gate signal generation: don't care
				PSEL[1:0]	00	Connecting PPG: don't care
				PGEN[1:0]	00	PPG reflection: don't care
				DMOD	0	Output polarity: Active High
				Reserved	000	-
	WFTM10	HW	WFTM	0x0010		Set dead time value
	ADCMP	ACSA	HW	CE0[1:0]	00	ch.0 operation state: Operation disabled
				CE1[1:0]	Other	ch.1 operation state:
				CE2[1:0]	Other	ch.2 operation state:
				Reserved	00	-
				SEL0[1:0]	01	Selection of ch.0 start timing: At Up-count only
				SEL1[1:0]	Other	Selection of ch.1 start timing:
				SEL2[1:0]	Other	Selection of ch.2 start timing:
				Reserved	00	-
		ACSB	BW	BDIS0	1	ch.0 buffer function: Disable
				BDIS1	Other	ch.1 buffer function:
				BDIS2	Other	ch.2 buffer function:
				Reserved	0	-
				BTS0	0	ch.0 buffer transfer: don't care
				BTS1	Other	ch.1 buffer transfer:
				BTS2	Other	ch.2 buffer transfer:
				Reserved	0	-
	ACCP0	HW	ACCP	0x2800		Specify start timing for ADC0
	ATSA	ATSA	HW	AD0S[1:0]	00	ADC0 scan conversion start: ADCMPch.0
				AD1S[1:0]	Other	ADC1 scan conversion function:
				AD2S[1:0]	Other	ADC2 scan conversion function:
				Reserved	00	-
				AD0P[1:0]	00	ADC0 priority conversion start: ADCMPch.0
				AD1P[1:0]	Other	ADC1 priority conversion start:
				AD2P[1:0]	Other	ADC2 priority conversion start:
				Reserved	00	-

Table 3-7 Example of Operation 2 – Register Settings 3

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
2	FRT	TCSA0	HW	CLK[3:0]	NM	Clock division pre-scaler setting:
				SCLR	NM	Soft clear:
				MODE	NM	Count mode setting:
				STOP	0	FRT count operation: Start counting
				BFE	NM	TCCP buffer function:
				ICRE	NM	Peak value detection interrupt:
				ICLR	1(RMW)	Peak value detection: Do nothing
				Reserved	NM	-
				IRQZE	NM	Zero value detection interrupt:
				IRQZF	1(RMW)	Zero value detection: Do nothing
				ECKE	NM	Selection of clock to be used: Internal clock
3	OCU	OCSA10	BW	CST0	NM	ch.0 operation state:
				CST1	1	ch.1 operation state: Operation enabled
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Do nothing
				IOP1	1	ch.1 match detection: Do nothing
6	ADCMP	ACSA	HW	CE0[1:0]	01	ch.0 operation state: Operation enabled Connecting FRT: FRT ch.0
				CE1[1:0]	NM	ch.1 operation state:
				CE2[1:0]	NM	ch.2 operation state:
				Reserved	NM	-
				SEL0[1:0]	NM	Selection of ch.0 start timing:
				SEL1[1:0]	NM	Selection of ch.1 start timing:
				SEL2[1:0]	NM	Selection of ch.2 start timing:
				Reserved	NM	-
6	OCU	OCCP1	HW	OCCP1	0x4000	Specify ch.1 change timing

Table 3-8 Example of Operation 2 – Register Settings 4

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
7	FRT	TCSA0	HR	CLK[3:0]	DC	Clock division pre-scaler setting:
				SCLR	DC	Soft clear:
				MODE	DC	Count mode setting:
				STOP	DC	FRT count operation:
				BFE	DC	TCCP buffer function:
				ICRE	DC	Peak value detection interrupt:
				ICLR	DC	Peak value detection:
				Reserved	DC	-
				IRQZE	DC	Zero value detection interrupt:
				IRQZF	1	Zero value detection: Zero value detected
				ECKE	DC	Selection of clock used:
		TCSA0	HW	CLK[3:0]	NM	Clock division pre-scaler setting:
				SCLR	NM	Soft clear:
				MODE	NM	Count mode setting:
				STOP	NM	FRT count operation:
				BFE	NM	TCCP buffer function:
				ICRE	NM	Peak value detection interrupt:
				ICLR	1(RMW)	Peak value detection: Do nothing
				Reserved	NM	-
				IRQZE	NM	Zero value detection interrupt:
				IRQZF	0	Zero value detection: Flag cleared
				ECKE	NM	Selection of clock used:

Table 3-9 Example of Operation 2 – Register Settings 5

Setting Timing	Name of Target Block	Name of Register	Operation	Bit Field	Value	Description of Setting
9	OCU	OCSA10	BW	CST0	NM	ch.0 operation state:
				CST1	0	ch.1 operation state: Disable
				BDIS0	NM	ch.0 OCCP buffer function:
				BDIS1	NM	ch.1 OCCP buffer function:
				IOE0	NM	ch.0 interrupt:
				IOE1	NM	ch.1 interrupt:
				IOP0	1	ch.0 match detection: Do nothing
				IOP1	1	ch.1 match detection: Do nothing
	ADCMP	OCSB10	BW	OTD0	NM	RT0 output level:
				OTD1	0	RT1 output level: Low
				Reserved	NM	-
				CMOD	NM	ch.0/ch.1 operation mode:
				BTS0	NM	ch.0 buffer transfer:
				BTS1	NM	ch.1 buffer transfer:
				Reserved	NM	-
				CE0[1:0]	00	ch.0 operation state: Operation disabled
FRT	ACSA	TCSA0	HW	CE1[1:0]	NM	ch.1 operation state:
				CE2[1:0]	NM	ch.2 operation state:
				Reserved	NM	-
				SEL0[1:0]	NM	Selection of ch.0 start timing:
				SEL1[1:0]	NM	Selection of ch.1 start timing:
				SEL2[1:0]	NM	Selection of ch.2 start timing:
				Reserved	NM	-
				CLK[3:0]	NM	Clock division pre-scaler setting:
				SCLR	1	Soft clear: Initialize FRT
				MODE	NM	Count mode setting:
TCDT0	HW	TCDT	HW	STOP	1	FRT count operation: Stop counting
				BFE	NM	TCCP buffer function:
				ICRE	NM	Peak value detection interrupt:
				ICLR	1	Peak value detection: Do nothing
				Reserved	NM	-
				IRQZE	NM	Zero value detection interrupt:
				IRQZF	1	Zero value detection: Do nothing
				ECKE	NM	Selection of clock used:
				TCDT0	HW	Initialize FRT count value

4. Registers of Multifunction Timer

This chapter describes the registers of the multifunction timer.

- 4.1 Individual Notation and Common Notation of Channel Numbers in Descriptions of Functions
- 4.2 List of Registers of Multifunction Timer
- 4.3 Details of Register Functions
- 4.4 Details of OCU Output Waveform
- 4.5 Details of WFG Output Waveform

4.1. Individual Notation and Common Notation of Channel Numbers in Descriptions of Functions

This section explains the individual notation and common notation of channel numbers in descriptions of the functions in this chapter.

As the multifunction timer unit contains multiple blocks of the same function and consists of multiple channel circuits, there are some common matters across the channels.

Where there is no need to distinguish among the channels, and functions that are common to all the channels are to be explained, a notation without channel numbers and a notation with parentheses (common notation) are used to avoid repeated explanations and simplify their explanation.

Where there is a need to make distinctions in explaining operation among channels, I/O signals or control registers, a notation clearly stating channel numbers (individual notation) is used in such explanation.

The notation rules and examples are provided below.

- Where channel numbers are notated directly, that indicates individual notation.
This notation indicates that the operation, I/O signal or control register of the corresponding channel is explained.
- Some control registers control 2 channels at the same time. In such cases, the two corresponding channel numbers are stated in individual notation to distinguish between them.
- Where channel numbers are omitted from a notation, that indicates the common notation.
This notation indicates that the operation, I/O signal or control register which is common to all the channels is explained to omit the repetition of such explanation.
- Where channel numbers are stated with a figure in parentheses, that indicates the common notation for some channels.
Where there is a need for distinguishing between even-numbered channels and odd-numbered channels among the channels mounted, (0) and (1) are stated respectively.
In this case, (0) indicates that a function that is common to the even-numbered channels is explained, while (1) indicates that a function that is common to the odd-numbered channels is explained.

Example 1: ICU-ch.3 of MFT unit 0 can select the calibration input of the internal CR oscillator.

Example 1 is an example of the individual notation, which indicates that the calibration input of the internal CR oscillator can be selected by only ICU-ch.3 of MFT unit 0. This notation indicates that the calibration input of the internal CR oscillator cannot be selected by ICU-ch.0 to ch.2 of MFT unit 0 or ICU ch.0 to ch.3 of other MFT units.

Example 2: The ICFS10 register is a register that selects FRT to be connected to ICU-ch.1 and ICU-ch.0.

Example 3: The ICFS32 register is a register that selects FRT to be connected to ICU-ch.3 and ICU-ch.2.

Examples 2 and 3 are examples of the individual notation that states a control register (ICFS) with two channel numbers (10 and 32).

Example 4: The ICFS register is a register that selects FRT to be connected to ICU.

Example 4 is an example of the common notation that omits the channel numbers of the control register (ICFS). What the description explains means that similar to Examples 2 and 3, repeated explanations are omitted by the common notation.

Example 5: ICFS10.FSI0[3:0] is a register that selects FRT to be connected to ICU-ch.0.

Example 6: ICFS10.FSI1[3:0] is a register that selects FRT to be connected to ICU-ch.1.

Example 7: ICFS32.FSI0[3:0] is a register that selects FRT to be connected to ICU-ch.2.

Example 8: ICFS32.FSI1[3:0] is a register that selects FRT to be connected to ICU-ch.3.

Examples 5 to 8 are examples of the individual notation that clearly identifies the correspondence between the control bit and the channel in the control registers by stating two channel numbers in the control register (ICFS).

Example 9: ICFS.FSI0[3:0] is a register that selects FRT to be connected to ICU-ch.(0).

Example 10: ICFS.FSI1[3:0] is a register that selects FRT to be connected to ICU-ch.(1).

Examples 9 and 10 are examples of the common notation with parentheses that omits the channel numbers of the control register. What the description explains means that similar to Examples 5 to 8, repeated explanations are omitted by the common notation.

It should be noted that where the common notation is used in explanation of each function block, as shown above, it must be converted into the individual notation for the relevant channel when it is read.

Table 4-1 and Table 4-3 show the correspondence table between the individual notation and common notation. For the correspondence between the individual notation and common notation regarding register names, see the list of registers.

Table 4-1 Individual Notation and Common Notation of OCU

Channel Number		5	4	3	2	1	0
Notation for explaining OCU operation	Individual	ch.5	ch.4	ch.3	ch.2	ch.1	ch.0
	Common	ch.(1)	ch.(0)	ch.(1)	ch.(0)	ch.(1)	ch.(0)
Notation of names of signals output from OCU	Individual	RT5	RT4	RT3	RT2	RT1	RT0
	Common	RT(1)	RT(0)	RT(1)	RT(0)	RT(1)	RT(0)

Table 4-2 Individual Notation and Common Notation of WFG

Channel Number		54		32		10	
Notation for explaining WFG operation	Individual	ch.54		ch.32		ch.10	
	Common			No notation			
Names of signals input from OCU	Individual	RT5	RT4	RT3	RT2	RT1	RT0
	Common	RT(1)	RT(0)	RT(1)	RT(0)	RT(1)	RT(0)
Names of signals output from WFG	Individual	RTO5	RTO4	RTO3	RTO2	RTO1	RTO0
	Common	RTO(1)	RTO(0)	RTO(1)	RTO(0)	RTO(1)	RTO(0)
Names of PPG input signals after selected to be input from PPG	Individual	CH10_PPG		CH32_PPG		CH54_PPG	
	Common			CH_PPG			
Names of GATE signals before selected to be output to PPG	Signal Name	CH10_GATE		CH32_GATE		CH54_GATE	
	Common			CH_GATE			

Table 4-3 Individual Notation and Common Notation of ICU

Channel Number		3	2	1	0
Notation for explaining ICU operation	Individual	ch.3	ch.2	ch.1	ch.0
	Common	ch.(1)	ch.(0)	ch.(1)	ch.(0)
Notation of names of ICU input signals	Individual	IC3	IC2	IC1	IC0
	Common	IC(1)	IC(0)	IC(1)	IC(0)

4.2. List of Registers of Multifunction Timer

This section provides a list of the registers that exist in the multifunction timer unit.

Table 4-4 shows a list of the registers that exist in the multifunction timer unit.

The control registers of the multifunction timer unit are in the same configuration across the mounted channels. In this section, the operation of registers of the same function is explained using the common notation. The List of Registers states names in the individual notation and the common notation for each register. Replace the name in the common notation that appears in descriptions with the name in the individual notation when reading the descriptions.

Registers shown in the List of Registers refer to the registers that exist in the Multifunction Timer 1 unit. A model containing more than one multifunction timer unit has sets of the same registers for the number of the multifunction timer units.

Table 4-4 List of Registers of Multifunction Timer Unit

Block Name	Register Name (Individual Notation)	Register Function	Bit Width	Access	See	Register Name (Common Notation)
FRT	TCSA0	FRT ch.0 control register A	16	B, H	4.3.1	TCSA
	TCSA1	FRT ch.1 control register A				
	TCSA2	FRT ch.2 control register A				
	TCSB0	FRT ch.0 control register B	16	B, H	4.3.2	TCSB
	TCSB1	FRT ch.1 control register B				
	TCSB2	FRT ch.2 control register B				
	TCCP0	FRT ch.0 cycle setting register	16	H	4.3.3	TCCP
	TCCP1	FRT ch.1 cycle setting register				
	TCCP2	FRT ch.2 cycle setting register				
	TCDT0	FRT ch.0 count value register	16	H	4.3.4	TCDT
	TCDT1	FRT ch.1 count value register				
	TCDT2	FRT ch.2 count value register				
OCU	OCFS10	OCU ch.1, ch.0 connecting FRT select register	8	B, H	4.3.5	OCFS
	OCFS32	OCU ch.3, ch.2 connecting FRT select register				
	OCFS54	OCU ch.5, ch.4 connecting FRT select register		B		
	OCSA10	OCU ch.1, ch.0 control register A	8	B, H	4.3.6	OCSA
	OCSA32	OCU ch.3, ch.2 control register A				
	OCSA54	OCU ch.5, ch.4 control register A				
	OCSB10	OCU ch.1, ch.0 control register B	8	B, H	4.3.7	OCSB
	OCSB32	OCU ch.3, ch.2 control register B				
	OCSB54	OCU ch.5, ch.4 control register B				
	OCSC	OCU ch.5 ~ ch.0 control register C	8	B	4.3.8	OCSC
	OCCP0	OCU ch.0 compare value store register	16	H	4.3.9	OCCP(0)
	OCCP1	OCU ch.1 compare value store register				OCCP(1)
	OCCP2	OCU ch.2 compare value store register				OCCP(0)
	OCCP3	OCU ch.3 compare value store register				OCCP(1)
	OCCP4	OCU ch.4 compare value store register				OCCP(0)
	OCCP5	OCU ch.5 compare value store register				OCCP(1)

Block Name	Register Name (Individual Notation)	Register Function	Bit Width	Access	See	Register Name (Common Notation)
WFG	WFSA10	WFG ch.10 control register A	16	H	4.3.10	WFSA
	WFSA32	WFG ch.32 control register A				
	WFSA54	WFG ch.54 control register A				
	WFTM10	WFG ch.10 timer value register	16	H	4.3.11	WFTM
	WFTM32	WFG ch.32 timer value register				
	WFTM54	WFG ch.54 timer value register				
NZCL	NZCL	NZCL control register	16	H	4.3.12	NZCL
	WFIR	Interrupt control register	16	H	4.3.13	WFIR
ICU	ICFS10	ICU ch.1, ch.0 connecting FRT select register	8	B, H	4.3.14	ICFS
	ICFS32	ICU ch.3, ch.2 connecting FRT select register				
	ICSA10	ICU ch.1, ch.0 control register A	8	B, H	4.3.15	ICSA
	ICSA32	ICU ch.3, ch.2 control register A				
	ICSB10	ICU ch.1, ch.0 control register B	8	B, H	4.3.16	ICSB
	ICSB32	ICU ch.3, ch.2 control register B				
	ICCP0	ICU ch.0 capture value store register	16	H	4.3.17	ICCP
	ICCP1	ICU ch.1 capture value store register				
	ICCP2	ICU ch.2 capture value store register				
	ICCP3	ICU ch.3 capture value store register				
ADCMP	ACSA	ADCMP ch.2 to ch.0 control register A	16	B, H	4.3.18	ACSA
	ACSB	ADCMP ch.2 to ch.0 control register B	8	B	4.3.19	ACSB
	ACCP0	ADCMP ch.0 compare value store	16	H	4.3.20	ACCP
	ACCP1	ADCMP ch.1 compare value store				
	ACCP2	ADCMP ch.2 compare value store				
	ACCPDN0	ADCMP ch.0 compare value store	16	H	4.3.21	ACCPDN
	ACCPDN1	ADCMP ch.1 compare value store				
	ACCPDN2	ADCMP ch.2 compare value store				
ATSA	ATSA	ADC start trigger select register	16	H	4.3.22	ATSA

4.3. Details of Register Functions

This section explains details of the registers that exist in the multifunction timer unit.

- 4.3.1 FRT Control Register A (TCSA)
- 4.3.2 FRT Control Register B (TCSB)
- 4.3.3 FRT Cycle Setting Register (TCCP)
- 4.3.4 FRT Count Value Register (TCDT)
- 4.3.5 OCU Connecting FRT Select Register (OCFS)
- 4.3.6 OCU Control Register A (OCSA)
- 4.3.7 OCU Control Register B (OCSB)
- 4.3.8 OCU Control Register C (OCSC)
- 4.3.9 OCU Compare Value Store Register (OCCP)
- 4.3.10 WFG Control Register A (WFSA)
- 4.3.11 WFG Timer Value Register (WFTM)
- 4.3.12 NZCL Control Register (NZCL)
- 4.3.13 WFG Interrupt Control Register (WFIR)
- 4.3.14 ICU Connecting FRT Select Register (ICFS)
- 4.3.15 ICU Control Register A (ICSA)
- 4.3.16 ICU Control Register B (ICSB)
- 4.3.17 ICU Capture value store register (ICCP)
- 4.3.18 ADCMP Control Register A (ACSA)
- 4.3.19 ADCMP Control Register B (ACSB)
- 4.3.20 ADCMP Compare Value Store Register (ACCP)
- 4.3.21 ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN)
- 4.3.22 ADC Start Trigger Select Register (ATSA)

4.3.1. FRT Control Register A (TCSA)

TCSA is a 16-bit register that controls FRT.

Each mounted channel has three registers: TCSA0, TCSA1 and TCSA2.

TCSA0 controls FRT-ch.0.

TCSA1 controls FRT-ch.1.

TCSA2 controls FRT-ch.2.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	ECKE	IRQZF	IRQZE		Reserved		ICLR	ICRE
Attribute	R/W	R/W	R/W		-		R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Field	BFE	STOP	MODE	SCLR		CLK[3:0]		
Attribute	R/W	R/W	R/W	W		R/W		
Initial value	0	1	0	0	0	0	0	0

■ Functions of Register

[bit3:0] TCSA.CLK[3:0]

Process	Value	Function
Write	0000	Sets FRT's count clock cycle to the same value as PCLK.
	0001	Sets FRT's count clock cycle to PCLK multiplied by 2.
	0010	Sets FRT's count clock cycle to PCLK multiplied by 4.
	0011	Sets FRT's count clock cycle to PCLK multiplied by 8.
	0100	Sets FRT's count clock cycle to PCLK multiplied by 16.
	0101	Sets FRT's count clock cycle to PCLK multiplied by 32.
	0110	Sets FRT's count clock cycle to PCLK multiplied by 64.
	0111	Sets FRT's count clock cycle to PCLK multiplied by 128.
	1000	Sets FRT's count clock cycle to PCLK multiplied by 256.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

TCSA.CLK[3:0] is a register that sets the count clock cycle of FRT counter (16-bit Up/Down counter).

Change the setting of this register while FRT is stopping.

As for FRT count clock, either the PCLK in LSI which is divided by the pre-scaler or an external clock input can be selected for use. As this register setting is the setting for the pre-scaler, its value has no meaning if an external clock input is selected.

FRT's count clock cycle is determined based on the PCLK cycle and the clock division ratio set by this register.

The following table shows examples of CLK[3:0] settings and FRT count clock cycles.

CLK[3:0]	Clock Ratio	FRT Count Clock Cycle		
		PCLK=25ns (40MHz)	PCLK=33.3ns (33MHz)	PCLK=50ns (25MHz)
0000	1	25ns	30ns	50ns
0001	2	50ns	61ns	100ns
0010	4	100ns	121ns	200ns
0011	8	200ns	242ns	400ns
0100	16	400ns	485ns	800ns
0101	32	800ns	970ns	1.6μs
0110	64	1.6μs	1.9μs	3.2μs
0111	128	3.2μs	3.9μs	6.4μs
1000	256	6.4μs	7.8μs	12.8μs

[bit4] TCSA.SCLR

Process	Value	Function
Write	0	Cancels FRT operation state initialization request.
	1	Issues FRT operation state initialization request.
Read	-	"0" is always read.

TCSA.SCLR is a register that requests FRT operation state initialization.

There are two ways to use this register, as described below.

1. When stopping FRT counter
 Write "1" to issue an initialization request for FRT's operation state when stopping FRT counter.
2. When clearing FRT counter through clock synchronization
 Write "1" to issue a request to clear FRT's count value to "0x0000" through synchronization when operating FRT in Up-count mode.

For information about how to use this register, see the section regarding TCSA.STOP.

[bit5] TCSA.MODE

Process	Value	Function
Write	0	Sets FRT's count mode to Up-count mode.
	1	Sets FRT's count mode to Up/Down-count mode.
Read	-	Read the register setting.

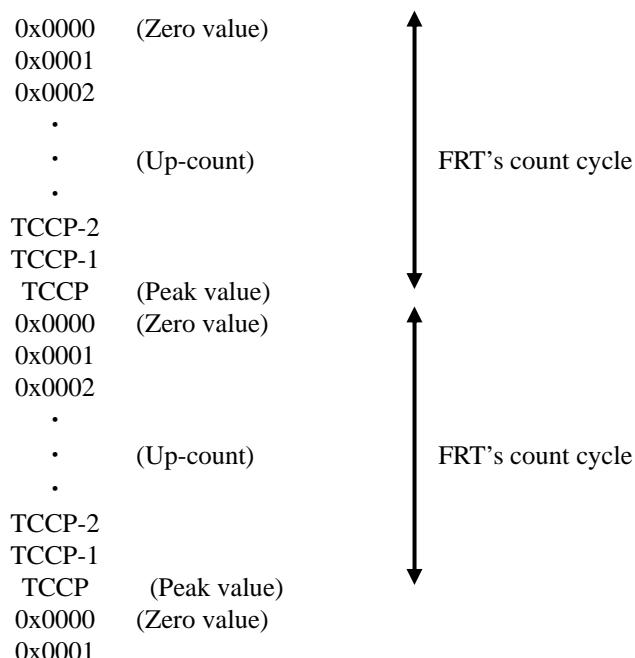
TCSA.MODE is a register that selects FRT's count mode.

Change the setting of this register while FRT is stopping.

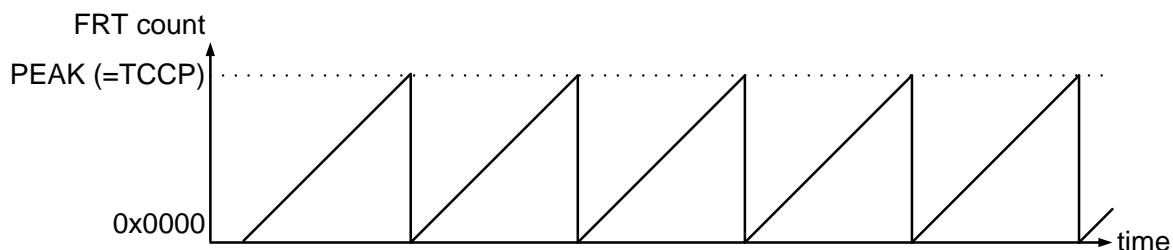
In Up-count mode, FRT performs the following operation.

FRT's counter starts Up-count operation from "0x0000". After up-counting to the value set by the TCCP register, the value of the counter becomes "0x0000". Then, the Up-count operation is repeated. FRT's count cycle is "(TCCP+1) x Count clock cycle".

Change in the value of FRT's counter is shown below.



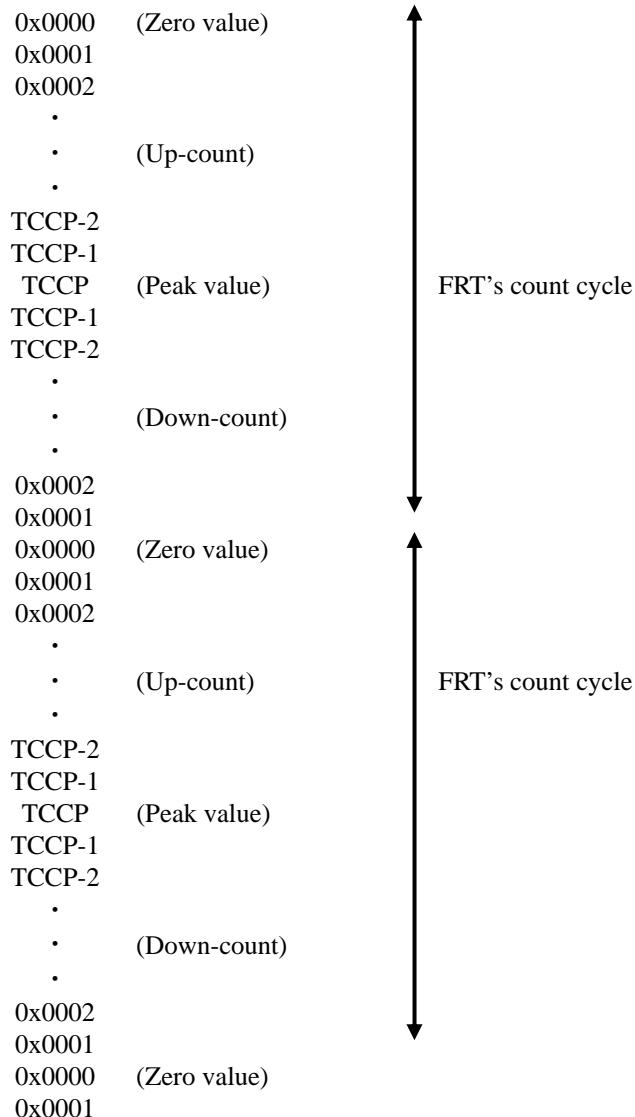
TCSA.MODE=0 Up count mode



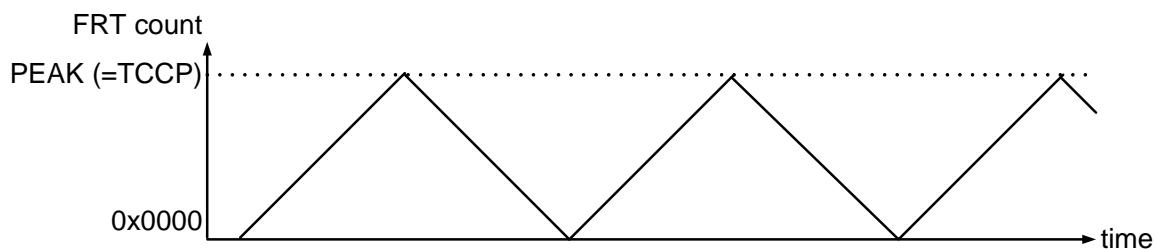
In Up/Down-count mode, FRT performs the following operation.

FRT's counter starts Up-count operation from "0x0000". After up-counting to the value set by the TCCP register, it starts Down-count operation. When it returns to "0x0000", it starts up-counting again and repeats the count operation. FRT's count cycle is "(TCCP) x 2 x Count clock cycle".

Change in the value of FRT's counter is shown below.



TCSA.MODE=1 Up-down count mode



[bit6] TCSA.STOP

Process	Value	Function
Write	0	Puts FRT in operating state.
	1	Puts FRT in stopping state.
Read	-	Reads the register setting

TCSA.STOP is a register that controls the start and stop of FRT's operation. This register is used in the combination with TCSA.SCLR, as shown below.

- When starting FRT's counter operation:

When "0" is written to TCSA.STOP and TCSA.SCLR while FRT's count operation is stopped, FRT starts counting.

- When clearing the count value of FRT's counter to "0x0000" through synchronization in Up-count mode:

If "0" is written to TCSA.STOP and "1" is written to TCSA.SCLR during FRT's count operation in Up-count mode, FRT's count value is cleared to "0x0000" in FRT's next count clock. If "0" is written to TCSA.SCLR before the counter is cleared, the counter clear request is cancelled and the counter value is not cleared. Do not write TCSA.SCLR=0 until it can be checked that the counter value is cleared. This operation cannot be performed in Up/Down-count mode.

- When stopping the operation of FRT's counter:

If "1" is written to TCSA.STOP and TCSA.SCLR during FRT's count operation, FRT stops the count operation. In some cases, FRT's counter value is not initialized to "0x0000" even after FRT stops, depending on the state of FRT's count clock. Always write "0x0000" to TCDT afterwards to clear FRT's counter value to "0x0000".

To rewrite to another register in the same address area during FRT's count operation, write "0" to TCSA.STOP and TCSA.SCLR.

To rewrite to another register in the same address area while FRT's count operation is stopped, write "1" to TCSA.STOP and "0" to TCSA.SCLR.

Figure 4-1 FRT Counter Start, Clear and Stop (Up-count Mode)

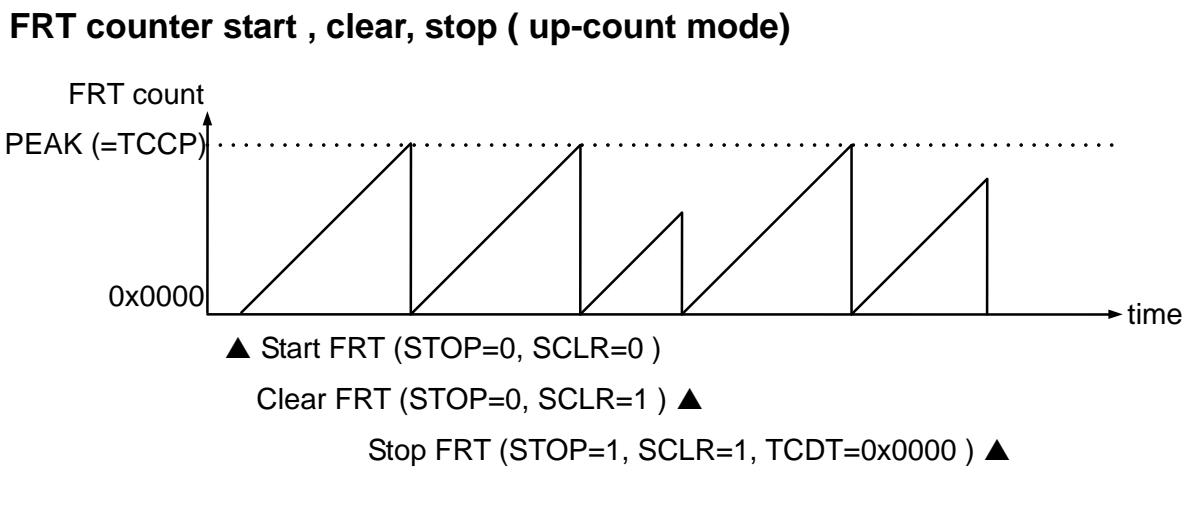
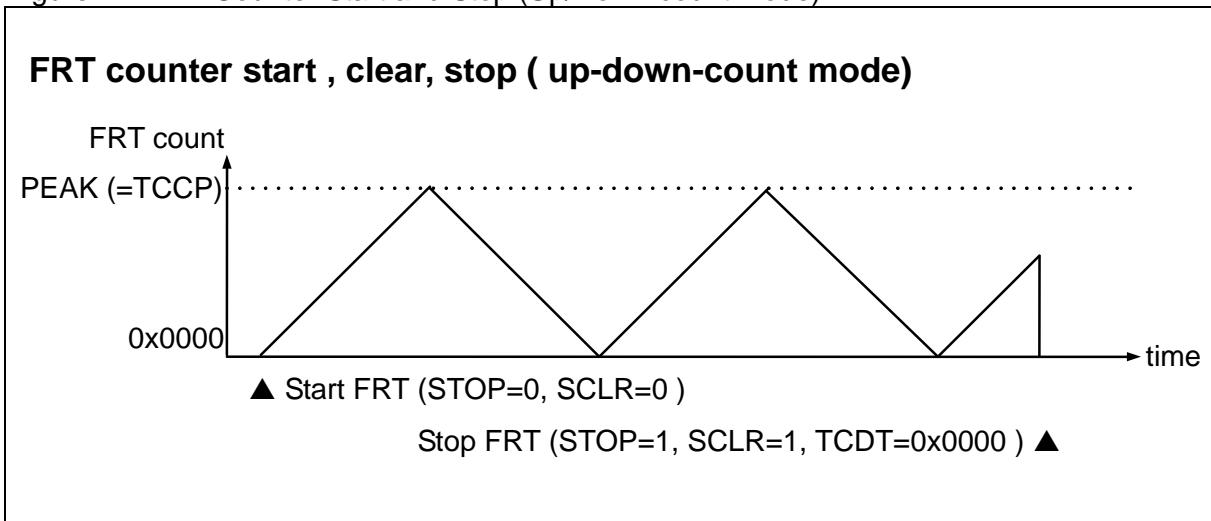


Figure 4-2 FRT Counter Start and Stop (Up/Down-count Mode)



[bit7] TCSA.BFE

Process	Value	Function
Write	0	Disables TCCP's buffer function.
	1	Enables TCCP's buffer function.
Read	-	Reads the register setting.

TCSA.BFE is a register that specifies whether to enable or disable the buffer function of the TCCP register. See "4.3.3 FRT Cycle Setting Register (TCCP)".

[bit8] TCSA.ICRE

Process	Value	Function
Write	0	Does not generate interrupt when "1" is set to TCSA.ICLR.
	1	Generates interrupt when "1" is set to TCSA.ICLR.
Read	-	Reads the register setting.

TCSA.ICRE is a register that specifies whether to notify CPU of the event that TCSA.ICLR is set as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit9] TCSA.ICLR

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that no match has been detected between FRT's count value and TCCP value.
	1	Indicates that a match has already been detected between FRT's count value and TCCP value.
Read at RMW access	"1"	is always read.

TCSA.ICLR is a register that is set to "1" when a match is detected between FRT's count value and TCCP value during FRT operation (hereinafter referred to as "Peak value detection").

By reading this register, whether FRT's count value has reached the TCCP value or not can be determined.

This register can be cleared by writing "0".

This register does nothing, if "1" is written. Always write "1" to the register when rewriting to another register in the same address area.

"1" is always read from this register at RMW access.

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit12:10] Reserved

Process	Function	
Write	"0" must be written at write access.	
Read	"0" is read.	

[bit13] TCSA.IRQZE

Process	Value	Function
Write	0	Does not generate interrupt when "1" is set to TCSA.IRQZF.
	1	Generates interrupt, when "1" is set to TCSA.IRQZF.
Read	-	Reads the register setting.

TCSA.IRQZE is a register that specifies whether to notify CPU of the event that TCSA.IRQZF is set as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit14] TCSA.IRQZF

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that a match between FRT's count value and "0x0000" has not been detected.
	1	Indicates that a match between FRT's count value and "0x0000" has already been detected.
Read at RMW access	"1"	is always read.

TCSA.IRQZF is a register that is set to "1" when a match between FRT's count value and "0x0000" has been detected (hereinafter referred to as "Zero value detection").

By reading this register, whether FRT's count value has reached "0x0000" or not can be determined.

This register is not set at "0x0000" from which FRT starts counting or at "0x0000" to which the counter value has been cleared by TCSA.SCLR.

This register can be cleared by writing "0".

This register does nothing, if "1" is written. Always write "1" to the register when rewriting to another register in the same address area.

"1" is always read from this register at RMW access.

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit15] TCSA.ECKE

Process	Value	Function
Write	0	Uses the internal clock (PCLK) as FRT's count clock.
	1	Uses an external input clock (FRCK) as FRT's count clock.
Read	-	Reads the register setting.

TCSA.ECKE is a register that selects the clock signal to be used as FRT's count clock.

Change the setting of this register while FRT is stopping.

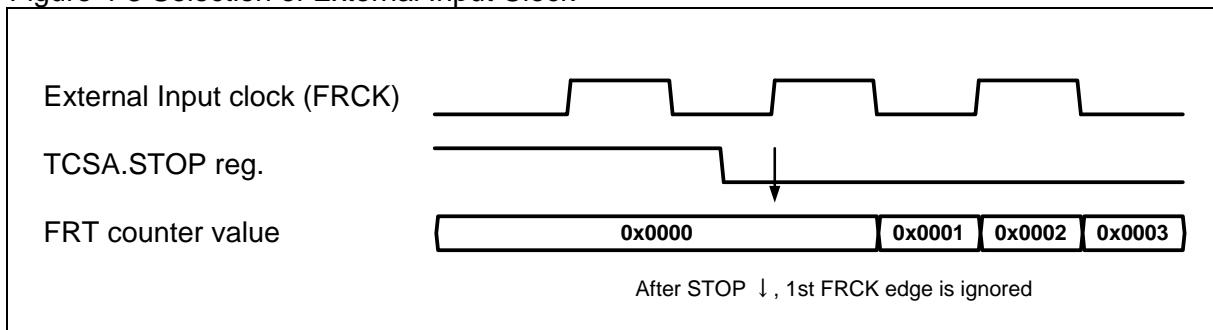
To select the internal clock, the clock division ratio must be set by TCSA.CLK[3:0].

To select an external input clock, the FRCK pin to be used in the GPIO block must be predetermined.

To operate it with an external input clock, the count operation is performed both at the rising edge and falling edge of an external input clock signal.

To operate it with an external input clock, the first edge from the external input clock after FRT operation starts (writing "0" to TCSA.STOP) is ignored, irrespective of the rising or falling edge, and the count operation starts from the next edge.

Figure 4-3 Selection of External Input Clock



4.3.2. FRT Control Register B (TCSB)

TCSB is a 16-bit register that controls FRT.

Each mounted channel has three registers: TCSB0, TCSB1 and TCSB2.

TCSB0 controls FRTch0.

TCSB1 controls FRTch1.

TCSB2 controls FRTch2.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute								
Initial Value	0	0	0	0	-	0	0	0
Bit	7	6	5	4	3	2	1	0
Field	Reserved					AD2E	AD1E	AD0E
Attribute						R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit0] TCSB.AD0E

Process	Value	Function
Write	0	Does not output AD conversion start signal to ADCunit0 upon Zero value detection by FRT.
	1	Outputs AD conversion start signal to ADCunit0 upon Zero value detection by FRT.
Read	-	Reads the register setting.

[bit1] TCSB.AD1E

Process	Value	Function
Write	0	Does not output AD conversion start signal to ADCunit1 upon Zero value detection by FRT.
	1	Outputs AD conversion start signal to ADCunit1 upon Zero value detection by FRT.
Read	-	Reads the register setting.

[bit2] TCSB.AD2E

Process	Value	Function
Write	0	Does not output AD conversion start signal to ADCunit2 upon Zero value detection by FRT.
	1	Outputs AD conversion start signal to ADCunit2 upon Zero value detection by FRT.
Read	-	Reads the register setting.

TCSB.AD0E, TCSB.AD1E and TCSB.AD2E are registers that select AD conversion start signal output upon Zero value detection by FRT.

These registers are used to start ADC conversion upon Zero value detection by FRT. Each of the AD conversion start signals for the 3 channels of FRT undergoes logic OR by ADC unit to which they are to be output. See the entire block diagram.

The conversion start signal from FRT ch.0, FRTch.1 and FRTch.2 to ADCunit0 has undergone logic OR.

The conversion start signal from FRT ch.0, FRTch.1 and FRTch.2 to ADCunit1 has undergone logic OR.

The conversion start signal from FRT ch.0, FRTch.1 and FRTch.2 to ADCunit2 has undergone logic OR.

Due to the above configuration, attention must be paid when starting AD conversion from multiple FRT's to the same ADC. To start AD conversion at FRT's count value other than Zero value detection, ADCMP can be used to output the AD conversion start signal.

It can be selected in the ATSA block whether to use the ADC conversion start signal achieved upon Zero value detection by FRT and the ADC conversion start signal achieved by ADCMP for starting ADC's scan conversion or priority conversion.

[bit15:3] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

4.3.3. FRT Cycle Setting Register (TCCP)

TCCP is a 16-bit register that sets FRT's count cycle.
 Each mounted channel has three registers: TCCP0, TCCP1 and TCCP2.
 TCCP0 sets the cycle for FRTch0.
 TCCP1 sets the cycle for FRTch1.
 TCCP2 sets the cycle for FRTch2.
 It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TCCP[15:0]															
Attribute	R/W															
Initial Value	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1															

■ Functions of Register

[bit15:0] TCCP.TCCP[15:0]

Process	Function
Write	Sets FRT's cycle. Stores the written value to the TCCP buffer register.
Read	Reads the value in the TCCP register (not the value in the TCCP buffer register).

TCCP is a 16-bit register that sets FRT's count cycle.

Depending on the TCCP value and FRT's count mode, FRT's count cycle varies, as shown below.

In Up-count mode:

FRT's count cycle = $(TCCP+1) \times FRT's\ count\ clock\ cycle$

In Up/Down-count mode:

FRT's count cycle = $TCCP \times 2 \times FRT's\ count\ clock\ cycle$

When data is written to this address area, the data is first stored in the buffer register. And then, the data is transferred from the buffer register to the TCCP register under the following conditions.

When the buffer function is disabled:

Data is transferred immediately after it is written to the buffer register.

When the buffer function is enabled:

Data is transferred, when FRT is stopped or when FRT's count value has reached "0x0000".

Whether the buffer function is enabled or disabled is determined by the value in the TCSA.BFE register.

FRT's count cycle can be changed by rewriting this register during FRT's count operation. If data is read from this address area, the value in the TCCP register is read, rather than the value in the buffer register. Therefore, it should be noted that no bit can be rewritten by RMW access to this address area when the buffer function is enabled.

It is prohibited to write "0x0000" to this register.

Figure 4-4 shows an example of changing FRT's cycle when the buffer function is enabled.

When TCCP's buffer function is enabled, a value written to the buffer register is transferred to the TCCP register upon the next Zero value detection. FRT's count cycle is changed in the next FRT cycle after the writing.

Figure 4-4 Example of Changing FRT's Cycle (When Buffer Function is Enabled)

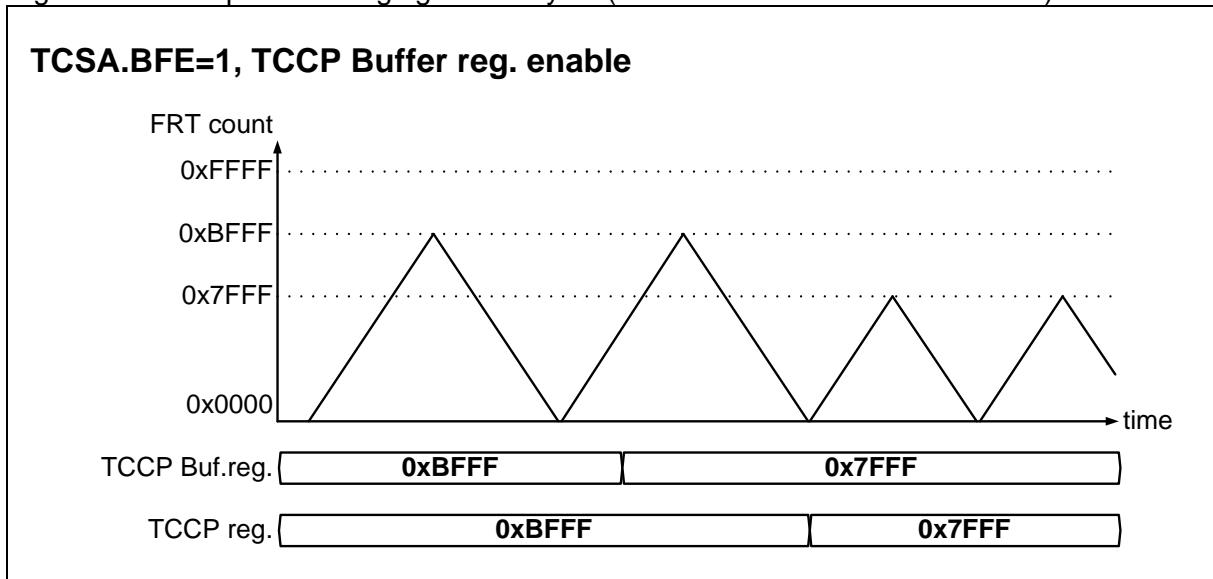
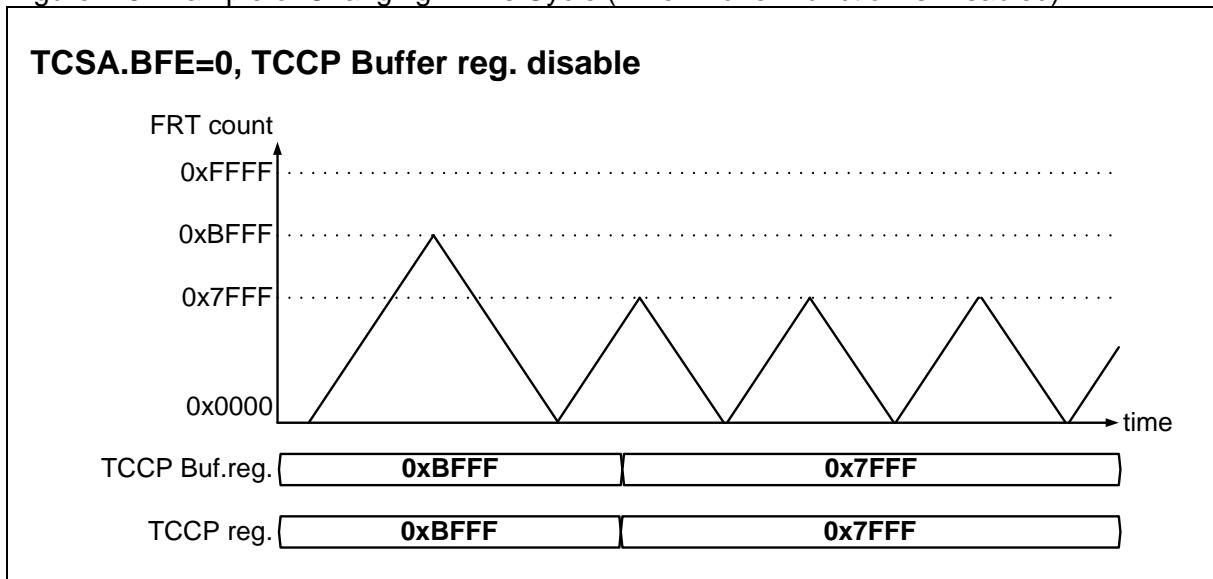


Figure 4-5 shows an example of changing FRT's cycle when the buffer function is disabled.

When TCCP's buffer function is disabled, the value in the buffer register is immediately reflected on the TCCP register; therefore, FRT's cycle can be changed in the same cycle in which the value was written. In this case, it should be noted that if a value smaller than FRT's count value is written as the TCCP value at this point, FRT's counter counts up to "0xFFFF".

Figure 4-5 Example of Changing FRT's Cycle (When Buffer Function is Disabled)



4.3.4. FRT Count Value Register (TCDT)

TCDT is a 16-bit register that reads and writes FRT's count value.
 Each mounted channel has three registers: TCDT0, TCDT1 and TCDT2.
 TCDT0 is the timer count value of FRT-ch.0.
 TCDT1 is the timer count value of FRT-ch.1.
 TCDT2 is the timer count value of FRT-ch.2.
 It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TCDT[15:0]															
Attribute	R/W															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Functions of Register

[bit15:0] TCDT.TCDT[15:0]

Process	Value	Function
Write	0x0000	Sets FRT's count value to "0x0000" (possible only when FRT is stopped).
	Other than above	Setting prohibited
Read	-	Reads FRT's current value.

TCDT is a 16-bit register that reads and writes FRT's count value.

The value read from TCDT is FRT's count value of that point.

Do not write any data during FRT's operation or a value other than "0x0000".

If FRT is operated, and then stopped, make sure to write "0x0000" to TCDT and initialize FRT's count value in order to prepare for its restart.

4.3.5. OCU Connecting FRT Select Register (OCFS)

OCFS is an 8-bit register that selects and sets FRT to be connected to OCU. Each mounted channel has three registers: OCFS10, OCFS32 and OCFS54. OCFS10 controls OCU ch1 and OCU ch0. OCFS32 controls OCU ch3 and OCU ch2. OCFS54 controls OCU ch5 and OCU ch4. OCFS10 and OCFS54 are located at even-numbered addresses, while OCFS32 is located at an odd-numbered address. Therefore, their bit positions are [7:0] and [15:8].

■ Configuration of Register

Bit Field	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
Attribute	FSO1[3:0]				FSO0[3:0]			
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit3:0/11:8] OCFS.FSO0[3:0]

Process	Value	Function
Write	0000	Connects FRT ch.0 to OCU ch.(0).
	0001	Connects FRT ch.1 to OCU ch.(0).
	0010	Connects FRT ch.2 to OCU ch.(0).
	0011 0100	For models with multiple MFT units: Connects FRT of an external MFT. For models with one MFT unit: Setting prohibited
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit7:4/15:12] OCFS.FSO1[3:0]

Process	Value	Function
Write	0000	Connects FRT ch.0 to OCU ch.(1).
	0001	Connects FRT ch.1 to OCU ch.(1).
	0010	Connects FRT ch.2 to OCU ch.(1).
	0011 0100	For models with multiple MFT units: Connects FRT of an external MFT. For models with one MFT unit: Setting prohibited
	Other than above	Setting prohibited
Read	-	Reads the register setting.

OCFS.FSO0[3:0] is a register that selects FRT to be connected to ch.(0) of OCU and uses it.

OCFS.FSO1[3:0] is a register that selects FRT to be connected to ch.(1) of OCU and uses it.

Change the setting of this register, while the operation of the OCU to be connected is prohibited.

For models with multiple MFT units, the connection to FRT that exists in another MFT unit can be selected. For related settings, see "5.1 Connection of Model Containing Multiple MFT's".

4.3.6. OCU Control Register A (OCSA)

OCSA is an 8-bit register that controls OCU's operation.

Each mounted channel has three registers: OCSA10, OCSA32 and OCSA54.

OCSA10 controls OCU ch1 and OCU ch0.

OCSA32 controls OCU ch3 and OCU ch2.

OCSA54 controls OCU ch5 and OCU ch4.

■ Configuration of Register

Bit	7	6	5	4	3	2	1	0
Field	IOP1	IOP0	IOE1	IOE0	BDIS1	BDIS0	CST1	CST0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	1	1	0	0

■ Functions of Register

[bit0] OCSA.CST0

Process	Value	Function
Write	0	Disables the operation of OCU ch.(0). Reflects the value written to OCSB.OTD0 on the RT(0) output pin.
	1	Enables the operation of OCU ch.(0). Ignores the value written to OCSB.OTD0.
Read	-	Reads the register setting.

[bit1] OCSA.CST1

Process	Value	Function
Write	0	Disables the operation of OCU ch.(1). Reflects the value written to OCSB.OTD1 on the RT(1) output pin.
	1	Enables the operation of OCU ch.(1). Ignores the value written to OCSB.OTD1.
Read	-	Reads the register setting.

OCSA.CST0 is a register that selects the operation state of OCU-ch(0).

OCSA.CST1 is a register that selects the operation state of OCU-ch(1).

Each channel of OCU, when the operation is enabled, performs the following operation according to the operation mode setting, at the timing when the value specified in the OCCP register matches FRT's count value.

- It changes the output level of the RT0 to RT5 output pins and outputs the PWM signal.
- It sets "1" to the OCSA.IOP0 and OCSA.IOP1 registers and notifies CPU of the state change.

If the values don't match and the operation is disabled, the output level of the output pins maintains the last state.

For OCU's operation modes, see "4.4 Details of OCU Output Waveform".

If OCU's operation is enabled, the writing to the OCSB.OTD0 and OCSB.OTD1 is ignored and not reflected on the level of the output pins.

<Note>

Always follow the procedure below and perform control when starting PWM signal output by OCU.

1. Initial setting
Set FRT operation mode (FRT control register other than TCSA.STOP).
Set OCU operation mode and initialize the output level (OCU control register other than OCSA.CST0 and OCSA.CST1).
Set the OCCP compare value (writing the OCCP value).
2. Start FRT count operation (writing "0" to TCSA.STOP).
3. Enable OCU's operation (writing "1" to OCSB.CST0 and OCSB.CST1).

Always follow the procedure below and perform control when finishing PWM signal output by OCU.

1. Disable OCU's operation (writing "0" to OCSB.CST0 and OCSB.CST1).
2. Reset the output level of the OCU output pins (writing to OCSB.OTD0 and OCSB.OTD1, if necessary).
3. Stop FRT's count operation (writing "1" to TCSA.STOP and TCSA.SCLR, writing 0x0000 to TCDT).

[bit2] OCSA.BDIS0

Process	Value	Function
Write	0	Enables the buffer function of the OCCP(0) register.
	1	Disables the buffer function of the OCCP(0) register.
Read	-	Reads the register setting.

[bit3] OCSA.BDIS1

Process	Value	Function
Write	0	Enables the buffer function of the OCCP(1) register.
	1	Disables the buffer function of the OCCP(1) register.
Read	-	Reads the register setting.

OCSA.BDIS0 is a register that specifies whether to enable or disable the buffer register function of OCCP(0).

OCSA.BDIS1 is a register that specifies whether to enable or disable the buffer register function of OCCP(1).

Change the setting of these registers, while OCU's operation is disabled.

See "4.3.9 OCU Compare Value Store Register (OCCP)".

<Note>

When using FRT in Up/Down-count mode, make sure to enable the buffer function of OCCP and use it as the transfer mode at Zero value detection.

[bit4] OCSA.IOE0

Process	Value	Function
Write	0	Does not generate interrupt, when "1" is set to OCSA.IOP0.
	1	Generates interrupt, when "1" is set to OCSA.IOP0.
Read	-	Reads the register setting.

[bit5] OCSA.IOE1

Process	Value	Function
Write	0	Does not generate interrupt, when "1" is set to OCSA.IOP1.
	1	Generates interrupt, when "1" is set to OCSA.IOP1.
Read	-	Reads the register setting.

OCSA.IOE0 is a register that specifies whether to notify CPU of the event that "1" is set to OCSA.IOP0 as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

OCSA.IOE1 is a register that specifies whether to notify CPU of the event that "1" is set to OCSA.IOP1 as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit6] OCSA.IOP0

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that no match has been detected between FRT's count value and OCCP(0) value at OCU ch.(0).
	1	Indicates that a match has already been detected between FRT's count value and OCCP(0) value at OCU ch.(0).
Read at RMW access	"1"	is always read.

[bit6] OCSA.IOP1

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that no match has been detected between FRT's count value and OCCP(1) value at OCU ch.(1).
	1	Indicates that a match has already been detected between FRT's count value and OCCP(1) value at OCU ch.(1).
Read at RMW access	"1"	is always read.

OCSA.IOP0 is a register that is set to "1" when a match is detected between FRT's count value and OCCP(0) value when the operation of OCU-ch.(0) is enabled.

OCSA.IOP1 is a register that is set to "1" when a match is detected between FRT's count value and OCCP(1) value when the operation of OCU-ch.(1) is enabled.

By reading from this register, whether FRT's count value has reached the OCCP value or not can be determined.

This register can be cleared by writing "0".

This register does nothing, if "1" is written. Always write "1" to the register when rewriting to another register in the same address area.

"1" is always read from this register at RMW access.

See "5.2 Treatment of Event Detect Register and Interrupt".

<Note>

When FRT is in Up/Down-count mode, these registers are not set, even if FRT's count value has matched the OCCP value at its peak.

4.3.7. OCU Control Register B (OCSB)

OCSB is an 8-bit register that controls OCU's operation.

Each mounted channel has three registers: OCSB10, OCSB32 and OCSB54.

OCSB10 controls OCU ch1 and OCU ch0.

OCSB32 controls OCU ch3 and OCU ch2.

OCSB54 controls OCU ch5 and OCU ch4.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved	BTS1	BTS0	CMOD	Reserved	Reserved	OTD1	OTD0
Attribute	-	R/W	R/W	R/W	-	-	R/W	R/W
Initial Value	-	1	1	0	-	-	0	0

■ Functions of Register

[bit8] OCSB.OTD0

Process	Value	Function
Write	0	Sets the output level of the RT(0) pin to the Low level, when OCSA.CST0=0. Does nothing, when OCSA.CST0=1.
	1	Sets the output level of the RT(0) pin to the High level, when OCSA.CST0=0. Does nothing, when OCSA.CST0=1.
Read	0	Indicates that the RT(0) output pin is in the Low-level output state.
	1	Indicates that the RT(0) output pin is in the High-level output state.

[bit9] OCSB.OTD1

Process	Value	Function
Write	0	Sets the output level of the RT(1) pin to the Low level, when OCSA.CST1=0. Does nothing, when OCSA.CST1=1.
	1	Sets the output level of the RT(1) pin to the High level, when OCSA.CST1=0. Does nothing, when OCSA.CST1=1.
Read	0	Indicates that the RT(1) output pin is in the Low-level output state.
	1	Indicates that the RT(1) output pin is in the High-level output state.

OCSA.OTD0 is a register that reads the state of the RT(0) output pin of OCU-ch.(0) and sets its output level.

OCSA.OTD1 is a register that reads the state of the RT(1) output pin of OCU-ch.(1) and sets its output level.

The output level of the OCU output pins (RT0 to RT5) can be set by writing to these registers when OCU's operation is disabled. When OCU's operation is enabled, the writing to these registers is ignored. The read value of these registers indicates the output level of the OCU output pins, irrespective of OCU's operation state.

<Notes>

- After being processed by WFG, OCU's output pins (RT0 to RT5) become LSI's external output pins (RTO0 to RTO5). For this reason, the level of OCU's output pins does not match the level of LSI's external output pins in some of WFG's operation modes; therefore care must be taken. The state of LSI's external output pins can be read from the PDIR register of the I/O port block.
- Follow the procedure below to set the output level to Low by stopping OCU's operation when CST0=1 (OCU operation enabled) and OTD0=1 (High-level output).
 - No value can be written to OTD0 while OCU's operation is enabled; therefore, first write "0" to CST0 to stop OCU's operation.
 - Then, write "0" to OTD0 to set the output level to Low.

It should be noted that if the above steps were reversed, the value written to OTD0 would be ignored. It should also be noted that if CST0=0 and OTD0=0 were written to the OCSA and OCSB registers by half-word access, the value written to OTD0 would be ignored because OCU's operation is enabled. Similarly, care must be taken to writing to OTD1.

[bit11:10] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

[bit12] OCSB.CMOD

Process	Value	Function
Write	0	Writes "0" to this register.
	1	Writes "1" to this register.
Read	-	Reads the register setting.

OCSB.CMOD is a register that selects OCU's operation mode in combination with OCSC.MOD0 to MOD5.

Change the setting of this register, while OCU's operation is disabled.

For details of operation modes by this register setting, see "4.4 Details of OCU Output Waveform".

When setting OCSB10.CMOD, the common setting applies to ch.1 and ch.0.

When setting OCSB32.CMOD, the common setting applies to ch.3 and ch.2.

When setting OCSB54.CMOD, the common setting applies to ch.5 and ch.4.

[bit13] OCSB.BTS0

Process	Value	Function
Write	0	Performs buffer transfer of the OCCP(0) register upon Zero value detection by FRT.
	1	Performs buffer transfer of the OCCP(0) register upon Peak value detection by FRT.
Read	-	Reads the register setting.

[bit14] OCSB.BTS1

Process	Value	Function
Write	0	Performs buffer transfer of the OCCP(1) register upon Zero value detection by FRT.
	1	Performs buffer transfer of the OCCP(1) register upon Peak value detection by FRT.
Read	-	Reads the register setting.

OCSB.BTS0 is a register that specifies the timing of transfer from the buffer register to the OCCP(0) register when the buffer function of the OCCP(0) register is enabled.

OCSB.BTS1 is a register that specifies the timing of transfer from the buffer register to the OCCP(1) register when the buffer function of the OCCP(1) register is enabled.

Change the setting of these registers while OCU's operation is disabled.

The setting of these registers has no meaning, when the buffer function is disabled (OCSA.BDIS1=1, OCSA.BDIS0=1).

See "4.3.9 OCU Compare Value Store Register (OCCP)".

<Note>

When using FRT in Up/Down-count mode, make sure to enable OCCP's buffer function and select the transfer mode for Zero value detection.

[bit15] Reserved

Process	Function
Write	The written value is ignored.
Read	An undefined value is read.

4.3.8. OCU Control Register C (OCSC)

OCSC is an 8-bit register that controls OCU's operation.
This register controls all of OCU ch0 to ch.5.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	-	-	0	0	0	0	0	0

■ Functions of Register

[bit5:0] OCSC.MOD5, OCSC.MOD4, OCSC.MOD3, OCSC.MOD2, OCSC.MOD1, OCSC.MOD0

Process	Value	Function
Write	0	Writes "0" to this register.
	1	Writes "1" to this register.
Read	-	Reads the register setting.

OCSC.MOD0 and OCSC.MOD1 determines the operation mode of OCU ch.0/ch.1 in combination with OCSB10.CMOD.

OCSC.MOD2 and OCSC.MOD3 determines the operation mode of OCU ch.2/ch.3 in combination with OCSB32.CMOD.

OCSC.MOD4 and OCSC.MOD5 determines the operation mode of OCU ch.4/ch.5 in combination with OCSB54.CMOD.

Change the setting of this register while OCU's operation is disabled.

For the operation modes by this register setting, see "4.4 Details of OCU Output Waveform".

[bit7:6] Reserved

Process	Function
Write	The written value is ignored.
Read	An undefined value is read.

4.3.9. OCU Compare Value Store Register (OCCP)

OCCP is a 16-bit register that specifies the timing of changing OCU's output signal as the compare value of FRT's count value.

Each mounted channel has six registers: OCCP0 to OCCP5.

OCCP0 stores the compare value of OCU ch.0 (2-change mode, ch.1 compare value).

OCCP1 stores the compare value of OCU ch.1.

OCCP2 stores the compare value of OCU ch.2 (2-change mode, ch.3 compare value).

OCCP3 stores the compare value of OCU ch.3.

OCCP4 stores the compare value of OCU ch.4 (2-change mode, ch.5 compare value).

OCCP5 stores the compare value of OCU ch.5.

It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OCCP[15:0]															
Attribute	R/W															
Initial Value	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

■ Functions of Register

[bit15:0] OCCP.OCCP[15:0]

Process	Function
Write	Specifies the timing of changing OCU's output signal. Stores the written value to the buffer register.
Read	Reads the value in the OCCP register (not the value in the OCCP buffer register).

OCCP is a 16-bit register that specifies the timing of changing OCU's output signal as the compare value of FRT's count value.

When data is written to this address area, the data is first stored in the buffer register. And then, the data is transferred from the buffer register to the OCCP register under the following conditions.

When the buffer function is disabled:

Data is transferred immediately after it is written to the buffer register.

When the buffer function is enabled and the transfer upon Zero value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has reached "0x0000".

When the buffer function is enabled and the transfer upon Peak value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has matched the TCCP value.

The enabling/disabling of the buffer function and the timing of data transfer are determined by the value of the corresponding register OCSA.BDIS1/BDSI0 or OCSB.BTS1/BTS0.

When OCU's operation is enabled, the pulse width of OCU's output signal can be changed by rewriting to this register. When the buffer function is disabled, the written value can be immediately reflected on the OCCP register. When the buffer function is enabled, the settings in the OCCP register for multiple channels can be synchronized.

If data is read from this address area, the value in the OCCP register is read, rather than the value in the buffer register. Therefore, it should be noted that no bit can be rewritten by RMW access to this address area when the buffer function is enabled.

If "0x0000" or "0xFFFF" is written to this register when FRT is in Up/Down-count mode, a fixed value can be output. For details, see "4.4 Details of OCU Output Waveform".

4.3.10. WFG Control Register A (WFSA)

WFSA is a 16-bit register that controls WFG's operation.

Each mounted channel has three registers: WFSA10, WFSA32 and WFSA54.

WFSA10 controls WFG ch.10 (the output processing block of OCU ch.1 and OCU ch.0).

WFSA32 controls WFG ch.32 (the output processing block of OCU ch.3 and OCU ch.2).

WFSA54 controls WFG ch.54 (the output processing block of OCU ch.5 and OCU ch.4).

It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	DMOD	PGEN[1:0]	PSEL[1:0]		
Attribute	-	-	-	R/W	R/W		R/W	
Initial Value	-	-	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Field	GTEN[1:0]		TMD[2:0]		DCK[2:0]			
Attribute	R/W		R/W		R/W			
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit2:0] WFSA.DCK[2:0]

Process	Value	Function
Write	000	Sets the count clock cycle of the WFG timer to the same value as PCLK.
	001	Sets the count clock cycle of the WFG timer to PCLK multiplied by 2.
	010	Sets the count clock cycle of the WFG timer to PCLK multiplied by 4.
	011	Sets the count clock cycle of the WFG timer to PCLK multiplied by 8.
	100	Sets the count clock cycle of the WFG timer to PCLK multiplied by 16.
	101	Sets the count clock cycle of the WFG timer to PCLK multiplied by 32.
	110	Sets the count clock cycle of the WFG timer to PCLK multiplied by 64
	Other than above	Setting prohibited
Read	-	Reads the register setting.

WFSA.DCK[2:0] is a register that sets the count clock cycle of the WFG timer.

Change the setting of this register, while the WFG timer is stopping.

The count clock of the WFG timer is generated by dividing the PCLK in LSI by the pre-scaler. This register sets the division ratio of the pre-scaler.

The count clock cycle of the WFG timer is determined according to PCLK cycle and the clock division ratio set by this register.

The table below shows examples of DCK[2:0] settings and the count clock cycle of the WFG timer.

DCK[2:0]	Clock Ratio	Count Clock Cycle of WFG Timer		
		PCLK=25ns (40MHz)	PCLK=33.3ns (33MHz)	PCLK=50ns (25MHz)
000	1	25ns	30ns	50ns
001	2	50ns	61ns	100ns
010	4	100ns	121ns	200ns
011	8	200ns	242ns	400ns
100	16	400ns	485ns	800ns
101	32	800ns	970ns	1.6μs
110	64	1.6μs	1.9μs	3.2μs

[bit5:3] WFSA.TMD[2:0]

Process	Value	Function
Write	000	Sets WFG's operation mode to Through mode.
	001	Sets WFG's operation mode to RT-PPG mode.
	010	Sets WFG's operation mode to Timer- PPG mode.
	100	Sets WFG's operation mode to RT dead timer mode.
	111	Sets WFG's operation mode to PPG dead timer mode.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

WFSA.TMD[2:0] is a register that selects WFG's operation mode.

For the operation modes by this register setting, see "4.5 Details of WFG Output Waveform".

When WFG's operation mode is set to Through mode (TMD[2:0]=000) or RT-PPG mode (TMD[2:0]=001), the WFG timer can be used as an independent reload timer.

Change the setting of this register while OCU and PPG timer unit to be connected are stopping. If the value set in this register is rewritten to a different value, the count state of the WFG timer is reset.

[bit7:6] WFSA.GTEN[1:0]

Process	Value	Function
Write	00	Does not generate the CH_GATE signal.
	Other than above	Generates the CH_GATE signal. For details, see "4.5 Details of WFG Output Waveform".
Read	-	Reads the register setting.

WFSA.GTEN[1:0] is a register that selects the output condition of the CH_GATE for each channel of WFG, in combination with WFSA.TMD[2:0].

This register has no meaning, when WFSA.TMD[2:0] is set to 000,100,. Change the setting, while OCU and PPG timer units to be connected are stopping.

The CH_GATE signal is generated based on the RT input signal and WFG timer operation at each channel of WFG (for details, see "4.5 Details of WFG Output Waveform").

The start trigger signal (GATE signal) for the PPG timer unit is generated from the CH_GATE signal (for details, see the section regarding WFSA.PSEL[1:0]).

The PPG timer unit to be connected to WFG can start the output of the PPG signal, using the GATE signal, and each channel of WFG can superimpose the PPG signal from the PPG timer unit on the RTO signal output (for details, see the section regarding WFSA.PGEN[1:0] and "4.5 Details of WFG Output Waveform").

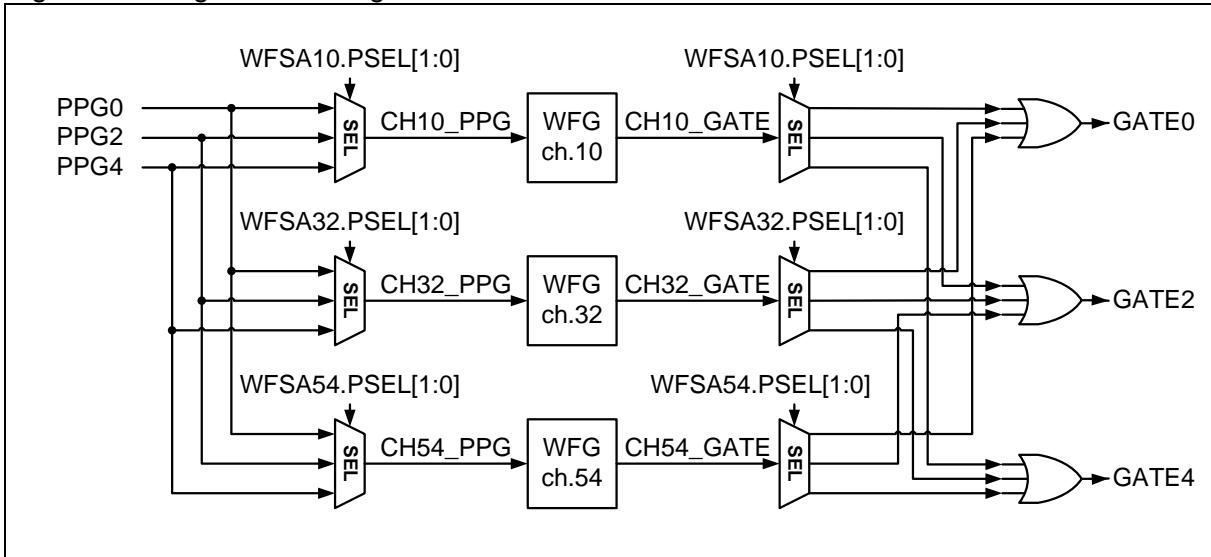
[bit9:8] WFSA.PSEL[1:0]

Process	Value	Function
Write	00	Sets the output destination of the GATE signal to ch.0 of the PPG timer unit. Sets the input source of the PPG signal to ch.0 of the PPG timer unit.
	01	Sets the output destination of the GATE signal to ch.2 of the PPG timer unit. Sets the input source of the PPG signal to ch.2 of the PPG timer unit.
	10	Sets the output destination of the GATE signal to ch.4 of the PPG timer unit. Sets the input source of the PPG signal to ch.4 of the PPG timer unit.
	11	Setting prohibited
Read	-	Reads the register setting.

WFSA.PSEL[1:0] is a register that selects the PPG timer unit to be used at each channel of WFG.

This register selects the PPG timer unit to be used as the output destination of the GATE signal and the input source of the PPG signal at once. Change the setting of this register, while OCU and PPG timer unit to be connected are stopping. Figure 4-6 shows a diagram of configuration of the PPG selector for MFTunit0.

Figure 4-6 Diagram of Configuration of PPG Selector for MFTunit0



The following section describes the configuration and operation of the PPG selector.

Each channel of WFG can output a trigger signal (CH_GATE signal) to start the PPG timer unit.

The CH10_GATE signal, CH32_GATE signal and CH54_GATE signal refer to the GATE signal for each channel of WFG, which has been generated at WFG ch10, WFG ch32 and WFG ch54, respectively.

After its output is selected by WFSA.PSEL[1:0] for each PPG timer unit to be connected, each CH_GATE signal undergoes logic OR by PPG timer unit and is output to each PPG unit.

The GATE0signal, GATE2 signal and GATE4 signal refer to the GATE signal that is output to ch.0, ch.2 and ch.4 of the PPG timer unit, respectively.

Each PPG timer unit can be started by the GATE signal and output the PPG signal.

The PPG0 signal, PPG2 signal and PPG4 signal refer to the PPG signal that is output from ch.0, ch.2 and ch.4 of the PPG timer unit, respectively, and input to WFG.

The CH10_PPG signal, CH32_PPG signal and CH54_PPG signal refer to the PPG signal that is used at WFG ch10, WFG ch32 and WFG ch54 respectively, whose input has been selected by WFSA.PSEL[1:0].

- Setting example 1)
WFSA10.PSEL[1:0]=00, WFSA32.PSEL[1:0]=00, and WFSA54.PSEL[1:0]=00 selects the common use of ch.0 of the PPG timer unit at all the channels of WFG.

GATE0 becomes the logic OR signal of CH10_GATE, CH32_GATE and CH54_GATE. Both GATE2 and GATE4 are set to fixed Low output. Each channel of WFG instructs ch.0 of the PPG timer unit to start up.

All of CH10_PPG, CH32_PPG and CH54_PPG become the PPG0 signal. Each channel of WFG uses the output signal of ch.0 of the PPG timer unit for waveform generation.

- Setting example 2)
WFSA10.PSEL[1:0]=00, WFSA32.PSEL[1:0]=01, and WFSA54.PSEL[1:0]=10 selects the individual use of ch.0, ch.2 and ch.4 of the PPG timer unit for each channel of WFG.

GATE0=CH10_GATE, GATE2=CH32_GATE, and GATE4=CH54_GATE are output, separately. Each channel instructs ch.0, ch.2 or ch.4 of the PPG timer unit to start up, individually.

CH10_PPG = PPG0, CH32_PPG=PPG2, and CH54_PPG=PPG4 are set. Each channel of WFG uses the output signal of the corresponding PPG timer unit for waveform generation.

<Notes>

- WFSA.PSEL[1:0] is set differently between MFTunit0 and MFTunit1 and the channel number of the PPG timer unit to be connected is also different. The descriptions above are intended for WFSA.PSEL[1:0] in MFTunit0. For information about MFTunit1, see "5.1 Connection of Model Containing Multiple MFT's".
 - To use the GATE signal, the PPG timer unit must be set beforehand. For details, see the chapter "PPG".
 - Even without the use of the GATE signal, the PPG timer unit can start outputting upon instruction by CPU.
-

[bit11:10] WFSA.PGEN[1:0]

Process	Value	Function
Write	00	Does not reflect the CH_PPG signal on WFG output (RTO output).
	Other than above	Specifies the condition to be used to reflect the CH_PPG signal on WFG output. For details of the reflection conditions, see "4.5 Details of WFG Output Waveform".
Read	-	Reads the register setting.

WFSA.PGEN[1:0] is a register that specifies how to reflect the CH_PPG signal that is input to each channel of WFG on WFG output, in combination with WFSA.TMD[2:0].

When WFG's operation mode is set to Through mode, the CH_PPG signal can be output to the RTO pin without any change, according to the setting of WFSA.PGEN[1:0]. This register setting has no meaning, if WFSA.TMD[2:0] is set to 100, 111. Change the setting, while OCU and PPG timer unit to be connected are stopping.

[bit12] WFSA.DMOD

Process	Value	Function
Write	0	Sets the output polarity for the output of the non-overlap signal to normal polarity (Active High).
	1	Sets the output polarity of the non-overlap signal to reversed polarity (Active Low).
Read	-	Reads the register setting.

WFSA.DMOD is a register that specifies which polarity will be used to output the non-overlap signal.

In the case of WFSA.TMD[2:0]=100 or WFSA.TMD[2:0]=111, the non-overlap signal is output for WFG's RTO(0) and RTO(1) output. The output polarity can be selected by this register setting. The register setting has no meaning, unless WFSA.TMD[2:0] is set to "100", "111". Change the setting, while OCU and PPG timer unit to be connected are stopping.

[bit13] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

[bit15:14] Reserved

Process	Function
Write	Writing is ignored.
Read	An undefined value is read.

4.3.11. WFG Timer Value Register (WFTM)

WFTM is a 16-bit register that sets the initial value of the WFG timer. Each mounted channel has three registers: WFTM10, WFTM32 and WFTM54. WFTM10 sets the initial value of the WFG timer for WFG ch10 (the output processing block of OCU ch1 and ch0). WFTM32 sets the initial value of the WFG timer for WFG ch32 (the output processing block of OCU ch3 and ch2). WFTM54 sets the initial value of the WFG timer for WFG ch54 (the output processing block of OCU ch5 and ch4). It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	WFTM[15:0]															
Attribute	R/W															
Initial Value	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

■ Functions of Register

[bit15:0] WFTM.WFTM[15:0]

Process	Function
Write	Sets the initial value of the WFG timer. Setting "0x0000" means 65536.
Read	Reads the register setting.

WFTM[15:0] is a 16-bit register that sets the initial value of the WFG timer.

The operating time of the WFG timer can be set as shown below, according to the setting of WFTM.

Operating time of WFG timer = WFTM value x Operation clock cycle of WFG timer

When WFG's operation mode is Timer PPG mode, RT dead timer mode or PPG dead timer mode (WFSA.TMD[2:0]=010, 100, 111), the WFG timer loads the initial value from the WFTM register, starts Down-count operation, when instructed to start up, and then stops once the counting is completed.

- In Timer PPG mode:
WFG timer counts the time set in the WFG timer operation flag.
- In RT dead timer mode and PPG dead timer mode:
WFG timer counts the dead time of the non-overlap signal.

When WFG's operation mode is Through mode or RT-PPG mode (WFSA.TMD[2:0]=000, 001), the WFG timer is not used for generation of output waveforms. In these modes, therefore, it can be used as a reload timer that generates interrupts to CPU in the intervals set by WFTM. For information about how to use it as a reload timer, see "4.3.13 WFG Interrupt Control Register (WFIR)".

This register can be rewritten, regardless of whether the WFG timer is currently operating or stopping. A new value rewritten to this register becomes valid from the next startup of the timer.

4.3.12. NZCL Control Register (NZCL)

NZCL is a 16-bit register that controls DTIF interrupt (interrupt for emergency motor shutdown by signal input from the DTTIX pin).

It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit Field	15	14	13	12	11	10	9	8
Attribute	Reserved							
Initial Value	0	0	0	0	-	0	0	0
Bit Field	7	6	5	4	3	2	1	0
Attribute	Reserved SDTI NWS[2:0] DTIE							
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

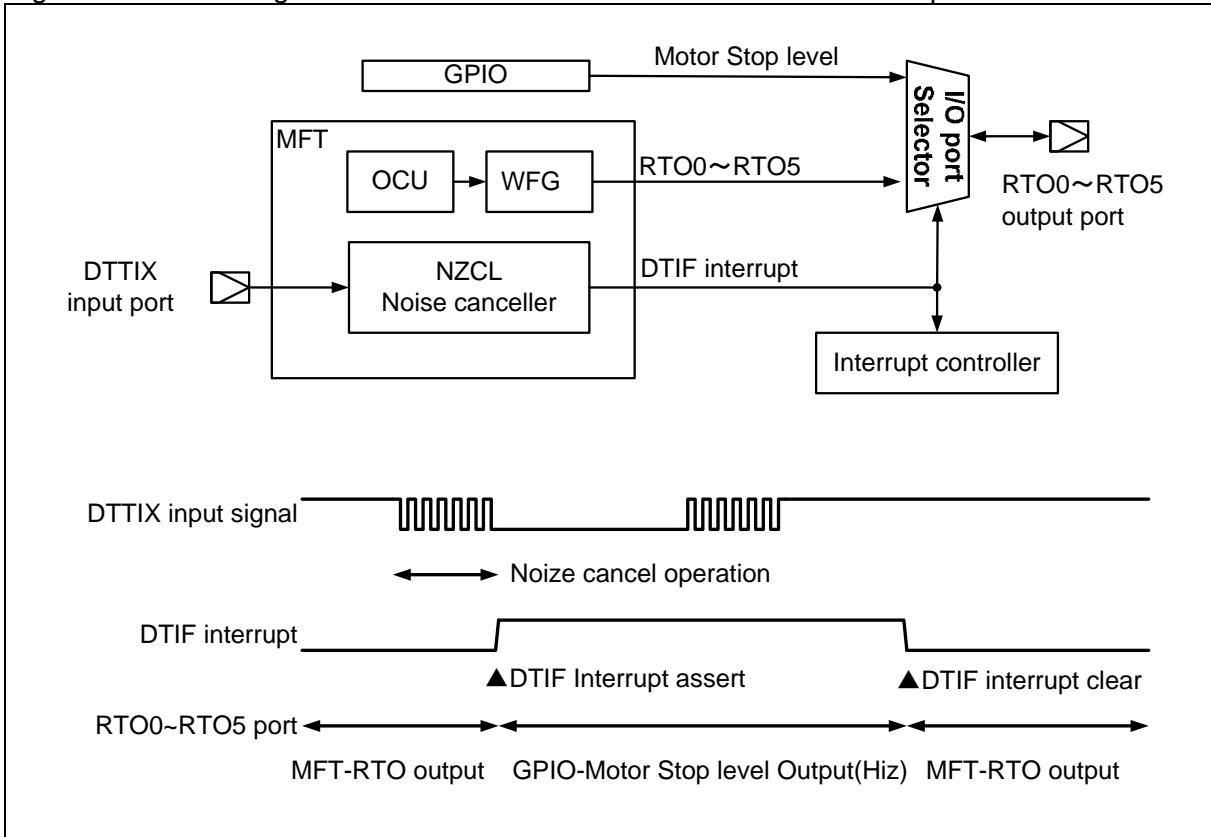
[bit0] NZCL.DTIE

Process	Value	Function
Write	0	Ignores the signal input from the DTTIX pin.
	1	Generates DTIF interrupt by signal input from the DTTIX pin.
Read	-	Reads the register setting.

NZCL.DTIE is a register that specifies whether or not to generate DTIF interrupt by signal input from the DTTIX pin.

Figure 4-7 shows a block diagram and time chart of the DTIX pin and DTIF interrupt.

Figure 4-7 Block Diagram and Time Chart of DTIX Pin and DTIF Interrupt



The DTTIX pin is a special pin dedicated to inputting an external interrupt signal for emergency motor shutdown. When the Low level is input, it recognizes the signal as a request for emergency motor shutdown. The input signal of this pin is input to the noise canceler. If a Low-level pulse no less than the value set by the noise canceler is input, the WFIR.DTIF register is set, the DTIF interrupt signal is asserted, and an interrupt is generated to CPU.

The DTIF interrupt signal is connected to the interrupt controller and the I/O port selector.

The I/O port selector can switch the state of the RTO0 to RTO5 output pins to the setting state of the sharing GPIO port, while DTIF interrupt is being generated.

The signal required for emergency motor shutdown can be output to the RTO0 to RTO5 pins by setting the GPIO pin shared with the RTO0 to RTO5 pins to the motor non-operating level beforehand.

The generated interrupt signal is deasserted by clearing the WFIR.DTIF register (writing "1" to the WFIR.DTIC).

Table 4-5 shows a list of function settings of the GPIO pin.
PFR, DDR and PDOR in the table refer to the corresponding registers of the GPIO port that are shared with the RTO0 to RTO5 pins.

Table 4-5 Setting List of Motor Non-operating Level by DTTIX Pin Interrupt

	Setting of GPIO Register					DTIF Signal Level	State of RTO Pin
	PFR	EPFR1 [11:0]	EPFR1 [12]	DDR	PDOR		
When switching the output state of the pin by DTIF interrupt	1	101010101010 Or 010101010101	1	1	1	0	Output RTO0 to RTO5
				1	0	1	Output High level
				0	don't care	0	Output RTO0 to RTO5
			0	1	don't care	1	Output Low level
				0	don't care	0	Output RTO0 to RTO5
				1	don't care	1	Hi-Z state
When not switching the output state of the pin by DTIF interrupt			0	don't care	don't care	0	Output RTO0 to RTO5
						1	

- PFR, EPFR1[11:0] is the basic setting for using the LSI pin as RTO output of MFT.
- EPFR1[12] is a bit that specifies whether or not to switch the pin function by interrupt.
- The EPFR1 register controls the pin used in MFTunit0. In the case of MFTunit1, the EPFR2 is used.
- Setting the DDR, PDOR register specifies the motor non-operating level when the pin function is switched.

If the output state is not to be switched by DTIF interrupt (EPFR1[12]=0), the state of the output pin is not switched, but DTIF interrupt is generated; therefore, CPU can receive interrupt notification.

[bit3:1] NZCL.NWS[2:0]

Process	Value	Function
Write	000	DTIF interrupt is generated immediately after Low-leve input from the DTTIX pin. (No noise-canceling)
	001	Sets the noise-canceling width to 4 PCLK cycles.
	010	Sets the noise-canceling width to 8 PCLK cycles.
	011	Sets the noise-canceling width to 16 PCLK cycles.
	100	Sets the noise-canceling width to 32 PCLK cycles.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

NZCL.NWS[2:0] is a register that sets the noise-canceling width of the noise-canceler for the DTTIX pin input signal.

[bit4] NZCL.SDTI

Process	Value	Function
Write	0	Does nothing.
	1	Forcibly generates DTIF interrupt, rather than by NZCL.DTIE setting.
Read	-	"0" is always read.

NZCL.SDTI is a register that generates DTIF interrupt by writing to the register via software.

Writing "1" to this register sets the WFIR.DTIF and generates interrupt, irrespective of NZCL.DTIE setting and the state of the DTTIX pin. Writing to this register allows for the use of the output switch function for the RTO pin in the I/O port controller. The generated interrupt signal is deasserted by clearing the WFIR.DTIF register (i.e. writing "1" to the WFIR.DTIC register).

[bit15:5] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

4.3.13. WFG Interrupt Control Register (WFIR)

WFIR is a register that controls DTIF interrupt and the interrupt from the WFG timer. This register is a special register dedicated to interrupt control, and each register bit is configured so that its state is not affected by writing "0". For this reason, reading before writing to the register is not required. Also, each register bit is configured so that its state is not affected by writing the read value back. It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	TMIS54	TMIE54	TMIC54	TMIF54	TMIS32	TMIE32	TMIC32	TMIF32
Attribute	W	R/W	W	R	W	R/W	W	R
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TMIS10	TMIE10	TMIC10	TMIF10	Reserved		DTIC	DTIF
Attribute	W	R/W	W	R	-		W	R
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit0] WFIR.DTIF

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that DTIF interrupt has not been generated.
	1	Indicates that DTIF interrupt has been generated.

[bit1] WFIR.DTIC

Process	Value	Function
Write	0	Does nothing.
	1	Clears WFIR.DTIF and deasserts the DTIF interrupt signal.
Read	-	"0" is always read.

WFIR.DTIF is a register that checks the state of DTIF interrupt.

WFIR.DTIC is a register that clears WFIR.DTIF and deasserts the DTIF interrupt signal.

The WFIR.DTIF register is set by inputting the emergency motor shutdown signal from the DTTIX pin or writing "1" to the NZCL.SDTI register. When WFIR.DTIF is set, the DTIF interrupt signal is asserted and an interrupt is generated to CPU.

Writing "1" to WFIR.DTIC clears WFIR.DTIF and deasserts the DTIF interrupt signal.

If interrupt processing has been performed by DTIF interrupt, make sure to clear DTIF when returning from the interrupt.

[bit3:2] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

[bit4] WFIR.TMIF10

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that WFG10 timer interrupt has not been generated.
	1	Indicates that WFG10 timer interrupt has been generated.

[bit5] WFIR.TMIC10

Process	Value	Function
Write	0	Does nothing.
	1	Clears WFIR.TMIF10 and deasserts the WFG10 timer interrupt signal.
Read	-	"0" is always read.

[bit6] WFIR.TMIE10

Process	Value	Function
Write	0	Does nothing.
	1	Starts the WFG10 timer (or does nothing, if it has already been started).
Read	0	Indicates that the WFG10 timer is currently stopped.
	1	Indicates that the WFG10 timer is currently in operation.

[bit7] WFIR.TMIS10

Process	Value	Function
Write	0	Does nothing.
	1	Stops the WFG10 timer (and also clears an interrupt at the same time, if it occurs, and deasserts the interrupt signal).
Read	-	"0" is always read.

WFIR.TMIF10 is a register that checks the state of WFG10 timer interrupt.

WFIR.TMIC10 is a register that clears WFG10 timer interrupt and deasserts the interrupt signal.

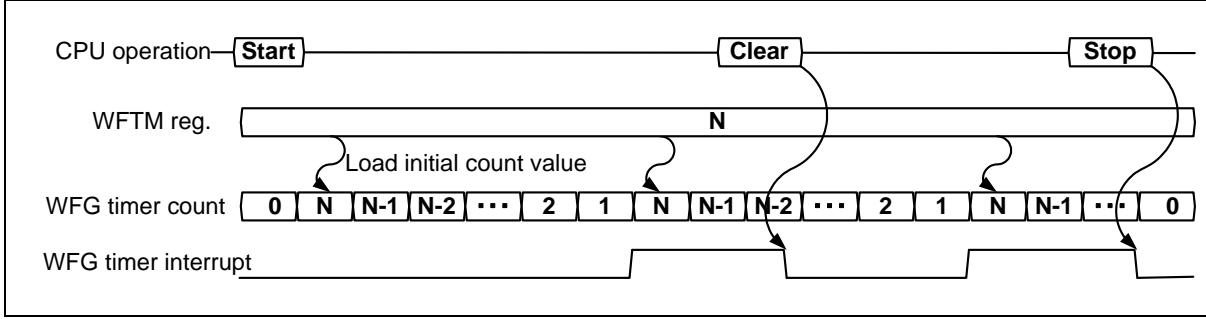
WFIR.TMIE10 is a register that starts the WFG10 timer.

WFIR.TMIS10 is a register that stops the WFG10 timer, clears the interrupt and deasserts the interrupt signal.

If the WFG timer for ch.10 of WFG is not used for waveform generation (WFSA10.TMD[2:0]=000, 001), the WFG10 timer can be used as an independent reload timer which generates interrupts regularly to CPU.

Figure 4-8 shows a diagram of the operation when the WFG timer is used as a reload timer.

Figure 4-8 Diagram of Operation when WFG Timer is Used as Reload Timer



Below is the procedure for using the WFG timer as a reload timer.

- First, set the initial value of the timer to the WFTM register and the clock division ratio to WFSA.DCK.
- Interval time of the interrupt generated from the timer = (WFTM value << WFSA.DCK) * PCLK cycle
- Writing "1" to WFIR.TMIE starts the timer.
- The WFG timer loads the initial value for the WFTM register, performs Down-count operation, and generates an interrupt when the count value is set to "1".
- At the same time, it reloads the initial value from the WFTM and continues the Down-count operation.
- If "1" is read from WFIR.TMIE, it indicates that the WFG timer is operating as a reload timer.
- If "1" is read from WFIR.TMIF, it indicates that an interrupt has occurred.
- Writing "1" to WFIR.TMIC allows WFIR.TMIF to be cleared and the interrupt signal to be deasserted.
- Writing "1" to WFIR.TMIS allows the count operation of the WFG timer to be stopped and no further interrupts to occur.
- If an interrupt has already occurred when "1" is written to WFIR.TMIS, the timer is stopped, WFIR.TMIF is cleared and the interrupt signal is deasserted at the same time.
- The value in the WFTM register can be rewritten during the timer operation. The changed value is reflected from the next reload time.
- If interrupt processing has been performed by WFG timer interrupt, make sure to clear WFIR.TMIF when returning from the interrupt.
- The following priority order applies to the processing, if "1" is written to WFIR.TMIS, WFIR.TMIC and WFIR.TMIE at the same time:
(Highest priority) Stopping the timer > Clearing the timer interrupt > Starting the timer (Lowest priority)

[bit8] WFIR.TMIF32

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that WFG32 timer interrupt has not been generated.
	1	Indicates that WFG32 timer interrupt has been generated.

[bit9] WFIR.TMIC32

Process	Value	Function
Write	0	Does nothing.
	1	Clears WFIR.TMIF32 and deasserts the interrupt signal of the WFG32 timer.
Read	-	"0" is always read.

[bit10] WFIR.TMIE32

Process	Value	Function
Write	0	Does nothing.
	1	Starts the WFG32 timer (or does nothing, if it has already been started).
Read	0	Indicates that the WFG32 timer is currently stopped.
	1	Indicates that the WFG32 timer is currently in operation.

[bit11] WFIR.TMIS32

Process	Value	Function
Write	0	Does nothing.
	1	Stops the WFG32 timer (and also clears an interrupt at the same time, if it occurs, and deasserts the interrupt signal).
Read	-	"0" is always read.

WFIR.TMIF32 is a register that checks the state of WFG32 timer interrupt.

WFIR.TMIC32 is a register that clears WFG32 timer interrupt and deasserts the interrupt signal.

WFIR.TMIE32 is a register that starts the WFG32 timer.

WFIR.TMIS32 is a register that stops the WFG32 timer, clears the interrupt and deasserts the interrupt signal.

If the WFG timer for ch.32 of WFG is not used for waveform generation (WFSA32.TMD[2:0]=000, 001), the WFG32 timer can be used as an independent reload timer which generates interrupts regularly to CPU.

This register is used in the same way as for WFIR.TMIF10, WFIR.TMIC10, WFIR.TMIE10 and WFIR.TMIS10.

[bit12] WFIR.TMIF54

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that the WFG54 timer interrupt has not been generated.
	1	Indicates that the WFG54 timer interrupt has been generated.

[bit13] WFIR.TMIC54

Process	Value	Function
Write	0	Does nothing.
	1	Clears WFIR.TMIF54 and deasserts the interrupt signal of the WFG54 timer.
Read	-	"0" is always read.

[bit14] WFIR.TMIE54

Process	Value	Function
Write	0	Does nothing.
	1	Starts the WFG54 timer (or does nothing, if it has already been started).
Read	0	Indicates that the WFG54 timer is currently stopped.
	1	Indicates that the WFG54 timer is currently in operation.

[bit15] WFIR.TMIS54

Process	Value	Function
Write	0	Does nothing.
	1	Stops the WFG54 timer (and also clears an interrupt at the same time, if it occurs, and deasserts the interrupt signal).
Read	-	"0" is always read.

WFIR.TMIF54 is a register that checks the state of WFG54 timer interrupt.

WFIR.TMIC54 is a register that clears WFG54 timer interrupt and deasserts the interrupt signal.

WFIR.TMIE54 is a register that starts the WFG54 timer.

WFIR.TMIS54 is a register that stops the WFG54 timer, clears the interrupt and deasserts the interrupt signal.

If the WFG timer for ch.54 of WFG is not used for waveform generation (WFSA54.TMD[2:0]=000, 001), the WFG54 timer can be used as an independent reload timer which generates interrupts regularly to CPU.

This register is used in the same way as for WFIR.TMIF10, WFIR.TMIC10, WFIR.TMIE10 and WFIR.TMIS10.

4.3.14. ICU Connecting FRT Select Register (ICFS)

ICFS is an 8-bit register that selects and sets FRT to be connected to ICU.

Each mounted channel has two registers: ICFS10 and ICFS32.

ICFS10 controls ICU ch1 and ICU ch0.

ICFS32 controls ICU ch3 and ICU ch2.

ICFS10 is located at an even-numbered address, while ICFS32 is located at an odd-numbered address; therefore, their bit positions are [7:0] and [15:8].

■ Configuration of Register

Bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
Field	FSI1[3:0]				FSI0[3:0]			
Attribute	R/W				R/W			
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit3:0/11:8] ICFS.FSI0[3:0]

Process	Value	Function
Write	0000	Connects FRT ch.0 to ICU ch.(0).
	0001	Connects FRT ch.1 to ICU ch.(0).
	0010	Connects FRT ch.2 to ICU ch.(0).
	0011 0100	For models with multiple MFT units: Connects FRT of an external MFT. For models with one MFT unit: Setting prohibited
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit7:4/15:12] ICFS.FSI1[3:0]

Process	Value	Function
Write	0000	Connects FRT ch.0 to ICU ch.(1).
	0001	Connects FRT ch.1 to ICU ch.(1).
	0010	Connects FRT ch.2 to ICU ch.(1).
	0011 0100	For models with multiple MFT units: Connects FRT of an external MFT. For models with one MFT unit: Setting prohibited
	Other than above	Setting prohibited
Read	-	Reads the register setting.

ICFS.FSI0[3:0] is a register that selects FRT to be connected to ICU-ch.(0) for use.

ICFS.FSI1[3:0] is a register that selects FRT to be connected to ICU-ch.(1) for use.

For models with multiple MFT units, the connection to FRT that exists in another MFT unit can be selected.
For related settings, see "5.1 Connection of Model Containing Multiple MFT's".

Change the setting of these registers, while the operation of the ICU to be connected is disabled.

4.3.15. ICU Control Register A (ICSA)

ICSA is an 8-bit register that controls ICU's operation.
Each mounted channel has two registers: ICSA10 and ICSA32.
ICSA10 controls ICU ch1 and ICU ch0.
ICSA32 controls ICU ch3 and ICU ch2.

■ Configuration of Register

Bit Field	7	6	5	4	3	2	1	0
Attribute	ICP1	ICP0	ICE1	ICE0	EG1[1:0]	EG0[1:0]		
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit1:0] ICSA.EG0[1:0]

Process	Value	Function
Write	00	Disables the operation of ICU ch.(0). Ignores IC(0) signal input.
	01	Enables the operation of ICU ch.(0). Treats only the rising edge of IC(0) signal input as a valid edge.
	10	Enables the operation of ICU ch.(0). Treats only the falling edge of IC(0) signal input as a valid edge.
	11	Enables the operation of ICU ch.(0). Treats both the rising and falling edges of IC(0) signal input as valid edges.
Read	-	Reads the register setting.

[bit3:2] ICSA.EG1[1:0]

Process	Value	Function
Write	00	Disables the operation of ICU ch.(1). Ignores IC(1) signal input.
	01	Enables the operation of ICU ch.(1). Treats only the rising edge of IC(1) signal input as a valid edge.
	10	Enables the operation of ICU ch.(1). Treats only the falling edge of IC(1) signal input as a valid edge.
	11	Enables the operation of ICU ch.(1). Treats both the rising and falling edges of IC(1) signal input as valid edges.
Read	-	Reads the register setting.

ICSA.EG0[1:0] is a register that enables/disables the operation of ICU-ch.(0) and selects a valid edge(s).

ICSA.EG1[1:0] is a register that enables/disables the operation of ICU-ch.(1) and selects a valid edge(s).

If a valid edge is detected at the input signal when ICU's operation is enabled, it performs the capture operation that captures FRT's count output to the ICCP register. At the same time, it notifies CPU that the valid edge has been detected. The valid edge of the input signal can be selected from the rising edge only, the falling edge only, or both rising and falling edges.

When the operation is disabled, it does nothing and ignores the input signal.

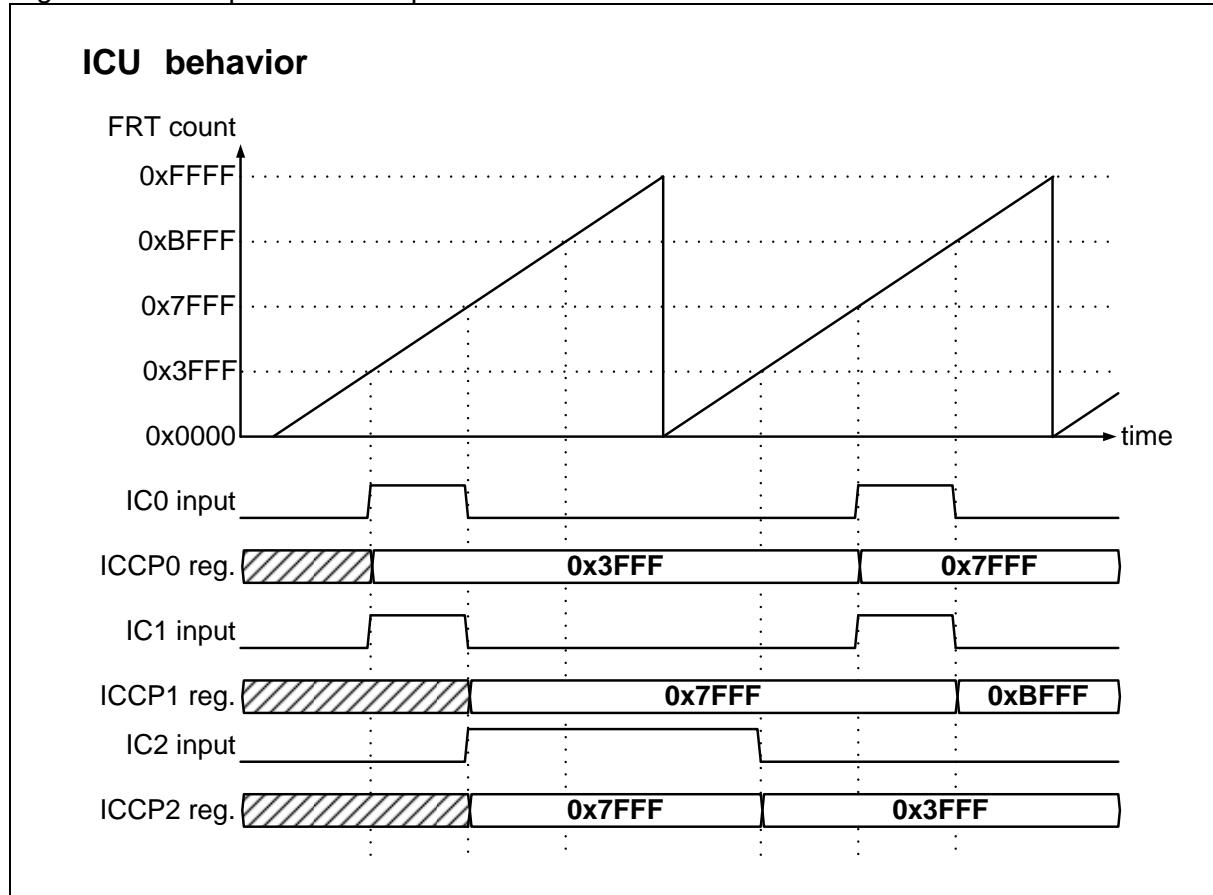
Figure 4-9 shows an example of ICU's operation.

ICU-ch.0 indicates the operation to be performed upon detection of the rising edge of IC0 signal input.

ICU-ch.1 indicates the operation to be performed upon detection of the falling edge of IC1 signal input.

ICU-ch.2 indicates the operation to be performed upon detection of both the rising and falling edges of IC signal input.

Figure 4-9 Example of ICU's Operation



[bit4] ICSA.ICE0

Process	Value	Function
Write	0	Does not generate interrupt, when "1" is set to ICSA.ICP0.
	1	Generates interrupt, when "1" is set to ICSA.ICP0.
Read	-	Reads the register setting.

[bit5] ICSA.ICE1

Process	Value	Function
Write	0	Does not generate interrupt, when "1" is set to ICSA.ICP1.
	1	Generates interrupt, when "1" is set to ICSA.ICP1.
Read	-	Reads the register setting

ICSA.ICE0 is a register that specifies whether to notify CPU of the event that "1" is set to ICSA.ICP0 as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

ICSA.ICE1 is a register that specifies whether to notify CPU of the event that "1" is set to ICSA.ICP1 as an interrupt (enabling interrupt) or not to notify it (disabling interrupt).

See "5.2 Treatment of Event Detect Register and Interrupt".

[bit6] ICSA.ICP0

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that no valid edge has been detected at ICU ch.(0) and no capture operation has been performed.
	1	Indicates that a valid edge has been detected at ICU ch.(0) and the capture operation has been performed.
Read at RMW access	"1"	is always read.

[bit7] ICSA.ICP1

Process	Value	Function
Write	0	Clears this register to "0".
	1	Does nothing.
Read	0	Indicates that no valid edge has been detected at ICU ch.(1) and no capture operation has been performed.
	1	Indicates that a valid edge has been detected at ICU ch.(1) and the capture operation has been performed.
Read at RMW access	"1"	is always read.

ICSA.ICP0 is a register to which "1" is set upon detection/capture of a valid edge, when the operation of ICU-ch.(0) is enabled.

ICSA.ICP1 is a register to which "1" is set upon detection/capture of a valid edge, when the operation of ICU-ch.(1) is enabled.

By reading from this register, it can be determined whether or not a valid edge has been detected and the capture operation has been performed.

This register can be cleared by writing "0".

This register does nothing, if "1" is written. Always write "1" to the register when rewriting to another register in the same address area.

"1" is always read from this register at RMW access.

See "5.2 Treatment of Event Detect Register and Interrupt".

4.3.16. ICU Control Register B (ICSB)

ICSB is an 8-bit register that reads the operation state of ICU. Each mounted channel has two registers: ICSB10 and ICSB32. ICSB10 reads the operation state of ICU ch1 and ICU ch0. ICSB32 reads the operation state of ICU ch3 and ICU ch2.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IEI1	IE10
Attribute	-	-	-	-	-	-	R	R
Initial Value	-	-	-	-	-	-	0	0

■ Functions of Register

[bit8] ICSB.IE10

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that the latest capture operation of ICU ch.(0) was performed at a falling edge.
	1	Indicates that the latest capture operation of ICU ch.(0) was performed at a rising edge.

[bit9] ICSB.IEI1

Process	Value	Function
Write	-	Writing is ignored.
Read	0	Indicates that the latest capture operation of ICU ch.(1) was performed at a falling edge.
	1	Indicates that the latest capture operation of ICU ch.(1) was performed at a rising edge.

ICSB.IE10 is a register that indicates the latest valid edge of ICU-ch.(0).

ICSB.IEI1 is a register that indicates the latest valid edge of ICU-ch.(1).

By reading from this register, at which edge the latest capture operation was performed can be determined.

As the initial value of this register is "0", "0" can be read if the capture operation has never been performed. It is also updated every time the valid edge of an input signal is detected. After the capture operation is performed, it is necessary to read from this register before the next valid edge.

[bit15:10] Reserved

Process	Function
Write	The written value is ignored.
Read	An undefined value is read.

4.3.17. ICU Capture value store register (ICCP)

ICCP is a 16-bit register that reads the value captured to ICU.
 Each mounted channel has four registers: ICCP0, ICCP1, ICCP2 and ICCP3.
 ICCP0 stores the capture value of ICU ch0.
 ICCP1 stores the capture value of ICU ch1.
 ICCP2 stores the capture value of ICU ch2.
 ICCP3 stores the capture value of ICU ch3.
 It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ICCP[15:0]															
Attribute	R															
Initial Value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

■ Functions of Register

[bit15:0] ICCP.ICCP[15:0]

Process	Function
Write	Writing is ignored.
Read	Reads the data captured to ICU.

ICCP is a 16-bit register that reads the value captured at each channel of ICU.

As the initial value of this register is undefined, a meaningless value is read if the capture operation has never been performed.

This register is updated every time the valid edge of an input signal is detected. After the capture operation is performed, it is necessary to read from this register before the next valid edge.

4.3.18. ADCMP Control Register A (ACSA)

ACSA is a 16-bit register that controls ADCMP's operation.
This register controls all of ch0, ch1 and ch2 of ADCMP.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	SEL2[1:0]		SEL1[1:0]		SEL0[1:0]	
Attribute	-	-	R/W		R/W		R/W	
Initial Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	CE2[1:0]		CE1[1:0]		CE0[1:0]	
Attribute	-	-	R/W		R/W		R/W	
Initial Value	0	0	0	0	0	0	0	0

■ Functions of Register

[bit1:0] ACSA.CE0[1:0]

Process	Value	Function
Write	00	Disables the operation of ADCMP ch.0.
	01	Enables the operation of ADCMP ch.0. Connects FRT ch.0 to ADCMP ch.0.
	10	Enables the operation of ADCMP ch.0. Connects FRT ch.1 to ADCMP ch.0.
	11	Enables the operation of ADCMP ch.0. Connects FRT ch.2 to ADCMP ch.0.
Read	-	Reads the register setting.

[bit3:2] ACSA.CE1[1:0]

Process	Value	Function
Write	00	Disables the operation of ADCMP ch.1.
	01	Enables the operation of ADCMP ch.1. Connects FRT ch.0 to ADCMP ch.1.
	10	Enables the operation of ADCMP ch.1. Connects FRT ch.1 to ADCMP ch.1.
	11	Enables the operation of ADCMP ch.1. Connects FRT ch.2 to ADCMP ch.1.
Read	-	Reads the register setting.

[bit5:4] ACSA.CE2[1:0]

Process	Value	Function
Write	00	Disables the operation of ADCMP ch.2.
	01	Enables the operation of ADCMP ch.2. Connects FRT ch.0 to ADCMP ch.2.
	10	Enables the operation of ADCMP ch.2. Connects FRT ch.1 to ADCMP ch.2.
	11	Enables the operation of ADCMP ch.2. Connects FRT ch.2 to ADCMP ch.2.
Read	-	Reads the register setting.

ACSA.CE0[1:0], ACSA.CE1[1:0], and ACSA.CE2[1:0] are registers that specify whether to enable or disable the operation of ADCMP-ch.0, ADCMP-ch.1, and ADCMP-ch.2 respectively and also select the FRT to be connected.

When the operation is enabled, ADCMP outputs the start instruction signal of AD conversion to ADC at the timing when there is a match between the compare value specified by the ADCMPDN register and the count value of the FRT connected. To enable the operation of ADCMP, make sure to set the values of the ACCP and ACCPDN registers beforehand.

ADCMP ch.0 instructs ADC unit0 to start AD conversion.

ADCMP ch.1 instructs ADC unit1 to start AD conversion.

ADCMP ch.2 instructs ADC unit2 to start AD conversion.

The AD conversion start signal output from ADCMP is connected to each ADC unit, after its output is selected by ATSA.

When the operation is disabled, ADCMP does nothing. ADCMP can only select the connection to the FRT that exists within its own MFT.

If the buffer function of ACCP and ACCPDN registers is to be used, use FRT-ch.0 for FRT to be connected.

[bit7:6] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

[bit9:8] ACSA.SEL0[1:0]

Process	Value	Function
Write	00	Instructs AD to be started, when FRT is in Up-count/Peak/Down-count state and it matches the setting value of ACCP0. Ignores the setting value of ACCPDN0.
	01	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP0. Ignores the setting value of ACCPDN0.
	10	Instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCP0. Ignores the setting value of ACCPDN0.
	11	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP0. Or instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCPDN0.
Read	-	Reads the register setting.

[bit11:10] ACSA.SEL1[1:0]

Process	Value	Function
Write	00	Instructs AD to be started, when FRT is in Up-count/Peak/Down-count state and it matches the setting value of ACCP1. Ignores the setting value of ACCPDN1.
	01	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP1. Ignores the setting value of ACCPDN1.
	10	Instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCP1. Ignores the setting value of ACCPDN1.
	11	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP1. Or instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCPDN1.
Read	-	Reads the register setting.

[bit13:12] ACSA.SEL2[1:0]

Process	Value	Function
Write	00	Instructs AD to be started, when FRT is in Up-count/Peak/Down-count state and it matches the setting value of ACCP2. Ignores the setting value of ACCPDN2.
	01	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP2. Ignores the setting value of ACCPDN2.
	10	Instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCP2. Ignores the setting value of ACCPDN2.
	11	Instructs AD to be started, when FRT is in Up-count state and it matches the setting value of ACCP2. Or instructs AD to be started, when FRT is in Peak/Down-count state and it matches the setting value of ACCPDN2.
Read	-	Reads the register setting.

ACSA.SEL0[1:0], ACSA.SEL1[1:0], and ACSA.SEL2[1:0] are registers that specify which count state FRT should be in to instruct AD conversion to be started at each channel of ADCMP.

Change the setting of these register, when the operation of ADCMP to be connected is disabled.

When using FRT in Up-count mode, set "00" to these registers for use.

Figure 4-10 shows an example of the operation when SEL is set to 00.

Figure 4-10 Example of Operation when SEL=00

ACSA.SEL=00

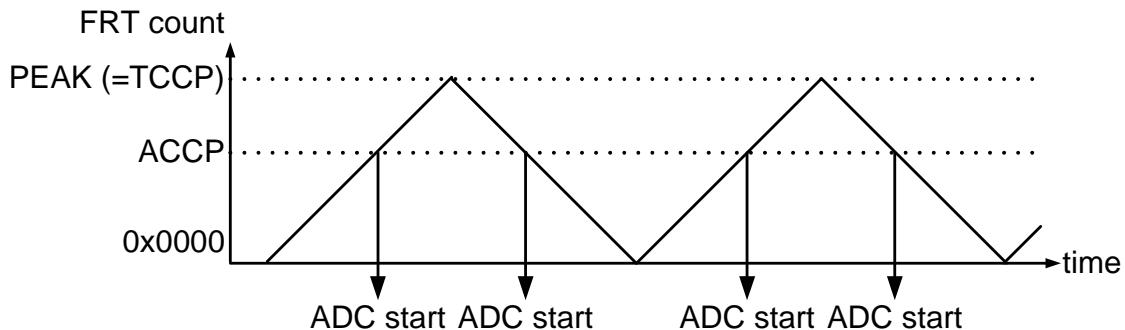


Figure 4-11 shows an example of the operation when SEL is set to 01.

Figure 4-11 Example of Operation when SEL=01

ACSA.SEL=01

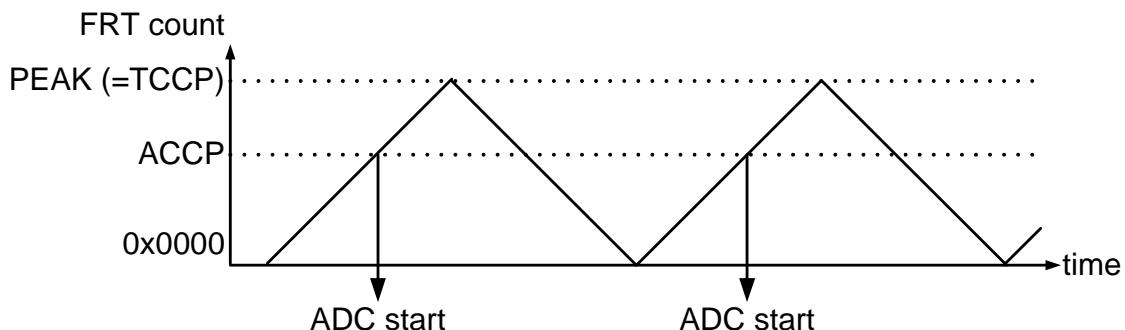


Figure 4-12 shows an example of the operation when SEL is set to 10.

Figure 4-12 Example of Operation when SEL=10

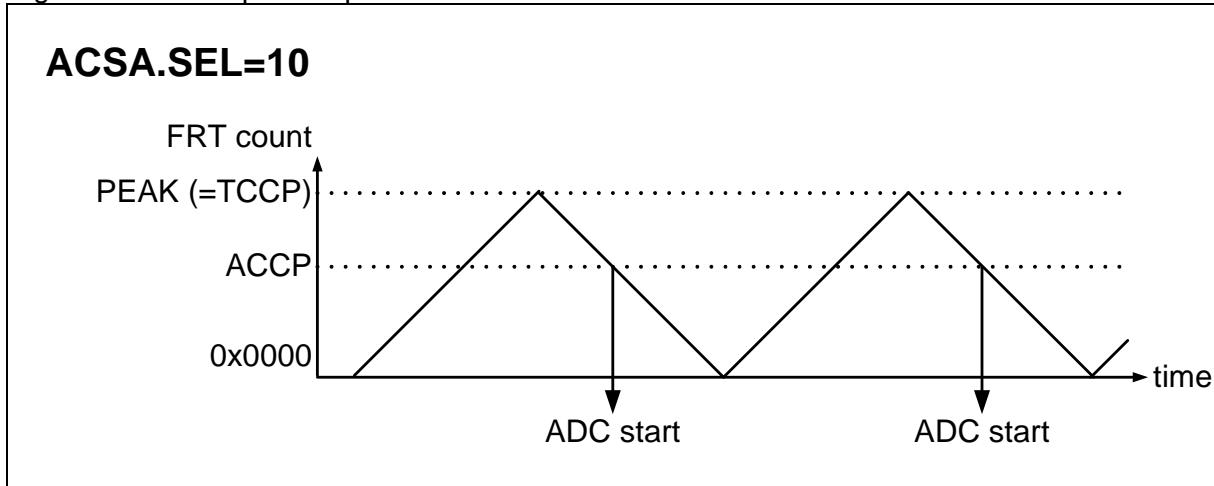
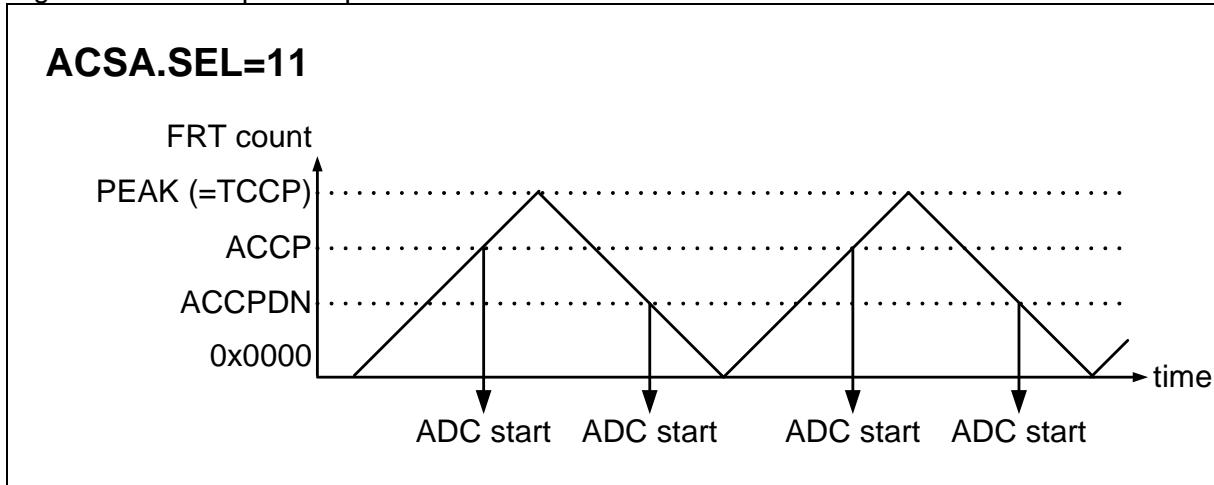


Figure 4-13 shows an example of the operation when SEL is set to 11.

Figure 4-13 Example of Operation when SEL=11



[bit15:14] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

4.3.19. ADCMP Control Register B (ACSB)

ACSB is an 8-bit register that controls ADCMP's operation.
This register controls all of ch.0, ch.1 and ch.2 of ADCMP.

■ Configuration of Register

Bit	7	6	5	4	3	2	1	0
Field	Reserved	BTS2	BTS1	BTS0	Reserved	BDIS2	BDIS1	BDIS0
Attribute	-	R/W	R/W	R/W	-	R/W	R/W	R/W
Initial Value	0	0	0	0	0	1	1	1

■ Functions of Register

[bit0] ACSB.BDIS0

Process	Value	Function
Write	0	Enables the buffer function of the ACCP0 and ACCPDN0 registers.
	1	Disables the buffer function of the ACCP0 and ACCPDN0 registers.
Read	-	Reads the register setting.

[bit1] ACSB.BDIS1

Process	Value	Function
Write	0	Enables the buffer function of the ACCP1 and ACCPDN1 registers.
	1	Disables the buffer function of the ACCP1 and ACCPDN1 registers.
Read	-	Reads the register setting.

[bit2] ACSB.BDIS2

Process	Value	Function
Write	0	Enables the buffer function of the ACCP2 and ACCPDN2 registers.
	1	Disables the buffer function of the ACCP2 and ACCPDN2 registers.
Read	-	Reads the register setting.

ACSB.BDIS0 is a register that specifies whether to enable or disable the buffer function of the ACCP0 and ACCPDN0 registers.

ACSB.BDIS1 is a register that specifies whether to enable or disable the buffer function of the ACCP1 and ACCPDN1 registers.

ACSB.BDIS2 is a register that specifies whether to enable or disable the buffer function of the ACCP2 and ACCPDN2 registers.

Change the setting of these registers, when the operation of ADCMP to be connected is disabled.

If the buffer function of ACCP and ACCPDN registers is to be used, use FRT-ch.0 for FRT to be connected.

See "4.3.20 ADCMP Compare Value Store Register (ACCP)" and "4.3.21 ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN)".

[bit3] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

[bit4] ACSB.BTS0

Process	Value	Function
Write	0	Performs buffer transfer of the ACCP0 and ACCPDN0 registers upon Zero value detection by FRT.
	1	Performs buffer transfer of the ACCP0 and ACCPDN0 registers upon Peak value detection by FRT.
Read	-	Reads the register setting.

[bit5] ACSB.BTS1

Process	Value	Function
Write	0	Performs buffer transfer of the ACCP1 and ACCPDN1 registers upon Zero value detection by FRT.
	1	Performs buffer transfer of the ACCP1 and ACCPDN1 registers upon Peak value detection by FRT.
Read	-	Reads the register setting.

[bit6] ACSB.BTS2

Process	Value	Function
Write	0	Performs buffer transfer of the ACCP2 and ACCPDN2 registers upon Zero value detection by FRT.
	1	Performs buffer transfer of the ACCP2 and ACCPDN2 registers upon Peak value detection by FRT.
Read	-	Reads the register setting.

ACSB.BTS0 is a register that specifies the timing of transferring data from the buffer register to the ACCP0 and ACCPDN0 registers when the buffer function is enabled.

ACSB.BTS1 is a register that specifies the timing of transferring data from the buffer register to the ACCP1 and ACCPDN1 registers when the buffer function is enabled.

ACSB.BTS2 is a register that specifies the timing of transferring data from the buffer register to the ACCP2 and ACCPDN2 registers when the buffer function is enabled.

Change the setting of these registers, when the operation of ADCMP to be connected is disabled.

See "4.3.20 ADCMP Compare Value Store Register (ACCP)" and "4.3.21 ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN)".

[bit7] Reserved

Process	Function
Write	"0" must be written at write access.
Read	"0" is read.

4.3.20. ADCMP Compare Value Store Register (ACCP)

ACCP is a 16-bit register that specifies the timing of starting AD conversion at ADCMP as the compare value of the FRT count value.

Each mounted channel has three registers: ACCP0, ACCP1 and ACCP2.

ACCP0 stores the compare value of ADCMP ch0.

ACCP1 stores the compare value of ADCMP ch1.

ACCP2 stores the compare value of ADCMP ch2.

It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ACCP[15:0]															
Attribute	R/W															
Initial Value	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

■ Functions of Register

[bit15:0] ACCP.ACCT[15:0]

Process	Function
Write	Specifies the timing of starting AD conversion. The value is written to the ACCP buffer register.
Read	Reads the value in the ACCP register (not the value in the ACCP buffer register).

ACCP is a register that specifies the timing of starting AD conversion.

Each specifies the timing of starting AD conversion in combination with the settings of ACSA.SEL0[1:0], ACSA.SEL1[1:0], and ACSA.SEL2[1:0].

When data is written to this address area, the data is first stored in the buffer register. And then, the data is transferred from the buffer register to the ACCP register under the following conditions.

When the buffer function is disabled:

Data is transferred immediately after it is written to the buffer register.

When the buffer function is enabled and the transfer upon Zero value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has reached "0x0000".

When the buffer function is enabled and the transfer upon Peak value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has matched the TCCP value.

The enabling/disabling of the buffer function and the timing of data transfer are determined by the value of the corresponding register ACSB.BDIS0, BDIS1, BDIS2, BTS0, BDIS1, or BDIS2.

During FRT's count operation, the timing of starting AD conversion can be changed by rewriting to this register. When the buffer function is disabled, the written value can be immediately reflected on the ACCP register. When the buffer function is enabled, the settings in the ACCP register for multiple channels can be synchronized.

If data is read from this address area, the value in the ACCP register is read, rather than the value in the buffer register. Therefore, it should be noted that no bit can be rewritten by RMW access to this address area when the buffer function is enabled.

<Notes>

- AD conversion cannot be started by writing "0x0000" to this register.
- As the initial value of this register is "0x0000", make sure to rewrite it to another value before use.
- To start AD conversion upon Zero value detection, use the starting method by the TCSB.AD0E, TCSB.AD1E and TCSB.AD2E registers.
- If the buffer function of ACCP register is to be used, use FRT-ch.0 for FRT to be connected.
- It should be noted that when ACSA.SEL0[1:0], SEL1[1:0], SEL[1:0] = "01", "10", "11" are set, FRT's Peak value (=TCCP) will be ignored, even if it is set to ACCP.

4.3.21. ADCMP Compare Value Store Register, Down-count Direction Only (ACCPDN)

ACCPDN is a 16-bit register that specifies the timing of starting AD conversion at ADCMP as the compare value of the FRT count value.

Each mounted channel has three registers: ACCPDN0, ACCPDN1 and ACCPDN2.

ACCPDN0 stores the compare value of ADCMP ch0.

ACCPDN1 stores the compare value of ADCMP ch1.

ACCPDN2 stores the compare value of ADCMP ch2.

It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ACCPDN[15:0]															
Attribute	R/W															
Initial Value	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

■ Functions of Register

[bit15:0] ACCPDN.ACCPDN[15:0]

Process	Function
Write	Specifies the timing of starting AD conversion. The value is written to the ACCPDN buffer register.
Read	Reads the value in the ACCPDN register (not the value in the ACCPDN buffer register).

ACCPDN is a register that specifies the timing of starting AD conversion.

Each specifies the timing of starting AD conversion, only when the setting values of ACSA.SEL0[1:0], ACSA.SEL1[1:0], ACSA.SEL2[1:0] is 11.

When data is written to this address area, the data is first stored in the buffer register. And then, the data is transferred from the buffer register to the ACCP register under the following conditions.

When the buffer function is disabled:

Data is transferred immediately after it is written to the buffer register.

When the buffer function is enabled and the transfer upon Zero value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has reached "0x0000".

When the buffer function is enabled and the transfer upon Peak value detection is enabled:

Data is transferred, when FRT's counter is stopped or when FRT's count value has matched the TCCP value.

The enabling/disabling of the buffer function and the timing of data transfer are determined by the value of the corresponding register ACSB.BDIS0, BDIS1, BDIS2, BTS0, BDIS1, or BDIS2.

During FRT's count operation, the timing of starting AD conversion can be changed by rewriting to this register. When the buffer function is disabled, the written value can be immediately reflected on the ACCPDN register. When the buffer function is enabled, the settings in the ACCPDN register for multiple channels can be synchronized.

If data is read from this address area, the value in the ACCPDN register is read, rather than the value in the buffer register. Therefore, it should be noted that no bit can be rewritten by RMW access to this address area when the buffer function is enabled.

<Notes>

- AD conversion cannot be started by writing "0x0000" to this register.
 - If the buffer function of ACCPDN register is to be used, use FRT-ch.0 for FRT to be connected.
-

4.3.22. ADC Start Trigger Select Register (ATSA)

ATSA is a 16-bit register that selects ADC's start signal which is output from MFT. This register is used to select a start trigger for ADCunit0, unit1 and unit2. It should be noted that this register does not allow for byte access.

■ Configuration of Register

Bit	15	14	13	12	11	10	9	8
Field	Reserved		AD2P		AD1P		AD0P	
Attribute	-		R/W		R/W		R/W	
Initial Value	-	-	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Field	Reserved		AD2S		AD1S		AD0S	
Attribute	-		R/W		R/W		R/W	
Initial Value	-	-	0	0	0	0	0	0

■ Functions of Register

[bit1:0] ATSA.AD0S[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.0 as ADC unit0 scan conversion start signal.
	01	Selects the OR signal of FRTch.0 to ch.2 start signal as ADC unit0 scan conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit3:2] ATSA.AD1S[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.1 as ADC unit1 scan conversion start signal.
	01	Selects the OR signal of FRTch.0 to ch.2 start signal as ADC unit1 scan conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit5:4] ATSA.AD2S[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.2 as ADC unit2 scan conversion start signal.
	01	Selects the OR signal of FRTch.0 to ch.2 start signal as ADC unit2 scan conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

ATSA.AD0S[1:0] is a register that selects the start signal to be used to start the scan conversion of ADC unit0.

ATSA.AD1S[1:0] is a register that selects the start signal to be used to start the scan conversion of ADC unit1.

ATSA.AD2S[1:0] is a register that selects the start signal to be used to start the scan conversion of ADC unit2.

The starting method used for ADC's scan conversion start signal that is output from MFT can be selected from starting by ADCMP or starting by FRTch.0 to ch.2. The above is selected by the setting in this register. Change the setting of this register, when the operation of ADCMP to be connected is disabled.

For models containing multiple MFT's, the ADC scan conversion start signal from MFT undergoes logic OR for each MFT unit, and then it is connected to ADC. For details, see the chapter "A/D Converter".

[bit7:6] Reserved

Process	Function	
Write	The written value is ignored.	
Read	An undefined value is read.	

[bit9:8] ATSA.AD0P[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.0 as ADC unit0 priority conversion start signal.
	01	Selects the logic OR signal of FRTch.0 to ch.2 as ADC unit0 priority conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit11:10] ATSA.AD1P[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.1 as ADC unit1 priority conversion start signal.
	01	Selects the logic OR signal of FRTch.0 to ch.2 as ADC unit1 priority conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

[bit13:12] ATSA.AD2P[1:0]

Process	Value	Function
Write	00	Selects the start signal of ADCMP ch.2 as ADC unit2 priority conversion start signal.
	01	Selects the logic OR signal of FRTch.0 to ch.2 as ADC unit2 priority conversion start signal.
	Other than above	Setting prohibited
Read	-	Reads the register setting.

ATSA.AD0P[1:0] is a register that selects the start signal to be used to start priority conversion of ADC unit0.

ATSA.AD1P[1:0] is a register that selects the start signal to be used to start priority conversion of ADC unit1.

ATSA.AD2P[1:0] is a register that selects the start signal to be used to start priority conversion of ADC unit2.

The starting method used for ADC's priority conversion start signal that is output from MFT can be selected from starting by ADCMP or starting by FRTch.0 to ch.2. The above is selected by the setting in this register. Change the setting of this register, when the operation of ADCMP to be connected is disabled.

For models containing multiple MFT's, the ADC priority conversion start signal from MFT undergoes logic OR for each MFT unit, and then it is connected to ADC. For details, see the chapter "A/D Converter".

[bit15:14] Reserved

Process	Function
Write	The written value is ignored.
Read	An undefined value is read.

4.4. Details of OCU Output Waveform

This section provides details of the output waveform of the RT output signal in each mode of OCU.

■ List of OCU Operation Modes

The operation modes of the OCU are selected by the following register settings. Table 4-6 shows a list of register setting values and the operation modes of OCU-ch.(0) and OCU-ch.(1).

Table 4-6 Register Setting Values and the Operation Modes of OCU-ch.(0) and OCU-ch.(1)

Register Setting				Operation Mode Selected	
TCSA. MODE -Ch.(1) (*1)	TCSA. MODE -Ch.(0) (*2)	OCSB. CMOD (*3)	OCSC. MOD (*4)	CH(1) Operation Mode	CH(0) Operation Mode
0	0	0	00	Up-count mode (1-change)	Up-count mode (1-change)
0	0	1	00	Up-count mode (2-change)	Up-count mode (1-change)
0	1	0	01	Up-count mode (1-change)	Up/Down-count mode (Active High)
1	0	0	10	Up/Down-count mode (Active High)	Up-count mode (1-change)
1	0	1	10	Up/Down-count mode (Active Low)	Up-count mode (1-change)
1	1	0	11	Up/Down-count mode (Active High)	Up/Down-count mode (Active High)
1	1	1	11	Up/Down-count mode (Active Low)	Up/Down-count mode (Active Low)

*1 TCSA.MODE-ch.(1) indicates the TCSA.MODE value of FRT to be connected to OCU-ch.(1) selected by the OCFS register.

*2 TCSA.MODE-ch.(0) indicates the TCSA.MODE value of FRT to be connected to OCU-ch.(0) selected by the OCFS register.

*3 OCSB.CMOD indicates the OCSB10.CMOD value for ch.1-ch.0. It indicates the OCSB32.CMOD value for ch.3-ch.2. It indicates the OCSB54.CMOD value for ch.5-ch.4.

*4 OCSC.MOD indicates the OCSC.MOD[1:0] value for ch.1-ch.0. It indicates the OCSC.MOD[3:2] value for ch.3-ch.2. It indicates the OCSC.MOD[5:4] value for ch.5-ch.4.

*5 OCSB.CMOD and OCSC.MOD[5:0] cannot be used in combinations other than listed above.

*6 OCU ch.(0) cannot use Up-count mode (2-change).

■ List of Changes of the RT(0) and RT(1) Signals in OCU Operation Modes

When each channel of OCU is in the state of Operation enabled, if the FRT counter value matches the OCCP register value, the output signal level changes. In addition, the changes of the output signal level are determined by the operation mode of OCU, the value of OCCP, and the count state of FRT. Table 4-7 shows a list of OCU-ch.(0) operation modes, register settings and RT(0) signal outputs. Table 4-8 shows a list of OCU-ch.(1) operation modes, register settings and RT(1) signal outputs.

Table 4-7 Details of OCU-ch.(0) Operation and RT(0) Signal Outputs

Name of Operation Mode	OCCP(0) value				
	0x0000	0xFFFF	Other than 0x0000 & 0xFFFF		
			Up	Peak	Down
Up-count mode (1-change)	M:Rev U:No	M:Rev U:No	M:Rev U:No	M:Rev U:No	-
Up/Down-count mode (Active High)	All-Act	All-Ina	M:Act U:No	M:No U:No (*7)	M:Ina U:No
Up/Down-count mode (Active Low)	All-Act	All-Ina	M:Act U:No		M:Ina U:No

Table 4-8 Details of OCU-ch.(1) Operation and RT(1) Signal

Name of Operation Mode	OCCP(1) value					OCCP(0) value	
	0x0000	0xFFFF	Other than 0x0000, 0xFFFF				
			Up	Peak	Down		
Up-count mode (1-change)	M:Rev U:No	M:Rev U:No	M:Rev U:No	M:Rev U:No	-	-	
Up-count mode (2-change)	M:Rev U:No	M:Rev U:No	M:Rev U:No	M:Rev U:No	-	M:Rev U:No	
Up/Down-count mode (Active High)	All-Act	All-Ina	M:Act U:No	M:No U:No (*7)	M:Ina U:No	-	
Up/Down-count mode (Active Low)	All-Act	All-Ina	M:Act U:No		M:Ina U:No	-	

* Meanings of symbols in Table 4-7 and Table 4-8

Up : Operation when FRT is up-counting

Peak : Operation when FRT's count value is the Peak value (=TCCP value)

Down : Operation when FRT is down-counting

M : Operation when FRT's count value matches the OCCP value

U : Operation when FRT's count value does not match the OCCP value

Rev : Change of the output signal level to the Reversed level.

Act : Change of the output signal level to the Active level. No change, if the previous output level was already Active.

Ina : Change of the output signal level to the Inactive level. No change, if the previous output level was already Inactive.

No : No change of the output signal level.

All-Act : Change of the output signal level to the Active level, while the OCCP value is that value.

All-Ina : Change of the output signal level to the Inactive level, while the OCCP value is that value.

*7 In Up/Down count mode, if the peak FRT count value matches OCCP value, the RT(0) and RT(1) output signals do not change, and the OCSA.IOP0 and IOP1 flags are not set.

■ Up-count Mode (1-change)

When Up-count mode (1-change) is selected, the following operation applies.

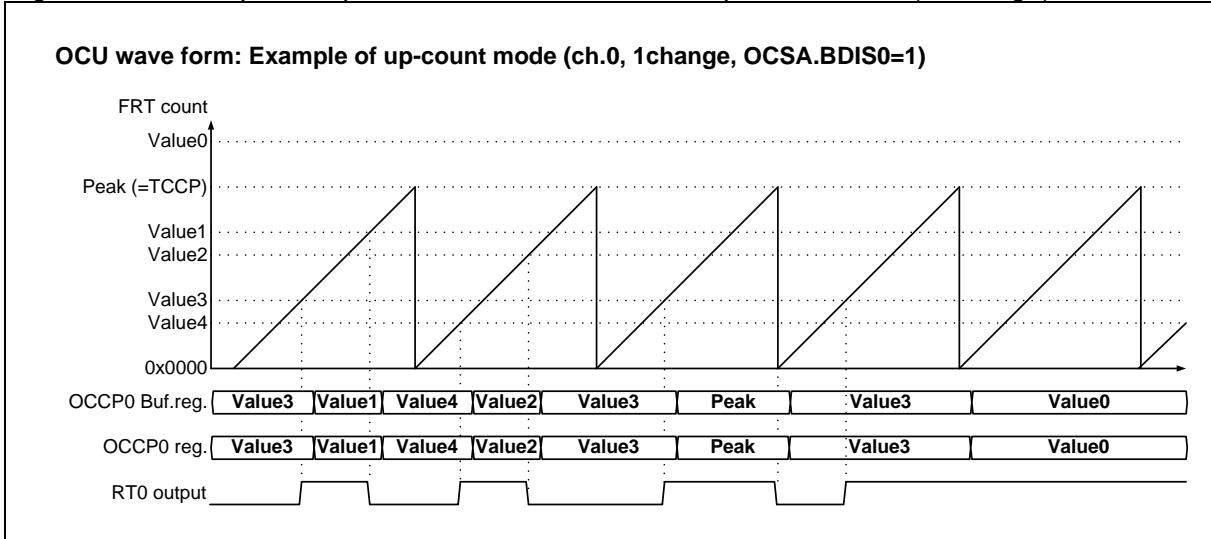
Regardless of FRT's count state, the output level of the RT(0) signal is reversed when FRT's count value matches OCCP(0).

Regardless of FRT's count state, the output level of the RT(1) signal is reversed when FRT's count value matches OCCP(1).

In this mode, OCU-ch.(0) and ch.(1) can operate independently from each other.

Figure 4-14 shows an example of operation waveform when OCU-ch.0 is in Up-count mode (1-change). This figure illustrates the state in which the buffer function of the OCCP0 register is disabled.

Figure 4-14 Example of Operation Waveform in OCU Up-count Mode (1-change)



<Note>

A note on Up-count mode (1-change) is as follows:

- If a value larger than the Peak value of FRT's counter (e.g. Value0 in Figure 4-14) is set to OCCP, the output does not change.

■ Up-count Mode (2-change)

When Up-count mode (2-change) is selected, the following operation applies.

Regardless of FRT's count state, the output level of the RT(1) signal is reversed when FRT's count value matches OCCP(0) or OCCP(1).

This mode can be used only by OCU ch.(1), not by ch.(0). Also, as OCU-ch.(0) and ch.(1) perform interlocked operation, they cannot operate independently from each other. If 2-change mode is selected for OCU-ch(1), OCU-ch.(0) operates in 1-change mode to perform the operation that changes according to the OCCP(0) value.

Figure 4-15 Example 1 of Operation Waveform in OCU Up-count Mode (2-change)

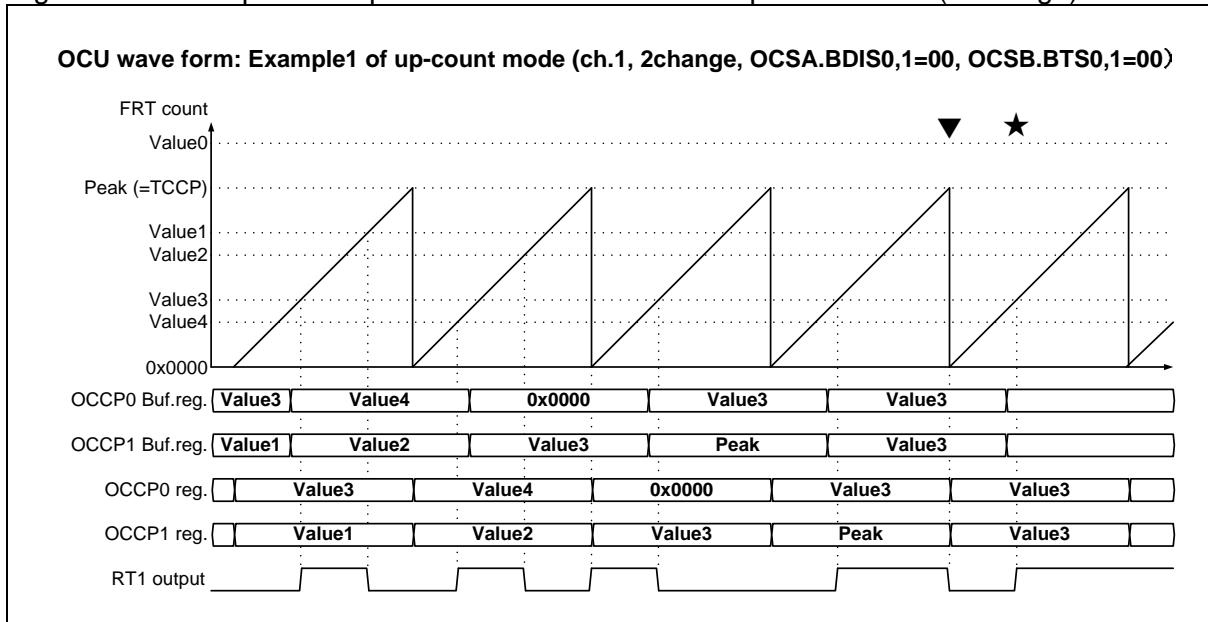


Figure 4-16 Example 2 of Operation Waveform in OCU Up-count Mode (2-change)

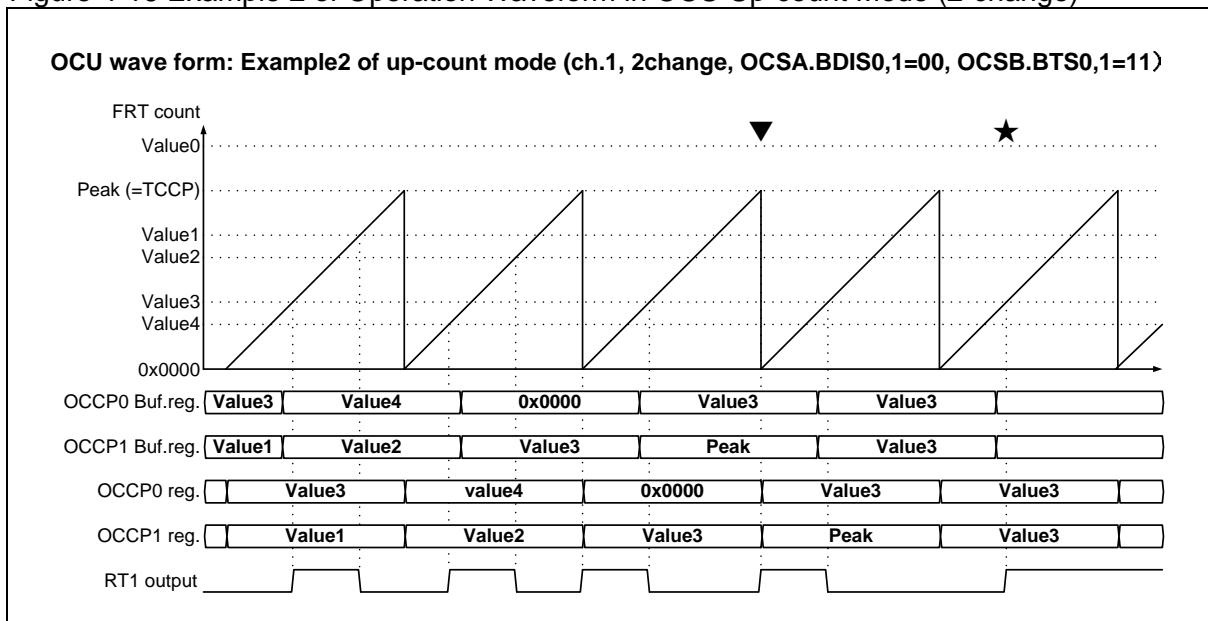


Figure 4-15 shows Example 1 of the operation waveform when OCU-ch.1 is in Up-count mode (2-change). This figure is based on the conditions that the buffer function of the OCCP1 register is enabled and Zero value transfer is set.

Figure 4-16 shows Example 2 of the operation waveform when OCU-ch.1 is in Up-count mode (2-change). This figure is based on the conditions that the buffer function of the OCCP1 register is enabled and Peak value transfer is set.

Due to the difference in the timing of OCCP data transfer between Figure 4-15 and Figure 4-16, they operate differently when the Peak value is set to OCCP1.

In the case of Figure 4-15, data is transferred from the OCCP1 buffer register to the OCCP1 register, when FRT's counter value is "0x0000". The output level is reversed at the timing indicated by ▼, under the condition: OCCP1=Peak value.

In the case of Figure 4-16, data is transferred from the OCCP1 buffer register to the OCCP1 register, when FRT's counter value reaches the Peak value. As the register values are compared immediately after the transfer, the output level is reversed at the timing indicated by ▼, under the condition: OCCP1=Peak value.

<Note>

Notes on Up-count mode (2-change) are as follows:

- If a value larger than the Peak value of FRT's counter (e.g. Value0 in Figure 4-15) is set to OCCP, the output does not change.
 - If the same value is set to OCCP(0) and OCCP(1), the output level is reversed at the timing indicated by ★, as shown in Figure 4-15 and Figure 4-16.
 - It is necessary to set the operation enable flags of both OCSA.CST0 and OCSA.CST1.
 - OCSA.IOP0 is set when FRT's count value matches OCCP(0).
 - OCSA.IOP1 is set when FRT's count value matches OCCP(1).
 - It is necessary to apply the same settings for which FRT is to be connected; whether to enable or disable the buffer function; and the transfer timing, for both OCU ch.(0) and ch.(1).
-

■ Up/Down-count Mode (Active-High)

When Up/Down-count mode (Active-High) is selected, the following operation applies.

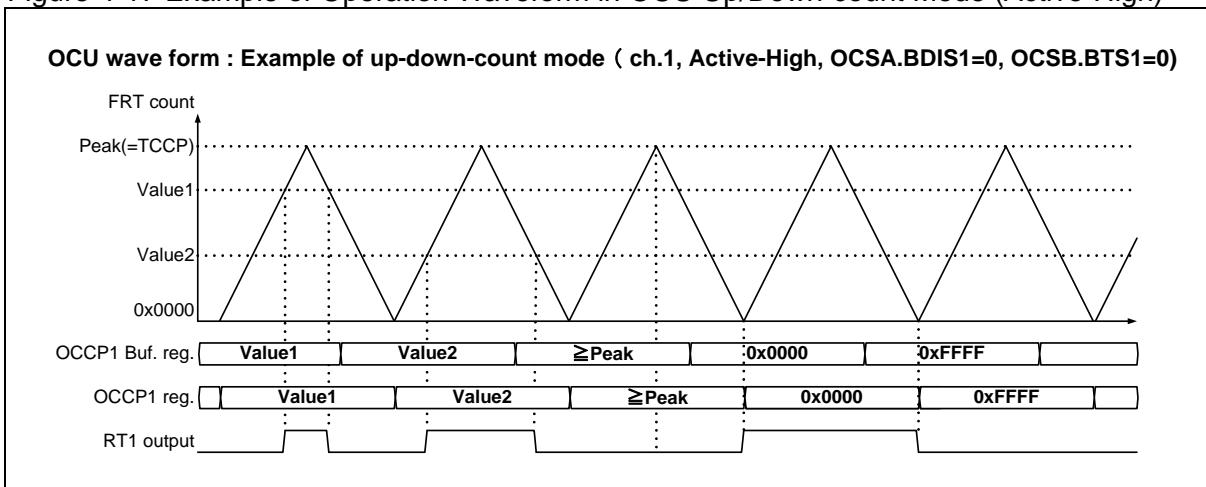
If FRT is up-counting, the output level of RT(0) signal is changed to Active level (High), when FRT's count value matches OCCP(0). If FRT is down-counting, the output level is changed to Inactive level (Low), when FRT's count value matches OCCP(0). When "0x0000" is set to OCCP(0), all-active (High) is output, and then, it returns to Inactive level (Low), when a value other than "0x0000" is set. As long as "0xFFFF" is set to OCCP(0), the Inactive level (Low) is output.

The operation of the RT(1) signal output is the same as for RT(0) according to the value in OCCP(1).

In this mode, OCU-ch.(0) and ch.(1) can operate independently from each other.

Figure 4-17 shows an example of the operation waveform when OCU-ch.1 is in Up/Down-count mode (Active-High). This figure is based on the conditions that the buffer function of the OCCP register is enabled and the Zero value transfer is selected.

Figure 4-17 Example of Operation Waveform in OCU Up/Down-count Mode (Active-High)



<Note>

Notes on Up/Down-count mode (Active-High) are as follows:

- In this mode, enable the buffer function of OCCP and select Zero value transfer for use.
- If "0x0000" is set to OCCP register when OCU's operation is enabled, the output level is changed to Active level immediately, regardless of FRT's count value.
- If a value no less than the Peak value of FRT's counter is set to OCCP, the output level does not change, even when FRT's counter value reaches its peak value. Also, the IOP0 and IOP1 registers are not set.

■ Up/Down-count Mode (Active-Low)

When Up/Down-count mode (Active- Low) is selected, the following operation applies.

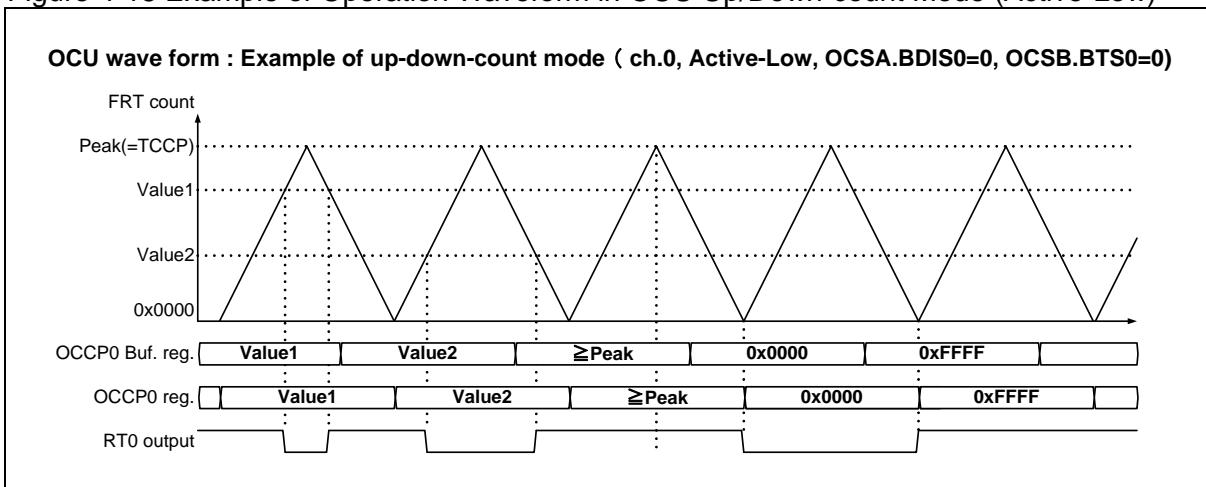
If FRT is up-counting, the output level of RT(0) signal is changed to Active level (Low), when FRT's count value matches OCCP(0). If FRT is down-counting, the output level is changed to Inactive level (High), when FRT's count value matches OCCP(0). When "0x0000" is set to OCCP(0), all-active (Low) is output, and then, it returns to Inactive level (High), when a value other than "0x0000" is set. As long as "0xFFFF" is set to OCCP(0), the Inactive level (High) is output.

The operation of the RT(1) signal output is the same as for RT(0) according to the value in OCCP(1).

In this mode, OCU-ch.(0) and ch.(1) can operate independently from each other.

Figure 4-18 shows an example of the operation waveform when OCU-ch.0 is in Up/Down-count mode (Active-Low). This figure is based on the conditions that the buffer function of the OCCP register is enabled and the Zero value transfer is selected.

Figure 4-18 Example of Operation Waveform in OCU Up/Down-count Mode (Active-Low)



<Note>

Notes on Up/Down-count mode (Active-Low) are as follows:

- In this mode, enable the buffer function of OCCP and select Zero value transfer for use.
- If "0x0000" is set to OCCP register when OCU's operation is enabled, the output level is changed to Active level immediately, regardless of FRT's count value.
- If a value no less than the Peak value of FRT's counter is set to OCCP, the output level does not change, even when FRT's counter value reaches its peak value. Also, the IOP0 and IOP1 registers are not set.

4.5. Details of WFG Output Waveform

This section provides details of the output waveform in each mode of WFG.

■ List of WFG Operation Modes

Table 4-9 shows a list of WFG operation modes, register settings and CH_GATE signal outputs.

Table 4-10 shows a list of WFG operation modes, register settings, RTO(1) and RTO(0) signal outputs.

Table 4-9 List of Details of CH_GATE Signal Outputs

Operation Mode	WFSA. TMD[2:0]	WFSA. GTEN[1:0]	CH_GATE Signal Output
Through mode	000	don't care	Always outputs Low-level signals
RT-PPG mode	001	00	Always outputs Low-level signals
		01	Outputs RT(0) without change
		10	Outputs RT(1) without change
		11	Outputs High-level signals when either RT(1) or RT(0) signal is High-level Outputs Low-level signals when both RT(1) and RT(0) signals are Low-level
Timer-PPG mode	010	00	Always outputs Low-level signals
		01	Outputs WFG timer active flag0
		10	Outputs WFG timer active flag1
		11	Outputs High-level signals when either of WFG timer active flags is "1" Outputs Low-level signals when both of WFG timer active flags are "0"
RT dead timer mode	100	don't care	Always outputs Low-level signals
PPG dead timer mode	111	00	Always outputs Low-level signals
		01	Outputs RT(0) without change
		10	Outputs RT(1) without change
		11	Outputs High-level signals when either RT(1) or RT(0) signal is High-level Outputs Low-level signals when both RT(1) and RT(0) signals are Low-level

* The CH_GATE signals in the table refer to CH10_GATE, CH32_GATE and CH54_GATE before being selected by WFSA.PSEL[1:0], as shown in the diagram of WFG-PPG connection.

Table 4-10 List of Output Details of RTO Pin

Operation Mode	WFSA. TMD [2:0]	WFSA. PGEN [1:0]	WFSA. DMOD	Output of RTO(1) Signal	Output of RTO(0) Signal
Through mode	000	00	don't care	Outputs RT(1) signal through	Outputs RT(0) signal through
		01		Outputs RT(1) signal through	Outputs CH_PPG signal through
		10		Outputs CH_PPG signal through	Outputs RT(0) signal through
		11		Outputs CH_PPG signal through	Outputs CH_PPG signal through
RT-PPG mode	001	00	don't care	Outputs RT(1) signal through	Outputs RT(0) signal through
		01		Outputs RT(1) signal through	(*A) Outputs Low-level signals when RT(0) is Low-level Outputs CH_PPG signal when RT(0) signal is High-level
		10		(*B) Outputs Low-level signals when RT(1) signal is Low-level Outputs CH_PPG signal when RT(1) signal is High-level	Outputs RT(0) signal through
		11		Same as *B	Same as *A
Timer-PPG mode	010	00	don't care	(*D) Outputs Low-level signals when WFG timer active flag1 is "0" Outputs High-level signals when WFG timer active flag1 is "1"	(*C) Outputs Low-level signals when WFG timer active flag0 is "0" Outputs High-level signals when WFG timer active flag0 is "1"
		01		Same as *D	(*E) Outputs Low-level signals when WFG timer active flag0 is "0" Outputs CH_PPG signal when WFG timer active flag0 is "1"
		10		(*F) Outputs Low-level signals when WFG timer active flag1 is "0" Outputs CH_PPG signal when WFG timer active flag1 is "1"	Same as *C
		11		Same as *F	Same as *E
RT RT dead timer mode	100	don't care	0	Starts WFG timer at the rising and falling edges of the RT(1) signal and generates the non-overlap signal. Outputs the generated non-overlap signal with normal polarity (Active High)	
			1	Starts WFG timer at the rising and falling edges of the RT(1) signal and generates the non-overlap signal. Outputs the generated non-overlap signal with reversed polarity (Active Low)	
PPG dead timer mode	111	don't care	0	Starts WFG timer at the rising and falling edges of the CH_PPG signal and generates the non-overlap signal. Outputs the generated non-overlap signal with normal polarity (Active High)	
			1	Starts WFG timer at the rising and falling edges of the CH_PPG signal and generates the non-overlap signal. Outputs the generated non-overlap signal with reversed polarity (Active Low)	

* The CH_PPG signals in the table refer to CH10_PPG, CH32_PPG and CH54_PPG selected by WFSA.PSEL[1:0], as shown in the diagram of WFG-PPG connection.

■ Through Mode

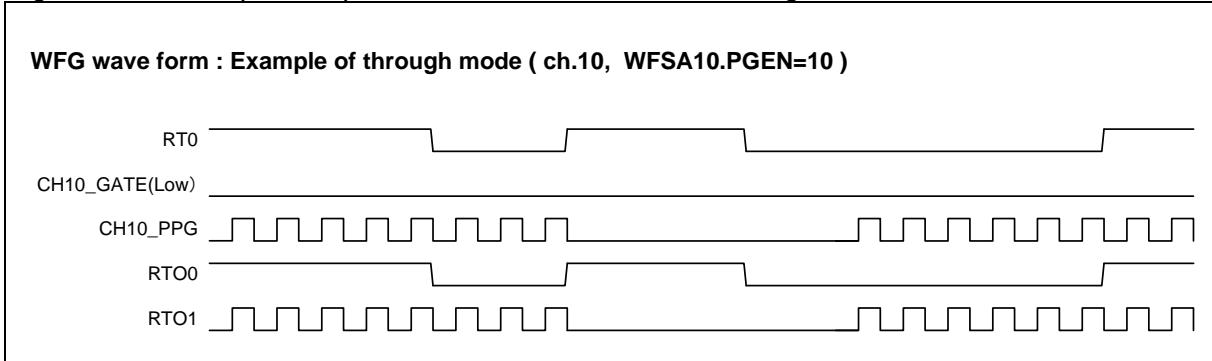
The operation in Through mode is as follows (see the List of Output Details).

The output of the CH_GATE signal is always fixed to the Low level.

The RTO(1) and RTO(0) signals output the RT(1), RT(0), and CH_PPG signals through without change by PGEN[1:0] setting.

Figure 4-19 shows an example of the operation waveform in Through mode of WFG-ch.10. In this example, the RT0 signal and the CH10_PPG signal are output through to RTO0 and RTO1, respectively (PPG timer unit can start outputting without the use of the GATE signal).

Figure 4-19 Example of Operation Waveform in WFG-Through Mode



■ RT-PPG Mode

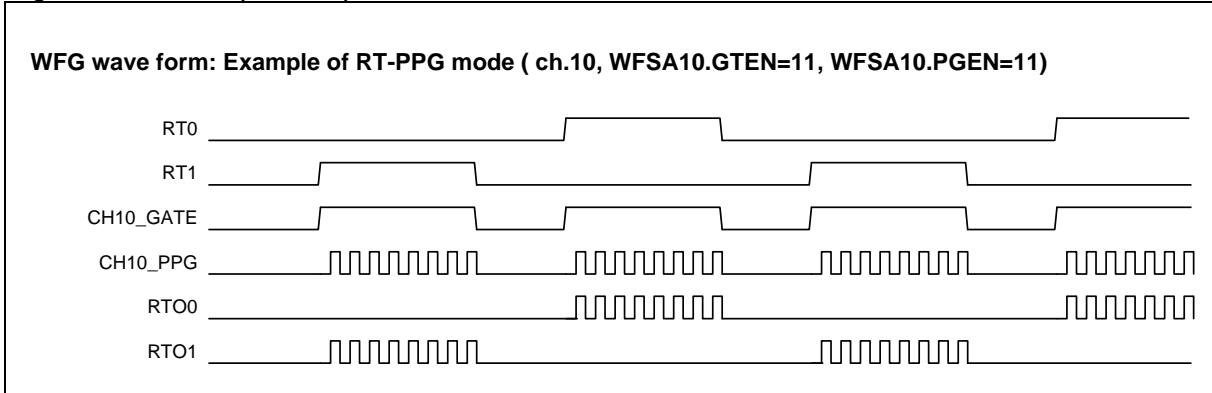
The operation in RT-PPG mode is as follows (see the List of Output Details).

The CH_GATE signal outputs the RT(1) signal, RT(0) signal or the logic OR signal of each signal by GTEN[1:0] setting.

The RTO(1) and RTO(0) signals output the RT(1) signal, RT(0) signal, CH_PPG signal, or the logic AND signal of each signal by PGEN[1:0] setting.

Figure 4-20 shows an example of the operation waveform in RT-PPG mode of WFG-ch.10. In this example, the CH0_GATE signal is generated from both RT1 and RT0 to start PPG-ch.0. The CH0_PPG signal is superimposed on RTO0 and RTO1 to output.

Figure 4-20 Example of Operation Waveform in WFG-RT-PPG Mode



■ Timer PPG Mode

The operation in Timer-PPG mode is as follows (see the List of Output Details).

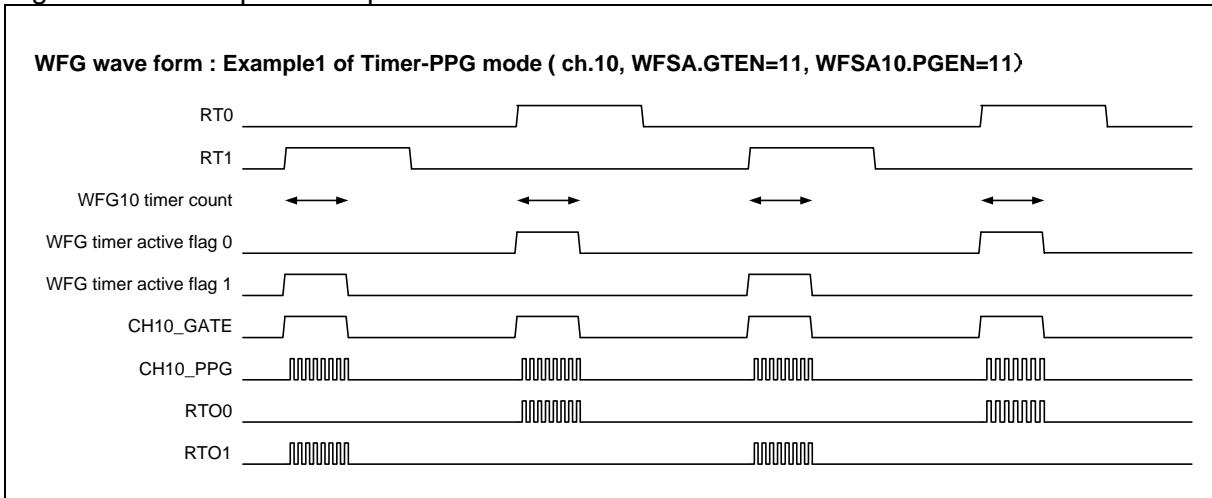
Each channel of WFG has two flags: WFG timer active flag0 and WFG timer active flag1. This mode outputs a waveform using these flags.

The CH_GATE signal outputs WFG timer active flag1, WFG timer active flag0, or the logic OR signal of each signal by GTEN[1:0] setting.

The RTO(1) and RTO(0) signals output these active flags, CH_PPG signal, or the logic AND signal of each signal by PGEN[1:0] setting.

Figure 4-21 shows Example 1 of the operation waveform in Timer PPG mode of WFG-ch.10.

Figure 4-21 Example 1 of Operation Waveform in WFG-Timer PPG Mode



The WFG timer active flags operate as follows:

WFG timer active flag0 is set to "1", when the rising edge of the RT(0) signal is detected.

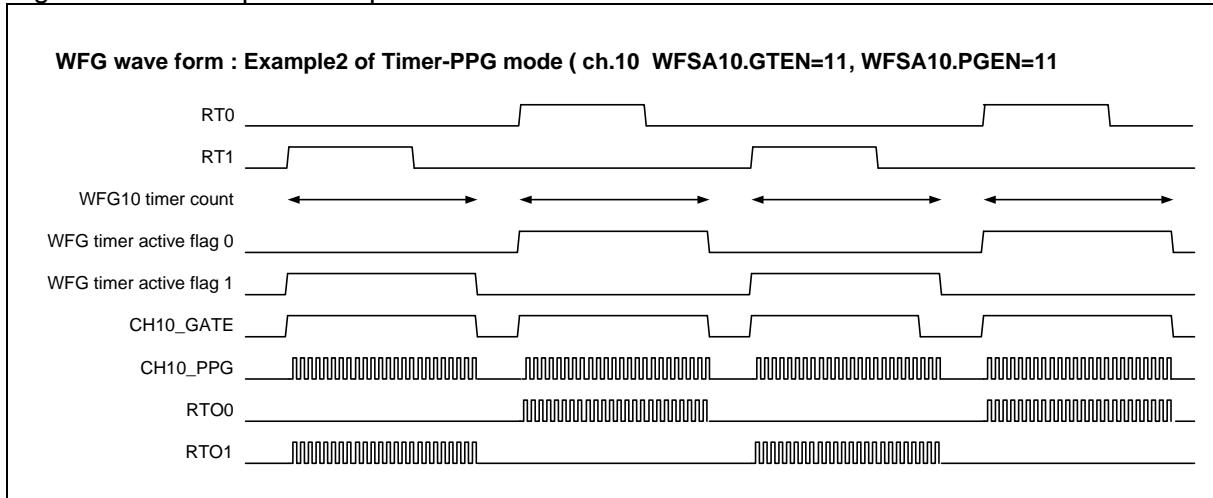
WFG timer active flag1 is set to "1", when the rising edge of the RT(1) signal is detected.

When either of the WFG timer active flags is set, the WFG timer loads the initial value from the WFTM register and starts Down-count operation. After counting, it resets both of the WFG timer active flags to "0". Therefore, irrespective of the pulse width of the RT(0) and RT(1) signals, the WFG timer active flags are set for the cycle setting time of the WFG timer from the rising edge of each signal. During this period, the CH_PPG output can be superimposed on RTO.

By the time this mode is selected by writing to the WFSA register, each WFG timer active flag is already reset. When the mode is selected, the output of the RTO(0) and RTO(1) signals is set to the Low level, regardless of the output level of the RT(0) signal, RT(1) signal and CH_PPG signal.

Figure 4-22 shows Example 2 of the operation waveform in Timer PPG mode of WFG-ch.10.

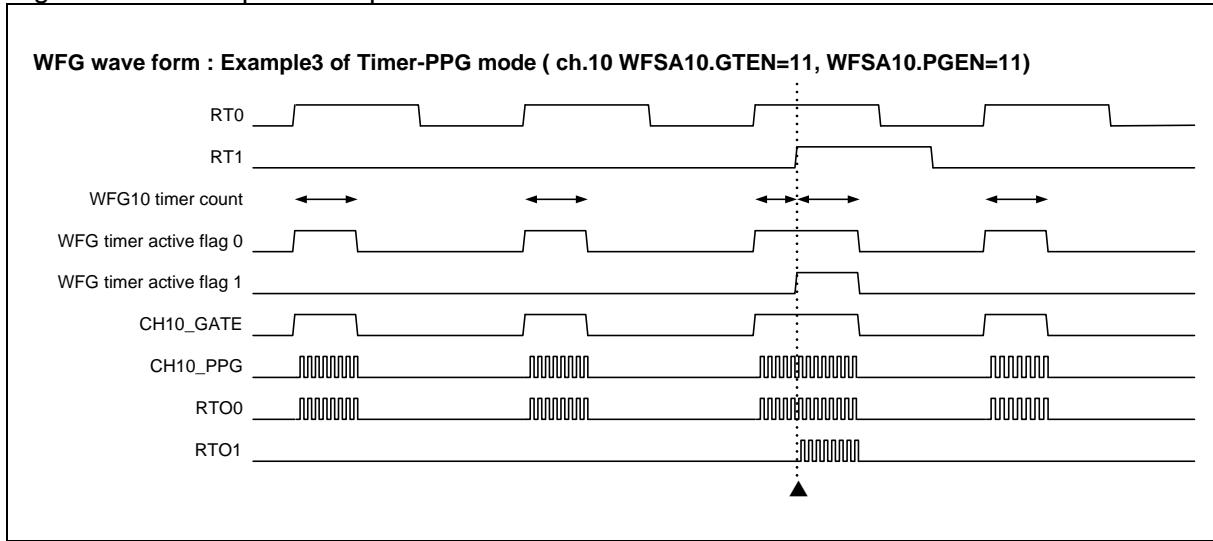
Figure 4-22 Example 2 of Operation Waveform in WFG-Timer PPG Mode



This figure shows an example of making the time setting of the WFG timer (WFTM) longer than the pulse length of RT0 and RT1. It indicates that although the same signals as in Figure 4-21 are input for the RT0 and RT1 signals, the outputs that are different from the ones shown in Figure 4-21 can be achieved because of the timer setting time.

Figure 4-23 shows Example 3 of the operation waveform in Timer PPG mode of WFG-ch.10.

Figure 4-23 Example 3 of Operation Waveform in WFG-Timer PPG Mode



This figure shows an exceptional case. The following operation is performed at the point indicated by ▲ in the figure. WFG timer active flag0 is set at the rising edge of the RT0 signal and WFG10 timer is in operation. In the meantime, the rising edge of the RT1 signal is detected and WFG timer active flag1 is set.

In this case, WFG10 timer reloads the initial value and performs the operation that will restart the timer count. Each WFG timer active flag is reset, when the counting by WFG10 timer is completed. For this reason, the period in which WFG timer active flag0 is set becomes longer than the timer setting, as shown in the figure. Therefore, the output of the waveform shown in the figure can be achieved for RTO0 and RTO1.

■ RT Dead Timer Mode

The operation in RT dead timer mode is as follows (see the List of Output Details).

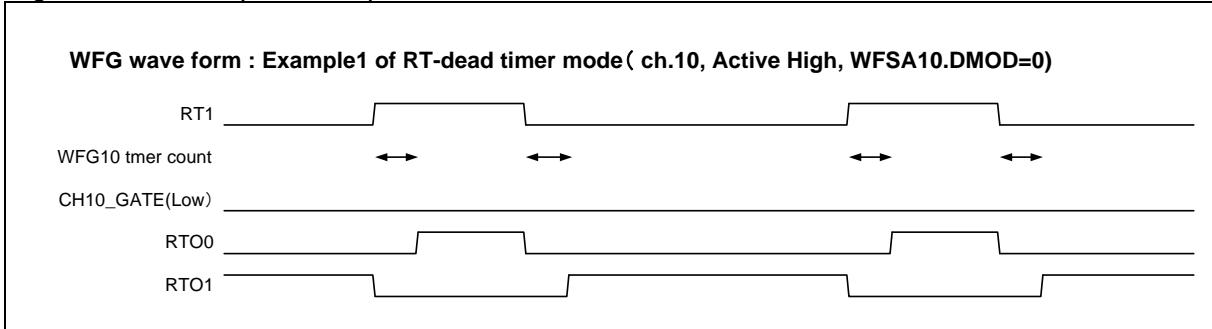
The output of the CH_GATE signal is always fixed to the Low level.

As for the RTO(1) and RTO(0) signals, the non-overlap signal that has the dead time set by WFG timer based on the RT(1) signal is output.

In this mode, the RT(0) signal and the CH_PPG signal are not used. This mode assumes that the output polarity of OCU's RT(1) output is Active High. The output polarity of RTO(0) and RTO(1) can be selected by WFSA.DMOD.

Figure 4-24 shows Example 1 of the operation waveform in RT dead timer mode of WFG-ch.10.

Figure 4-24 Example 1 of Operation Waveform in WFG-RT Dead Timer Mode



This figure shows an example of the case where normal polarity (Active High) is selected by WFSA.DMOD=0.

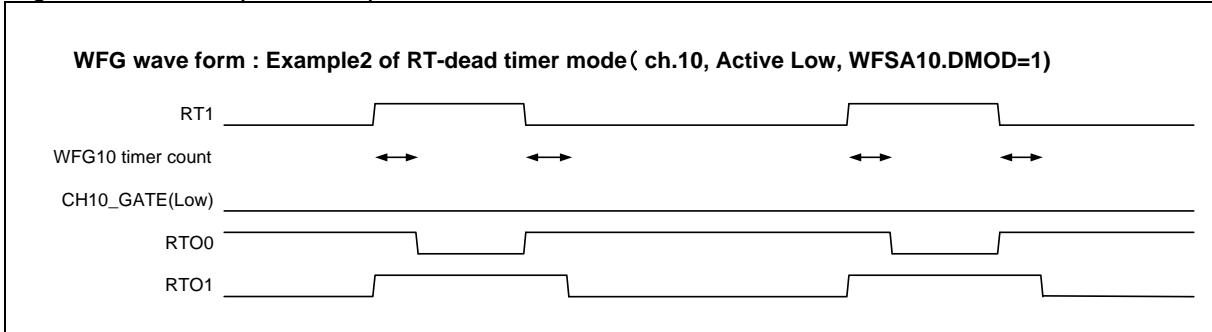
When the rising edge of the RT(1) is detected, the output of the RTO(0) signal is set to the Low level and WFG timer starts. Then, when the delay time by the WFG timer setting elapses, the RTO(0) signal is set to the High level.

When the falling edge of the RT(1) is detected, the output of the RTO(0) signal is set to the Low level and WFG timer starts. Then, when the delay time by the WFG timer setting elapses, the RTO(1) signal is set to the High level.

When this mode is selected by writing to the WFSA register, the RTO(0) signal is set to the same output level as for the RT(1) signal, while the RTO(1) signal is set to the output level that is opposite from that of the RT(1) signal.

Figure 4-25 shows Example 2 of the operation waveform in RT dead timer mode of WFG-ch.10.

Figure 4-25 Example 2 of Operation Waveform in WFG-RT Dead Timer Mode



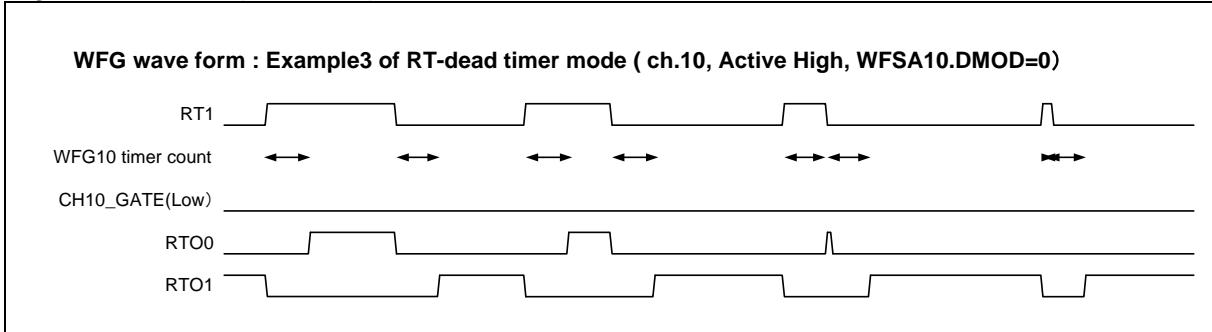
This figure shows an example of the case where reversed polarity (Active Low) is selected by WFSA.DMOD=1.

When reversed polarity (Active Low) is specified, the non-overlap signal is output with polarity reversed from the output level of the RTO(0) and RTO(1) signals.

When this mode is selected by writing to the WFSA register, the RTO(0) signal is set to the output level that is opposite from that of the RT(1) signal and the RTO(1) signal is set to the same output level as for the RT(1) signal.

Figure 4-26 shows Example 3 of the operation waveform in RT dead timer mode of WFG-ch.10.

Figure 4-26 Example 3 of Operation Waveform in WFG-RT Dead Timer Mode



This figure shows a case of normal polarity (Active High). A pulse shorter than the dead time set by the WFTM register is input to the last RT1 signal in the figure. In this case, WFG10 timer starts counting at the rising edge of the RT1 signal and loads the initial value at the next falling edge to restart the operation. Therefore, no pulse will be output to RTO0. After the falling edge of RT1, RTO1 is set to the High level when the timer setting time has elapsed.

In the case of reversed polarity (Active Low), no pulse is output to RTO(0), just like the above.

■ PPG Dead Timer Mode

The operation in PPG dead timer mode is as follows (see the List of Output Details).

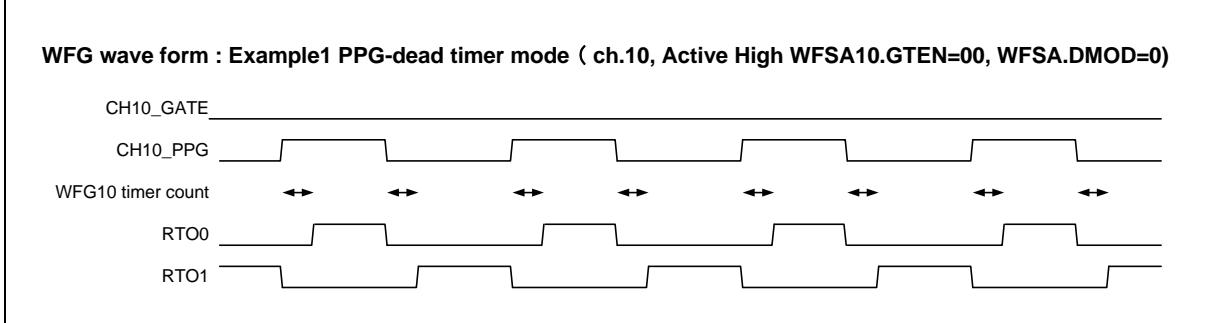
The CH_GATE signal outputs the RT(1) signal, RT(0) signal or logic OR signal by GTEN[1:0] setting.

As for the RTO(1) and RTO(0) signals, the non-overlap signal that has the dead time set by WFG timer based on the CH_PPG signal is output.

In this mode, the RT(0) and RT(1) signals are used only for the output of the CH_GATE signal.

Figure 4-27 shows Example 1 of the operation waveform in PPG dead timer mode of WFG-ch.10.

Figure 4-27 Example 1 of Operation Waveform in WFG-PPG Dead Timer Mode

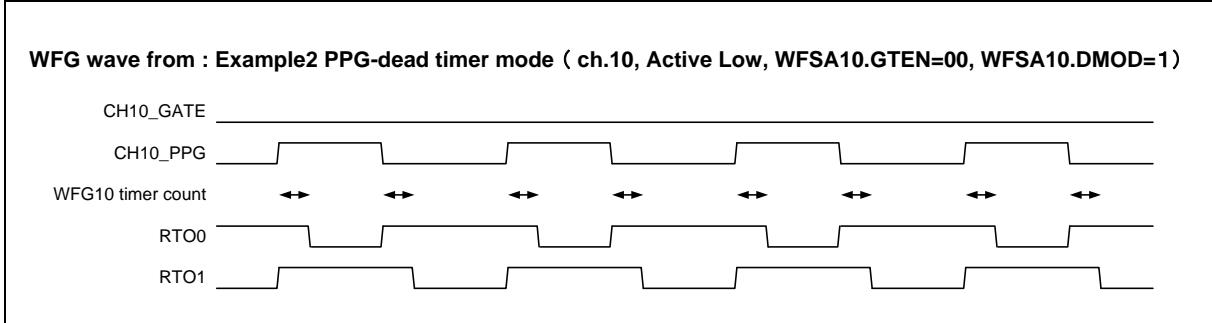


This figure shows an example of the case where WFSA.DMOD=0 and normal polarity (Active High) are selected. As shown in the figure, the WFG timer starts at the rising and falling edges of the CH_PPG signal and the non-overlap signal with the delay time set by the WFTM applied to the RTO(0) and RTO(1) signals is output.

When this mode is selected by writing to the WFSA register, the RTO(0) signal is set to the same output level as for the CH_PPG signal and the RTO(1) signal is output at the output level that is opposite from that of the CH_PPG signal.

Figure 4-28 shows Example 2 of the operation waveform in PPG dead timer mode of WFG-ch.10.

Figure 4-28 Example 2 of Operation Waveform in WFG-PPG Dead Timer Mode



This figure shows an example of the case where WFSA.DMOD=1 and reversed polarity (Activator) are selected. As shown in the figure, the non-overlap signal with the output level reversed from that of the RTO(0) and RTO(1) signals is output.

When this mode is selected by writing to the WFSA register, the RTO(0) signal is set to the output level that is opposite from that of the CH_PPG signal and the RTO(1) signal is output at the same output level as for the CH_PPG signal.

If the pulse width of the CH_PPG signal is shorter than the WFG timer, no pulse will be output to RTO(0), just like the case in Figure 4-26.

5. Other Matters

- 5.1 Connection of Model Containing Multiple MFT's
- 5.2 Treatment of Event Detect Register and Interrupt

5.1. Connection of Model Containing Multiple MFT's

This section describes the connection of models that contain multiple MFT's.

For models containing more than one multifunction timer unit, the connection of the I/O signals of the multifunction timer varies depending on the unit.

This section describes such connection differences for each multifunction timer unit.

5.1.1. Selection of FRT Connected to OCU and ICU

OCU and ICU are configured to be able to select FRT for other multifunction timer units. This section explains FRT connection between multifunction timer units and the selection method.

■ Model Containing Two MFT's

Figure 5-1 shows a diagram of FRT connected between multifunction timer units for a model containing 2 multifunction timer units.

Figure 5-1 Diagram of FRT Connected between Multifunction Timer Units
(For Model Containing 2 Multifunction Timer Units)

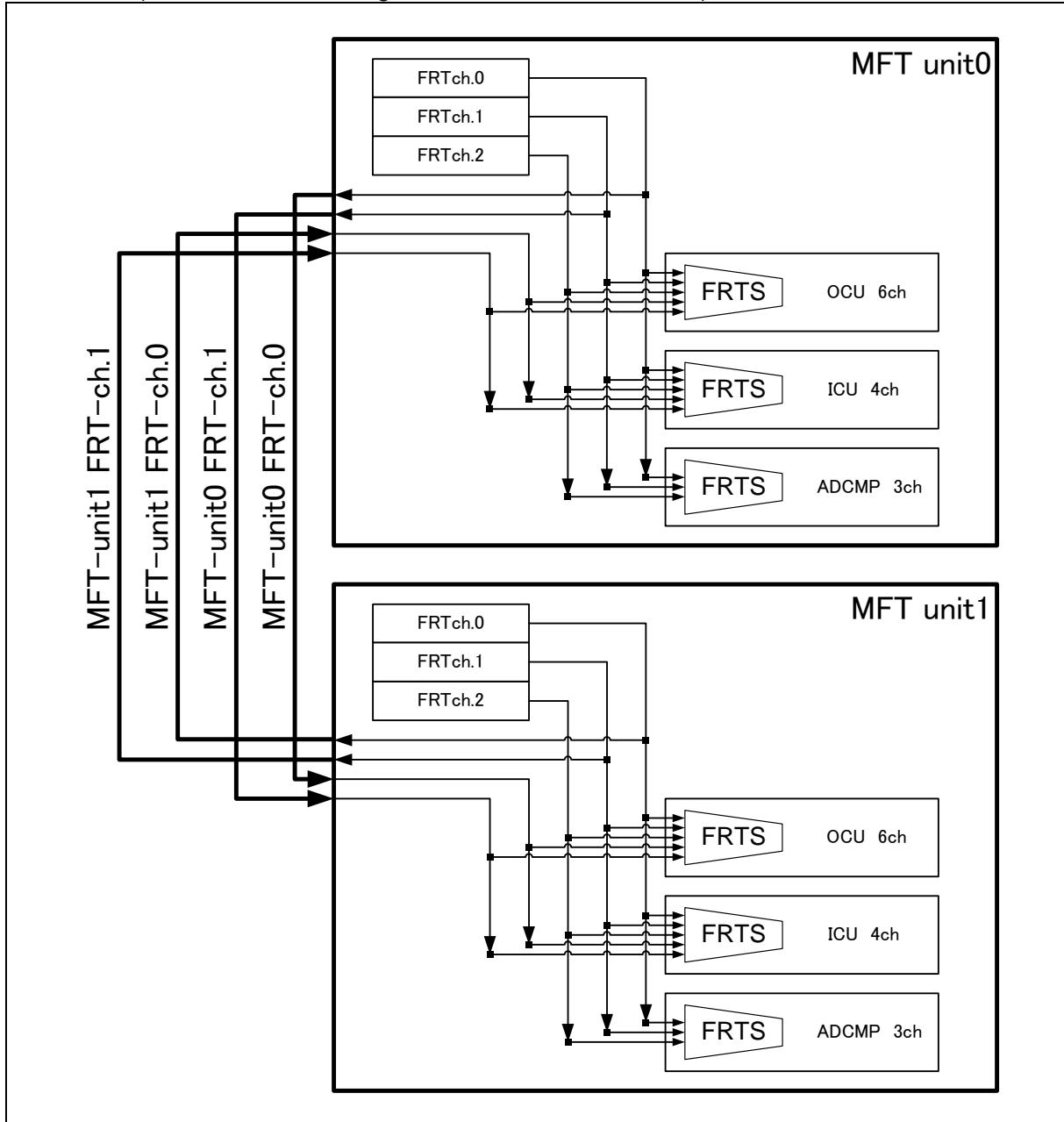


Table 5-1 shows the register settings of OCFS and ICFS of MFT-unit0 and where they are connected.

**Table 5-1 OCFS and ICFS Register Settings for MFT-unit0
(For Model Containing 2 Multifunction Timer Units)**

Register Name	Setting	Function
OCFS	FSO0[3:0]	0011 Connects FRT-ch.0 of MFT unit1 to OCU ch.(0).
		0100 Connects FRT-ch.1 of MFT unit1 to OCU ch.(0).
	FSO1[3:0]	0011 Connects FRT-ch.0 of MFT unit1 to OCU ch.(1).
		0100 Connects FRT-ch.1 of MFT unit1 to OCU ch.(1).
ICFS	FSI0[3:0]	0011 Connects FRT-ch.0 of MFT unit1 to ICU ch.(0).
		0100 Connects FRT-ch.1 of MFT unit1 to ICU ch.(0).
	FSI1[3:0]	0011 Connects FRT-ch.0 of MFT unit1 to ICU ch.(1).
		0100 Connects FRT-ch.1 of MFT unit1 to ICU ch.(1).

Table 5-2 shows the register settings of OCFS and ICFS of MFT-unit1 and where they are connected.

**Table 5-2 OCFS and ICFS Register Settings for MFT-unit1
(For Model Containing 2 Multifunction Timer Units)**

Register Name	Setting	Function
OCFS	FSO0[3:0]	0011 Connects FRT-ch.0 of MFT unit0 to OCU ch.(0).
		0100 Connects FRT-ch.1 of MFT unit0 to OCU ch.(0).
	FSO1[3:0]	0011 Connects FRT-ch.0 of MFT unit0 to OCU ch.(1).
		0100 Connects FRT-ch.1 of MFT unit0 to OCU ch.(1).
ICFS	FSI0[3:0]	0011 Connects FRT-ch.0 of MFT unit0 to ICU ch.(0).
		0100 Connects FRT-ch.1 of MFT unit0 to ICU ch.(0).
	FSI1[3:0]	0011 Connects FRT-ch.0 of MFT unit0 to ICU ch.(1).
		0100 Connects FRT-ch.1 of MFT unit0 to ICU ch.(1).

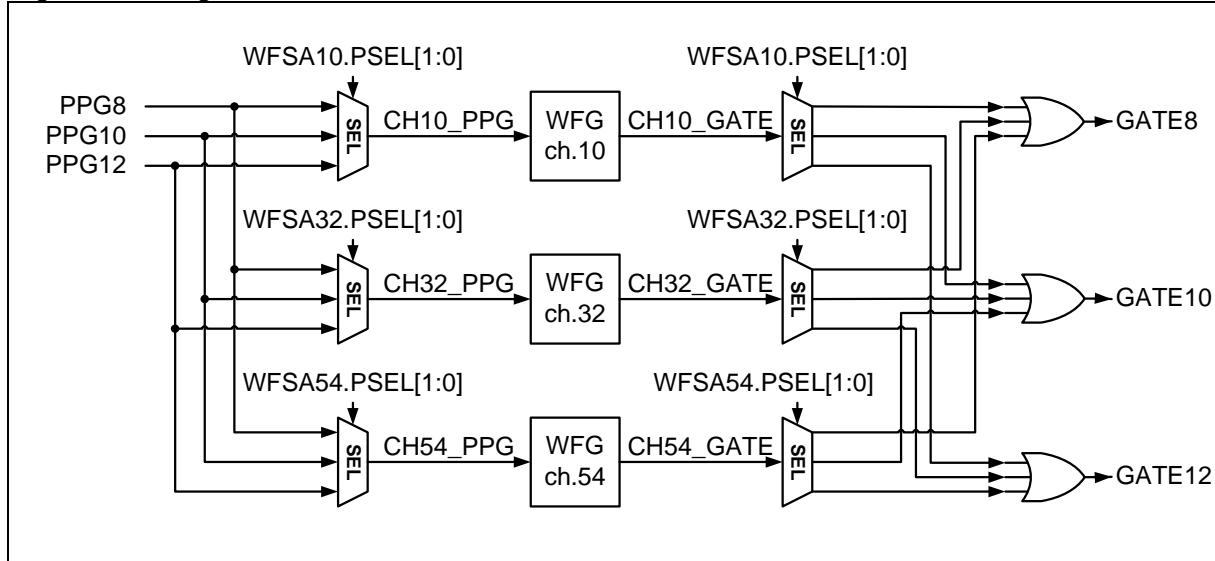
5.1.2. PPG Timer Unit Connected to WFG

The PPG timer unit to be connected to WFG varies depending on the multifunction timer unit used. This section explains the connection of the PPG timer unit and the selection method.

■ MFT-unit1

PPG timer unit ch.8, ch.10 and ch.12 are connected to WFG of MFTunit1, as shown in Figure 5-2.

Figure 5-2 Diagram of WFG-PPG Connection at MFTunit1



In case of WFG in MFTunit1, the following is selected by the setting of the WFSA.PSEL[1:0] register.

[bit9:8] WFSA.PSEL[1:0]

Process	Value	Function
Write	00	Sets the output destination of the GATE signal to PPG timer unit ch.8. Sets the input source of the PPG signal to PPG timer unit ch.8.
	01	Sets the output destination of the GATE signal to PPG timer unit ch.10. Sets the input source of the PPG signal to PPG timer unit ch.10.
	10	Sets the output destination of the GATE signal to PPG timer unit ch.12. Sets the input source of the PPG signal to PPG timer unit ch.12.
	11	Setting prohibited
Read	-	Reads the register setting.

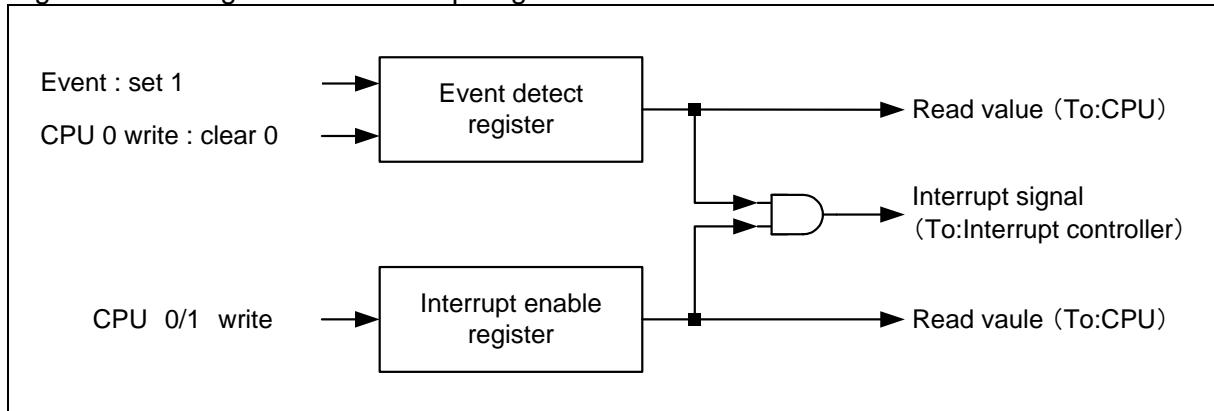
5.2. Treatment of Event Detect Register and Interrupt

This section provides notes on the event detect register in the multifunction timer unit, the operation and control of interrupt-related circuits.

■ Configuration of Circuit

Figure 5-3 shows the configuration of the interrupt signal generator.

Figure 5-3 Configuration of Interrupt Signal Generator



- Event detect register

Each function block has an event detect register to notify CPU that a specific event (e.g. detection of the rising edge of the input signal at ICU) has occurred. This register indicates "0" when the relevant event has not occurred. It is set to "1", when the event occurs.

- Interrupt enable register

There is an interrupt enable register to specify whether or not to notify CPU of the above event as an interrupt. As shown in the figure, the logic AND of the values in the event detect register and the interrupt enable register is connected to the interrupt controller (NVIC) as an interrupt signal.

- Writing to and reading from each register

The event detect register can be read from CPU at any time, regardless of the value in the interrupt enable register.

It can be cleared by writing "0", but cannot be set by writing "1". The interrupt enable register allows any value to be set from CPU and can be read.

■ Circuit Operation

- Operation when the interrupt enable register is set to "0" (interrupt disabled)

Even when an event occurs and "1" is set to the event detect register, no interrupt occurs. In this case, the occurrence of the event can be recognized by reading from the event detect register regularly via CPU.

- Operation when the interrupt enable register is set to "1" (interrupt enabled)

When an event occurs and "1" is set to the event detect register, the interrupt signal is asserted and an interrupt occurs. CPU can recognize the occurrence of the event by the interrupt.

■ Clearing Event Detect Register

Generally, the event detect register cannot be cleared automatically. In order to recognize the occurrence of the next event after "1" is set to the event detect register, the event detect register must be cleared via CPU beforehand. If it is not cleared via CPU, CPU cannot recognize the occurrence of the succeeding events.

■ Returning from Interrupt Processing

When an interrupt is processed using an interrupt signal, it is necessary to clear the event detect register when returning from the interrupt processing, deassert the interrupt signal, and then return from the interrupt. Returning from an interrupt without deasserting the interrupt signal will result in the same interrupt process taking place again with no way out of that process.

■ Value Written to Event Detect Register

The write value and read value of the event detect register have the following meanings:

- Writing "0" : Clears the register.
- Writing "1" : Does nothing.
- Reading "0" : No event occurred.
- Reading "1" : Event occurred.

Because the event detect register is in the configuration described above, when a value is read from the event detect register via CPU, the value can not be normally written back. This is due to the following reason.

When "0" is successfully read from the event detect register at a certain point, it indicates that the event has yet to occur at that point. Next, writing the value back to the event detect register without change (i.e. writing "0") means instructing the event detect register to be cleared. If an event occurs during the period from the reading via CPU to the writing the value back, the register will be cleared, preventing that event from being recognized.

For the above reason, when writing to the event detect register, "1" must be always written (i.e. doing nothing), unless the register is intended to be cleared. An example is provided below.

The ICSA10 register is in the following configuration based on the 8-bit register.

Bit	7	6	5	4	3	2	1	0
Field	ICP1	ICP0	ICE1	ICE0	EG1[1:0]		EG0[1:0]	

The ICP1 and ICP0 registers are event detect registers that notify CPU of an event upon edge detection at ICU-ch1 and ICU-ch.0, respectively.

If "01111111" is read from these registers at a certain point, for example, it indicates that a valid edge is detected (ICP0=1) at ch.0 and no valid edge is detected (ICP1=0) at ch.1.

Then, write "0" back to bit6 in order to clear the ICP0 register. At that point, it is not possible to set the value in the ICP0 register to "0" and write "00111111" back due to the reason explained above. It is because information about any possible detection of an event at ch.1 will be cleared during the period from reading from the register to writing the value back.

Therefore, in order to clear the ICP0 register, it is necessary to write "10111111" back with bit6=0 and bit7=1.

■ Read Value Mask Function at RMW Access

Since the above procedure is complicated, a masking function is provided to mask the read value of the event detect register to "1" at RMW access for the value to be written back.

In this model, RMW access occurs, when write access is made to the bit-banding alias area.

Write access to the bit-banding alias area is the RMW access used to read all of the register bits in the address area where the target bit exists, rewrite only the target bit and write all the register bits back.

In the example of the ICSA10 register provided earlier, assume that the value "01111111" is read at a certain point.

To write "0" to bit6 so that the ICP0 register will be cleared, write access to the normal address area requires bit7=1 and bit6=0 to be written. However, if "0" is written to bit6 by write access to the bit-banding alias area, the hardware performs the following operation:

- It read the value in the ICSA10 register.
- At this point, the ICP1 and ICP0 registers return a read value masked to "1" because of the RMW access. In other words, the value to be read is "11111111".
- Write to the ICSA10 register the value "10111111", where only the value of bit6(ICP0) has been replaced with "0".

bit7 can not be cleared because the device operates as described above. How to write back the value of bit6 is described in this example. In case of writing back the values of bit7 and bit5 to bit0, the read values of bit7 and bit6 are masked to "1" also; therefore, it is unnecessary to consider the writing back value. For this reason, this configuration allows rewriting the register without considering the writing back value to the event detection register in case of writing access to the bit-banding alias area.

* Read access to the bit-banding alias area is not RMW access; therefore the value of the register is unmasked when reading.

■ List of Event Detect Registers and Interrupt Enable Registers

Table 5-3 shows a list of the event detect registers and interrupt enable registers that exist in the multifunction timer unit as well as their interrupt signals.

Table 5-3 List of Event Detect Registers and Interrupt Enable Registers

Block Name	Target Event	Event Detect Register	Interrupt Enable Register	Name of Interrupt Signal
FRT ch.0	Detection of FRT0 == 0x0000	TCSA0.IRQZF	TCSA0.IRQZE	Zero value detection interrupt
FRT ch.1	Detection of FRT1 == 0x0000	TCSA1.IRQZF	TCSA1.IRQZE	Zero value detection interrupt
FRT ch.2	Detection of FRT2 == 0x0000	TCSA2.IRQZF	TCSA2.IRQZE	Zero value detection interrupt
FRT ch.0	Detection of FRT0 == TCCP0	TCSA0.ICLR	TCSA0.ICRE	Peak value detection interrupt
FRT ch.1	Detection of FRT1 == TCCP1	TCSA1.ICLR	TCSA1.ICRE	Peak value detection interrupt
FRT ch.2	Detection of FRT2 == TCCP2	TCSA2.ICLR	TCSA2.ICRE	Peak value detection interrupt
OCU ch.0	Detection of FRT == OCCP0	OCSA10.IOP0	OCSA10.IOE0	Match detection interrupt
OCU ch.1	Detection of FRT == OCCP1	OCSA10.IOP1	OCSA10.IOE1	Match detection interrupt
OCU ch.2	Detection of FRT == OCCP2	OCSA32.IOP0	OCSA32.IOE0	Match detection interrupt
OCU ch.3	Detection of FRT == OCCP3	OCSA32.IOP1	OCSA32.IOE1	Match detection interrupt
OCU ch.4	Detection of FRT == OCCP4	OCSA54.IOP0	OCSA54.IOE0	Match detection interrupt
OCU ch.5	Detection of FRT == OCCP5	OCSA54.IOP1	OCSA54.IOE1	Match detection interrupt
ICU ch.0	Detection of valid edge	ICSA10.ICP0	ICSA10.ICE0	Input signal edge detection interrupt
ICU ch.1	Detection of valid edge	ICSA10.ICP1	ICSA10.ICE1	Input signal edge detection interrupt
ICU ch.2	Detection of valid edge	ICSA32.ICP0	ICSA32.ICE0	Input signal edge detection interrupt
ICU ch.3	Detection of valid edge	ICSA32.ICP1	ICSA32.ICE1	Input signal edge detection interrupt

The interrupts shown in Table 5-4 below do not have an interrupt enable register, as they are dedicated to interrupts (i.e. polling not assumed). If "1" is set to the interrupt flag when the target event occurs, an interrupt occurs.

Table 5-4 List of Interrupt Flag Registers and Interrupt Enable Registers

Block Name	Target Event	Interrupt Flag Register	Interrupt Enable Register	Name of Interrupt Signal
NZCL	Input of emergency motor shutdown signal	WFIR.DTIF	None	DTIF interrupt
WFG ch.10	Completion of WFG10 timer count	WFIR.TMIF10	None	WFG10 timer interrupt
WFG ch.32	Completion of WFG32 timer count	WFIR.TMIF32	None	WFG32 timer interrupt
WFG ch.54	Completion of WFG54 timer count	WFIR.TMIF54	None	WFG54 timer interrupt

Chapter: PPG Configuration

This chapter explains PPG configuration.

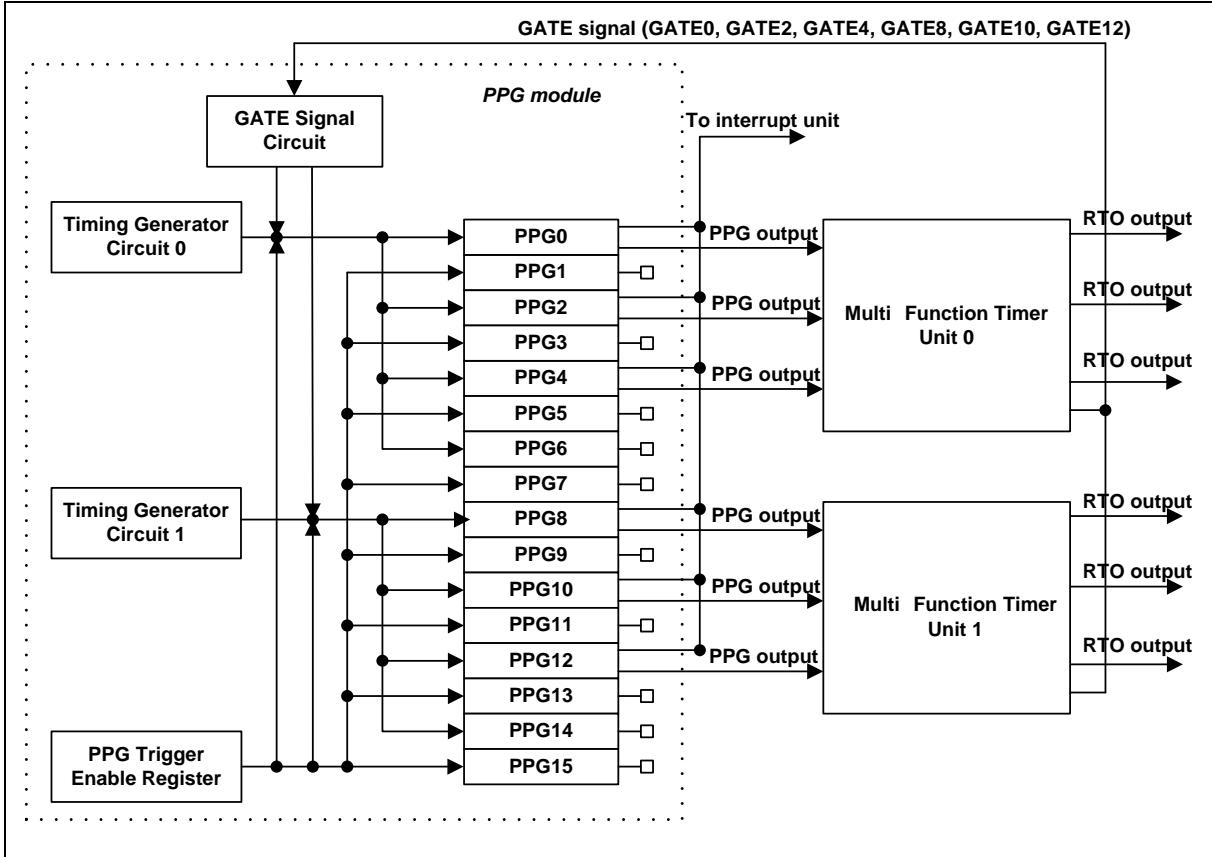
1. Configuration

CODE: 9BFPPGTOP-E01.2

1. Configuration

This section explains the unique PPG configuration.

The following shows the unique PPG configuration.



● PPG connection

- PPG output is transferred from the output RTO pin of the multifunction timer via the multifunction timer module.
- PPG output and PPG interrupt are connected only to the PPG0, PPG2, PPG4, PPG8, PPG10, and PPG12. Therefore, no output is obtained from other PPG channels.
- A PPG start factor can be set to a PPG channel with no output connected, but no output is obtained from such a PPG channel.
- Furthermore, any PPG operation mode (8-bit, 8+8-bit, 16-bit, or 16+16-bit mode) can be selected, but no output is obtained from a PPG channel with no output connected.

● Differences between timing generators 0 and 1

- Timing generator 0
 - Compare Register : COMP0/COMP2/COMP4/COMP6
 - PPG channel to be triggered : ch.0/ch.2/ch.4/ch.6
- Timing generator 1
 - Compare Register : COMP1/COMP3/COMP5/COMP7
 - PPG channel to be triggered : ch.8/ch.10/ch.12/ch.14

● Combinations of operation modes and PPG channels with output enabled

PPG channel	8-bit mode	8+8-bit mode	16-bit mode	16+16-bit mode
PPG ch.0	PPG0 output	PPG0 output	PPG0 output	PPG0 output
PPG ch.1	Not available	PPG0 prescaler		
PPG ch.2	PPG2 output	PPG2 output	PPG2 output	PPG0 prescaler
PPG ch.3	Not available	PPG2 prescaler		
PPG ch.4	PPG4 output	PPG4 output	PPG4 output	PPG4 output
PPG ch.5	Not available	PPG4 prescaler		
PPG ch.6	Not available	Not available	Not available	PPG4 prescaler
PPG ch.7	Not available	Not available		
PPG ch.8	PPG8 output	PPG8 output	PPG8 output	PPG8 output
PPG ch.9	Not available	PPG8 prescaler		
PPG ch.10	PPG10 output	PPG10 output	PPG10 output	PPG8 prescaler
PPG ch.11	Not available	PPG10 prescaler		
PPG ch.12	PPG12 output	PPG12 output	PPG12 output	PPG12 output
PPG ch.13	Not available	PPG12 prescaler		
PPG ch.14	Not available	Not available	Not available	PPG12 prescaler
PPG ch.15	Not available	Not available		

● Setting the EDGE bit in the PPG GATE Function Control Register

The EDGE bit in the PPG GATE Function Control Register can be set only to "EDGE=0".

* : Started only at the rising edge of the GATE signal.

Chapter: PPG

This chapter describes the PPG timer.

1. Overview
2. Configuration and Block Diagrams
3. Operations
4. Setup Procedure Example
5. Registers
6. Notes

1. Overview

This section describes the overview of PPG timer.

The Programmable Pulse Generator (PPG) module can perform pulse output of arbitrary cycle and duty ratio controlled by timer operation.

● Features of PPG module

- 8-bit PPG output 2-channel independent operating mode is supported.
- 16-bit PPG output operating mode is supported.
- 8+8-bit PPG output operating mode is supported.
- 16+16-bit PPG output operating mode is supported.
- PPG can be inverted the output level, including the initial output level during PPG stop.
- PPG can be selected arbitrary cycle by selecting PPG count clock.
- PPG can output the pulses with arbitrary duty ratio by register setting.
This module can also be used in conjunction with an external circuit to form a D/A converter.
- When the reload value count is ended and an underflow occurs, PPG is activated. This activation occurs if an interrupt is enabled by the Control Register (PPGC Register).

● PPG start trigger conditions

PPG start trigger can be selected from following three conditions.

- Start triggered by register writing
- Start triggered by the Timing Generator Circuit
- Start triggered by GATE signal from the multifunction timer

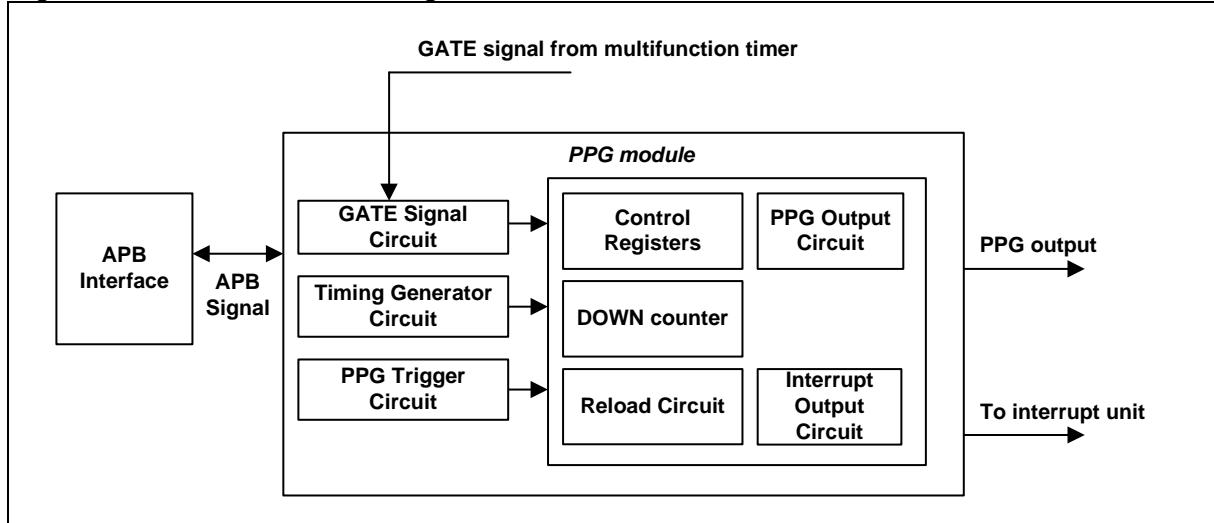
2. Configuration and Block Diagrams

This section shows the configuration and block diagrams of PPG timer.

■ PPG circuit block diagram

Figure 2-1 shows the block diagram of the PPG circuit.

Figure 2-1 PPG circuit block diagram



The PPG module consists of the following functional blocks.

- GATE Signal Circuit
This circuit sets the start or stop signal to PPG module when receiving a GATE signal from the multifunction timer.
- Timing Generator Circuit
This is a PPG start timing generator circuit. This circuit generates a PPG start timing signal, using its built-in compare register. Simultaneous startup or a delayed startup of PPG is allowed by the compare register setting. The details are given in the following sections.
- PPG trigger circuit
This is a PPG trigger circuit containing PPG start register. This circuit generates a start trigger signal when the register is enabled. Up to eight channels can be started simultaneously (in 16-bit mode).
- Control registers
These registers contain the setting registers of count clock cycle and PPG operation mode.
- DOWN Counter
This is a down counter for PPG pulse generation. This counter loads the value of reload register and generates output pulses.
- Reload Circuit
This circuit sets the LOW width and HIGH width of output pulse to be reloaded into the down counter. This circuit inverts the pin output level at reloading.
- PPG Output Circuit
This circuit outputs PPG pulse. An output level at PPG STOP can be set by the register setting.
- Interrupt Output Circuit
This circuit outputs a preset interrupt signal.

■ Timing generator block diagram

Figure 2-2 and Figure 2-3 show the block diagram of timing generator circuits 0 and 1, respectively.

Figure 2-2 Block diagram of timing generator circuit 0

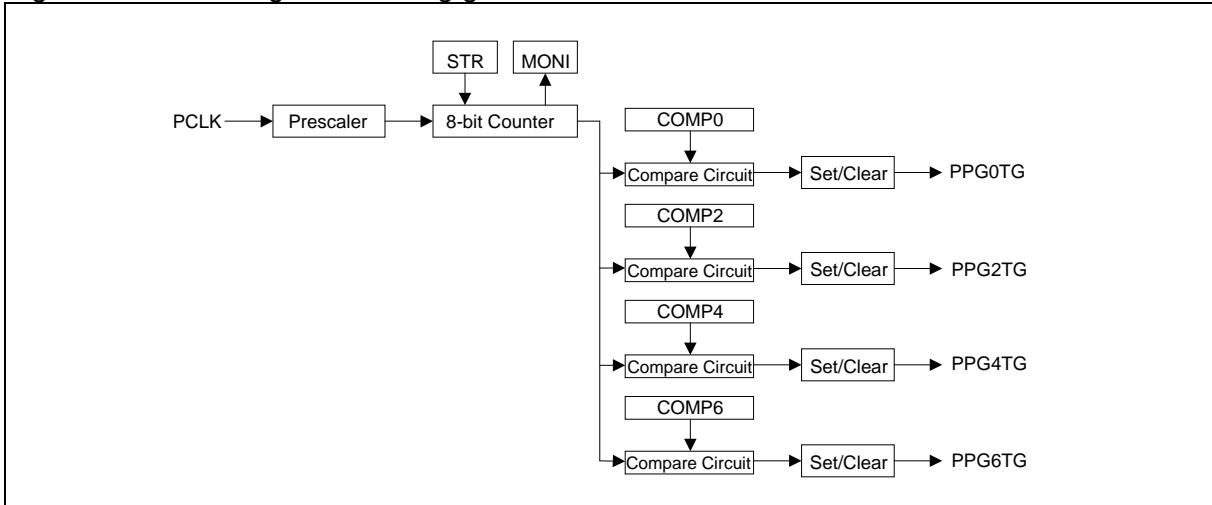
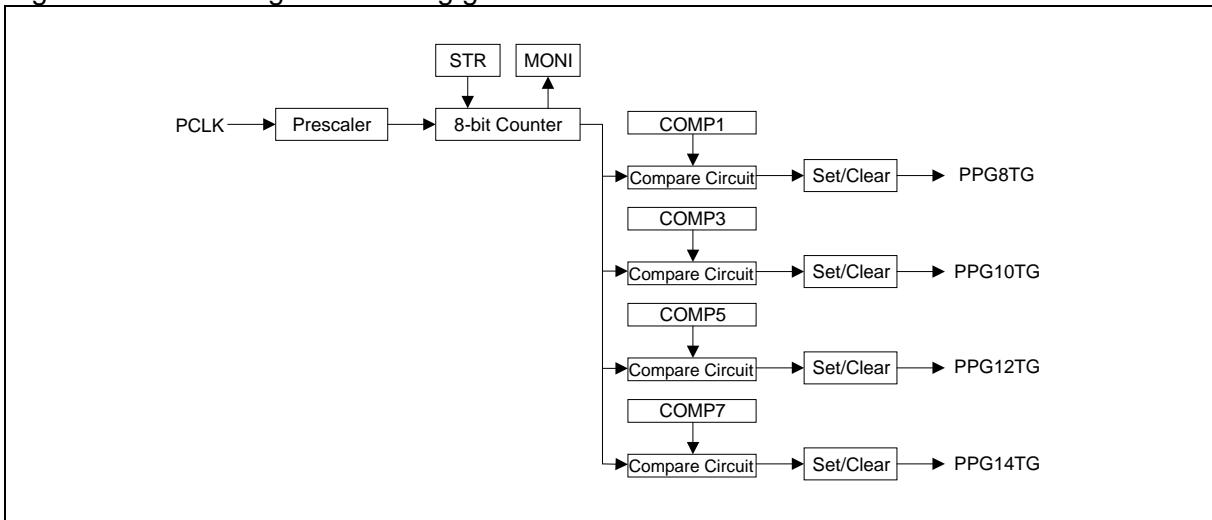


Figure 2-3 Block diagram of timing generator circuit 1



In Figure 2-2 and Figure 2-3, "PPGxTG"(x=0,2,4,6,8,10,12,14) is a PPG start trigger signal.

The Timing Generator Circuit consists of the following functional blocks.

- Prescaler
This circuit sets the divided clock of PCLK (the APB bus clock) for the counter clock.
- 8-bit Counter
This is an UP Counter. The count clock is used from the prescaler.
- Compare circuit
This circuit compares an 8-bit counter value with the COMP register value, and generates a PPG start trigger signal.
- COMP Register
This register sets a generation timing of each PPG start trigger signal.

3. Operations

This section shows the operation of PPG timer.

- 3.1 PPG circuit operations
- 3.2 Timing generator circuit operations

3.1. PPG circuit operations

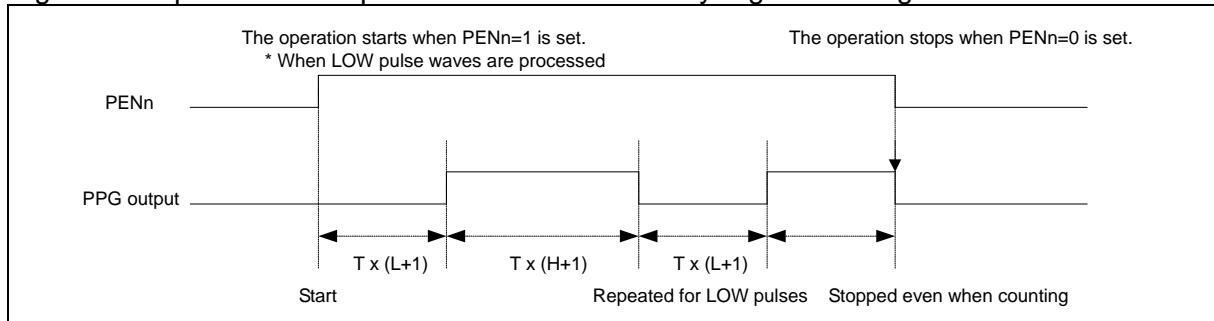
The PPG module can output pulse signals having arbitrary cycle and duty ratio. The pulse output can be controlled based on the timer operation.

■ PPG functions

The PPG timer starts decrementing the down counter value being set in the LOW width reload register by triggered. During this time, the output is set to LOW. After an underflow occurs, the output is inverted to HIGH. Then, the PPG timer starts decrementing the value being set in the HIGH width reload register. When an underflow occurs, the output is inverted to LOW again. Thus, PPG can generate pulses having arbitrary LOW and HIGH pulse width.

Figure 3-1 shows an operation example of 8/16bit-PPG operating mode when PPG is started by register writing.

Figure 3-1 Operation example when PPG is started by register writing



Explanation of symbols used in Figure 3-1

	L : PRLL value
P1 = T x (L+1)	H : PRLH value
	T : Input clock cycle
Ph = T x (H+1)	P1 : LOW pulse width
	Ph : HIGH pulse width

1. When PENn Register is set to "1", PPG output starts.
2. The 8/16-bit PPG timer has two reload registers (PRLL/PRLH) for each channel. The values set in the reload registers are reloaded alternately into the down counters.
3. After LOW level pulses are output during $T \times (L+1)$ counting, HIGH level pulses are output during $T \times (H+1)$ counting. After $T \times (H+1)$ counting are finished, $T \times (L+1)$ counting start again and LOW level pulses output again. This output repeats until the PENn Register is set to "0".
4. The operation stops if PENn Register is set to "0".
The pulse output is stopped even when pulses are counted.

■ Relation between reload value and pulse width

The pulse width to be output is the value that multiplies the cycle of the count clock by the value in the reload register plus 1. Note that the pulse width will be one cycle of the count clock when the reload register value is set to "00H" at operating the 8-bit PPG and when the reload register value is set to "0000H" at operating the 16-bit PPG. Note that the pulse width will be 256 cycles of the count clock when the reload register value is set to "FFH" at operating the 8-bit PPG and the pulse width will be 65536 cycles of the count clock when the reload register value is set to "FFFFH" at operating the 16-bit PPG.

■ Interrupts

If an interrupt is enabled by the Control Register (PPGC Register) setting, an interrupt occurs when the reload value is counted out and an underflow occurs.

The interrupt occurs when LOW pulse ends and HIGH pulse ends.

However, if the PPGC.INTM bit is set to "1", an interrupt only occurs when HIGH width pulse ends.

■ Explanation of operation modes

- 8-bit PPG operation mode
One channel can operate as 8-bit PPG independently.
- 16-bit PPG operation mode
Two channels are combined, and the combined channel operates as 16-bit PPG.
- 8+8-bit PPG operation mode
In this mode, it needs two PPGs. One channel operates as an 8-bit prescaler. Another channel operates as 8-bit PPG timer. The borrow output of prescaler is counted, and allows 8-bit PPG pulse in any cycle.
- 16+16-bit PPG operation mode
In this mode, it needs totally four PPGs. Two channels are combined, and combined channel operates as 16-bit prescaler. The other two channels are combined, and combined channel operates as 16-bit PPG timer. In this mode, the borrow output of the prescaler is counted, and allows 16-bit PPG pulse in any cycle.

■ Relation between PPG channels and operation modes

The PPG has an 8-bit Counter for each channel. In 16-bit operation mode, two channels are combined and a 16-bit PPG is operated. The combination of 16-bit operation modes and PPG channel concatenation is defined on Table 3-1.

Table 3-1 Combination of operation modes and PPG channel concatenation

PPG channel	8-bit mode	8+8-bit mode	16-bit mode	16+16-bit mode
PPG0	PPG0	PPG0	PPG0	PPG0
PPG1	PPG1	PPG0 prescaler		PPG0 prescaler
PPG2	PPG2	PPG2	PPG2	PPG0 prescaler
PPG3	PPG3	PPG2 prescaler		PPG4
PPG4	PPG4	PPG4	PPG4	PPG4
PPG5	PPG5	PPG4 prescaler		PPG4 prescaler
PPG6	PPG6	PPG6	PPG6	PPG8
PPG7	PPG7	PPG6 prescaler		PPG8 prescaler
PPG8	PPG8	PPG8	PPG8	PPG8
PPG9	PPG9	PPG8 prescaler		PPG8 prescaler
PPG10	PPG10	PPG10	PPG10	PPG12
PPG11	PPG11	PPG10 prescaler		PPG12 prescaler
PPG12	PPG12	PPG12	PPG12	PPG14
PPG13	PPG13	PPG12 prescaler		PPG14 prescaler
PPG14	PPG14	PPG14	PPG14	PPG14
PPG15	PPG15	PPG14 prescaler		PPG14 prescaler

■ Selecting the count clock

The count clocks to be used are PCLK clocks, and can be selected from one of the following four types of count clock inputs. The count clock operates as shown on Table 3-2.

Table 3-2 Count clock selection

PCS1	PCS0	Count clock operation
0	0	The Count Clock is counted for PCLK cycle.
0	1	The Count Clock is counted for 4 cycles of PCLK.
1	0	The Count Clock is counted for 16 cycles of PCLK.
1	1	The Count Clock is counted for 64 cycles of PCLK.

Note: The period of the initial count may vary if the PPG is started when the prescaler is running and the PPG is halted in 8+8-bit PPG operation mode and 16+16-bit PPG operation mode.

■ Control of pulse output

When the PPG is halted, the LOW level pulse is output at default setting.

The signal level inverting, including the initial signal level, can be specified by the REVC register.

■ PPG startup conditions

The following three PPG start trigger modes can be selected.

- Start triggered by the Timing Generator Circuit
- Start triggered by GATE signal from the multifunction timer
- Start triggered by register writing

Table 3-3 below defines the register settings and PPG start/stop conditions.

Table 3-3 PPG start condition settings

PPGC Register TTGR bit	GATEC Register STGR bit	PPG start/stop conditions
1	-	Start triggered by the Timing Generator Circuit
0	1	Start triggered by GATE signal from the multifunction timer
0	0	Start triggered by TRG Register writing

- Start triggered by the Timing Generator Circuit

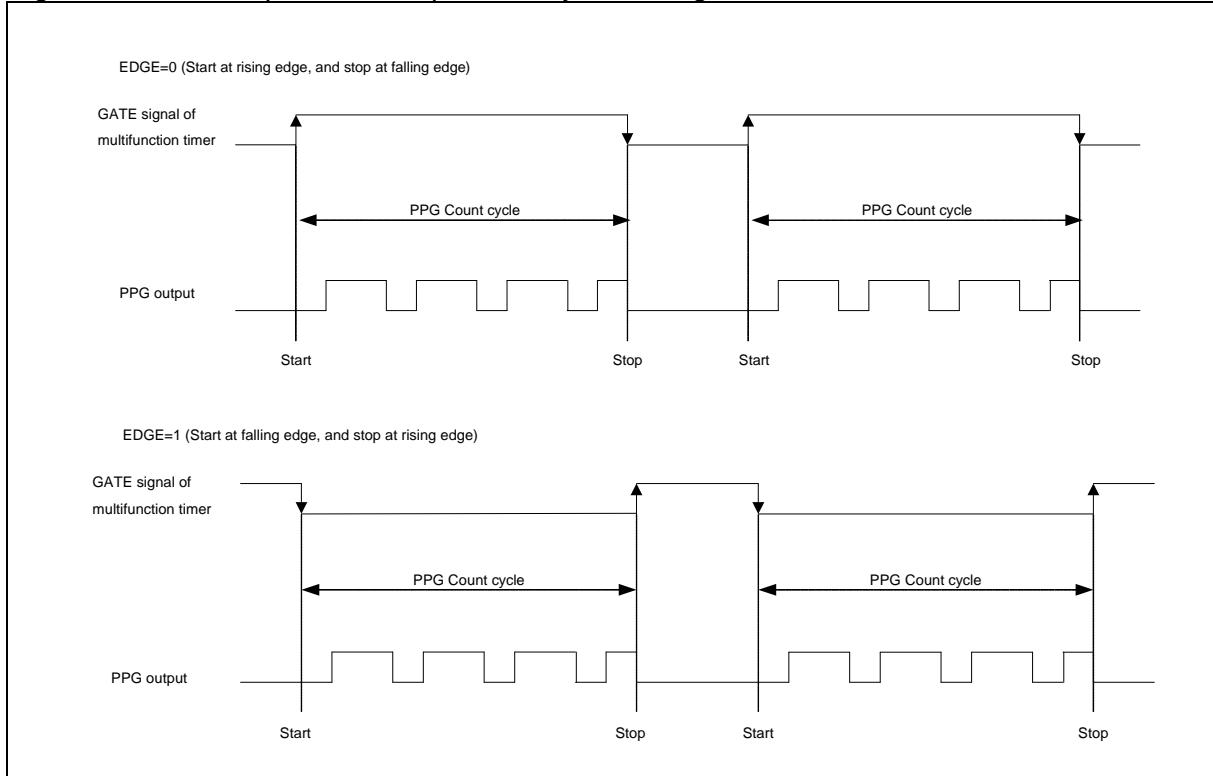
The Timing Generator Circuit compares its built-in, 8-bit UP counter value with the 8-bit Compare register value, and generates/outputs a start trigger signal of PPG Timer when they match. The details are given in "3.2 Timing generator circuit operations."

- Start triggered by GATE signal from the multifunction timer

The PPG can be started and stopped by the GATE signal from the multifunction timer.

Also, an effective startup period of PPG can be set by combination of EDGE bit of GATEC register and the GATE signal from the multifunction timer. Figure 3-2 shows an example of PPG startup by the GATE signal from the multifunction timer.

Figure 3-2 An example of PPG operation by GATE signal from the multifunction timer



1. PPG continues output operation during the GATE signal is active.
2. After LOW level pulses for $T \times (L+1)$ count are output, HIGH level pulses for $T \times (H+1)$ count are output.
3. After HIGH level pulses are counted, LOW level pulses are counted again and output.
The pulse output is continued when the GATE signal is active.
4. When the GATE signal is made inactive, PPG pulse output is stopped.

Note: PPG startup at a rising edge or a falling edge of GATE signal can be set by the EDGE bit.

- Start triggered by TRG register writing
When the PEN bit TRG register (PPG start register) is set to "1" at each channel, the PPG is started and it starts pulse counting. The PPG stops counting when the PEN bit of each channel is set to "0". The details are shown in Figure 3-1.

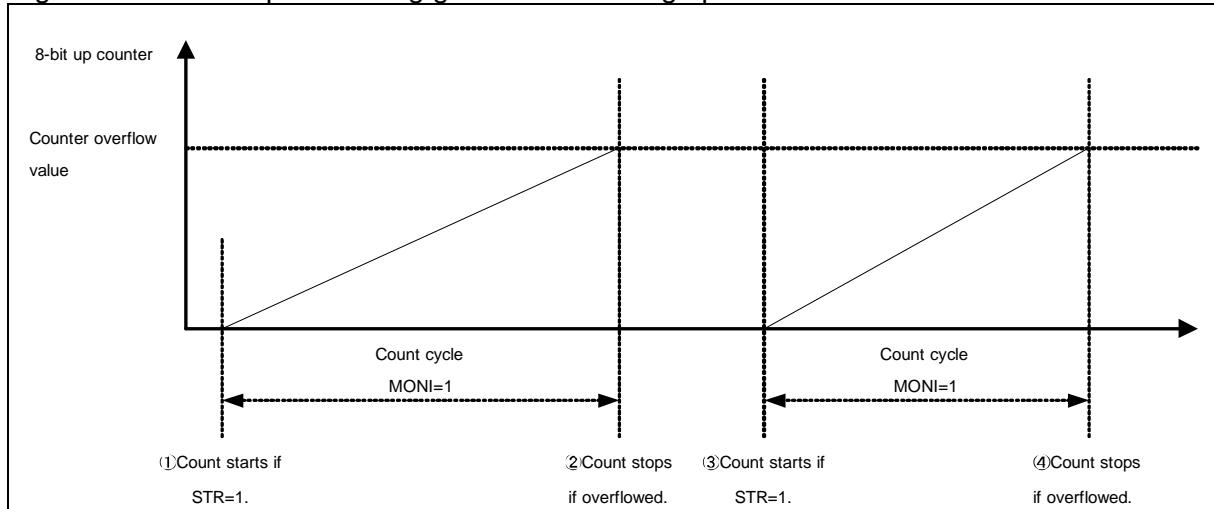
3.2. Timing generator circuit operations

The Timing generator circuit compares its built-in, 8-bit UP counter value with the 8-bit compare register value, and generates/outputs a start trigger signal of PPG Timer when they match.

If a compare register is set for each PPG timer, multiple PPG timers can be synchronized and they can be started with a delay of each PPG start timing.

- A delay period can be set by the four compare registers relating to each PPG channel. The 8-bit UP counter value is entered in each comparator and when it matches the compare register value, each PPG start trigger signal is generated.
- Four channels of PPGs can be synchronized for a single timing generator, and those PPGs can be started with a delay.
- Four counter operation clocks (PCLK/2, PCLK/8, PCLK/32, and PCLK/64) can be selected.
- When the 8-bit UP counter is running, the MONI bit is read as "1". When stopped, the MONI bit is read as "0".
- The PPG starts when the STR bit of TTCSR register (PPG start trigger control register) is set to "1". The PPG stops when TRGnO=0 is set. Also, the counting stops if the UP counter overflows.
- Figure 3-3 gives an example of Timing Generator counting operation.

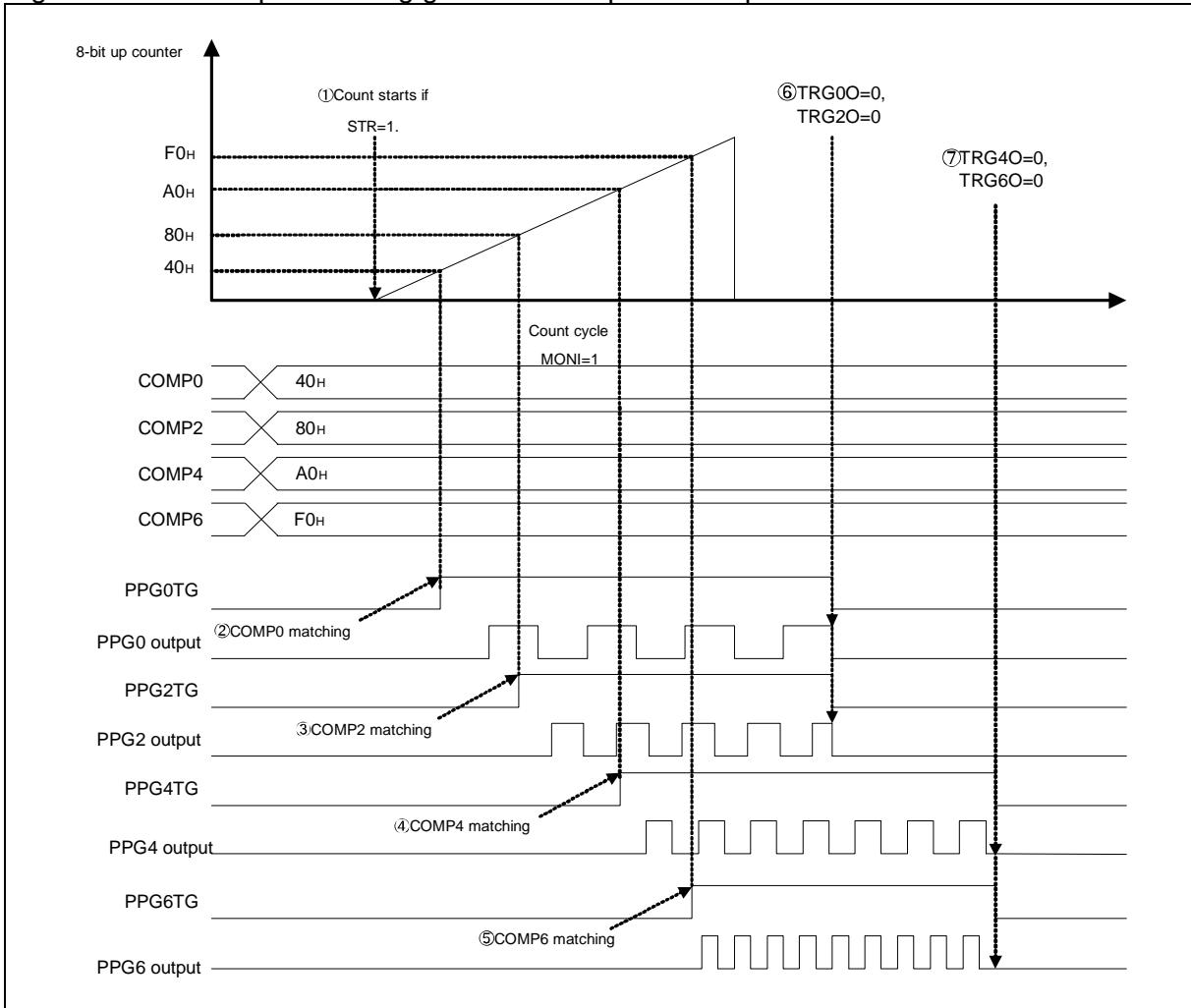
Figure 3-3 An example of timing generator counting operation



1. When TTCSR.STR=1 is set, the 8-bit UP counter starts counting.
2. During counting, TTCSR.MONI=1 is read.
When counting is stopped, TTCSR.MONI=0 is read.
3. If the 8-bit UP counter overflows, TTCSR.STR=0 is set and the counting is stopped.
4. When TTCSR.STR=1 is set again, the 8-bit UP counter starts counting.

- Figure 3-4 gives an example of Timing Generator compare startup.

Figure 3-4 An example of timing generator compare startup



- When TTCR.STR=1 is set, the 8-bit UP counter starts counting.
During counting, TTCR.MONI=1 is read.
- If the 8-bit UP counter value matches the COMP0 value, the PPG0 trigger signal (PPG0TG) is made active.
- If the 8-bit UP counter value matches the COMP2 value, the PPG2 trigger signal (PPG2TG) is made active.
- If the 8-bit UP counter value matches the COMP4 value, the PPG4 trigger signal (PPG4TG) is made active.
- If the 8-bit UP counter value matches the COMP6 value, the PPG6 trigger signal (PPG6TG) is made active.
- If TTCR.TRG0O=0 and TTCR.TRG2O=0 are set, the PPG0 and PPG2 trigger signals are made inactive.
- If TTCR.TRG4O=0 and TTCR.TRG6O=0 are set, the PPG4 and PPG6 trigger signals are made inactive.

Notes: Figure 3-4 shows the operation for Timing Generator 0.

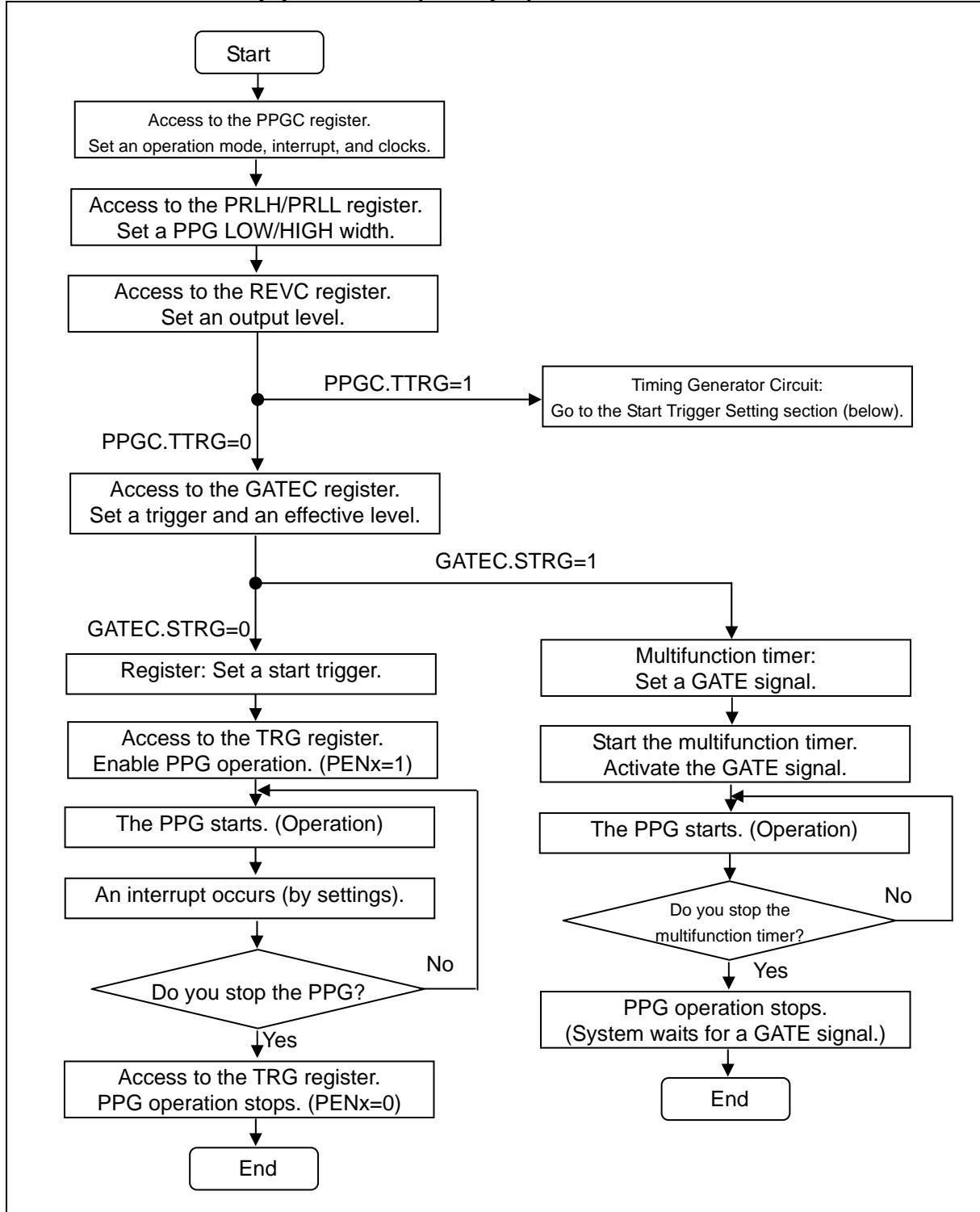
The COMP Register value must be written before TTCR.STR=1 is set.

In Figure 3-4, when the counter value matches the COMP register value, PPG output starts the LOW pulses.

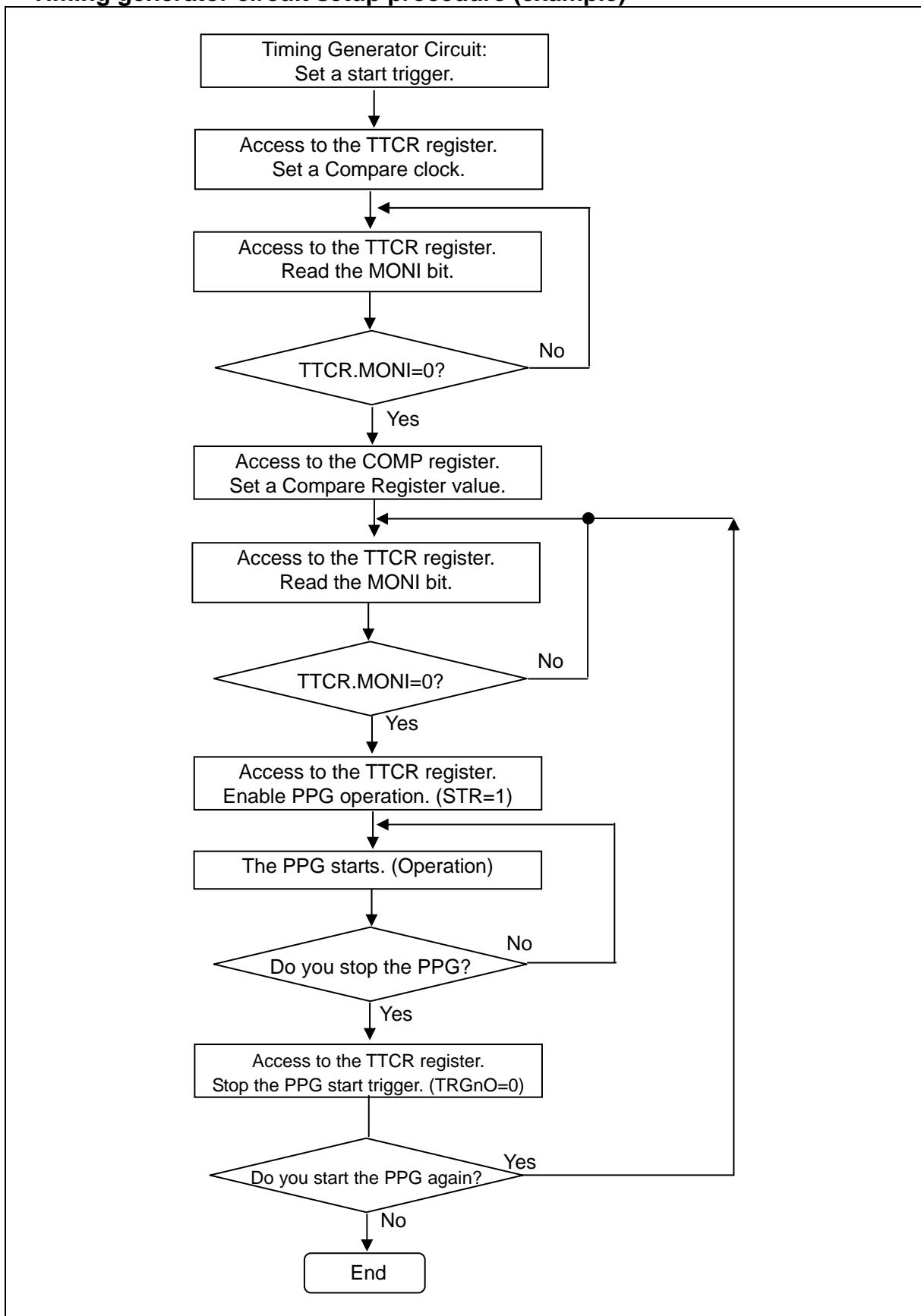
4. Setup Procedure Example

This section explains a setting procedure example of PPG.

■ PPG function setup procedure (example)



■ Timing generator circuit setup procedure (example)



5. Registers

This section explain the registers of PPG.

Table 5-1 lists and explains PPG Registers.

Table 5-1 PPG Register list

Abbreviation	Register name	See
TTCR0	PPG Start Trigger Control Register 0	5.1
TTCR1	PPG Start Trigger Control Register 1	5.2
COMP0	PPG Compare Register 0	5.3
COMP1	PPG Compare Register 1	
COMP2	PPG Compare Register 2	
COMP3	PPG Compare Register 3	
COMP4	PPG Compare Register 4	
COMP5	PPG Compare Register 5	
COMP6	PPG Compare Register 6	
COMP7	PPG Compare Register 7	
TRG	PPG Start Register	5.4
REVC	Output Reverse Register	5.5
PPGC0	PPG Operation Mode Control Register 0	5.6
PPGC1	PPG Operation Mode Control Register 1	
PPGC2	PPG Operation Mode Control Register 2	
PPGC3	PPG Operation Mode Control Register 3	
PPGC4	PPG Operation Mode Control Register 4	
PPGC5	PPG Operation Mode Control Register 5	
PPGC6	PPG Operation Mode Control Register 6	
PPGC7	PPG Operation Mode Control Register 7	
PPGC8	PPG Operation Mode Control Register 8	
PPGC9	PPG Operation Mode Control Register 9	
PPGC10	PPG Operation Mode Control Register 10	
PPGC11	PPG Operation Mode Control Register 11	
PPGC12	PPG Operation Mode Control Register 12	
PPGC13	PPG Operation Mode Control Register 13	
PPGC14	PPG Operation Mode Control Register 14	
PPGC15	PPG Operation Mode Control Register 15	
PRLH0	PPG Reload Register H0	5.7
PRLL0	PPG Reload Register L0	
PRLH1	PPG Reload Register H1	
PRLL1	PPG Reload Register L1	
PRLH2	PPG Reload Register H2	
PRLL2	PPG Reload Register L2	

Abbreviation	Register name	See
PRLH3	PPG Reload Register H3	5.7
PRLL3	PPG Reload Register L3	
PRLH4	PPG Reload Register H4	
PRLL4	PPG Reload Register L4	
PRLH5	PPG Reload Register H5	
PRLL5	PPG Reload Register L5	
PRLH6	PPG Reload Register H6	
PRLL6	PPG Reload Register L6	
PRLH7	PPG Reload Register H7	
PRLL7	PPG Reload Register L7	
PRLH8	PPG Reload Register H8	
PRLL8	PPG Reload Register L8	
PRLH9	PPG Reload Register H9	
PRLL9	PPG Reload Register L9	
PRLH10	PPG Reload Register H10	
PRLL10	PPG Reload Register L10	
PRLH11	PPG Reload Register H11	
PRLL11	PPG Reload Register L11	
PRLH12	PPG Reload Register H12	
PRLL12	PPG Reload Register L12	
PRLH13	PPG Reload Register H13	
PRLL13	PPG Reload Register L13	
PRLH14	PPG Reload Register H14	
PRLL14	PPG Reload Register L14	
PRLH15	PPG Reload Register H15	
PRLL15	PPG Reload Register L15	
GATEC0	Gate Function Control Register 0	5.8
GATEC4	Gate Function Control Register 4	
GATEC8	Gate Function Control Register 8	
GATEC12	Gate Function Control Register 12	

5.1. PPG Start Trigger Control Register 0 (TTCR0)

The TTCR0 Register controls a start of PPG0/PPG2/PPG4/PPG6 Register.

■ Register configuration

Bit	15	14	13	12	11	10	9	8
Field	TRG6O	TRG4O	TRG2O	TRG0O	CS01	CS00	MONI0	STR0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	1'b1	1'b1	1'b1	1'b1	1'b0	1'b0	1'b0	1'b0
Bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	-							
Initial value	-							

■ Register functions

[Bits 15:12] TRG6O, TRG4O, TRG2O, TRG0O: PPG trigger stop bits

These bits control the PPG start trigger signal.

Bits 15:12	Function
Read	"1" is always read.
Writing by 0	Disables the PPG start trigger signal. (LOW output)
Writing by 1	No effect

[Bit 11:10] CS01, CS00: Count clock select bits of UP counter comparing built-in compare register.

These bits set an operation clock of UP counter.

Bit 11	Bit 10	Function
0	0	PCLK/2 [Initial value]
0	1	PCLK/8
1	0	PCLK/32
1	1	PCLK/64

[Bit 9] MONI0: 8-bit UP counter operation state monitor bit

This bit indicates the PPG's 8-bit UP counter operation state.

Bit	Function
Reading as 0	The PPG UP counter is stopped. [Initial value]
Reading as 1	The PPG UP counter is operating.
During writing	No effect

[Bit 8] STR0: 8-bit UP counter operation Enable bit
This bit enables the 8-bit UP counter operation.

Bit	Function
Read	"0" is always read.
Writing by 0	No effect
Writing by 1	Starts the PPG UP counter operation.

[Bits 7:0] RES: Reserved bits
"0b00000000" is read from these bits.
Set these bits to "0b00000000" when writing.

<Notes>

- If TRGnO=1 is set for matching by compare register value and if TRGnO=0 is set simultaneously, the start trigger clear operation by TRGnO=0 setting preceded.
 - If TRGnO=0 is set by register writing before the TRGnO=1 is set for matching by compare register value, it does not have an affect on other operations.
 - The STR bit writing is prohibited when the 8-bit UPcounter is operating. The STR bit must be set to "1" when the MONI bit is "0".
 - If the STR bit is set to "1" when the 8-bit UP counter is operating, this writing is ignored.
 - The CS bit writing is prohibited after the 8-bit UP counter has operated.
-

5.2. PPG Start Trigger Control Register 1 (TTCR1)

The TTCR1 Register controls a start of PPG8/PPG10/PPG12/PPG14 Register.

■ Register configuration

Bit	15	14	13	12	11	10	9	8
Field	TRG7O	TRG5O	TRG3O	TRG1O	CS11	CS10	MONI1	STR1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	1'b1	1'b1	1'b1	1'b1	1'b0	1'b0	1'b0	1'b0
Bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	-							
Initial value	-							

■ Register functions

[Bits 15:12] TRG7O, TRG5O, TRG3O, TRG1O: PPG trigger stop bits

These bits control the PPG start trigger signal.

Bits 15:12	Function
Read	"1" is always read.
Writing by 0	Disables the PPG start trigger signal. (LOW output)
Writing by 1	No effect

[Bit 11:10] CS11, CS10: Count clock select bits of UP counter comparing built-in compare register.

These bits set an operation clock of UP counter.

Bit 11	Bit 10	Function
0	0	PCLK/2 [Initial value]
0	1	PCLK/8
1	0	PCLK/32
1	1	PCLK/64

[Bit 9] MONI1: 8-bit UP counter operation state monitor bit

This bit indicates the PPG's 8-bit UP counter operation state.

Bit	Function
Reading as 0	The PPG UP counter is stopped. [Initial value]
Reading as 1	The PPG UP counter is operating.
During writing	No effect

[Bit 8] STR1: 8-bit Counter Operation Enable bit
This bit enables the 8-bit UP counter operation.

Bit	Function
Read	"0" is always read.
Writing by 0	No effect
Writing by 1	Starts the PPG Counter operation.

[Bits 7:0] RES: Reserved bits
"0b00000000" is read from these bits.
Set these bits to "0b00000000" when writing.

<Notes>

- If TRGnO=1 is set for matching by compare register value and if TRGnO=0 is set simultaneously, the start trigger clear operation by TRGnO=0 setting preceded.
 - If TRGnO=0 is set by register writing before the TRGnO=1 is set for matching by compare register value, it does not have an affect on other operations.
 - The STR bit writing is prohibited when the 8-bit UPcounter is operating. The STR bit must be set to "1" when the MONI bit is "0".
 - If the STR bit is set to "1" when the 8-bit UP counter is operating, this writing is ignored.
 - The CS bit writing is prohibited after the 8-bit UP counter has operated.
-

5.3. PPG Compare Register "n" (COMPn, where n=0 to 7)

The COMPn Register sets a Compare Register value of the Timing Generator.

■ Register configuration

Bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
Field					COMPn			
Attribute					R/W			
Initial value					8'b00000000			

Note: Bits 7 to 0 are set for an odd number address, but bits 15 to 8 are set for an even number address.

■ Register functions

[Bits 15:8, or bits 7:0] COMP7 to COMP0: Compare Register channels 7 to 0
Sets a PPG Compare Register value.

Bits 15:8, or bits 7:0	Function
Read	Reads the Compare Register value. Initial value is "0b00000000".
Write	Writes a Compare Register value.

<Notes>

- This is an 8-bit Compare Register, and it is assigned to each of PPG start trigger signals.
- When the register value matches the 8-bit counter value, a PPG start trigger signal of the channel corresponding to the matching register value is switched from LOW to HIGH state and output.
- If this register value is "0b00000000", it is not compared with the 8-bit counter value. The PPG start trigger signal is never output HIGH but is kept LOW.
- This register writing is prohibited when the 8-bit Counter is operating.

5.4. PPG Start Register (TRG)

The TRG Register allows a start of the PPG.

■ Register configuration

Bit	15	14	13	12	11	10	9	8
Field	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08
Attribute	R/W							
Initial value	1'b0							
Bit	7	6	5	4	3	2	1	0
Field	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00
Attribute	R/W							
Initial value	1'b0							

■ Register functions

[Bits 15:0] PEN15 to PEN00: PPG Operation Enable bits

Starts PPG operation and sets its operation mode.

Bits 15:0	Function
0	Disables PPG operation. (The LOW output is held.) [Initial value]
1	Enables PPG operation.

<Note>

In 16-bit PPG mode, a combination of TRG Register PENn bits of each channel must be started or stopped simultaneously.

5.5. Output Reverse Register (REVC)

The REVC Register sets an output polarity of PPG output value.

■ Register configuration

Bit	15	14	13	12	11	10	9	8
Field	REV15	REV14	REV13	REV12	REV11	REV10	REV09	REV08
Attribute	R/W							
Initial value	1'b0							
Bit	7	6	5	4	3	2	1	0
Field	REV07	REV06	REV05	REV04	REV03	REV02	REV01	REV00
Attribute	R/W							
Initial value	1'b0							

■ Register functions

[Bits 15:0] REV15 to REV00: PPG Output Reverse Enable bits

Sets a polarity of PPG output value.

Bits 15:0	Function
0	Normal output (LOW output when PPG is not operating) [Initial value]
1	Invert the output. (HIGH output when PPG is stopped)

<Note>

In 16-bit operation mode, the PPG values of even-numbered channels (PPG0, PPG2, PPG4, PPG6, PPG8, PPG10, PPG12, and PPG14) are only output. If the REVC Registers of odd-numbered channels are written, it has no effect.

5.6. PPG Operation Mode Control Register (PPGC)

PPGC Register sets an interrupt, an operation mode, and the prescaler data.

■ PPGC Register configuration list

15	8	7	0	Initial value	Access	Corresponding PPG
PPGC0	PPGC1	0x0000	R/W	PPG0, PPG1		
PPGC2	PPGC3	0x0000	R/W	PPG2, PPG3		
PPGC4	PPGC5	0x0000	R/W	PPG4, PPG5		
PPGC6	PPGC7	0x0000	R/W	PPG6, PPG7		
PPGC8	PPGC9	0x0000	R/W	PPG8, PPG9		
PPGC10	PPGC11	0x0000	R/W	PPG10, PPG11		
PPGC12	PPGC13	0x0000	R/W	PPG12, PPG13		
PPGC14	PPGC15	0x0000	R/W	PPG14, PPG15		

■ Register configuration details

Bit Field	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
	PIE	PUF	INTM	PCS1	PCS0	MD1	MD0	TTRG

Note: Bits 7 to 0 are set for an odd number address, but bits 15 to 8 are set for an even number address.

■ Register functions

[Bit 15 or 7] PIE: PPG Interrupt Enable bit

Enables a PPG interrupt.

Bit 15 or 7	Function
0	Disables an interrupt. [Initial value]
1	Enables an interrupt.

[Bit 14 or 6] PUF: PPG Counter Underflow bit

Controls the underflow bits of PPG Counter.

Bit 14 or 6	Function
0	No underflow of PPG Counter was detected. [Initial value]
1	An underflow of PPG Counter was detected.

[Bit 13 or 5] INTM: Interrupt Mode Select bit

Sets an interrupt mode.

Bit 13 or 5	Function
0	Sets the PUF bit to "1" when a PPLH or PPLL underflow occurs. [Initial value]
1	Sets the PUF bit to "1" only when a PPLH underflow occurs.

[Bit 12 or 11, or bit 4 or 3] CS1, CS0: PPG DOWN Counter Operation Clock Select bits
 Sets an operation clock of PPG's DOWN Counter.

Bit 12 or 4	Bit 11 or 3	Function
0	0	PCLK [Initial value]
0	1	PCLK/4
1	0	PCLK/16
1	1	PCLK/64

[Bit 10 or 9, or bit 9 or 1] MD1, MD0: PPG Operation Mode Set bits
 PPG output value can be set to be reversed.

Bit 10 or 9	Bit 9 or 1	Function
0	0	Sets 8-bit operation mode. [Initial value]
0	1	Sets 8+8-bit operation mode.
1	0	Sets 16-bit operation mode.
1	1	Sets 16+16-bit operation mode.

[Bit 8 or 0] TTRG: PPG start trigger select bit
 Selects the PPG start trigger.

Bit 8 or 0	Function
0	Uses a TRG Register value or a multifunction timer value. [Initial value]
1	Uses a Timing Generator Circuit.

<Notes>

- If PPGC.PIF=1 and PPGC.PUF=1, an interrupt occurs.
- If an underflow occurs, the PPGC.PUF bits of Channel 0 and Channel 1 are set in the same timing.
- The PPGC.PUF bits are cleared when they are set to "0". The bit setting to "1" is ignored.
- When a Read-Modify-Write instruction is executed, the PPGC.PUF bit is always held to "1" regardless of the bit value.
- In 16-bit PPG mode, the values are the same between Channel 0 and Channel 1, and between Channel 2 and Channel 3.
 Therefore, the count value is within a range of 0x0000 to 0xFFFF.
- The PPGC.CS1, PPGC.CS0 and PPGC.TTRG bits are set only for Channel 0 and Channel 2 of PPGC Register.

5.7. PPG Reload Registers (PRLH, PRLL)

The PRLH and PRLL Registers set the LOW and HIGH width of PPG.

■ PPGC Register configuration list

15	8	7	0	Initial value	Access	Corresponding PPG
PRLH0	PRLL0			0xxxxx	R/W	PPG0
PRLH1	PRLL1			0xxxxx	R/W	PPG1
PRLH2	PRLL2			0xxxxx	R/W	PPG2
PRLH3	PRLL3			0xxxxx	R/W	PPG3
PRLH4	PRLL4			0xxxxx	R/W	PPG4
PRLH5	PRLL5			0xxxxx	R/W	PPG5
PRLH6	PRLL6			0xxxxx	R/W	PPG6
PRLH7	PRLL7			0xxxxx	R/W	PPG7
PRLH8	PRLL8			0xxxxx	R/W	PPG8
PRLH9	PRLL9			0xxxxx	R/W	PPG9
PRLH10	PRLL10			0xxxxx	R/W	PPG10
PRLH11	PRLL11			0xxxxx	R/W	PPG11
PRLH12	PRLL12			0xxxxx	R/W	PPG12
PRLH13	PRLL13			0xxxxx	R/W	PPG13
PRLH14	PRLL14			0xxxxx	R/W	PPG14
PRLH15	PRLL15			0xxxxx	R/W	PPG15

■ Register configuration

Bit	15	14	13	12	11	10	9	8
Field					PRLH			
Attribute					R/W			
Initial value					8'bXXXXXXXXX			
Bit	7	6	5	4	3	2	1	0
Field					PRLL			
Attribute					R/W			
Initial value					8'bXXXXXXXXX			

■ Register functions

[Bits 15:8] PRLH: PPG Reload Register Level-HIGH Set bits
Sets a HIGH value of PPG.

Bit	Function
During writing	Any value can be written.
During reading	The register value is read. The initial value is undefined.

[Bits 7:0] PRLL: PPG Reload Register Level-LOW Set bits
Sets a LOW value of PPG.

Bit	Function
During writing	Any value can be written.
During reading	The register value is read. The initial value is undefined.

<Notes>

This register operation is determined based on PPG operation mode.
The following defines each operation mode.

- 8-bit operation mode combination

15	8 7	0
PRLH0		PRLL0
Sets the HIGH width of PPG0.		Sets the LOW width of PPG0.
PRLH1		PRLL1
Sets the HIGH width of PPG1.		Sets the LOW width of PPG1.
PRLH2		PRLL2
Sets the HIGH width of PPG2.		Sets the LOW width of PPG2.
PRLH3		PRLL3
Sets the HIGH width of PPG3.		Sets the LOW width of PPG3.
•		•
•		•

- Sets the PPGn channel. (n=0 to 15)

The 8-bit PRLH bits of PPGn set the HIGH width of PPGn. The 8-bit length PRLL bits set the LOW width of PPGn.

- 8+8-bit operation mode combination

15	8 7	0
PRLH0		PRLL0
Sets the HIGH width of PPG0.		Sets the LOW width of PPG0.
PRLH1		PRLL1
Sets the HIGH width of PPG0-pri.		Sets the LOW width of PPG0-pri.
PRLH2		PRLL2
Sets the HIGH width of PPG2.		Sets the LOW width of PPG2.
PRLH3		PRLL3
Sets the HIGH width of PPG2-pri.		Sets the LOW width of PPG2-pri.
•		•
•		•

- PPGn and PPGn+1 channel settings (where, n=0, 2, 4, 6, 8, 10, 12, or 14)

The 8-bit PRLH bits of PPGn set the HIGH width of PPGn. The 8-bit length PRLL bits set the LOW width of PPGn. The 8-bit length PRLH bits of PPGn+1 set the HIGH width of PPGn prescaler. The 8-bit length PRLL bits of PPGn+1 set the LOW width of PPGn prescaler.

- 16-bit operation mode combination

15	8 7	0
PRLH0		PRLL0
Sets the HIGH width of PPG0.		
PRLH1		PRLL1
Sets the LOW width of PPG0.		
PRLH2		PRLL2
Sets the HIGH width of PPG2.		
PRLH3		PRLL3
Sets the LOW width of PPG2.		
•		•
•		•

- PPGn and PPGn+1 channel settings (where, n=0, 2, 4, 6, 8, 10, 12, or 14)

The 16-bit length that combined PRLH with PRLL bits of PPGn set the HIGH width of PPGn. The 16-bit length that combined PRLH with PRLL bits of PPGn+1 set the LOW width of PPGn.

- 16+16-bit operation mode combination

15	8 7	0
PRLH0		PRLL0
	Sets the HIGH width of PPG0.	
PRLH1		PRLL1
	Sets the LOW width of PPG0.	
PRLH2		PRLL2
	Sets the HIGH width of PPG0-pri.	
PRLH3		PRLL3
	Sets the LOW width of PPG0-pri.	
•		•
•		•

- Settings of PPGn channel (n=0, 4, 8, or 12) and PPGn+1 channels, PPGn+1 channel, or PPGn+2 channel
The 16-bit length that combined PRLH with PRLL bits of PPGn set the HIGH width of PPGn. The
16-bit length that combined PRLH with PRLL bits of PPGn+1 set the LOW width of PPGn. The 16-bit
length that combined PRLH with PRLL bits of PPGn+2 set the HIGH pulse of PPGn prescaler. The
16-bit length that combined PRLH with PRLL bits of PPGn+3 set the LOW pulse of PPGn prescaler.

5.8. PPG Gate Function Control Registers (GATEC0/GATEC4/GATEC8/GATEC12)

The GATEC Registers specify the start of the PPG using a GATE signal sent from the multifunction timer.

■ GATEC Register configuration list

15	8	7	0	Initial value	Access	Corresponding PPG
Reserved		GATEC0		0x00	R/W	PPG2, PPG0
Reserved		GATEC4		0x00	R/W	PPG6, PPG4
Reserved		GATEC8		0x00	R/W	PPG10, PPG8
Reserved		GATEC12		0x00	R/W	PPG14, PPG12

■ Register configuration (n=0, 4, 8, or 12)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	STRGn+2	EDGEN+2	Reserved	STRGn	EDGEN		
Attribute	-	R/W	R/W	-	R/W	R/W		
Initial value	-	1'b0	1'b0	-	1'b0	1'b0		

■ Register functions

[Bits 7:6] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[Bit 5] STRGn+2: Select trigger bit 2 (n=0, 4, 8, or 12)

Selects an operation trigger signal for PPGn+2.

Bit	Function
0	Start by the TRG Register setting. [Initial value]
1	Start by GATE signal from the multifunction timer.

[Bit 4] EDGEN+2: Start Effective Level Select bit "n+2" (where, n=0, 4, 8, or 12)

Sets an effective level of GATEn+2 signal from the multifunction timer.

Bit	Function
0	Start at HIGH level. [Initial value]
1	Start at the LOW level.

[Bits 3:2] RES: Reserved bits

"0b00" is read from these bits.

Set these bits to "0b00" when writing.

[Bit 1] STRGn: Select trigger bit "n" (where, n=0, 4, 8, or 12)

Selects an operation trigger signal for PPGn.

Bit	Function
0	Start by the TRG Register setting. [Initial value]
1	Start by GATE signal from the multifunction timer.

[Bit 0] EDGEN: Start Effective Level Select bit "n" (where, n=0, 4, 8, or 12)

Sets an effective level of GATEn signal from the multifunction timer.

Bit	Function
0	Start at HIGH level. [Initial value]
1	Start at the LOW level.

6. Notes

This section explains the notes when using the PPG.

● PPG output operations

When the PPG is operating, the pulses of LOW period and HIGH level period are continuously output. Once the pulse output has started, the PPG does not stop the output until PPG operation is stopped.

A reset signal must be entered or the PPG stop setting must be set to stop the operation.
The following explains PPG stop conditions.

- Start triggered by the Timing Generator Circuit
Disable the start trigger by setting PPGC.TRGnO=0.
- Start triggered by GATE signal from the multifunction timer
Disable the start trigger by setting GATE=0 (at HIGH edge detection) which is sent from the Multifunction timer.
- Start triggered by TRG Register writing
Disable the start trigger by setting PEN=0.

● PPG operation mode setting

The mode startup setting is selected by the MD bit setting of each PPGC Register.
The MD bits must always be set to select the desired operation mode before the PPG is started.

● Other module settings

PPG pulses are output via the I/O port of multifunction timer. The multifunction timer settings are explained in Chapter "Multifunction Timer". For details on pulse output to I/O ports, see Chapter "I/O Ports". Also, for details on interrupts, see Chapter "Interrupts".

● Interrupts

In 8-bit operation mode or 8+8-bit operation mode, when an underflow occurs on each counter, an interrupt request is issued for each of them. However, in 16-bit operation mode or 16+16-bit operation mode, when an underflow occurs on a 16-bit counter, the PUF(m) and PUF(m+1) bits are set simultaneously. Therefore, it is recommend to enable either of PIE(m) and PIE(m+1) bits to use a single interrupt source. Also, it is recommend to clear both PUF(m) and PUF(m+1) bits simultaneously to clear the cause of the interrupt ($m=0, 2, 4, 6, 8, 10, 12, 14$).

Chapter: Quad Position & Revolution Counter

This chapter explains the functions and operations of the Quad Position & Revolution Counter (QPRC).

1. Overview
2. Configuration
3. Operations
4. Registers

1. Overview

The Quad Position & Revolution Counter is used to measure the position of Position Encoder. Also, it can be used as an up/down counter by its setting. The Quad Position & Revolution Counter contains a 16-bit position counter, a 16-bit revolution counter, two 16-bit compare registers, a control register, and its control circuit.

■ Features of Quad Position & Revolution Counter

- **The position counter can be operated in one of the following 3 counting modes:**

- PC_Mode1 : Up/down count mode
- PC_Mode2 : Phase difference count mode (supporting the 2-time and 4-time frequency multiplication)
- PC_Mode3 : Count mode with direction

- **The revolution counter can be operated in one of the following 3 counting modes:**

- RC_Mode1 : The revolution counter can count up or down at a ZIN active edge only.
- RC_Mode2 : The revolution counter can count up or down with an output value of position counter only.
- RC_Mode3 : The revolution counter can count up or down both with an output value of position counter and a signal at ZIN active edge.

- **A signal edge detection can be set for detecting an input event from three AIN, BIN and ZIN external pins**

- Detection of falling edge
- Detection of rising edge
- Detection of both rising and falling edges

- **The following two functions can be selected for input in ZIN pin**

- Counter clear function
- Gate function

- **An interrupt request can be generated if:**

- The position counter value matches the Position Compare Register,
- The position counter value matches the Position and Revolution Compare Register value, or the revolution counter value matches the Position and Revolution Compare Register value,
- The position counter underflows,
- The position counter overflows (that is, the position counter value matches the value of the QPRC Maximum Position Register),
- The position counter is reset at a ZIN active edge,
- The counting of position counter is inverted,
- The position counter matches the Position Compare Register value, and the revolution counter matches the Position and Revolution Compare Register value, or
- An outrange revolution counter value is detected.

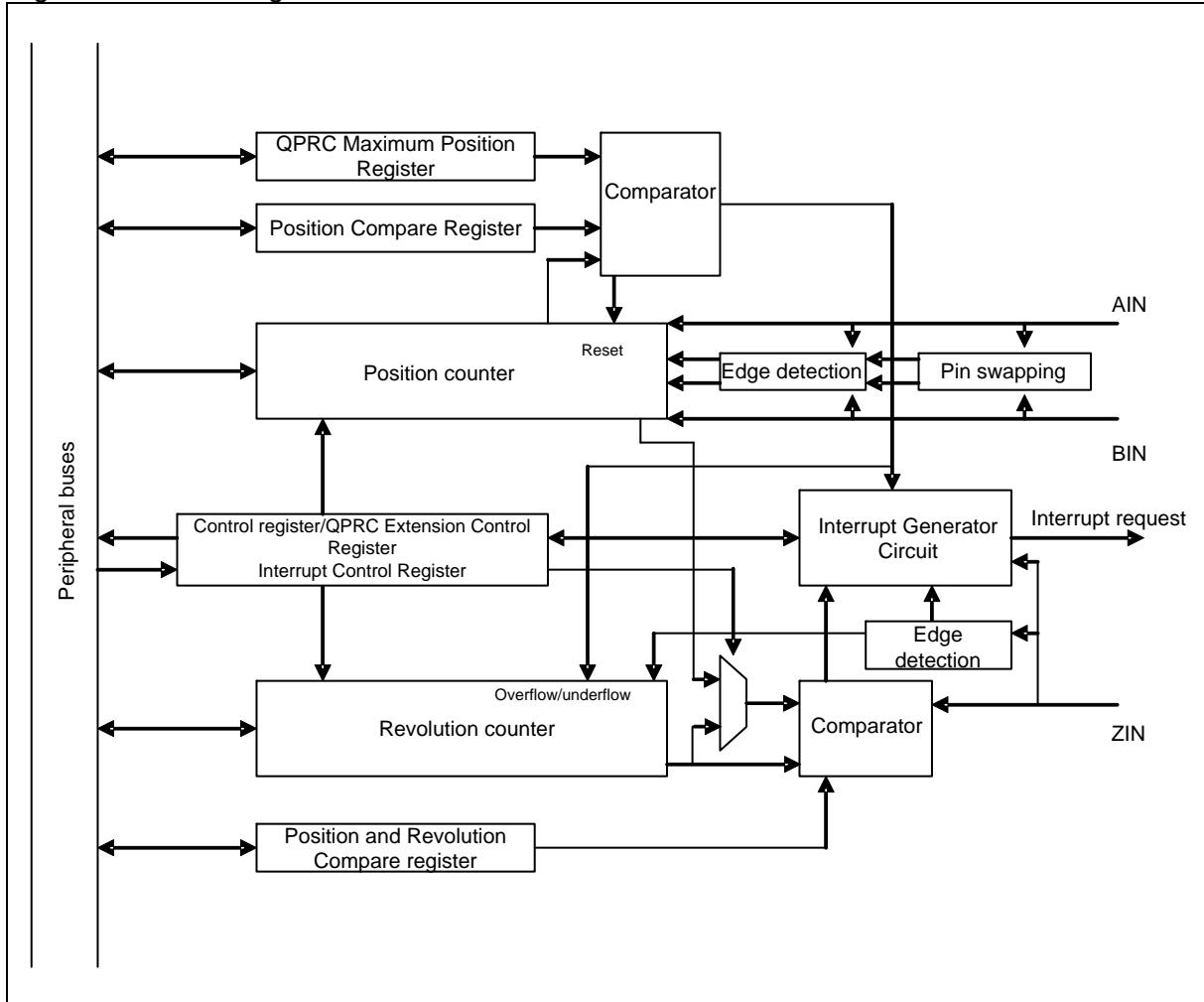
- **The following useful functions are provided for counting**

- Swap function of AIN and BIN external pins
- Mask reset function of the position counter
- Count direction check function during position counter operation or during overflow/underflow occurrence

2. Configuration

The following shows the configuration of Quad Position & Revolution Counter.

Figure 2-1 Block diagram of Quad Position & Revolution Counter



3. Operations

This section explains the operation of Quad Position & Revolution Counter.

■ Operation of position counter

The position counter receives an input signal from AIN or BIN external pin as an event of count clock, and increments or decrements the counter. As listed in Table 3-1, the position counter can select a counting mode by setting of the position counter mode bits (QCR:PCM[1:0]) of a control register. The counting conditions depend on the selected count mode.

The position counter is counted up or down in the following ZIN conditions only.

- If the ZIN function is set to the count clear function (QCR:CGSC="0")
- If the ZIN function is set to the Gate function (QCR:CGSC="1"), the ZIN low-level detection (QCR:CGE[1:0]=="01") is set, and the ZIN is logical low
- If the ZIN function is set to the Gate function (QCR:CGSC="1"), the ZIN high-level detection (QCR:CGE[1:0]=="10") is set, and the ZIN is logical high

If the ZIN function is set to the Gate function (QCR:CGSC="1") and if a level other than ZIN high- or low-level detection (QCR:CGE[1:0]=="00" or "11") is set, the position counter is not counted up or down.

Also, if the AIN and BIN configuration is swapped by SWAP bits of a control register, the AIN and BIN pins are swapped and the position counter is counted up or down.

For example, if PC_Mode1 (QCR:PCM[1:0]=="01") and AES[1:0]=="10" (rising edge) and BES[1:0]=="01" (falling edge) are set, the following occurs.

- If QCR:SWAP="0" and when a rising edge of AIN signal is detected, the position counter is counted up. If a falling edge of BIN signal is detected, the position counter is counted down.
- If QCR:SWAP="1", the position counter is counted down at a falling edge of AIN signal but it is counted up at a rising edge of BIN signal.

Table 3-1 Counting conditions of AIN and BIN pin position counter

Position count mode (PC_MODE)	AIN counting conditions	BIN counting conditions
Count disable PC_Mode0:QCR:PCM[1:0]=="00"	Position counter disable	Position counter disable
Up/down counting PC_Mode1: QCR:PCM[1:0]=="01"	AIN Active edge	BIN Active edge
Phase difference count PC_Mode2:QCR:PCM[1:0]=="10"	AIN Active edge or high/low level	High/low level or BIN active edge
Counting with direction PC_Mode3:QCR:PCM[1:0]=="11"	High/low level	BIN Active edge

<Note>

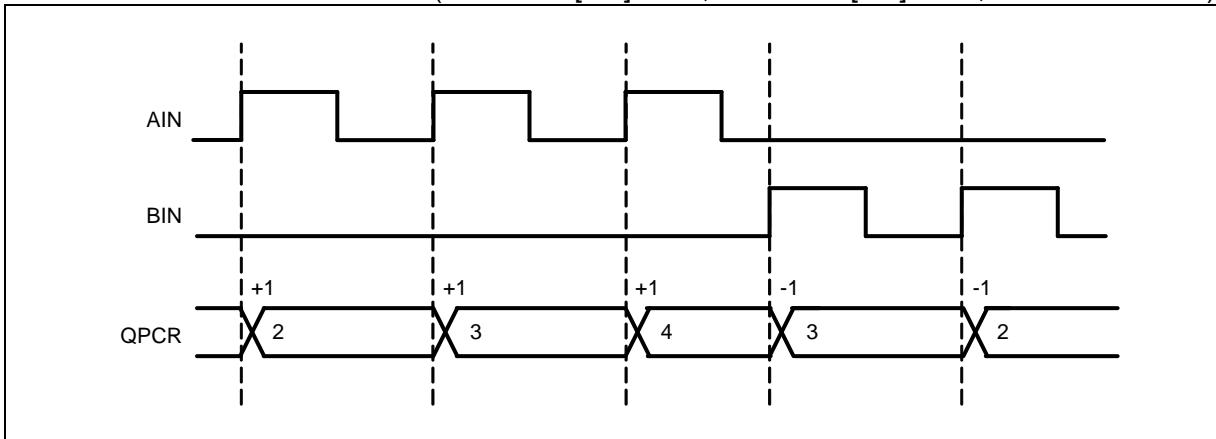
The active edge of AIN signal and the active edge of BIN signal mean a rising edge, a falling edge, or both of the respective signal if they are set by the AIN Detection Edge Select bits (QCR:AES[1:0]=="01" or "10" or "11") or by the BIN Detection Edge Select bits (QCR:BES[1:0]=="01" or "10" or "11").

● **PC_Mode1: Up/down count mode**

- An external signal entered from AIN or BIN external pin is received as the counting clock, and the position counter is counted up or down.
- In this mode, the position counter is counted up when an active edge of AIN signal is detected. When an active edge of BIN signal is detected, the position counter is counted down.

Figure 3-1 Operations in up/down count mode

(QCR:AES[1:0] = "10", QCR:BES[1:0] = "10", QCR:SWAP = "0")



● PC_Mode2: Phase difference count mode (supporting the 2-time and 4-time frequency multiplication)

- This mode is useful for counting the difference between phases A and B of "encoder output signal." If the phase-A and phase-B outputs are respectively connected to the AIN and BIN pins and if phase A is leading phase B, the counter is counted up. If delayed, the counter is counted down.
- In this mode, when an active edge of AIN signal is detected, the BIN signal level is checked and the position counter counts it. In the opposite case, the position counter also counts it.
- Counting in the 4-time or 2-time frequency multiplication can be made by setting the AES and BES bits of QPRC Control Register (QCR). The counting in these frequency multiplication modes allows more accurate position measurement as its counting resolution is very high.

Table 3-2 AES and BES bit settings in frequency multiplication mode

Frequency multiplication mode	AES[1:0] setting	BES[1:0] setting
1-time frequency multiplication mode	01	00
	10	00
	00	01
	00	10
2-time frequency multiplication mode	11	00
	00	11
4-time frequency multiplication mode	11	11

Table 3-3 Counting in 2-time frequency multiplication mode
(QCR:AES[1:0]="00", QCR:BES[1:0]="11", QCR:SWAP="0")

Edge detection pin	Detection edge	Level Check pin	Input level	Counting direction	Figure 3-2 Timing
BIN	Rising edge	AIN	High	Up	(1)
	Rising edge		Low	Down	(2)
	Falling edge		High	Down	(3)
	Falling edge		Low	Up	(4)

Figure 3-2 Operation in 2-time frequency multiplication mode
(QCR:AES[1:0]="00", QCR:BES[1:0]="11", QCR:SWAP="0")

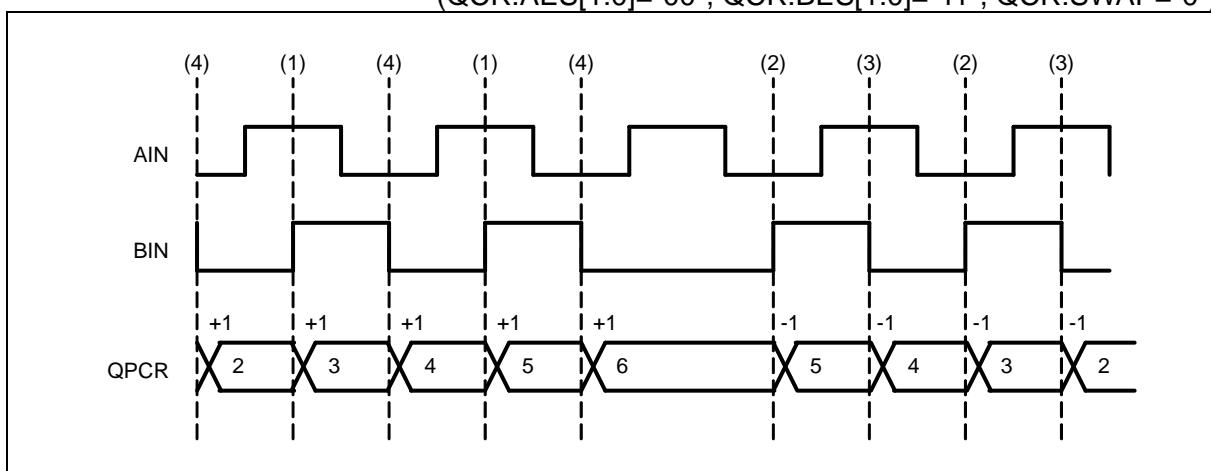
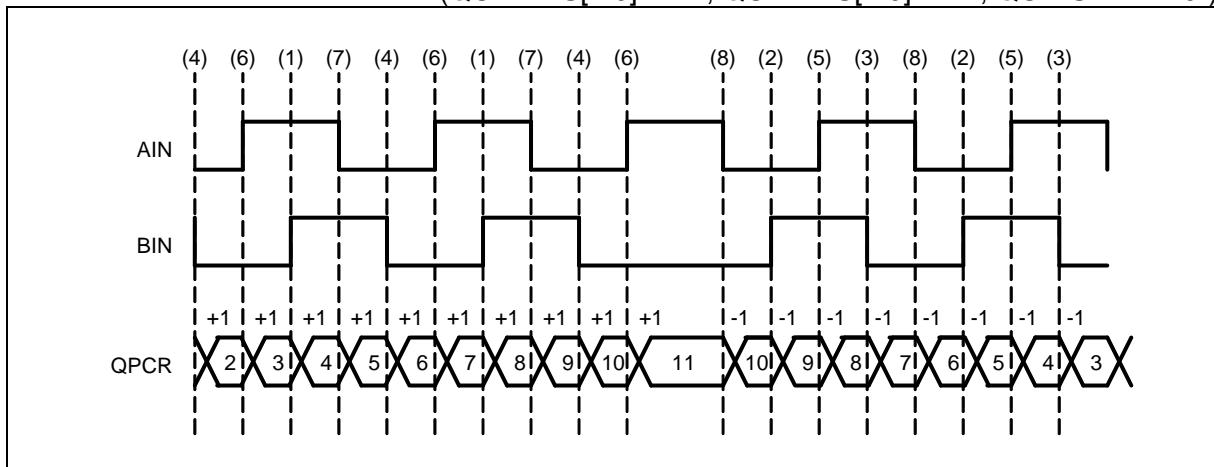


Table 3-4 Counting in 4-time frequency multiplication mode
 (QCR:AES[1:0] = "11", QCR:BES[1:0] = "11")

Edge detection pin	Detection edge	Level Check pin	Input level	Counting direction	Figure 3-3 Timing
BIN	Rising edge	AIN	High	Up	(1)
	Rising edge		Low	Down	(2)
	Falling edge		High	Down	(3)
	Falling edge		Low	Up	(4)
AIN	Rising edge	BIN	High	Down	(5)
	Rising edge		Low	Up	(6)
	Falling edge		High	Up	(7)
	Falling edge		Low	Down	(8)

Figure 3-3 Operation in 4-time frequency multiplication mode
 (QCR:AES[1:0] = "11", QCR:BES[1:0] = "11", QCR:SWAP = "0")



● **PC_Mode3: Count mode with direction**

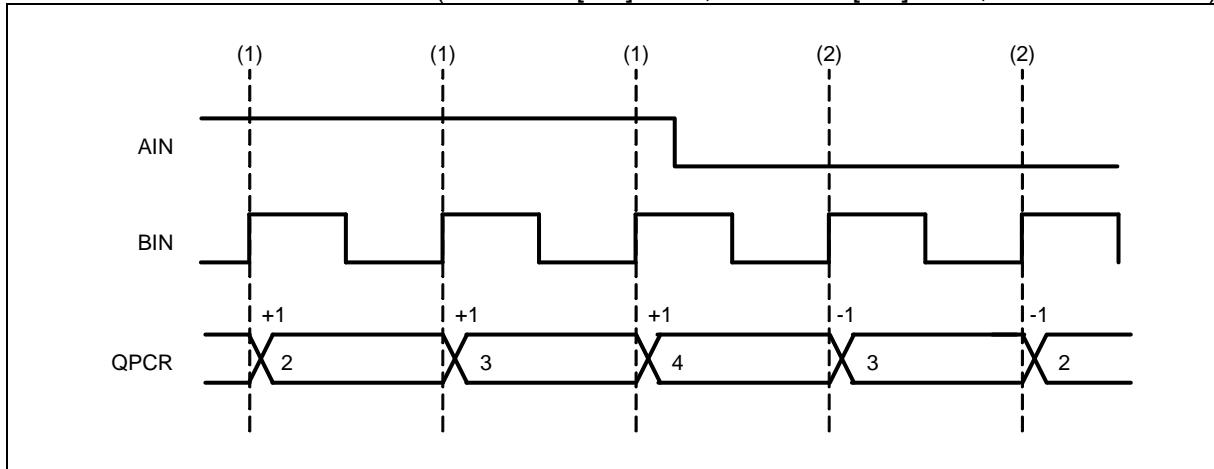
- A signal entered from the BIN external pin is received as the counting clock, and an input level of the signal entered from the AIN external pin is used for count direction control for counter up/down counting.
- In this mode, when an active edge of BIN signal is detected, the AIN signal level is checked and the position counter counted up or down. A rising edge, a falling edge, or both can be set as the active edge.

Table 3-5 Counting in the direction control counting mode

Edge detection pin	Detection edge	Level Check pin	Input level	Counting direction	Figure 3-4 Timing
BIN	Active edge	AIN	High	Up	(1)
	Active edge		Low	Down	(2)

Figure 3-4 Operation in the direction control counting mode

(QCR:AES[1:0] = "00", QCR:BES[1:0] = "10", QCR:SWAP = "0")



■ Operation of revolution counter

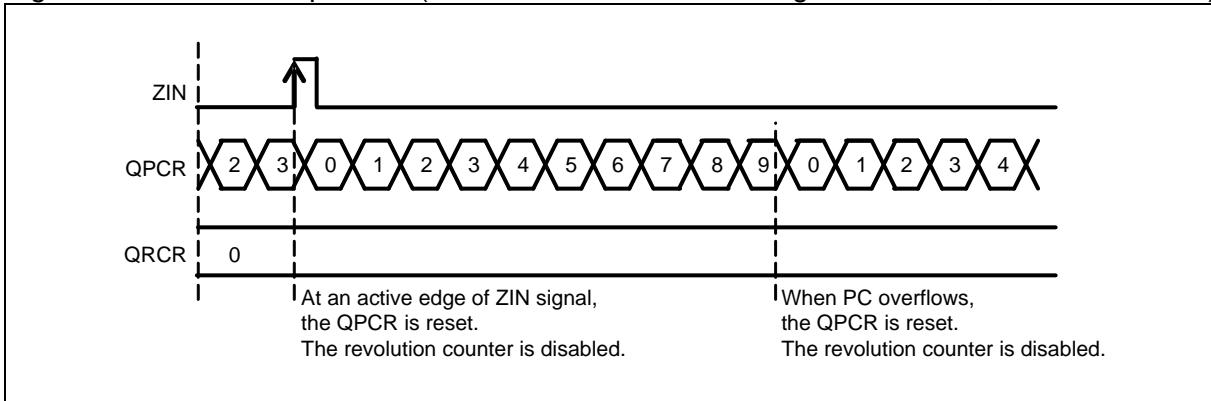
When the revolution counter receives an input from the ZIN pin (having the counter clear function) or an output of position counter (underflow or overflow), it is counted up or down. A rising edge, a falling edge, or both can be set as the active edge of ZIN signal.

The counting conditions of revolution counter depend on the selected mode as follows.

● RC_Mode0 (QCR:RCM[1:0] = "00")

- The revolution counter is disabled.
- When the ZIN signal is used for counter clear function (QCR:CGSC = "0"), the active edge of ZIN signal is reset. Also, the position counter is reset when this counter overflows.

Figure 3-5 RC_Mode0 operation (QPRC Maximum Position Register QMPR=9, QCR:CGSC = "0")



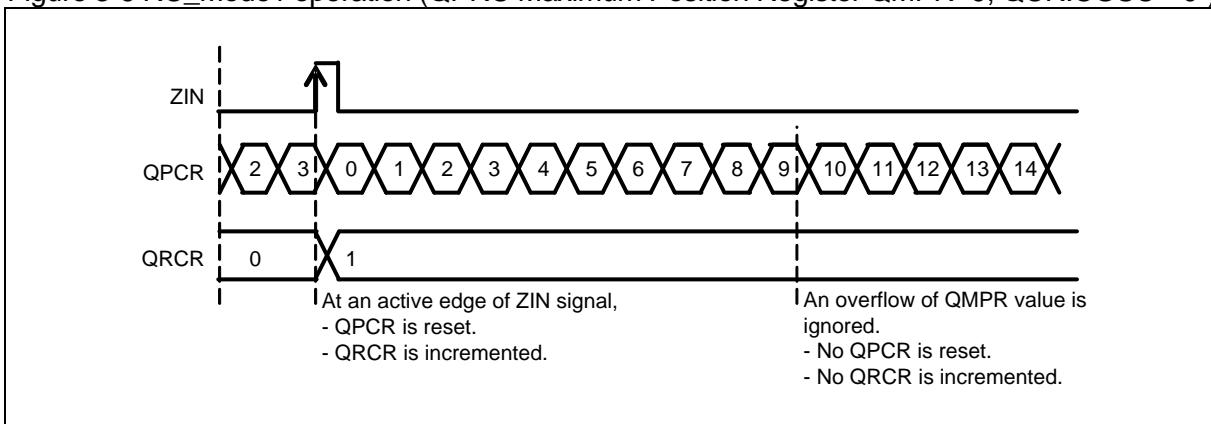
QPCR: QPRC Position Count Register

QRCCR: QPRC Revolution Count Register

● RC_Mode1 (QCR:RCM[1:0] = "01")

- When ZIN signal is used for the counter clear function (QCR:CGSC = "0"), the revolution counter is operated only an active edge of ZIN signal (but an input from the position counter is ignored).
- When an active edge of ZIN signal is detected during incrementing of position counter (QICR:DIRPC = "0"), the revolution counter is counted up. When an active edge of ZIN is detected during decrementing of position counter (QICR:DIRPC = "1"), it is counted down.
- When the ZIN signal is used for counter clear function (QCR:CGSC = "0"), the position counter is reset only at an active edge of ZIN signal.
- The position counter is not reset even when an overflow of position counter is detected. When an overflow of position counter is detected, the position counter is counted up and the overflow flag (QICR:OFDF) is set to logical 1.

Figure 3-6 RC_Mode1 operation (QPRC Maximum Position Register QMPR=9, QCR:CGSC = "0")



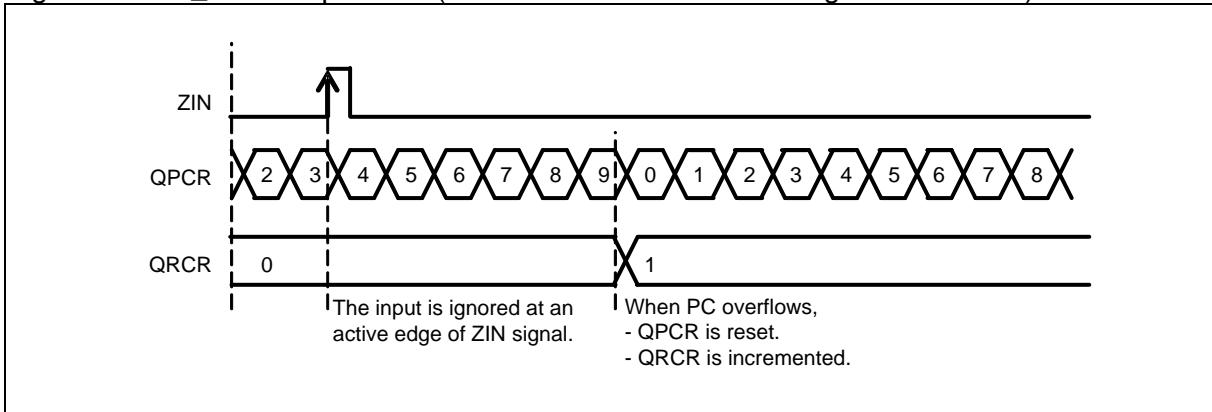
<Notes>

- When an active edge of ZIN signal and an active edge which counts down position counter are detected at the same time during incrementing of position counter (QICR:DIRPC="0"), the revolution counter is counted down.
 - When an active edge of ZIN signal and an active edge which counts up position counter are detected at the same time during decrementing of position counter (QICR:DIRPC="1"), the revolution counter is counted up.
 - When an active edge of ZIN signal, an active edge of AIN signal, and an active edge of BIN signal are detected at the same time, the revolution counter is counted up or down in accordance with the last position counter direction bit (QICR:DIRPC).

- RC_Mode2 (QCR:RCM[1:0] = "10")

- The revolution counter is counted up or down only by the output value of position counter.
 - The position counter is reset only when an overflow of position counter is detected (but an event of ZIN signal is ignored).
 - If an overflow of position counter is detected in any of 3 position counter modes (PC_Mode1, PC_Mode2 and PC_Mode3), the position counter is counted up. If an underflow of it is detected, the position counter is counted down.

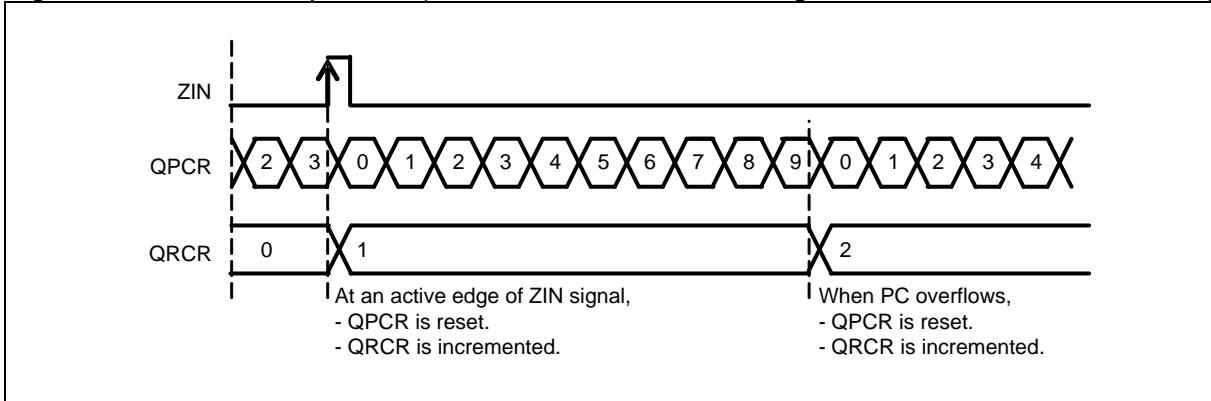
Figure 3-7 RC_Mode2 operation (QPRC Maximum Position Register QMPR=9)



● RC_Mode3 (QCR:RCM[1:0] = "11")

- In this mode, the revolution counter is operated with an output of position counter and when the ZIN signal is used for the counter clear function (QCR:CGSC="0"), the revolution counter is also counted up or down at an active edge of ZIN signal.
 - When an active edge of ZIN signal is detected during incrementing of position counter (QICR:DIRPC="0") or when an overflow of position counter is detected, the revolution counter is counted up.
 - When an active edge of ZIN signal is detected during decrementing of position counter (QICR:DIRPC="1") or when an underflow of position counter is detected, the revolution counter is counted down.
 - When the ZIN signal is used for the counter clear function (QCR:CGSC="0"), the position counter is reset at an active edge of ZIN signal or at detection of position counter overflow.

Figure 3-8 RC_Mode3 operation (QPRC Maximum Position Register QMPR=9, QCR:CGSC="0")



<Notes>

- When an active edge of ZIN signal and an active edge which counts down position counter are detected at the same time during incrementing of position counter (QICR:DIRPC="0"), the revolution counter is counted down.
- When an active edge of ZIN signal and an active edge which counts up position counter are detected at the same time during decrementing of position counter (QICR:DIRPC="1"), the revolution counter is counted up.
- When an active edge of ZIN signal, an active edge of AIN signal, and an active edge of BIN signal are detected at the same time, the revolution counter is counted up or down in accordance with the last position counter direction bit (QICR:DIRPC).

■ Absolute value of positions

In RC_Mode2 and 3 mode (when the revolution counter operates with an output of position counter), each position has the following absolute value.

QPRC Position Count Register (QPCR) + QPRC Revolution Count Register (QRCCR) × (QPRC Maximum Position Register (QMPR) +1)

Example: Time measurement

The revolution counter counts the "hours", and the position counter counts the "minutes".

If QMPR="59", QPCR="20", and QRCCR="5"

$$\text{Time} = 20 + 5 \times (59 + 1)$$

$$= 320 \text{ minutes.}$$

This is the absolute value in position counter units (minutes).

■ Quad Position & Revolution Counter interrupts

The following table defines the conditions where an interrupt request of Quad Position & Revolution Counter can generate.

Table 3-6 Generation conditions of Quad Position & Revolution Counter interrupt requests

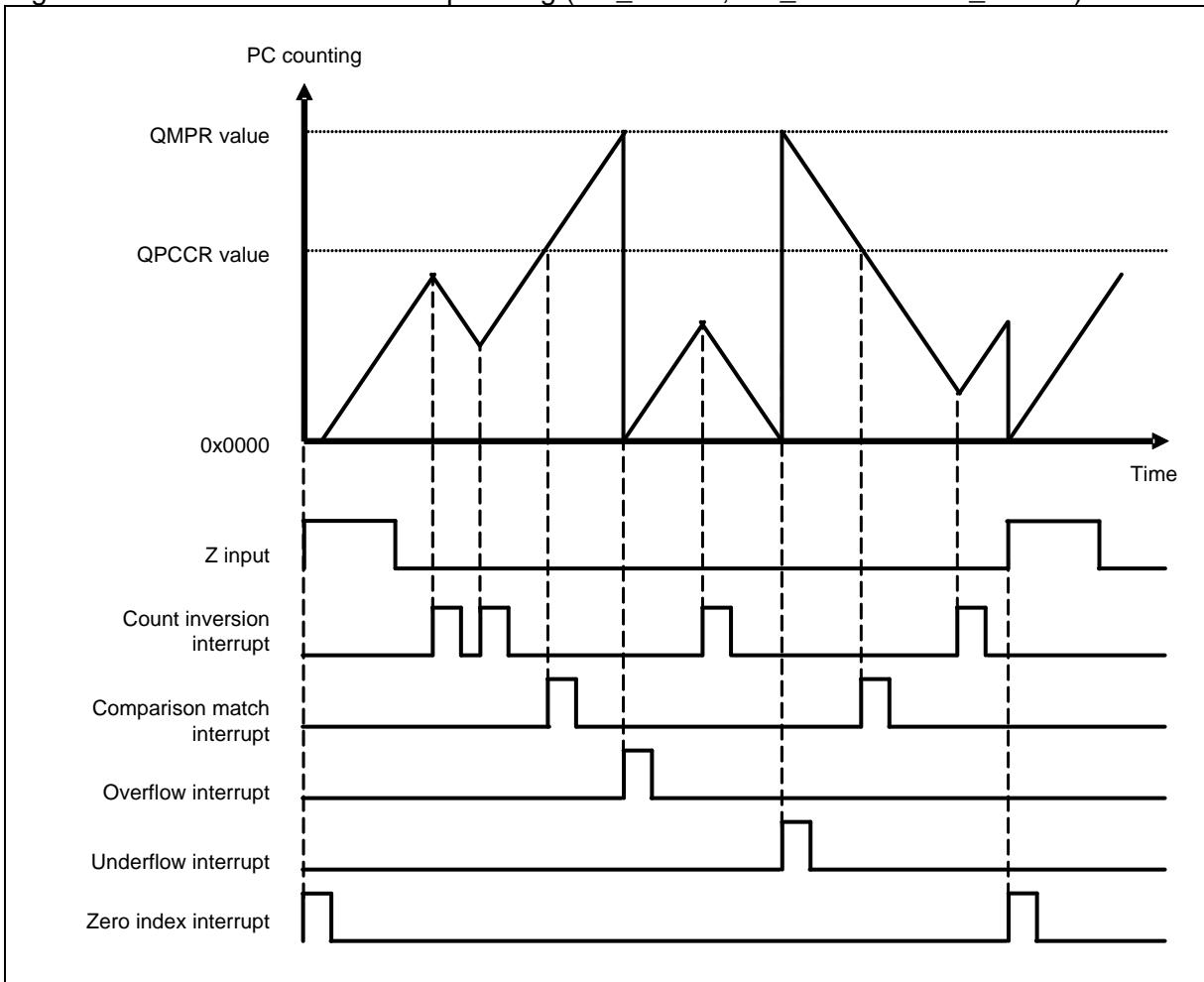
Interrupt request	Interrupt request flag	Interrupt request is enabled if	Interrupt request is cleared if
Count inversion interrupt request	QICR: CDCF="1"	QICR: CDCIE="1"	QICR: CDCF is set to "0".
Zero index interrupt request	QICR: ZIIF="1"		QICR: ZIIF is set to "0".
Overflow interrupt request	QICR: OFDF="1"	QICR: OUZIE="1"	QICR: OFDF is set to "0".
Underflow interrupt request	QICR: UFDF="1"		QICR: UFDF is set to "0".
PC and RC match interrupt request	QICR: QPRCMF="1"	QICR: QPRCMIE="1"	QICR: QPRCMF is set to "0".
PC match interrupt request	QICR: QPCMFI="1"	QICR: QPCMIE="1"	QICR: QPCMFI is set to "0".
PC match and RC match interrupt request	QICR: QPCNRCMF="1"	QICR: QPCNRCMIE="1"	QICR: QPCNRCMF is set to "0".
Outrange interrupt request	QEGR: ORNGF="1"	QICR: ORNGIE="1"	QEGR: QRNGF is set to "0".

QICR: QPRC Interrupt Control Register

QEGR: QPRC Extension Control Register

■ Interrupts of position counter

Figure 3-9 Position counter interrupt timing (RC_Mode0, RC_Mode2 or RC_Mode3)



QPCCR: QPRC Position Counter Compare Register

■ Operation example of QPRC Maximum Position Register (QMPR) interrupt

The QPRC Maximum Position Register (QMPR) value is used as the reload data to the position counter when an overflow or underflow of position counter is detected.

When the position counter value matches the QPRC Maximum Position Register (QMPR) value, the operation of the revolution counter depends on the selected mode as follows:

- When the position counter is counted up in RC_Mode0 (QCR:RCM[1:0] = "00"), RC_Mode2 (QCR:RCM[1:0] = "10") or RC_Mode3 (QCR:RCM[1:0] = "11"), the overflow flag (QICR:OFDF) is set to "1" and the position counter is reset.
- When the position counter is counted up in RC_Mode1 (QCR:RCM[1:0] = "01"), the overflow flag (QICR:OFDF) is set to "1". During this time, the position counter is not reset but is counted up.

The following gives an operation example where the QPRC Maximum Position Register (QMPR) is used in RC_Mode2 (QCR:RCM[1:0] = "10").

During counting up

When the position counter maximum value overflows to "0x0000", the revolution counter is counted up. During this time, the overflow flag (QICRL:OFDF) is set to logical 1.

Example: If the QPRC Maximum Position Register (QMPR) is set to 18

Position counter	15	16	17	18	0	1	2
Revolution counter	1	1	1	1	2	2	2

During counting down

When an underflow is detected with "0x0000" and when the value of Quad Counter Maximum Position Counter Register (QMPR) is reloaded to the position counter, the revolution counter is counted down. During this time, the underflow flag (QICRL:UFDF) is set to logical 1.

Example: If the QPRC Maximum Position Register (QMPR) is set to 5

Position counter	4	3	2	1	0	5	4	3	2	1	0	5
Revolution counter	1	1	1	1	1	0	0	0	0	0	0	0xFFFF

<Note>

The counting direction of position counter depends on the AIN and BIN external input signals only.

■ Position counter reset mask function

The position counter reset mask function can be used only when RC_Mode0 (QCR:RCM[1:0] = "00") or RC_Mode3 (QCR:RCM[1:0] = "11") is selected. This function operates regardless of setting of the position counter mode (PC_Mode1, PC_Mode2 or PC_Mode3).

The position counter reset mask function is executed in the following sequence.

1. If an active event of ZIN signal, an overflow of position counter, or an underflow of position counter are detected, a value being set by the position counter reset mask bits (QCR:PCRM[1:0]) is set to the mask counter (*1).
2. When the position counter is counted up or down in the same counting direction, the mask counter (*1) is counted down.
The position counter is reset only when the mask counter (*1) is set to "0x0". Also, the revolution counter is not counted up or down.
When a count inversion of the position counter is detected, the mask counter (*1) is set to "0x0".
3. If the mask counter (*1) is set to "0x0", the position counter is set to "0x0000" when an active edge of ZIN signal or an overflow of position counter is detected.

*1 : The number of times to mask both the reset of position counter and the counting up/down of revolution counter is counted. The masking continues until this counter value reaches "0x0".

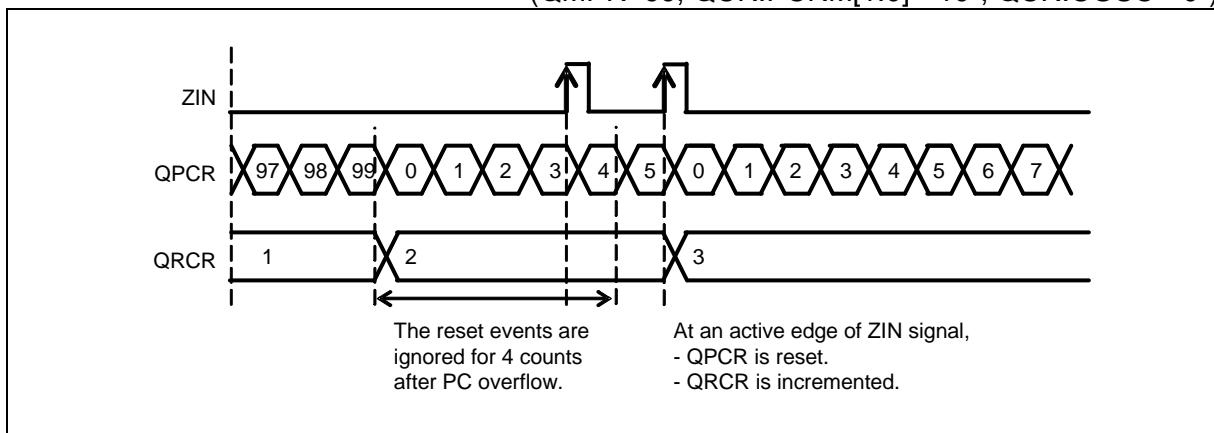
The following gives an operation example where the position counter reset mask function is used in RC_Mode3 (QCR:RCM[1:0] = "11").

Example 1:

An active edge of ZIN signal is ignored for four (4) counts of position counter after occurrence of position counter overflow.

Figure 3-10 Position counter reset mask operation example 1

(QMPR=99, QCR:PCRM[1:0] = "10", QCR:CGSC = "0")

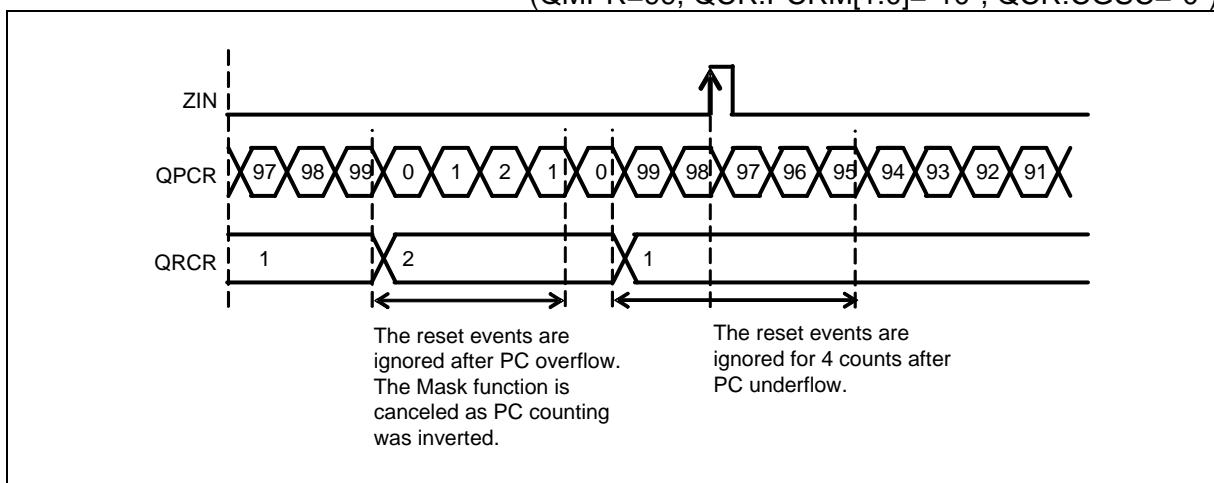


Example 2:

An active edge of ZIN signal is ignored for four (4) counts of position counter after count inversion of position counter.

Figure 3-11 Position counter reset mask operation example 2

(QMPR=99, QCR:PCRM[1:0] = "10", QCR:CGSC = "0")



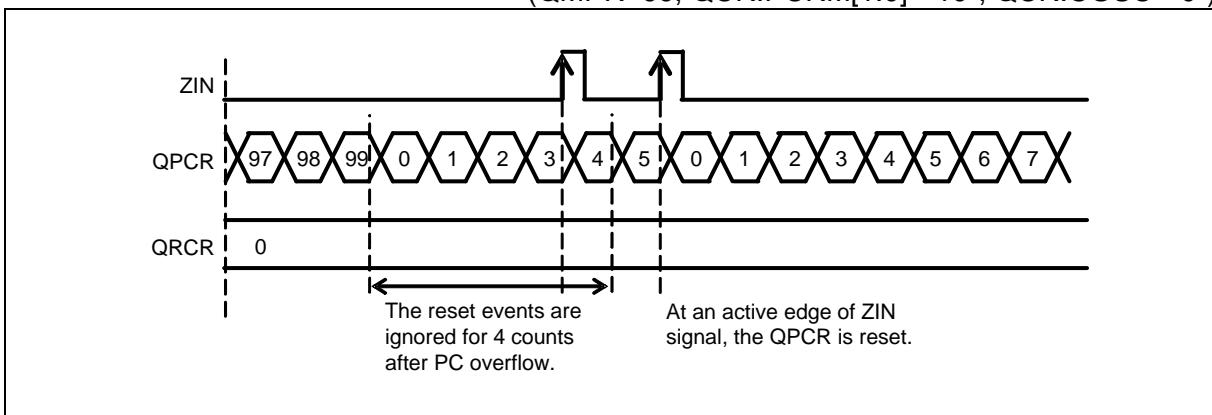
The following gives an operation example where the position counter reset mask function is used in RC_Mode0 (QCR:RCM[1:0] = "00").

Example 3:

An active edge of ZIN signal is ignored for four (4) counts of position counter after occurrence of position counter overflow if the revolution counter is disabled.

Figure 3-12 Position counter reset mask operation example 3

(QMPR=99, QCR:PCRM[1:0] = "10", QCR:CGSC = "0")



<Notes>

- While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions.
 - When the position counter mode bit (QCR:PCM[1:0]) is changed
 - When the revolution counter mode bit (QCR:RCM[1:0]) is changed
 - When the direction of the position counter is changed
- Even if an overflow or underflow of the position counter occurs without inversion of the position counter while the position counter reset mask function is operating in RC_Mode0 (QCR:RCM[1:0] = "00") or RC_Mode3 (QCR:RCM[1:0] = "11"), the revolution counter is not counted up or down. However, if an overflow occurs, the position counter becomes "0". If an underflow occurs, the QMPR is reloaded to the position counter. The overflow interrupt request flag bit (QICR:OFDF) or the underflow interrupt request flag bit (QICR:UFDF) is set to "1".

4. Registers

This section explains the configuration and functions of the registers used for the Quad Position & Revolution Counter (QPRC).

■ List of Quad Position & Revolution Counter registers

Abbreviation	Register name	See
QPCR	Quad Position & Revolution Counter Position Count Register	4.1
QRCR	QPRC Revolution Count Register	4.2
QPCCR	QPRC Position Counter Compare Register	4.3
QPRCR	QPRC Position and Revolution Counter Compare Register	4.4
QCR	QPRC Control Register	4.5
QECSR	QPRC Extension Control Register	4.6
QICRL	Low-Order Bytes of QPRC Interrupt Control Register	4.7
QICRH	High-Order Bytes of QPRC Interrupt Control Register	4.8
QMPCR	QPRC Maximum Position Register	4.9

4.1. Quad Position & Revolution Counter Position Count Register (QPCR)

The Quad Position & Revolution Counter Position Count Register (QPCR) indicates the position counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPCR[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] QPCR:

Reading this register reads out the current value of the position counter. While the position counter stops counting (QCR:PSTP="1"), the count value can be written to this register.

This register is set to 0x0000 in the following one of conditions.

- Reset
- A ZIN active edge is detected in the following conditions.
 - The ZIN function is set to the counter clear function (QCR:CGSC="0") in RC_Mode1 (QCR:RCM[1:0]="01").
 - After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is valid (QCR:PCRM[1:0]="01" or "10" or "11") in RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11")
 - The ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is invalid (QCR:PCRM[1:0]="00") in RC_Mode0(QCR:RCM[1:0]="00") or RC_Mode3(QCR:RCM[1:0]="11").
- A position counter overflow is detected in the following conditions.
 - RC_Mode2(QCR:RCM[1:0]="10")
 - After the position counter has been incremented or decremented by the mask set value when the count inversion of the position counter is not detected where the ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is valid (QCR:PCRM[1:0]="01" or "10" or "11") in RC_Mode0 (QCR:RCM[1:0]="00") or RC_Mode3 (QCR:RCM[1:0]="11")
 - The ZIN function is set to the counter clear function (QCR:CGSC="0") and the reset mask function of the position counter is invalid (QCR:PCRM[1:0]="00") in RC_Mode0(QCR:RCM[1:0]="00") or RC_Mode3(QCR:RCM[1:0]="11").
- 0x0000 is written to this QPCR while the position counter is under suspension (QCR:PSTP="1").

The value of the QPRC Maximum Position Register (QMPR) is set to this register in the following condition.

- A position counter underflow is detected.

<Notes>

- Do not access the Quad Position & Revolution Counter Position Count Register (QPCR) with a byte access instruction.
 - After the count value was written to the Quad Position & Revolution Counter Position Count Register (QPCR) while the position counter was under suspension (QCR:PSTP="1") in RC_Mode0 (QCR:RCM[1:0] = "00"), RC_Mode1 (QCR:RCM[1:0] = "01"), or RC_Mode3 (QCR:RCM[1:0] = "11"), if a ZIN active edge is detected with the count function (QCR:CGSC = "0"), the Quad Position & Revolution Counter Position Count Register (QPCR) will be set to 0x0000.
To write the count value to the Quad Position & Revolution Counter Position Count Register (QPCR), make the ZIN detection edge invalid (QCR:CGE[1:0] = "00") before writing it to the QPCR.
-

4.2. QPRC Revolution Count Register (QRCR)

The QPRC Revolution Count Register (QRCR) indicates the revolution counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QRCR[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] QRCR:

Reading this register reads out the current value of the revolution counter. While the revolution counter stops counting (QCR:RCM[1:0] = "00"), the count value can be written to this register.

This register is set to 0x0000 in the following one of conditions.

- Reset
- 0x0000 is written to this register while the revolution counter is under suspension (QCR.RCM[1:0] = "00").

<Notes>

- Do not access the QPRC Revolution Count Register (QRCR) with a byte access instruction.
- As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0] = "00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (QCR:PCM[1:0] = "00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.
 - The position counter is reset if the mode is RC_Mode0 (QCR:RCM[1:0] = "00"), RC_Mode1 (QCR:RCM[1:0] = "01"), or RC_Mode3 (QCR:RCM[1:0] = "11")
 - The revolution counter is not counted up or down

4.3. QPRC Position Counter Compare Register (QPCCR)

The QPRC Position Counter Compare Register (QPCCR) is used to compare with the count value of the position counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPCCR[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] QPCCR:

If the value of this register matches that of the position counter, the QPRC position counter comparison match flag (QICR:QPCMCF) is set to "1". This Compare Register can be used only to compare with the count value of the position counter.

<Note>

Do not access the QPRC Position Counter Compare Register (QPCCR) with a byte access instruction.

4.4. QPRC Position and Revolution Counter Compare Register (QPRCR)

The QPRC Position and Revolution Counter Compare Register (QPRCR) is used to compare with the selected count value of the position or revolution counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QPRCR[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] QPRCR:

Select whether to compare with the count value of the position or revolution counter using the RSEL bit of the QPRC Control Register (QCR). If the value of this register matches that of the position or revolution counter, the QPRC position and revolution counter comparison match flag (QICR:QPRCMF) is set to "1".

<Note>

Do not access the QPRC Position and Revolution Counter Compare Register (QPRCR) with a byte access instruction.

4.5. QPRC Control Register (QCR)

The QPRC Control Register (QCR) is used to specify the operation mode of the position counter or 16-bit revolution counter. It is also used to start or stop each counter.

■ Low-Order Bytes of QPRC Control Register (QCRL)

bit	7	6	5	4	3	2	1	0
Field	SWAP	RSEL	CGSC	PSTP	RCM1	RCM0	PCM1	PCM0
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	0

[bit 7] SWAP: Swap bit

This bit is used to swap the connections of the AIN input and BIN input to the position counter.

When this bit is set to "0", the AIN pin is used for the AIN input of the position counter, and the BIN pin is used for the BIN input of the position counter. When this bit is set to "1", the AIN pin is used for the BIN input of the position counter, and the BIN pin is used for the AIN input of the position counter.

Bit	Description
0	No swap
1	Swaps AIN and BIN inputs.

<Note>

Change the swap bit (SWAP) when the position counter is disabled (PCM[1:0]="00").

[bit 6] RSEL: Register function selection bit

This bit is used to select whether to compare the value of the QPRC Position and Revolution Counter Compare Register (QPRCCR) with that of the position or revolution counter.

Bit	Description
0	Compares the value of the QPRC Position and Revolution Counter Compare Register (QPRCCR) with that of the position counter.
1	Compares the value of the QPRC Position and Revolution Counter Compare Register (QPRCCR) with that of the revolution counter.

<Note>

When the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and also the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCCR), the PC match and RC match interrupt request flag bit (QICR: QPCNRCMF) is set to "1" regardless of the setting of this bit.

[bit 5] CGSC: Count clear or gate selection bit

This bit is used to select the function of the ZIN external pin.

When the counter clear function is enabled (QGSC="0"), the ZIN pin clears the position counter if the revolution count mode is set to RC_Mode0 (RCM[1:0]="00"), RC_Mode1 (RCM[1:0]="01"), or RC_Mode3 (RCM[1:0]="11"). The CGE1 and CGE0 bits of the QCR register clear the position counter by selecting a valid edge of the ZIN pin and detecting the selected edge.

When the gate function is enabled (QGSC="1"), the ZIN pin controls the count operation of the position counter. The CGE1 and CGE0 bits of the QCR register count the position counter at the valid level of the ZIN pin.

Bit	Description
0	Counter clear function
1	Gate function

[bit 4] PSTP: Position counter stop bit

This bit is used to stop the position counter.

Bit	Description
0	Enables count operation.
1	Stops count operation.

[bit 3:2] RCM1, RCM0: Revolution counter mode bits

These bits are used to select the count mode of the revolution counter and the reset mode of the position counter. For the effect on the position counter, see "Operation of revolution counter".

bit3	bit2	Description
0	0	Disables the revolution counter (RC_Mode0).
0	1	The revolution counter is counted up or down only with a ZIN active edge (RC_Mode1).
1	0	The revolution counter is counted up or down only when overflow or underflow is detected in the position counter that matches QMPR (RC_Mode2).
1	1	The revolution counter is counted up or down in two cases: a position counter overflow or underflow is detected and a ZIN active edge is detected (RC_Mode3).

[bit 1:0] PCM1, PCM0: Position counter mode bits

These bits are used to select the count mode of the position counter.

bit1	bit0	Description
0	0	Disables the position counter (PC_Mode0) to stop it.
0	1	Up-down count mode (PC_Mode1) Increments the value with an AIN active edge and decrements it with a BIN active edge.
1	0	Phase difference count mode (PC_Mode2) Counts up if AIN is leading BIN and down if BIN is leading AIN.
1	1	Directional count mode (PC_Mode3) Counts up or down with the BIN active edge and AIN level.

<Note>

As the direction of the position counter is not detected in PC_Mode0 (PCM[1:0] = "00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (PCM[1:0] = "00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.

- The position counter is reset if the mode is RC_Mode0 (RCM[1:0] = "00"), RC_Mode1 (RCM[1:0] = "01"), or RC_Mode3 (RCM[1:0] = "11")
- The revolution counter is not counted up or down

■ High-Order Bytes of QPRC Control Register (QCRH)

bit	15	14	13	12	11	10	9	8
Field	CGE1	CGE0	BES1	BES0	AES1	AES0	PCRM1	PCRM0
Attribute	R/W	R/W						
Initial value	0	0	0	0	0	0	0	0

[bit 15:14] CGE1, CGE0: Detection edge selection bits

These bits are used to select the detection edge when the ZIN external pin is used for the counter clear function (CGSC="0"). They are also used to select the detection level when the ZIN external pin is used for the gate function (CGSC="1").

bit15	bit14	ZIN used for counter clear function (CGSC="0")	ZIN used for gate function (CGSC="0")
0	0	Disables edge detection.	Disables level detection.
0	1	Detects a falling edge.	Detects level "L".
1	0	Detects a rising edge.	Detects level "H".
1	1	Detects a rising or falling edge.	Disables level detection.

[bit 13:12] BES1, BES0: BIN detection edge selection bits

These bits are used to select the detection edge of the BIN external pin.

bit13	bit12	Description
0	0	Disables edge detection.
0	1	Detects a falling edge.
1	0	Detects a rising edge.
1	1	Detects rising and falling edges.

[bit 11:10] AES1, AES0: AIN detection edge selection bits

These bits are used to select the detection edge of the AIN external pin.

bit11	bit10	Description
0	0	Disables edge detection.
0	1	Detects a falling edge.
1	0	Detects a rising edge.
1	1	Detects rising and falling edges.

[bit 9:8] PCRM1, PCRM0: Position counter reset mask bits

These bits are used to specify the period (mask time) to ignore the events shown below after detecting a position counter overflow or underflow or detecting a ZIN active edge.

- Position counter resetting
- Revolution counter increment or decrement

This mask function is released when the count direction of the position counter is changed, and restarts when a position counter overflow or underflow is detected or a ZIN active edge is detected.

bit9	bit8	Description
0	0	No reset mask
0	1	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes twice.
1	0	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes four times.
1	1	The position counter reset or a revolution counter count-up or -down events are ignored until the position counter changes eight times.

<Notes>

- The position counter reset mask function is available only in RC_Mode0 (RCM[1:0] = "00") and RC_Mode3 (RCM[1:0] = "11"). This function operates regardless of the setting of the position counter mode (PC_Mode1, PC_Mode2, or PC_Mode3).
- While the position counter reset mask function is operating, the mask function is released and the position counter can be reset in the following conditions.
 - When the position counter mode bit (PCM[1:0]) is changed
 - When the revolution counter mode bit (RCM[1:0]) is changed
 - When the direction of the position counter is changed

4.6. QPRC Extension Control Register (QEGR)

The QPRC Extension Control Register (QEGR) is used to select that the revolution counter is inside the count range, indicate that the revolution counter is outside the count range, or control whether or not to generate an interrupt when the revolution counter gets out of the range.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												ORNGIE	ORNGF	ORNGMD	
Attribute	-	-	-	-	-	-	-	-	-	-	-	-	-	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:3] Reserved bit

Always write "0" to these bits. The read value is "0".

[bit 2] ORNGIE: Outrange interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the outrange interrupt request flag (ORNGF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the revolution counter gets out of the range (ORNGF="1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 1] ORNGF: Outrange interrupt request flag bit

This flag indicates that the revolution counter is outside the count range.

If a positive number is selected as the outrange mode of the revolution counter (ORNGMD="0"), this flag is set to "1" when the revolution counter changes from 0x0001 to 0x0000 after counting down or when it changes from 0xFFFF to 0xFFFF after counting up.

If the 8K value is selected as the outrange mode of the revolution counter (ORNGMD="1"), this flag is set to "1" when the revolution counter changes from 0x8001 to 0x8000 after counting down or when it changes from 0x7FFE to 0x7FFF after counting up.

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Out of range is not detected.	Clears this bit.
1	Out of range is detected.	No effect.

[bit 0] ORNGMD: Outrange mode selection bit

This bit defines the outrange mode of the revolution counter.

Bit	Description
0	Selects a positive number (in the range from 0x0000 to 0xFFFF).
1	Selects the 8K value (in the range from 0x0000 to 0x7FFF).

4.7. Low-Order Bytes of QPRC Interrupt Control Register (QICRL)

The Low-Order Bytes of QPRC Interrupt Control Register (QICRL) are used to control a position counter overflow or underflow interrupt, zero index interrupt, QPRC position counter comparison match interrupt, or QPRC position and revolution counter comparison match interrupt.

bit	7	6	5	4	3	2	1	0
Field	ZIIF	OFDF	UFDF	OUZIE	QPRCMF	QPRCMIE	QPCMIE	QPCMIE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] ZIIF: Zero index interrupt request flag bit

This flag is set to "1" when the position counter is reset by the ZIN input.

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no zero index.	Clears this bit.
1	Detects zero index.	No effect.

<Note>

The zero index interrupt request flag bit (ZIIF) is not set to "1" even if ZIN is used as the gate function (QCR:CGSC="1") or the position counter is reset in RC_Mode2 (QCR:RCM[1:0]="10").

[bit 6] OFDF: Overflow interrupt request flag bit

This flag indicates that a position counter overflow occurs. When the position counter is counted up, this bit is set to "1" if the value of the position counter matches that of the QPRC Maximum Position Register (QMPR).

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no overflow.	Clears this bit.
1	Detects overflow.	No effect.

[bit 5] UFDF: Underflow interrupt request flag bit

This flag indicates that a position counter underflow occurs. When the position counter is counted down, this bit is set to "1" if the position counter is 0x0000.

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no underflow.	Clears this bit.
1	Detects underflow.	No effect.

[bit 4] OUZIE: Overflow, underflow, or zero index interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the overflow interrupt request flag bit (OFDF), underflow interrupt request flag bit (UFDF), or zero index interrupt request flag bit (ZIIF) is set to "1". When this bit is set to "1", an interrupt is generated if overflow is detected (OFDF="1"), underflow is detected (UFDF="1"), or zero index is detected (ZIIF="1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 3] QPRCMF: PC and RC match interrupt request flag bit

This flag indicates whether the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) or the value of the revolution counter (QRCCR) matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).

When the comparison between the position counter and QPRC Position and Revolution Counter Compare Register (QPRCR) is selected (QCR:RSEL="0"), this flag is set to "1" if the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).

When the comparison between the revolution counter and QPRC Position and Revolution Counter Compare Register (QPRCR) is selected (QCR:RSEL="1"), this flag is set to "1" if the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no comparison match with the QPRCR value.	Clears this bit.
1	Detects a comparison match with the QPRCR value.	No effect.

<Notes>

- If the register function selection bit (QCR:RSEL) is set to "0", the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately when the following one of conditions is satisfied.
 - The mode is changed to PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11") when the position counter is disabled (QCR:PCM[1:0] = "00") and the value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).
 - The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when data is written to the Quad Position & Revolution Counter Position Count Register (QPCR) or QPRC Position and Revolution Counter Compare Register (QPRCR) in PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11").
- If the register function selection bit (QCR:RSEL) is set to "1", the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately when the following condition is satisfied.
 - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) by writing data to the QPRC Position and Revolution Counter Compare Register (QPRCR) when the mode is RC_Mode1 (QCR:RCM[1:0] = "01"), RC_Mode2 (QCR:RCM[1:0] = "10"), or RC_Mode3 (QCR:RCM[1:0] = "11").
 - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) by changing the mode from RC_Mode0 (QCR:RCM[1:0] = "00") to another mode.
- If the register function selection bit (QCR:RSEL) is changed, the PC and RC match interrupt request flag bit (QPRCMF) is set to "1" immediately if the following one of conditions is satisfied.
 - The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the register function selection bit (QCR:RSEL) is changed from "0" to "1" in the mode other than RC_Mode0 (QCR:RCM[1:0] = "00").
 - The value of the position counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the register function selection bit (QCR:RSEL) is changed from "1" to "0" in the mode other than RC_Mode0 (QCR:RCM[1:0] = "00").

[bit 2] QPRCMIE: PC and RC match interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC and RC match interrupt request flag (QPRCMF) is set to "1". When this bit is set to "1", an interrupt is generated if the value of the position or revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) (QPRCMF = "1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 1] QPCMFI: PC match interrupt request flag bit

This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR).

This flag is set to "1" if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR).

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no comparison match with the QPCCR value.	Clears this bit.
1	Detects a comparison match with the QPCCR value.	No effect.

<Note>

The PC match interrupt request flag bit (QPCMFI) is set to "1" immediately when the following one of conditions is satisfied.

- The mode is changed to PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11") when the position counter is disabled (QCR:PCM[1:0]="00") and the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR).
- The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) by writing to the Quad Position & Revolution Counter Position Count Register (QPCR) when the position counter stop bit (QCR:PSTP) is "1" and when the mode is PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11").
- The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) by writing to the QPRC Position Counter Compare Register (QPCCR) when the mode is PC_Mode1 (QCR:PCM[1:0]="01"), PC_Mode2 (QCR:PCM[1:0]="10"), or PC_Mode3 (QCR:PCM[1:0]="11").

[bit 0] QPCMIE: PC match interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC match interrupt request flag (QPCMFI) is set to "1".

When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) (QPCMFI="1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

4.8. High-Order Bytes of QPRC Interrupt Control Register (QICRH)

The High-Order Bytes of QPRC Interrupt Control Register (QICRH) are used to control a match between the position counter and QPCCR, a match between the revolution counter and QPRCR, and a count inversion interrupt. They are also used to indicate the direction of the position counter when the last underflow or overflow interrupt was detected or the last value of the position counter was changed.

bit	15	14	13	12	11	10	9	8
Field	Reserved		QPCNRCMF	QPCNRCMIE	DIROU	DIRPC	CDCF	CDCIE
Attribute	-	-	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 15:14] Reserved

Always write "0" to these bits. The read value is "0".

[bit 13] QPCNRCMF: PC match and RC match interrupt request flag bit

This flag indicates whether or not the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).

This flag is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) (QPCM=1") and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Detects no match.	Clears this bit.
1	Detects a match.	No effect.

<Note>

The PC match and RC match interrupt request flag bit (QPCNRCMF) is set to "1" immediately when the following one of conditions is satisfied.

- The mode is changed to PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11") when the position counter is disabled (QCR:PCM[1:0] = "00") and the revolution counter is in the mode other than RC_Mode0 (QCR:RCM[1:0] = "00") while the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR).
- The value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) when data is written to the Quad Position & Revolution Counter Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) where the value of the revolution counter matches that of the QPRC Position & Revolution Counter Compare Register (QPRCR) when the mode is PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11") and the revolution counter is in the mode other than RC_Mode0 (QCR:RCM[1:0] = "00").
- The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the data is written to the QPRC Position and Revolution Counter Compare Register (QPRCR) in the mode other than RC_Mode0 (QCR:RCM[1:0] = "00") where the specified value matches that of the QPRC Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) in PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11").
- The value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) when the mode is changed from RC_Mode0 (QCR:RCM[1:0] = "00") to another mode where the specified value matches that of the QPRC Position Count Register (QPCR) or QPRC Position Counter Compare Register (QPCCR) in PC_Mode1 (QCR:PCM[1:0] = "01"), PC_Mode2 (QCR:PCM[1:0] = "10"), or PC_Mode3 (QCR:PCM[1:0] = "11").
- This bit is set to "1" when the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) regardless of the setting of the register function selection bit (QCR:RSEL).

[bit 12] QPCNRCMIE: PC match and RC match interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the PC match and RC match interrupt request flag (QPCNRCMF) is set to "1".

When this bit is set to "1", an interrupt is generated if the value of the position counter matches that of the QPRC Position Counter Compare Register (QPCCR) and the value of the revolution counter matches that of the QPRC Position and Revolution Counter Compare Register (QPRCR) (QPCNRCMF = "1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 11] DIROU: Last position counter flow direction bit

This bit indicates the direction of the position counter when the last position counter overflow or underflow was detected.

Bit	Description
0	The position counter was incremented.
1	The position counter was decremented.

[bit 10] DIRPC: Last position counter direction bit

This bit indicates the count direction when the position counter was last changed.

Bit	Description
0	The position counter was incremented.
1	The position counter was decremented.

<Note>

As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, if the mode is changed from PC_Mode0 (QCR:PCM[1:0]="00") to another mode, when a ZIN active edge is detected before an AIN/BIN active edge is detected, the following operations apply.

- The position counter is reset if the mode is RC_Mode0 (QCR:RCM[1:0]="00"), RC_Mode1 (QCR:RCM[1:0]="01"), or RC_Mode3 (QCR:RCM[1:0]="11")
- The revolution counter is not counted up or down

[bit 9] CDCF: Count inversion interrupt request flag bit

This bit indicates whether or not the position counter inverted the count direction.

This bit is set to "1" when the position counter inverts the count direction. Inverting the count direction means that the counter counts down at the next counting after counting up, or the counter counts up at the next counting after counting down.

This flag can only clear to "0" in write mode. Setting "1" has no effect.

"1" is read by the read-modify-write access operation.

Bit	Description	
	Read	Write
0	Does not invert the count direction of the position counter.	Clears this bit.
1	Inverts the count direction of the position counter at least once.	No effect.

<Note>

As the direction of the position counter is not detected in PC_Mode0 (QCR:PCM[1:0]="00"), the last position counter direction bit (QICR:DIRPC) becomes indefinite. Therefore, after the mode is changed from PC_Mode0 (QCR:PCM[1:0]="00") to another mode, even if an AIN/BIN active edge is detected and the direction of the position counter is inverted, the count inversion interrupt request flag bit (QICR:CDCF) is not set to "1".

[bit 8] CDCIE: Count inversion interrupt enable bit

This bit is used to control whether or not to issue an interrupt notification to the CPU when the count inversion interrupt request flag (CDCF) is set to "1".

When this bit is set to "1", an interrupt is generated if the count direction of the position counter is inverted (CDCF="1").

Bit	Description
0	Interrupt disabled
1	Interrupt enabled

4.9. QPRC Maximum Position Register(QMPR)

The QPRC Maximum Position Register (QMPR) is used to specify the maximum value of the position counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QMPR[15:0]															
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[bit 15:0] QMPR:

When the position counter is counted up, a position counter overflow is detected (QICR:OFDF="1") if the set value of the QPRC Maximum Position Register (QMPR) matches the value of the position counter.

When the position counter is counted down, the set value of the QPRC Maximum Position Register (QMPR) is reloaded to the position counter if a position counter underflow is detected (QICR:UFDF="1").

<Note>

Do not access the QPRC Maximum Position Register (QMPR) with a byte access instruction.

CHAPTER: A/D Converter

This chapter explains the functions and operations of the A/D converter.

1. Configuration
2. Functions and Operations
3. Notes

1. Configuration

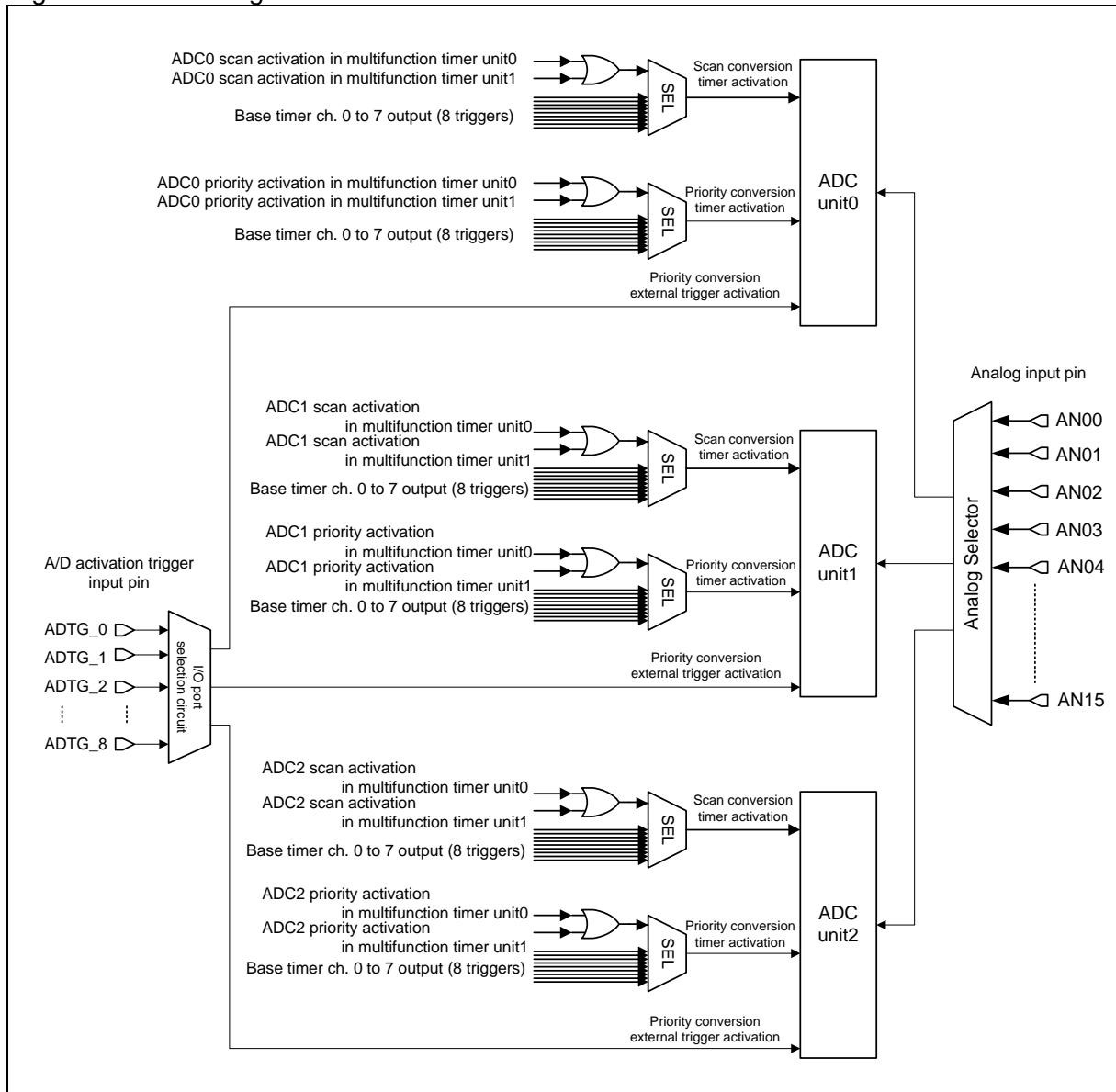
A/D converter converts analog input voltage from an external pin to a digital value.

■ A/D converter configuration

- 3 units of A/D converters with 10-bit resolution or 12-bit resolution are installed.
- Any channel can be selected to any unit from 16 channels of analog input.
- The following triggers can be selected as an activation trigger for A/D conversion.
 - Priority conversion activation trigger
 - Trigger input from an external pin
 - Timer trigger input (base timer or multifunction timer)
 - Software activation
 - Scan conversion activation trigger
 - Timer trigger input (base timer or multifunction timer)
 - Software activation

Figure 1-1 shows a connection diagram of the A/D converter with the related circuits.

Figure 1-1 Block diagram of the A/D converter with the related circuits



2. Functions and Operations

See descriptions of the following related chapters for functions and operations of the A/D converter.

■ 10-bit A/D converter operation

See the chapter of "10-bit A/D Converter" for conversion operations of 10-bit A/D converter.

■ 10-bit A/D timer trigger select operation

See the chapter of "A/D Timer Trigger Selection" for operations of 10-bit A/D converter timer trigger selection.

■ 12-bit A/D converter operation

See the chapter of "12-bit A/D Converter" for conversion operations of 12-bit A/D converter.

■ 12-bit A/D timer trigger select operation

See the chapter of "A/D Timer Trigger Selection" for operations of 12-bit A/D converter timer trigger selection.

3. Notes

This section shows the notes.

■ Notes on 10-bit A/D converter

- Simultaneous A/D conversion of 3 channels is possible because 3 units of A/D converters are installed.
Do not select the same input channel with the multiple units.
- Some channels of an analog input cannot be used for certain models. Do not change the selection registers (SCIS0, SCIS1, SCIS2, and SCIS3) and the sampling time selection registers (ADSS0, ADSS1, ADSS2, and ADSS3) for the channels which cannot be used from their initial values.
- In this series, P1A[2:0] of the priority conversion input selection register (PCIS) should be selected for an analog input channel during priority conversion. Always write "0" to ESCE bit of the priority conversion control register (PCCR) of the 10-bit A/D converter.
- DMA transfer using the A/D interrupt request generation of this series supports only DMA transfer using generation of a scan conversion interrupt request. DMA transfer using a priority conversion interrupt request is not supported.

■ Notes on 12-bit A/D converter

- Simultaneous A/D conversion of 3 channels is possible because 3 units of A/D converters are installed.
Do not select the same input channel with the multiple units.
- Some channels of an analog input cannot be used for certain models. Do not change the selection registers (SCIS0, SCIS1, SCIS2, and SCIS3) and the sampling time selection registers (ADSS0, ADSS1, ADSS2, and ADSS3) for the channels which cannot be used from their initial values.
- In this series, P1A[2:0] of the priority conversion input selection register (PCIS) should be selected for an analog input channel during priority conversion. Always write "0" to ESCE bit of the priority conversion control register (PCCR) of the 10-bit A/D converter.
- DMA transfer using the A/D interrupt request generation of this series supports only DMA transfer using generation of a scan conversion interrupt request. DMA transfer using a priority conversion interrupt request is not supported.

Chapter: 10-bit A/D Converter

This chapter explains the functions and operations of the 10-bit A/D converter.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup procedure examples
5. Registers

1. Overview

The 10-bit A/D converter is a function that converts analog input voltages into 10-bit digital values using a type of the RC Successive Approximation Register.

■ Features of the 10-bit A/D converter

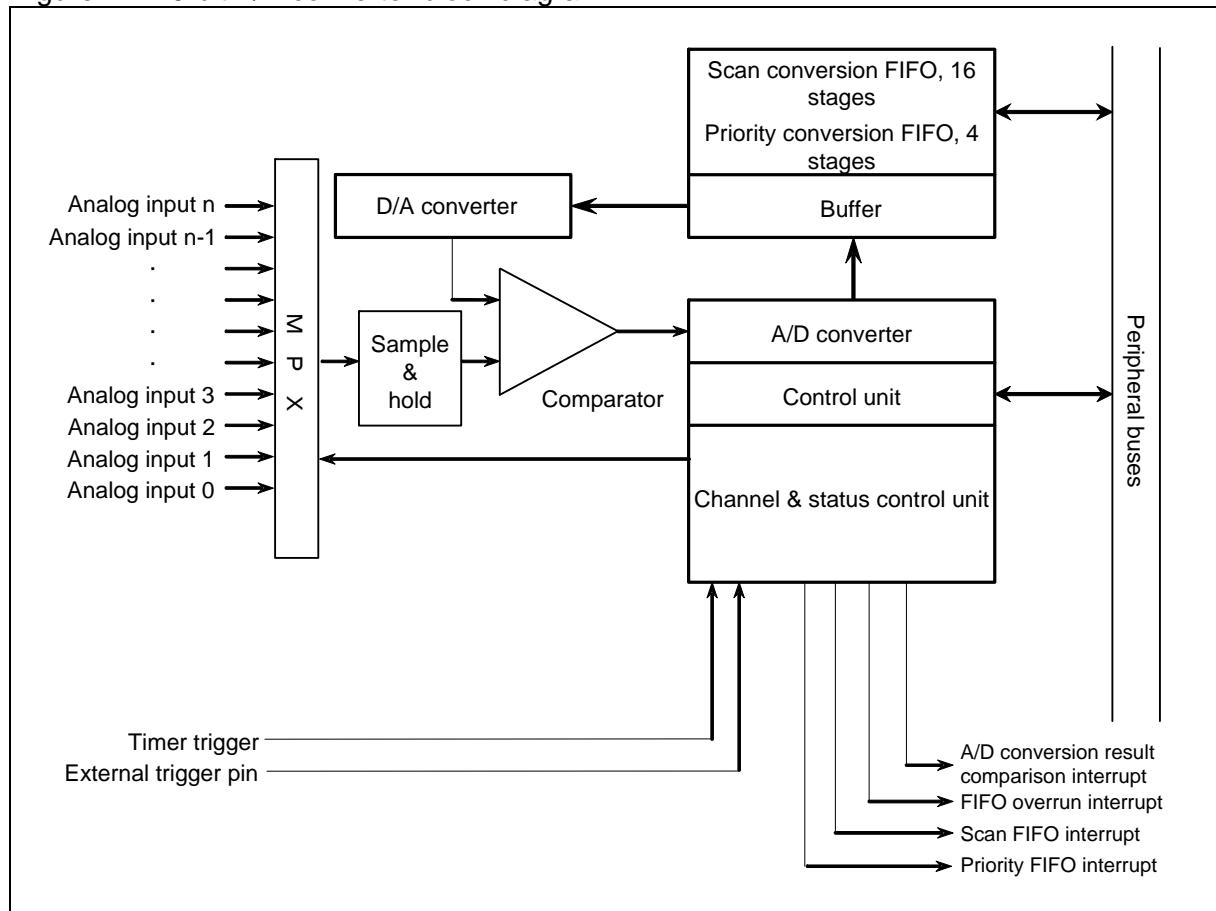
- 10-bit resolution
- Converter using a type of RC Successive Approximation Register with sample and hold circuits
- Conversion time of 1.2 µs (at a peripheral clock frequency of 30 MHz)
- Two sampling times selectable for each input channel
- Scan conversion operation:
 - Multiple analog inputs can be selected from multiple channels.
 - Start factors are software and timers.
 - Repeat mode is available.
- Priority conversion operation:
 - Even during scan operation, if a start factor of priority conversion occurs, it is possible to interrupt the ongoing scan conversion and perform conversion with high priority (There are two priority levels: 1 and 2. Priority level 1 is higher than priority level 2.).
 - Start factors are software and timers (priority level 2), and external triggers (priority level 1).
- FIFO function:
 - Sixteen FIFO stages for scan conversion and four FIFO stages for priority conversion are incorporated.
 - An interrupt is generated when data is written in the specified count of FIFO stages.
- Changeable A/D conversion data placement (selectable between shift to the MSB side and shift to LSB side)
- The A/D conversion result comparison function is available.
- There are four interrupt sources as follows:
 1. Scan conversion FIFO stage count interrupt
 2. Priority conversion FIFO stage count interrupt
 3. FIFO overrun interrupt (for both scan and priority conversion processes)
 4. A/D conversion result comparison interrupt
- DMA transfer triggered by an interrupt request

2. Configuration

This section provides the configuration of the 10-bit A/D converter.

■ 10-bit A/D converter block diagram

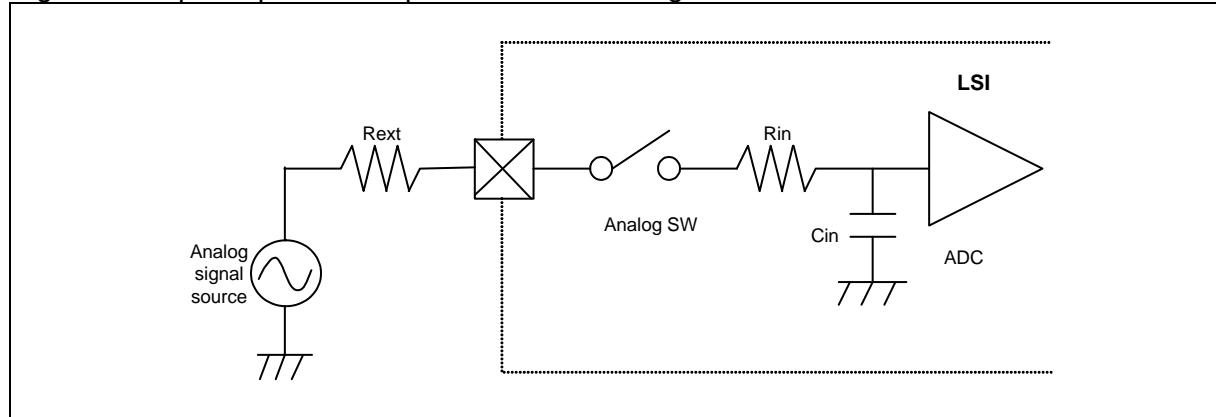
Figure 2-1 10-bit A/D converter block diagram



■ Input impedance

The sampling circuit of the A/D converter is shown as an equivalent circuit in Figure 2-2. Refer to the "Electrical Characteristics" in the data sheet to make sure that the external impedance R_{ext} should be selected not to exceed the sampling time.

Figure 2-2 Input impedance equivalence circuit diagram



3. Explanation of Operations

This chapter explains the operations of the 10-bit A/D converter.

- 3.1 A/D conversion operation
- 3.2 FIFO operations
- 3.3 A/D comparison function
- 3.4 Starting DMA

3.1. A/D conversion operation

The A/D converter can perform two types of conversion processes: scan conversion and priority conversion.

- 3.1.1 Scan conversion operation
- 3.1.2 Priority conversion operation
- 3.1.3 Priority levels and state transitions

3.1.1. Scan conversion operation

This section explains the scan conversion operation.

The input channels are selected in the Scan Conversion Input Selection Register (SCIS). By setting the corresponding bit in the SCIS to "1", any necessary channel can be selected from among multiple analog input channels.

The A/D converter can be started by software or a timer. To start the converter by software, set the SSTR bit in the SCCR register to "1". Then conversion starts. To start the converter by timers, set the SHEN bit in the SCCR register to "1" to enable timer start. Conversion starts when the timer's rising edge is detected. When conversion starts, the SCS bit in the ADSR register is set to "1". When the conversion is completed, the SCS bit is reset to "0".

When the SSTR bit in the SCCR register is set to "1" again during A/D conversion or the timer's rising edge is detected again while timer start is enabled, the ongoing conversion operation is immediately stopped and initialized and the A/D conversion is performed again (the operation is restarted).

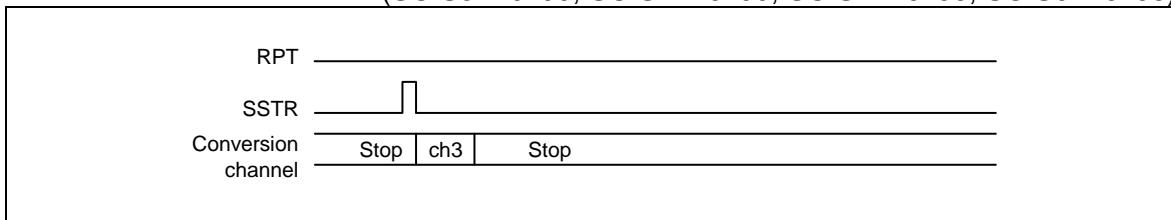
The available scan conversion modes are as follows:

1. One-shot mode for a single channel

This mode is selected when only one analog priority conversion is specified for scan conversion and RPT = 0 in the SCCR register. When the selected priority conversion is completed, the operation stops.

Figure 3-1 Stop of operation in one-shot mode for a single channel

(SCIS3 = 0x00, SCIS2 = 0x00, SCIS1 = 0x00, SCIS0 = 0x08)

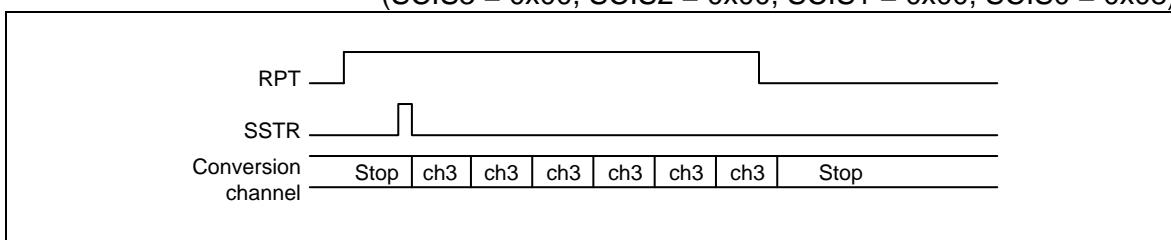


2. Continuous mode for a single channel

This mode is selected when only one analog priority conversion process is specified for scan conversion and RPT = 1 in the SCCR register. When the selected priority conversion is completed, the same priority conversion is started again. To stop A/D conversion, set RPT to "0". The operation stops when the ongoing A/D conversion is completed.

Figure 3-2 Stop of operation in continuous mode for a single channel

(SCIS3 = 0x00, SCIS2 = 0x00, SCIS1 = 0x00, SCIS0 = 0x08)

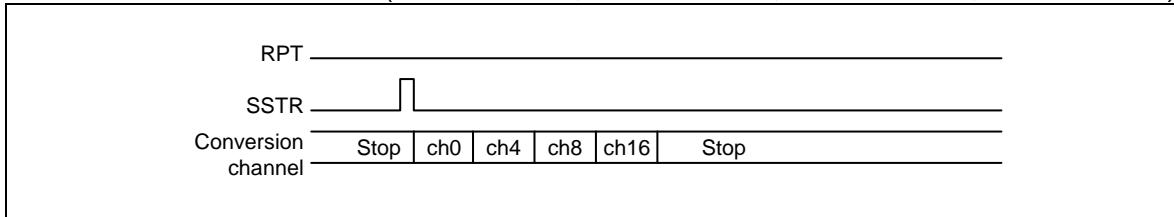


3. One-shot mode for multiple channels

This mode is selected when multiple analog channels are specified for scan conversion and RPT = 0 in the SCCR register. When the conversion starts, the existence of each channel is automatically checked. While the channels are switched from one to another, A/D conversion is started and the conversion result is written to FIFO when the conversion is completed. The conversion channels are selected in descending order of channel number (starting from ch.0). Channels not selected in the SCIS register are skipped and the conversion operation targets the next selected channel. When the A/D conversion of the last one of the selected channels is completed, the A/D conversion is stopped.

Figure 3-3 Stop of operation in one-shot mode for multiple channels

(SCIS3 = 0x00, SCIS2 = 0x01, SCIS1 = 0x01, SCIS0 = 0x11)

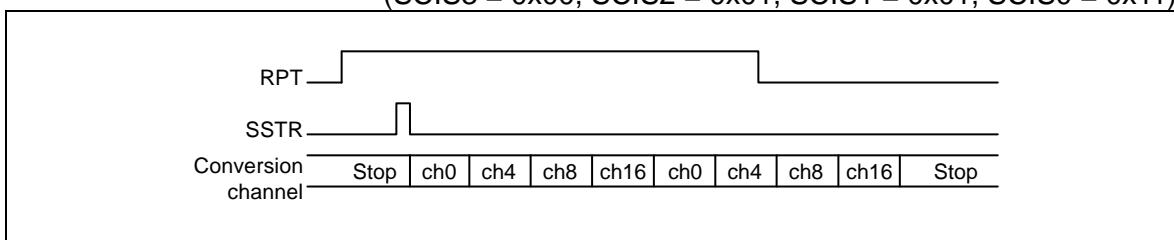


4. Continuous mode for multiple channels

This mode is selected when multiple analog channels are specified for scan conversion and RPT = 1 in the SCCR register. When the conversion starts, the existence of each channel is automatically checked. While the channels are switched from one to another, A/D conversion is started and the conversion result is written to FIFO when the conversion is completed. The conversion channels are selected in descending order of channel number (starting from ch.0). Channels not selected in the SCIS register are skipped and the conversion operation targets the next selected channel. When the A/D conversion of the last one of the selected channels is completed, the conversion operation starts again from ch.0. To end A/D conversion, set RPT "0". The operation stops when the A/D conversion of the last one of the selected channels is completed.

Figure 3-4 Stop of operation in continuous mode for multiple channels

(SCIS3 = 0x00, SCIS2 = 0x01, SCIS1 = 0x01, SCIS0 = 0x11)



3.1.2. Priority conversion operation

This section explains the priority conversion operation.

This mode is used to give priority to a specific conversion process. Even when scan conversion is in progress, if priority conversion is started, the scan conversion is interrupted immediately and the priority conversion is performed. When the priority conversion is completed, the scan operation restarts from the channel where it was interrupted. If conversion with higher priority (priority level 1) is started while the conversion with lower priority (priority level 2) is performed, the priority level 2 conversion is interrupted immediately and the priority level 1 conversion is performed. When the priority level 1 conversion is completed, the priority level 2 conversion is restarted.

Two levels of priority are given to priority conversion. Priority level 1 is the highest and level 2 is the second. Trigger start by an external pin is assigned as the start factor at priority level 1 and software/timer start is assigned as that at priority level 2.

The input channels are selected in the Priority Conversion Input Selection (PCIS) register.

- The procedure for selecting channels at priority level 1 differs depending on the ESCE bit in the Priority Conversion Control (PCCR) register.
 - When ESCE = 0: The P1A [2:0] bits in the PCIS register are used. Only one of the eight channels, ch.0 to 7, can be selected.
 - When ESCE = 1: The setting of the P1A [2:0] bits in the PCIS register is ignored. Only one of the eight channels, ch.0 to 7, can be selected with input from the external pin (ECS [2:0]).

Example: ECS [2:0] = 0b000 -> ch.0
= 0b010 -> ch.2
= 0b111 -> ch.7

- The P2A [4:0] bits in the PCIS register are used for selecting the channel at priority level 2. Only one of the multiple input channels can be selected.

The start factor of A/D conversion differs depending on the priority level.

- Priority level 1 (highest priority) conversion can be started by a falling edge of external trigger input. To enable external trigger start, set the PEEN bit to "1" in the PCCR register.
- Priority level 2 conversion can be started by software or a timer. To start conversion by software, set the PSTR bit in the PCCR register to "1". To start conversion by a timer, set the PHEN bit in the PCCR register to "1" to enable timer start. Conversion starts when the timer's rising edge is detected. When conversion starts, the PCS bit in the ADSR register is set to "1". When the conversion is completed, the PCS bit is reset to "0".

In priority conversion mode, the conversion cannot be restarted. In addition, start factors at the same priority level are ignored.

(A timer start factor is ignored during software-started operation.)

If a priority level 1 start factor (external trigger) occurs during conversion started by a priority level 2 start factor (software or timer), the PCNS bit in the A/D Status Register (ADSR) is set to "1" and the priority level 2 conversion is interrupted immediately. When the priority level 1 conversion is completed, PCNS is reset to "0" and the interrupted priority level 2 conversion is restarted. If a priority level 2 start factor occurs during priority level 1 conversion, the priority level 2 start factor is reserved (retained) and PCNS is set to "1". When the priority level 1 conversion is completed, PCNS is reset to "0" and the priority level 2 conversion is started.

Priority conversion can only be performed in one-shot mode for a single channel.

3.1.3. Priority levels and state transitions

This section explains priority levels and state transitions.

■ Priority levels

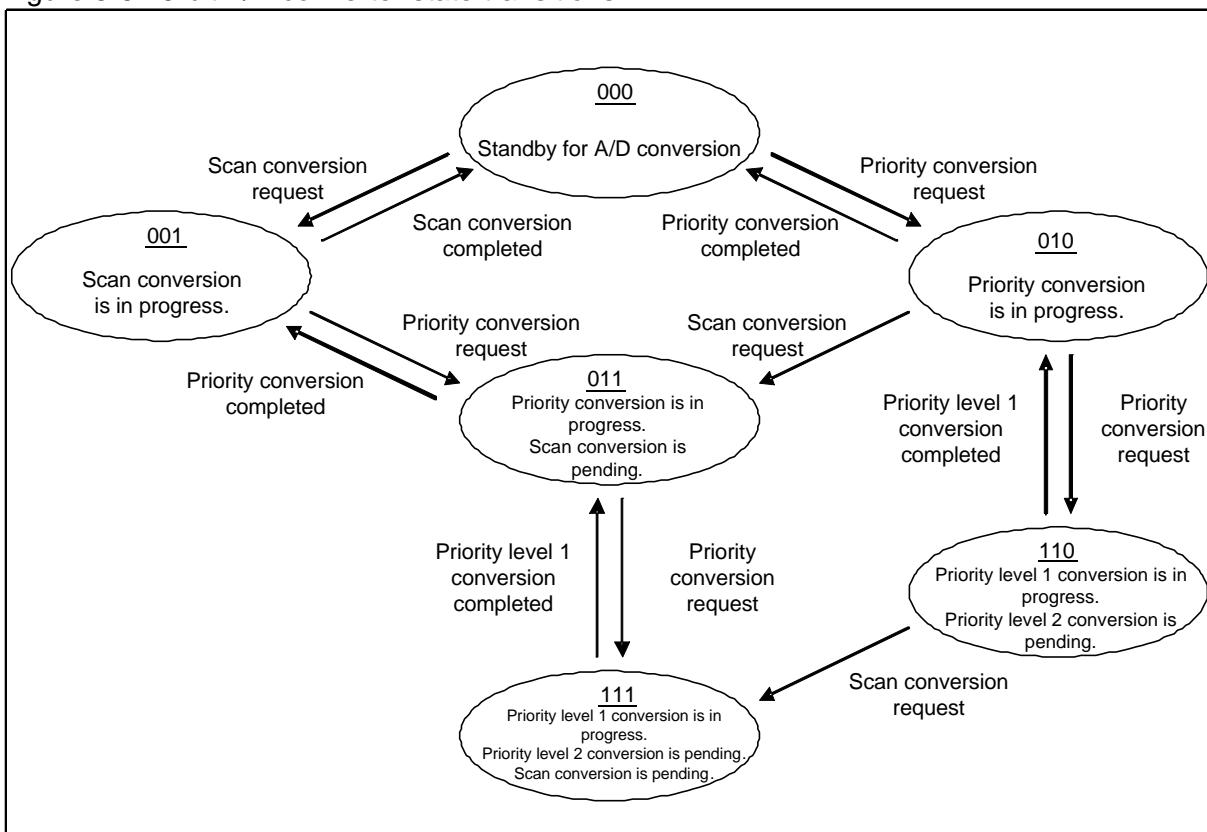
Table 3-1 Priority levels for the A/D converter

Priority level	Conversion type	Start factor
1	Priority level 1 conversion	<ul style="list-style-type: none">Input from external trigger pin (at falling edge)
2	Priority level 2 conversion	<ul style="list-style-type: none">Software (when the PSTR bit is set to "1")Trigger input from timer (at rising edge)
3	Scan conversion	<ul style="list-style-type: none">Software (when the SSTR bit is set to "1")Trigger input from timer (at rising edge)

- When a startup by priority conversion occurs during scan conversion
 - The scan conversion operation is interrupted and priority conversion operation is performed. When the priority conversion operation is completed, the scan conversion is restarted from the channel where it was interrupted.
- When a startup at priority level 1 occurs during conversion at priority level 2
 - The priority level 2 conversion is interrupted and the operation by the startup at priority level 1 is performed. When the priority level 1 operation is completed, the priority level 2 conversion is restarted automatically.
- When a startup at priority level 2 occurs during conversion at priority level 1
 - The start factor at priority level 2 is retained. When the priority level 1 conversion is completed, the priority level 2 conversion is started automatically.
- When a startup of scan conversion occurs during priority level 1 conversion
 - The start factor of the scan conversion is retained. When the priority level 1 conversion is completed, the scan conversion operation is started automatically.
- When a startup of scan conversion occurs during priority level 2 conversion
 - The start factor of the scan conversion is retained. When the priority level 2 conversion is completed, the scan conversion operation is started automatically.
- While priority conversion is performed, start factor at the same priority level are masked (the operation is not restarted).

■ State transitions

Figure 3-5 10-bit A/D converter state transitions



The operation states can be read from the SCS, PCS, and PCNS bits of the ADSR register.

Table 3-2 Correspondence between bits and operation states

PCNS	PCS	SCS	Explanation of states
0	0	0	Standby for A/D conversion.
0	0	1	Scan A/D conversion is in progress.
0	1	0	Priority A/D conversion (priority level 1 or 2) is in progress.
0	1	1	Priority A/D conversion (priority level 1 or 2) is in progress. Scan conversion is pending.
1	1	0	Priority A/D conversion (priority level 1) is in progress. Priority conversion (priority level 2) is pending.
1	1	1	Priority A/D conversion (priority level 1) is in progress. Scan conversion and priority conversion (priority level 2) are pending.

3.2. FIFO operations

The A/D converter has 16 FIFO stages for scan conversion and 4 FIFO stages for priority conversion. When conversion data is written in the specified count of FIFO stages, an interrupt is generated to the CPU.

- 3.2.1 FIFO operations in scan conversion
- 3.2.2 Interrupts in scan conversion
- 3.2.3 FIFO operations in priority conversion
- 3.2.4 Interrupts in priority conversion
- 3.2.5 Restrictions on reading FIFO data registers in empty state
- 3.2.6 Bit placement selection for FIFO data registers

3.2.1. FIFO operations in scan conversion

This section explains FIFO operations in scan conversion.

Sixteen FIFO stages are incorporated for writing scan conversion data. After reset, they are in empty state and the SEMP bit in the Scan Conversion Control Register is set to "1". When A/D conversion of one channel is completed, the conversion result and conversion channel are written in the first FIFO stage. This resets SEMP to "0". The conversion result and conversion channel for the subsequent channels are written in the corresponding FIFO stages.

When such data is written in all of the 16 stages, the SFUL bit is set to "1" to indicate that FIFO is in full state. If conversion is performed and an attempt is made to write data in FIFO when FIFO is in full state, the SOVR bit is set to "1" and the data is discarded (cannot overwrite the existing data).

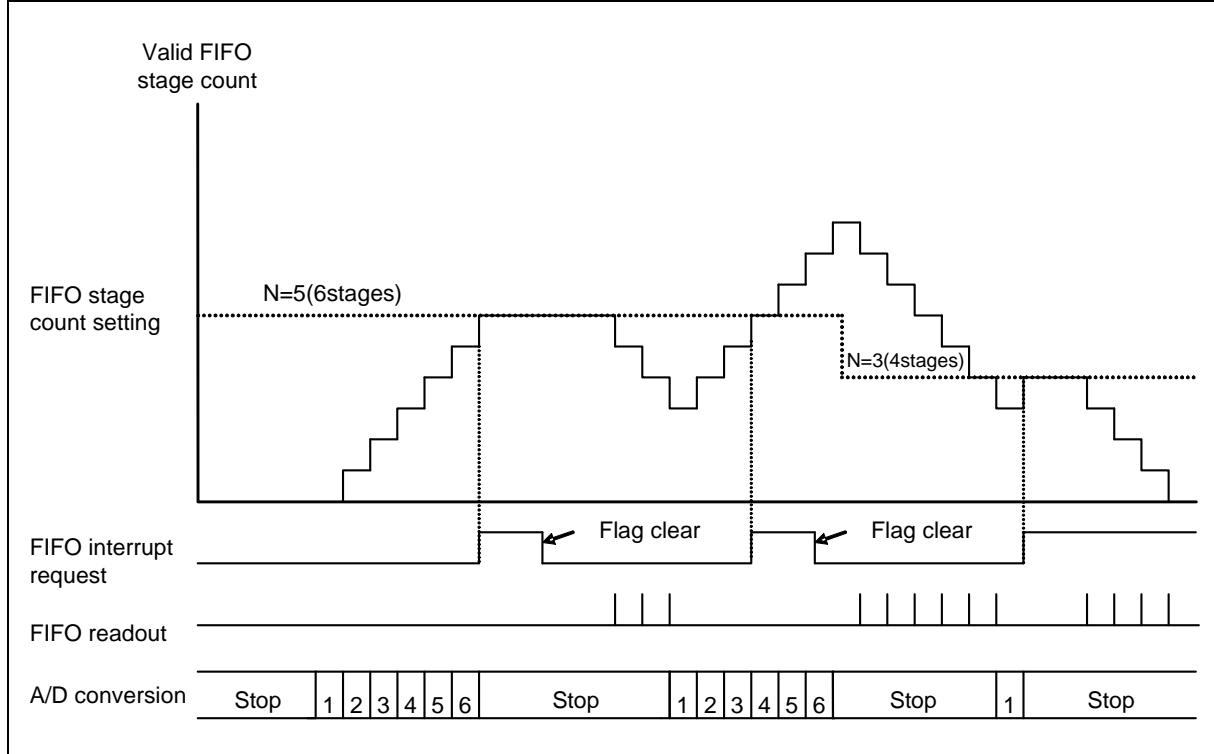
To clear the data in FIFO, set the SFCLR bit in the Scan Conversion Control register to "1". FIFO goes to the empty state and the SEMP bit is set to "1".

Data in FIFO can be read sequentially by reading the Scan FIFO Data Register (SCFD). To perform a byte access to this register, read the most significant byte (bit 15:8) to shift FIFO (reading the least significant byte (bit 7:0) does not shift FIFO).

3.2.2. Interrupts in scan conversion

This section explains interrupts in scan conversion.

Figure 3-6 FIFO interrupt settings and FIFO operations



When conversion data for the number of FIFO stages ($N + 1$) set in SFS [3:0] in the Scan Conversion FIFO Stage Count Setup Register (SFNS) is written in FIFO, the interrupt request bit (SCIF) in the A/D Control Register (ADCR) is set to "1". If the interrupt enable bit (SCIE) is set to "1", an interrupt request is generated to the CPU.

The following explains FIFO stage count interrupt methods for each scan conversion mode.

1. One-shot mode for a single channel

To generate an interrupt after the completion of one conversion process for the specified channel, set SFS [3:0] = 0x0. When conversion data is written in the first FIFO stage, SCIF is set to "1".

<Note>

If SFS [3:0] is set to 0x1 or more (two stages or more), interrupts are not generated until the specified number of times of conversion is completed.

2. Continuous mode for a single channel

To generate an interrupt after the completion of one conversion process for the specified channel, set SFS [3:0] = 0x0. When conversion data is written in the first FIFO stage, SCIF is set to "1".

To generate an interrupt at the completion of a number of times of conversion of the specified channel, set SFS [3:0] to 0x1 or more (two stages or more). For example, set SFS [3:0] = 0x3 to generate an interrupt after four repeats.

3. One-shot mode for multiple channels

To generate an interrupt after the completion of conversion of the multiple specified channels, set the FIFO stage count according to the number of channels. If eight channels are selected, set the FIFO stage count by setting SFS [3:0] = 0x7. When the conversion of the last one of the selected channels is completed, SCIF is set to "1".

An interrupt can be generated at any timing before scan completion by setting SFS [3:0] to a value less than the number of selected channels.

4. Continuous mode for multiple channels

To generate an interrupt after the completion of the first scan of the multiple specified channels, set the FIFO stage count according to the number of channels. If eight channels are selected, set the FIFO stage count by setting SFS [3:0] = 0x7. When the conversion of the last one of the selected channels is completed, SCIF is set to "1".

To generate an interrupt after the completion of the second scan, set the FIFO stage count to twice the number of selected channels. For example, when four channels are selected, set the FIFO stage count to 8 (SFS [3:0] = 0x7). An interrupt is generated when the second scan is completed.

Because the FIFO stage count can be set to any value, an interrupt can be generated at any desired timing.

3.2.3. FIFO operations in priority conversion

This section explains FIFO operations in priority conversion.

Four FIFO stages are incorporated for writing priority conversion data. After reset, they are in empty state and the PEMP bit in the Priority Conversion Control Register is set to "1". When one A/D conversion process is completed, the conversion result, start factor (priority level), and conversion channels are written in the first FIFO stage. This resets SEMP to "0". The conversion result and conversion channels for the subsequent conversion processes are written in the corresponding FIFO stages.

When such data is written in all of the 4 stages, the PFUL bit is set to "1" to indicate that FIFO is in full state. If conversion is performed and an attempt is made to write data in FIFO when FIFO is in full state, the POVR bit is set to "1" and the data is discarded (cannot overwrite the existing data).

To clear the data in FIFO, set the PFCLR bit in the Priority Conversion Control Register to "1". FIFO goes to the empty state and the PEMP bit is set to "1".

Data in FIFO can be read sequentially by reading the Priority FIFO Data Register (PCFD). To perform byte access to this register, read the most significant byte (bit 15:8) to shift FIFO (reading the least significant byte (bit 7:0) does not shift FIFO).

3.2.4. Interrupts in priority conversion

This section explains interrupts in priority conversion.

When conversion data for the number of FIFO stages ($N + 1$) set in PFS [1:0] in the Priority Conversion FIFO Stage Count Setup Register (PFNS) is written in FIFO, the interrupt request bit (PCIF) in the A/D Control Register (ADCR) is set to "1". If the interrupt enable bit (PCIE) is set to "1", an interrupt request is generated to the CPU.

The following explains FIFO stage count interrupt methods in priority conversion.

To generate an interrupt after the completion of one conversion process for the specified channel, set PFS [1:0] = 0x0. When conversion data is written in the first FIFO stage, PCIF is set to "1".

<Note>

If PFS [1:0] is set to 0x1 or more (two stages or more), interrupts are not generated until the specified number of times of conversion is completed.

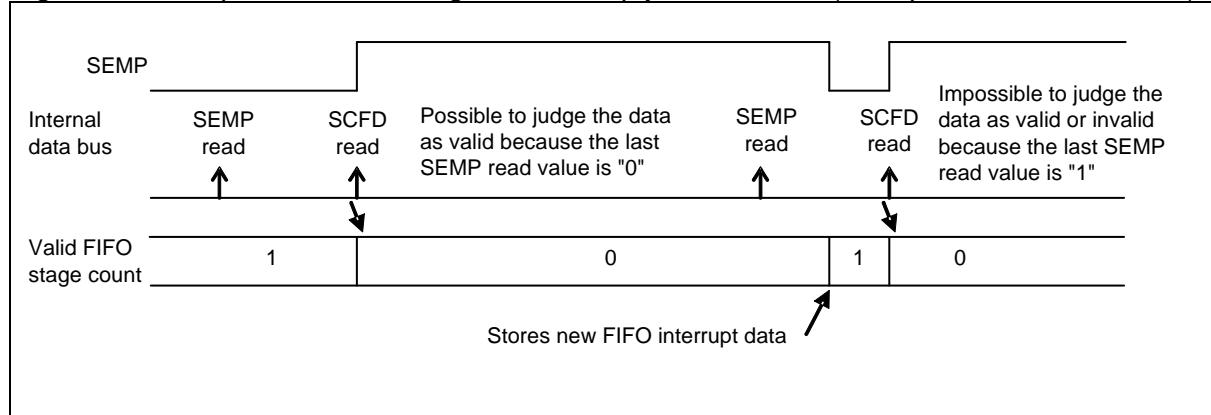
3.2.5. Restrictions on reading FIFO data registers in empty state

This section explains restrictions on reading FIFO data registers.

Read SCFD (Scan FIFO Data Register) only when SEMP (scan conversion FIFO empty bit) = 0. Likewise, read the PCFD (Priority FIFO Data Register) only when PEMP (priority conversion FIFO empty bit) = 0.

If the SCFD or PCFD is read after confirming that the SEMP or PEMP is "1" (FIFO is empty), an A/D conversion result may have been stored immediately before the reading. In other words, whether the value read from the SCFD or PCFD is valid or invalid cannot be judged and valid FIFO data may be discarded.

Figure 3-7 Example of FIFO reading after the empty bit is set to 1 (example in scan conversion)



3.2.6. Bit placement selection for FIFO data registers

This section explains bit placement selection for FIFO data registers.

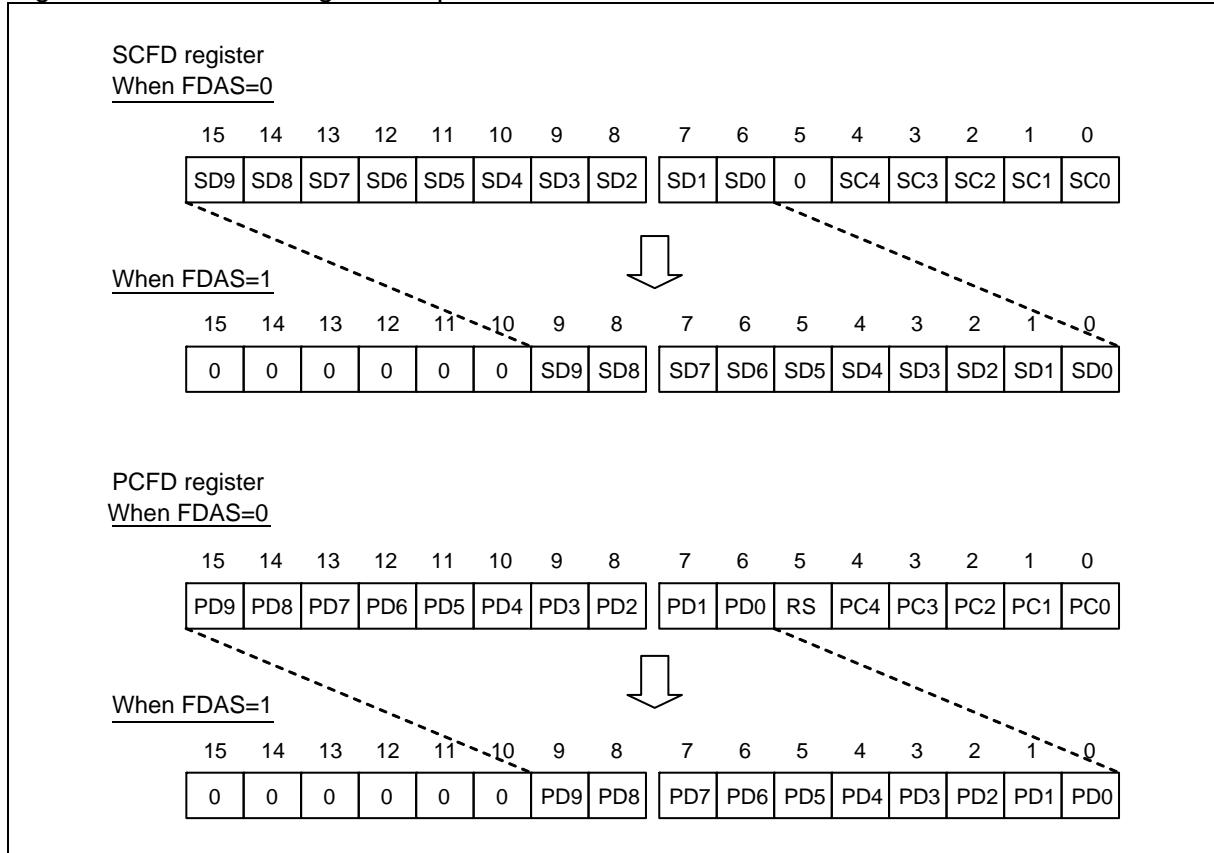
The A/D converter can change the bit placement for the conversion results in the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) with the FDAS bit in the A/D Status Register (ADSR).

The bit placement in the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) can be changed by setting the FDAS bit in the A/D Status Register (ADSR)(Figure 3-8).

When the FDAS bit is set to "1", the 10-bit A/D conversion result (SD9 to SD0 and PD9 to PD0) are assigned to the LSB side (bit 9:0). This placement should be used when channel information is not required (for example, conversion of only one channel is performed) because the converted channel information (SC4 to SC0 and PC4 to PC0) is lost.

In addition, because there is no information on the priority A/D start factor (RS) when the FDAS bit is "1", conversion at priority levels 1 and 2 must not be started together.

Figure 3-8 FIFO data register bit placement



3.3. A/D comparison function

The A/D comparison function compares A/D conversion results and generates interrupts.

To use the comparison function, set the CMPEN bit in the A/D Comparison Control Register (bit 7 in the CMPCR register) to "1".

The values set in the A/D Comparison Value Setup Register (CMPD) are compared with the most significant eight bits (AD9 to 2) of the A/D conversion result. If the comparison result satisfies the conditions set in the A/D Comparison Control Register (CMPCR), the A/D comparison interrupt bit (CMPIF) in the ADCR register is set to "1". If the interrupt enable bit (CMPIE) is "1", an interrupt is generated to the CPU.

<Note>

Two bits (AD1 to 0) on the LSB side are not compared.

Because the result of A/D conversion, regardless of scan or priority, is compared before it is written to FIFO, comparison is possible when FIFO is full.

If CMD1 is set to "1" (to generate an interrupt when the result is equal to or more than the CMPD set value), CMPIF is set to "1" when the conversion result is equal to the value in the A/D Comparison Value Setup Register.

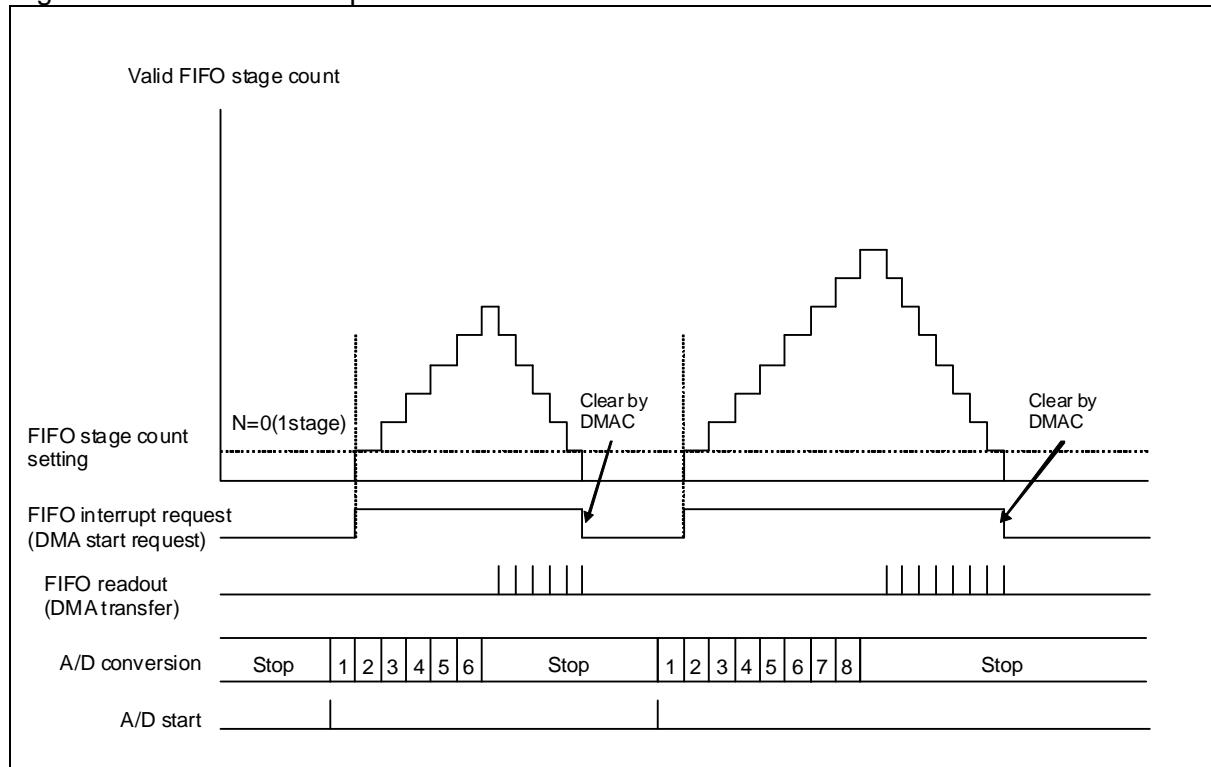
3.4. Starting DMA

The A/D converter can start DMA transfer with a scan conversion FIFO stage count interrupt request.

The A/D converter can transfer scan FIFO data by connecting the interrupt signal of scan conversion from the A/D converter to DMA and starting DMA. By setting the scan FIFO stage count to "0" (an interrupt request is generated when a conversion result is stored in the first stage of FIFO), DMA transfer can be performed in conjunction with A/D conversion.

The setting in the DMA Transfer Request Selection Register as to whether the A/D converter interrupt signal is connected to the interrupt controller or DMAC should be made for DMA transfer of the interrupt controller.

Figure 3-9 DMA transfer operation



4. Setup procedure examples

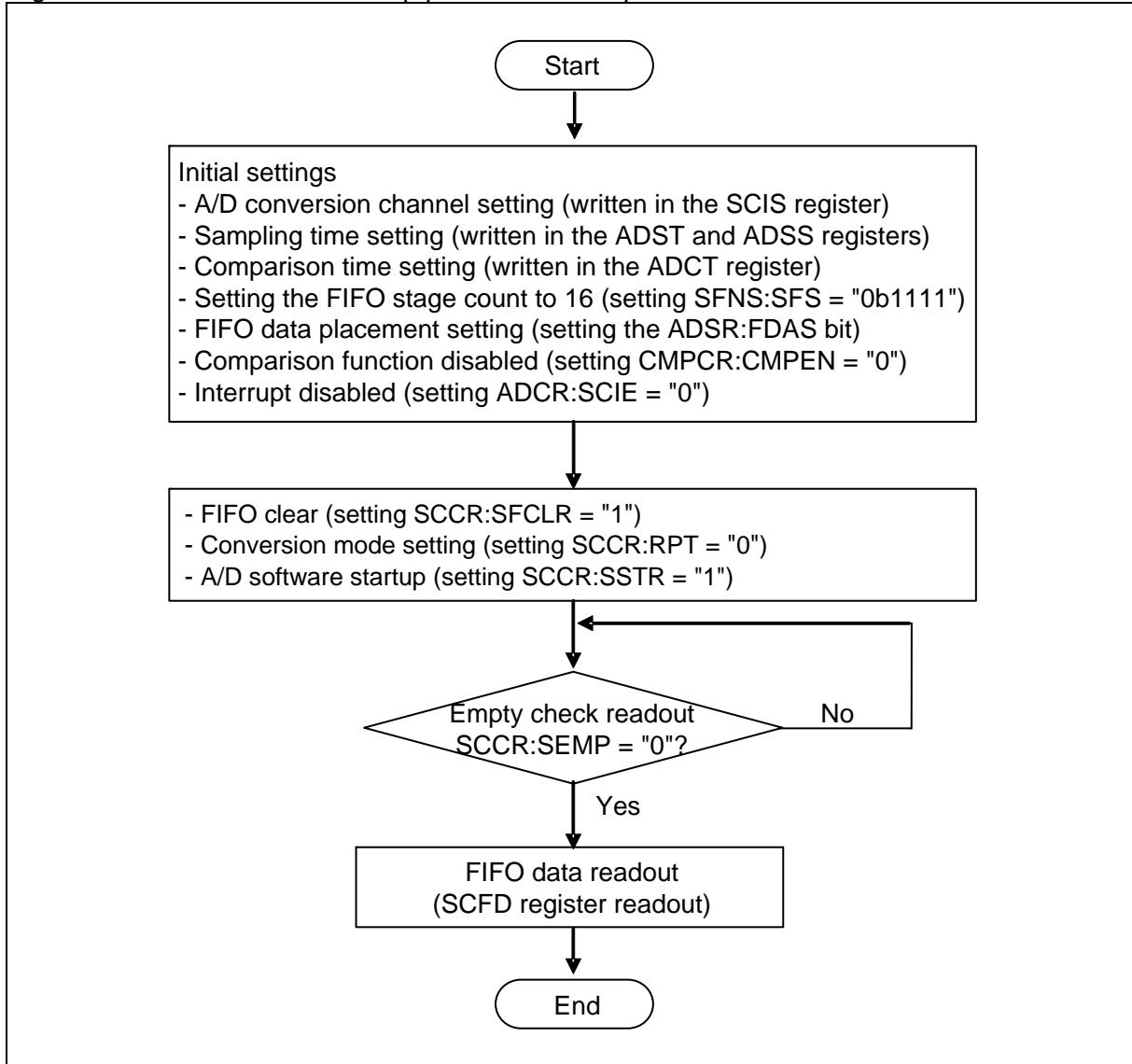
This section provides examples of setup procedures for the 10-bit A/D converter.

- 4.1 Scan conversion setup procedure example
- 4.2 Priority conversion setup procedure example
- 4.3 Setting the conversion time

4.1. Scan conversion setup procedure example

This section provides a scan conversion setup procedure example.

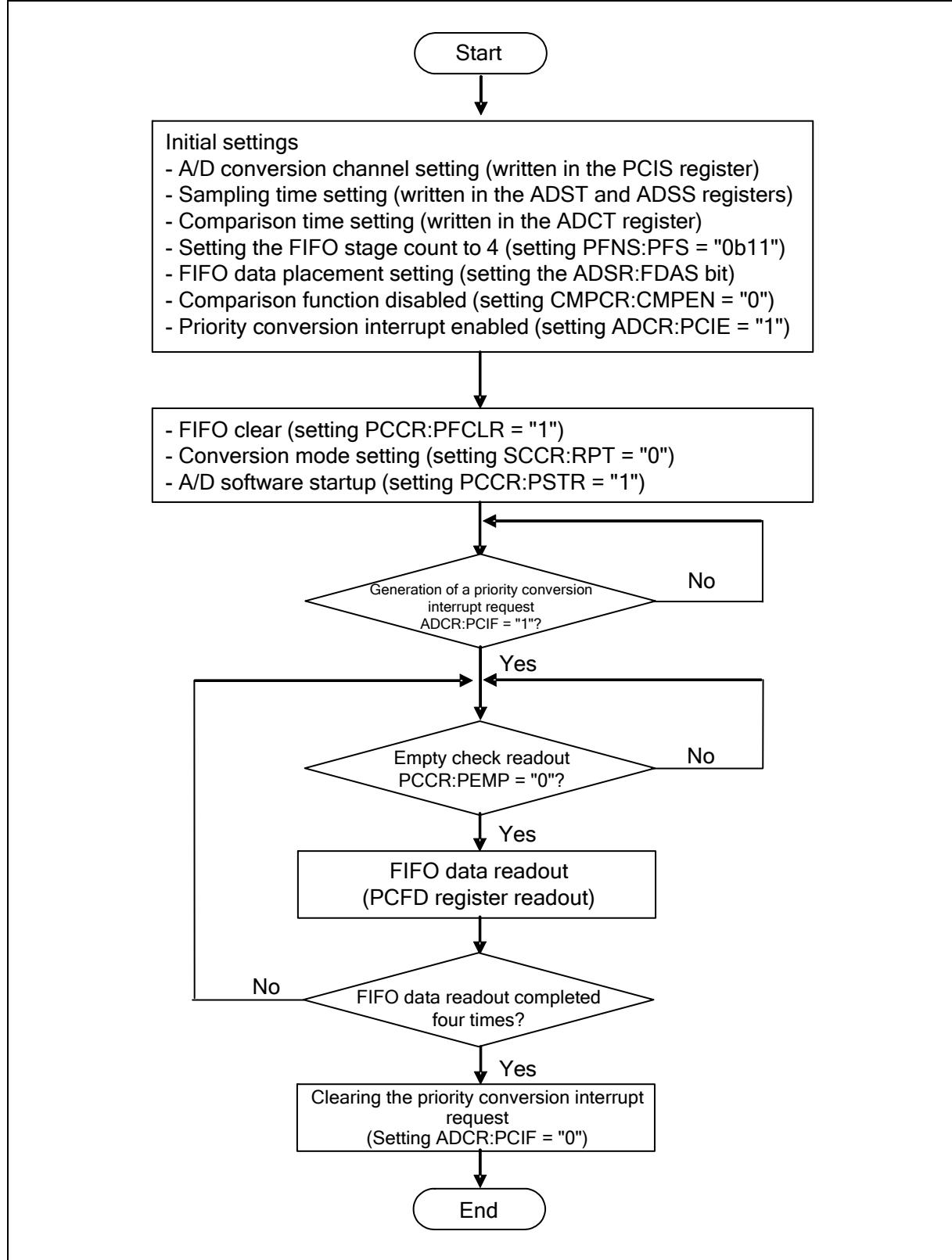
Figure 4-1 Scan conversion setup procedure example



4.2. Priority conversion setup procedure example

This section provides a priority conversion setup procedure example.

Figure 4-2 Priority conversion setup procedure example



4.3. Setting the conversion time

The conversion time of the A/D converter is "sampling time" + "comparison time". Two sampling time settings can be applied to each channel. This section explains how to set and calculate the conversion time.

■ Example of setting the sampling time

A sampling time is set in each of Sampling Time Setup Registers 0 and 1 (ADST0 and ADST1). Using Sampling Time Selection Registers (ADSS3 to 0), whether Sampling Time Setup Registers 0 or 1 is used to provide the value can be selected for each channel. This allows you to set different sampling times for channels with different external impedances.

$$\text{Sampling time} = \text{Peripheral clock (PCLK) cycle} \times (\text{ST set value} + 1) \times \text{STX setting multiplier}$$

<Note>

When STXx1 and STXx0 = 00 (STx5 to STx0 set values multiplied by 1) are set, set STx5 to STx0 to "3" or more ("2" or less must not be set). Furthermore, the sampling time conditions in "Electric Characteristics" of the Data Sheet the must be satisfied.

■ Example of setting the comparison time

The comparison time is set in the Comparison Time Setup Register (ADCT).

$$\text{Comparison time} = \{(CT \text{ set value (N)} + 1) \times 10 + 4\} \times \text{Peripheral clock (PCLK) cycle}$$

<Note>

If the sampling or comparison time fails to meet the electrical characteristics of the A/D converter, the A/D conversion accuracy may be degraded.

■ Example of conversion time calculation (when PCLK = 20 MHz (50 ns cycle))

(1) Sampling time

- When ST05 to 00 = 7 and STX01 and STX00 = 00 (multiplied by 1)
Sampling time = $50 \text{ ns} \times (7 + 1) \times 1 = 400 \text{ ns}$
- When ST15 to 10 = 19 and STX11 and STX10 = 01 (multiplied by 4)
Sampling time = $50 \text{ ns} \times (19 + 1) \times 4 = 4 \mu\text{s}$

(2) Comparison time

- When CT02 to 00 = 1
Sampling time = $\{(1 + 1) \times 10 + 4\} \times 50 \text{ ns} = 1.2 \mu\text{s}$

(3) Conversion time

By adding (1) and (2) together:

- Conversion time for channels specified with the ADST0 register = $1.6 \mu\text{s}$
- Conversion time for channels specified with the ADST1 register = $5.2 \mu\text{s}$

5. Registers

This section explains the configuration and functions of the registers used for the 10-bit A/D converter.

■ List of registers for the 10-bit A/D converter

Abbreviation	Register name	See
ADCR	A/D Control Register	5.1
ADSR	A/D Status Register	5.2
SCCR	Scan Conversion Control Register	5.3
SFNS	Scan Conversion FIFO Stage Count Setup Register	5.4
SCFD	Scan Conversion FIFO Data Register	5.5
SCIS	Scan Conversion Input Selection Register	5.6
PCCR	Priority Conversion Control Register	5.7
PFNS	Priority Conversion FIFO Stage Count Setup Register	5.8
PCFD	Priority Conversion FIFO Data Register	5.9
PCIS	Priority Conversion Input Selection Register	5.10
CMPD	A/D Comparison Value Setup Register	5.11
CMPCR	A/D Comparison Control Register	5.12
ADSS	Sampling Time Selection Register	5.13
ADST	Sampling Time Setup Register	5.14
ADCT	Comparison Time Setup Register	5.15

5.1. A/D Control Register (ADCR)

The A/D Control Register (ADCR) performs interrupt flag display and interrupt enable control.

bit	15	14	13	12	11	10	9	8
Field	SCIF	PCIF	CMPIF	Reserved	SCIE	PCIE	CMPIE	OVRIE
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	X	0	0	0	0

[bit 15] SCIF: Scan conversion interrupt request bit

When conversion values are written up to the stage count specified in the Scan Conversion FIFO Stage Count Setup Register (SFNS), this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Conversion result is not stored.	Clears this bit.
1	Conversion result is stored.	No effect.

[bit 14] PCIF: Priority conversion interrupt request bit

When conversion values are written up to the stage specified in the Priority Conversion FIFO Stage Count Setup Register (PFNS), this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Conversion result is not stored.	Clears this bit.
1	Conversion result is stored.	No effect.

[bit 13] CMPIF: Conversion result comparison interrupt request bit

When the condition set in the A/D Comparison Value Setup Register (CMPD) or A/D Comparison Control Register (CMPCR) is satisfied during the operation of the A/D conversion result comparison function, this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Specified condition is not satisfied.	Clears this bit.
1	Specified condition is satisfied.	No effect.

[bit 12] Reserved: Reserved bit

Write	Has no effect on operation.
Read	The value is undefined.

[bit 11] SCIE: Scan conversion interrupt enable bit

This bit controls the interrupt request of SCIF. When the SCIE bit is enabled, and the SCIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 10] PCIE: Priority conversion interrupt enable bit

This bit controls the interrupt request of PCIF. When the PCIE bit is enabled, and the PCIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 9] CMPIE: Conversion result comparison interrupt enable bit

This bit controls the interrupt request of CMPIF. When the CMPIE bit is enabled, and the CMPIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 8] OVRIE: FIFO overrun interrupt enable bit

This bit controls the interrupt request of the SOVR bit in the SCCR register or the POVR bit in the PCCR register. When the OVRIE bit is enabled, and the SOVR or POVR bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

5.2. A/D Status Register (ADSR)

The A/D Status Register (ADSR) displays scan and priority conversion statuses.

bit	7	6	5	4	3	2	1	0
Field	ADSTP	FDAS	Reserved			PCNS	PCS	SCS
Attribute	R/W	R/W	-	-	-	R	R	R
Initial value	0	0	X	X	X	0	0	0

[bit 7] ADSTP: A/D conversion forced stop bit

Setting the ADSTP bit to "1" stops the A/D conversion operation forcibly (both scan and priority conversion operations are stopped). Forced stop of A/D conversion initializes the PCNS, PCS, and SCS bits in the ADSR register to "0". However, other register bits are not reset.

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
1		Stops the conversion operation forcibly.

[bit 6] FDAS: FIFO data placement selection bit

Setting the FDAS bit to "1" shifts the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) values by 6 bits to the LSB side, placing the conversion result in bit 9 to 0.

When the FDAS bit is set to "1", the FIFO Data Registers (SCFD and PCFD) have no information about the converted channels. For this reason, use this bit when channel information is not required (for example, conversion of only one channel is performed).

When the FDAS bit is set to "1", the information of the RS bit in the Priority Conversion FIFO Data Register (PCFD) is also lost. Use this bit only for priority level 1 or 2.

Bit	Description
0	Places conversion result on the MSB side (with channel information).
1	Places conversion result on the LSB side (without channel information).

[bit 5:3] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 2] PCNS: Priority conversion pending flag

This flag indicates that conversion at priority level 2 (software/timer) is pending. This flag is set when priority conversion at priority level 2 (software/timer) is started while priority conversion at priority level 1 (external trigger start) is performed or when conversion at priority level 1 is started while priority conversion at priority level 2 is performed. Writing is ignored.

Bit	Description
0	Priority level 2 conversion is not pending.
1	Priority level 2 conversion is pending.

[bit 1] PCS: Priority conversion status flag

This flag indicates that priority A/D conversion is in progress. This flag is set while priority conversion at priority level 1 or 2 is performed. Writing is ignored.

Bit	Description
0	Priority conversion is stopped.
1	Priority conversion is in progress.

[bit 0] SCS: Scan conversion status flag

This flag indicates that scan A/D conversion is in progress. Writing is ignored.

Bit	Description
0	Scan conversion is stopped.
1	Scan conversion is in progress.

5.3. Scan Conversion Control Register (SCCR)

The Scan Conversion Control Register (SCCR) controls the scan conversion mode.

bit	15	14	13	12	11	10	9	8
Field	SEMP	SFUL	SOVR	SFCLR	Reserved	RPT	SHEN	SSTR
Attribute	R	R	R/W	R/W	-	R/W	R/W	R/W
Initial value	1	0	0	0	X	0	0	0

[bit 15] SEMP: Scan conversion FIFO empty bit

This bit is set when FIFO goes to the empty state. When conversion data is written in the Scan Conversion FIFO Data Register (SCFD), this bit is set to "0". Writing is ignored.

Bit	Description
0	Data remains in FIFO.
1	FIFO is empty.

[bit 14] SFUL: Scan conversion FIFO full bit

This bit is set when FIFO goes to full state. When SFCLR is set to "1" or the Scan Conversion FIFO Data Register (SCFD) is read, this bit is set to "0". Writing is ignored.

Bit	Description
0	Data can be input to FIFO.
1	FIFO is full.

[bit 13] SOVR: Scan conversion overrun flag

This bit is set when an attempt to write data to a full FIFO is made (conversion data in a full FIFO is not overwritten). The read value of Read-Modify-Write access is "1" regardless of the bit value. When the OVRIE bit in the ADCR register is "1", an interrupt is generated to the CPU if the SOVR bit is "1".

Bit	Description	
	Read	Write
0	No overrun has occurred.	Clears this bit.
1	Overrun has occurred.	No effect.

[bit 12] SFCLR: Scan conversion FIFO clear bit

Setting this bit to "1" clears the scan conversion FIFO. FIFO becomes empty and the SEMP bit is set to "1".

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
1		Clears FIFO.

[bit 11] Reserved: Reserved bit

Write	Has no effect on operation.
Read	The value is undefined.

[bit 10] RPT: Scan conversion repeat bit

Setting this bit to "1" places the converter in the repeat mode. When the conversion of all analog input channels selected in the Scan Conversion Input Selection Register (SCIS) is completed, the conversion is started again.

Setting the RPT bit to "0" ends the repeat conversion. The operation stops when the conversion of the analog input channels selected in the SCIS bit is completed.

Setting the RPT bit to "1" must be performed while scan conversion is stopped (the SCS bit in the ADSR register = "0"). (Setting the SSTR bit to "1" may be performed simultaneously with setting the RPT bit to "1".)

Bit	Description
0	Single conversion mode
1	Repeat conversion mode

[bit 9] SHEN: Scan conversion timer start enable bit

Set this bit to "1" to start scan conversion using a rising edge from a timer. Software startup (SSTR = 1) is valid even when this bit is set to "1".

Bit	Description
0	Timer start disable
1	Timer start enable

[bit 8] SSTR: Scan conversion start bit

Setting this bit to "1" starts A/D conversion. Setting this bit to "1" again during conversion stops the ongoing conversion immediately and restarts the conversion.

Bit	Description	
	Read	Write
0		No effect.
1	The value is always "0".	Starts conversion or restarts the conversion (during conversion).

<Note>

If a startup by a timer occurs simultaneously with the setting of the SSTR bit to "1", the setting of the SSTR bit to "1" takes preference and the startup by the timer is ignored.

5.4. Scan Conversion FIFO Stage Count Setup Register (SFNS)

The Scan Conversion FIFO Stage Count Setup Register (SFNS) sets up the generation of interrupt requests in scan conversion. When the specified count of FIFO stages store A/D conversion data, the interrupt request bit (SCIF) is set.

bit	7	6	5	4	3	2	1	0
Field	Reserved				SFS [3:0]			
Attribute	-	-	-	-	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

[bit 7:4] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 3:0] SFS [3:0]: Scan conversion FIFO stage count setting bit

When A/D conversion data for the FIFO stage count ($N + 1$) set in SFS [3:0] is written, the interrupt request flag (SCIF) is set to "1".

Bit [3:0]	Description
0b0000	Generates an interrupt request when conversion result is stored in the first FIFO stage.
0b0001	Generates an interrupt request when conversion result is stored in the second FIFO stage.
0b0010	Generates an interrupt request when conversion result is stored in the third FIFO stage.
...	...
0b1101	Generates an interrupt request when conversion result is stored in the 14th FIFO stage.
0b1110	Generates an interrupt request when conversion result is stored in the 15th FIFO stage.
0b1111	Generates an interrupt request when conversion result is stored in the 16th FIFO stage.

5.5. Scan Conversion FIFO Data Register (SCFD)

The Scan Conversion FIFO Data Register (SCFD) consists of 16 FIFO stages and stores analog conversion results. Data can be retrieved sequentially by reading the register.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0	Reserved	SC4	SC3	SC2	SC1	SC0
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit 15:6] SD9:SD0: Scan conversion result

The result of 10-bit scan A/D conversion is written.

[bit 5] Reserved: Reserved bit

The read value is "0".

[bit 4:0] SC4:SC0: Conversion input channel bits

The analog input channels corresponding to the conversion result written in SD9 to SD0 are written.

Settings for channels not defined in the product specifications are not written. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

<Note>

This register has different bit configurations depending on the FDAS bit setting in the A/D Status Register (ADSR). When the FDAS bit is "1", see "3.2.6 Bit placement selection for FIFO data registers".

To perform a byte access to this register, read the most significant byte (bit 15:8) to shift the FIFO data. Reading the least significant byte (bit 7:0) does not shift FIFO.

5.6. Scan Conversion Input Selection Register (SCIS)

The Scan Conversion Input Selection Register (SCIS) is used to select analog input channels for which scan conversion is performed. Any channels can be selected from multiple analog inputs. The selected channels are converted in ascending order of channel number.

■ SCIS3 (most significant byte: AN31 to AN24) and SCIS2 (least significant byte: AN23 to AN16)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	AN31	AN30	AN29	AN28	AN27	AN26	AN25	AN24	AN23	AN22	AN21	AN20	AN19	AN18	AN17	AN16
Attribute	R/W															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] AN31:AN16: Analog input selection bits

When these bits are set to "1", the corresponding channels are selected for analog conversion.

■ SCIS1 (most significant byte: AN15 to AN8) and SCIS0 (least significant byte: AN7 to AN0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] AN15:AN0: Analog input selection bits

When these bits are set to "1", the corresponding channels are selected for analog conversion.

<Note>

It is not possible to change the channels during A/D conversion. Be sure to set SCIS3 to SCIS0 while the A/D conversion is stopped.

It is not possible to set "1" in the bit corresponding to a channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

■ Example of scan conversion order

The selected channels are converted in ascending order of channel number.

Example: When the AN1, AN3, AN5, and AN23 bits are set to "1", the analog conversion proceeds from ch.1, ch.3, ch.5, and to ch.23.

5.7. Priority Conversion Control Register (PCCR)

The Priority Conversion Control Register (PCCR) controls the priority conversion mode.

Priority conversion can be performed even while scan conversion is being performed.

In addition, different priority levels (two levels) can be given to priority conversion processes.

bit	15	14	13	12	11	10	9	8
Field	PEMP	PFUL	POVR	PFCLR	ESCE	PEEN	PHEN	PSTR
Attribute	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	0	0	0	0

[bit 15] PEMP: Priority conversion FIFO empty bit

This bit is set when FIFO goes to the empty state. When conversion data is written in the Priority Conversion FIFO Data Register (PCFD), this bit is set to "0". Writing is ignored.

Bit	Description
0	Data remains in FIFO.
1	FIFO is empty.

[bit 14] PFUL: Priority conversion FIFO full bit

This bit is set when FIFO goes to full state. When PFCLR is set to "1" or the Priority Conversion FIFO Data Register (PCFD) is read, this bit is set to "0". Writing is ignored.

Bit	Description
0	Data can be input to FIFO.
1	FIFO is full.

[bit 13] POVR: Priority conversion overrun flag

This bit is set when an attempt to write data to a full FIFO is made (conversion data in a full FIFO is not overwritten). The read value of Read-Modify-Write access is "1" regardless of the bit value. When the OVRIE bit in the ADCR register is "1", an interrupt is generated to the CPU if the POVR bit is "1".

Bit	Description	
	Read	Write
0	No overrun has occurred.	Clears this bit.
1	Overrun has occurred.	No effect.

[bit 12] PFCLR: Priority conversion FIFO clear bit

Setting this bit to "1" clears the priority conversion FIFO. FIFO becomes empty and the PEMP bit is set to "1".

Bit	Description	
	Read	Write
0		No effect.
1	The value is always "0".	Clears FIFO.

[bit 11] ESCE: External trigger analog input selection bit

This bit selects whether the external trigger analog input is selected with the P1A [2:0] bits in the Priority Conversion Input Selection Register (PCIS) or the external input pin ECS [2:0] bits.

Bit	Description
0	The external trigger analog inputs are selected with P1A [2:0].
1	The external trigger analog inputs are selected with an external input.

<Note>

It is not possible to change the setting of the ESCE bit during A/D conversion. To change the setting, make sure the A/D conversion is stopped.

If channel selection with external pins ECS [2:0] cannot be used due to the product specifications, be sure to set the ESCE bit to "0".

[bit 10] PEEN: Priority conversion external start enable bit

Set this bit to "1" to start priority conversion using a falling edge of an external trigger pin input. Conversion started with an external trigger has priority level 1 (highest priority).

Bit	Description
0	External trigger start disable
1	External trigger start enable

[bit 9] PHEN: Priority conversion timer start enable bit

Set this bit to "1" to start priority conversion using a rising edge from a timer. Software startup (PSTR = 1) is valid even when this bit is set to "1". Conversion started with an external trigger has priority level 2 (lower than level 1).

Bit	Description
0	Timer start disable
1	Timer start enable

[bit 8] PSTR: Priority conversion start bit

Setting this bit to "1" starts A/D conversion. Conversion started with this bit has priority level 2 (lower than level 1). It is not possible to restart the conversion started with this bit.

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
		Starts priority conversion.

5.8. Priority Conversion FIFO Stage Count Setup Register (PFNS)

The Priority Conversion FIFO Stage Count Setup Register (PFNS) sets up the generation of interrupt requests in priority conversion. When the specified count of FIFO stages store A/D conversion data, the interrupt request bit (PCIF) is set.

bit	7	6	5	4	3	2	1	0
Field	Reserved		TEST [1:0]		Reserved		PFS [1:0]	
Attribute	-	-	R	R	-	-	R/W	R/W
Initial value	X	X	X	X	X	X	0	0

[bit 7:6] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 5:4] TEST [1:0]: Test bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 3:2] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 1:0] PFS [1:0]: Priority conversion FIFO stage count setting bits

When A/D conversion data for the FIFO stage count ($N + 1$) set in PFS [1:0] is written, the interrupt request flag (PCIF) is set to "1".

Bit [1:0]	Description
0b00	Generates an interrupt request when conversion result is stored in the first FIFO stage.
0b01	Generates an interrupt request when conversion result is stored in the second FIFO stage.
0b10	Generates an interrupt request when conversion result is stored in the third FIFO stage.
0b11	Generates an interrupt request when conversion result is stored in the fourth FIFO stage.

5.9. Priority Conversion FIFO Data Register (PCFD)

The Priority Conversion FIFO Data Register (PCFD) consists of four FIFO stages and stores analog conversion results. Data can be retrieved sequentially by reading the register.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	RS	PC4	PC3	PC2	PC1	PC0
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit 15:6] PD9:PD0: Priority conversion result

The result of 10-bit priority A/D conversion is written.

[bit 5] RS: Priority A/D start factor

The start factor of the priority conversion corresponding to the conversion result written in PD9 to PD0 is shown.

When RS = "0", it is not possible to indicate whether the conversion was started by software or by a timer.

Bit	Description
0	Software/timer start (priority level 2)
1	External trigger (priority level 1)

[bit 4:0] PC4:PC0: Conversion input channel bits

The analog input channels corresponding to the conversion result written in PD9 to PD0 are written.

Settings for channels not defined in the product specifications are not written. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

<Note>

This register has different bit configurations depending on the FDAS bit setting in the A/D Status Register (ADSR). When the FDAS bit is "1", see "3.2.6 Bit placement selection for FIFO data registers".

To perform a byte access to this register, read the most significant byte (bit 15:8) to shift the FIFO data. Reading the least significant byte (bit 7:0) does not shift FIFO.

Conversion started with an external trigger can be performed only when the analog input channel is between ch.0 to 7.

5.10. Priority Conversion Input Selection Register (PCIS)

The Priority Conversion Input Selection Register (PCIS) is used to select the analog input channels for which priority conversion is performed. For software or timer start at priority level 2, only one channel can be selected from multiple analog input channels. For external trigger start at priority level 1, one channel can be selected from eight channels (ch.0 to ch.7).

bit	7	6	5	4	3	2	1	0
Field	P2A [4:0]					P1A [2:0]		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7:3] P2A [4:0]: Priority level 2 analog input selection

This bit specifies the analog input channel for a start at priority level 2 (software/timer). It can be selected from all channels. It is not possible to set the channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [7:3]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

[bit 2:0] P1A [2:0]: Priority level 1 analog input selection

This bit specifies the analog input channel for a start at priority level 1 (external trigger). It can be selected from eight channels (ch.0 to ch.7).

Bit [2:0]	Description
0b000	ch.0
0b001	ch.1
0b010	ch.2
...	...
0b101	ch.5
0b110	ch.6
0b111	ch.7

5.11. A/D Comparison Value Setup Register (CMRD)

The A/D Comparison Value Setup Register (CMRD) sets the value to be compared with the A/D conversion result. When the conditions set in both this register and the A/D Comparison Control Register (CMPCR) are satisfied, the conversion result comparison interrupt request bit (CMPIF) in the A/D Control Register (ADCR) is set.

bit	15	14	13	12	11	10	9	8
Field	CMAD9	CMAD8	CMAD7	CMAD6	CMAD5	CMAD4	CMAD3	CMAD2
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	0

[bit 15:8] CMAD9:CMAD2: A/D conversion result value setting bits

These bits set the value to be compared with the A/D conversion result.

The most significant eight bits (bit 9:2) of the A/D conversion result are compared with the value in this register (CMAD9 to CMAD2). The least significant two bits (bit 1 and 0) of the A/D conversion result are not compared.

5.12. A/D Comparison Control Register (CMPCR)

The A/D Comparison Control Register (CMPCR) controls the A/D comparison function. When the converted value is compared with the value in the A/D Comparison Value Setup Register (CMPD) and the comparison condition in this register is satisfied, the conversion result comparison interrupt request bit (CMPIF) in the A/D Control Register (ADCR) is set.

bit	7	6	5	4	3	2	1	0
Field	CMPEN	CMD1	CMD0		CCH [4:0]			
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] CMPEN: Conversion result comparison function operation enable bit

This bit enables the operation of the A/D comparison function.

Bit	Description
0	Stops the comparison function operation.
1	Enables the comparison function operation.

[bit 6] CMD1: Comparison mode 1

This bit sets the condition for generating a conversion interrupt request.

Bit	Description
0	Generates an interrupt request when the result is smaller than the CMPD set value.
1	Generates an interrupt request when the result is equal to or greater than the CMPD set value.

[bit 5] CMD0: Comparison mode 0

This bit selects the comparison target. When this bit is "1", the setting of CCH [4:0] is invalid.

Bit	Description
0	Compares the conversion result of the channel set in CCH [4:0].
1	Compares the conversion results of all channels.

[bit 4:0] CCH [4:0]: Comparison target analog input channel

This bit sets the analog channel to be compared. When the CMD0 bit is "1", setting of this bit is invalid. It is not possible to set the channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

5.13. Sampling Time Selection Register (ADSS)

The Sampling Time Selection Register (ADSS3 to 0) allows you to set the sampling time for each bit. Which of the sampling times set in Sampling Time Setup Registers 0 and 1 (ADST0 and 1) is used is specified in this register.

■ ADSS3 (most significant byte: TS31 to TS24) and ADSS2 (least significant byte: TS23 to TS16)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Attribute	R/W															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] TS31:TS16: Sampling time selection bits

Set the sampling time specified in the Sampling Time Setup Register (ADST) for the corresponding channel.

Setting "0" specifies the time set in ADST0 and setting "1" specifies the time set in ADST1. TS31 to TS16 correspond respectively to ch.31 to ch.16.

■ ADSS1 (most significant byte: TS15 to TS8) and ADSS0 (least significant byte: TS7 to TS0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] TS15:TS0: Sampling time selection bits

Set the sampling time specified in the Sampling Time Setup Register (ADST) for the corresponding channel.

Setting "0" specifies the time set in ADST0 and setting "1" specifies the time set in ADST1. TS15 to TS0 correspond respectively to ch.15 to ch.0.

<Note>

It is not possible to write to the ADSS register during A/D conversion.

It is not possible to set "1" in the bit corresponding to a channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

5.14. Sampling Time Setup Register (ADST)

Sampling Time Setup Registers 0 and 1 (ADST0 and 1) set the sampling times for A/D conversion. ADST0 and 1 are provided for setting two sampling times, and which one is used is selected in the Sampling Time Selection Register (ADSS3 to 0).

■ ADST0 (most significant byte)

bit	15	14	13	12	11	10	9	8
Field	STX01	STX00	ST05	ST04	ST03	ST02	ST01	ST00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

[bit 15:14] STX01:STX00: Sampling time N times setting bits

These bits multiply the sampling time set values in the ST05 to ST00 bits by N.

Bit 15	Bit 14	Description
0	0	Set value x 1
0	1	Set value x 4
1	0	Set value x 8
1	1	Set value x 16

[bit 13:8] ST05:ST00: Sampling time setting bits

These bit set the sampling time for A/D conversion.

Sampling time = PCLK cycle × (ST set value + 1) × STX setting multiplier

Example: When ST05 to ST00 = 9, STX01 and STX00 = 01 (4 times), and PCLK = 20 MHz (50 ns),
Sampling time = $50 \text{ ns} \times (9 + 1) \times 4 = 2 \mu\text{s}$

<Note>

It is not possible to write to the ADST0 register during A/D conversion.

When STX01 and STX00 = 00 (ST05 to ST00 set values multiplied by 1) are set, set ST05 to ST00 to "3" or more ("2" or less must not be set). Furthermore, ADST0 must be set so that the sampling time conditions in "Electric Characteristics" of the Data Sheet are satisfied.

■ ADST1 (least significant byte)

bit	7	6	5	4	3	2	1	0
Field	STX11	STX10	ST15	ST14	ST13	ST12	ST11	ST10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

[bit 7:6] STX11:STX10: Sampling time N times setting bits

These bits multiply the sampling time set values in the ST15 to ST10 bits by N.

Bit 7	Bit 6	Description
0	0	Set value x 1
0	1	Set value x 4
1	0	Set value x 8
1	1	Set value x 16

[bit 5:0] ST15:ST10: Sampling time setting bits

These bit set the sampling time for A/D conversion.

Sampling time = PCLK cycle × (ST set value + 1) × STX setting multiplier

Example: When ST15 to ST10 = 9, STX11 and STX10 = 01 (4 times), and PCLK = 20 MHz (50 ns),
Sampling time = $50 \text{ ns} \times (9 + 1) \times 4 = 2 \mu\text{s}$

<Note>

It is not possible to write to the ADST1 register during A/D conversion.

When STX11 and STX10 = 00 (ST15 to ST10 set values multiplied by 1) are set, set ST15 to ST10 to "3" or more ("2" or less must not be set). Furthermore, ADST1 must be set so that the sampling time conditions in "Electric Characteristics" of the Data Sheet are satisfied.

5.15. Comparison Time Setup Register (ADCT)

The Comparison Time Setup Register (ADCT) sets the comparison time, which is part of the A/D conversion time.

bit	15	14	13	12	11	10	9	8
Field	Reserved					CT2	CT1	CT0
Attribute	-	-	-	-	-	R/W	R/W	R/W
Initial value	X	X	X	X	X	1	1	1

[bit 15:11] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 10:8] CT2:CT0: Comparison time setting bits

These bits set the comparison time of the A/D conversion time.

The comparison time setting is common to Sampling Setup Registers 0 and 1.

Comparison time = $\{(CT \text{ set value } (N) + 1) \times 10 + 4\} \times \text{PCLK cycle}$

Example: When the CT set value = 1 and PCLK = 20 MHz (50 ns),

Comparison time = $\{(1 + 1) \times 10 + 4\} \times 50 = 1.2 \mu\text{s}$

<Note>

It is not possible to write to the ADCT register during A/D conversion.

Chapter: 12-bit A/D Converter

This chapter explains the functions and operations of the 12-bit A/D converter.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup procedure examples
5. Registers

1. Overview

The 12-bit A/D converter is a function that converts analog input voltages into 12-bit digital values using a type of the RC Successive Approximation Register.

■ Features of the 12-bit A/D converter

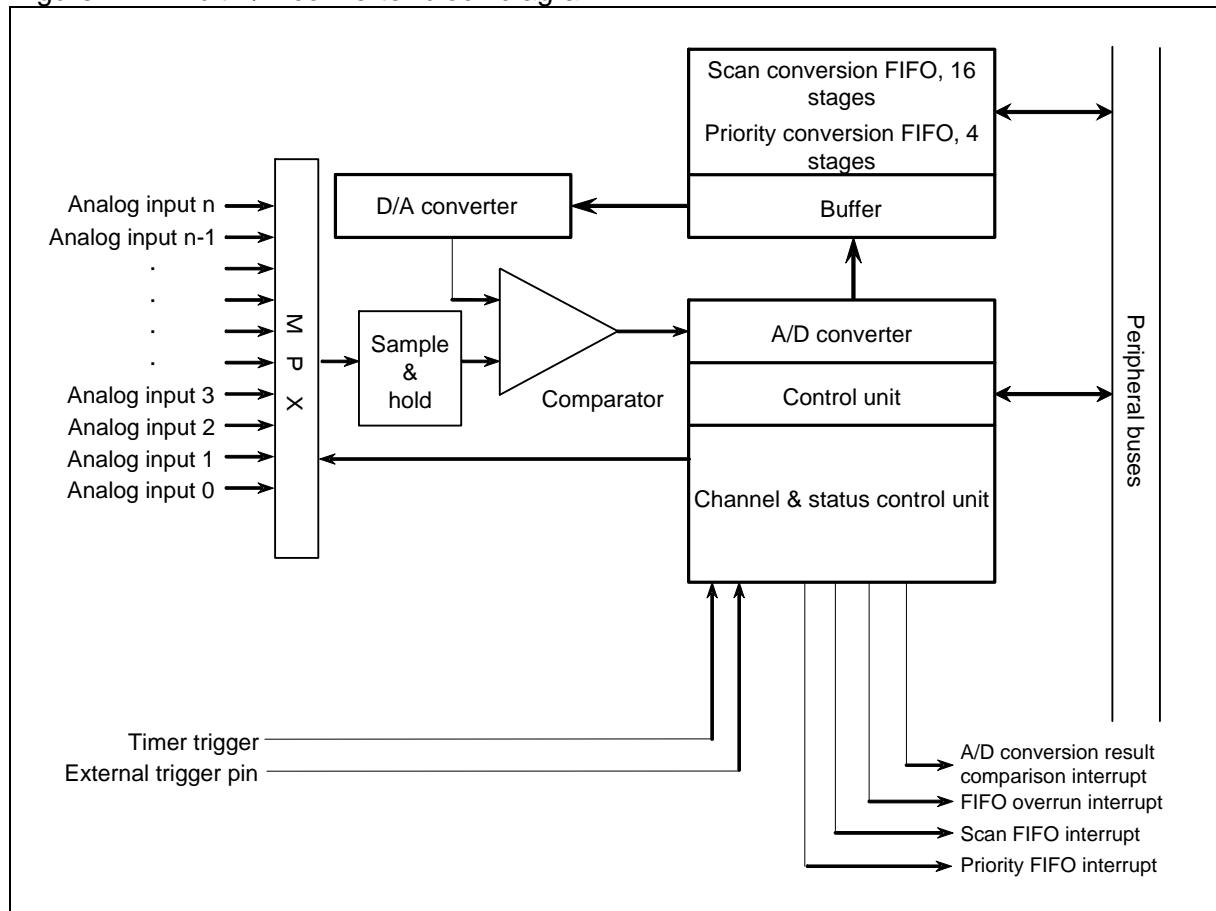
- 12-bit resolution
- Converter using a type of RC Successive Approximation Register with sample and hold circuits
- Conversion time of 1.0 μ s (at a base clock (HCLK) frequency of 72 MHz)
- Two sampling times selectable for each input channel
- Scan conversion operation:
 - Multiple analog inputs can be selected from multiple channels.
 - Start factors are software and timers.
 - Repeat mode is available.
- Priority conversion operation:
 - Even during scan operation, if a start factor of priority conversion occurs, it is possible to interrupt the ongoing scan conversion and perform conversion with high priority (There are two priority levels: 1 and 2. Priority level 1 is higher than priority level 2.).
 - Start factors are software and timers (priority level 2), and external triggers (priority level 1).
- FIFO function:
 - Sixteen FIFO stages for scan conversion and four FIFO stages for priority conversion are incorporated.
 - An interrupt is generated when data is written in the specified count of FIFO stages.
- Changeable A/D conversion data placement (selectable between shift to the MSB side and shift to LSB side)
- The A/D conversion result comparison function is available.
- There are four interrupt sources as follows:
 1. Scan conversion FIFO stage count interrupt
 2. Priority conversion FIFO stage count interrupt
 3. FIFO overrun interrupt (for both scan and priority conversion processes)
 4. A/D conversion result comparison interrupt
- DMA transfer triggered by an interrupt request

2. Configuration

This section provides the configuration of the 12-bit A/D converter.

■ 12-bit A/D converter block diagram

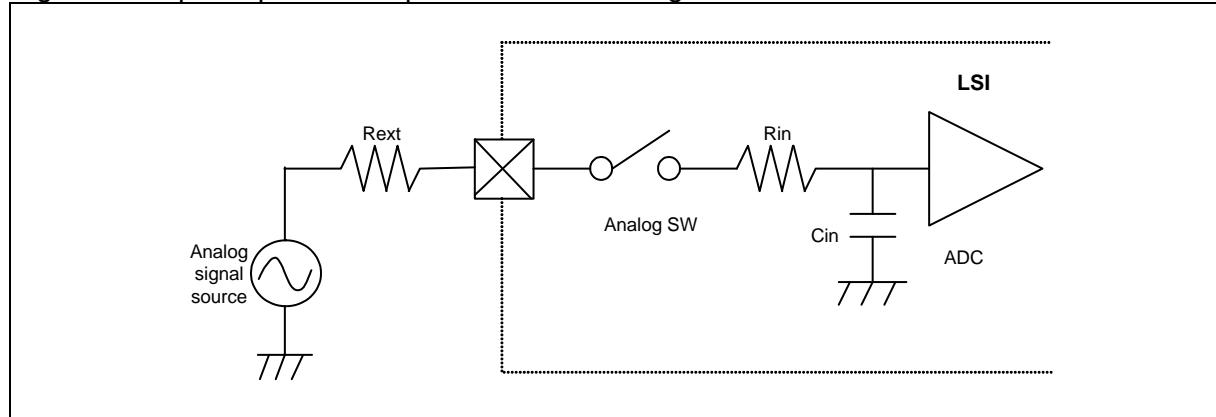
Figure 2-1 12-bit A/D converter block diagram



■ Input impedance

The sampling circuit of the A/D converter is shown as an equivalent circuit in Figure 2-2. Refer to the "Electrical Characteristics" in the data sheet to make sure that the external impedance R_{ext} should be selected not to exceed the sampling time.

Figure 2-2 Input impedance equivalence circuit diagram



3. Explanation of Operations

This chapter explains the operations of the 12-bit A/D converter.

- 3.1 Enabling operations of the A/D converter
- 3.2 A/D conversion operation
- 3.3 FIFO operations
- 3.4 A/D comparison function
- 3.5 Starting DMA

3.1. Enabling operations of the A/D converter

This section explains enabling operations of the A/D converter

The A/D converter must be in the operation enable state prior to A/D conversion. Writing "1" to the ENBL bit of the ADCEN register turns the A/D converter from the operation stop state to the operation enable state after the period of operation enable state transitions. On the other hand, writing "0" to the ENBL bit of the ADCEN register turns the A/D converter immediately to the operation stop state.

A/D conversion can be performed only in the operation enable state. An A/D conversion request in the operation stop state is ignored. If the A/D converter enters the operation stop state during A/D conversion, A/D conversion stops immediately.

Reading the READY bit of the ADCEN register allows you to check whether the A/D converter is in the operation enable state or not.

<Note>

When setting the CPU to the timer mode or the stop mode, set the ENBL bit to "0" and turn the A/D converter to the operation stop state.

3.2. A/D conversion operation

The A/D converter can perform two types of conversion processes: scan conversion and priority conversion.

3.2.1 Scan conversion operation

3.2.2 Priority conversion operation

3.2.3 Priority levels and state transitions

3.2.1. Scan conversion operation

This section explains the scan conversion operation.

The input channels are selected in the Scan Conversion Input Selection Register (SCIS). By setting the corresponding bit in the SCIS to "1", any necessary channel can be selected from among multiple analog input channels.

The A/D converter can be started by software or a timer. To start the converter by software, set the SSTR bit in the SCCR register to "1". Then conversion starts. To start the converter by timers, set the SHEN bit in the SCCR register to "1" to enable timer start. Conversion starts when the timer's rising edge is detected. When conversion starts, the SCS bit in the ADSR register is set to "1". When the conversion is completed, the SCS bit is reset to "0".

When the SSTR bit in the SCCR register is set to "1" again during A/D conversion or the timer's rising edge is detected again while timer start is enabled, the ongoing conversion operation is immediately stopped and initialized and the A/D conversion is performed again (the operation is restarted).

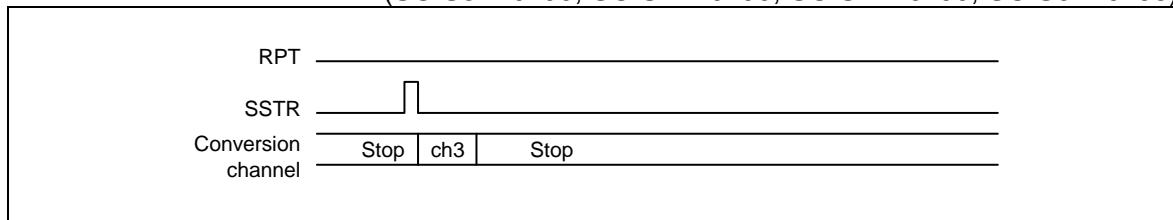
The available scan conversion modes are as follows:

1. One-shot mode for a single channel

This mode is selected when only one analog priority conversion is specified for scan conversion and RPT = 0 in the SCCR register. When the selected priority conversion is completed, the operation stops.

Figure 3-1 Stop of operation in one-shot mode for a single channel

(SCIS3 = 0x00, SCIS2 = 0x00, SCIS1 = 0x00, SCIS0 = 0x08)

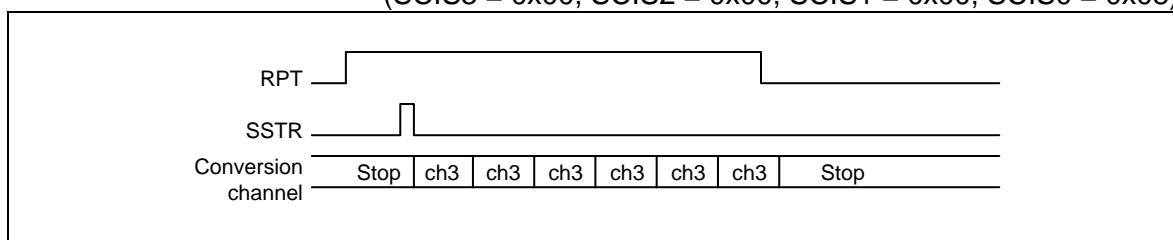


2. Continuous mode for a single channel

This mode is selected when only one analog priority conversion process is specified for scan conversion and RPT = 1 in the SCCR register. When the selected priority conversion is completed, the same priority conversion is started again. To stop A/D conversion, set RPT to "0". The operation stops when the ongoing A/D conversion is completed.

Figure 3-2 Stop of operation in continuous mode for a single channel

(SCIS3 = 0x00, SCIS2 = 0x00, SCIS1 = 0x00, SCIS0 = 0x08)

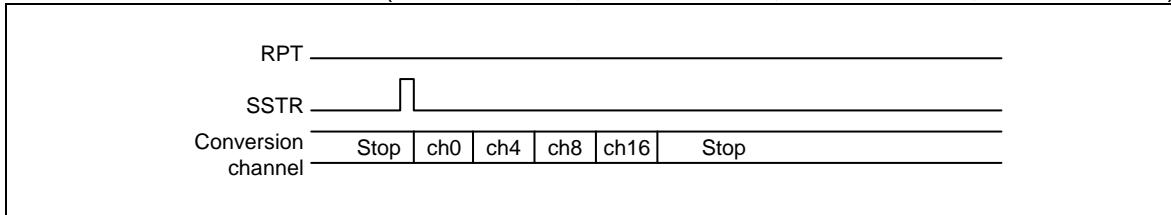


3. One-shot mode for multiple channels

This mode is selected when multiple analog channels are specified for scan conversion and RPT = 0 in the SCCR register. When the conversion starts, the existence of each channel is automatically checked. While the channels are switched from one to another, A/D conversion is started and the conversion result is written to FIFO when the conversion is completed. The conversion channels are selected in descending order of channel number (starting from ch.0). Channels not selected in the SCIS register are skipped and the conversion operation targets the next selected channel. When the A/D conversion of the last one of the selected channels is completed, the A/D conversion is stopped.

Figure 3-3 Stop of operation in one-shot mode for multiple channels

(SCIS3 = 0x00, SCIS2 = 0x01, SCIS1 = 0x01, SCIS0 = 0x11)

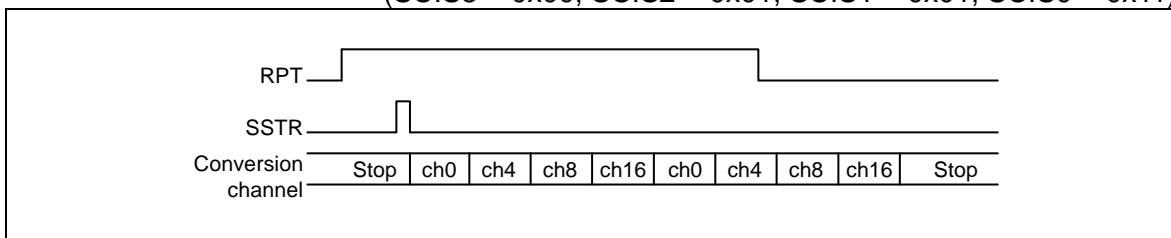


4. Continuous mode for multiple channels

This mode is selected when multiple analog channels are specified for scan conversion and RPT = 1 in the SCCR register. When the conversion starts, the existence of each channel is automatically checked. While the channels are switched from one to another, A/D conversion is started and the conversion result is written to FIFO when the conversion is completed. The conversion channels are selected in descending order of channel number (starting from ch.0). Channels not selected in the SCIS register are skipped and the conversion operation targets the next selected channel. When the A/D conversion of the last one of the selected channels is completed, the conversion operation starts again from ch.0. To end A/D conversion, set RPT "0". The operation stops when the A/D conversion of the last one of the selected channels is completed.

Figure 3-4 Stop of operation in continuous mode for multiple channels

(SCIS3 = 0x00, SCIS2 = 0x01, SCIS1 = 0x01, SCIS0 = 0x11)



3.2.2. Priority conversion operation

This section explains the priority conversion operation.

This mode is used to give priority to a specific conversion process. Even when scan conversion is in progress, if priority conversion is started, the scan conversion is interrupted immediately and the priority conversion is performed. When the priority conversion is completed, the scan operation restarts from the channel where it was interrupted. If conversion with higher priority (priority level 1) is started while the conversion with lower priority (priority level 2) is performed, the priority level 2 conversion is interrupted immediately and the priority level 1 conversion is performed. When the priority level 1 conversion is completed, the priority level 2 conversion is restarted.

Two levels of priority are given to priority conversion. Priority level 1 is the highest and level 2 is the second. Trigger start by an external pin is assigned as the start factor at priority level 1 and software/timer start is assigned as that at priority level 2.

The input channels are selected in the Priority Conversion Input Selection (PCIS) register.

- The procedure for selecting channels at priority level 1 differs depending on the ESCE bit in the Priority Conversion Control (PCCR) register.
 - When ESCE = 0: The P1A [2:0] bits in the PCIS register are used. Only one of the eight channels, ch.0 to 7, can be selected.
 - When ESCE = 1: The setting of the P1A [2:0] bits in the PCIS register is ignored. Only one of the eight channels, ch.0 to 7, can be selected with input from the external pin (ECS [2:0]).

Example: ECS [2:0] = 0b000 -> ch.0
= 0b010 -> ch.2
= 0b111 -> ch.7

- The P2A [4:0] bits in the PCIS register are used for selecting the channel at priority level 2. Only one of the multiple input channels can be selected.

The start factor of A/D conversion differs depending on the priority level.

- Priority level 1 (highest priority) conversion can be started by a falling edge of external trigger input. To enable external trigger start, set the PEEN bit to "1" in the PCCR register.
- Priority level 2 conversion can be started by software or a timer. To start conversion by software, set the PSTR bit in the PCCR register to "1". To start conversion by a timer, set the PHEN bit in the PCCR register to "1" to enable timer start. Conversion starts when the timer's rising edge is detected. When conversion starts, the PCS bit in the ADSR register is set to "1". When the conversion is completed, the PCS bit is reset to "0".

In priority conversion mode, the conversion cannot be restarted. In addition, start factors at the same priority level are ignored.

(A timer start factor is ignored during software-started operation.)

If a priority level 1 start factor (external trigger) occurs during conversion started by a priority level 2 start factor (software or timer), the PCNS bit in the A/D Status Register (ADSR) is set to "1" and the priority level 2 conversion is interrupted immediately. When the priority level 1 conversion is completed, PCNS is reset to "0" and the interrupted priority level 2 conversion is restarted. If a priority level 2 start factor occurs during priority level 1 conversion, the priority level 2 start factor is reserved (retained) and PCNS is set to "1". When the priority level 1 conversion is completed, PCNS is reset to "0" and the priority level 2 conversion is started.

Priority conversion can only be performed in one-shot mode for a single channel.

3.2.3. Priority levels and state transitions

This section explains priority levels and state transitions.

■ Priority levels

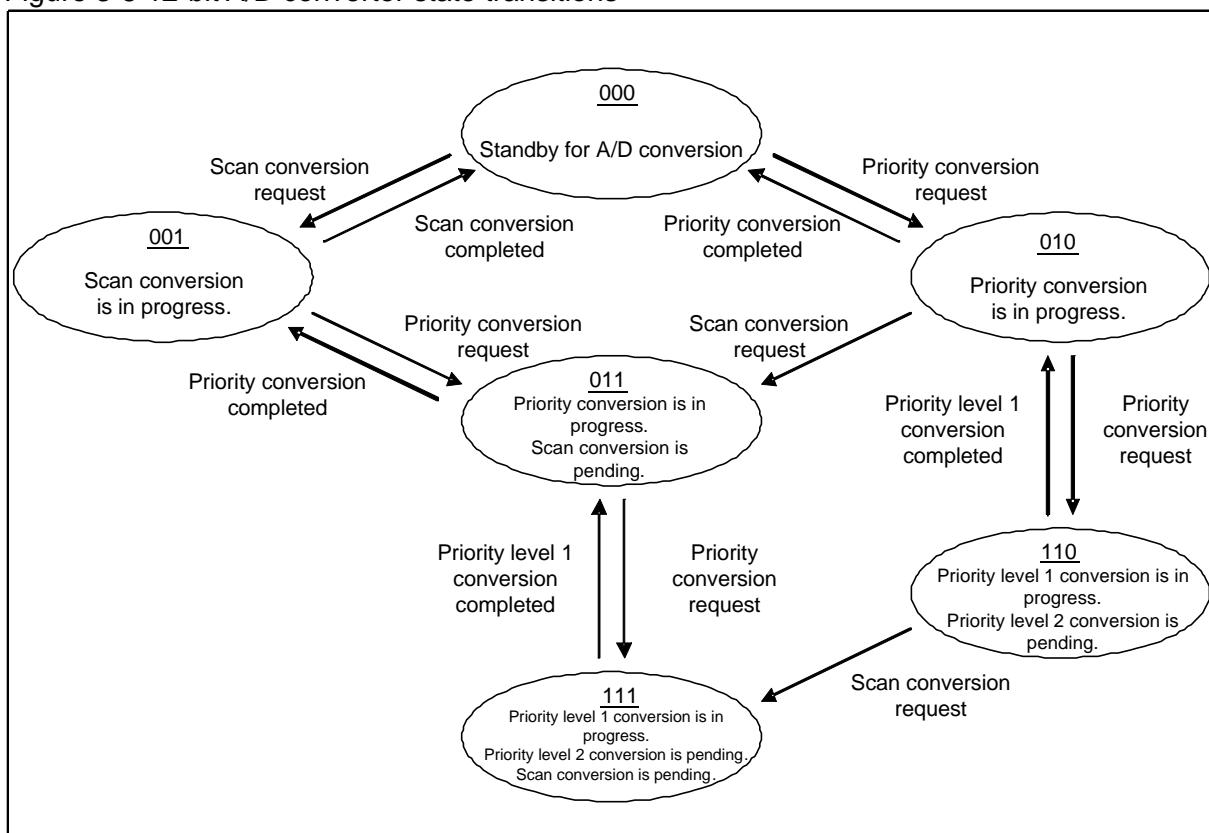
Table 3-1 Priority levels for the A/D converter

Priority level	Conversion type	Start factor
1	Priority level 1 conversion	<ul style="list-style-type: none">Input from external trigger pin (at falling edge)
2	Priority level 2 conversion	<ul style="list-style-type: none">Software (when the PSTR bit is set to "1")Trigger input from timer (at rising edge)
3	Scan conversion	<ul style="list-style-type: none">Software (when the SSTR bit is set to "1")Trigger input from timer (at rising edge)

- When a startup by priority conversion occurs during scan conversion
 - The scan conversion operation is interrupted and priority conversion operation is performed. When the priority conversion operation is completed, the scan conversion is restarted from the channel where it was interrupted.
- When a startup at priority level 1 occurs during conversion at priority level 2
 - The priority level 2 conversion is interrupted and the operation by the startup at priority level 1 is performed. When the priority level 1 operation is completed, the priority level 2 conversion is restarted automatically.
- When a startup at priority level 2 occurs during conversion at priority level 1
 - The start factor at priority level 2 is retained. When the priority level 1 conversion is completed, the priority level 2 conversion is started automatically.
- When a startup of scan conversion occurs during priority level 1 conversion
 - The start factor of the scan conversion is retained. When the priority level 1 conversion is completed, the scan conversion operation is started automatically.
- When a startup of scan conversion occurs during priority level 2 conversion
 - The start factor of the scan conversion is retained. When the priority level 2 conversion is completed, the scan conversion operation is started automatically.
- While priority conversion is performed, start factor at the same priority level are masked (the operation is not restarted).

■ State transitions

Figure 3-5 12-bit A/D converter state transitions



The operation states can be read from the SCS, PCS, and PCNS bits of the ADSR register.

Table 3-2 Correspondence between bits and operation states

PCNS	PCS	SCS	Explanation of states
0	0	0	Standby for A/D conversion.
0	0	1	Scan A/D conversion is in progress.
0	1	0	Priority A/D conversion (priority level 1 or 2) is in progress.
0	1	1	Priority A/D conversion (priority level 1 or 2) is in progress. Scan conversion is pending.
1	1	0	Priority A/D conversion (priority level 1) is in progress. Priority conversion (priority level 2) is pending.
1	1	1	Priority A/D conversion (priority level 1) is in progress. Scan conversion and priority conversion (priority level 2) are pending.

3.3. FIFO operations

The A/D converter has 16 FIFO stages for scan conversion and 4 FIFO stages for priority conversion. When conversion data is written in the specified count of FIFO stages, an interrupt is generated to the CPU.

- 3.3.1 FIFO operations in scan conversion
- 3.3.2 Interrupts in scan conversion
- 3.3.3 FIFO operations in priority conversion
- 3.3.4 Interrupts in priority conversion
- 3.3.5 Validity of FIFO data
- 3.3.6 Bit placement selection for FIFO data registers

3.3.1. FIFO operations in scan conversion

This section explains FIFO operations in scan conversion.

Sixteen FIFO stages are incorporated for writing scan conversion data. After reset, they are in empty state and the SEMP bit in the Scan Conversion Control Register is set to "1". When A/D conversion of one channel is completed, the conversion result, start factor, and conversion channel are written in the first FIFO stage. This resets SEMP to "0". The conversion result, start factor, and conversion channel for the next channel are written sequentially in the second FIFO stage.

When such data is written in all of the 16 stages, the SFUL bit is set to "1" to indicate that FIFO is in full state. If conversion is performed and an attempt is made to write data in FIFO when FIFO is in full state, the SOVR bit is set to "1" and the data is discarded (cannot overwrite the existing data).

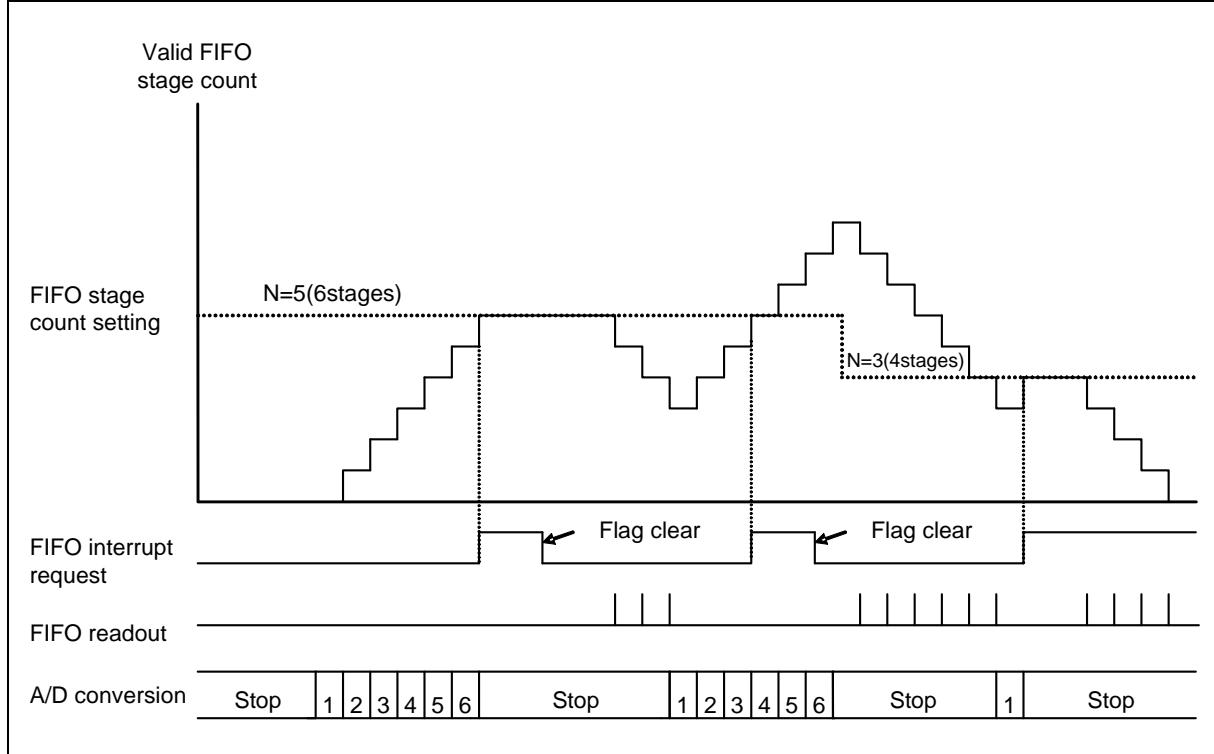
To clear the data in FIFO, set the SFCLR bit in the Scan Conversion Control register to "1". FIFO goes to the empty state and the SEMP bit is set to "1".

Data in FIFO can be read sequentially by reading the Scan FIFO Data Register (SCFD). To perform a byte (8 bits) access to this register, read the most significant byte (bit 31:24) to shift FIFO (reading the other bytes (bit23:16, bit15:8, bit7:0) does not shift FIFO). To perform a half word (16 bits) access to this register, read the most significant half word (bit 31:16) to shift FIFO (reading the other byte (bit 15:0) does not shift FIFO). Performing a word (32 bits) access to this register shifts FIFO.

3.3.2. Interrupts in scan conversion

This section explains interrupts in scan conversion.

Figure 3-6 FIFO interrupt settings and FIFO operations



When conversion data for the number of FIFO stages ($N + 1$) set in SFS [3:0] in the Scan Conversion FIFO Stage Count Setup Register (SFNS) is written in FIFO, the interrupt request bit (SCIF) in the A/D Control Register (ADCR) is set to "1". If the interrupt enable bit (SCIE) is set to "1", an interrupt request is generated to the CPU.

The following explains FIFO stage count interrupt methods for each scan conversion mode.

1. One-shot mode for a single channel

To generate an interrupt after the completion of one conversion process for the specified channel, set SFS [3:0] = 0x0. When conversion data is written in the first FIFO stage, SCIF is set to "1".

<Note>

If SFS [3:0] is set to 0x1 or more (two stages or more), interrupts are not generated until conversion data is written into FIFO by the specified stage count.

2. Continuous mode for a single channel

To generate an interrupt after the completion of one conversion process for the specified channel, set SFS [3:0] = 0x0. When conversion data is written in the first FIFO stage, SCIF is set to "1".

To generate an interrupt at the completion of a number of times of conversion of the specified channel, set SFS [3:0] to 0x1 or more (two stages or more). For example, set SFS [3:0] = 0x3 to generate an interrupt after four repeats.

3. One-shot mode for multiple channels

To generate an interrupt after the completion of conversion of the multiple specified channels, set the FIFO stage count according to the number of channels. If eight channels are selected, set the FIFO stage count by setting SFS [3:0] = 0x7. When the conversion of the last one of the selected channels is completed, SCIF is set to "1".

An interrupt can be generated at any timing before scan completion by setting SFS [3:0] to a value less than the number of selected channels.

4. Continuous mode for multiple channels

To generate an interrupt after the completion of the first scan of the multiple specified channels, set the FIFO stage count according to the number of channels. If eight channels are selected, set the FIFO stage count by setting SFS [3:0] = 0x7. When the conversion of the last one of the selected channels is completed, SCIF is set to "1".

To generate an interrupt after the completion of the second scan, set the FIFO stage count to twice the number of selected channels. For example, when four channels are selected, set the FIFO stage count to 8 (SFS [3:0] = 0x7). An interrupt is generated when the second scan is completed.

Because the FIFO stage count can be set to any value, an interrupt can be generated at any desired timing.

3.3.3. FIFO operations in priority conversion

This section explains FIFO operations in priority conversion.

Four FIFO stages are incorporated for writing priority conversion data. After reset, they are in empty state and the PEMP bit in the Priority Conversion Control Register is set to "1". When one A/D conversion process is completed, the conversion result, start factor, and conversion channels are written in the first FIFO stage. This resets SEMP to "0". The conversion result and conversion channels for the subsequent conversion processes are written in the corresponding FIFO stages.

When such data is written in all of the 4 stages, the PFUL bit is set to "1" to indicate that FIFO is in full state. If conversion is performed and an attempt is made to write data in FIFO when FIFO is in full state, the POVR bit is set to "1" and the data is discarded (cannot overwrite the existing data).

To clear the data in FIFO, set the PFCLR bit in the Priority Conversion Control Register to "1". FIFO goes to the empty state and the PEMP bit is set to "1".

Data in FIFO can be read sequentially by reading the Priority FIFO Data Register (PCFD). To perform byte (8 bits) access to this register, read the most significant byte (bit 31:24) to shift FIFO (reading the other bytes (bit23:16, bit15:8, bit7:0) does not shift FIFO). To perform a half word (16 bits) access to this register, read the most significant half word (bit 31:16) to shift FIFO (reading the other byte (bit 15:0) does not shift FIFO). Performing a word (32 bits) access to this register shifts FIFO.

3.3.4. Interrupts in priority conversion

This section explains interrupts in priority conversion.

When conversion data for the number of FIFO stages ($N + 1$) set in PFS [1:0] in the Priority Conversion FIFO Stage Count Setup Register (PFNS) is written in FIFO, the interrupt request bit (PCIF) in the A/D Control Register (ADCR) is set to "1". If the interrupt enable bit (PCIE) is set to "1", an interrupt request is generated to the CPU.

The following explains FIFO stage count interrupt methods in priority conversion.

To generate an interrupt after the completion of one conversion process for the specified channel, set PFS [1:0] = 0x0. When conversion data is written in the first FIFO stage, PCIF is set to "1".

<Note>

If PFS[1:0] is set to 0x1 or more (two stages or more), interrupts are not generated until conversion data is written into FIFO by the specified stage count.

3.3.5. Validity of FIFO data

This section explains restrictions on reading FIFO data registers.

The bit 12 of the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) comes with the INVL (A/D conversion result disable) bit which indicates data validity. During reading FIFO data registers, the INVL bit is set to "0" if data is valid while the INVL bit is set to "1" if data is invalid.

For word (32 bits) reading, data validity can be checked by the INVL bit.

For half word (16 bits) reading which does not use interrupts or empty bits (SEMP, PEMP), always start reading from the least significant 16 bits including the INVL bit. If the INVL bit is "1" at this time, reading the most significant 16 bits is prohibited. The most significant 16 bits must be read only when the INVL bit is "0".

For byte (8 bits) reading which does not use interrupts or empty bits (SEMP, PEMP), always start reading from bit15:8 including the INVL bit. If the INVL bit is "1" at this time, reading bit 31:24, bit 23:16, or bit 7:0 is prohibited. They must be read only when the INVL bit is "0".

3.3.6. Bit placement selection for FIFO data registers

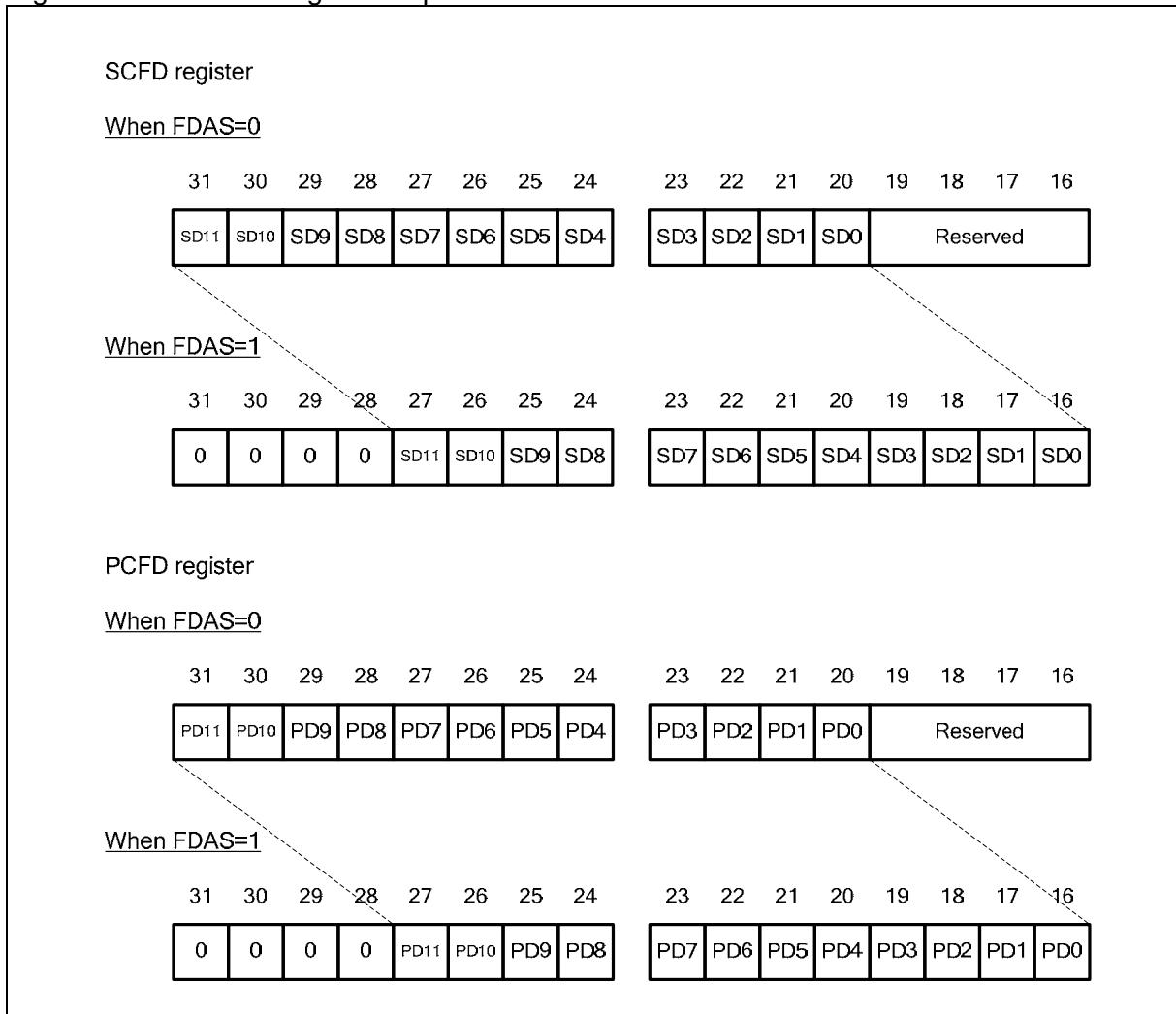
This section explains bit placement selection for FIFO data registers.

The A/D converter can change the bit placement for the conversion results in the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) with the FDAS bit in the A/D Status Register (ADSR) (Figure 3-7).

Setting the FDAS bit to "1" places 12-bit A/D conversion results (SD11 to SD0, PD11 to PD0) on the LSB side (bit 27:16) when a FIFO data register is read. Placement of the least significant 16 bits of a FIFO data register does not change.

FIFO is shifted, regardless of the set value of the FDAS bit, by reading bit 31:24 (for a byte access), bit 31:16 (for a half word access), or bit 31:0 (for a word access) of a FIFO data register.

Figure 3-7 FIFO data register bit placement



3.4. A/D comparison function

The A/D comparison function compares A/D conversion results and generates interrupts.

To use the comparison function, set the CMPEN bit in the A/D Comparison Control Register (bit 7 in the CMPCR register) to "1".

The values set in the A/D Comparison Value Setup Register (CMPD) are compared with the most significant 10 bits (AD11:2) of the A/D conversion result. If the comparison result satisfies the conditions set in the A/D Comparison Control Register (CMPCR), the A/D comparison interrupt bit (CMPIF) in the ADCR register is set to "1". If the interrupt enable bit (CMPIE) is "1", an interrupt is generated to the CPU.

<Note>

Two bits (bit 1, bit 0) on the LSB side are not compared.

Because the result of A/D conversion, regardless of scan or priority, is compared before it is written to FIFO, comparison is possible when FIFO is full.

If CMD1 is set to "1" (to generate an interrupt when the result is equal to or more than the CMPD set value), CMPIF is set to "1" when the conversion result is equal to the value in the A/D Comparison Value Setup Register.

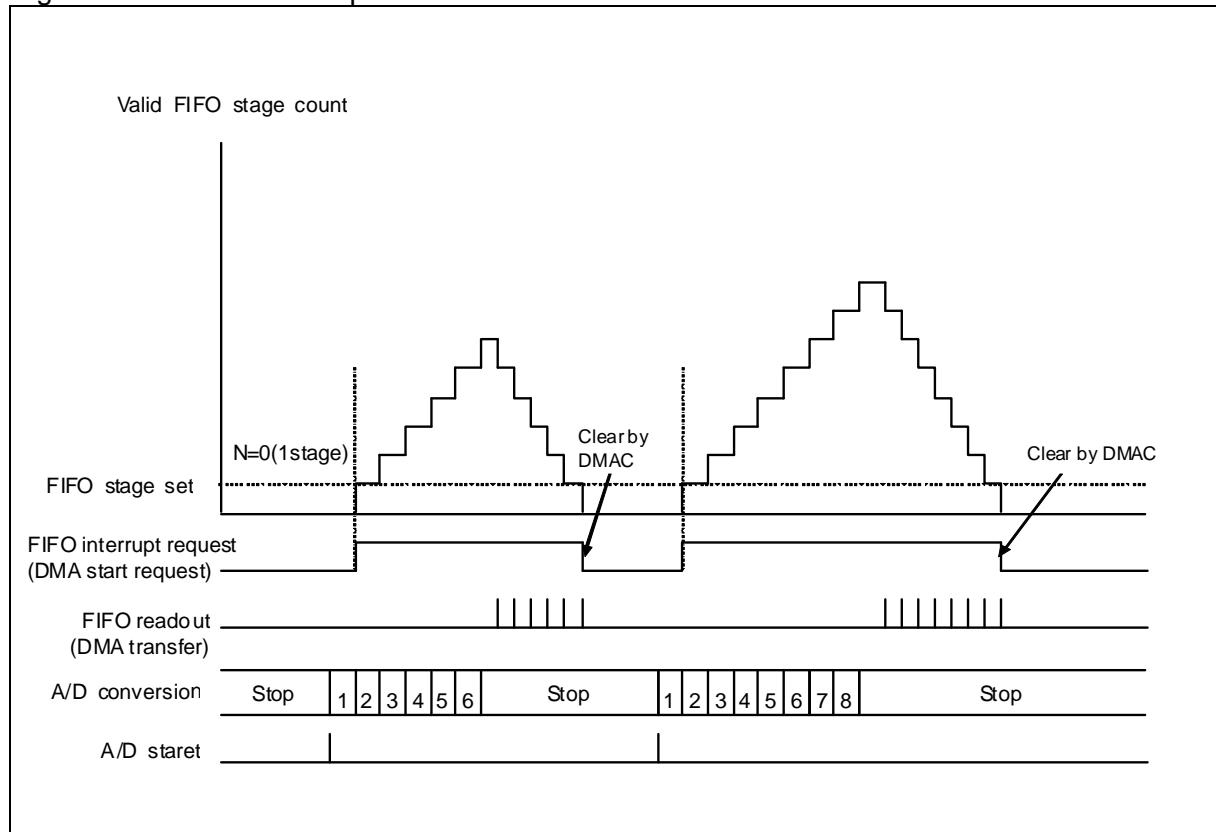
3.5. Starting DMA

The A/D converter can start DMA transfer with a scan conversion FIFO stage count interrupt request.

The A/D converter can transfer scan FIFO data by connecting the interrupt signal of scan conversion from the A/D converter to DMA and starting DMA. By setting the scan FIFO stage count for interrupt generation to "0" (an interrupt is generated when a conversion result is stored in the first FIFO stage), DMA transfer can be performed in conjunction with A/D conversion.

The setting in the DMA Transfer Request Selection Register of the interrupt controller as to whether the A/D converter scan conversion interrupt signal is connected to the CPU or DMAC should be made for DMA transfer.

Figure 3-8 DMA transfer operation



4. Setup procedure examples

This section provides examples of setup procedures for the 12-bit A/D converter.

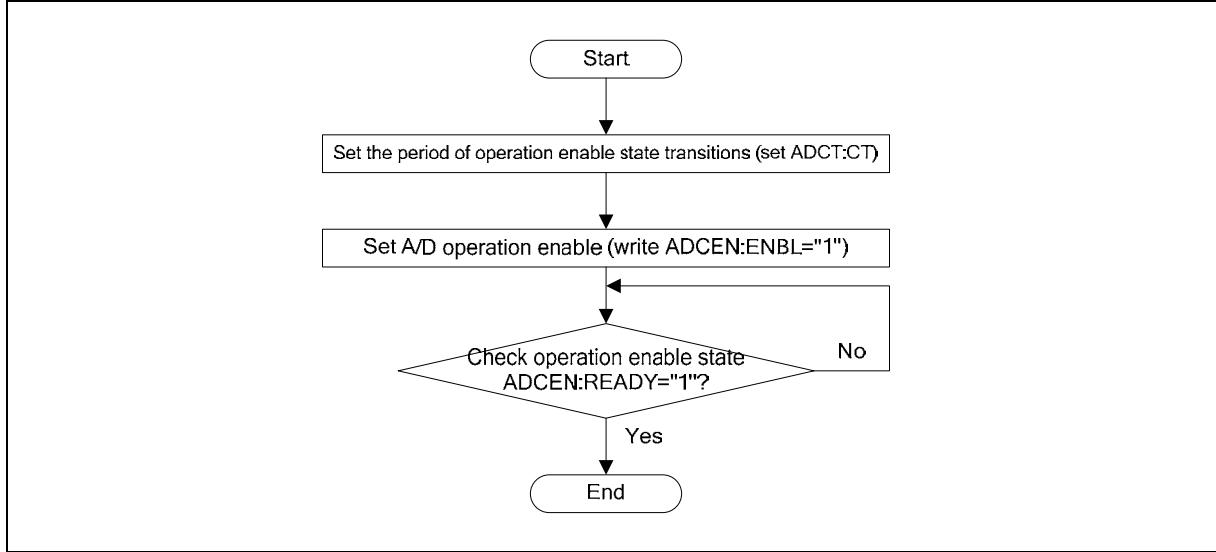
- 4.1 A/D Operation Enable Setup Procedure Example
- 4.2 Scan conversion setup procedure example
- 4.3 Priority conversion setup procedure example
- 4.4 Setting the conversion time

4.1. A/D Operation Enable Setup Procedure Example

This section provides an A/D operation enable setup procedure example.

- Set the period of operation enable state transitions
- Poll the operation enable state

Figure 4-1 A/D Operation Enable Setup Procedure Example

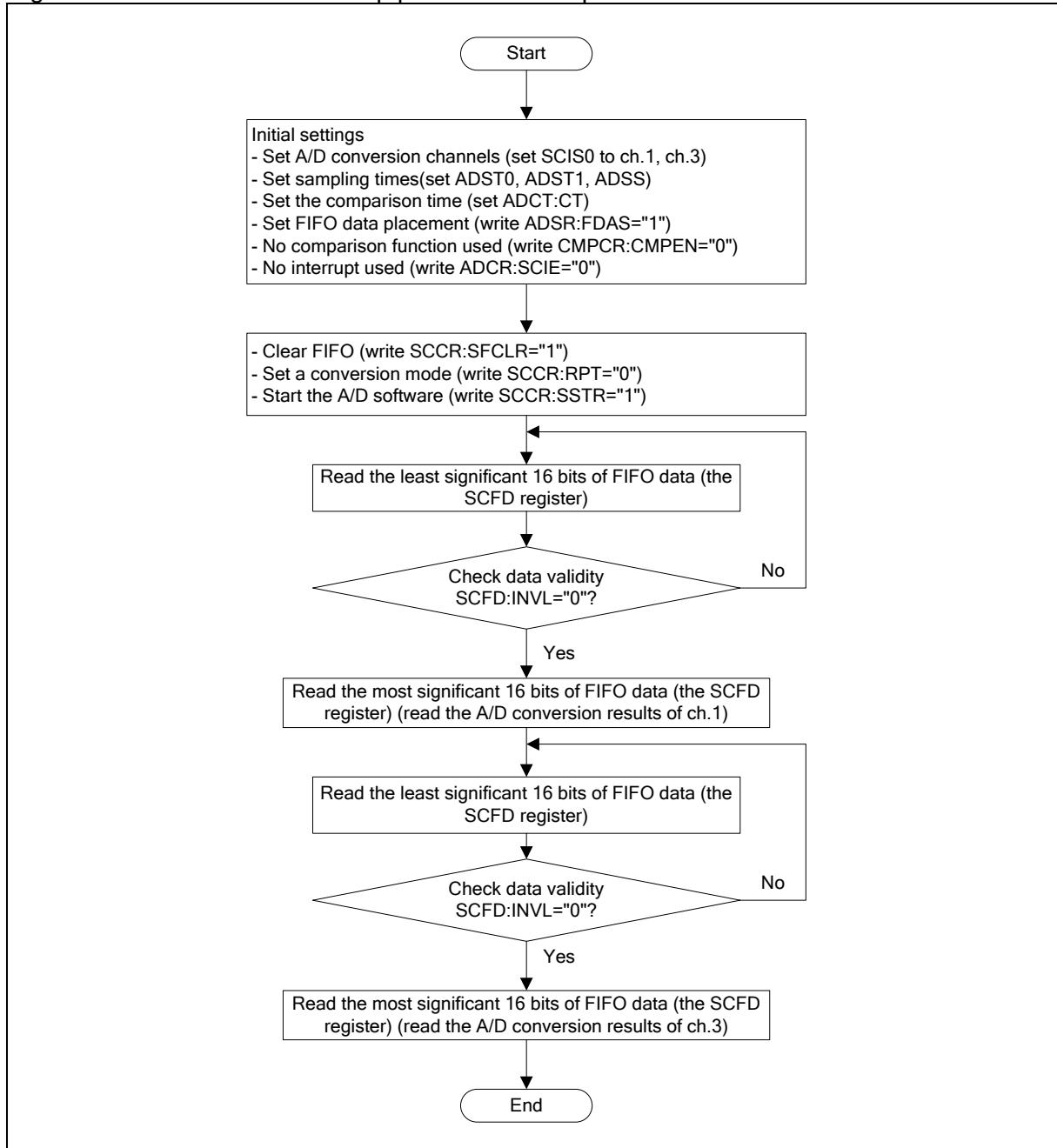


4.2. Scan conversion setup procedure example

This section provides a scan conversion setup procedure example.

- Scan conversion by software startup
- Set A/D conversion channels to ch.1 and ch.3
- Set different sampling times for ch.1 and ch.3
- Set the comparison time
- Read the least significant 16 bits of FIFO data and check data validity by the INVL bit
- After checking that data is valid, read the most significant 16 bits of FIFO data

Figure 4-2 Scan conversion setup procedure example

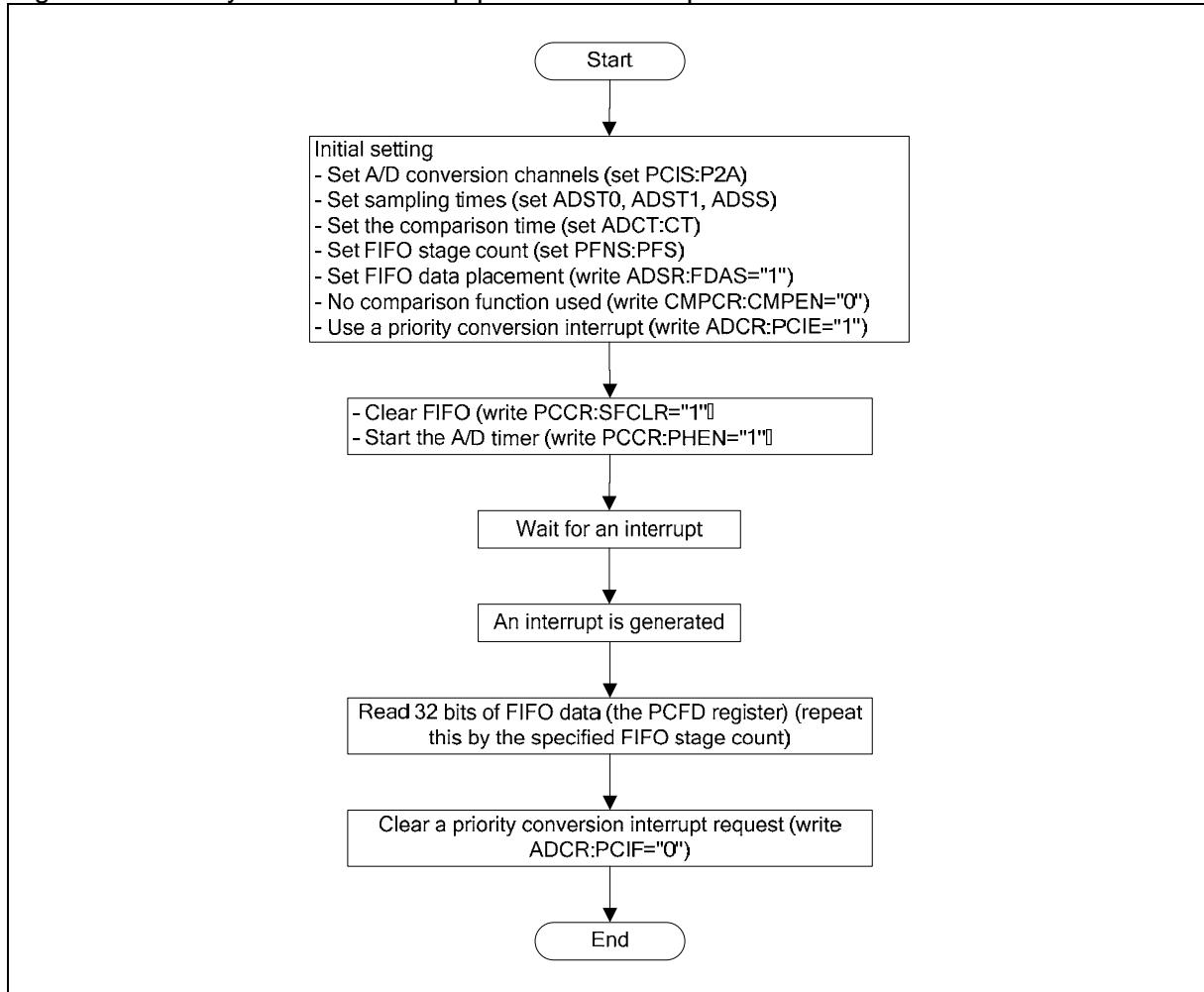


4.3. Priority conversion setup procedure example

This section provides a priority conversion setup procedure example.

- Priority conversion at priority level 2 by timer start
- Conversion channels are ch.1 and ch.3
- Set different sampling times for ch.1 and ch.3
- Set the comparison time
- Read 32 bits of FIFO data by using an interrupt
- Read FIFO by the specified stage count

Figure 4-3 Priority conversion setup procedure example



4.4. Setting the conversion time

The conversion time of the A/D converter is "sampling time" + "comparison time". Two sampling time settings can be applied to each channel. This section explains how to set and calculate the conversion time.

■ Example of setting the sampling time

A sampling time is set in each of Sampling Time Setup Registers 0 and 1 (ADST0 and ADST1). Using Sampling Time Selection Registers (ADSS3 to 0), whether Sampling Time Setup Registers 0 or 1 is used to provide the value can be selected for each channel. This allows you to set different sampling times for channels with different external impedances.

$$\text{Sampling time} = \text{Base clock (HCLK) cycle} \times \{(\text{ST set value} + 1) \times \text{STX setting multiplier} + 1\}$$

<Notes>

- When STXx2, STXx1, and STXx0 = 000 (STx4 to STx0 set values multiplied by 1) are set, set STx4 to STx0 to "3" or more ("2" or less must not be set).
- For setting the sampling time, refer to the "Electrical Characteristics" in the data sheet to make sure that an appropriate time should be selected in accordance with an external impedance of an input channel, an analog power supply voltage (AVCC), and a base clock (HCLK) cycle.

■ Example of setting the comparison time

The comparison time is set in the Comparison Time Setup Register (ADCT).

$$\text{Comparison time} = \text{Compare clock cycle} \times 14$$

$$\text{Compare clock cycle} = \text{Base clock (HCLK) cycle} \times (\text{CT set value} + 2)$$

<Notes>

- For setting the compare clock cycle, refer to the "Electrical Characteristics" in the data sheet to make sure that an appropriate time should be selected in accordance with an analog power supply voltage (AVCC) and a base clock (HCLK) cycle.
- If the sampling time or compare clock cycle fails to meet the electrical characteristics of the A/D converter, the A/D conversion accuracy may be degraded.

■ Example of conversion time calculation (when HCLK = 80 MHz (12.5 ns cycle))

(1) Sampling time

- When ST04 to 00 = 17 and STX02, STX01, and STX00 = 000 (multiplied by 1)
 $\text{Sampling time} = 12.5 \text{ ns} \times \{(17 + 1) \times 1 + 1\} = \underline{\underline{237.5 \text{ ns}}}$

- When ST14 to 10 = 19 and STX12, STX11, and STX10 = 001 (multiplied by 4)
 $\text{Sampling time} = 12.5 \text{ ns} \times \{(19 + 1) \times 4 + 1\} = \underline{\underline{1012.5 \text{ ns}}}$

(2) Comparison time

- When CT02 to 00 = 3
 $\text{Compare clock cycle} = 12.5 \text{ ns} \times (3 + 2) = \underline{\underline{62.5 \text{ ns}}}$
 $\text{Comparison time} = 62.5 \text{ ns} \times 14 = \underline{\underline{875 \text{ ns}}}$

(3) Conversion time

By adding (1) and (2) together:

- Conversion time for channels specified with the ADST0 register = 1112.5 ns
- Conversion time for channels specified with the ADST1 register = 1887.5 ns

5. Registers

This section explains the configuration and functions of the registers used for the 12-bit A/D converter.

■ List of registers for the 12-bit A/D converter

Abbreviation	Register name	See
ADCR	A/D Control Register	5.1
ADSR	A/D Status Register	5.2
SCCR	Scan Conversion Control Register	5.3
SFNS	Scan Conversion FIFO Stage Count Setup Register	5.4
SCFD	Scan Conversion FIFO Data Register	5.5
SCIS	Scan Conversion Input Selection Register	5.6
PCCR	Priority Conversion Control Register	5.7
PFNS	Priority Conversion FIFO Stage Count Setup Register	5.8
PCFD	Priority Conversion FIFO Data Register	5.9
PCIS	Priority Conversion Input Selection Register	5.10
CMPD	A/D Comparison Value Setup Register	5.11
CMPCR	A/D Comparison Control Register	5.12
ADSS	Sampling Time Selection Register	5.13
ADST	Sampling Time Setup Register	5.14
ADCT	Comparison Time Setup Register	5.15
ADCEN	A/D Operation Enable Setup Register	5.16

5.1. A/D Control Register (ADCR)

The A/D Control Register (ADCR) performs interrupt flag display and interrupt enable control.

bit	15	14	13	12	11	10	9	8
Field	SCIF	PCIF	CMPIF	Reserved	SCIE	PCIE	CMPIE	OVRIE
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	X	0	0	0	0

[bit 15] SCIF: Scan conversion interrupt request bit

When conversion values are written up to the stage count specified in the Scan Conversion FIFO Stage Count Setup Register (SFNS), this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Conversion result is not stored.	Clears this bit.
1	Conversion result is stored.	No effect.

[bit 14] PCIF: Priority conversion interrupt request bit

When conversion values are written up to the stage specified in the Priority Conversion FIFO Stage Count Setup Register (PFNS), this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Conversion result is not stored.	Clears this bit.
1	Conversion result is stored.	No effect.

[bit 13] CMPIF: Conversion result comparison interrupt request bit

When the condition set in the A/D Comparison Value Setup Register (CMPD) or A/D Comparison Control Register (CMPCR) is satisfied during the operation of the A/D conversion result comparison function, this bit is set to "1". The read value of Read-Modify-Write access is "1" regardless of the bit value.

Bit	Description	
	Read	Write
0	Specified condition is not satisfied.	Clears this bit.
1	Specified condition is satisfied.	No effect.

[bit 12] Reserved: Reserved bit

Write	Has no effect on operation.
Read	The value is undefined.

[bit 11] SCIE: Scan conversion interrupt enable bit

This bit controls the interrupt request of SCIF. When the SCIE bit is enabled, and the SCIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 10] PCIE: Priority conversion interrupt enable bit

This bit controls the interrupt request of PCIF. When the PCIE bit is enabled, and the PCIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 9] CMPIE: Conversion result comparison interrupt enable bit

This bit controls the interrupt request of CMPIF. When the CMPIE bit is enabled, and the CMPIF bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

[bit 8] OVRIE: FIFO overrun interrupt enable bit

This bit controls the interrupt request of the SOVR bit in the SCCR register or the POVR bit in the PCCR register. When the OVRIE bit is enabled, and the SOVR or POVR bit is set, an interrupt request to the CPU is generated.

Bit	Description
0	Interrupt request disable
1	Interrupt request enable

5.2. A/D Status Register (ADSR)

The A/D Status Register (ADSR) displays scan and priority conversion statuses.

bit	7	6	5	4	3	2	1	0
Field	ADSTP	FDAS		Reserved		PCNS	PCS	SCS
Attribute	R/W	R/W	-	-	-	R	R	R
Initial value	0	0	X	X	X	0	0	0

[bit 7] ADSTP: A/D conversion forced stop bit

Setting the ADSTP bit to "1" stops the A/D conversion operation forcibly (both scan and priority conversion operations are stopped). Forced stop of A/D conversion initializes the PCNS, PCS, and SCS bits in the ADSR register to "0". However, other register bits are not reset.

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
1		Stops the conversion operation forcibly.

[bit 6] FDAS: FIFO data placement selection bit

Setting the FDAS bit to "1" shifts the Scan Conversion FIFO Data Register (SCFD) and Priority Conversion FIFO Data Register (PCFD) conversion result values by 4 bits to the LSB side, placing them in bit 27 to 16. The position of the FIFO data register of lower 16-bit doesn't change.

Bit	Description
0	Places conversion result on the MSB side.
1	Places conversion result on the LSB side.

[bit 5:3] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 2] PCNS: Priority conversion pending flag

This flag indicates that conversion at priority level 2 (software/timer) is pending. This flag is set when priority conversion at priority level 2 (software/timer) is started while priority conversion at priority level 1 (external trigger start) is performed or when conversion at priority level 1 is started while priority conversion at priority level 2 is performed. Writing is ignored.

Bit	Description
0	Priority level 2 conversion is not pending.
1	Priority level 2 conversion is pending.

[bit 1] PCS: Priority conversion status flag

This flag indicates that priority A/D conversion is in progress. This flag is set while priority conversion at priority level 1 or 2 is performed. Writing is ignored.

Bit	Description
0	Priority conversion is stopped.
1	Priority conversion is in progress.

[bit 0] SCS: Scan conversion status flag

This flag indicates that scan A/D conversion is in progress. Writing is ignored.

Bit	Description
0	Scan conversion is stopped.
1	Scan conversion is in progress.

5.3. Scan Conversion Control Register (SCCR)

The Scan Conversion Control Register (SCCR) controls the scan conversion mode.

bit	15	14	13	12	11	10	9	8
Field	SEMP	SFUL	SOVR	SFCLR	Reserved	RPT	SHEN	SSTR
Attribute	R	R	R/W	R/W	-	R/W	R/W	R/W
Initial value	1	0	0	0	X	0	0	0

[bit 15] SEMP: Scan conversion FIFO empty bit

This bit is set when FIFO goes to the empty state. When conversion data is written in the Scan Conversion FIFO Data Register (SCFD), this bit is set to "0". Writing is ignored.

Bit	Description
0	Data remains in FIFO.
1	FIFO is empty.

[bit 14] SFUL: Scan conversion FIFO full bit

This bit is set when FIFO goes to full state. When SFCLR is set to "1" or the Scan Conversion FIFO Data Register (SCFD) is read, this bit is set to "0". Writing is ignored.

Bit	Description
0	Data can be input to FIFO.
1	FIFO is full.

[bit 13] SOVR: Scan conversion overrun flag

This bit is set when an attempt to write data to a full FIFO is made (conversion data in a full FIFO is not overwritten). The read value of Read-Modify-Write access is "1" regardless of the bit value. When the OVRIE bit in the ADCR register is "1", an interrupt is generated to the CPU if the SOVR bit is "1".

Bit	Description	
	Read	Write
0	No overrun has occurred.	Clears this bit.
1	Overrun has occurred.	No effect.

[bit 12] SFCLR: Scan conversion FIFO clear bit

Setting this bit to "1" clears the scan conversion FIFO. FIFO becomes empty and the SEMP bit is set to "1".

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
1		Clears FIFO.

[bit 11] Reserved: Reserved bit

Write	Has no effect on operation.
Read	The value is undefined.

[bit 10] RPT: Scan conversion repeat bit

Setting this bit to "1" places the converter in the repeat mode. When the conversion of all analog input channels selected in the Scan Conversion Input Selection Register (SCIS) is completed, the conversion is started again.

Setting the RPT bit to "0" ends the repeat conversion. The operation stops when the conversion of the analog input channels selected in the SCIS bit is completed.

Setting the RPT bit to "1" must be performed while scan conversion is stopped (the SCS bit in the ADSR register = "0"). (Setting the SSTR bit to "1" may be performed simultaneously with setting the RPT bit to "1".)

Bit	Description
0	Single conversion mode
1	Repeat conversion mode

[bit 9] SHEN: Scan conversion timer start enable bit

Set this bit to "1" to start scan conversion using a rising edge from a timer. Software startup (SSTR = 1) is valid even when this bit is set to "1".

Bit	Description
0	Timer start disable
1	Timer start enable

[bit 8] SSTR: Scan conversion start bit

Setting this bit to "1" starts A/D conversion. Setting this bit to "1" again during conversion stops the ongoing conversion immediately and restarts the conversion.

Bit	Description	
	Read	Write
0		No effect.
1	The value is always "0".	Starts conversion or restarts the conversion (during conversion).

<Note>

If a startup by a timer occurs simultaneously with the setting of the SSTR bit to "1", the setting of the SSTR bit to "1" takes preference and the startup by the timer is ignored.

5.4. Scan Conversion FIFO Stage Count Setup Register (SFNS)

The Scan Conversion FIFO Stage Count Setup Register (SFNS) sets up the generation of interrupt requests in scan conversion. When the specified count of FIFO stages store A/D conversion data, the interrupt request bit (SCIF) is set.

bit	7	6	5	4	3	2	1	0
Field	Reserved				SFS [3:0]			
Attribute	-	-	-	-	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

[bit 7:4] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 3:0] SFS [3:0]: Scan conversion FIFO stage count setting bit

When A/D conversion data for the FIFO stage count ($N + 1$) set in SFS [3:0] is written, the interrupt request flag (SCIF) is set to "1".

Bit [3:0]	Description
0b0000	Generates an interrupt request when conversion result is stored in the first FIFO stage.
0b0001	Generates an interrupt request when conversion result is stored in the second FIFO stage.
0b0010	Generates an interrupt request when conversion result is stored in the third FIFO stage.
...	...
0b1101	Generates an interrupt request when conversion result is stored in the 14th FIFO stage.
0b1110	Generates an interrupt request when conversion result is stored in the 15th FIFO stage.
0b1111	Generates an interrupt request when conversion result is stored in the 16th FIFO stage.

5.5. Scan Conversion FIFO Data Register (SCFD)

The Scan Conversion FIFO Data Register (SCFD) consists of 16 FIFO stages and stores analog conversion results. Data can be retrieved sequentially by reading the register.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0				Reserved
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field		Reserved		INVL	Reserved	RS1	RS0		Reserved		SC4	SC3	SC2	SC1	SC0	
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit 31:20] SD11:SD0: Scan conversion result

The result of 12-bit scan A/D conversion is written.

[bit 19:13] Reserved: Reserved bits

The read value is undefined.

[bit 12] INVL : A/D conversion result disable bit

This bit is set when this register value is invalid.

Bit	Description
0	This register value is valid
1	This register value is invalid

[bit 11:10] Reserved: Reserved bits

The read value is undefined.

[bit 9:8] RS1:RS0 : Scan conversion start factor

The start factor of the scan conversion corresponding to this register value is shown.

Bit [9:8]	Description
0b01	Software start
0b10	Timer start

[bit 7:5] Reserved: Reserved bits

The read value is undefined.

[bit 4:0] SC4:SC0: Conversion input channel bits

The analog input channels corresponding to the conversion result written in SD11 to SD0 are written. Settings for channels not defined in the product specifications are not written. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

<Note>

This register has different bit configurations depending on the FDAS bit setting in the A/D Status Register (ADSR). When the FDAS bit is "1", see "3.3.6 Bit placement selection for FIFO data registers".

To perform a byte access to this register, read the most significant byte (bit 31:24) to shift the FIFO data. Reading the other bytes (bit 23:16, bit 15:8, bit 7:0) does not shift FIFO. To perform a half byte access to this register, read the most significant half byte (bit 31:16) to shift the FIFO data. Reading the other byte (bit 15:0) does not shift FIFO. Performing a word access to this register shifts FIFO.

If software and a timer are started simultaneously, "0b11" may be read from the RS1:RS0 bit.

5.6. Scan Conversion Input Selection Register (SCIS)

The Scan Conversion Input Selection Register (SCIS) is used to select analog input channels for which scan conversion is performed. Any channels can be selected from multiple analog inputs. The selected channels are converted in ascending order of channel number.

■ SCIS3 (most significant byte: AN31 to AN24) and SCIS2 (least significant byte: AN23 to AN16)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	AN31	AN30	AN29	AN28	AN27	AN26	AN25	AN24	AN23	AN22	AN21	AN20	AN19	AN18	AN17	AN16
Attribute	R/W															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] AN31:AN16: Analog input selection bits

When these bits are set to "1", the corresponding channels are selected for analog conversion.

■ SCIS1 (most significant byte: AN15 to AN8) and SCIS0 (least significant byte: AN7 to AN0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] AN15:AN0: Analog input selection bits

When these bits are set to "1", the corresponding channels are selected for analog conversion.

<Note>

It is not possible to change the channels during A/D conversion. Be sure to set SCIS3 to SCIS0 while the A/D conversion is stopped.

It is not possible to set "1" in the bit corresponding to a channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

■ Example of scan conversion order

The selected channels are converted in ascending order of channel number.

Example: When the AN1, AN3, AN5, and AN23 bits are set to "1", the analog conversion proceeds from ch.1, ch.3, ch.5, and to ch.23.

5.7. Priority Conversion Control Register (PCCR)

The Priority Conversion Control Register (PCCR) controls the priority conversion mode.

Priority conversion can be performed even while scan conversion is being performed.

In addition, different priority levels (two levels) can be given to priority conversion processes.

bit	15	14	13	12	11	10	9	8
Field	PEMP	PFUL	POVR	PFCLR	ESCE	PEEN	PHEN	PSTR
Attribute	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	0	0	0	0

[bit 15] PEMP: Priority conversion FIFO empty bit

This bit is set when FIFO goes to the empty state. When conversion data is written in the Priority Conversion FIFO Data Register (PCFD), this bit is set to "0". Writing is ignored.

Bit	Description
0	Data remains in FIFO.
1	FIFO is empty.

[bit 14] PFUL: Priority conversion FIFO full bit

This bit is set when FIFO goes to full state. When PFCLR is set to "1" or the Priority Conversion FIFO Data Register (PCFD) is read, this bit is set to "0". Writing is ignored.

Bit	Description
0	Data can be input to FIFO.
1	FIFO is full.

[bit 13] POVR: Priority conversion overrun flag

This bit is set when an attempt to write data to a full FIFO is made (conversion data in a full FIFO is not overwritten). The read value of Read-Modify-Write access is "1" regardless of the bit value. When the OVRIE bit in the ADCR register is "1", an interrupt is generated to the CPU if the POVR bit is "1".

Bit	Description	
	Read	Write
0	No overrun has occurred.	Clears this bit.
1	Overrun has occurred.	No effect.

[bit 12] PFCLR: Priority conversion FIFO clear bit

Setting this bit to "1" clears the priority conversion FIFO. FIFO becomes empty and the PEMP bit is set to "1".

Bit	Description	
	Read	Write
0		No effect.
1	The value is always "0".	Clears FIFO.

[bit 11] ESCE: External trigger analog input selection bit

This bit selects whether the external trigger analog input is selected with the P1A [2:0] bits in the Priority Conversion Input Selection Register (PCIS) or the external input pin ECS [2:0] bits.

Bit	Description
0	The external trigger analog inputs are selected with P1A [2:0].
1	The external trigger analog inputs are selected with an external input.

<Note>

It is not possible to change the setting of the ESCE bit during A/D conversion. To change the setting, make sure the A/D conversion is stopped.

If channel selection with external pins ECS [2:0] cannot be used due to the product specifications, be sure to set the ESCE bit to "0".

[bit 10] PEEN: Priority conversion external start enable bit

Set this bit to "1" to start priority conversion using a falling edge of an external trigger pin input. Conversion started with an external trigger has priority level 1 (highest priority).

Bit	Description
0	External trigger start disable
1	External trigger start enable

[bit 9] PHEN: Priority conversion timer start enable bit

Set this bit to "1" to start priority conversion using a rising edge from a timer. Software startup (PSTR = 1) is valid even when this bit is set to "1". Conversion started with an external trigger has priority level 2 (lower than level 1).

Bit	Description
0	Timer start disable
1	Timer start enable

[bit 8] PSTR: Priority conversion start bit

Setting this bit to "1" starts A/D conversion. Conversion started with this bit has priority level 2 (lower than level 1). It is not possible to restart the conversion started with this bit.

Bit	Description	
	Read	Write
0	The value is always "0".	No effect.
		Starts priority conversion.

5.8. Priority Conversion FIFO Stage Count Setup Register (PFNS)

The Priority Conversion FIFO Stage Count Setup Register (PFNS) sets up the generation of interrupt requests in priority conversion. When the specified count of FIFO stages store A/D conversion data, the interrupt request bit (PCIF) is set.

bit	7	6	5	4	3	2	1	0
Field	Reserved		TEST [1:0]		Reserved		PFS [1:0]	
Attribute	-	-	R	R	-	-	R/W	R/W
Initial value	X	X	X	X	X	X	0	0

[bit 7:6] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 5:4] TEST [1:0]: Test bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 3:2] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 1:0] PFS [1:0]: Priority conversion FIFO stage count setting bits

When A/D conversion data for the FIFO stage count ($N + 1$) set in PFS [1:0] is written, the interrupt request flag (PCIF) is set to "1".

Bit [1:0]	Description
0b00	Generates an interrupt request when conversion result is stored in the first FIFO stage.
0b01	Generates an interrupt request when conversion result is stored in the second FIFO stage.
0b10	Generates an interrupt request when conversion result is stored in the third FIFO stage.
0b11	Generates an interrupt request when conversion result is stored in the fourth FIFO stage.

5.9. Priority Conversion FIFO Data Register (PCFD)

The Priority Conversion FIFO Data Register (PCFD) consists of four FIFO stages and stores analog conversion results. Data can be retrieved sequentially by reading the register.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	Reserved			
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved			INVL	Reser ved	RS2	RS1	RS0	Reserved			PC4	PC3	PC2	PC1	PC0
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit 31:20] PD11:PD0: Priority conversion result

The result of 12-bit priority A/D conversion is written.

[bit 19:13] Reserved: Reserved bits

The read value is undefined.

[bit 12] INVL : A/D conversion result disable bit

This bit is set when this register value is invalid.

Bit	Description
0	This register value is valid
1	This register value is invalid

[bit 11] Reserved: Reserved bit

The read value is undefined.

[bit 10:8] RS2:RS0 : Scan conversion start factor

The start factor of the priority conversion corresponding to this register value is shown.

Bit [10:8]	Description
0b001	Software start (priority level 2)
0b010	timer start (priority level 2)
0b100	External trigger (priority level 1)

[bit 7:5] Reserved: Reserved bits
The read value is undefined.

[bit 4:0] PC4:PC0: Conversion input channel bits

The analog input channels corresponding to the conversion result written in PD11 to PD0 are written. Settings for channels not defined in the product specifications are not written. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

<Note>

This register has different bit configurations depending on the FDAS bit setting in the A/D Status Register (ADSR). When the FDAS bit is "1", see "3.3.6 Bit placement selection for FIFO data registers".

To perform a byte access to this register, read the most significant byte (bit 31:24) to shift the FIFO data. Reading the other bytes (bit 23:16, bit 15:8, bit 7:0) does not shift FIFO. To perform a half word access to this register, read the most significant half word (bit 31:16) to shift FIFO. Reading the other byte (bit 15:0) does not shift FIFO. Performing a word access to this register shifts FIFO.

If software and a timer are started simultaneously, "0b011" may be read from the RS2:RS0 bit.

Conversion started with an external trigger can be performed only when the analog input channel is between ch.0 to 7.

5.10. Priority Conversion Input Selection Register (PCIS)

The Priority Conversion Input Selection Register (PCIS) is used to select the analog input channels for which priority conversion is performed. For software or timer start at priority level 2, only one channel can be selected from multiple analog input channels. For external trigger start at priority level 1, one channel can be selected from eight channels (ch.0 to ch.7).

bit	7	6	5	4	3	2	1	0
Field	P2A [4:0]					P1A [2:0]		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7:3] P2A [4:0]: Priority level 2 analog input selection

This bit specifies the analog input channel for a start at priority level 2 (software/timer). It can be selected from all channels. It is not possible to set the channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [7:3]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

[bit 2:0] P1A [2:0]: Priority level 1 analog input selection

This bit specifies the analog input channel for a start at priority level 1 (external trigger). It can be selected from eight channels (ch.0 to ch.7).

Bit [2:0]	Description
0b000	ch.0
0b001	ch.1
0b010	ch.2
...	...
0b101	ch.5
0b110	ch.6
0b111	ch.7

5.11. A/D Comparison Value Setup Register (CMRD)

The A/D Comparison Value Setup Register (CMRD) sets the value to be compared with the A/D conversion result. When the conditions set in both this register and the A/D Comparison Control Register (CMPCR) are satisfied, the conversion result comparison interrupt request bit (CMPIF) in the A/D Control Register (ADCR) is set.

bit	31	30	29	28	27	26	25	24
Field	CMAD11	CMAD10	CMAD9	CMAD8	CMAD7	CMAD6	CMAD5	CMAD4
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
bit	23	22	21	20	19	18	17	16
Field	CMAD3	CMAD2	Reserved					
Attribute	R/W	R/W	-	-	-	-	-	-
Initial value	0	0	X	X	X	X	X	X

[bit 31:22] CMAD11:CMAD2: A/D conversion result value setting bits

These bits set the value to be compared with the A/D conversion result.

The most significant 10 bits (bit 11:2) of the A/D conversion result are compared with the value in this register (CMAD11 to CMAD2). The least significant two bits (bit 1 and 0) of the A/D conversion result are not compared.

[bit 21:16] Reserved: Reserved bits

The read value is undefined.

5.12. A/D Comparison Control Register (CMPCR)

The A/D Comparison Control Register (CMPCR) controls the A/D comparison function. When the converted value is compared with the value in the A/D Comparison Value Setup Register (CMPD) and the comparison condition in this register is satisfied, the conversion result comparison interrupt request bit (CMPIF) in the A/D Control Register (ADCR) is set.

bit	7	6	5	4	3	2	1	0
Field	CMPEN	CMD1	CMD0		CCH [4:0]			
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] CMPEN: Conversion result comparison function operation enable bit

This bit enables the operation of the A/D comparison function.

Bit	Description
0	Stops the comparison function operation.
1	Enables the comparison function operation.

[bit 6] CMD1: Comparison mode 1

This bit sets the condition for generating a conversion interrupt request.

Bit	Description
0	Generates an interrupt request when the most significant 10 bits (bit 11:2) of the A/D conversion result is smaller than the CMPD set value.
1	Generates an interrupt request when the most significant 10 bits (bit 11:2) of the A/D conversion result is equal to or greater than the CMPD set value.

[bit 5] CMD0: Comparison mode 0

This bit selects the comparison target. When this bit is "1", the setting of CCH [4:0] is invalid.

Bit	Description
0	Compares the conversion result of the channel set in CCH [4:0].
1	Compares the conversion results of all channels.

[bit 4:0] CCH [4:0]: Comparison target analog input channel

This bit sets the analog channel to be compared. When the CMD0 bit is "1", setting of this bit is invalid. It is not possible to set the channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

Bit [4:0]	Description
0b00000	ch.0
0b00001	ch.1
0b00010	ch.2
...	...
0b11101	ch.29
0b11110	ch.30
0b11111	ch.31

5.13. Sampling Time Selection Register (ADSS)

The Sampling Time Selection Register (ADSS3 to 0) allows you to set the sampling time for each bit. Which of the sampling times set in Sampling Time Setup Registers 0 and 1 (ADST0 and 1) is used is specified in this register.

■ ADSS3 (most significant byte: TS31 to TS24) and ADSS2 (least significant byte: TS23 to TS16)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Attribute	R/W															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] TS31:TS16: Sampling time selection bits

Set the sampling time specified in the Sampling Time Setup Register (ADST) for the corresponding channel.

Setting "0" specifies the time set in ADST0 and setting "1" specifies the time set in ADST1. TS31 to TS16 correspond respectively to ch.31 to ch.16.

■ ADSS1 (most significant byte: TS15 to TS8) and ADSS0 (least significant byte: TS7 to TS0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit 15:0] TS15:TS0: Sampling time selection bits

Set the sampling time specified in the Sampling Time Setup Register (ADST) for the corresponding channel.

Setting "0" specifies the time set in ADST0 and setting "1" specifies the time set in ADST1. TS15 to TS0 correspond respectively to ch.15 to ch.0.

<Note>

It is not possible to write to the ADSS register during A/D conversion.

It is not possible to set "1" in the bit corresponding to a channel that is not defined in the product specifications. See the specified number of the analog input channels in the "Data Sheet" of each product.

5.14. Sampling Time Setup Register (ADST)

Sampling Time Setup Registers 0 and 1 (ADST0 and 1) set the sampling times for A/D conversion. ADST0 and 1 are provided for setting two sampling times, and which one is used is selected in the Sampling Time Selection Register (ADSS3 to 0).

■ ADST0 (most significant byte)

bit	15	14	13	12	11	10	9	8
Field	STX02	STX01	STX00	ST04	ST03	ST02	ST01	ST00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	0	0	0	0

[bit 15:13] STX02:STX00: Sampling time N times setting bits

These bits multiply the sampling time set values in the ST04 to ST00 bits by N.

Bit 15	Bit 14	Bit 13	Description
0	0	0	Set value x 1
0	0	1	Set value x 4
0	1	0	Set value x 8
0	1	1	Set value x 16
1	0	0	Set value x 32
1	0	1	Set value x 64
1	1	0	Set value x 128
1	1	1	Set value x 256

[bit 12:8] ST04:ST00: Sampling time setting bits

These bit set the sampling time for A/D conversion.

Sampling time = HCLK cycle $\times \{(ST \text{ set value} + 1) \times STX \text{ setting multiplier} + 1\}$

Example: When ST04 to ST00 = 9, STX02, STX01, and STX00 = 001 (multiplied by 4), and

HCLK = 80 MHz (12.5 ns),

Sampling time = 12.5 ns $\times \{(9 + 1) \times 4 + 1\} = 512.5$ ns

<Note>

It is not possible to write to the ADST0 register during A/D conversion.

When STX02, STX01, and STX00 = 000 (ST04 to ST00 set values multiplied by 1) are set, set ST04 to ST00 to "3" or more ("2" or less must not be set).

For setting the sampling time, refer to the "Electrical Characteristics" in the data sheet to make sure that an appropriate time should be selected in accordance with an external impedance of an input channel, an analog power supply voltage (AVCC), and a base clock (HCLK) cycle.

■ ADST1 (least significant byte)

bit	7	6	5	4	3	2	1	0
Field	STX12	STX11	STX10	ST14	ST13	ST12	ST11	ST10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	0	0	0	0

[bit 7:5] STX12:STX10: Sampling time N times setting bits

These bits multiply the sampling time set values in the ST14 to ST10 bits by N.

Bit 7	Bit 6	Bit 5	Description
0	0	0	Set value x 1
0	0	1	Set value x 4
0	1	0	Set value x 8
0	1	1	Set value x 16
1	0	0	Set value x 32
1	0	1	Set value x 64
1	1	0	Set value x 128
1	1	1	Set value x 256

[bit 4:0] ST14:ST10: Sampling time setting bits

These bit set the sampling time for A/D conversion.

Sampling time = HCLK cycle × {(ST set value + 1) × STX setting multiplier + 1}

Example: When ST14 to ST10 = 9, STX12, STX11, and STX10 = 001 (multiplied by 4), and
HCLK = 80 MHz (12.5 ns),
Sampling time = $12.5 \text{ ns} \times \{(9 + 1) \times 4 + 1\} = 512.5 \text{ ns}$

<Note>

It is not possible to write to the ADST1 register during A/D conversion.

When STX12, STX11, and STX10 = 000 (ST14 to ST10 set values multiplied by 1) are set, set ST14 to ST10 to "3" or more ("2" or less must not be set).

For setting the sampling time, refer to the "Electrical Characteristics" in the data sheet to make sure that an appropriate time should be selected in accordance with an external impedance of an input channel, an analog power supply voltage (AVCC), and a base clock (HCLK) cycle.

5.15. Comparison Time Setup Register (ADCT)

The Comparison Time Setup Register (ADCT) sets the comparison time, which is part of the A/D conversion time.

bit	7	6	5	4	3	2	1	0
Field	Reserved					CT2	CT1	CT0
Attribute	-	-	-	-	-	R/W	R/W	R/W
Initial value	X	X	X	X	X	1	1	1

[bit 7:3] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit 2:0] CT2:CT0: Compare clock frequency division ratio setting bits

These bits set the division ratio of the HCLK for generating the compare clock of A/D conversion.

The frequency division ratio setting is common to Sampling Setup Registers 0 and 1.

Bit 2	Bit 1	Bit 0	Description
0	0	0	Frequency division ratio 2
0	0	1	Frequency division ratio 3
0	1	0	Frequency division ratio 4
0	1	1	Frequency division ratio 5
1	0	0	Frequency division ratio 6
1	0	1	Frequency division ratio 7
1	1	0	Frequency division ratio 8
1	1	1	Frequency division ratio 9

Frequency division ratio = CT set value + 2

Compare clock cycle = Base clock (HCLK) cycle × Frequency division ratio

Comparison time = Compare clock cycle × 14

Example: When the CT set value = 3 and HCLK = 80 MHz (12.5 ns),

Frequency division ratio = $3 + 2 = 5$

Compare clock cycle = $12.5 \text{ ns} \times 5 = 62.5 \text{ ns}$

Comparison time = $62.5 \text{ ns} \times 14 = 875 \text{ ns}$

<Note>

It is not possible to write to the ADCT register during the period of operation enable state transitions and A/D conversion.

For setting the compare clock cycle, refer to the "Electrical Characteristics" in the data sheet to make sure that an appropriate time should be selected in accordance with an analog power supply voltage (AVCC) and a base clock (HCLK) cycle.

5.16. A/D Operation Enable Setup Register (ADCEN)

The A/D Operation Enable Setup Register (ADCEN) is used to turn the 12-bit A/D converter to the operation enable state.

bit	7	6	5	4	3	2	1	0
Field	Reserved						READY	ENBL
Attribute	-	-	-	-	-	-	R	R/W
Initial value	X	X	X	X	X	X	0	0

[bit 7:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1] READY : A/D operation enable state bit

This bit indicates whether the A/D converter is in the operation enable state or not.

A/D conversion can be performed only in the operation enable state.

An A/D conversion request in the operation stop state is ignored.

If the A/D converter enters the operation stop state during A/D conversion, A/D conversion stops immediately.

Bit	Description
0	Operation stop state
1	Operation enable state

[bit0] ENBL : A/D operation enable bit

This bit enables the operation of the A/D converter.

Writing "1" to the ENBL bit turns the A/D converter to the operation enable state after the period of operation enable state transitions. On the other hand, writing "0" to this bit turns the A/D converter to the operation stop state.

Bit	Description
0	Stops operation
1	Enables operation

The cycle number of the base clock (HCLK) necessary as the period of operation enable state transitions depends on the CT[2:0] set value of the ADCT register. The cycle number for the set value is as follows:

ADCT.CT[2:0]	Description
0b000	72 cycles
0b001	108 cycles
0b010	144 cycles
0b011	180 cycles
0b100	216 cycles
0b101	252 cycles
0b110	288 cycles
0b111	324 cycles

Period of operation enable state transitions = Base clock (HCLK) cycle × cycle number

例) When the CT set value = 3 and HCLK = 80 MHz (12.5 ns),

$$\text{Period of operation enable state transitions} = 12.5 \times 180 = 2250 \text{ ns}$$

<Note>

It is not possible to write to the ADCT register during the period of operation enable state transitions and A/D conversion.

Set the ADCEN after setting the ADCT. Set the ADCT so that it may satisfy the period of operation enable state transitions of "Electrical Characteristics" in the data sheet.

When setting the CPU to the timer mode or the stop mode, set the ENBL bit to "0" and turn the A/D converter to the operation stop state.

Chapter: A/D Timer Trigger Selection

This chapter explains the functions and operations to select a timer trigger of the A/D converter.

-
- 1. Overview
 - 2. Registers

1. Overview

This section explains the operations to select a timer trigger of the A/D converter.

■ Selecting a timer trigger of the A/D converter

The 10-bit A/D converter and the 12-bit A/D converter are started by the factors shown in Table 1-1.

Table 1-1 A/D converter start factor

Conversion type	Start factor
Priority level 1 conversion	<ul style="list-style-type: none">Input from an external trigger pin (at falling edge)
Priority level 2 conversion	<ul style="list-style-type: none">Software (when the PCCR:PSTR bit is set to "1")Trigger input from timer (at rising edge)
Scan conversion	<ul style="list-style-type: none">Software (when the SCCR:SSTR bit is set to "1")Trigger input from timer (at rising edge)

The A/D converter can be started with two types of timers: base timer and multifunction timer.

A timer start factor can be selected using the Scan Conversion Timer Trigger Selection Register (SCTSL) or Priority Conversion Timer Trigger Selection Register (PRTSL). The A/D converter starts A/D conversion if a rising edge of the selected timer is detected while timer starting is enabled.

For details on the operations of the 10-bit A/D converter, see the explanation of operations in the "10-bit A/D Converter".

For details on the operations of the 12-bit A/D converter, see the explanation of operations in the "12-bit A/D Converter".

2. Registers

This section explains the configuration and functions of the registers used to select an A/D timer trigger.

■ List of timer trigger selection registers for A/D converter

Abbreviation	Register name	See
SCTSL	Scan Conversion Timer Trigger Selection Register	2.1
PRTSL	Priority Conversion Timer Trigger Selection Register	2.2

The functions of the timer trigger selection registers for the A/D converter are common to the 10-bit A/D converter and the 12-bit A/D converter. For the other registers of the A/D converter, see "10-bit A/D Converter" and "12-bit A/D Converter".

2.1. Scan Conversion Timer Trigger Selection Register (SCTSL)

The Scan Conversion Timer Trigger Selection Register (SCTSL) is used to select a timer trigger when performing scan conversion.

bit	15	14	13	12	11	10	9	8
Field	Reserved				SCTSL[3:0]			
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

[bit 15:12] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 11:8] SCTSL: Scan conversion timer trigger selection bit

Bit [11:8]	Description
0b0000	No selected trigger (Input is fixed to "0".)
0b0001	Starts scan conversion with the multifunction timer.
0b0010	Base timer ch.0
0b0011	Base timer ch.1
0b0100	Base timer ch.2
0b0101	Base timer ch.3
0b0110	Base timer ch.4
0b0111	Base timer ch.5
0b1000	Base timer ch.6
0b1001	Base timer ch.7
0b1010 to 0b1111	Setting disabled

2.2. Priority Conversion Timer Trigger Selection Register (PRTSL)

The Priority Conversion Timer Trigger Selection Register (PRTSL) is used to select a timer trigger when performing priority conversion.

bit	7	6	5	4	3	2	1	0
Field	Reserved				PRTSL[3:0]			
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

[bit 7:4] Reserved: Reserved bits

Write	Has no effect on operation.
Read	The value is undefined.

[bit 3:0] PRTSL: Priority conversion timer trigger selection bit

Bit [3:0]	Description
0b0000	No selected trigger (Input is fixed to "0".)
0b0001	Starts priority conversion with the multifunction timer.
0b0010	Base timer ch.0
0b0011	Base timer ch.1
0b0100	Base timer ch.2
0b0101	Base timer ch.3
0b0110	Base timer ch.4
0b0111	Base timer ch.5
0b1000	Base timer ch.6
0b1001	Base timer ch.7
0b1010 to 0b1111	Setting disabled

CHAPTER: Multi-function Serial Interface

This chapter describes the overview of the multi-function serial interface.

1. Overview of the Multi-function Serial Interface

CODE: 9BFMFS-E01.2

1. Overview of the Multi-function Serial Interface

This multi-function serial interface has the following characteristics.

■ Interface Mode

The following interface modes are selectable for the multi-function serial interface depending on the operation mode settings.

- UART0 (Asynchronous normal serial interface)
- UART1 (Asynchronous multi-processor serial interface)
- CSIO (Clock synchronous serial interface) (SPI can be supported)
- LIN(LIN bus interface)
- I²C (I²C bus interface)

<Note>

See explanations of each chapter for details about each interface.

■ Switching the Interface Mode

To communicate through each serial interface, the serial mode register (SMR) shown in Table 1-1 should be used to set the operation mode before starting the communication.

Table 1-1 Switching Interface Mode

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (Asynchronous normal serial interface)
0	0	1	UART1 (Asynchronous multi-processor serial interface)
0	1	0	CSIO (Clock synchronization serial interface) (SPI can be supported)
0	1	1	LIN(LIN bus interface)
1	0	0	I ² C (I ² C bus interface)

<Notes>

- Transmission and reception cannot be guaranteed when the operation mode is switched while one of the serial interfaces is still in use for transmission or reception operation.
- To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) or disable the I²C (ISMK:EN="0") , and switch the operation mode continuously.
- The settings not listed in Table 1-1 are prohibited.

■ Transmission/Reception FIFO

This UART has a 16-byte transmission FIFO and 16-byte reception FIFO. The FIFO steps should be converted to 16 bytes when reading through this text.

■ LIN Sync field Detection: LSYN

If you are to use an ICU in the LIN bus interface mode, use the ICU of the multifunction timer.
For switching an input to an ICU, see the section for Extended Function Pin Setting Register in the chapter "I/O PORT".

CHAPTER: UART (Async Serial Interface)

This chapter explains the UART (async serial interface) function supported in operation mode 0 and 1 of the multifunctional serial interface.

1. Overview of UART (Async Serial Interface)
2. UART Interrupt
3. UART Operation
4. Dedicated Baud Rate Generator
5. Setting Procedure and Program Flow in Operation Mode 0 (Async Normal Mode)
6. Setting Procedure and Program Flow in Operation Mode 1 (Async Multiprocessor Mode)
7. UART (Async Serial Interface) Registers

CODE: FM15U-E05.2

1. Overview of UART (Async Serial Interface)

UART (async serial interface) is a general-purpose serial data communications interface for asynchronous communications with external devices. It supports a bi-directional communications function (normal mode) and a master/slave type communications function (multi-processor mode: both master and slave modes supported). It also has transmit/receive FIFO installed.

■ Functions of UART (Async Serial Interface)

		Function
1	Data	<ul style="list-style-type: none"> Full duplex double buffer (when FIFO is not used) Transmit/receive FIFO (size: max 128×9 bits each)^{*1} (when FIFO is used)
2	Serial input	Run oversampling three times with the bus clock and determine the value of received data based on the majority sampling value.
3	Transfer system	Asynchronous
4	Baud rate	<ul style="list-style-type: none"> Complete with a dedicated baud rate generator (constructed with a 15-bit reload counter) The external clock input can be adjusted with the reload counter.
5	Data length	<ul style="list-style-type: none"> 5-9 bits (in normal mode)/7 or 8 bits (in multiprocessor mode)
6	Signaling system	NRZ (Non Return to Zero), inverted NRZ
7	Start bit detection	<ul style="list-style-type: none"> In synch with the falling edge of the start bit (in the NRZ system) In synch with the rising edge of the start bit (in the inverted NRZ system)
8	Receive error detection	<ul style="list-style-type: none"> Framing error Overrun error Parity error^{*2}
9	Hardware flow control	CTS/RTS-based automatic transmission/reception control
10	Interrupt request	<ul style="list-style-type: none"> Receive interrupt (upon reception completed, framing error, overrun error or parity error^{*2}) Transmit interrupts (transmit data empty, transmit bus idle) Transmit FIFO interrupt (when transmit FIFO is empty) For both transmission and reception, the extended intelligent I/O service (EIOS) and the DMA function are available.
11	Master/slave communications functions (in multiprocessor mode)	One (master)-to-n (slaves) communication is enabled. (Both master and slave systems are supported.)
12	FIFO options	<ul style="list-style-type: none"> Transmit/receive FIFO installed (maximum capacity: 128×9 bits for transmit FIFO, $128 \text{ bytes} \times 9$ bits for receive FIFO)^{*1} Transmit FIFO or receive FIFO can be selected. Transmit data can be resent. Receive FIFO interrupt timing can be changed via software. FIFO resetting is supported independently.

*1: The FIFO capacity size varies from model type to model type.

*2: Parity errors are only generated in normal mode.

2. UART Interrupt

UART generates transmit or receive interrupts. These interrupt requests can be generated if:

- Incoming data is set in the Receive Data Register (RDR) or a data receive error occurs.
- Outgoing data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started.
- The transmit bus is idle (No data transmission occurs).
- Transmit FIFO data is requested.

■ UART Interrupt

Table 2-1 shows the relationships between the UART interrupt control bits and the interrupt causes.

Table 2-1 UART interrupt control bits and interrupt causes

Interrupt type	Interrupt request flag Bit	Flag register	Operation mode		Interrupt cause	Interrupt cause enable bit	Operation to clear interrupt request flag
			0	1			
Reception	RDRF	SSR	○	○	A single-byte reception	SCR:RIE	Reading from the received data register (RDR)
					Reception of a data volume matching the value set for FBYTE.		
					While the FRIIE bit is "1" and the receive FIFO contains valid data, a receive idle state continues for 8 bits or longer period.		Reading from the Received Data Register (RDR) until receive FIFO is emptied
	ORE	SSR	○	○	Overrun error		Setting the reception error flag clear bit (SSR:REC) to "1"
	FRE	SSR	○	○	Framing error		
	PE	SSR	○	x	Parity error		
Transmission	TDRE	SSR	○	○	The Transmit Data Register is empty	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) ^{*1}
	TBI	SSR	○	○	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) ^{*1}
	FDRQ	FCR1	○	○	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.

*1: Set the TIE bit to "1" only after the TDRE bit has been set to "0".

2.1. Receive interrupt and flag set timing

Data reception can be interrupted by a Receive Completion (SSR:RDRF) or a Receive Error Occurrence (SSR:PE, ORE, FRE).

■ Receive interrupt and flag set timing

Upon detection of the first stop bit, received data are stored in the Receive Data Register (RDR). When the data reception is completed (SSR:RDRF=1) or when a data receive error occurs (SSR:PE, ORE, FRE=1), each flag is set. If receive interrupts are enabled (SSR:RIE=1) then, a receive interrupt occurs.

<Note>

If a receive error occurs, data in the Receive Data Register (RDR) becomes invalid.

Figure 2-1 RDRF (Receive Data Register Full) flag bit set timing

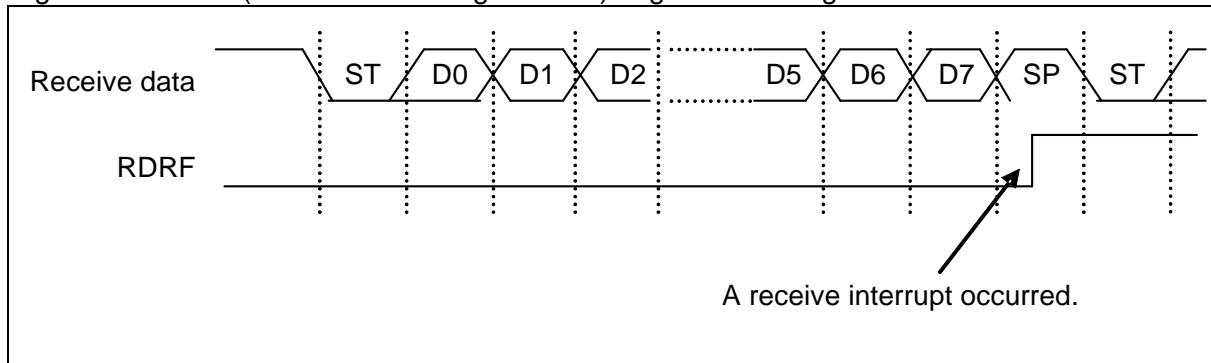
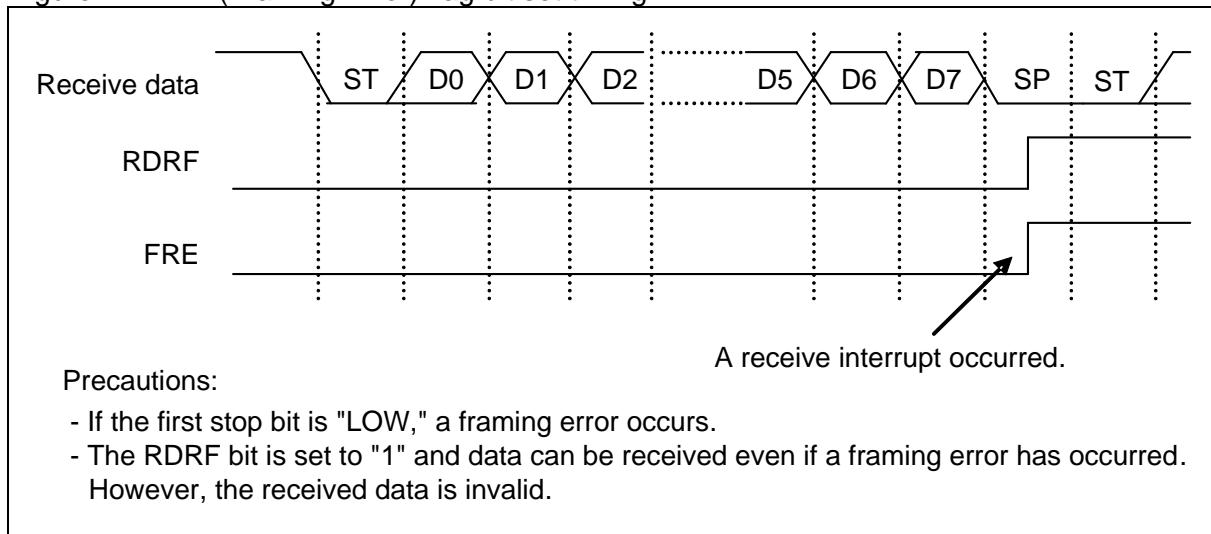


Figure 2-2 FRE (Framing Error) flag bit set timing

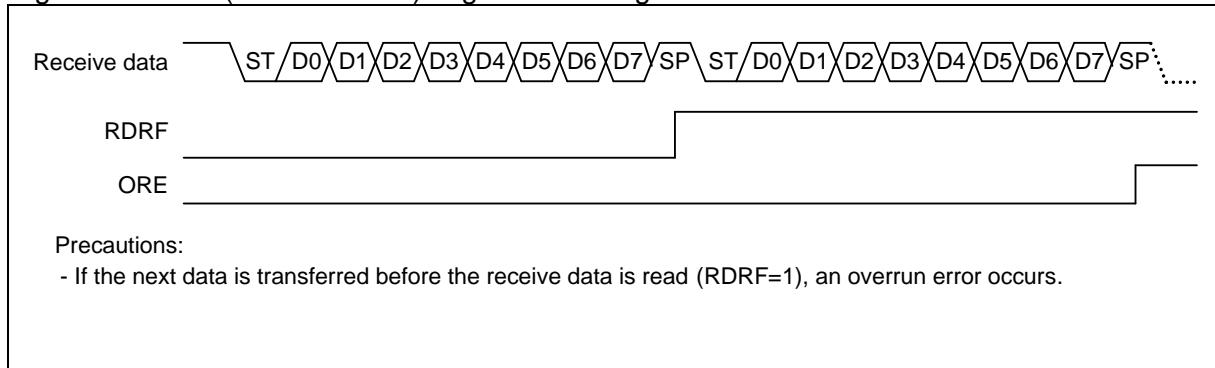


<Note>

During reception, if the following is detected at the same time as the stop bit sampling point or before the 1-2 bus clock, the relevant edge becomes invalid, which may disable normal reception of the next data. To output frames continuously, adequate intervals are required between frames.

- The falling edge of serial data (When ESCR:INV="0")
- The rising edge of serial data (When ESCR:INV="1")

Figure 2-3 ORE (Overrun Error) flag bit set timing



2.2. Interrupt and flag set timing when receive FIFO is used

If the receive FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register) is received.

■ Interrupt and flag set timing when receive FIFO is used

If the receive FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.

- When full FBYTE data is received, the receive data full flag (SSR:RDRF) of the Serial Status register is set to "1". If receive interrupts are enabled (SCR:RIE) during this time, a receive interrupt occurs.
 - If both of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (SSR:RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FCR:FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to zero (0). If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- When data is read from the Receive Data Register (RDR) until receive FIFO is emptied, the receive data full flag (SSR:RDRF) is cleared.
 - If the valid receive data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE=1) occurs.

Figure 2-4 Receive interrupt timing when Receive FIFO is used

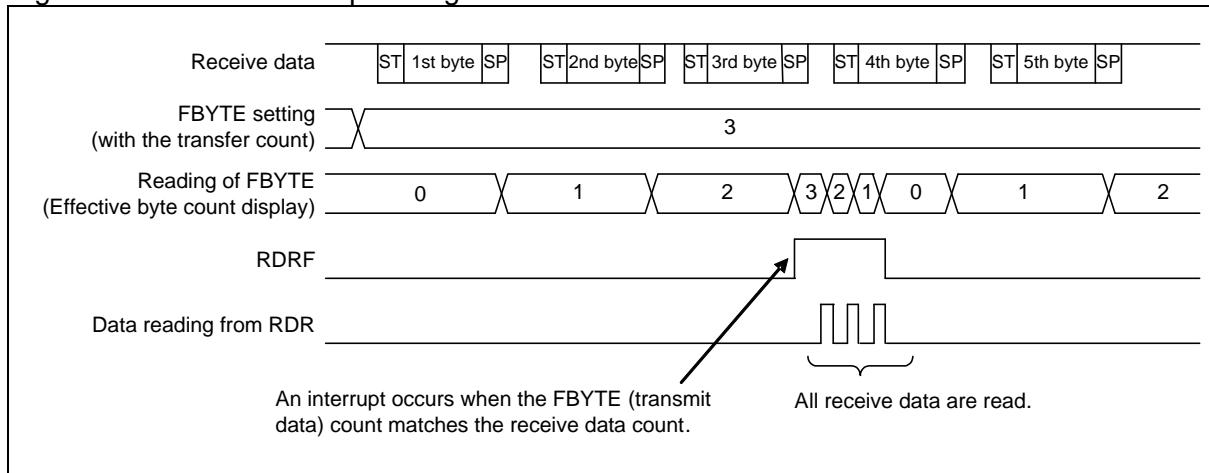
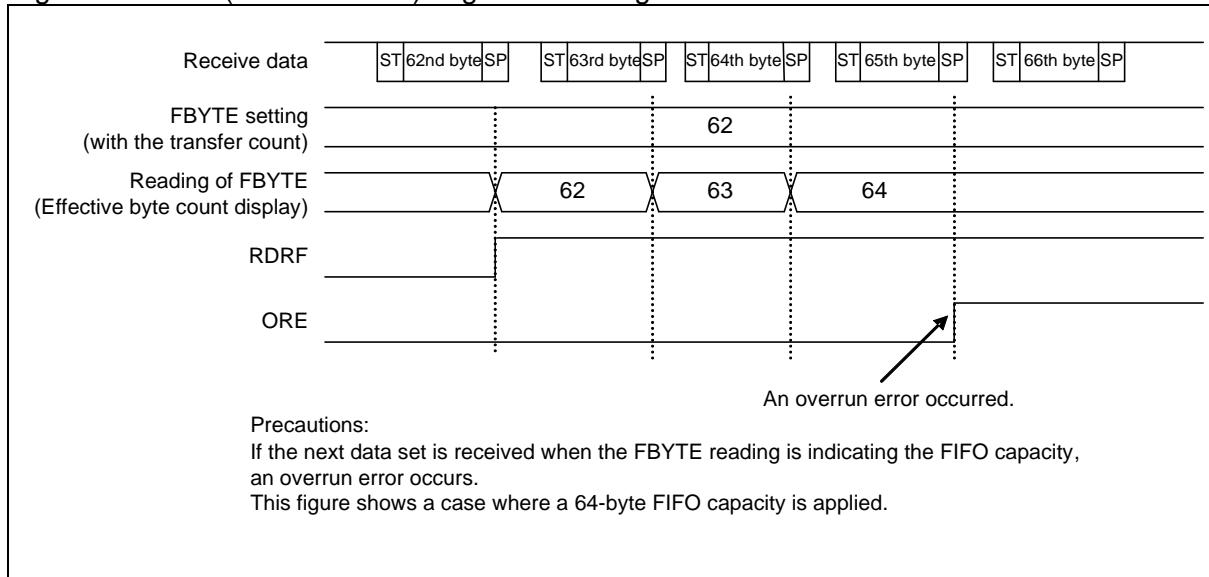


Figure 2-5 ORE (Overrun Error) flag bit set timing



2.3. Transmit interrupt and flag set timing

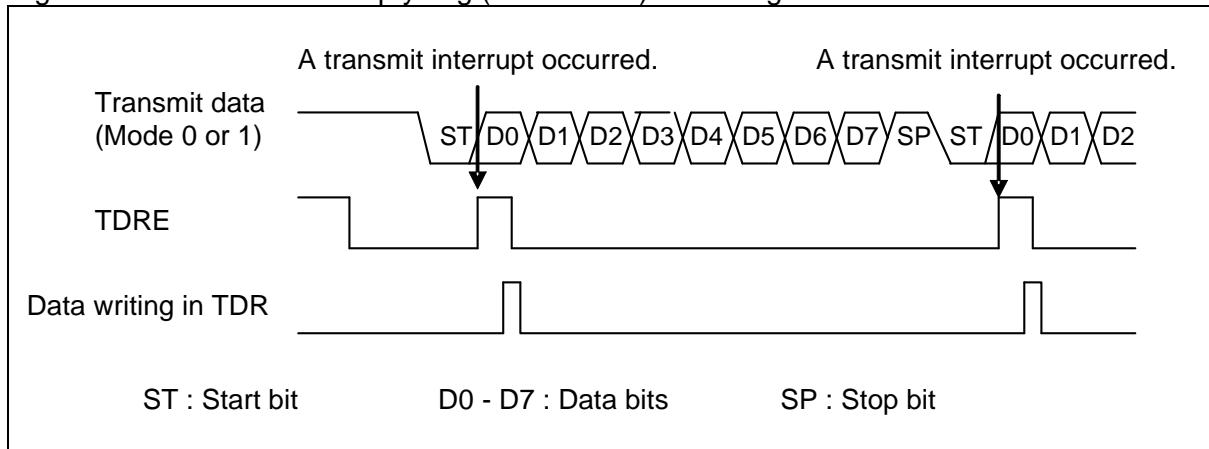
A transmit interrupt occurs when outgoing data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

■ Transmit interrupt and flag set timing

● Transmit data empty flag (SSR:TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written in the TDR (SSR:TDRE = 1). If transmit interrupts are enabled (SCR:TIE = 1) during this time, a transmit interrupt occurs. As the SSR:TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

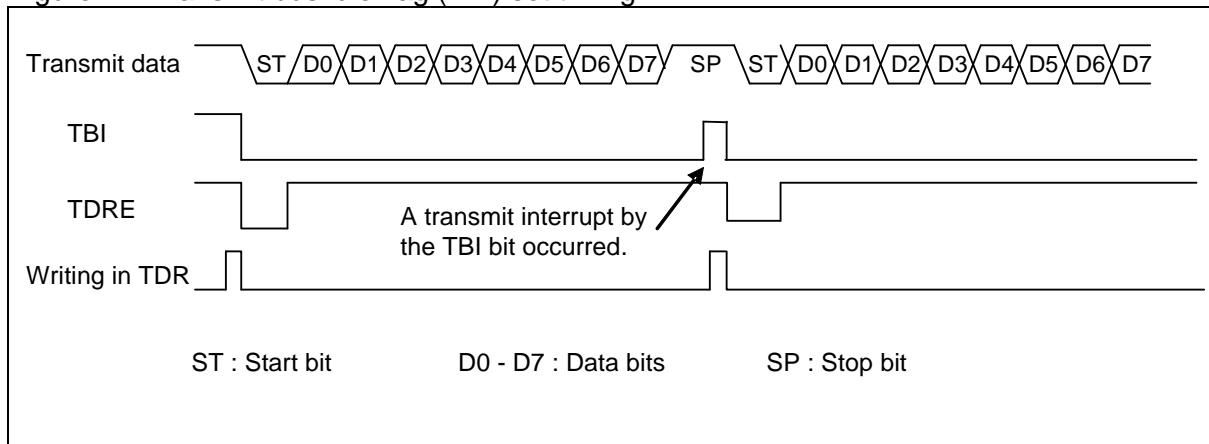
Figure 2-6 Transmit data empty flag (SSR:TDRE) set timing



● Transmit bus idle flag (SSR:TBI) set timing

If the Transmit Data Register is empty (SSR:TDRE=1) and no data is transmitted, the SSR:TBI bit is set to "1". If transmit bus idle interrupts are enabled (SCR:TBIE = 1) during this time, a transmit interrupt occurs. When transmit data is written to the Transmit Data Register (TDR), both the SSR:TBI bit and the transmit interrupt request are cleared.

Figure 2-7 Transmit bus idle flag (TBI) set timing



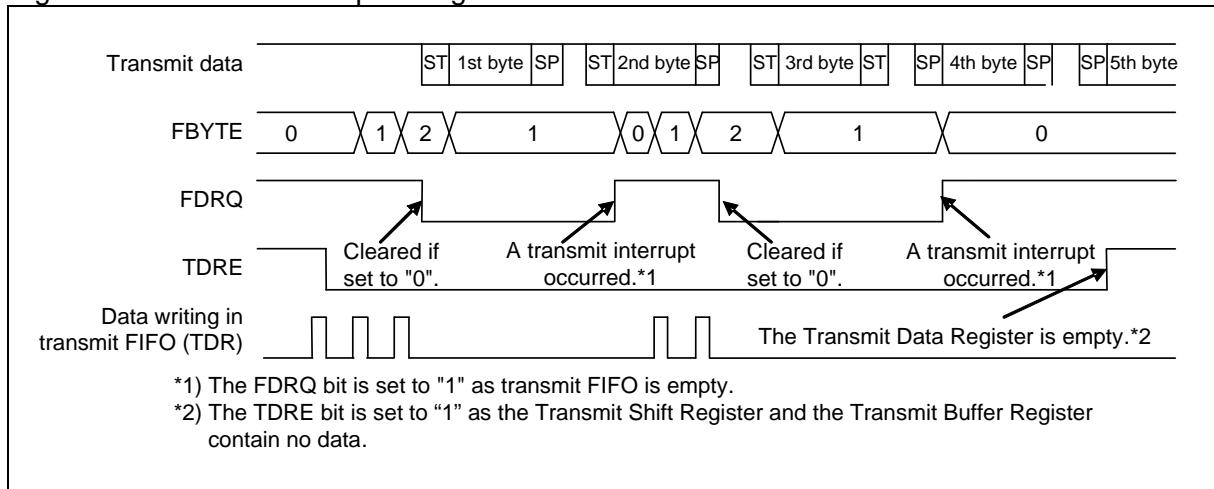
2.4. Interrupt and flag set timing when transmit FIFO is used

When the transmit FIFO is used, an interrupt occurs if the FIFO contains no data.

■ Transmit interrupt and flag set timing when transmit FIFO is used

- If the Transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1". If FIFO transmit interrupts are enabled (FCR1:FTIE="1"), a transmit interrupt occurs.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" when transmit FIFO becomes full.
- To check to see if transmit FIFO contains any data, read from the FIFO Byte Register (FBYTE). If FBYTE=0x00, no data exists in the transmit FIFO.

Figure 2-8 Transmit interrupt timing when transmit FIFO is used



3. UART Operation

UART operates in bi-directional serial asynchronous communications in mode 0 and master/slave multiprocessor communications in mode 1.

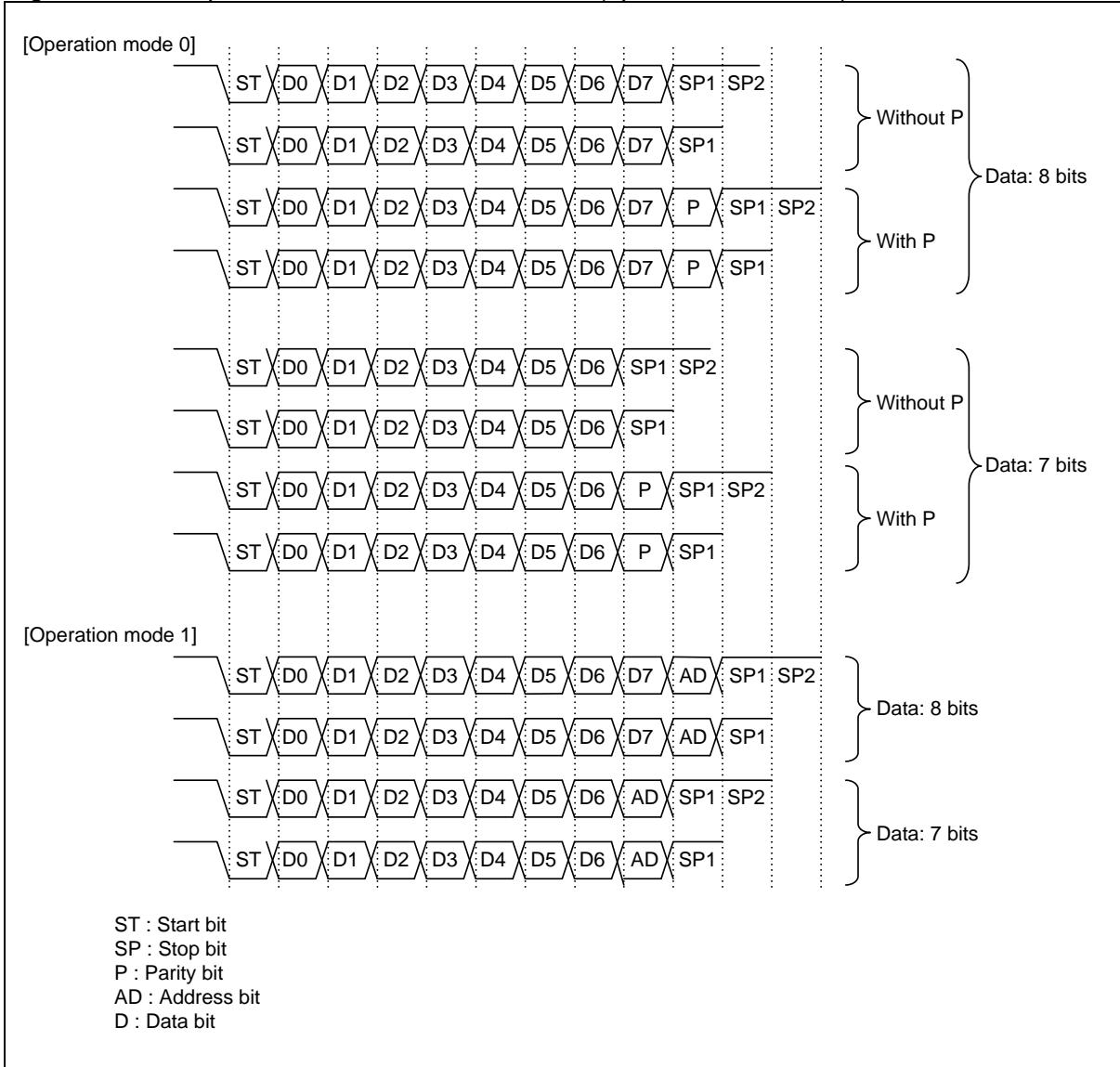
■ UART operation

● Transmit/receive data format

- Transmit/receive data always starts with a start bit, followed by transmission/reception of data with the specified data bit length, and ends with at least one-bit long stop bit.
- The BDS bit of the Serial Mode Register (SMR) determines the data transmit direction (LSB first or MSB first). If parity is used, the parity bit is always placed between the last data bit and the first stop bit.
- In operation mode 0 (normal mode), selection is possible to use or not to use parity.
- In operation mode 1 (multiprocessor mode), no parity is added, and instead, the AD bit is added.

Figure 3-1 shows the transmit/receive data formats for operation mode 0 and 1.

Figure 3-1 Example transmit/receive data format (operation mode 0/1)

**<Notes>**

- The above figure shows formats when the data length is set to 7 or 8 bits. (In operation mode 0, the data length can be set between 5 and 9 bits.)
- If the BDS bit of the Serial Mode Register (SMR) is set to "1" (MSB first), the bits are processed from D7, and then D6, D5, ... D1, and D0 (P), in that order.
- If the data length is set to X bits, the lower X bit of the Transmit/Receive Data Register (TDR/RDR) is enabled.

● Data transmission

- If the transmit data empty flag bit (TDRE) of the Serial Status Register (SSR) is "1", the transmit data can be written in the Transmit Data Register (TDR). (When transmit FIFO is enabled, transmit data can be written even if TDRE="0".)
- If transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag bit (SSR:TDRE) is set to "0".
- Setting the transmission enable bit of the serial control register (SCR:TXE) to "1" causes transmit data to be loaded to the transmit shift register, followed by sequential transmission starting with the start bit.
- When transmission starts, the transmit data empty flag bit (SSR:TDRE) is set to "1" again. If transmit interrupts are then enabled (SCR:TIE=1), a transmit interrupt is generated. In the interrupt processing, the next transmit data set can be written in the Transmit Data Register,

<Notes>

- As the transmit data empty flag bit (SSR:TDRE) is initially set to "1", a transmit interrupt occurs as soon as transmit interrupts are enabled (SCR:TIE).
 - As the FIFO transmit data request bit (FCR1:FDRQ) is initially set to "1", a transmit interrupt occurs as soon as FIFO transmit interrupts are enabled (FCR1:FTIE=1).
-

● Data reception

- When reception is enabled (SCR:RXE=1), the interface performs reception.
 - Upon detection of the start bit, one-frame reception takes place according to the data format set in the extended communications control register (ESCR:PEN, P, L2, L1, L0) and serial mode register (SMR:BDS). A start bit is detected when falling (ESCR:INV="0") is detected after passing the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) or if rising (ESCR:INV="1") is detected and "LOW" is detected for the data passing the sampling point.
 - When one-frame reception is completed, the receive data full flag bit (SSR:RDRF) is set to "1". If receive interrupts are then enabled (SCR:RIE=1), a receive interrupt is generated.
 - To read received data, perform reading of the received data after one-frame data reception is completed and check the state of the error flag of the Serial Status Register (SSR). Handle the receive error if it is occurring.
 - Reading of the received data causes the receive data full flag bit (SSR:RDRF) to be cleared to "0".
 - If receive FIFO is enabled, the receive data full flag bit (SSR:RDRF) is set to "1" when the number of received frames has reached the value set for receive FBYTE.
 - If all of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to zero (0). If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- If receive FIFO is enabled, receive FIFO does not store data in which an error has occurred when the error flag of the Serial Status Register (SSR) is set to "1". Also note that the receive data full flag bit (SSR:RDRF) is not set to "1". (However, the RDRF flag is set to "1" in an overrun error.) What the receive FBYTE indicates is the number of data sets received normally before the error occurred. Unless the error flag of the Serial Status Register (SSR) is cleared to "0", receive FIFO is not enabled.
 - If receive FIFO is enabled, the receive data full flag bit (SSR:RDRF) is cleared to "0" when all data in receive FIFO is out.

<Notes>

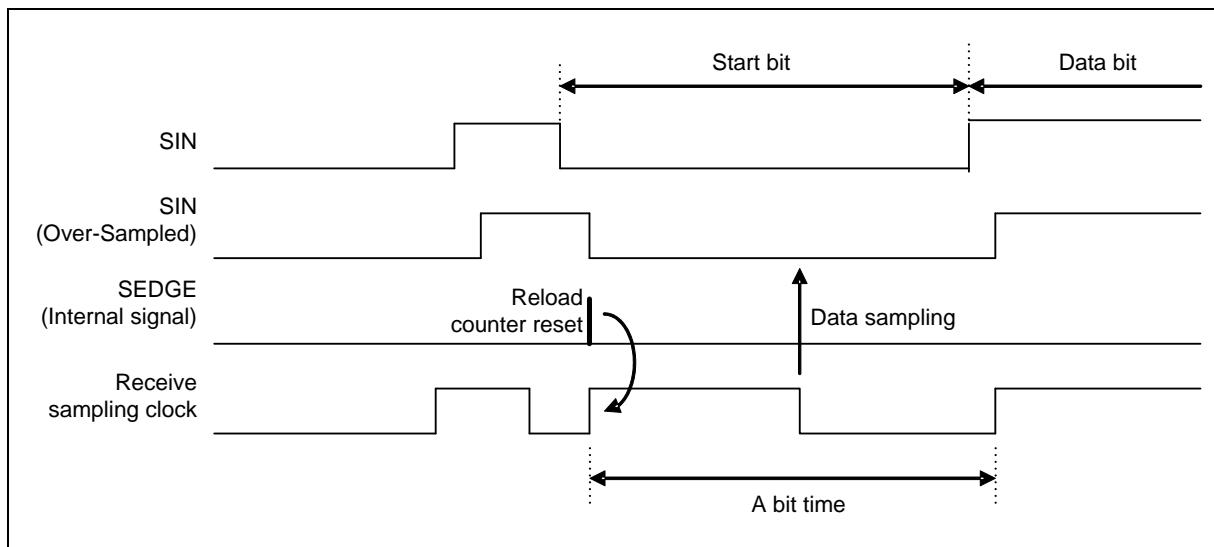
- Data in the Receive Data Register (RDR) becomes valid when the receive data register full flag bit (SSR:RDRF) is set to "1" and no receive error occurs (SSR:PE, ORE, FRE=0).
- Although a noise filter is built in (with the majority value applied after sampling serial data input three times with the bus clock), wrong data may be received if any noise passes through the filter. As a countermeasures, you can design the board so as not to allow noise to pass through this filter or perform communications so that noise that has passed may not cause any problem (by adding check sum of data at the end and resending the data if any error occurs, for example).
- During reception, if the following is detected at the same time as the stop bit sampling point or before the 1-2 bus clock, the relevant edge becomes invalid, which may disable normal reception of the next data. To output frames continuously, adequate intervals are required between frames.
 - The falling edge of serial data (When ESCR:INV="0")
 - The rising edge of serial data (When ESCR:INV="1")

● Clock selection

- You can use either an internal or external clock.
- To use the external clock, set SMR:EXT to "1". IN this case, the external clock is subject to frequency division by the baud rate generator.

● Start bit detection

- In asynchronous mode, the start bit is recognized based on detection of the falling edge of the SIN signal. For that reason, reception is not started unless the falling edge of the SIN signal is input even if reception is enabled (SCR:RXE=1).
- Upon detection of the start bit's falling edge, the receive reload counter of the baud rate generator is reset and reloaded to start countdown. Thus, sampling always takes place in the middle of data.



● Stop bit

- You can select the bit length to be between one and four.
- The receive data full flag bit (SSR:RDRF) is set to "1" upon detection of the first stop bit.

● Error detection

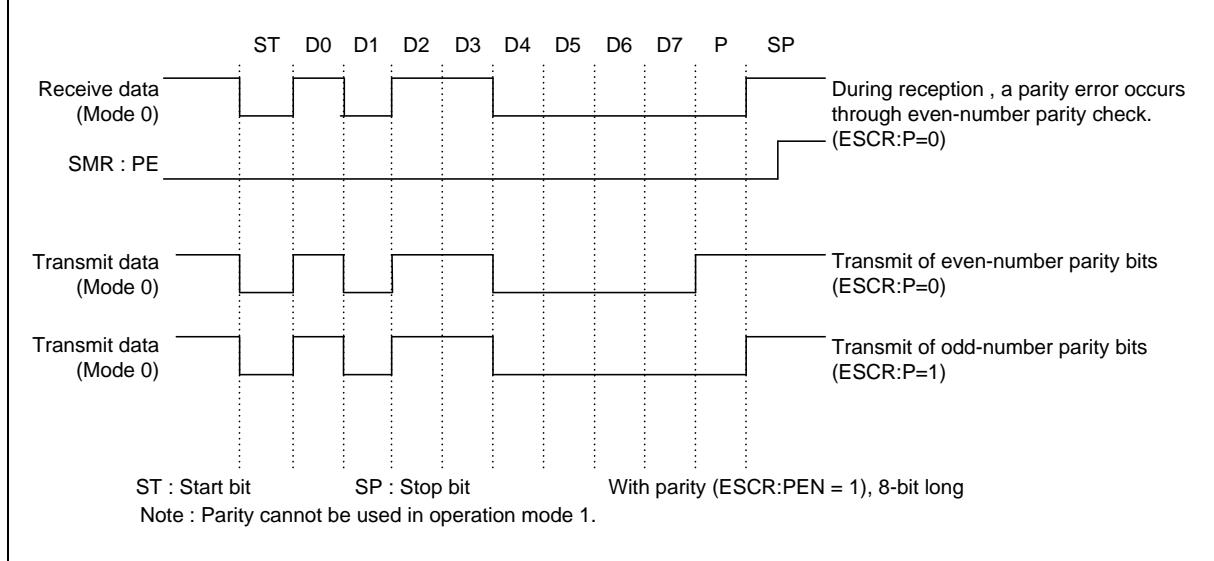
- In operation mode 0, parity, overrun and framing errors can be detected.
- In operation mode 1, overrun and framing errors can be detected but parity errors cannot be detected.

● Parity bit

- The parity bit can only be added in operation mode 0. The parity enable bit (ESCR:PE) can be used to specify use or non-use of parity and the parity selection bit (ESCR:P) to set even-number parity or odd-number parity.
- Parity cannot be used in operation mode 1.

Figure 3-2 shows transmit/receive data when parity is enabled.

Figure 3-2 Operation when parity is enabled

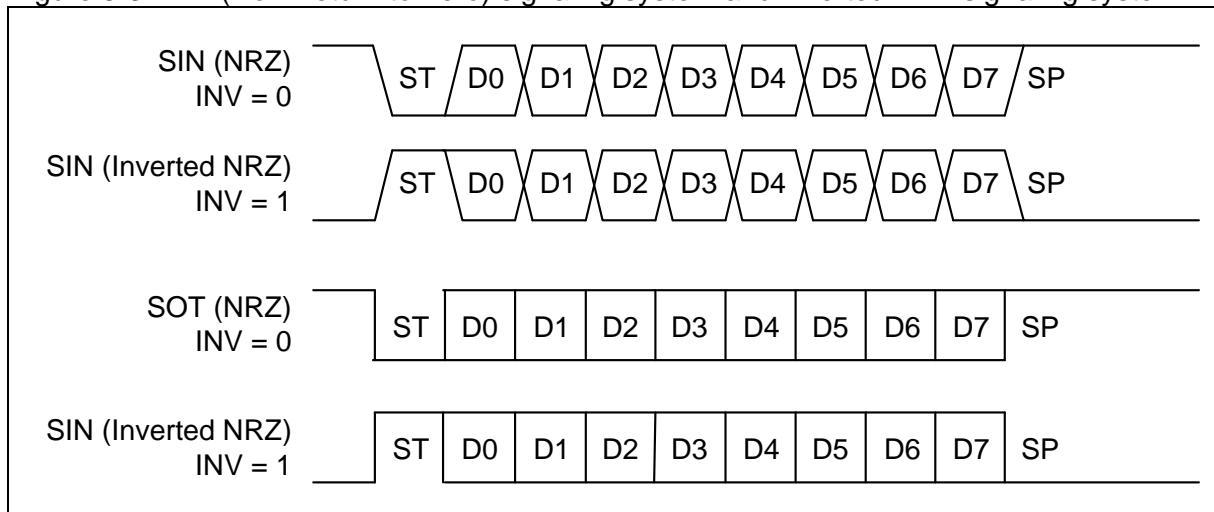


● Data signaling system

- By setting up the INV bit of the extended communications control register, you can select either the NRZ (Non Return to Zero) signaling system (ESCR:INV=0) or inverted NRZ signaling system (ESCR:INV=1).

Figure 3-3 shows the NRZ and inverted NRZ signaling systems.

Figure 3-3 NRZ (Non Return to Zero) signaling system and inverted NRZ signaling system



● Data transfer system

- For the data bit transfer method, either LSB first or MSB first can be selected.

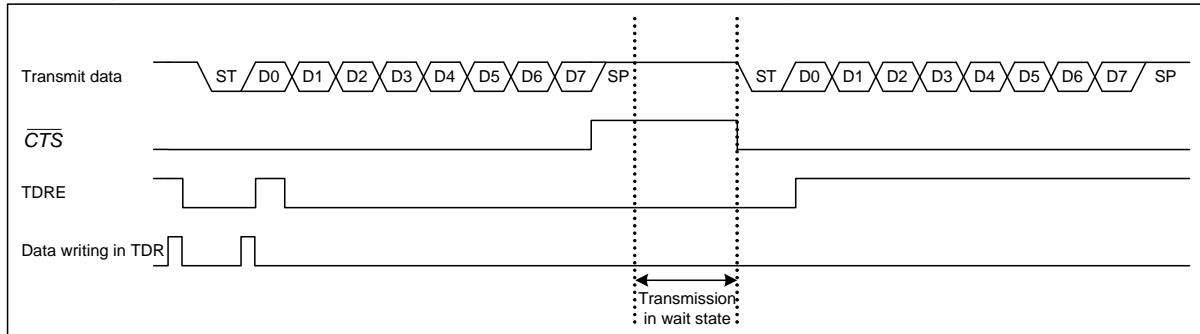
● Hardware flow control

When flow control is enabled (ESCR:FLWEN="1"), UART performs hardware flow control.

- During data transmission

If \overline{CTS} is "HIGH" after data is transmitted, the next data is not transmitted even if the transmit buffer contains data (TDRE="0") and the process waits until \overline{CTS} is set to "LOW". To have transmission wait, input "HIGH" in \overline{CTS} before the stop bit transmission is completed. Transmission continues up to the stop bit even if "HIGH" is input in \overline{CTS} during transmission.

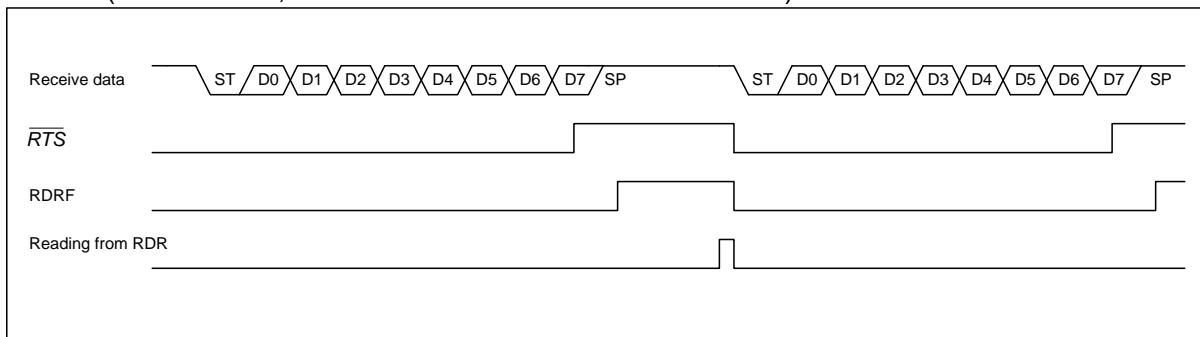
Figure 3-4 Hardware flow control during data transmission
(SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



- During data reception
- If FIFO is not used

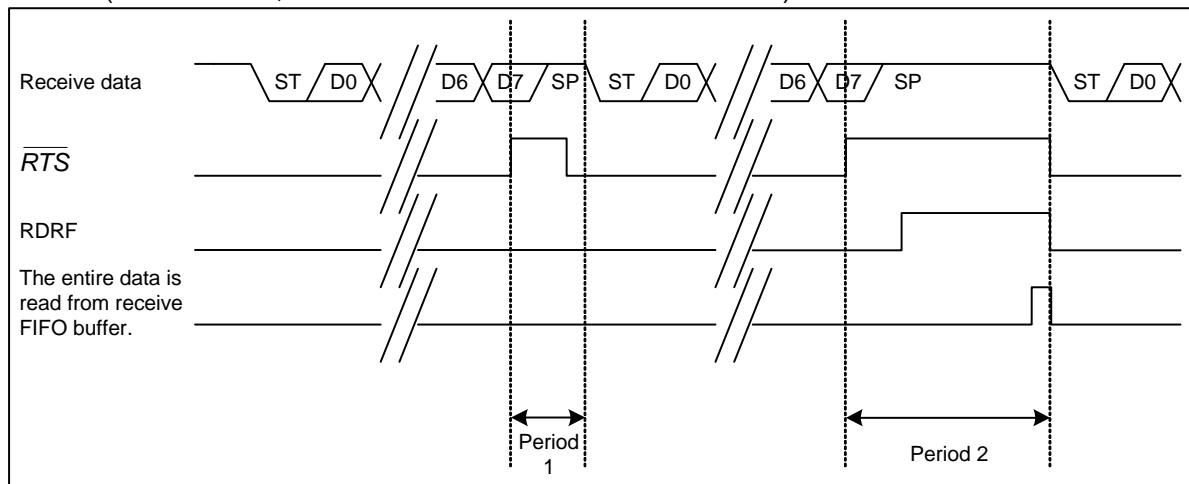
Upon reception of data one bit before the stop bit, "HIGH" is output to \overline{RTS} . After received data is read, "LOW" is output to \overline{RTS} .

Figure 3-5 Hardware flow control during data reception (with FIFO unused)
(SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



- If FIFO is used
 - If SSR:RDRF is not set (the specified number of data sets are not received in receive FIFO), \overline{RTS} outputs "HIGH" upon reception of data one bit before the stop bit, but \overline{RTS} outputs "LOW" upon detection of the stop bit. (For period 1)
 - If SSR:RDRF is set (the specified number of data sets are received in receive FIFO), \overline{RTS} outputs "HIGH" upon reception of data one bit before the stop bit. \overline{RTS} outputs "LOW" after all data is read from receive FIFO. (For period 2)

Figure 3-6 Hardware flow control during data reception (with FIFO used)
(SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



<Notes>

- When receive operation is disabled ($RXE=0$), the \overline{RTS} signal is fixed to "LOW".
- If both conditions below are satisfied when receive FIFO is used and if the receive idle state continues for more than 8 baud rate clocks, RDRF is set to "1" but "LOW" is maintained for the \overline{RTS} signal.
 - The receive FIFO idle detection enable bit (FCR1:FRIIE) is "1".
 - The preset data amount is not received and some data remains in receive FIFO.
- Performing programmable resetting (SCR:UPCL="1") clears the \overline{RTS} signal to "LOW".

4. Dedicated Baud Rate Generator

For the UART transmit/receive clock source, either of the following can be selected.

- Dedicated baud rate generator (reload counter)
- An external clock input to the baud rate generator (reload counter)

■ Selecting the UART baud rate

Select one of the following two baud rates.

● Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an internal clock by the set value.

To set the clock source, select an internal clock (BGR1:EXT=0).

● Baud rate obtained by dividing an external clock using the dedicated baud rate generator (reload counter)

Use an external clock for the clock source of the reload counter.

To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an external clock by the set value.

To set the clock source, select use of an external clock and the baud rate generator clock (BGR1:EXT=1).

This mode is designed for cases where an oscillator with a divided non-standard frequency is used.

<Notes>

- Set the external clock (BGR1:EXT=1) while the reload counter is suspended (BGR1/0=15' h00).
- If an external clock is selected (BGR1:EXT=1), its HIGH and LOW signals must have a width at least of two bus clocks.

4.1. Baud rate settings

The following explains how to set the baud rate, and also a result of serial clock frequency calculation.

■ Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$$V = \phi / b - 1$$

V : Reload value b : Baud rate ϕ : Bus clock frequency or external clock frequency

(2) Calculation example

To set the 16MHz bus clock, use the internal clock, and set the 19200-bps baud rate, set the reload value as follows.

Reload value:

$$V = (16 \times 1000000)/19200 - 1 = 832$$

Therefore, the baud rate is:

$$b = (16 \times 1000000)/(832 + 1) = 19208\text{bps}$$

(3) Baud rate error

The baud rate error can be calculated by the following equation.

$$\text{Error (\%)} = (\text{Calculated value} - \text{Target value})/\text{Target value} \times 100$$

Example: To set the 20MHz bus clock and 153600-bps target baud rate:

$$\text{Reload value} = (20 \times 1000000)/153600 - 1 = 129$$

$$\text{Baud rate (Calculated value)} = (20 \times 1000000)/(129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error (\%)} = (153846 - 153600)/153600 \times 100 = 0.16 (\%)$$

<Notes>

- If the reload value is set to "0", the reload counter is stopped.
- If the reload value is an even number, in the receive serial clock, the width of a "LOW" signal is longer than that of a "HIGH" signal by one bus clock cycle. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.
- Set the reload value to 4 or more. Note that data may not be received normally due to the baud rate error and reload value setting.

■ Reload value and baud rate for each bus clock frequency

Table 4-1 Reload values and baud rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4 M	-	-	-	-	-	0	4	0	5	0	7	0
2.5 M	-	-	-	0	-	-	-	-	-	-	-	-
2 M	-	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

- Value: BGR1/0 register set value (decimal)

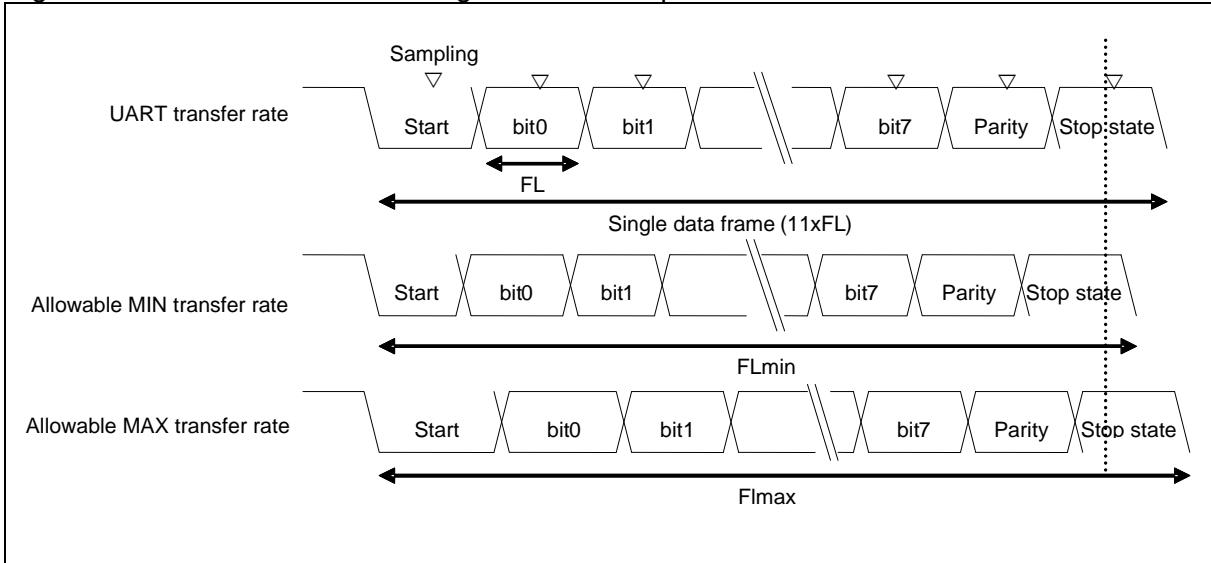
- ERR: Baud rate error (%)

■ Allowable baud rate range for data reception

The following shows the range of baud rate error allowed for the destination to receive data.

Set the reception baud rate error by using the following formulas to ensure that the value falls within the allowable range.

Figure 4-1 Allowable baud rate range for data reception



As shown in the figure, after detection of the start bit, the sampling timing of incoming data is determined by the counter set in the BGR1/0 register. Data can be received successfully if the bit sequence including the stop bit matches the sampling timing.

If this applies to a reception of 11 bits, a theoretical explanation can be given in the following.

Assuming that the sampling timing margin is one bus clock (ϕ), the minimum allowable transfer rate (FL_{min}) is determined as follows:

$$FL_{min} = (11 \text{bit} \times (V+1) - (V+1)/2 + 2)/\phi = (21V + 25)/2 \text{ } \phi \text{ (s)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

Thus, the maximum baud rate that allows the destination to receive data (BG_{max}) is determined as follows.

$$BG_{max} = 11/FL_{min} = 22\phi/(21V+25) \text{ (bps)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

When data is received at the maximum allowable transfer rate (FL_{max}), the starting point of the incoming 11th bit is sampled.

Thus, the maximum allowable transfer rate (FL_{max}) is determined as follows:

$$10/11 \times FL_{max} = (11 \text{bit} \times (V+1) - (V+1)/2)/\phi \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

$$FL_{max} = (21/20 \times 11 \times (V+1))/\phi$$

Assuming that the sampling timing margin (ϕ) is two clocks, the maximum allowable transfer rate (FL_{max}) is determined as follows:

$$10/11 \times FL_{max} = (11 \text{bit} \times (V+1) - (V+1)/2 - 2)/\phi \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

$$FL_{max} = (21/20 \times 11 \times (V+1) - 44/20)/\phi = (231V + 187)/20 \text{ } \phi \text{ (s)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

Accordingly, the minimum baud rate that allows the destination to receive data (BG_{min}) is determined as follows:

$$BG_{min} = 11/FL_{max} = 220\phi/(231V+187) \text{ (bps)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

From the above formulas for obtaining the minimum/maximum baud rate, the allowable error between UART and the destination is obtained as follows.

Reload value (V)	Maximum allowable baud rate error	Minimum allowable baud rate error
3	0%	0
10	+2.98%	-3.08%
50	+4.37%	-4.40%
100	+4.56%	-4.58%
200	+4.66%	-4.67%
32767	+4.76%	-4.76%

<Note>

Receive accuracy depends on the number of bits per frame, bus clock, and reload value. The higher the bus clock and frequency division ratio are, the higher the accuracy becomes.

■ External clock

Writing "1" to the EXT bit of the Baud Rate Generator Register (BGR) causes the baud rate generator to divide the external clock's frequency.

<Note>

The external clock signal synchronizes with the internal clock on UART. Therefore, an external clock that does not allow synchronization causes unstable operation.

■ Functions of reload counter

There are two types of reload counters: The transmit reload counter and the receive reload counter, both functioning as a dedicated baud rate generator. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from the external or internal clock.

■ Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

■ Restarting

The reload counter restarts counting in the following conditions.

● Common to transmit and receive reload counters

- A programmable reset (SCR:UPCL bit)

● Receive reload counter

- Detection of the start bit's falling edge in asynchronous mode

5. Setting Procedure and Program Flow in Operation Mode 0 (Async Normal Mode)

Operation mode 0 enables asynchronous bi-directional serial communications.

■ CPU-to-CPU connection

Select the bi-directional communication in operation mode 0 (normal mode). Connect two CPUs to each other as shown in Figure 5-1.

Figure 5-1 A connection example of bi-directional communications in UART operation mode 0 (with flow control disabled)

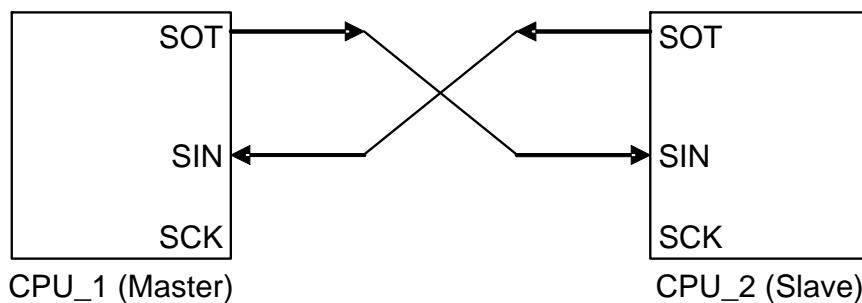
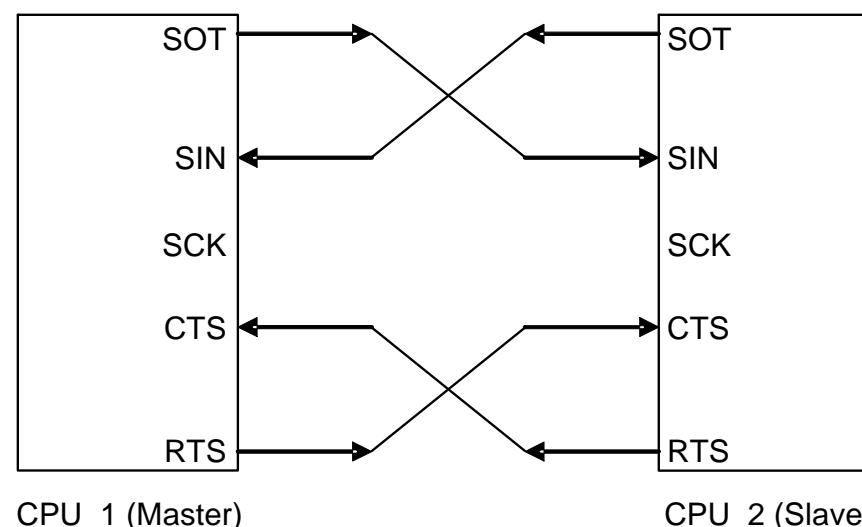


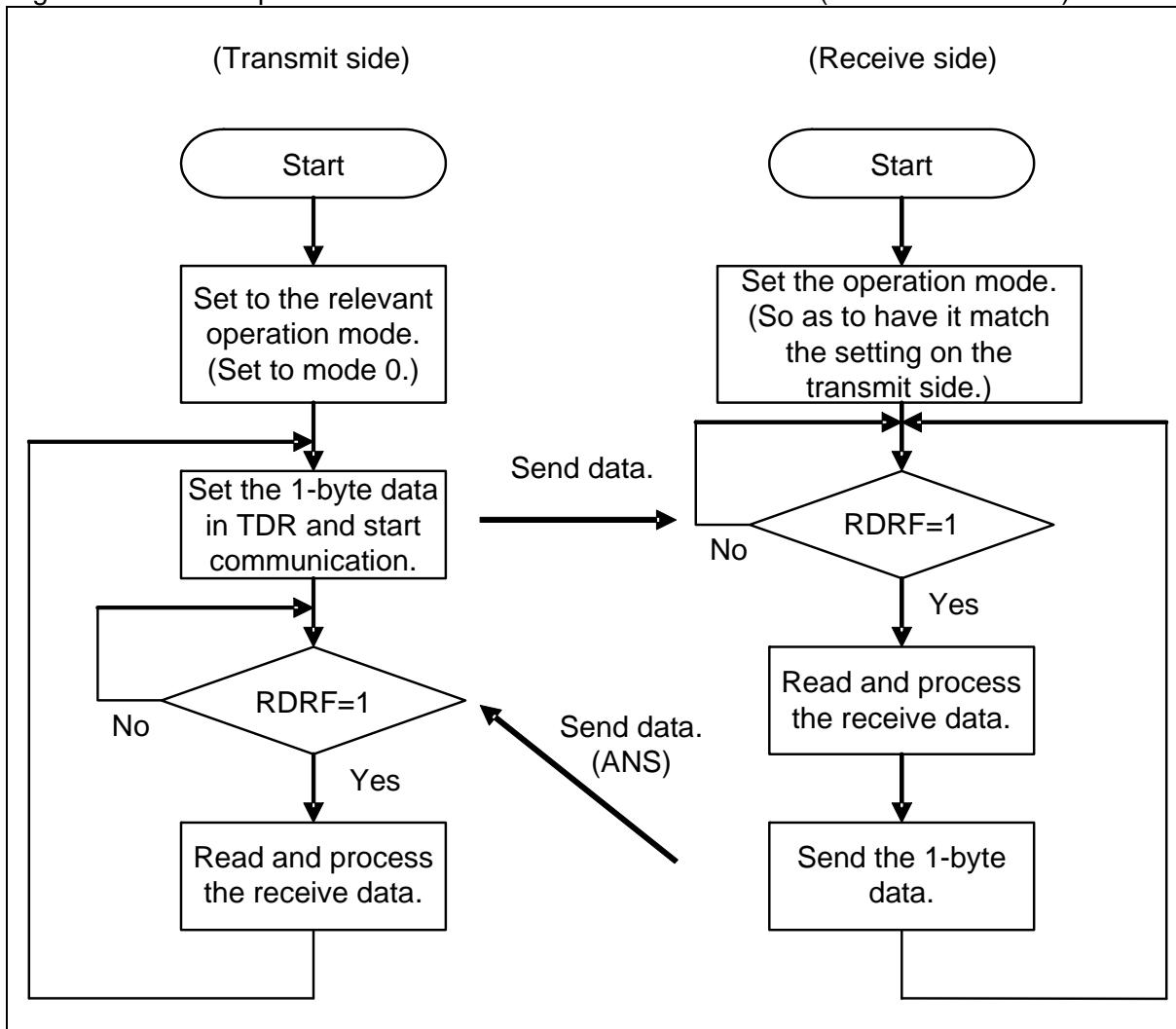
Figure 5-2 A connection example of bi-directional communications in UART operation mode 0 (with flow control disabled)



■ Flowcharts

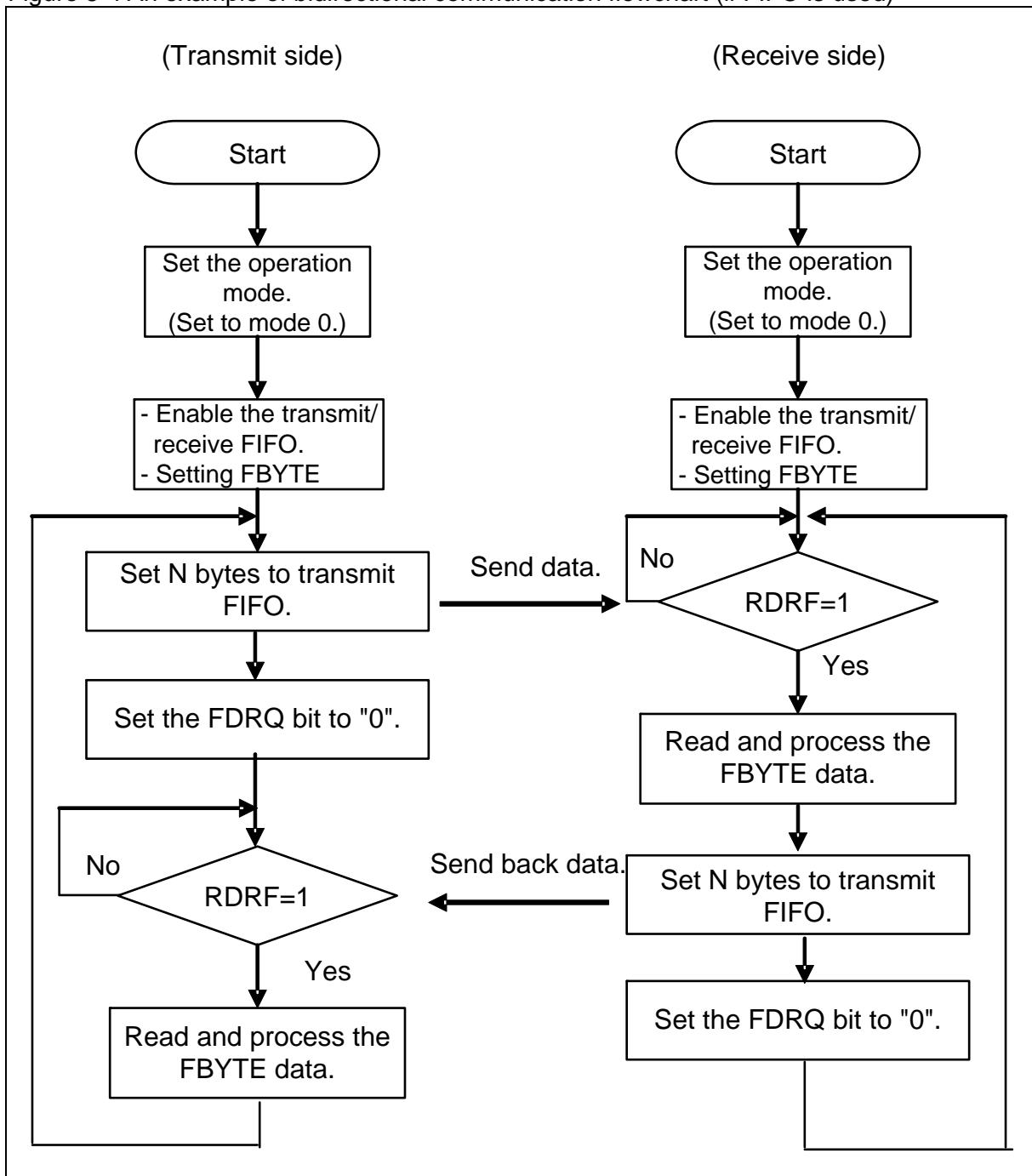
● If FIFO is not used

Figure 5-3 An example of bidirectional communication flowchart (if FIFO is not used)



● If FIFO is used

Figure 5-4 An example of bidirectional communication flowchart (if FIFO is used)



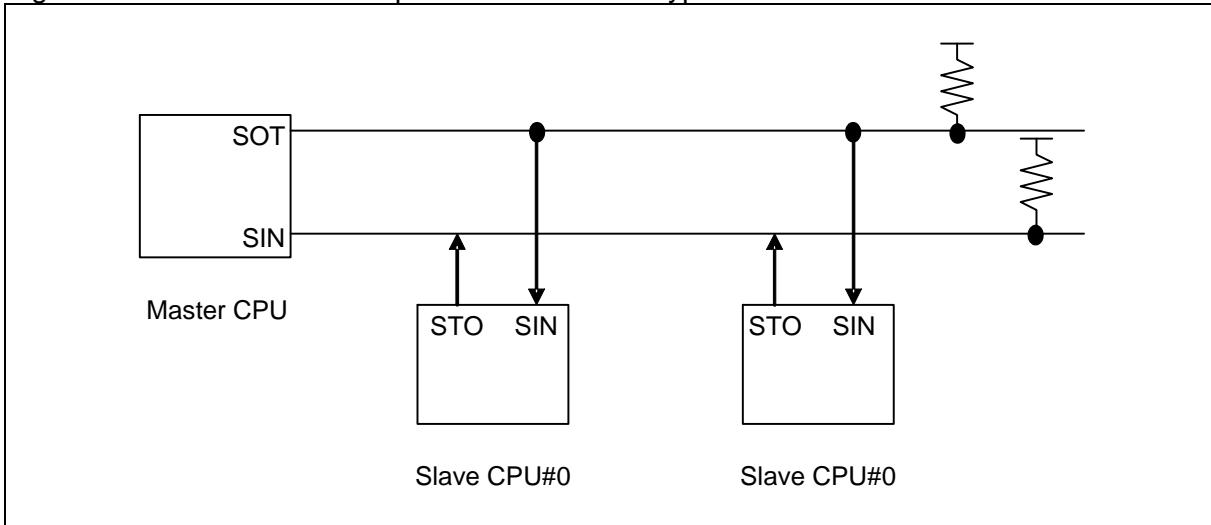
6. Setting Procedure and Program Flow in Operation Mode 1 (Async Multiprocessor Mode)

In operation mode 1 (multiprocessor mode), communications by master/slave connections with multiple CPUs. Either the master or slave function is available.

■ CPU-to-CPU connection

In a master/slave type communications, as shown in the figure, the communications system is configured with two common communication lines connected to the master CPU and multiple slave CPUs. UART can be used either as a master or a slave.

Figure 6-1 A connection example for master/slave type communications on UART



■ Function selection

In master/slave communications, select the operation mode and data transfer system, as shown in Table 6-1.

Table 6-1 Selection of master/slave type communications functions

	Operation mode		Data	Parity	Stop state Bit	Bit direction
	Master mode CPU	Slave mode CPU				
Address transmission and reception	Mode 1 (A/D bit transmission)	Mode 1 (A/D bit reception)	AD="1" + 7 or 8 bits Address	OFF	One bit or 2 bits	LSB or MSB first
Data transmission and reception			AD="0" + 7 or 8 bits Data			

<Note>

In operation mode 1, operate in word access mode for transmit/receive data (TDR/RDR).

● Communications procedure

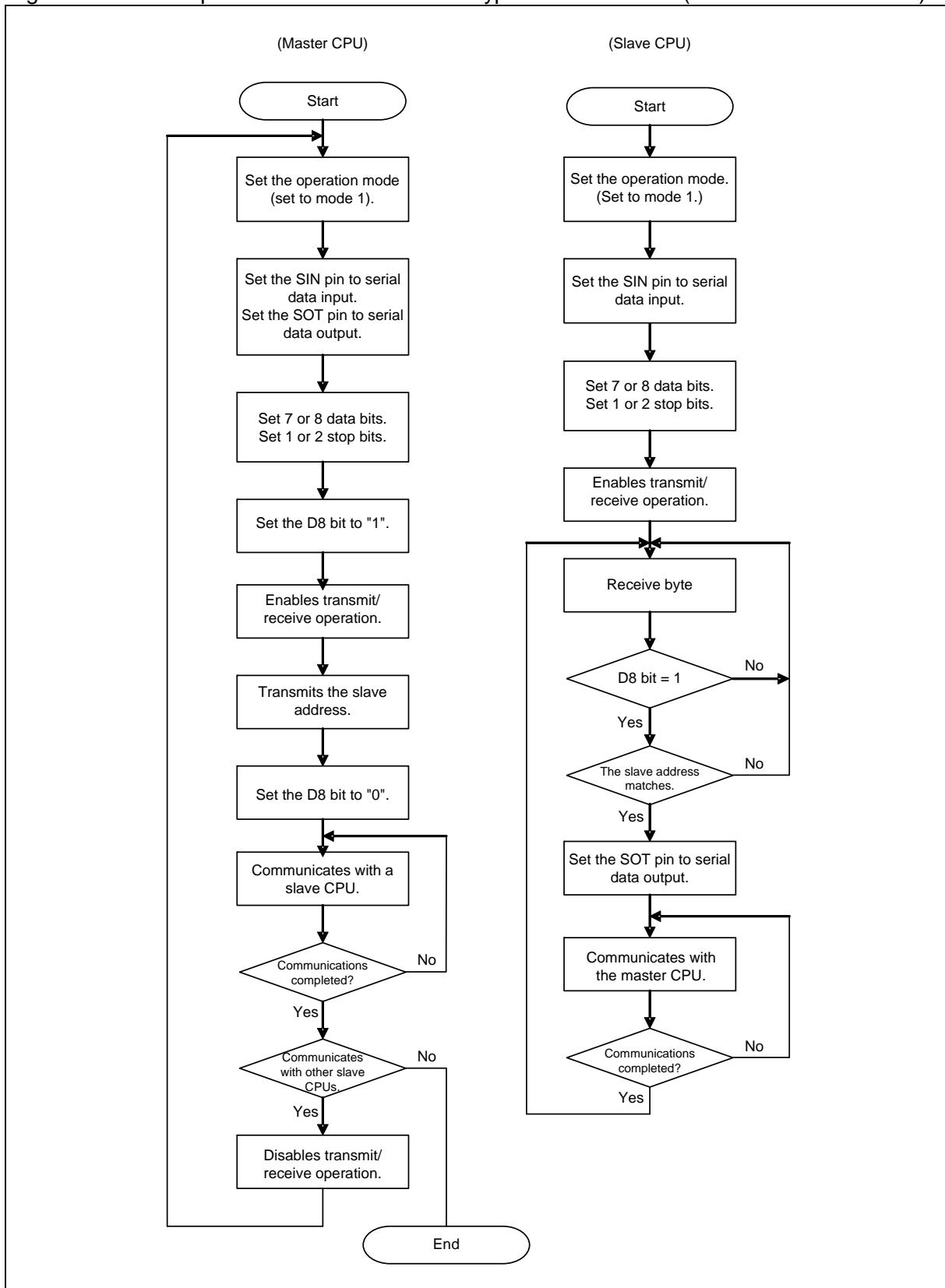
Communications start when the master CPU transmits address data. Address data is a data set whose D8 bit is "1", and used for selecting a slave CPU to communicate with. Each slave CPU judges the address as programmed, and communicates with the master CPU if that address matches the assigned address.

Figure 6-2 and Figure 6-3 show flowcharts of master/slave type communications (in multiprocessor mode).

■ Flowcharts

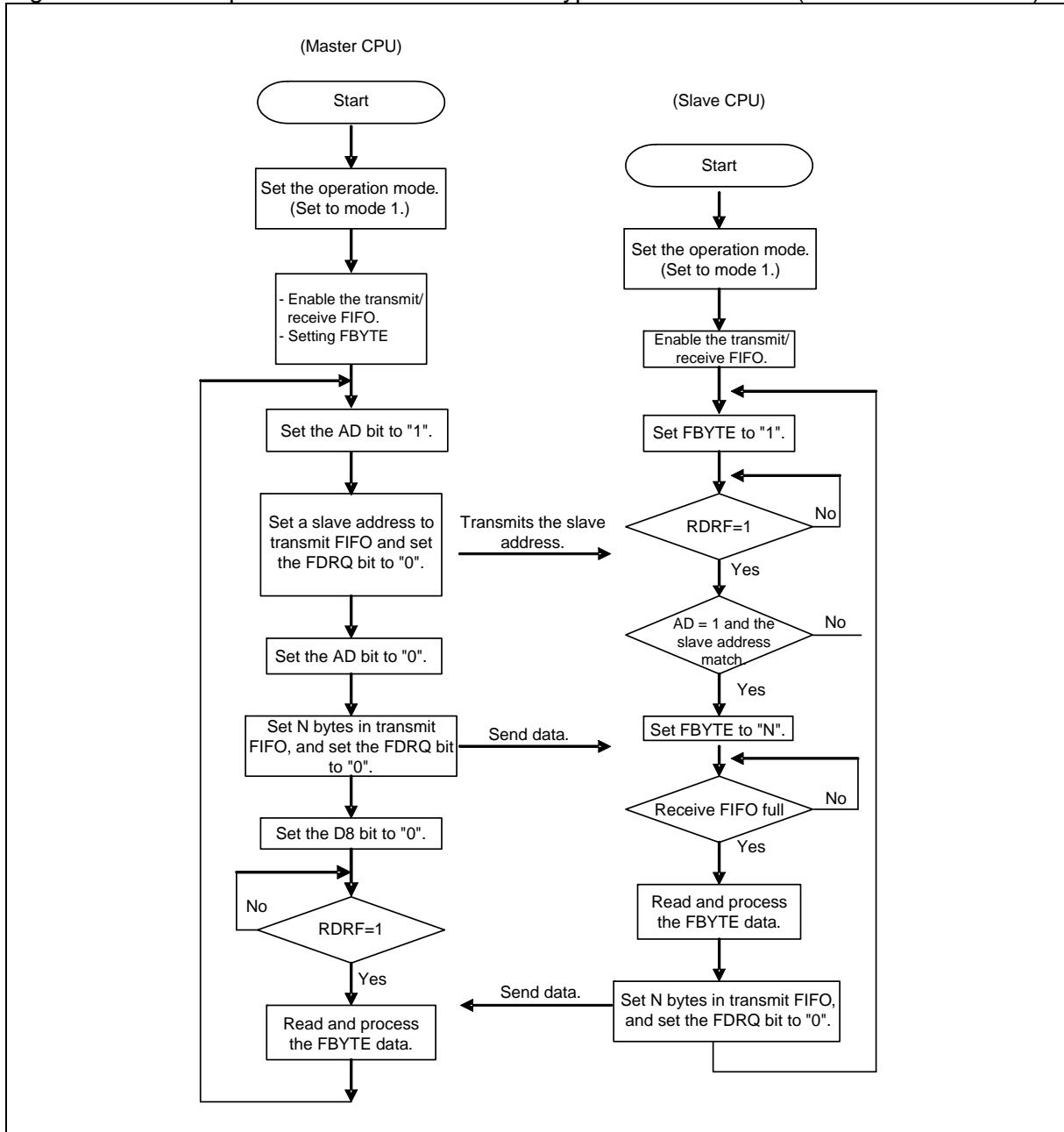
● If FIFO is not used

Figure 6-2 An example flowchart for master/slave type communications (if FIFO buffer is not used)



- If FIFO is used

Figure 6-3 An example flowchart for master/slave type communications (if FIFO buffer is used)



7. UART (Async Serial Interface) Registers

This section provides a list of UART (Async Serial Interface) registers.

■ UART (Async Serial Interface) registers list

Table 7-1 UART (Async Serial Interface) register list

	bit 15	bit 8	bit 7	bit 0
UART	SCR (Serial Control Register)		SMR (Serial Mode Register)	
	SSR (Serial Status Register)		ESCR (Extended Communication Control Register)	
	TDR1/RDR1 (Transmit/Receive Data Register 1)		TDR0/RDR0 (Transmit/Receive Data Register 0)	
	BGR1 (Baud Rate Generator Register 1)		BGR0 (Baud Rate Generator Register 0)	
	-		-	
FIFO	FCR1 (FIFO Control Register 1)		FCR0 (FIFO Control Register 0)	
	FBYTE2 (FIFO2 Byte Register)		FBYTE1 (FIFO1 Byte Register)	

Table 7-2 UART (Async Serial Interface) bit assignment

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
SCR/ SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SBL	BDS	-	SOE
SSR/ ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	FLWEN	ESBL	INV	PEN	P	L2	L1	L0
TDR/ (RDR)								D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-																
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

■ Operation mode

UART (Async Serial Interface) operates in two different modes. The Serial Mode Register (SMR) determines the mode to be enabled, depending on its setting, MD2, MD1 or MD0.

Table 7-3 UART (Async Serial Interface) operation modes

Operation mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (async normal mode)
1	0	0	1	UART1 (async multiprocessor mode)

7.1. Serial Control Register (SCR)

The Serial Control Register (SCR) can perform transmit/receive enable/disable, transmit/receive interrupt enable/disable, transmit bus idle interrupt enable/disable and UART reset operations.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	(SMR)		
Attribute	R/W	-	-	R/W	R/W	R/W	R/W	R/W			
Initial value	0	-	-	0	0	0	0	0			

[bit 15] UPCL: Programmable Clear bit

Initializes the UART internal state.

If set to "1",

- UART is reset directly (software reset). However, the current register settings are maintained. The transmit or receive state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/receive interrupt causes (SSR:PE, FRE, ORE, RDRF, TDRE and TBI) are initialized (to 0b000011).
- $\overline{\text{RTS}}$ signal is cleared to "LOW".

If set to "0", it has no effect.

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Programmable clear	

<Notes>

- Disable an interrupt first, and then execute the programmable clear instruction.
- If the FIFO operation is used, disable it (FCR0:FE2, FE1=0) first and then execute Programmable Clear.

[bit 14, 13] Unused bits

This bit value is unknown when read.

This bit has no effect when written.

[bit 12] RIE: Receive interrupt enable bit

- This bit enables or disables an output of receive interrupt request to the CPU.
- If the RIE bit and the receive data flag bit (SSR:RDRF) are "1", or if any of the error flag bits (SSR:PE, ORE or FRE) is "1", a receive interrupt request is output.

Bit	Description
0	Disables the receive interrupt.
1	Enables the receive interrupt.

[bit 11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of Transmit Interrupt Request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a Transmit Interrupt Request is output.

Bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit 10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and TBI bit are "1", a transmit bus idle interrupt request is output.

Bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit 9] RXE: Receive operation enable bit

Enables or disables UART receive operation.

- If set to "0", receive operation is disabled.
- If set to "1", receive operation is enabled.

Bit	Description
0	Disables data reception.
1	Enables data reception.

<Notes>

- Reception is not started unless the falling edge of the start bit (in NRZ format, when ESCR:INV=0) is input even if reception is enabled (RXE=1). (In the inverted NRZ format (ESCR:INV=1), reception is not started unless the falling edge is input.)
- If data reception is disabled (RXE=0), the current data reception is stopped immediately.
- When receive operation is disabled (RXE=0), $\overline{\text{RTS}}$ signal is fixed to "LOW".

[bit 8] TXE: Transmit operation enable bit
Enables or disables UART transmit operation.

- If set to "0", transmit operation is disabled.
- If set to "1", transmit operation is enabled.

Bit	Description
0	Disables the transmission.
1	Enables the transmission.

<Note>

If data transmission is disabled (TXE=0), the current data transmission is stopped immediately.

7.2. Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set operation mode, transfer direction data length and to select the stop bit length as well as to enable/disable output of serial data to their pins.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SCR)		MD2	MD1	MD0	WUCR	SBL	BDS	-	SOE	
Attribute			R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	
Initial value			0	0	0	0	0	0	0	0	

[bit 7:5] MD2, MD1, MD0: Operation mode set bit

Sets operation mode of the Async Serial Interface.

"0b000" : Sets operation mode 0 (async normal mode).

"0b001" : Sets operation mode 1 (async multiprocessor mode).

"0b010" : Sets operation mode 2 (clock sync mode).

"0b011" : Sets operation mode 3 (LIN communication mode).

"0b100" : Sets operation mode 4 (I²C mode).

This section explains the registers and their operation in operation mode 0 (async normal mode) and in operation mode 1 (async multiprocessor mode).

Bit 7	Bit 6	Bit 5	Description
0	0	0	Operation mode 0 (async normal mode)
0	0	1	Operation mode 1 (async multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I ² C mode)

* This section explains the registers and their operation in Operation mode 0 and in operation mode 1.

<Notes>

- Any bit setting other than above is inhibited.
- To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.
- After the operation mode has been switched, set each register correctly.

[bit 4] WUCR: Wake-up control bit

Selects a pin to be used for an external interrupt.

If set to "0", the INT pin is set as an external interrupt pin.

If set to "1", the SIN pin is set as an external interrupt pin.

Bit	Description
0	Disables the Wake-up function.
1	Enables the Wake-up function.

[bit 3] SBL: Stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

If set to SBL="0" and ESCR:ESBL="0", the stop bit length is set to one bit.

If set to SBL="1" and ESCR:ESBL="0", the stop bit length is set to two bits.

If set to SBL="0" and ESCR:ESBL="1", the stop bit length is set to three bits.

If set to SBL="1" and ESCR:ESBL="1", the stop bit length is set to four bits.

Bit	Description	
0	ESCR.ESBL=0	1 bit
	ESCR.ESBL=1	3 bits
1	ESCR.ESBL=0	2 bits
	ESCR.ESBL=1	4 bits

<Notes>

- In receive operation, only the first bit of the stop bit data is detected.
- Always set this bit when transmission is disabled (SCR:TXE=0).

[bit 2] BDS: Transfer direction select bit

Specifies to transmit the least significant bit of the transmit serial data first (LSB first; BDS=0) or the most significant bit first (MSB first; BDS=1).

Bit	Description
0	LSB first (The least significant bit is first transferred.)
1	MSB first (The most significant bit is first transferred.)

<Note>

Set this bit when transmission and reception are disabled (SCR:TXE=0, SCR:RXE=0).

[bit 1] Reserved bit

This is an undefined bit. The read value is "0". Be sure to write "0".

[bit 0] SOE: Serial data output enable bit
This bit enables or disables a serial data output.

Bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

<Note>

If this bit is used as the SOUT pin, the GPIO must also be set.

7.3. Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmission/reception state, check the receive error flag, and clears the receive error flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI		(ESCR)	
Attribute	R/W	-	R	R	R	R	R	R			
Initial value	0	-	0	0	0	0	1	1			

[bit 15] REC: Receive error flag clear bit

This bit clears the PE, FRE and ORE flags of the Serial Status Register (SSR).

- If this bit is set to "1", the error flag is cleared.
- This bit has no effect if set to "0".

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Clears the receive error flag (PE, FRE, ORE).	

[bit 14] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 13] PE: Parity error flag bit (only functions in operation mode 0)

- If a parity occurs during data reception with SMR:PEN=1, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the PE bit and SCR:RIE bit are "1", a receive interrupt request is output.
- If this flag is set, data in the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No parity error occurred.
1	No parity error occurred.

[bit 12] FRE: Framing error flag bit

- If a framing error occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the FRE bit and SCR:RIE bit are "1", a receive interrupt request is output.
- If this flag is set, data in the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No framing error occurred.
1	A framing error occurred.

[bit 11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SCR:RIE bits are "1", a receive interrupt request is output.
- If this flag is set, data in the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit 10] RDRF: Receive data full flag

- This flag shows the state of Receive Data Register (RDR).
- When the receive data is loaded in the RDR, this bit is set to "1". When data is read from the Receive Data Register (RDR), this bit is cleared to "0".
- If the RDRF bit and SCR:RIE bit are "1", a receive interrupt request is output.
- If the receive FIFO is used and if a certain count of data is received by the receive FIFO, the RDRF bit is set to "1".
- If receive FIFO is used, if both of the following conditions are satisfied, and if the Receive Idle state continues more than 8 baud rate clocks, the RDRF bit is set to "1".
 - The receive FIFO idle detection enable bit (FCR1:FRIIE) is "1".
 - The preset data amount is not received and some data remains in receive FIFO.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted.
- If the receive FIFO is used and if this buffer is emptied, this bit is cleared to "0".

Bit	Description
0	The Receive Data Register (RDR) is empty.
1	The Receive Data Register (RDR) contains data.

[bit 9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TDRE bit and SCR:TIE bit are "1", a transmit interrupt request is output.
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/reset timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

Bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register is empty.

[bit 8] TBI: Transmit bus idle flag

- This bit indicates that UART is not transmitting data.
- When transmit data is written in the Transmit Data Register (TDR), this bit is set to "0".
- If the Transmit Data Register is empty (TDRE=1) and not transmitting data, this bit is set to "1".
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TBI bit is set to "1".
- If this bit is "1" and if the transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

Bit	Description
0	During data transmission
1	No data transmission

7.4. Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to set a transmit/receive data length, enable/disable a parity bit, select a parity bit, invert the serial data format and set stop bit length selection.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SSR)		FLWEN	ESBL	INV	PEN	P	L2	L1	L0	
Attribute			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value			0	0	0	0	0	0	0	0	0

[bit 7] FLWEN: Flow control enable bit

Selects to enable or disable hardware flow control operation.

- If set to "0", hardware flow control is disabled.
- If set to "1", hardware flow control is enabled.

Bit	Description
0	Disables hardware flow control.
1	Enables hardware flow control.

<Notes>

- Set this bit when data transmission and reception is disabled (SCR:TXE=0, RXE=0).
- Set this bit to "1" only when hardware flow control is desired.

[bit 6] ESBL: Extension stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

If set to SBL="0" and ESCR:ESBL="0", the stop bit length is set to one bit.

If set to SBL="1" and ESCR:ESBL="0", the stop bit length is set to two bits.

If set to SBL="0" and ESCR:ESBL="1", the stop bit length is set to three bits.

If set to SBL="1" and ESCR:ESBL="1", the stop bit length is set to four bits.

Bit	Description	
0	SMR.SBL=0	1 bit
	SMR.SBL=1	2 bits
1	SMR.SBL=0	3 bits
	SMR.SBL=1	4 bits

<Notes>

- In receive operation, only the first bit of the stop bit data is detected.
- Always set this bit when transmission is disabled (SCR:TXE=0).

[bit 5] INV: Inverted serial data format bit

Selects NRZ or inverted NRZ for the serial data format.

Bit	Description
0	NRZ format
1	Inverted NRZ format

[bit 4] PEN: Parity enable bit (only functions in operation mode 0)

Sets to add (for transmission) and detect (for reception) a parity bit or not to.

- If set to "0", no parity is added.
- If set to "1", a parity is added.

Bit	Description
0	Disables parity.
1	Enables parity.

<Note>

In operation mode 1, this bit is internally fixed at "0".

[bit 3] P: Parity select bit (only functions in operation mode 0)

When set to enable parity (ESCR:PEN=1, this bit is set to either odd-number parity "1" or even-number parity "0").

- If set to "0", set to even-number parity.
- If set to "1", set to odd-number parity.

Bit	Description
0	Even-number parity
1	Odd-number parity

[bit 2:0] L2, L1, L0: Data length select bit

These bits set a length of transmit/receive data.

- If set to "0b000", the data length is set to eight bits.
- If set to "0b001", the data length is set to five bits.
- If set to "0b010", the data length is set to six bits.
- If set to "0b011", the data length is set to seven bits.
- If set to "0b100", the data length is set to nine bits.

Bit 2	Bit 1	Bit 0	Description
0	0	0	8-bit length
0	0	1	5-bit length
0	1	0	6-bit length
0	1	1	7-bit length
1	0	0	9-bit length

<Notes>

- Any setting other than the above is inhibited.
- In operation mode 1, set the data length to seven or eight bits. Any other setting is inhibited.

7.5. Receive Data Register/Transmit Data Register (RDR/TDR)

The Receive and Transmit Data Registers are allocated at the same address. This register functions as the Receive Data Register when data is read from it. This register operates as the Transmit Data Register when data is written in it.

When FIFO operation is enabled, the RDR/TDR address functions as the FIFO read/write address.

■ Receive Data Register (RDR)

bit	15	...	9	8	7	6	5	4	3	2	1	0
Field			D8	D7	D6	D5	D4	D3	D2	D1	D0	
Attribute			R	R	R	R	R	R	R	R	R	R
Initial value			0	0	0	0	0	0	0	0	0	0

The Receive Data Register (RDR) is a 9-bit data buffer register for serial data reception.

- When serial data signals are sent to the Serial Input pin (SIN pin), they are converted by a shift register and stored in the Receive Data Register (RDR).
- The high-order bits are sequentially set to "0" according to the data length, as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X represents the receive data bit.)

- When the received data is stored in the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is set to "1". If a receive interrupt is enabled (SSR:RIE=1), a receive interrupt request is generated.
- The Receive Data Register (RDR) must be read only when the receive data full flag bit (SSR:RDRF) is "1". When data is read from the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a receive error occurs (when SSR:PE, ORE or FRE is "1"), data in the Receive Data Register (RDR) becomes invalid.
- In operation mode 1 (multiprocessor mode), 7-bit or 8-bit long operation takes place and the received AD bit is stored in the D8 bit.
- For 9-bit long data transfer and in operation mode 1, data must be read from RDR by 16-bit data accessing.

<Notes>

- If the Receive FIFO is used and if the preset amount of data is received in the Receive FIFO buffer, SSR:RDRF is set to "1".
- If the receive FIFO is used and if this buffer is emptied, the SSR:RDRF bit is cleared to "0".
- If a receive error occurs when receive FIFO is used (SSR:PE, ORE, or FRE is "1"), the receive FIFO enable bit is cleared and the receive data is not stored in the receive FIFO buffer.

■ Transmit Data Register (TDR)

bit	15	...	9	8	7	6	5	4	3	2	1	0
Field			D8	D7	D6	D5	D4	D3	D2	D1	D0	
Attribute			W	W	W	W	W	W	W	W	W	W
Initial value			1	1	1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a 9-bit data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the Transmit Shift Register. The transmit data is then converted into serial data and sent out from the serial data output pin (the SOUT pin).
- The high-order bits are sequentially made invalid according to the data length as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", transmit data can be written. If a transmit interrupt is enabled, a transmit interrupt occurs. Perform transmit data write after a transmit interrupt is generated or when the transmit interrupt data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and transmit FIFO is disabled or the transmit FIFO buffer is full, no transmit data can be written.
- In operation mode 1 (multiprocessor mode), 7-bit or 8-bit long operation takes place and the AD bit is sent by writing to the D8 bit.
- For 9-bit long data transfer and in operation mode 1, data must be written in TDR by 16-bit data accessing.

<Notes>

- The Transmit Data Register is a write-only register. While the Receive Data Register is a read-only register. As the transmit and receive registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instructions cannot be used.
- For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

7.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks. Also, an external clock can be selected as the clock source of the reload counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXT	(BGR1)						(BGR0)								
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.
- The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.
- The EXT bit (bit 15) specifies to use the clock source of reload counter as the internal clock or use an external clock. If EXT=0 is set, an internal clock is used. If EXT=1 is set, an external clock is used.

[bit 15] EXT: External clock select bit

Bit	Description
0	Uses the internal clock.
1	Uses an external clock.

[bit 14:8] BGR1: Baud Rate Generator Register 1

Bit 14:8	Description
Write	Write data in reload counter bit 8 to 14.
Read	Reads the BGR1 set value.

[bit 7:0] BGR0: Baud Rate Generator Register 0

Bit 7:0	Description
Write	Write data in reload counter bit 0 to 7.
Read	Reads the BGR0 set value.

<Notes>

- Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) by 16-bit data accessing.
- If the current values of Baud Rate Generator Registers (BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/0 set values and execute the programmable clear (UPCL).
- If the reload value is an even number, in the receive serial clock, the width of a "LOW" signal is longer than that of a "HIGH" signal by one bus clock cycle. If the value is an odd number, the width of a LOW signal is the same as that of a HIGH signal.
- Set a value "4" or higher to BGR1/0. Note that data may not be received successfully depending on the baud rate error and reload value settings.
- To change the setting to an external clock (EXT=1) while the Baud Rate Generator is running, write "0" to the Baud Rate Generators 1 and 0 (BGR1, BGR0), execute Programmable Clear (UPCL) and then set for an external clock (EXT=1).

7.7. FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select the transmit or receive FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL		(FCR0)	
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W			
Initial value	0	0	-	0	0	1	0	0			

[bit 15:14] FTST1, FTST0: FIFO test bits

They are FIFO Test bits.

They must always be set to "0".

Bit 15:14	Description
0	Disables the FIFO test.
1	Enables the FIFO test.

<Note>

If this bit is set to "1", the FIFO test is executed.

[bit 13] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 12] FLSTE: Re-transmit data lost detect enable bit

This bit enables the FIFO re-transmit data lost flag (FLST) detection.

If set to "0", the FLST bit detection is disabled.

If set to "1", the FLST bit detection is enabled.

Bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

<Note>

If you wish to set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".

[bit 11] FRIIE: Receive FIFO idle detection enable bit

This bit sets to detect the receive idle state if the receive FIFO contains valid data and if it continues more than 8-bit hours. If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt is generated when the receive idle state is detected.

If set to "0", a detection of receive idle state is disabled.

If set to "1", a detection of receive idle state is enabled.

Bit	Description
0	Disables the receive FIFO idle detection.
1	Enables the receive FIFO idle detection.

<Note>

In case of using Receive FIFO, set this bit to "1".

[bit 10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. At this time, if a transmit FIFO interrupt is enabled (FTIE=1), a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

Bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

<Notes>

- "0" written when transmit FIFO is enabled is valid.
- If the FBYTE (for transmission) is "0", this bit cannot be set to "0".
- If this bit is set to "1", it has no effect on the operation.
- If a read-modify-write instruction is issued, "1" is read.

[bit 9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

Bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit 8] FSEL: FIFO select bit

This bit selects the transmit or receive FIFO.

If set to "0", transmit FIFO is assigned FIFO1, and the receive FIFO is assigned FIFO2.

If set to "1", transmit FIFO is assigned FIFO2, and the receive FIFO is assigned FIFO1.

Bit	Description
0	Transmit FIFO:FIFO1; Receive FIFO:FIFO2
1	Transmit FIFO:FIFO2; Receive FIFO:FIFO1

<Notes>

- This bit is not cleared by the FIFO Reset (FCR0:FCL2, FCL1=1).
 - To change this bit state, first disable the FIFO operation (FCR:FE2, FE1=0).
-

7.8. FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15	...	8	7	6	5	4	3	2	1	0	
Field	(FCR1)				-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute					-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value					0	0	0	0	0	0	0	0

[bit 7] Unused bit

When read, always "0" is read.

When written, always set this bit to "0".

[bit 6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- Data is written (overwritten) in the FIFO buffer when the FLSTE bit of FIFO Control Register 1 (FCR1) is "1" and the write pointer for transmit FIFO matches the read pointer which has been saved by the FSET bit.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten.

Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

Bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit 5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.

When the re-transmission setting has finished, this bit is set to "0".

Bit	Description
0	Not reloaded
1	Reloaded

<Notes>

- If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.
- When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".
- After you have set the TIE bit and TBIE bit to "0", set this bit to "1". After you have enabled transmit FIFO, set the SCR:TIE bit and SCR:TBIE bit to "1".

[bit 4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer value is saved before being transmitted and if the FLST bit is "0", the data can be re-transmitted even if a communication error or others have occurred.

If set to "1", the current read pointer value is saved.

If set to "0", it has no effect.

Bit	Description	
	During writing	During reading
0	Not saved	"0" is always read.
1	FIFO2 is reset.	

<Note>

This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".

[bit 3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST bit is initialized, and the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO2 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO2.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The valid data count of the FBYTE2 register is set to "0".

[bit 2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 state.

If this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST bit is initialized, and the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO1 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO1.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The valid data count of the FBYTE1 register is set to "0".

[bit 1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO (FCR1:FSEL=1) and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both SCR:TIE bit and SCR:TBIE bit to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO2 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

Bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit 0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- When the FIFO1 is set as transmit FIFO (FCR1:FSEL=0) and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the UART is set to enable data transmission (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both TIE bit and SCR:TBIE bit to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO1 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

Bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

7.9. FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a receive interrupt when certain number of data sets are received in the receive FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	R/W								R/W							
Initial value	0 0 0 0 0 0 0 0								0 0 0 0 0 0 0 0							

The FBYTE register indicates the effective data count of data written from or received in FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 7-4 Display of data count

FSEL	FIFO selection	Data count display
0	FIFO2: Receive FIFO, FIFO1: Transmit FIFO	FIFO2=FBYTE2, FIFO1=FBYTE1
1	FIFO2:Transmit FIFO, FIFO1:Receive FIFO	FIFO2=FBYTE2, FIFO1=FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to flag a receive interrupt for the FBYTE register of receive FIFO. If this specified transfer count matches the FBYTE register display, the interrupt flag (SSR:RDRF) is set to "1".
- If both conditions below are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to zero (0). If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.

FBYTE2, FBYTE1: FIFO2 data count display bit, FIFO1 data count display bit

During writing	Sets the transfer data count.
During reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in the FIFO buffer but not transmitted yet
During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
During reception: Set the data count to generate a receive interrupt.

<Notes>

- Set "0x00" in the FBYTE register of transmit FIFO.
- Set a data value equal to or greater than "2" in the FBYTE register of receive FIFO.
- This state can be changed only after the data reception has been disabled.
- A read-modify-write instruction cannot be used for this register.
- Any setting exceeding the FIFO capacity is inhibited.
- When all the following requirements are met, the receive data full flag bit (SSR:RDRF) is not set to "1" even though the effective data of FBYTE setting number exist in the receive FIFO. If the FBYTE register is set to "2" or greater, this operation will not occur.

- FBYTE is set to "1".
- The effective data count is "1", same as the number specified in FBYTE register.
- When the multi function serial interface macro receives the data, and writes received data in the reception FIFO, the data of the reception FIFO are read at the same time.

However, after that, the receive data full flag bit (SSR:RDRF) will be set to "1" at any of the following conditions.

- The next data is received.
- The receive idle state of 8 bits or longer is detected when the receive FIFO idle is enabled (FCR:FRIIE=1)

Chapter: CSIO (Clock Sync Serial Interface)

This chapter explains the Clock Sync Serial Interface (CSIO) function that is supported in Operation mode 2. This CSIO is a part of the multifunction serial interface functions.

1. Outline of CSIO (Clock Sync Serial Interface)
2. CSIO (Clock Sync Serial Interface) interrupts
3. CSIO (Clock Sync Serial Interface) operations
4. Dedicated baud rate generator
5. CSIO (Clock Sync Serial Interface) registers

1. Outline of CSIO (Clock Sync Serial Interface)

The CSIO is a generic serial data communication interface (supporting the SPI) to allow synchronous communication with an external device. It also has transmit/receive FIFO (up to 128×9 bits each) ^{*1}installed.

■ CSIO (Clock Sync Serial Interface) functions

		Function
1	Data buffer	<ul style="list-style-type: none"> Full duplex double buffer (when FIFO is not used) Transmit/Receive FIFO (up to 128×9 bits each) ^{*1} (if FIFO is used)
2	Transfer system	<ul style="list-style-type: none"> Clock synchronization (without start/stop bit) Master/slave function SPI supported (for both master and slave modes)
3	Baud rate	<ul style="list-style-type: none"> Dedicate baud rate generator provided (configured with a 15-bit reload counter; in master mode operation) An external clock can be entered (in the slave mode operation).
4	Data length	Variable from 5 bits to 9 bits
5	Receive error detection	Overrun error
6	Interrupt request	<ul style="list-style-type: none"> Receive interrupt (a receive completion, an overrun error) Transmit interrupt (a transmit data empty, a transmit bus idle) Transmit FIFO interrupt (when transmit FIFO is empty) Extended intelligent I/O service (EIIOS) and DMA transport support functions provided for both transmission and reception
7	Sync mode	Master or slave function
8	Pin access	The serial data output pin can be set to "1".
9	FIFO options	<ul style="list-style-type: none"> FIFO for transmit/receive installed (maximum capacity: 128×9 bits for transmit FIFO, 128×9 bits for receive FIFO) ^{*1} Transmit FIFO or receive FIFO can be selected. Transmit data can be resent. Receive FIFO interrupt timing can be changed via software. FIFO resetting is supported independently.

*1 The FIFO capacity depends on the package type.

2. CSIO (Clock Sync Serial Interface) interrupts

The CSIO interrupts contain the receive interrupt and the transmit interrupt. These interrupt requests can be generated if:

- A receive data is set in the Receive Data Register (RDR) or a data receive error occurs.
- A transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started
- The transmit bus is idle (No data transmission occurs).
- A transmit FIFO data is requested.

■ CSIO interrupts

Table 2-1 shows the CSIO interrupt control bits and the interrupt causes.

Table 2-1 CSIO interrupt control bits and interrupt causes

Interrupt type	Interrupt request flag bit	Flag register	Interrupt cause	Interrupt cause enable bit	Operation to clear interrupt request flag
Reception	RDRF	SSR	A single-byte reception	SCR:RIE	Reading from the received data register (RDR)
			Reception of a data volume matching the value set for FBYTE.		
			The FRIIE bit is "1", receive FIFO contains valid data, and the Receive Idle state continues more than 8 hours.		Reading from the Received Data Register (RDR) until receive FIFO is emptied
	ORE	SSR	Overrun error		Setting the Receive Error Flag Clear bit (SSR:REC) to "1"
Transmission	TDRE	SSR	The Transmit Data Register is empty.	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *1
	TBI	SSR	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *1
	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.

*1 Set the TIE bit to "1" only after the TDRE bit has been set to "0".

2.1. Receive interrupt and flag set timing

Data reception can be interrupted by a Receive Completion (SSR:RDRF) or a Receive Error Occurrence (SSR:ORE).

■ Receive interrupt and flag set timing

When the last data bit is detected, the received data is stored in the Receive Data Register (RDR). When the data reception is completed (SSR:RDRF=1) or when a data receive error occurs (SSR:ORE=1), each flag is set. If a receive interrupt is enabled (SSR:RIE=1) during this time, a receive interrupt occurs.

<Note>

If a receive error occurs, data in the Receive Data Register (RDR) is invalidated.

Figure 2-1 Data receiving and flag set timing

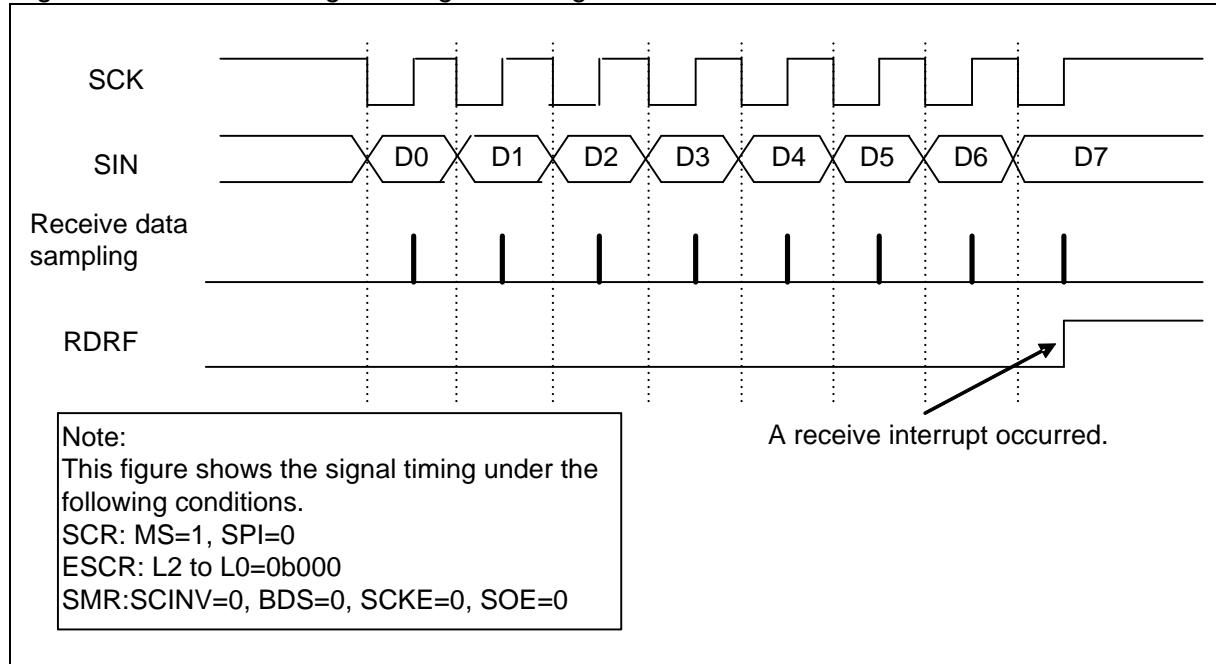
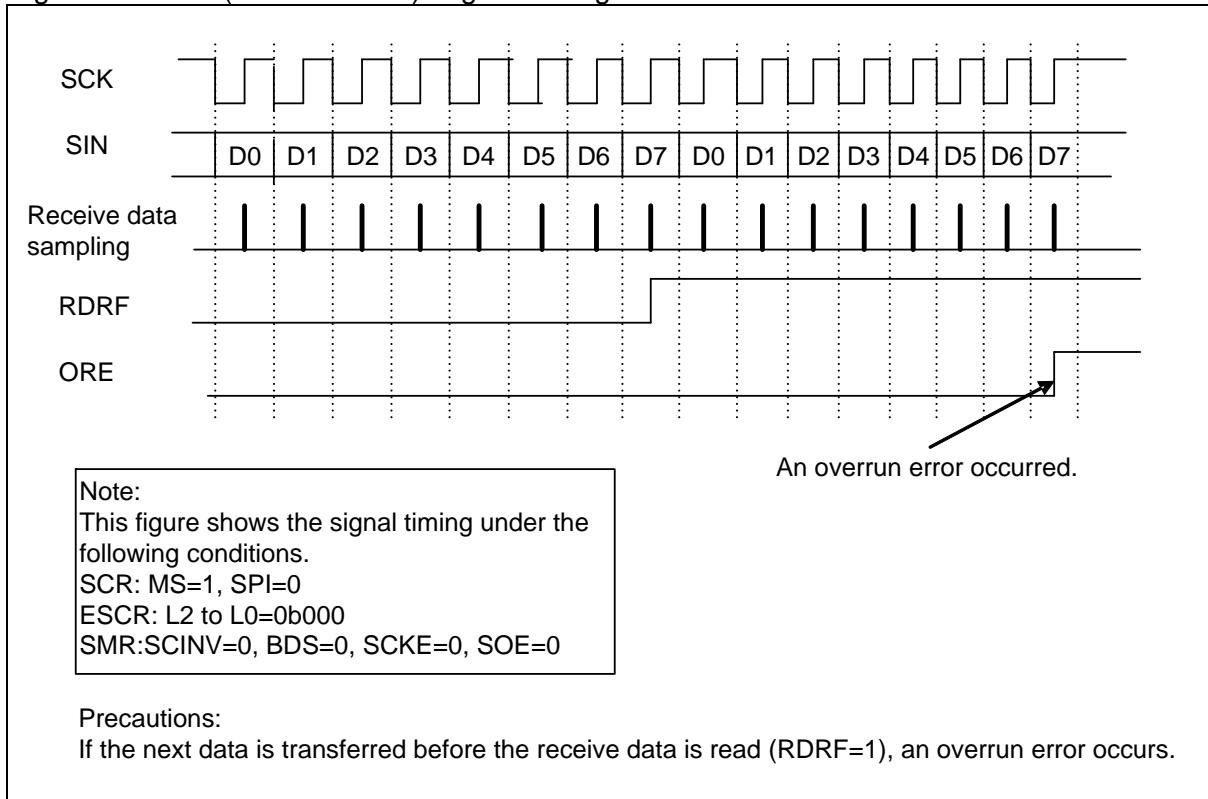


Figure 2-2 ORE (Overrun Error) flag set timing



2.2. Interrupt occurrence and flag set timing when receive FIFO is used

If receive FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register (FBYTE)) is received.

■ Receive interrupt and flag set timing when receive FIFO is used

If receive FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.

- When full FBYTE data is received, the receive data full flag (SSR:RDRF) of the Serial Status Register is set to "1". If a receive interrupt (SCR:RIE) is enabled during this time, a receive interrupt occurs.
- If all of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
 - The receive FIFO idle detect enable bit (FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to "0". If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- When the Receive Data Register (RDR) data is all read and receive FIFO is emptied, the receive data full flag (SSR:RDRF) is cleared.
- If the valid receive data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE=1) occurs.

Figure 2-3 Receive interrupt occurrence timing when receive FIFO is used

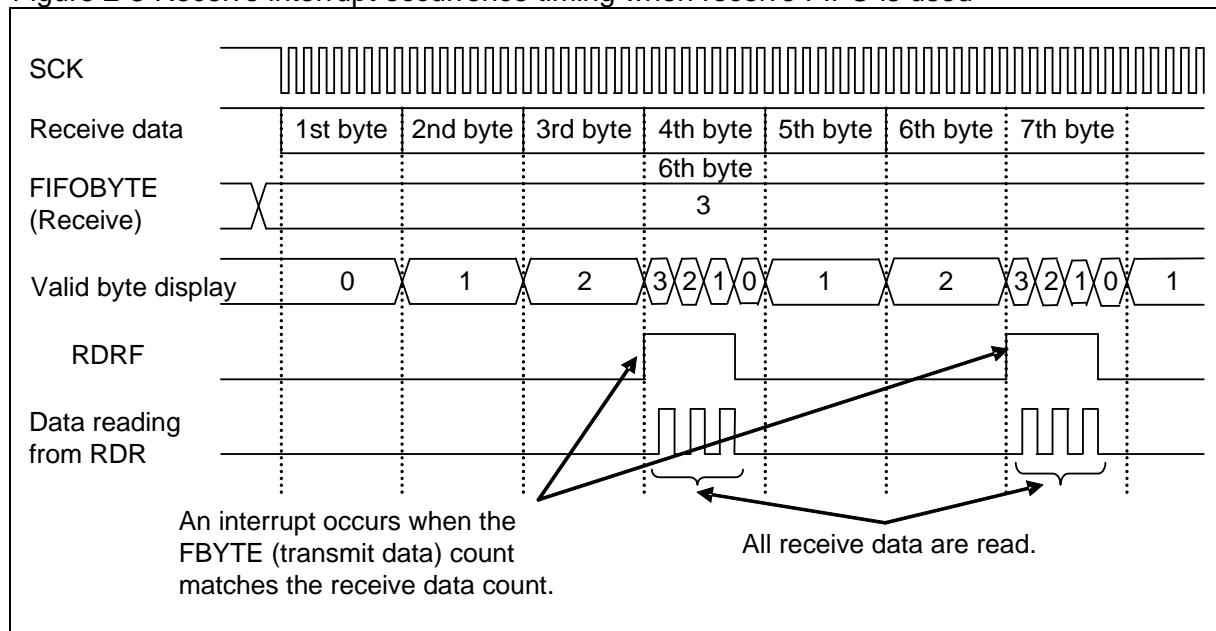
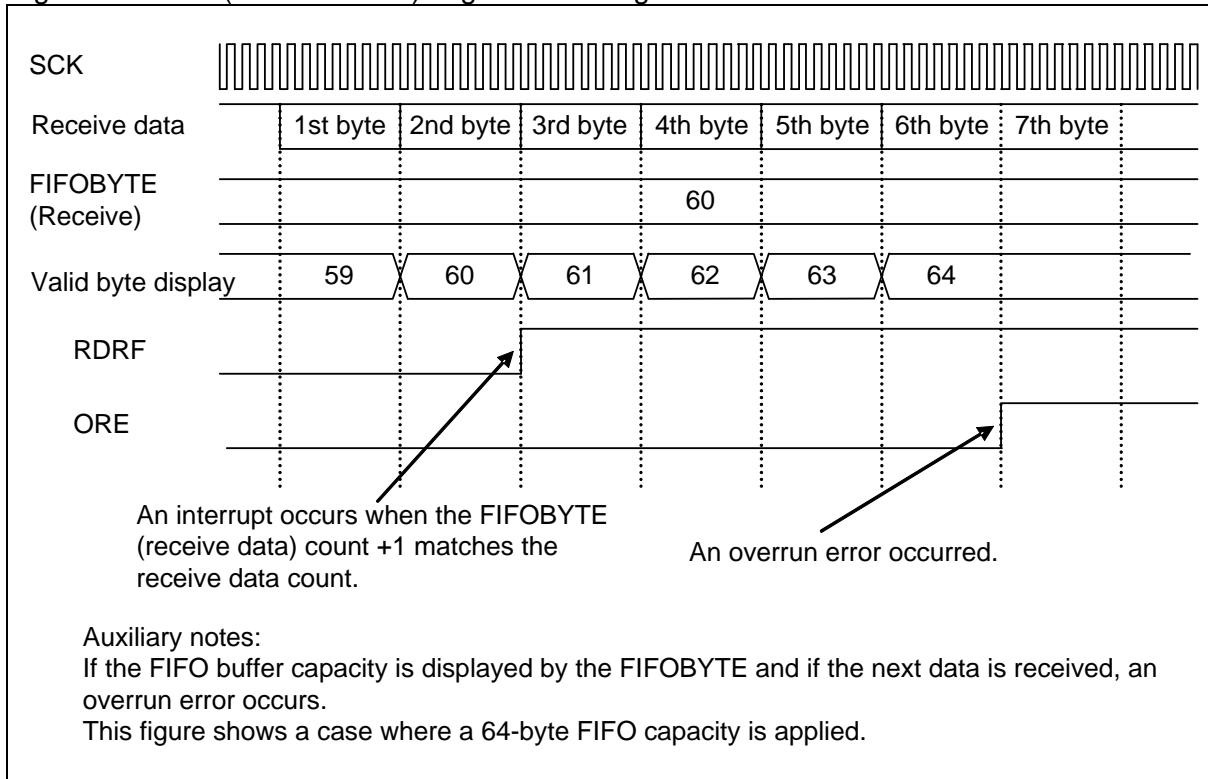


Figure 2-4 ORE (Overrun Error) flag bit set timing



2.3. Transmit interrupt occurrence and flag set timing

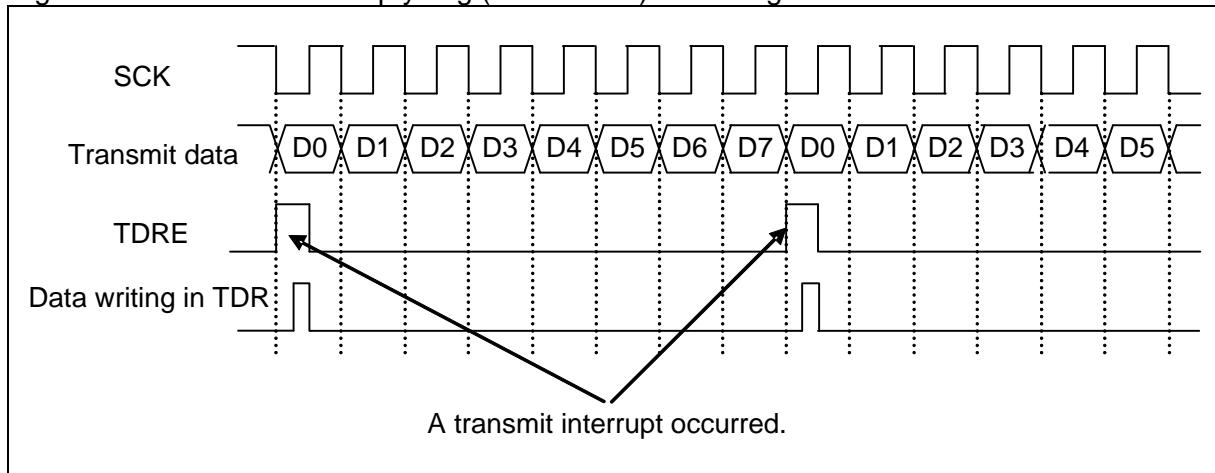
A transmit interrupt occurs if data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE=1) and it is transmitted, or if no data is transmitted (SSR:TBI=1).

■ Transmit interrupt occurrence and flag set timing

● Transmit data empty flag (SSR:TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written in the TDR (SSR:TDRE=1). If a transmit interrupt is enabled (SCR:TIE=1) during this time, a transmit interrupt occurs. As the SSR:TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

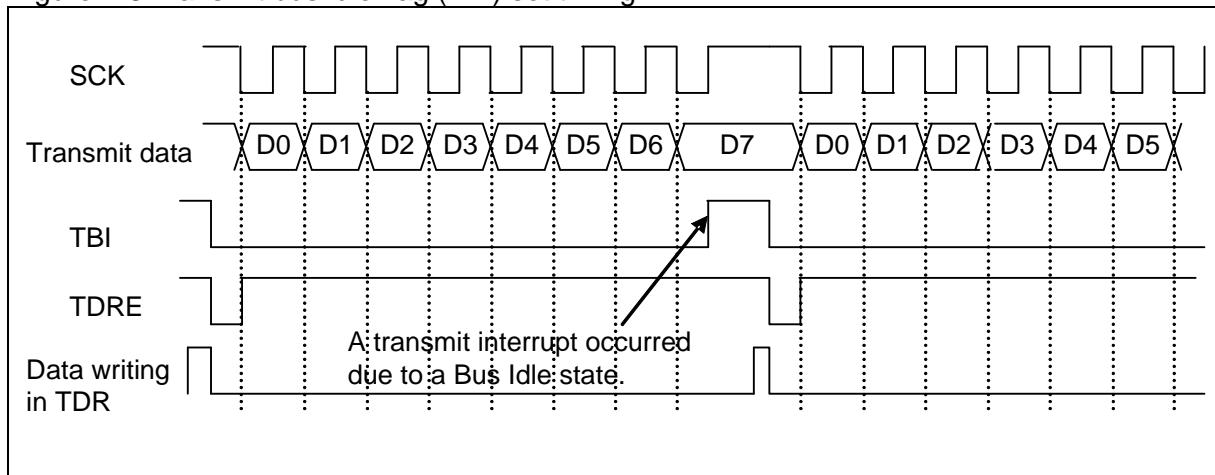
Figure 2-5 Transmit data empty flag (SSR:TDRE) set timing



● Transmit bus idle flag (SSR:TBI) set timing

If the Transmit Data Register is empty (SSR:TDRE=1) and no data is transmitted, the SSR:TBI bit is set to "1". If a transmit bus idle interrupt is enabled (SCR:TBIE=1) during this time, a transmit interrupt occurs. When transmit data is written to the Transmit Data Register (TDR), both the SSR:TBI bit and the transmit interrupt request are cleared.

Figure 2-6 Transmit bus idle flag (TBI) set timing



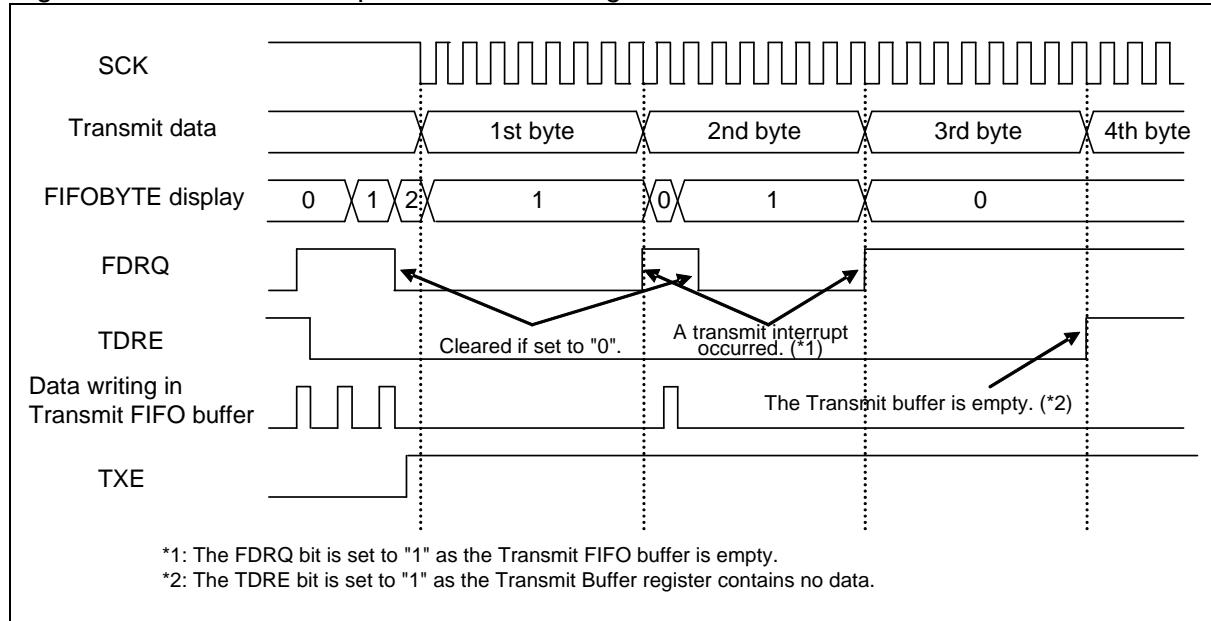
2.4. Interrupt occurrence and flag set timing when transmit FIFO is used

When transmit FIFO is used, an interrupt occurs if the buffer contains no data.

■ Transmit interrupt occurrence and flag set timing when transmit FIFO is used

- If transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1". If a FIFO transmit interrupt is enabled (FCR1:FTIE=1), a transmit interrupt occurs.
- If you have written the required data in transmit FIFO after occurrence of a transmit interrupt, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- When transmit FIFO is filled with data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "0".
- You can check a presence of data in transmit FIFO by reading the FIFO Byte Register (FBYTE). If FBYTE=0x00, no data exists in transmit FIFO.

Figure 2-7 Transmit interrupt occurrence timing when transmit FIFO is used



3. CSIO (Clock Sync Serial Interface) operations

The clock synchronous data transmission is used.

3.1. Normal transfer (I)

■ Features

	Item								Description							
1	Serial clock (SCK) signal detect level								"HIGH"							
2	Transmit data output timing								SCK signal falling edge							
3	Receive data sampling								SCK signal rising edge							
4	Data length								5 to 9 bits							

■ Register settings

The register values required for normal data transfer (I) are listed on the table below.

Table 3-1 Normal transfer (I) register settings

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE	
	0	1/0	0	*	*	*	*	*	0	1	0	0	0	*	1/0	*	
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0	
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*	
TDR/ RDR									D8	D7	D6	D5	D4	D3	D2	D1	D0
									*	*	*	*	*	*	*	*	*
BGR1/ BGR 0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

1 : Set to "1".

0 : Set to "0".

* : User-dependent values

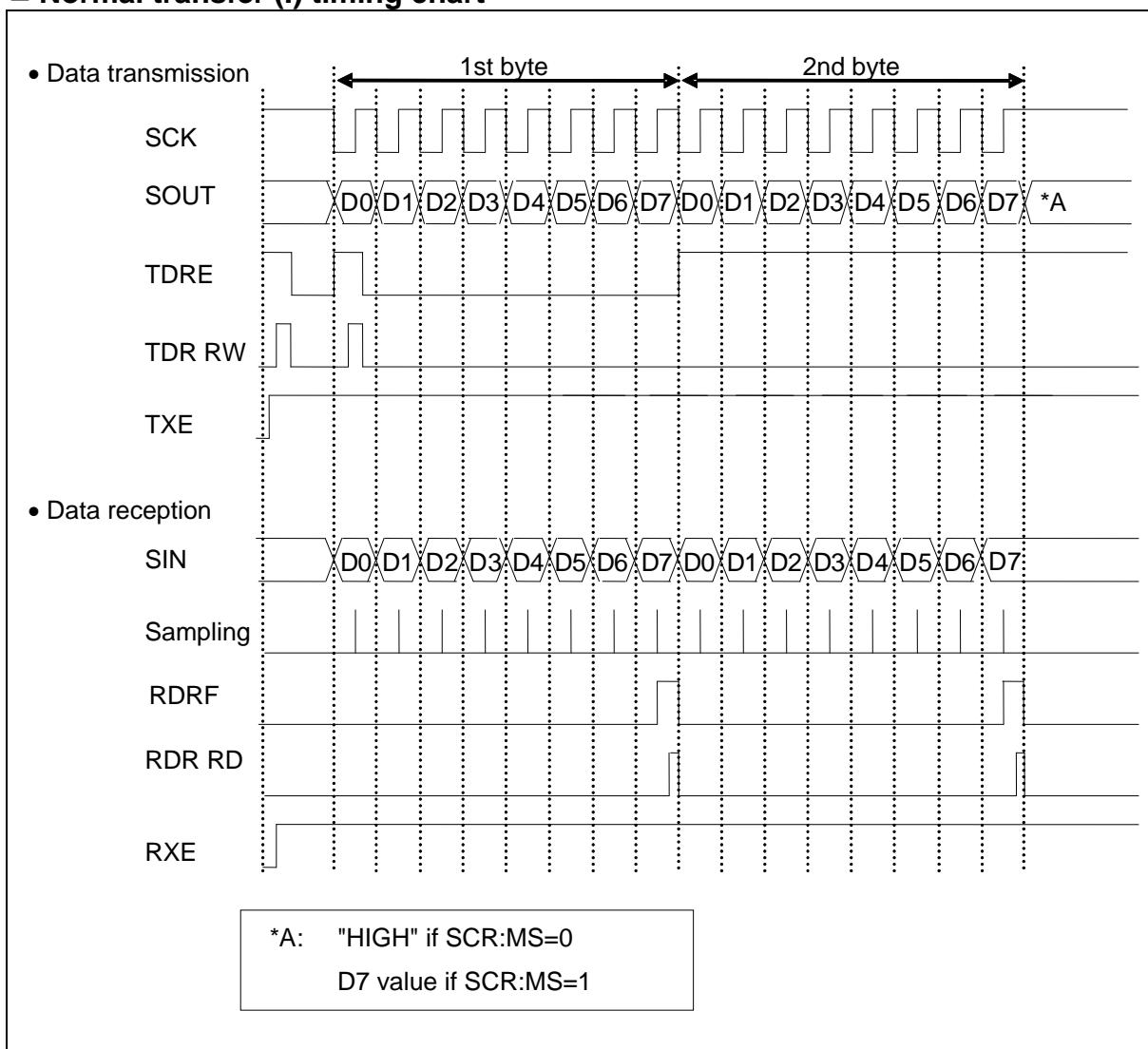
<Note>

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

During master mode operation: SCR:MS=0, SMR:SCKE=1

During slave mode operation: SCR:MS=1, SMR:SCKE=0

■ Normal transfer (I) timing chart



■ Master mode operation (SCR:MS=0, SMR:SCKE=1)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the receive data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1) during this time, a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

<Notes>

- To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.
- If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the transmit frames are set in the FBYTE register.

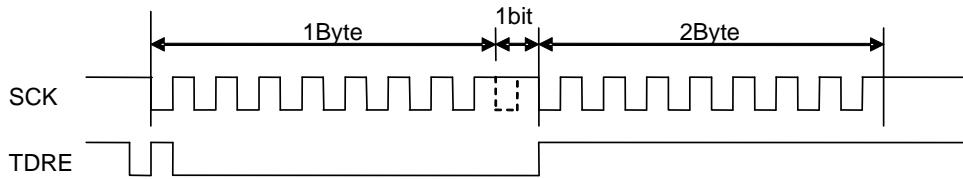
● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

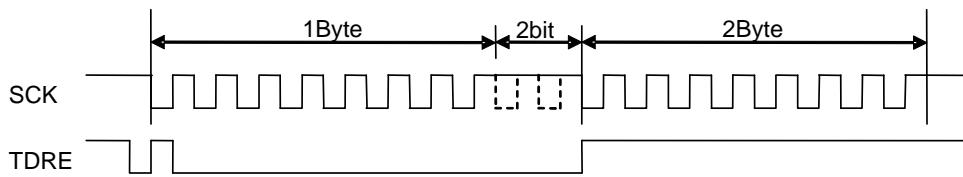
● **Continuous data transmit or receive waiting**

If anything other than (ESCR.WT1, ESCR.WT0)=(0, 0) is set for the continuous data transmission or reception, a wait is inserted between frames.

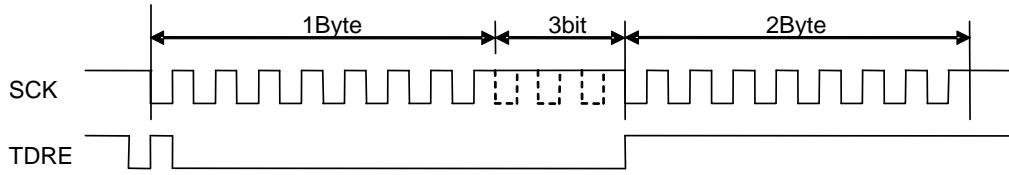
- ESCR.WT1=0, ESCR.WT0=1 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=0 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=1 (in master mode operation)



■ Slave mode operation (Set SCR:MS=1 and SMR:SCKE=0.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the receive data is sampled at a rising edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

3.2. Normal transfer (II)

■ Features

	Item	Description
1	Serial clock (SCK) signal detect level	"LOW"
2	Transmit data output timing	SCK signal rising edge
3	Receive data sampling	SCK signal falling edge
4	Data length	5 to 9 bits

■ Register settings

The register values required for normal data transfer (II) are listed on the table below.

Table 3-2 Normal transfer (II) register settings

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE	
	0	1/0	0	*	*	*	*	*	0	1	0	0	1	*	1/0	*	
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0	
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*	
TDR/ RDR									D8	D7	D6	D5	D4	D3	D2	D1	D0
									*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

1 : Set to "1".

0 : Set to "0".

* : User-dependent values

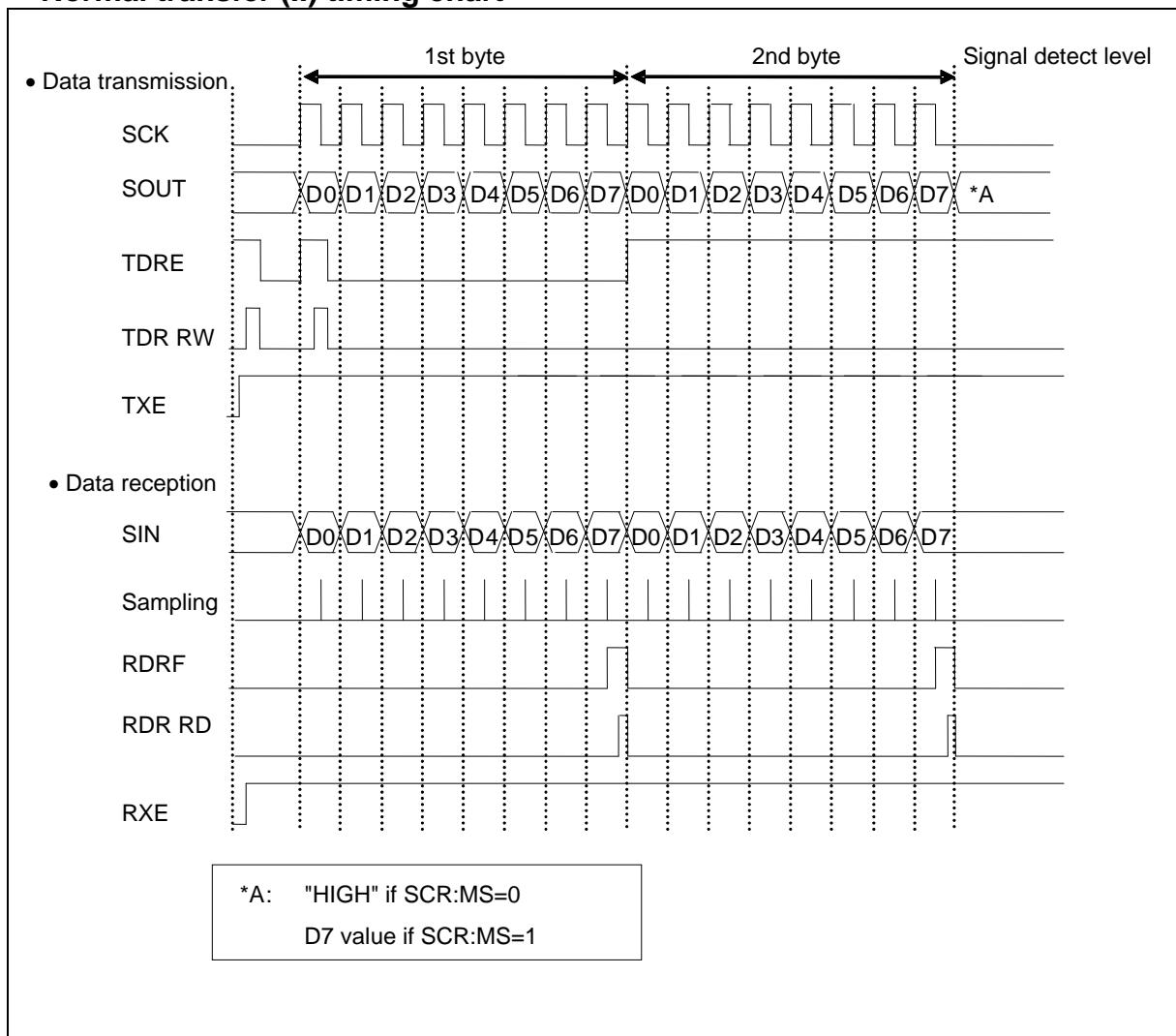
<Note>

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

During master mode operation: SCR:MS=0, SMR:SCKE=1

During slave mode operation: SCR:MS=1, SMR:SCKE=0

■ Normal transfer (II) timing chart



■ Master mode operation (Set SCR:MS=0 and SMR:SCKE=1.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the receive data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1) during this time, a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

<Notes>

- To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.
- If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the transmit frames are set in the FBYTE register.

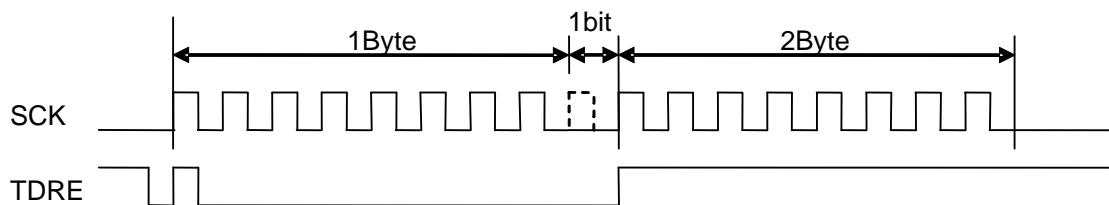
● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

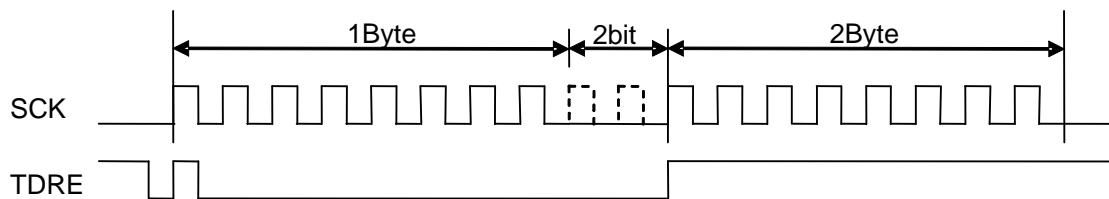
● **Continuous data transmit or receive waiting**

If anything other than (ESCR.WT1, ESCR.WT0)=(0, 0) is set for the continuous data transmission or reception, a wait is inserted between frames.

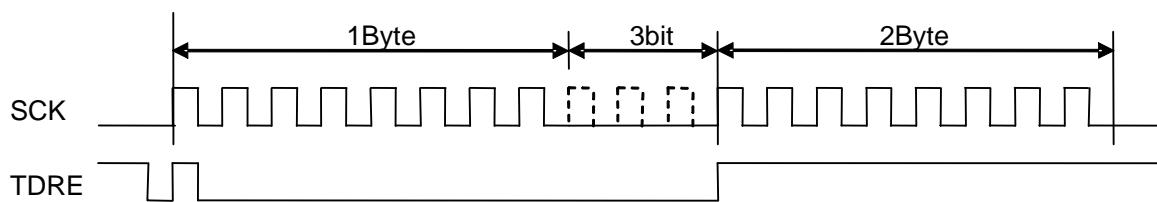
- ESCR.WT1=0, ESCR.WT0=1 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=0 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=1 (in master mode operation)



■ Slave mode operation (Set SCR:MS=1 and SMR:SCKE=0.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) input.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the receive data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

3.3. SPI transfer (I)

■ Features

	Item	Description
1	Serial clock (SCK) signal detect level	"HIGH"
2	Transmit data output timing	SCK signal rising edge
3	Receive data sampling	SCK signal falling edge
4	Data length	5 to 9 bits

■ Register settings

The register values required for SPI data transfer (I) are listed on the table below.

Table 3-3 SPI transfer (I) register settings

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE	
	0	1/0	1	*	*	*	*	*	0	1	0	0	0	*	1/0	*	
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0	
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*	
TDR/ RDR									D8	D7	D6	D5	D4	D3	D2	D1	D0
									*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

1 : Set to "1".

0 : Set to "0".

* : User-dependent values

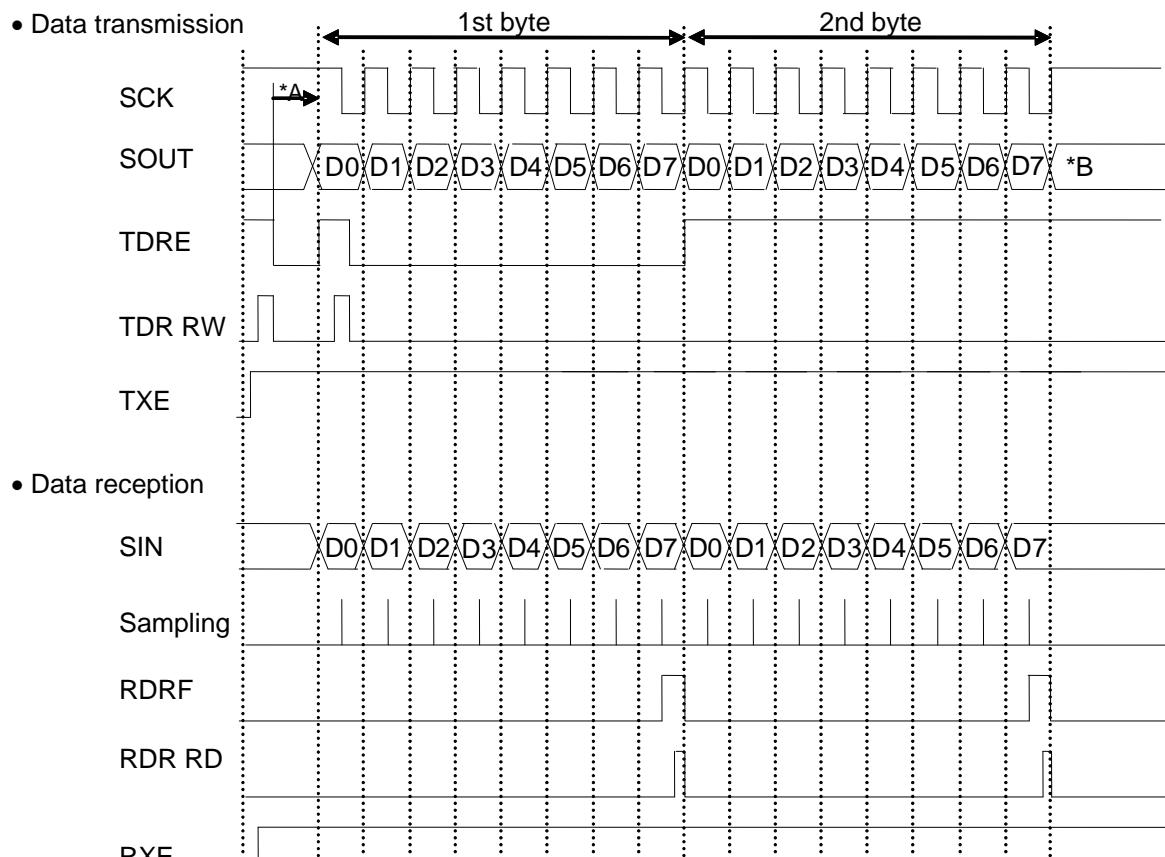
<Note>

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

During master mode operation: SCR:MS=0, SMR:SCKE=1

During slave mode operation: SCR:MS=1, SMR:SCKE=0

■ SPI transfer (I) timing chart



*A: During slave mode transmission(MS=1, SOE=1), 4 machine cycles or more time is required after writing data in the TDR

*B: "HIGH" if SCR:MS=0

"D0" of the 3rd byte if SCR:MS=1 and TDRE is "LOW"

"HIGH" if SCR:MS=1 and TDRE is "HIGH"

■ Master mode operation (Set SCR:MS=0 and SMR:SCKE=1.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the receive data is sampled at a falling edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1) during this time, a receive interrupt request is output. The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

<Notes>

- To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.
- If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the transmit frames are set in the FBYTE register.

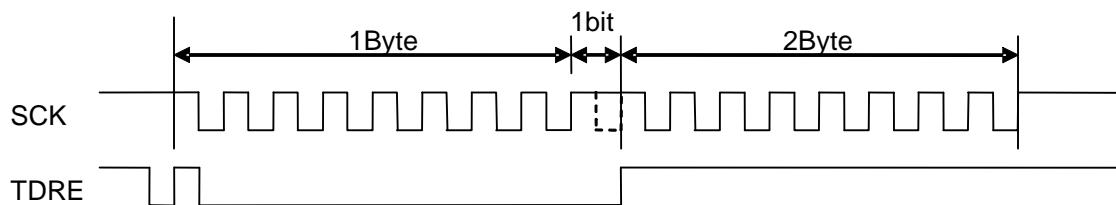
● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of the first serial clock. If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

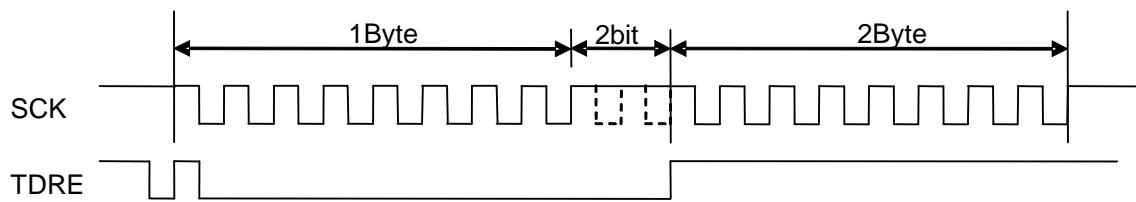
● **Continuous data transmit or receive waiting**

If anything other than (ESCR.WT1, ESCR.WT0)=(0, 0) is set for the continuous data transmission or reception, a wait is inserted between frames.

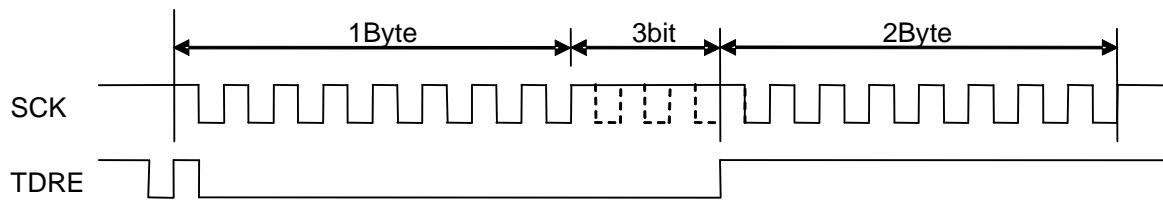
- ESCR.WT1=0, ESCR.WT0=1 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=0 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=1 (in master mode operation)



■ Slave mode operation (Set SCR:MS=1 and SMR:SCKE=0.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

<Note>

If data transmission is enabled (SCR:TXE=1) and if the first transmit data is written in the TDR at a time other than the serial clock (SCK) signal detect level, the first data bit is not output and the data transmission may fail. After the data transmission is enabled (SCR:TXE=1), the first data must be written in the TDR at a signal detect level of the serial clock (SCK).

● Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the receive data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

● Continuous switching from data reception to transmission

1. Disable the serial data output (SMR:SOE=0), enable a receive interrupt (SCR:RIE=1), enable data reception (SCR:RXE=1), and enable data transmission (SCR:TXE=1). If dummy data is written in the TDR at a signal detect level of serial clock (SCK), the receive data is sampled at a falling edge of serial clock (SCK) input.
2. To continue data reception, write a dummy data in the TDR between the time when a receive interrupt is requested and when the next serial clock (SCK) rises.
3. To switch the data reception to the data transmission, enable the serial data output (SMR:SOE=1), disable a receive interrupt (SCR:RIE=0), and disable data reception (SCR:RXE=0) between the time when a receive interrupt is requested and when the next serial clock (SCK) rises. Also, output the transmit data in synchronization with a rising edge of serial clock after the transmit data has been written in the TDR and the data reception has completed.

3.4. SPI transfer (II)

■ Features

	Item	Description
1	Serial clock (SCK) signal detect level	"LOW"
2	Transmit data output timing	SCK signal falling edge
3	Receive data sampling	SCK signal rising edge
4	Data length	5 to 9 bits

■ Register settings

The register values required for SPI data transfer (II) are listed on the table below.

Table 3-4 SPI transfer (II) register settings

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE	
	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	*	
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0	
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*	
TDR/ RDR									D8	D7	D6	D5	D4	D3	D2	D1	D0
									*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

1 : Set to "1".

0 : Set to "0".

* : User-dependent values

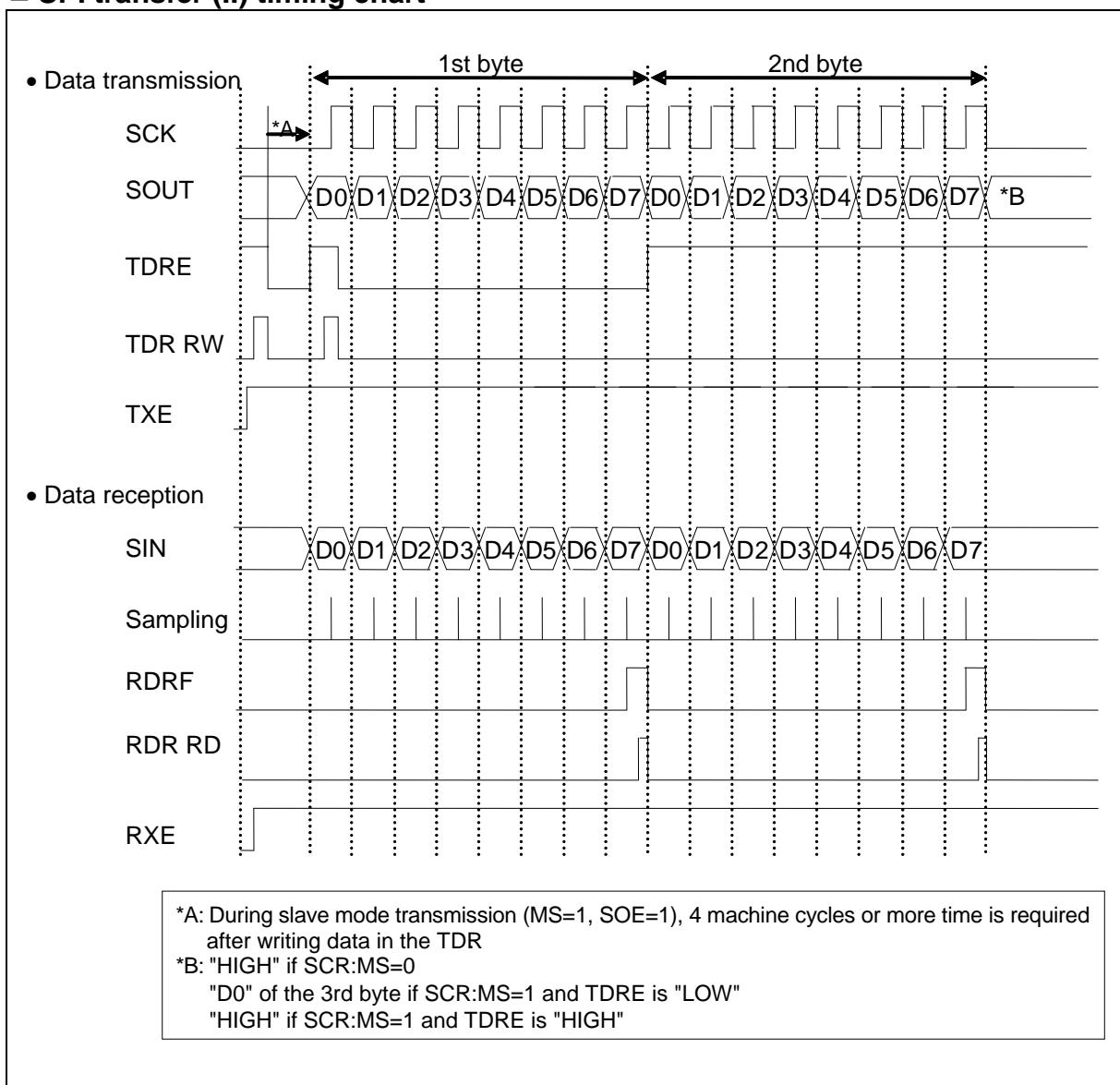
<Note>

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

During master mode operation: SCR:MS=0, SMR:SCKE=1

During slave mode operation: SCR:MS=1, SMR:SCKE=0

■ SPI transfer (II) timing chart



■ Master mode operation (Set SCR:MS=0 and SMR:SCKE=1.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
2. The SSR:TDRE bit is set to "1" before a half cycle of a rising edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

● Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the receive data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1) during this time, a receive interrupt request is output. The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

<Notes>

- To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.
- If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the transmit frames are set in the FBYTE register.

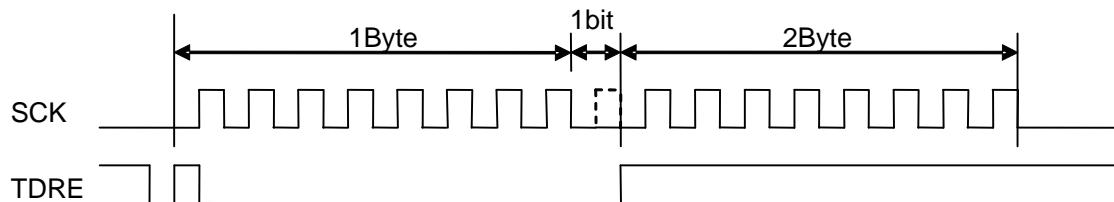
● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a rising edge of the first serial clock. If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

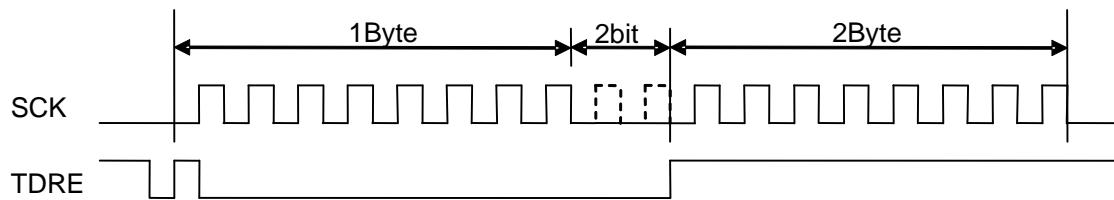
● **Continuous data transmit or receive waiting**

If anything other than (ESCR:WT1, ESCR.WT0)=(0, 0) is set for the continuous data transmission or reception, a wait is inserted between frames.

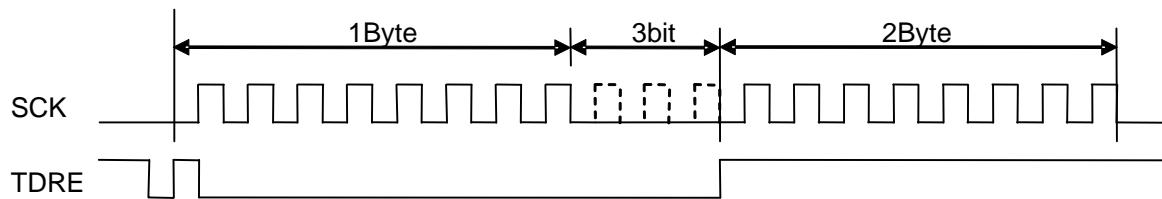
- ESCR.WT1=0, ESCR.WT0=1 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=0 (in master mode operation)



- ESCR.WT1=1, ESCR.WT0=1 (in master mode operation)



■ Slave mode operation (Set SCR:MS=1 and SMR:SCKE=0.)

● Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

<Note>

If data transmission is enabled (SCR:TXE=1) and if the first transmit data is written in the TDR at a time other than the serial clock (SCK) signal detect level, the first data bit is not output and the data transmission may fail. After the data transmission is enabled (SCR:TXE=1), the first data must be written in the TDR at a signal detect level of the serial clock (SCK).

● Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the receive data is sampled at a rising edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output.
The receive data (RDR) can be read during this time.
3. When the receive data (RDR) is read, the SSR:RDRF bit is cleared to "0".

● Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The receive data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of receive data is received, the SSR:RDRF bit is set to "1". If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt request is output. The receive data (RDR) can be read during this time. When the receive data is read, the SSR:RDRF bit is cleared to "0".

● Continuous switching from data reception to transmission

1. Disable the serial data output (SMR:SOE=0), enable a receive interrupt (SCR:RIE=1), enable data reception (SCR:RXE=1), and enable data transmission (SCR:TXE=1). If dummy data is written in the TDR at a signal detect level of serial clock (SCK), the receive data is sampled at a falling edge of serial clock (SCK) input.
2. To continue data reception, write a dummy data in the TDR between the time when a receive interrupt is requested and when the next serial clock (SCK) rises.
3. To switch the data reception to the data transmission, enable the serial data output (SMR:SOE=1), disable a receive interrupt (SCR:RIE=0), and disable data reception (SCR:RXE=0) between the time when a receive interrupt is requested and when the next serial clock (SCK) rises. Also, output the transmit data in synchronization with a rising edge of serial clock after the transmit data has been written in the TDR and the data reception has completed.

4. Dedicated baud rate generator

The dedicated baud rate generator functions in the master mode operation only. However, if receive FIFO is used, set the dedicated baud rate generator in the slave mode operation, too.

■ CSIO (Clock Sync Serial Interface) baud rate selection

The dedicated baud rate generator settings vary depending on the master or slave mode operation.

[1] During master mode operation

- **Divide the internal clock frequency using the dedicated baud rate generator, and select a baud rate.**

- This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
- The internal clock frequency is divided by the reload counter set value.

[2] During slave mode operation

- The dedicated baud rate generator does not function in the slave mode operation (SCR:MS=1).
(An external clock, entered from the SCK clock input pin, is used directly.)

<Note>

If receive FIFO is used, set the dedicated baud rate generator even in the slave mode operation.

4.1. Baud rate settings

This section explains how to set the baud rate. Also, the calculation result of serial clock frequency is shown.

■ Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$$V = \phi / b - 1$$

V : Reload value; b : Baud rate; ϕ : Bus clock frequency

(2) Calculation example

To set the 16MHz bus clock, use the internal clock, and set the 19200-bps baud rate, set the reload value as follows.

Reload value:

$$V = (16 \times 1000000)/19200 - 1 = 832$$

Therefore, the baud rate is:

$$b = (16 \times 1000000)/(832 + 1) = 19208\text{bps}$$

(3) Baud rate error

The baud rate error can be calculated by the following equation.

$$\text{Error (\%)} = (\text{Calculated value} - \text{Target value})/\text{Target value} \times 100$$

Example: To set the 20MHz bus clock and 153600-bps target baud rate:

$$\text{Reload value} = (20 \times 1000000)/153600 - 1 = 129$$

$$\text{Baud rate (Calculated value)} = (20 \times 1000000)/(129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error (\%)} = (153846 - 153600)/153600 \times 100 = 0.16 \text{ (\%)}$$

<Notes>

- If the reload value is set to "0", the reload counter is stopped.
- If the reload value is even, the "HIGH" and "LOW" width of serial clock are as follows. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.
If SMR:SCINV="0", the "HIGH" width of serial clock is longer for 1 cycle of bus clock.
If SMR:SCINV="1", the "LOW" width of serial clock is longer for 1 cycle of bus clock.
- Set the reload value to 3 or more.

■ Reload values and baud rates for each bus clock frequency

Table 4-1 Reload values and baud rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	-	-	-	-	-	-	-	-	-	-	3	0
6 M	-	-	-	-	-	-	-	-	3	0	-	-
5 M	-	-	-	-	-	-	3	0	-	-	-	-
4 M	-	-	-	-	3	0	4	0	5	0	7	0
2.5 M	-	-	3	0	-	-	-	-	-	-	-	-
2 M	3	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

- Value: BGR1/0 register set value

- ERR: Baud rate error (%)

■ Functions of reload counter

There are two types of reload counter: the transmit reload counter and the receive reload counter. They function as the dedicated baud rate generators. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from internal clocks.

■ Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

■ Restarting

The reload counter restarts counting in the following conditions.

● Common to transmit and receive reload counters

- A programmable reset (SCR:UPCL bit)

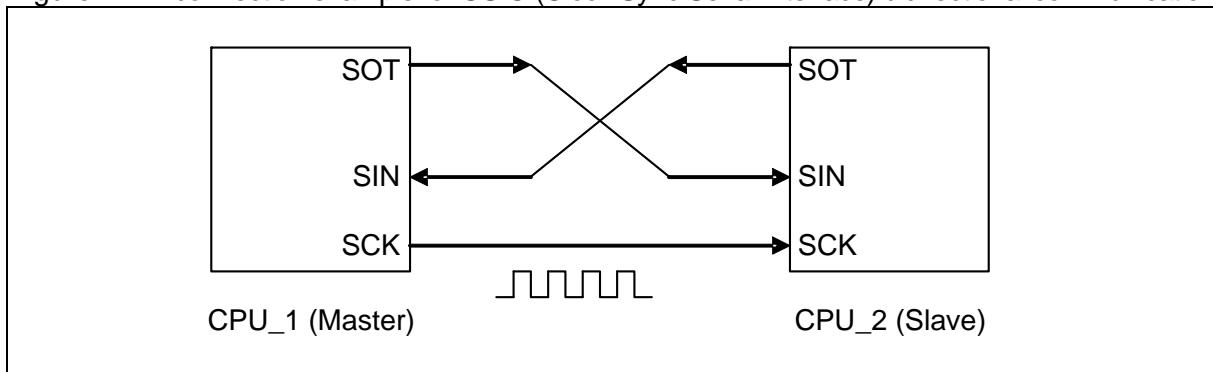
4.2. CSIO (Clock Sync Serial Interface) setup procedure and program flow

The CSIO (Clock Sync Serial Interface) allows bidirectional and synchronous serial data transmission.

● CPU-to-CPU connection

Select the bidirectional communication for the CSIO (Clock Sync Serial Interface). Connect two CPUs to each other as shown in Figure 4-1.

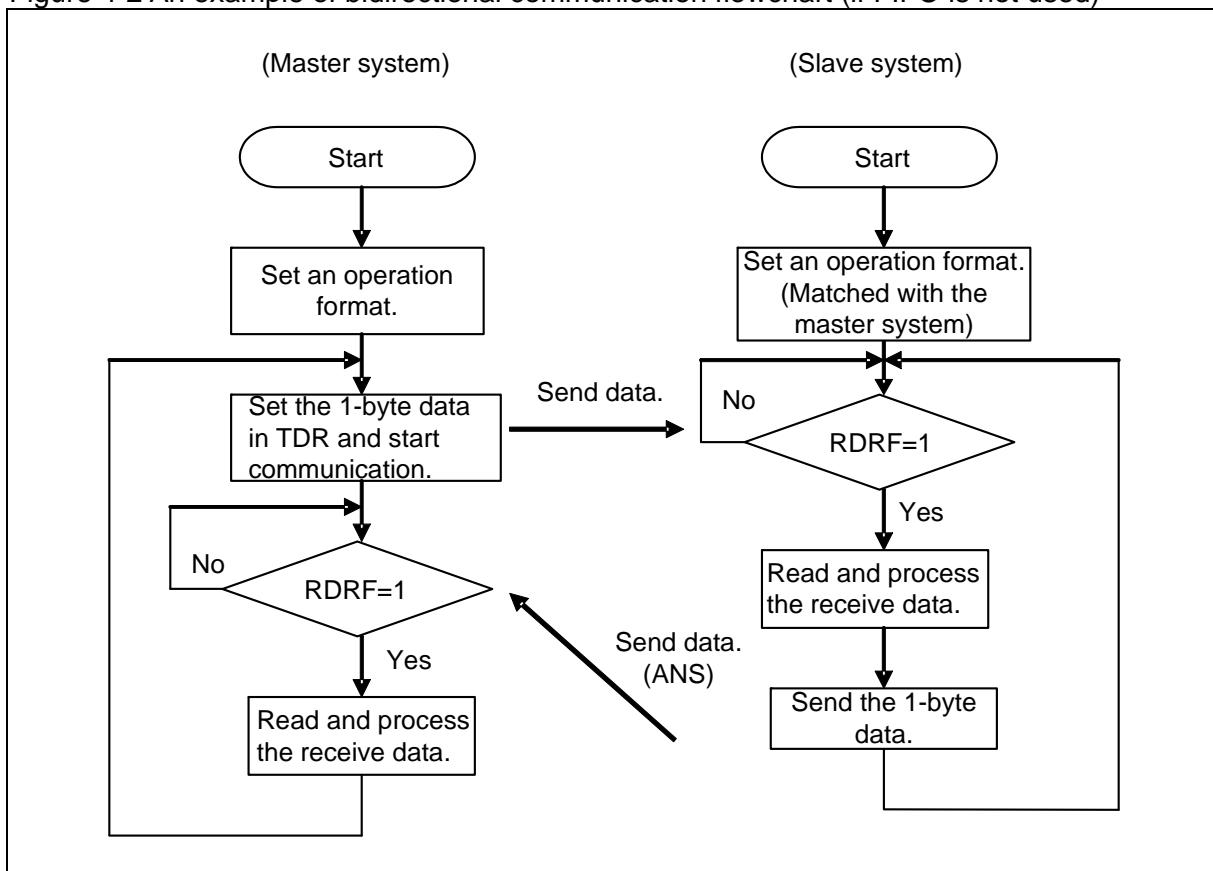
Figure 4-1 A connection example for CSIO (Clock Sync Serial Interface) bidirectional communication



■ Flowcharts

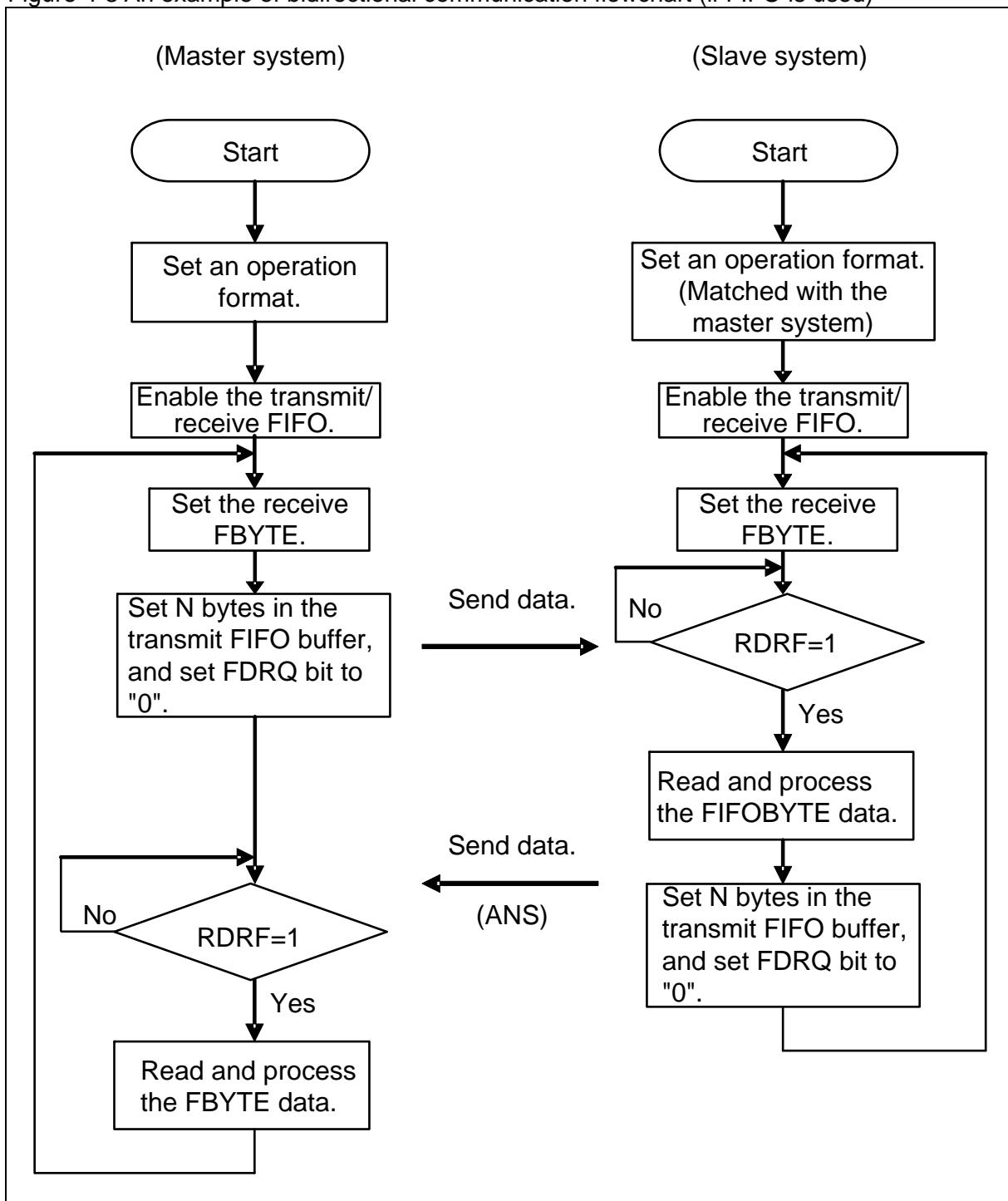
● If FIFO is not used

Figure 4-2 An example of bidirectional communication flowchart (if FIFO is not used)



● If FIFO is used

Figure 4-3 An example of bidirectional communication flowchart (if FIFO is used)



5. CSIO (Clock Sync Serial Interface) registers

This section provides a list of CSIO (Clock Sync Serial Interface) registers.

■ CSIO (Clock Sync Serial Interface) register list

Table 5-1 CSIO (Clock Sync Serial Interface) register list

	bit 15	bit 8	bit 7	bit 0
CSIO	SCR (Serial Control Register)		SMR (Serial Mode Register)	
	SSR (Serial Status Register)		ESCR (Extended Communication Control Register)	
	RDR1/TDR1 (Transmit/Receive Data register 1)		RDR0/TDR0 (Transmit/Receive Data register 0)	
	BGR1 (Baud Rate Generator Register 1)		BGR0 (Baud Rate Generator Register 0)	
	-		-	
FIFO	FCR1 (FIFO Control Register 1)		FCR0 (FIFO Control Register 0)	
	FBYTE2 (FIFO2 Byte Register)		FBYTE1 (FIFO1 Byte Register)	

Table 5-2 CSIO (Clock Sync Serial Interface) bit assignment

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
TDR/ RDR	-	-	-	-	-	-	-	D8	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

5.1. Serial Control Register (SCR)

The Serial Control Register (SCR) is used to enable/disable a transmit/receive interrupt, enable/disable a transmit idle interrupt, and enable/disable data transmission and reception. Also, the register can set the SPI connection and reset the CSIO settings.

Bit	15	14	13	12	11	10	9	8	7	...	0
Field	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	(SMR)		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Initial value	0	0	0	0	0	0	0	0			

[bit 15] UPCL: Programmable clear bit

Initializes the CSIO internal state.

If set to "1":

- The CSIO is reset directly (software reset). However, the current register settings are kept. The transmit or receive state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/receive interrupt causes (SSR:TDRE, TBI, RDRF, ORE) are cleared.

If set to "0":

No operation is affected.

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Programmable clear	

<Notes>

- Disable an interrupt first, and then execute the programmable clear instruction.
- If the FIFO operation is used, disable it (FCR0:FE2, FE1=0) first and then execute the programmable clear instruction.

[bit 14] MS: Master/Slave function select bit

Selects the master or slave mode.

- If set to "0": The master mode is selected.
- If set to "1": The slave mode is selected.

Bit	Description
0	Master mode
1	Slave mode

<Notes>

- If the slave mode is selected and if SMR:SCKE=0, the external clock is entered directly.
- After you have set the MS bit, enable data reception (RXE=1).

[bit 13] SPI: SPI corresponding bit

This bit allows the SPI communication.

- If set to "0": Normal synchronous communication is started.
- If set to "1": The SPI is corresponded.

Bit	Description
0	Normal Sync transfer
1	SPI correspond

[bit 12] RIE: Receive interrupt enable bit

- This bit enables or disables an output of receive interrupt request to the CPU.
- If the RIE bit and the receive data flag bit (SSR:RDRF) are "1", or if any of error flag bits (ORE) is "1", a receive interrupt request is output.

Bit	Description
0	Disables the receive interrupt.
1	Enables the receive interrupt.

[bit 11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of Transmit Interrupt Request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a Transmit Interrupt Request is output.

Bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit 10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and SSR:TBI bit are "1", a transmit bus idle interrupt request is output.

Bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit 9] RXE: Data receive enable bit

Enables or disables a CSIO data reception.

- If set to "0": The data frame reception is disabled.
- If set to "1": The data frame reception is enabled.

Bit	Description
0	Disables data reception.
1	Enables data reception.

<Notes>

- If data reception is disabled (RXE=0), the current data reception is stopped immediately.
- After you have set the MS bit and SMR:SCINV bit, enable the data reception (RXE=1).

[bit 8] TXE: Data transmission enable bit

Enables or disables a CSIO data transmission.

- If set to "0": The data frame transmission is disabled.
- If set to "1": The data frame transmission is enabled.

Bit	Description
0	Disables the transmission.
1	Enables the transmission.

<Note>

If data transmission is disabled (TXE=0), the current data transmission is stopped immediately.

5.2. Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to select an operation mode, to set a transmission direction, data length and serial clock inversion, and to enable or disable an output of serial data and clock to their pins.

Bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SCR)		MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE	
Attribute			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value			0	0	0	0	0	0	0	0	0

[bit 7:5] MD2, MD1, MD0: Operation mode set bits

These bits set an operation mode.

- "0b000": Sets operation mode 0 (async normal mode).
- "0b001": Sets operation mode 1 (async multiprocessor mode).
- "0b010": Sets operation mode 2 (clock sync mode).
- "0b011": Sets operation mode 3 (LIN communication mode).
- "0b100": Sets operation mode 4 (I²C mode).

This section explains the registers and their operation in operation mode 2 (clock sync mode).

Bit 7	Bit 6	Bit 5	Description
0	0	0	Operation mode 0 (async normal mode)
0	0	1	Operation mode 1 (async multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I ² C mode)

* This section explains the registers in operation mode 2.

<Notes>

- Any bit setting other than above is inhibited.
- To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.
- After the operation mode has been switched, set each register correctly.

[bit 4] WUCR: Wake-up control bit

Selects a pin to be used for an external interrupt.

If this bit is set to "0": The INT pin is set as an external interrupt pin.

If set to "1": The SCK pin is set as an external interrupt pin.

Bit	Description
0	Disables the Wake-up function.
1	Enables the Wake-up function.

[bit 3] SCINV: Serial clock invert bit

Inverts the serial clock format.

If set to "0":

- The signal detect level of serial clock output is set to "HIGH".
- The transmit data is output at a falling edge of serial clock during normal transfer, but it is output in synchronization with a rising edge of serial clock during SPI transfer.
- The receive data is sampled at a rising edge of serial clock during normal transfer, but it is sampled at a falling edge of serial clock during SPI transfer.

If set to "1":

- The signal detect level of serial clock output is set to "LOW".
- The transmit data is output at a rising edge of serial clock during normal transfer, but it is output in synchronization with a falling edge of serial clock during SPI transfer.
- The receive data is sampled at a falling edge of serial clock during normal transfer, but it is sampled at a rising edge of serial clock during SPI transfer.

Bit	Description
0	Signal level "HIGH" detection format
1	Signal level "LOW" detection format

<Notes>

- Always set this bit when transmission and reception are disabled (TXE=RXE=0).
- After you have set the SCINV bit, enable data reception (SCR:RXE=1).

[bit 2] BDS: Transfer direction select bit

Specifies to transmit the least significant bit of the transmit serial data first (LSB first; BDS=0) or the most significant bit first (MSB first; BDS=1).

Bit	Description
0	LSB first (The least significant bit is first transferred.)
1	MSB first (The most significant bit is first transferred.)

<Note>

Always set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).

[bit 1] SCKE: Master mode serial clock output enable bit

This bit controls the serial clock I/O port.

Bit	Description
0	Disables a serial clock output.
1	Enables a serial clock output.

<Note>

If this bit is used as the SCK pin, the GPIO must also be set.

[bit 0] SOE: Serial data output enable bit

This bit enables or disables a serial data output.

Bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

<Note>

If this bit is used as the SOUT pin, the GPIO must also be set.

5.3. Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmission/reception state, check the Receive Error flag, and clears the Receive Error flag.

Bit	15	14	13	12	11	10	9	8	7	...	0
Field	REC	-	-	-	ORE	RDRF	TDRE	TBI		(ESCR)	
Attribute	R/W	-	-	-	R	R	R	R			
Initial value	0	-	-	-	0	0	1	1			

[bit 15] REC: Receive error flag clear bit

This bit clears the ORE flag of the Serial Status Register (SSR).

- If this bit is set to "1", the error flag is cleared.
- This bit has no effect if set to "0".

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Clears the Receive Error flag (FRE, ORE).	

[bit 14:12] Reserved bits

This bit value is undefined when read.

This bit has no effect when written.

[bit 11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SCR:RIE bits are "1", a receive interrupt request is output.
- If this flag is set, data of the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit 10] RDRF: Receive data full flag bit

- This flag shows the state of Receive Data Register (RDR).
- When the receive data is loaded in the RDR, this bit is set to "1". When data is read from the Receive Data Register (RDR), this bit is cleared to "0".
- If the RDRF bit and SCR:RIE bit are "1", a receive interrupt request is output.
- If receive FIFO is used and if the preset amount of data is received in receive FIFO, the RDRF bit is set to "1".
- If receive FIFO is used, if both of the following conditions are satisfied, and if the Receive Idle state continues more than 8 baud rate clocks, the RDRF bit is set to "1".
 - The receive FIFO idle detect enable bit (FCR1:FRIIE) is "1".
 - The preset data amount is not received and some data remains in receive FIFO.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted.
- If the receive FIFO is used and if this buffer is emptied, this bit is cleared to "0".

Bit	Description
0	The Receive Data Register (RDR) is empty.
1	The Receive Data Register (RDR) contains data.

[bit 9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TDRE bit and SCR:TIE bit are "1", a transmit interrupt request is output.
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/reset timing when transmit FIFO is used, see "2.4 Interrupt occurrence and flag set timing when transmit FIFO is used".

Bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register (TDR) is empty.

[bit 8] TBI: Transmit bus idle flag bit

- This bit indicates that the CSIO is not transmitting data.
- When data is written in the Transmit Data Register (TDR), this bit is set to "0".
- If the Transmit Data Register (TDR) is empty (TDRE=1) and if no transmission is started, this bit is set to "1".
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- If this bit is "1" and if a transmit bus Idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

Bit	Description
0	During data transmission
1	No data transmission

5.4. Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to set a transmit/receive data length and to fix the serial data output to the "HIGH" state.

Bit	15	...	8	7	6	5	4	3	2	1	0
Field	-		SOP	-	-	WT1	WT0	L2	L1	L0	
Attribute		R/W		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0		-	-	0	0	0	0	0	0

[bit 7] SOP: Serial output pin set bit

- This bit sets the serial data output pin to the "HIGH" state. When this bit is set to "1", the SOUT pin is set to "HIGH". After that, this bit needs not be set to "0".
- When it is read, "0" is always read.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Sets the SOT pin to "HIGH" state.	

<Note>

Do not set this bit during serial data transmission.

[bit 6:5] Reserved bits

This bit value is undefined when read.
 This bit has no effect when written.

[bit 4:3] WT1, WT0: Data transmit/receive wait select bits

In master operation mode, these bits set a wait count for continuous data transmission or reception. In slave operation mode, these bits are set to "00".

- If set to "00": The SCK clocks are output continuously.
- If set to "01": The SCK clock is output after waiting for a single-bit time.
- If set to "10": The SCK clock is output after waiting for a two-bit time.
- If set to "11": The SCK clock is output after waiting for a three-bit time.

Bit 4	Bit 3	Description
0	0	0 bit
0	1	1 bit
1	0	2 bits
1	1	3 bits

[bit 2:0] L2, L1, L0: Data length select bits

These bits set a length of transmit/receive data.

- If set to "0b000": The 8-bit data length is set.
- If set to "0b001": The 5-bit data length is set.
- If set to "0b010": The 6-bit data length is set.
- If set to "0b011": The 7-bit data length is set.
- If set to "0b100": The 9-bit data length is set.

Bit 2	Bit 1	Bit 0	Description
0	0	0	8-bit length
0	0	1	5-bit length
0	1	0	6-bit length
0	1	1	7-bit length
1	0	0	9-bit length

<Note>

Any bit setting other than above is inhibited.

5.5. Receive Data Register/Transmit Data Register (RDR/TDR)

The Receive and Transmit Data Registers are allocated at the same address. This register functions as the Receive Data Register when data is read from it. This register operates as the Transmit Data Register when data is written in it.

■ Receive Data Register (RDR)

Bit	15	...	9	8	7	6	5	4	3	2	1	0
Field			D8	D7	D6	D5	D4	D3	D2	D1	D0	
Attribute			R	R	R	R	R	R	R	R	R	R
Initial value			0	0	0	0	0	0	0	0	0	0

The Receive Data Register (RDR) is a 9-bit data buffer register for serial data reception.

- When serial data signals are sent to the Serial Input pin (SIN pin), they are converted by a shift register and stored in the Receive Data Register (RDR).
- The high-order bits are sequentially set to "0" according to the data length as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

- When the received data is stored in the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is set to "1". If a receive interrupt is enabled (SSR:RIE=1), a receive interrupt request is generated.
- The Receive Data Register (RDR) must be read only when the receive data full flag bit (SSR:RDRF) is "1". When data is read from the Serial Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a receive error occurs (SSR:ORE), data in the Receive Data Register (RDR) is invalid.
- When the 9-bit length data is transferred, the RDR must be read in the 16-bit access mode.

<Notes>

- If the receive FIFO is used and if a certain count of data is received by the receive FIFO, the RDRF bit is set to "1".
- If receive FIFO is used and if this buffer is emptied, the RDRF bit is cleared to "0".
- If receive FIFO is used and if a receive error occurs (SSR:ORE), the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

■ Transmit Data Register (TDR)

Bit	15	...	9	8	7	6	5	4	3	2	1	0
Field			D8	D7	D6	D5	D4	D3	D2	D1	D0	
Attribute			W	W	W	W	W	W	W	W	W	W
Initial value			1	1	1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a 9-bit data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the transmit shift register. Then, the data is converted into serial data, and output at the serial data output pin (SOUT pin).
- The high-order bits are sequentially set to invalid according to the data length as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written in the buffer. If a transmit interrupt is enabled, a transmit interrupt occurs. The next transmit data must be written only after the transmit interrupt has occurred or when the transmit data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and if transmit FIFO is disabled or transmit FIFO is full, the transmit data cannot be written in the Transmit Data Register (TDR).
- When the 9-bit length data is transferred, data must be written in the TDR in the 16-bit access mode.

<Notes>

- The Transmit Data Register is a write-only register. While the Receive Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instructions cannot be used.
- For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt occurrence and flag set timing when transmit FIFO is used".

5.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-								(BGR1)							
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
Initial value	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- These bits set a clock frequency division in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
- The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.

[bit 15] Reserved bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 14:8] BGR1: Baud Rate Generator Register 1

Bit 14:8	Description
Write	Write data in reload counter bit 8 to 14.
Read	Reads the BGR1 set value.

[bit 7:0] BGR0: Baud Rate Generator Register 0

Bit 7:0	Description
Write	Write data in reload counter bit 0 to 7.
Read	Reads the BGR0 set value.

<Notes>

- Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) by 16-bit data accessing.
- If the reload value is even, the "HIGH" and "LOW" width of serial clock are as follows. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.
If SMR:SCINV="0", the "HIGH" width of serial clock is longer for 1 cycle of bus clock.
If SMR:SCINV="1", the "LOW" width of serial clock is longer for 1 cycle of bus clock.
- Set the reload value to 3 or more.
- If the current values of Baud Rate Generator Registers (BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/0 set values and execute the CSIO reset instruction (SCR:UPCL).
- If receive FIFO is used and if you wish to set the receive FIFO idle detect enable bit (FCR1:FRIIE) to "1" and starts the slave mode operation, set the desired baud rate in BGR1/0.

5.7. FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select the transmit or receive FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

Bit	15	14	13	12	11	10	9	8	7	...	0
Field	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL		(FCR0)	
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W			
Initial value	0	0	-	0	0	1	0	0			

[bit 15:14] FTST1, FTST0: FIFO test bits

They are FIFO Test bits.

They must always be set to "0".

Bit 15:14	Description
0	Disables the FIFO test.
1	Enables the FIFO test.

<Note>

If this bit is set to "1", the FIFO test is executed.

[bit 13] Reserved bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 12] FLSTE: Re-transmit data lost detect enable bit

This bit enables the FLST bit detection.

If set to "0": The FLST bit detection is disabled.

If set to "1": The FLST bit detection is enabled.

Bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

<Note>

If you wish to set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".

[bit 11] FRIIE: Receive FIFO idle detection enable bit

This bit sets to detect the receive idle state if the receive FIFO contains valid data and if it continues more than 8-bit hours. If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt is generated when the receive idle state is detected.

If set to "0": The receive idle state detection is disabled.

If set to "1": The receive idle state detection is enabled.

Bit	Description
0	Disables the receive FIFO idle detection.
1	Enables the receive FIFO idle detection.

<Note>

In case of using Receive FIFO, set this bit to "1".

[bit 10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. If the transmit FIFO interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

Bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

<Notes>

- If the FBYTE (for transmission) is "0", this bit cannot be set to "0".
- If this bit is "0", the FSEL bit state cannot be changed.
- If this bit is set to "1", it has no effect on the operation.
- If a read-modify-write instruction is issued, "1" is read.

[bit 9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

Bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit 8] FSEL: FIFO select bit

This bit selects the transmit or receive FIFO.

If set to "0": Set the transmit FIFO as FIFO1, and the receive FIFO as FIFO2.

If set to "1": Set the transmit FIFO as FIFO2, and the receive FIFO as FIFO1.

Bit	Description
0	Transmit FIFO:FIFO1; Receive FIFO:FIFO2
1	Transmit FIFO:FIFO2; Receive FIFO:FIFO1

<Notes>

- This bit is not cleared by FIFO reset (FCL2=1, FCL1=1).
 - To change this bit state, first disable the FIFO operation (FCR:OFE2=0, FE1=0).
-

5.8. FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

Bit	15	...	8	7	6	5	4	3	2	1	0	
Field	(FCR1)				-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute					-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value					0	0	0	0	0	0	0	0

[bit 7] Reserved bit

"0" is always read during reading.

"0" must always be written during writing.

[bit 6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- The FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in FIFO.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten.

Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

Bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit 5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.

When the re-transmission setting has finished, this bit is set to "0".

Bit	Description
0	Not reloaded
1	Reloaded

<Notes>

- If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.
- When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".
- After you have set the SCR:TIE bit and SCR:TBIE bit to "0", set this bit to "1". After you have enabled transmit FIFO, set the SCR:TIE bit and SCR:TBIE bit to "1".

[bit 4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer is saved before transmission and if the FLST bit is "0", data can be re-transmitted even when a communication error or others occur.

If set to "1": The current read pointer value is saved.

If set to "0": No effect.

Bit	Description	
	During writing	During reading
0	Not saved	"0" is always read.
1	Saved	

<Note>

This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".

[bit 3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

When this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST2 bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO2 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO2.
- Set the transmit FIFO interrupt enable bit to "0" before the execution.
- The valid data count of the FBYTE2 register is set to "0".

[bit 2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 state.

When this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST1 bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO1 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO1.
- Set the transmit FIFO interrupt enable bit to "0" before the execution.
- The valid data count of the FBYTE1 register is set to "0".

[bit 1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO (FCR1:FSEL=1) and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both SCR:TIE bit and SCR:TBIE bit to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO2 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

Bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit 0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If FIFO1 is set as transmit FIFO (FCR1:FSEL=0) and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both TIE bit and TBIE bit to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO1 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

Bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

5.9. FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field					(FBYTE2)							(FBYTE1)				
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count of FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 5-3 Display of data count

FSEL	FIFO selection	Byte count display
0	FIFO2: Receive FIFO, FIFO1: Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmit FIFO, FIFO1: Receive FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to flag a receive interrupt for the FBYTE register of receive FIFO. If this transfer data count matches the FBYTE register display, the interrupt flag (SSR:RDRF) is set to "1".
- If both conditions below are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to "0". If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- To receive data in the master mode operation (master mode reception), set both SCR:TIE and SCR:TBIE bits to "0", set the receive data count in the FBYTE register of transmit FIFO, and set the FCR1:FDRQ bit to "0". After that, when the SCR:TXE bit is "1", the serial clock is output for the preset data amount, and the preset amount of data can be received. Set the SCR:TIE bit and SCR:TBIE bit to "1" only after the FCR1:FDRQ bit has been set to "1".

FBYTE2, FBYTE1: FIFO2 data count display bit, FIFO1 data count display bit

During writing	Sets the transfer data count.
During reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in FIFO but not transmitted yet
 During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
 During reception: Set the data count to generate a receive interrupt.

<Notes>

- The FBYTE register of transmit FIFO must be "8'h00" except when data is received in the master mode operation.
 - During the master mode data reception, the transmit data count must be set only when transmit FIFO is empty and both SCR:TIE bit and SSR:TBIE bit are "0".
 - To disable the reception (SCR:RXE=0) when data is being received in the master mode operation, disable transmit FIFO first, and then disable the transmission and reception.
 - The FBYTE bit of receive FIFO must be set to "1" or larger.
 - Change the FBYTE data of receive FIFO only after you have disabled the data reception.
 - A read-modify-write instruction cannot be used for this register.
 - Any setting exceeding the FIFO capacity is inhibited.
 - When all the following requirements are met, the receive data full flag bit (SSR:RDRF) is not set to "1" even though the effective data of FBYTE setting number exist in the receive FIFO. If the FBYTE register is set to "2" or greater, this operation will not occur.
 - FBYTE is set to "1".
 - The effective data count is "1", same as the number specified in FBYTE register.
 - When the multi function serial interface macro receives the data, and writes received data in the reception FIFO, the data of the reception FIFO are read at the same time.
- However, after that, the receive data full flag bit (SSR:RDRF) will be set to "1" at any of the following conditions.
- The next data is received.
 - The receive idle state of 8 bits or longer is detected when the receive FIFO idle is enabled (FCR:FRIIE=1)

CHAPTER: LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)

This chapter explains the LIN communication function, a part of multifunctional serial interface functions and supported in Operation Mode 3.

1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)
2. LIN Interface (Ver. 2.1) Interrupts
3. Dedicated Baud Rate Generator
4. LIN Interface (Ver. 2.1) Operations
5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow
6. LIN Interface (ver. 2.1) Registers

1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)

The LIN interface (ver. 2.1) (LIN communication control interface ver. 2.1) supports functions complying with the LIN bus. It also has transmit/receive FIFO (up to 128×9 bits each)^{*1} installed.

■ Functions of LIN interface (ver. 2.1) (LIN communication control interface ver. 2.1)

		Function
1	Data buffer	<ul style="list-style-type: none"> Full duplex double buffer (when FIFO is not used) Transmit/receive FIFO ($max 128 \times 9$ bits each)^{*1} (when FIFO is used)
2	Serial input	Run oversampling three times with the bus clock and determine the value of received data based on the majority sampling value.
3	Transfer mode	Asynchronous
4	Baud rate	<ul style="list-style-type: none"> Complete with a dedicated baud rate generator (constructed with a 15-bit reload counter) The external clock can be adjusted with the reload counter.
5	Data length	8 bits
6	Signaling system	NRZ (Non Return to Zero)
7	Start bit detection	Synchronized with the falling edge of the start bit
8	Receive error detection	<ul style="list-style-type: none"> Framing error Overrun error
9	Interrupt request	<ul style="list-style-type: none"> Receive interrupts (reception completed, framing error, overrun error) Transmit interrupts (transmit data empty, transmit bus idle) Status interrupts (LIN break field detection) Interrupt request to ICU (LIN Sync field detection: LSYN) Transmit FIFO interrupt (when transmit FIFO is empty) For both transmission and reception, the extended intelligent I/O service (EIIOS) and the DMA function are available.
10	LIN bus option	<ul style="list-style-type: none"> Supports LIN Protocol Revision 2.1 Master device operations Slave device operations LIN break field generation (with variable bit length ranging from 13 to 16 bits) LIN break delimiter generation (with variable data length ranging from 1 to 4 bits) LIN break field detection Detection of LIN sync field start/stop edges connected to input capture
11	FIFO options	<ul style="list-style-type: none"> Transmit/receive FIFO installed (maximum capacity: 128×9 bits for transmit FIFO, 128×9 bits for receive FIFO)^{*1} Transmit FIFO or receive FIFO can be selected. Transmit data can be resent. Receive FIFO interrupt timing can be changed via software. FIFO resetting is supported independently.

*1: The FIFO capacity size varies from model type to model type.

2. LIN Interface (Ver. 2.1) Interrupts

Receive interrupts and transmit interrupts are provided for LIN interface (ver. 2.1). These interrupt requests can be generated if:

- Incoming data is set in the Receive Data Register (RDR) or a data receive error occurs.
- Outgoing data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started.
- The transmit bus is idle (No data transmission occurs).
- Transmit FIFO data is requested.
- A LIN break field is detected

■ LIN interface (ver. 2.1) interrupts

Table 2-1 shows the interrupt control bits and the interrupt causes of LIN interface (ver. 2.1).

Table 2-1 LIN interface (ver. 2.1) interrupt control bits and interrupt causes

Interrupt type	Interrupt request flag bit	Flag register	Interrupt cause	Interrupt cause enable bit	Operation to clear interrupt request flag
Reception	RDRF	SSR	A single-byte reception	SCR:RIE	Reading from the received data register (RDR)
			Reception of a data volume matching the value set for FBYTE.		
			While the FRIIE bit is "1" and the receive FIFO contains valid data, a receive idle state continues for 8 bits or longer period.		Reading from the Received Data Register (RDR) until receive FIFO is emptied
	ORE	SSR	Overrun error		Setting the Reception Error Flag Clear bit (SSR:REC) to "1"
	FRE	SSR	Framing error		
Transmission	TDRE	SSR	The Transmit Data Register is empty	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) ^{*1}
	TBI	SSR	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR), setting the LIN break field setting bit (LBR) to "1", or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data). ^{*1}
	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.
Status	LBD	SSR	LIN break field is detected	ESCR:LBI	The SSR:LBD bit is set to "0".
Input capture	ICP0	ICS0	The first rising edge in the LIN Sync field	ICS0:ICE0	Disables ICP0
	ICP0	ICS0	The fifth falling edge in the LIN Sync field		

*1: Set the TIE bit to "1" only after the TDRE bit has been set to "0".

2.1. Receive interrupt and flag set timing

Data reception can be interrupted by a receive completion (SSR:RDRF), a receive error occurrence (SSR:ORE, FRE), or a LIN break field detection.

■ Receive interrupt and flag set timing

Upon detection of the first stop bit, received data are stored in the Receive Data Register (RDR). When the data reception is completed (SSR:RDRF = 1) or when a data receive error occurs (SSR:ORE, FRE = 1), each flag is set. If receive interrupts are enabled (SSR:RIE = 1) during this time, a receive interrupt occurs.

<Note>

If a receive error occurs, data in the Receive Data Register (RDR) is invalidated.

Figure 2-1 RDRF (Receive Data Register Full) flag bit set timing

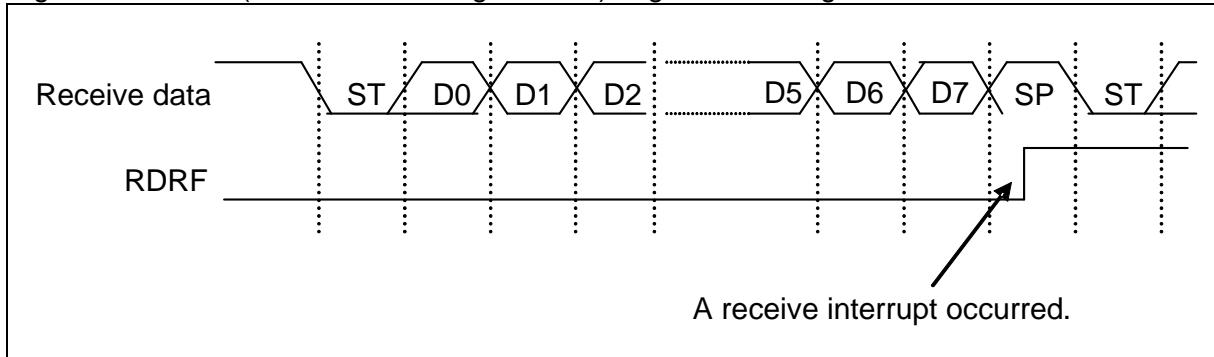
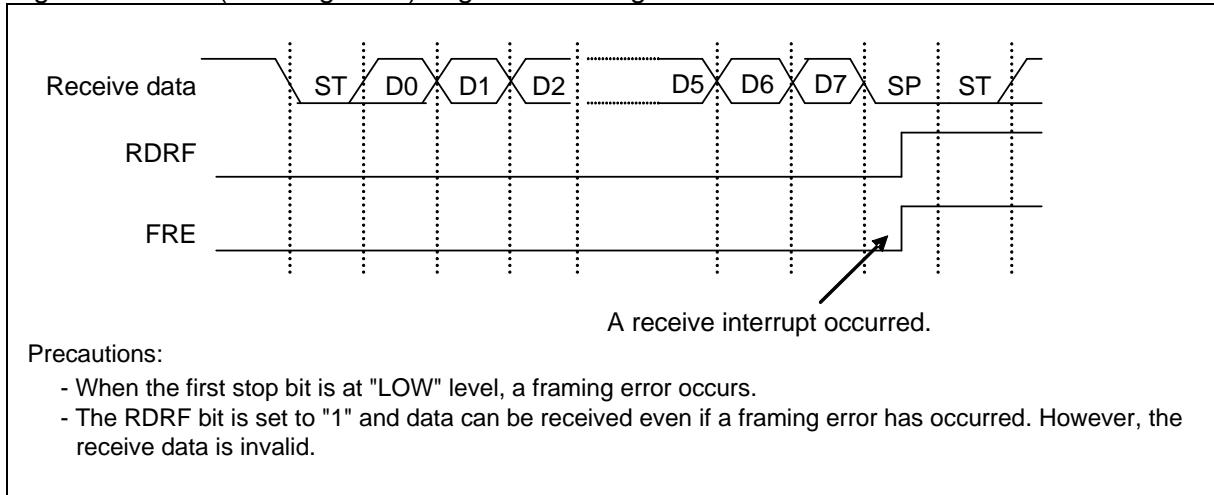


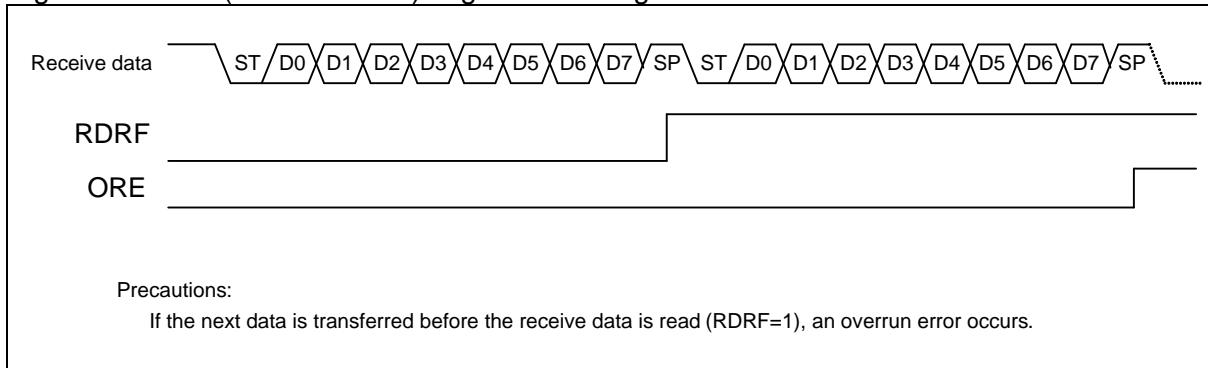
Figure 2-2 FRE (Framing Error) flag bit set timing



<Note>

During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data may not be received successfully. To output frames continuously, adequate intervals are required between frames.

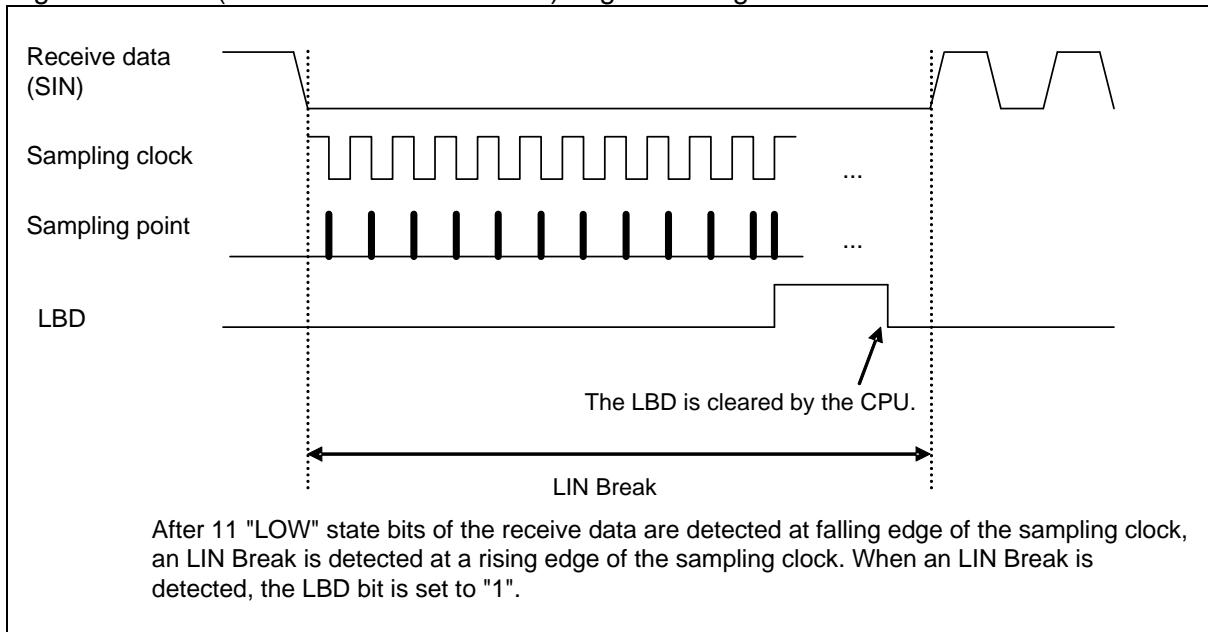
Figure 2-3 ORE (Overrun Error) flag bit set timing



■ LIN break field detection flag (LBD) set timing

If "0" is input for a width of 11 bits or more as Serial Input (SIN), the LBD bit is set to "1". If LIN break field interrupts are enabled (ESCR:LBIE = 1) then, a receive interrupt occurs.

Figure 2-4 LBD (LIN Break field Detection) flag set timing



2.2. Interrupt and flag set timing when receive FIFO is used

If receive FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register (FBYTE)) is received.

■ Interrupt and flag set timing when receive FIFO is used

If the receive FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.

- When full FBYTE data is received, the receive data full flag (SSR:RDRF) of the Serial Status register is set to "1". If receive interrupts are enabled (SCR:RIE) during this time, a receive interrupt occurs.
 - If both of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (SSR:RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FCR:FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to "0". If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- When the Receive Data Register (RDR) data is all read and receive FIFO is emptied, the receive data full flag (SSR:RDRF) is cleared.
 - If the valid receive data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE = 1) occurs.

Figure 2-5 Receive interrupt occurrence timing when receive FIFO is used

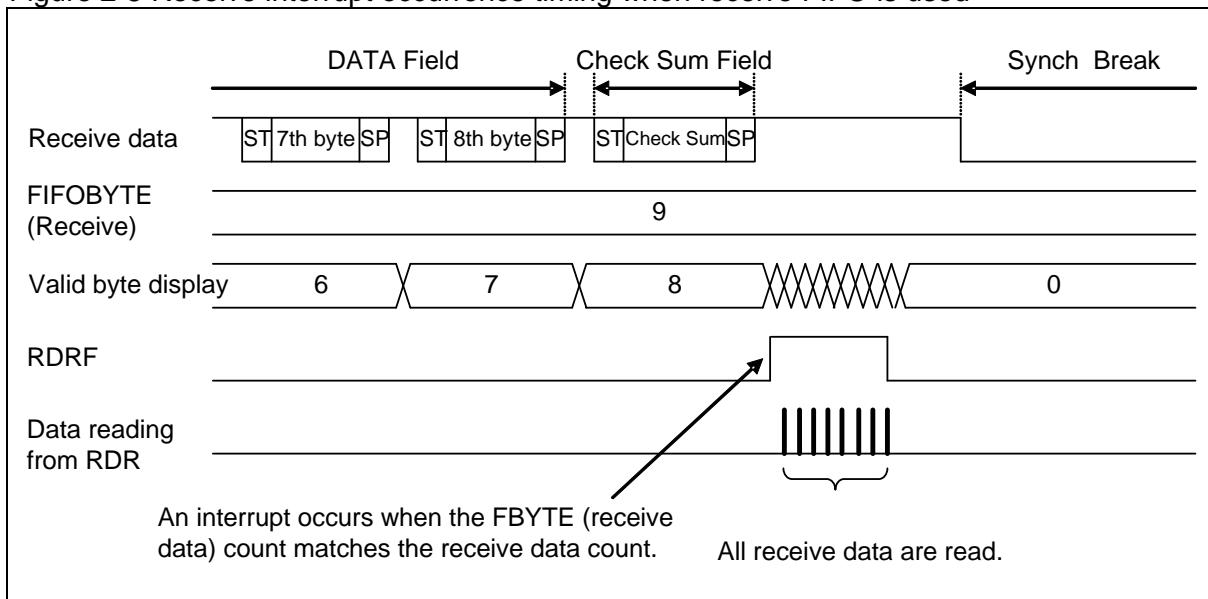
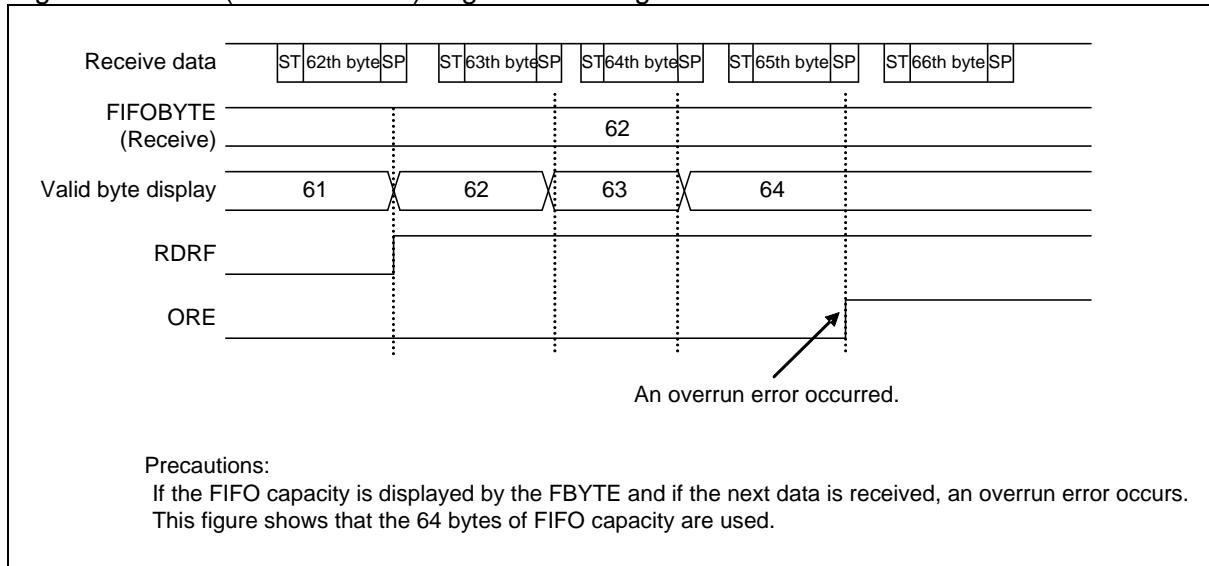


Figure 2-6 ORE (Overrun Error) flag bit set timing



2.3. Transmit interrupt and flag set timing

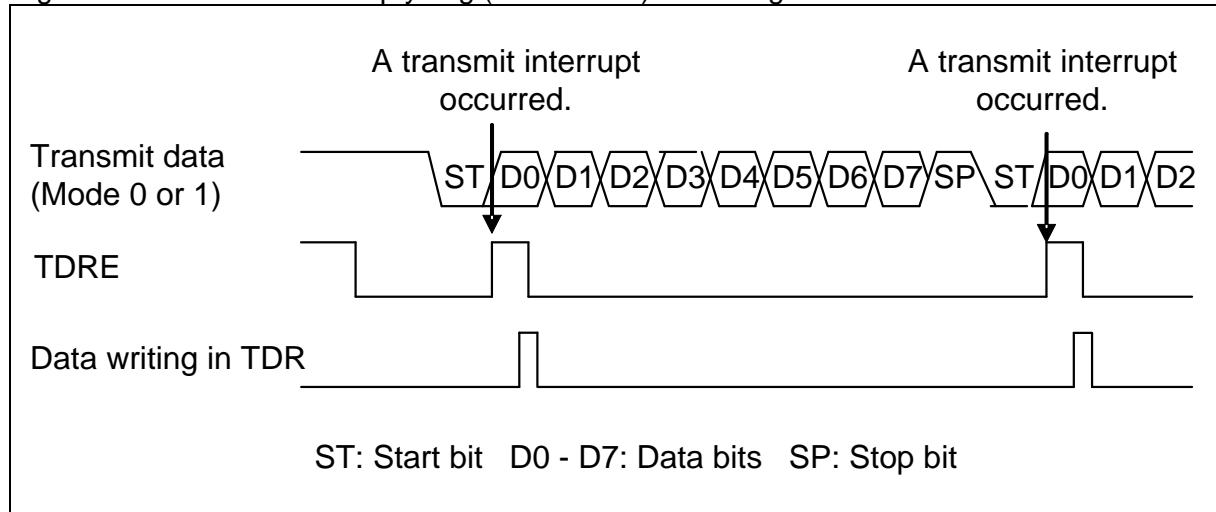
A transmit interrupt occurs when outgoing data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

■ Transmit interrupt and flag set timing

● Transmit data empty flag (TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written in the TDR (SSR:TDRE=1). If transmit interrupts are enabled (SCR:TIE=1) during this time, a transmit interrupt occurs. As the TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

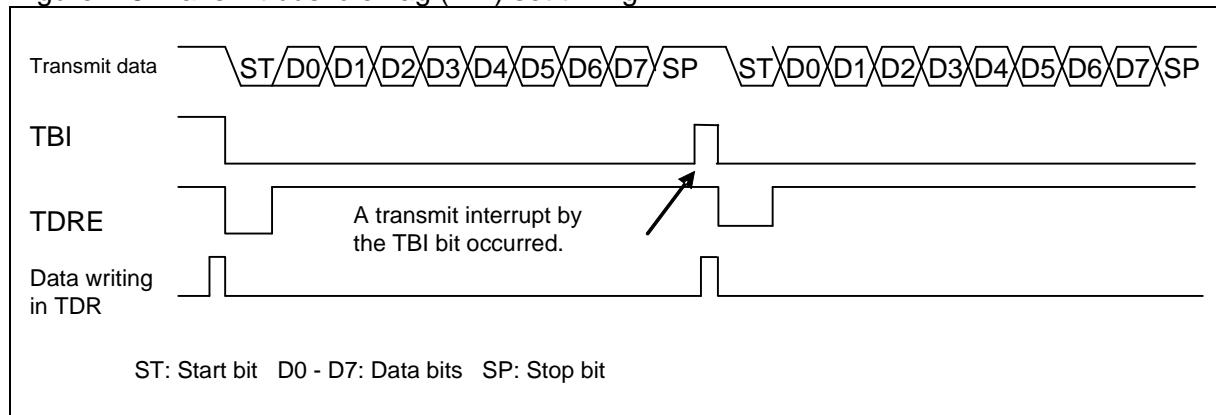
Figure 2-7 Transmit data empty flag (SSR:TDRE) set timing



● Transmit bus idle flag (TBI) set timing

If the Transmit Data Register is empty (TDRE="1") and no data is transmitted, the SSR:TBI bit is set to "1". If transmit bus idle interrupts are enabled (SCR:TBIE=1) during this time, a transmit interrupt occurs. When outgoing data is written to the Transmit Data Register (TDR), both the TBI bit and the transmit interrupt request are cleared.

Figure 2-8 Transmit bus idle flag (TBI) set timing



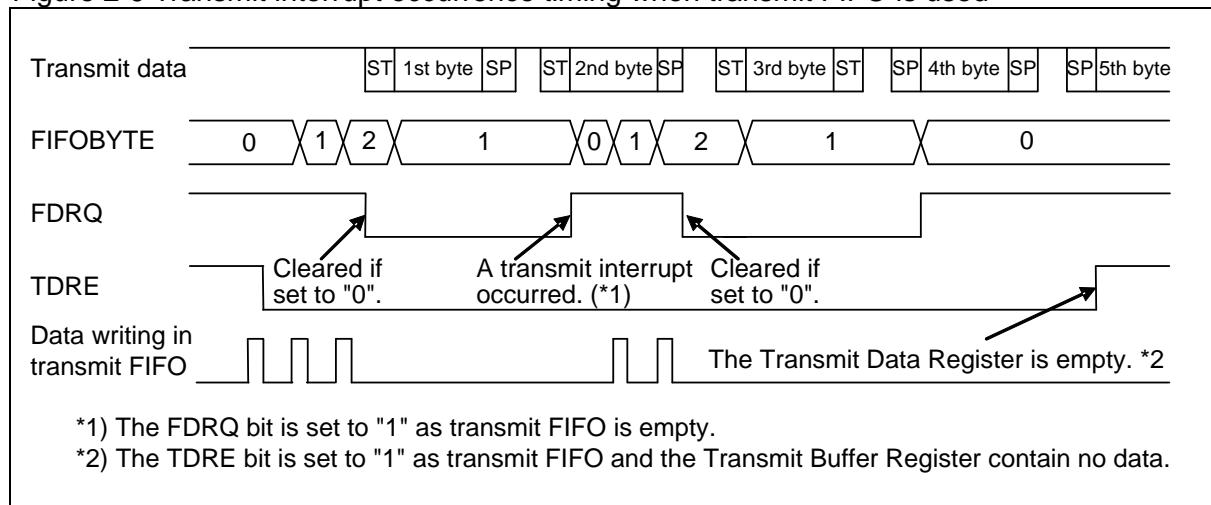
2.4. Interrupt and flag set timing when transmit FIFO is used

When the transmit FIFO is used, an interrupt occurs if the FIFO contains no data.

■ Transmit interrupt and flag set timing when transmit FIFO is used

- If the transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1". If FIFO transmit interrupts are enabled (FCR1:FTIE=1), a transmit interrupt occurs.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- When transmit FIFO is filled with data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "0".
- To check to see if transmit FIFO contains any data, read from the FIFO Byte Register (FBYTE). If FBYTE=0x00, no data exists in the transmit FIFO.

Figure 2-9 Transmit interrupt occurrence timing when transmit FIFO is used



3. Dedicated Baud Rate Generator

For the LIN interface (ver. 2.1) transmitting/receiving clock source, either of the following can be selected.

- Dedicated baud rate generator (reload counter)
- An external clock input to the baud rate generator (reload counter)

■ LIN interface (ver. 2.1) baud rate

Select one of the following two baud rates.

● Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an internal clock by the set value.

To set the clock source, select an internal clock (SMR:EXT = 0).

● Baud rate obtained by dividing an external clock using the dedicated baud rate generator (reload counter)

Use an external clock for the clock source of the reload counter.

To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an external clock by the set value.

To set the clock source, select use of an external clock and the baud rate generator clock (SMR:EXT = 1).

This mode is designed for cases where an oscillator with a divided non-standard frequency is used.

<Notes>

- Set the external clock (EXT = 1) while the reload counter is stopped (BGR1/0 = 15h00).
- If an external clock is selected (EXT = 1), its HIGH and LOW signals must have a width at least of two bus clocks.

3.1. Baud rate settings

The following explains how to set the baud rate, and also a result of serial clock frequency calculation.

■ Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$$V = \phi / b - 1$$

V : Reload value b: Baud rate ϕ : Bus clock frequency or external clock frequency

(2) Calculation example

To set the 16 MHz bus clock, to use the internal clock, and to set the 19200-bps baud rate, set the reload value as follows.

Reload value:

$$V = (16 \times 1000000) / 19200 - 1 = 832$$

Therefore, the baud rate is:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

(3) Baud rate error

The baud rate error can be obtained from the following equation.

$$\text{Error (\%)} = (\text{Calculated value} - \text{Target value}) / \text{Target value} \times 100$$

Example: To set the 20 MHz bus clock and 153600-bps target baud rate:

$$\text{Reload value} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{Baud rate (Calculated value)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error (\%)} = (153846 - 153600) / 153600 \times 100 = 0.16 \text{ (\%)}$$

<Notes>

- If the reload value is set to "0", the reload counter is stopped.
- If the reload value is even, the "LOW" signal width of serial clock is longer than the "HIGH" signal width for a single cycle of bus clock. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.
- Set the reload value to 3 or more. Note that data may not be received normally due to the baud rate error and reload value setting.

■ Reload value and baud rate for each bus clock frequency

Table 3-1 Reload values and baud rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	-	0
6M	-	-	-	-	-	-	-	-	-	-	-	-
5M	-	-	-	-	-	-	-	-	-	-	-	-
4M	-	-	-	-	-	-	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	-	-	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

- Value: BGR1/0 register set value

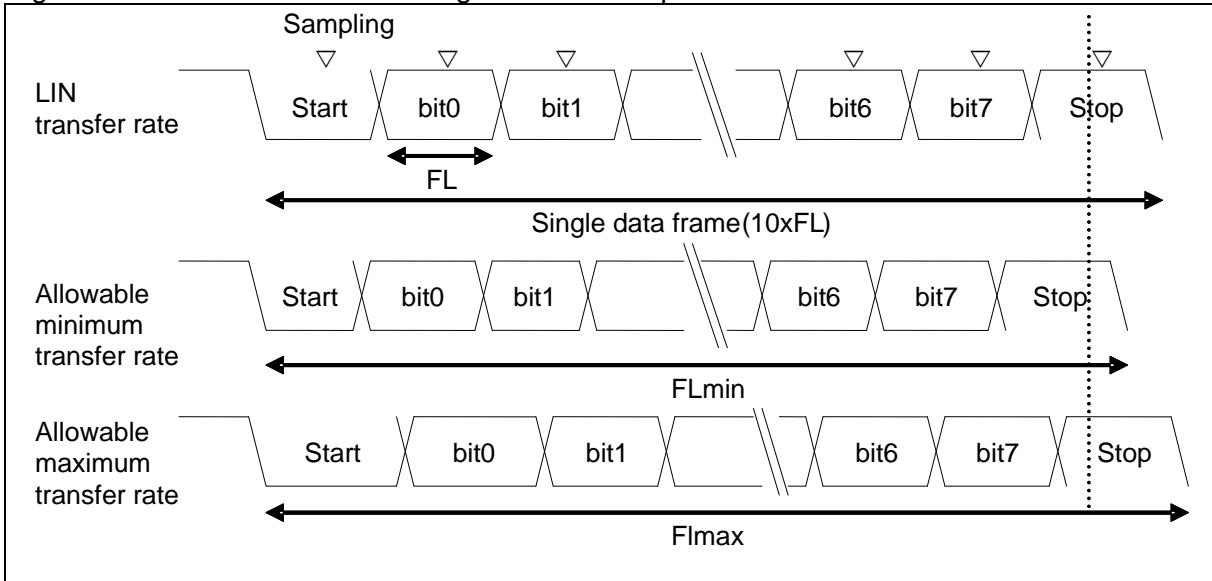
- ERR: Baud rate error (%)

■ Allowable baud rate range for data reception

The following shows the range of baud rate error allowed for the destination to receive data.

Set the reception baud rate error by using the following formulas to ensure that the value falls within the allowable range.

Figure 3-1 Allowable baud rate range for data reception



As shown in the figure, after detection of the start bit, the sampling timing of incoming data is determined by the counter set in the BGR1/0 register. Data can be received successfully if the bit sequence including the stop bit matches the sampling timing.

If this applies to a reception of 10 bits, a theoretical explanation can be given in the following.

Assuming that the sampling timing margin is one bus clock (ϕ), the minimum allowable transfer rate (FL_{min}) is determined as follows:

$$FL_{min} = (10 \text{bit} \times (V+1) - (V+1)/2 + 2) / \phi = (19V + 23)/2 \phi \text{ (s)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

Thus, the maximum baud rate that allows the destination to receive data (BG_{max}) is determined as follows.

$$BG_{max} = 10/FL_{min} = 20\phi/(19V+23) \text{ (bps)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

When data is received at the maximum allowable transfer rate (FL_{max}), the starting point of the incoming 10th bit is sampled.

Thus, the maximum allowable transfer rate (FL_{max}) is determined as follows:

$$9/10 \times FL_{max} = (10 \text{bit} \times (V+1) - (V+1)/2) / \phi \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

$$FL_{max} = (19/18 \times 10 \times (V+1)) / \phi$$

Assuming that the sampling timing margin (ϕ) is two clocks, the maximum allowable transfer rate (FL_{max}) is determined as follows:

$$9/10 \times FL_{max} = (10 \text{bit} \times (V+1) - (V+1)/2 - 2) / \phi \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

$$FL_{max} = (19/18 \times 10 \times (V+1) - 40/18) / \phi = (190V + 150)/20 \phi \text{ (s)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

Accordingly, the minimum baud rate that allows the destination to receive data (BG_{min}) is determined as follows:

$$BG_{min} = 10/FL_{max} = 18\phi/(19V+15) \text{ (bps)} \quad V: \text{Reload value}, \phi: \text{Bus clock}$$

From the above formulas that yields the minimum/maximum baud rates, the allowable baud rate errors between the LIN interface (ver. 2.1) and the destination can be obtained as shown in the following table.

Reload value (V)	Maximum allowable baud rate error	Minimum allowable baud rate error
3	0%	0
10	+3.28%	-3.41%
50	+4.83%	-4.87%
100	+5.04%	-5.07%
200	+5.15%	-5.16%
32767	+5.26%	-5.26%

<Note>

Receive accuracy depends on the number of bits per frame, bus clock, and reload value. The higher the bus clock and frequency division ratio are, the higher the accuracy becomes.

■ External clock

Writing "1" to the EXT bit of the Baud Rate Generator Register (BGR) causes the baud rate generator to divide the external clock's frequency.

<Note>

The external clock signal is synchronized with the internal clock on the LIN interface (ver. 2.1). Therefore, an external clock that does not allow synchronization causes unstable operation.

■ Functions of reload counter

There are two types of reload counters: The transmit reload counter and the receive reload counter, both functioning as a dedicated baud rate generator. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from the external or internal clock.

■ Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

■ Restarting

The reload counter restarts counting in the following conditions.

● Common to transmit and receive reload counters

- A programmable reset (SCR:UPCL bit)

● Receive reload counter

- Detection of the start bit's falling edge in asynchronous mode

4. LIN Interface (Ver. 2.1) Operations

The LIN interface (ver. 2.1) performs bi-directional LIN communication of master and slave.

■ Master mode operations

● Selecting master mode

To operate the LIN interface as a master, set the SCR:MS bit to "0".

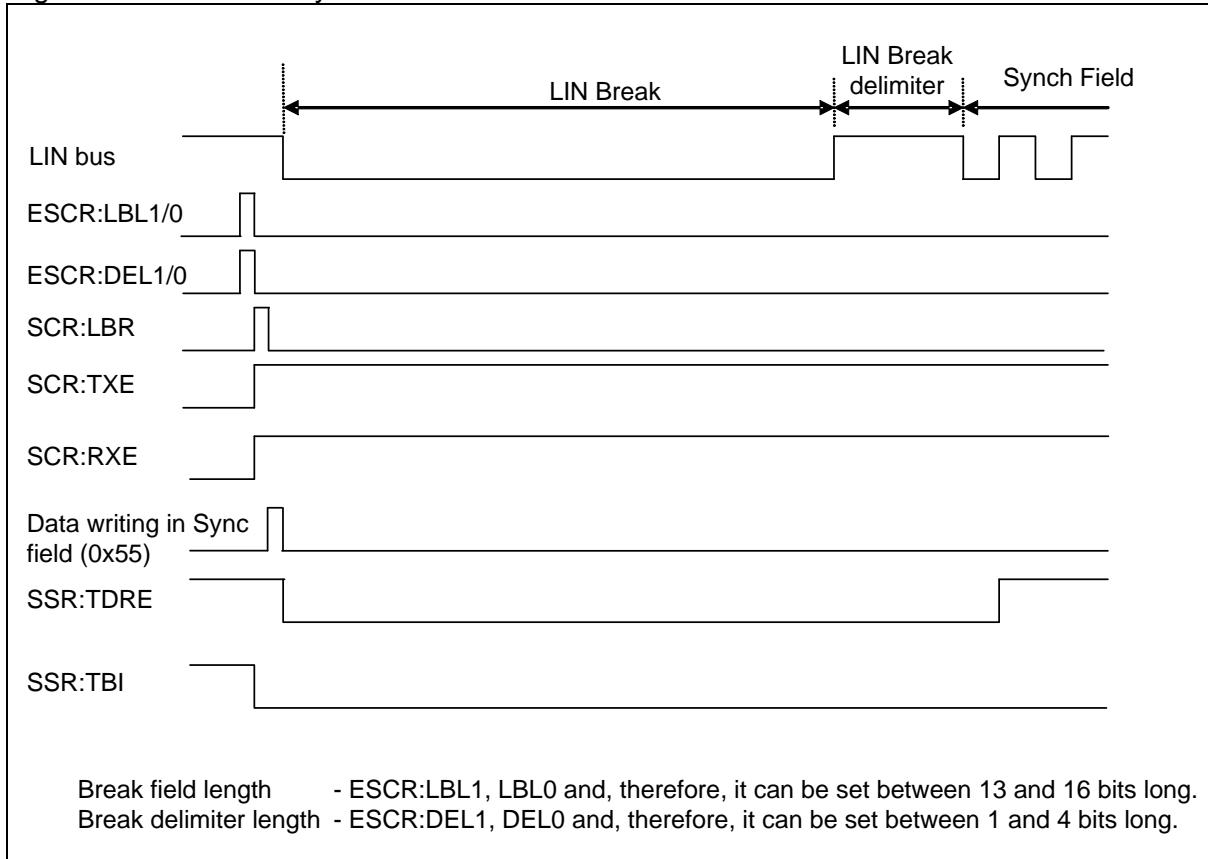
● Break field transmission-sync field transmission

- The break field length (ESCR:LBL1, LBL0) and the break field delimiter length (ESCR:DEL1, DEL0) can be selected.
- If transmission is enabled (SCR:TXE=1), and the SCR:LBR bit (LIN Break field setting bit) is set to "1", then the break field is transmitted.
- The sync field is transmitted when "0x55" is written to the Transmit Data Register (TDR).

<Notes>

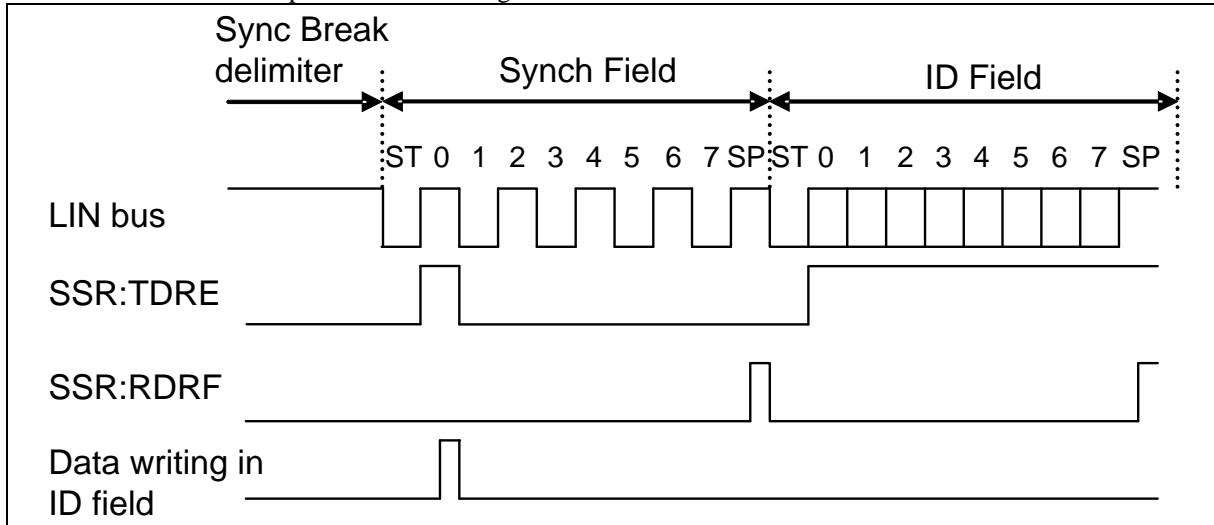
- Before setting the Transmit Data Register (TDR) to "0x55", set the SCR:LBR bit (LIN break field setting bit) to "1".
- Setting the SCR:RXE bit (reception enable bit) to "1" does not enable the break field to perform reception.

Figure 4-1 Break field-sync field transmission



● Sync field transmission-ID field transmission

- When the first bit of the sync field (0x55) is transmitted, the SSR:TDRE (transmit data empty) bit is set to "1". If transmit interrupts are enabled (SCR:TIE = 1) during this time, a transmit interrupt occurs.
- If a transmit interrupt occurs, the ID field can be written to the Transmit Data Register (TDR).
- If a receive interrupt occurs, compare the received data with the transmit data to make sure that no error has occurred.
- The ID field is output in 8-bit data length and LSB-first order.



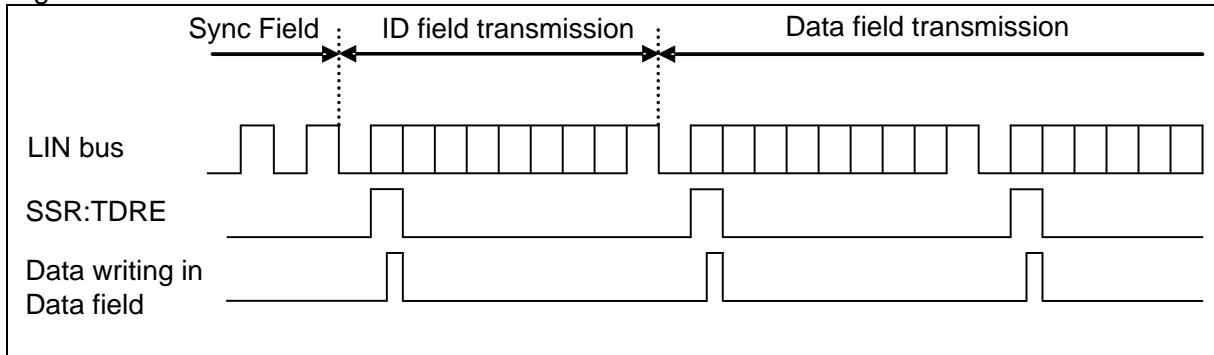
● ID field transmission-DATA field transmission/reception

Select whether to transmit the DATA field to a slave device or to receive the DATA field.

(To transmit the DATA field)

When the first bit of the ID field is transmitted, the SSR:TDRE bit is set to "1". Then data can be written to the DATA field.

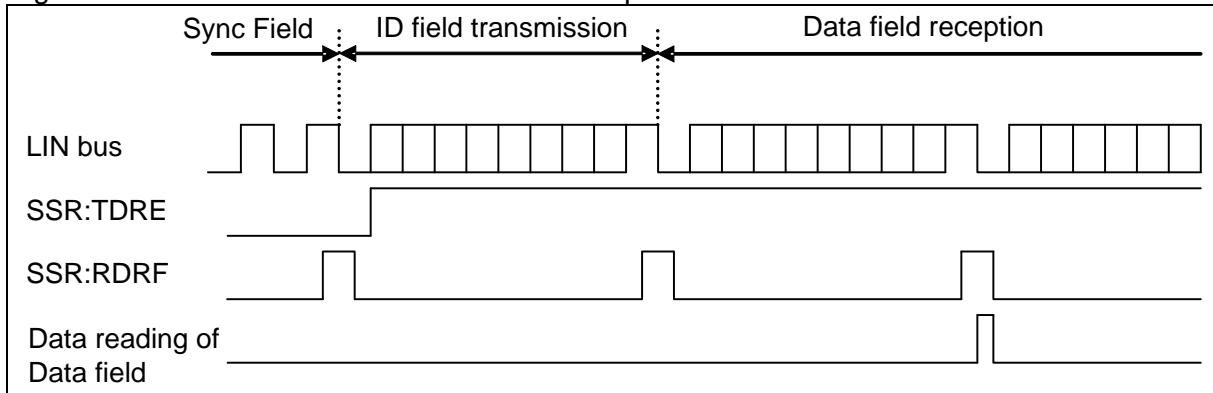
Figure 4-2 ID field transmission-DATA field transmission



(To receive the DATA field)

- When the first bit of the ID field is transmitted, the SSR:TDRE bit is set to "1". However, do not write any transmit data then. Also disable transmit interrupts (SCR:TIE = 0).
- When the DATA field is received, SSR:RDRF is set to "1". If receive interrupts are enabled (SSR:RIE = 1) then, a receive interrupt occurs.
- A start bit is detected when a falling edge is detected after data passes the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) and a LOW level is detected for the data passing the sampling point.

Figure 4-3 ID field transmission-DATA field reception



<Notes>

- The LIN interface (Ver. 2.1) includes noise filter (with the majority value applied after sampling serial data input three times with the bus clock). However, design the board so as not to allow noise to pass through this filter or perform communications so that any noise that has passed does not cause any problems (e.g., by adding a data checksum to the end and resending the data if any error occurs).
- During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data cannot be received successfully. To output frames continuously, adequate intervals should be considered between frames.

● Master mode operation timing chart (when FIFO is not used)

Figure 4-4 LIN bus timing (when DATA field is transmitted and FIFO is not used)

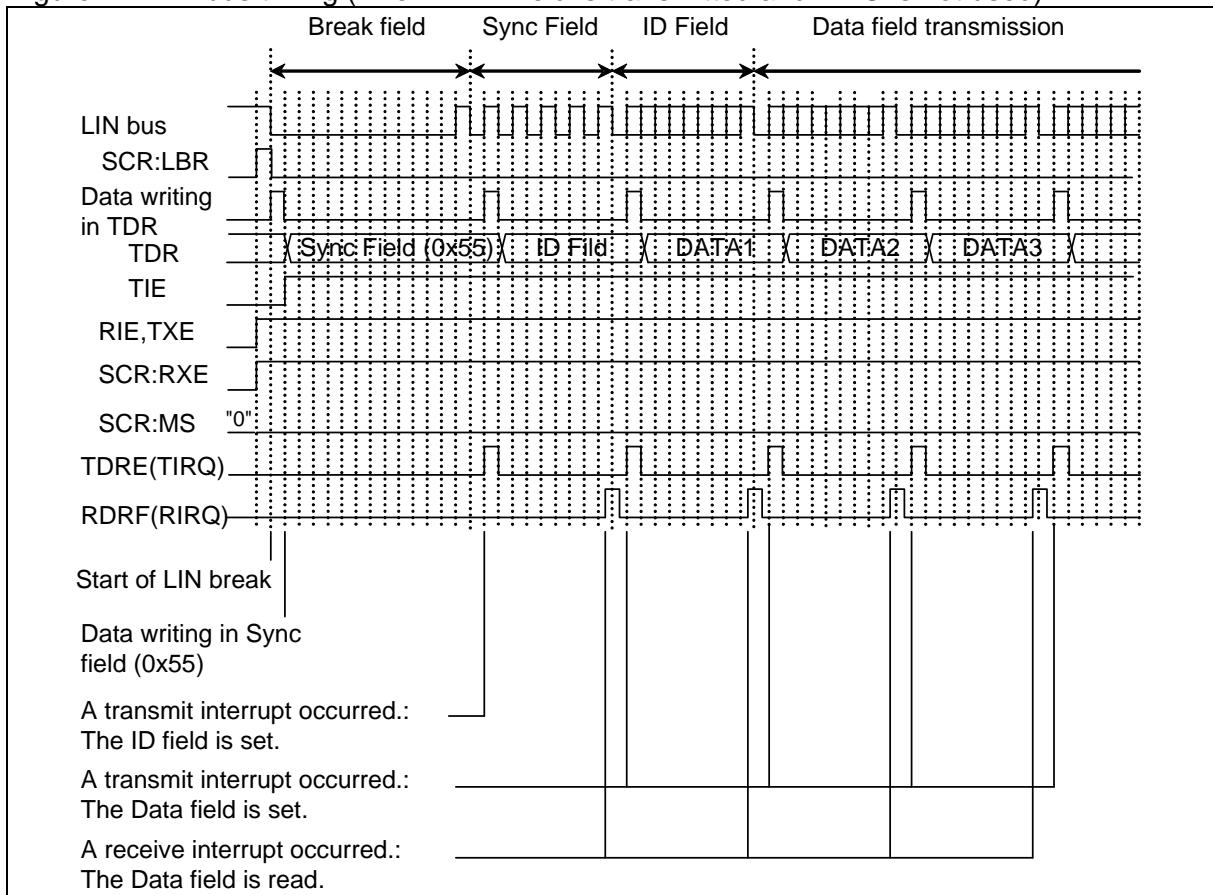
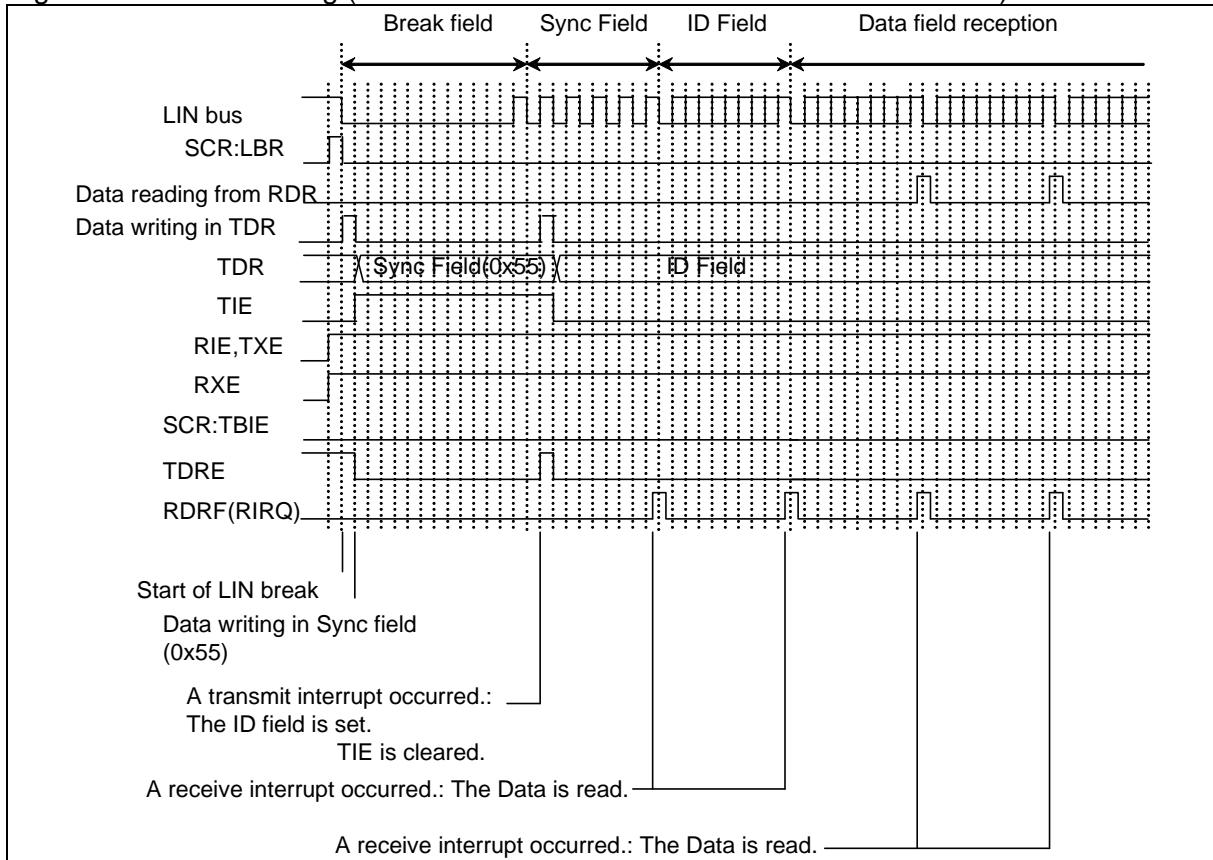


Figure 4-5 LIN bus timing (when DATA field is received and FIFO is not used)



● Master mode operation timing chart (when FIFO is used)

Figure 4-6 LIN bus timing (when DATA field is transmitted and FIFO is used)

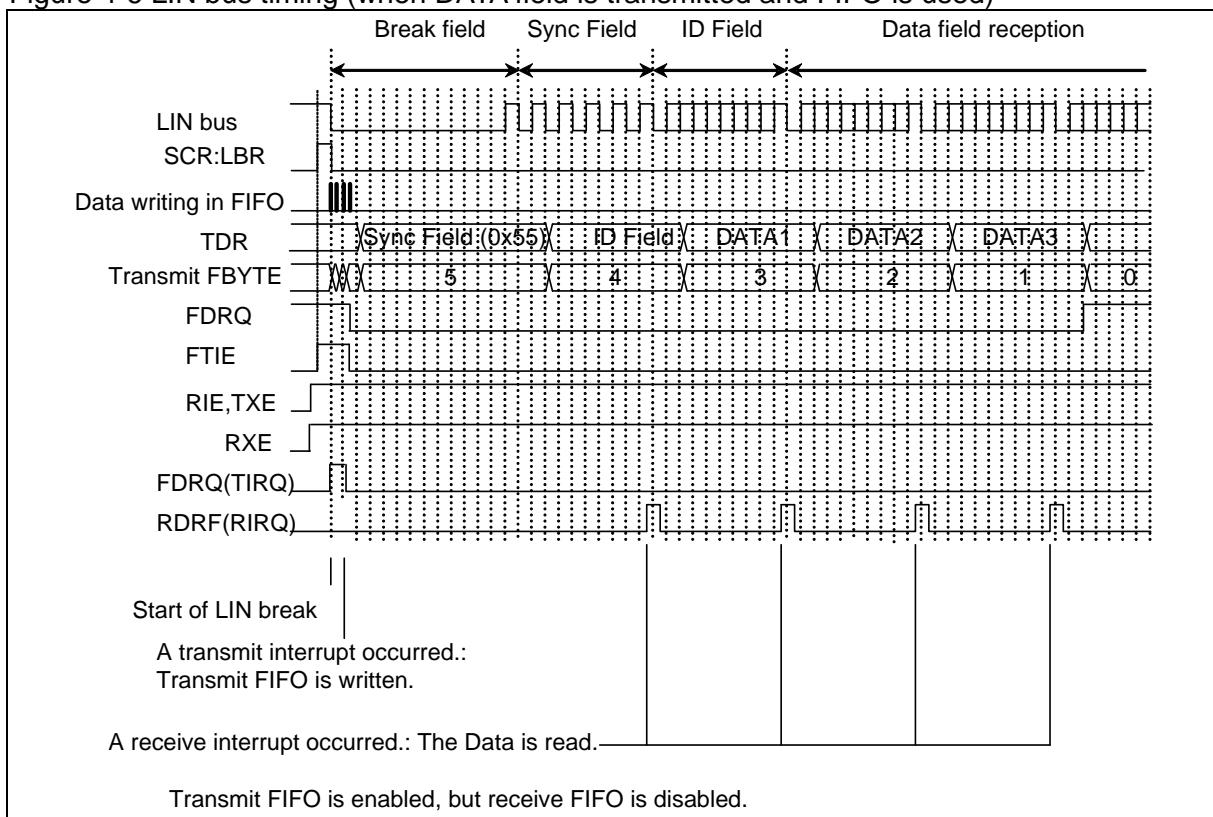
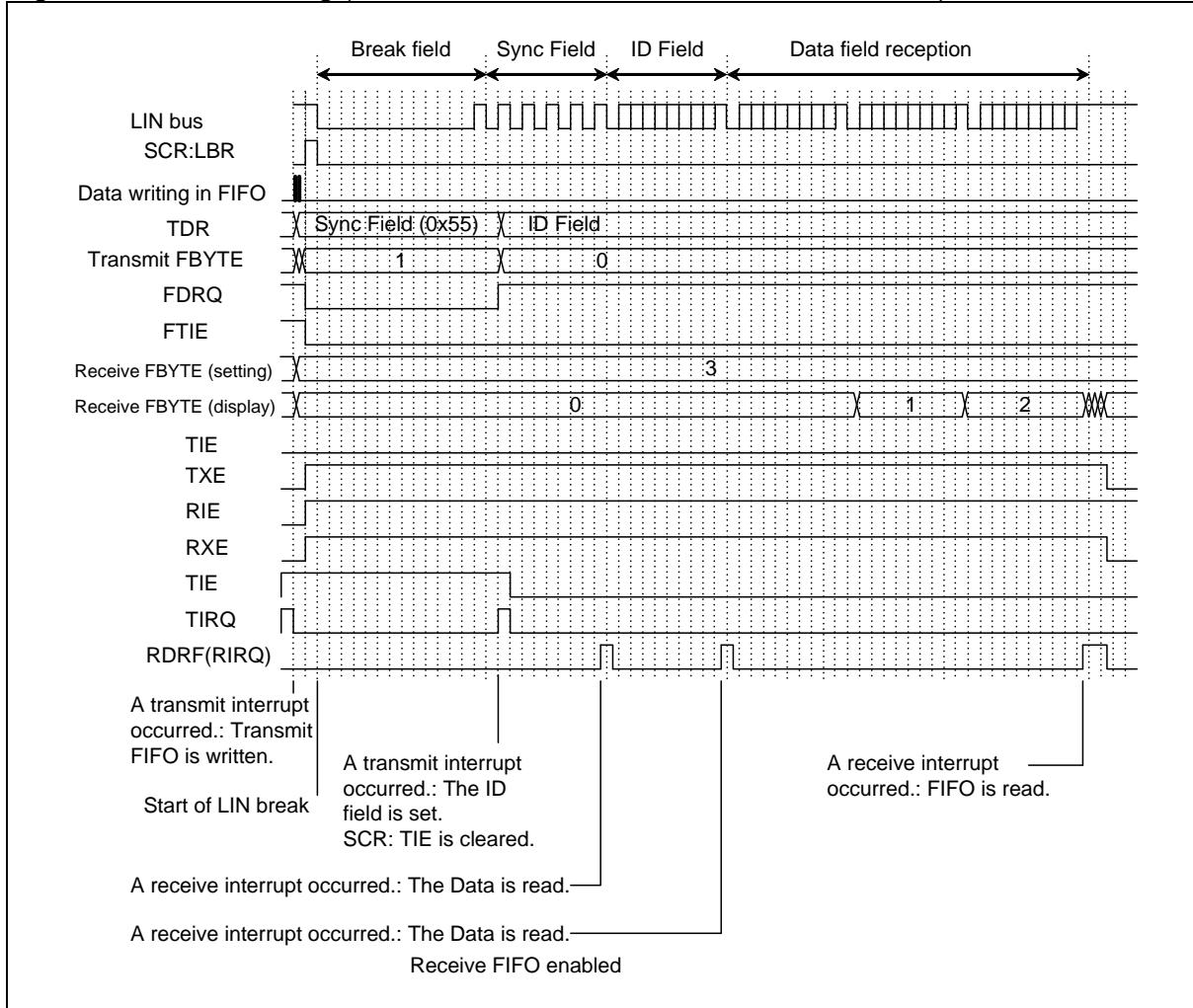


Figure 4-7 LIN bus timing (when DATA field is received and FIFO is used)



■ Slave mode operations

● Selecting slave mode

To operate the LIN interface as a slave, set the SCR:MS bit to "1".

● Break field reception-sync field reception

1. If the break field is input, the break field is detected (SSR:LBD = 1) at the 11th bit.
If the ESCR:LBIE bit is set to "1" then, a receive interrupt occurs.
2. Enable ICU interrupts then to detect both edges.
3. The LIN interface (ver. 2.1), upon the detection of the first falling edge in the sync field, sets the internal signal (LSYN) input to ICU to HIGH to start the ICU. This internal signal (LSYN) turns to LOW at the fifth falling edge.
4. The internal signal (LSYN) input to ICU is a value that the HIGH period multiplies the baud rate by eight. The baud rate set value is obtained as follows:

If the free run timer is not overflowed:

$$\text{BGR value} = (b - a) \times Fe / (8 \times \phi) - 1$$

If the free run timer is overflowed:

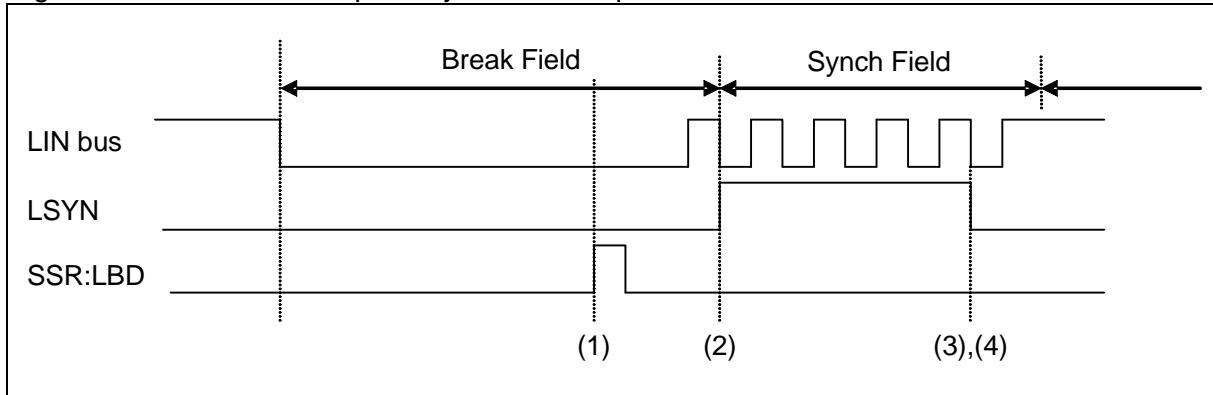
$$\text{BGR value} = (\max + 1 + b - a) \times Fe / (8 \times \phi) - 1$$

where, max : Maximum value of the free run timer
 a : The ICU data register value after the first interrupt
 b : The ICU data register value after the second interrupt
 ϕ : Bus clock frequency (MHz)
 Fe : External clock frequency (MHz). When the internal clock is used (EXT = 0),
 $Fe = \phi$ is assumed.

<Note>

To operate the break field the sync field, disable the reception (SCR:RXE = 0).

Figure 4-8 Break field reception-sync field reception



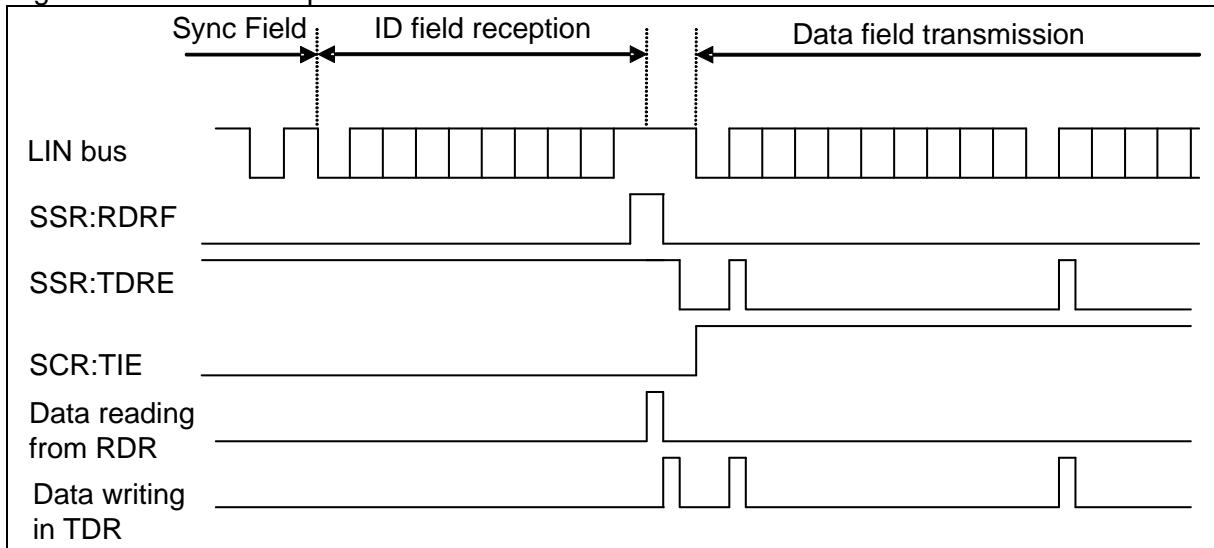
● ID field reception-DATA field transmission/reception

After reception of the ID field, whether to transmit or to receive the DATA field can be selected.

(To transmit the DATA field)

After reception of the ID field, write data to the Transmit Data Register (TDR). Enable transmit interrupts (SCR:TIE = 1) during this time.

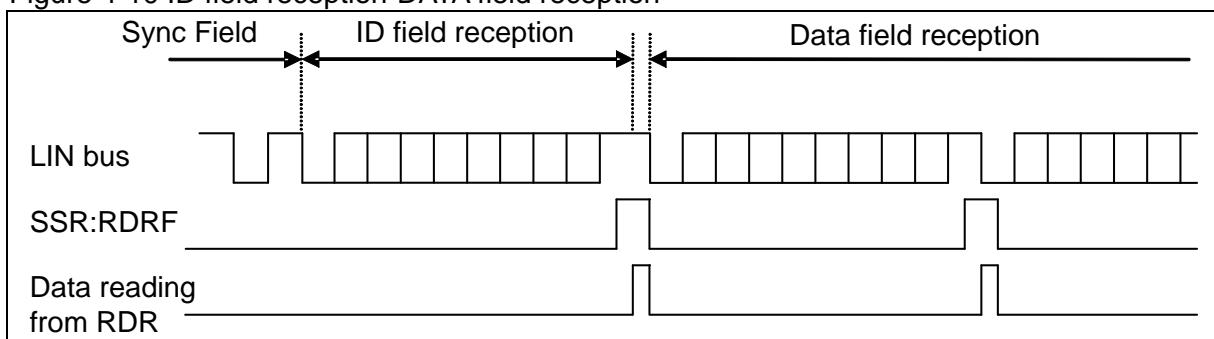
Figure 4-9 ID field reception-DATA field transmission



(To receive the DATA field)

- Every time the DATA field is received, SSR:RDRF is set to "1". If receive interrupts are enabled (SCR:RDRF = 1) then, a receive interrupt occurs.
- A start bit is detected when a falling edge is detected after data passes the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) and a LOW level is detected for the data passing the sampling point.

Figure 4-10 ID field reception-DATA field reception



<Notes>

- The LIN interface (Ver. 2.1) includes noise filter (with the majority value applied after sampling serial data input three times with the bus clock). However, design the board so as not to allow noise to pass through this filter or perform communications so that any noise that has passed does not cause any problems (e.g., by adding a data checksum to the end and resending the data if any error occurs).
- During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data cannot be received successfully. To output frames continuously, adequate intervals should be considered between frames.

● Slave mode operation timing chart

Figure 4-11 LIN bus timing (when DATA field is transmitted and FIFO is not used)

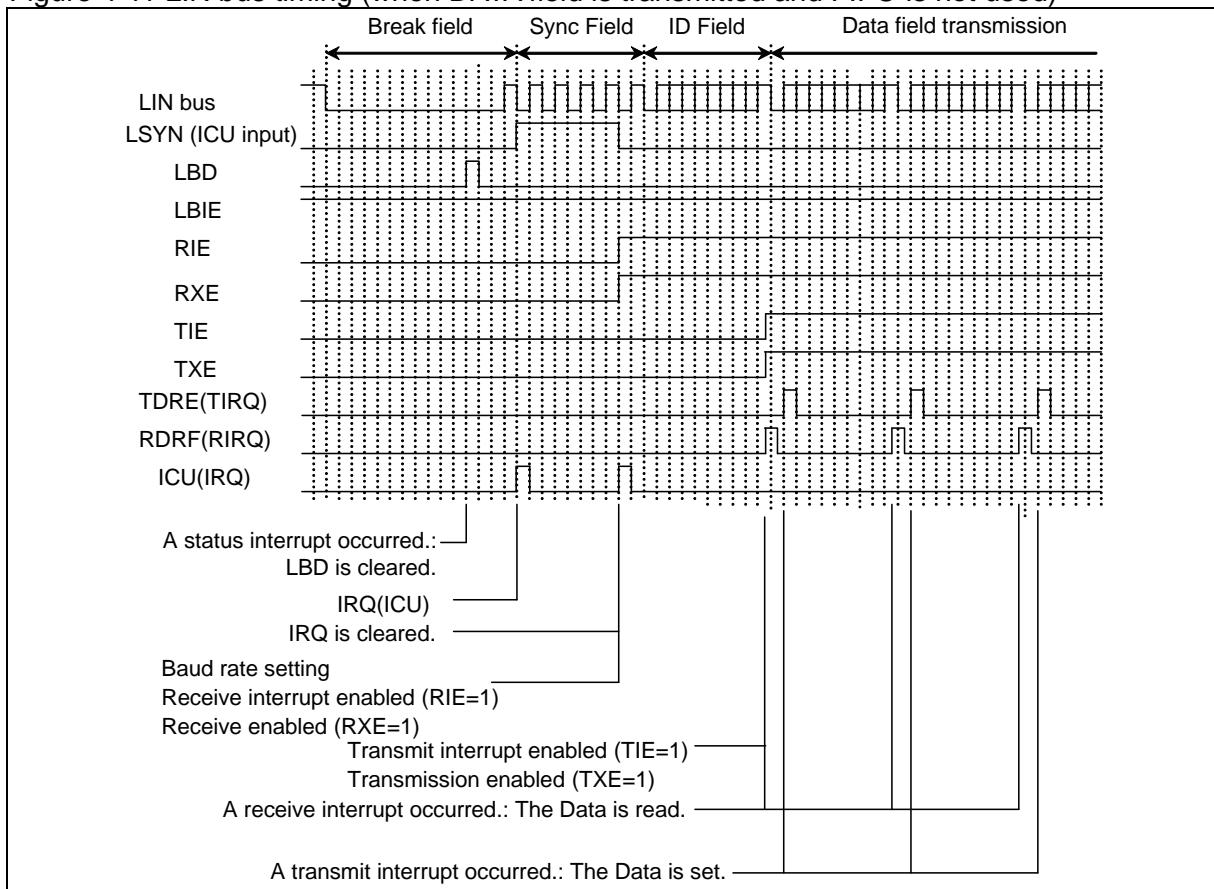
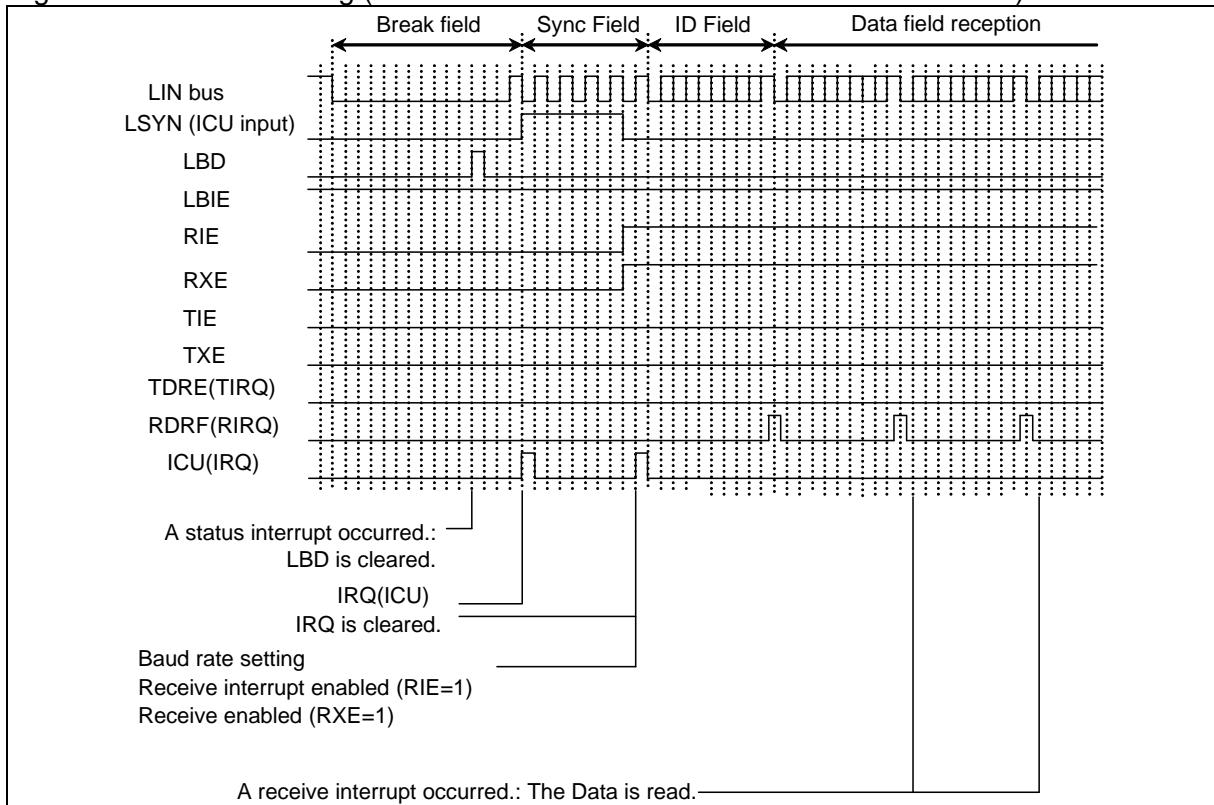


Figure 4-12 LIN bus timing (when DATA field is received and FIFO is not used)



● If FIFO is used

Figure 4-13 LIN bus timing (when DATA field is transmitted and FIFO is used)

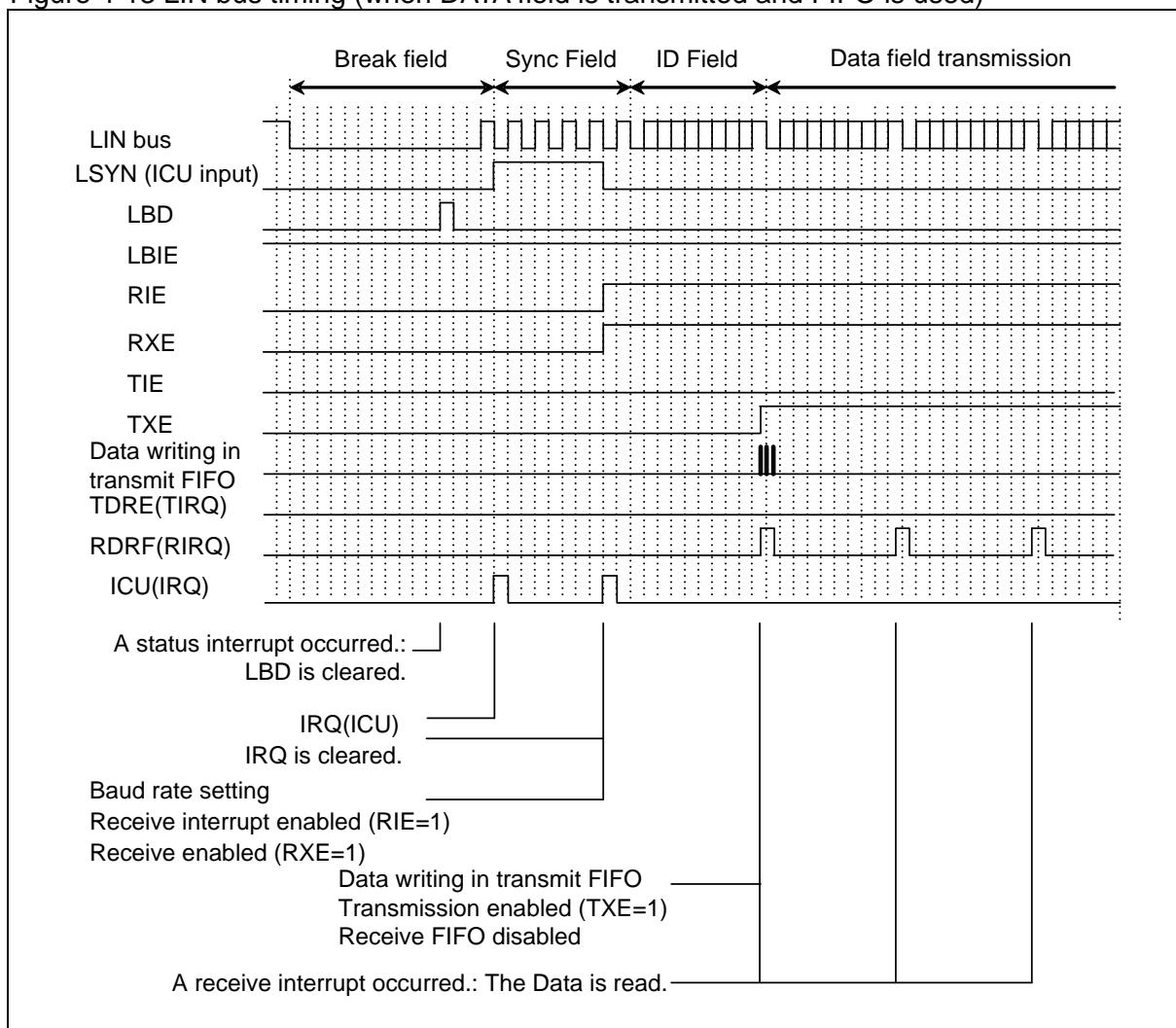
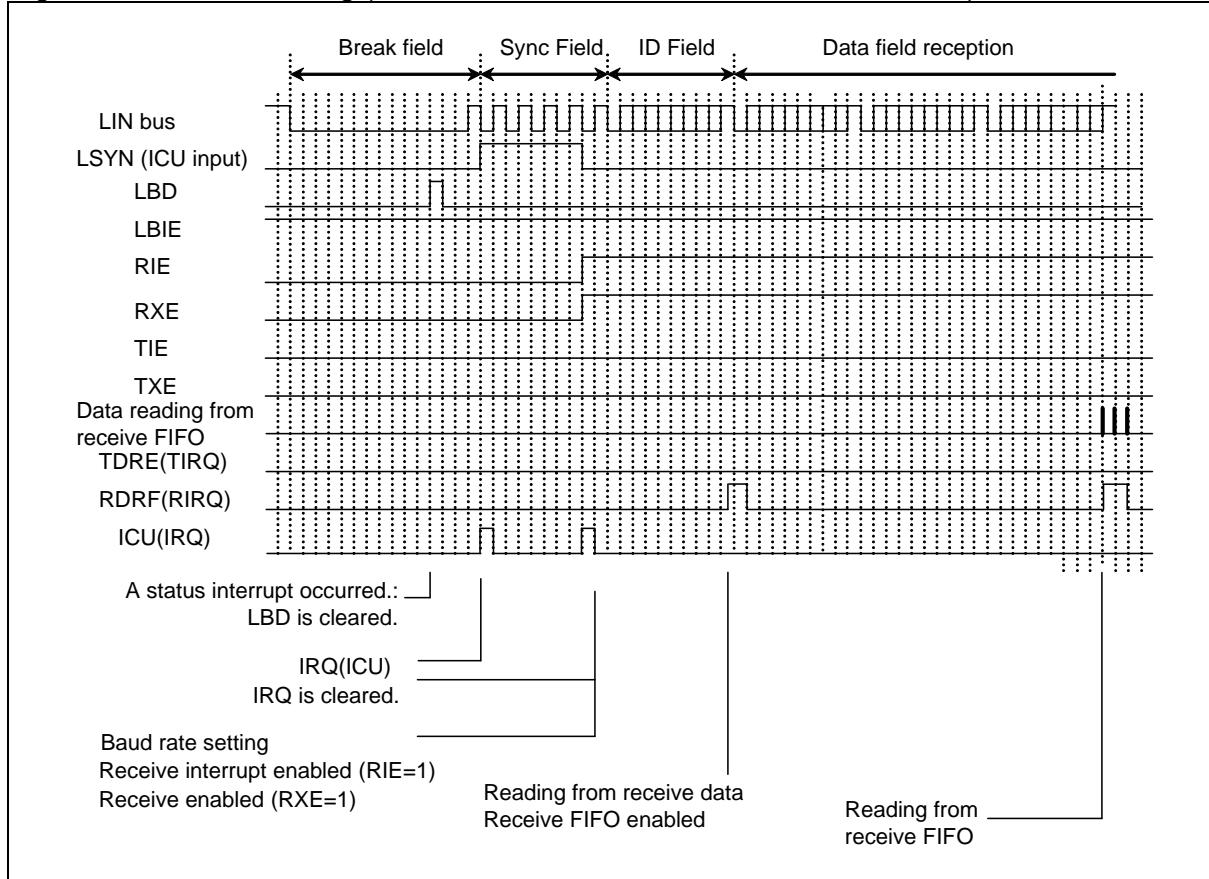


Figure 4-14 LIN bus timing (when DATA field is received and FIFO is used)



5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow

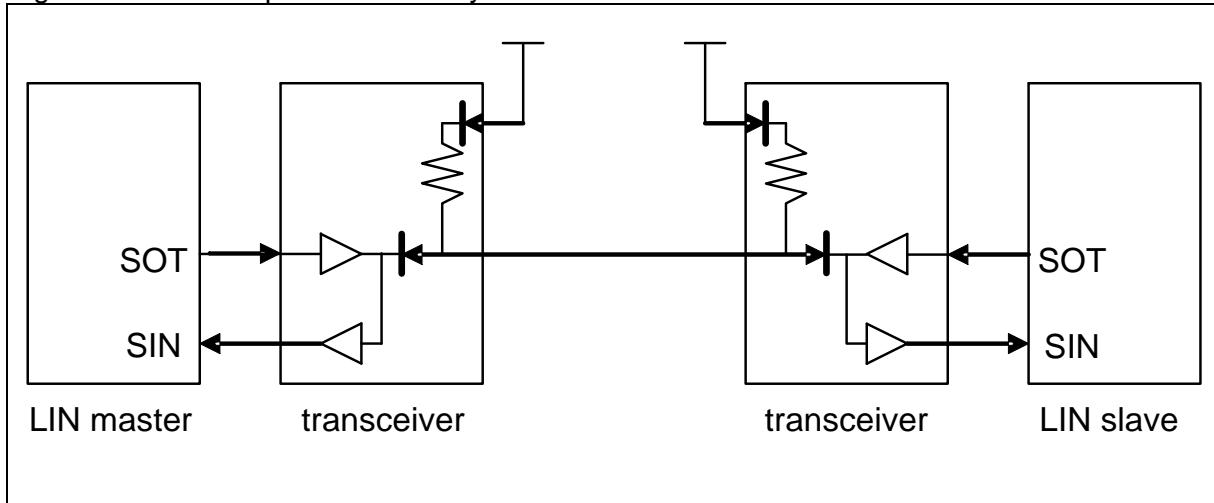
In Operation Mode 3 (LIN communication mode), the LIN interface (Ver. 2.1) can be used for a LIN master or LIN slave system.

■ Register settings

● CPU-to-CPU connection

Figure 5-1 shows a communication system consisting of one LIN master and one LIN slave. The LIN interface (ver. 2.1) can work as a LIN master or a LIN slave.

Figure 5-1 An example of LIN bus system communication



■ Example flowchart

● Master mode operations

Figure 5-2 An example flowchart of LIN communication in master mode (when FIFO is not used)

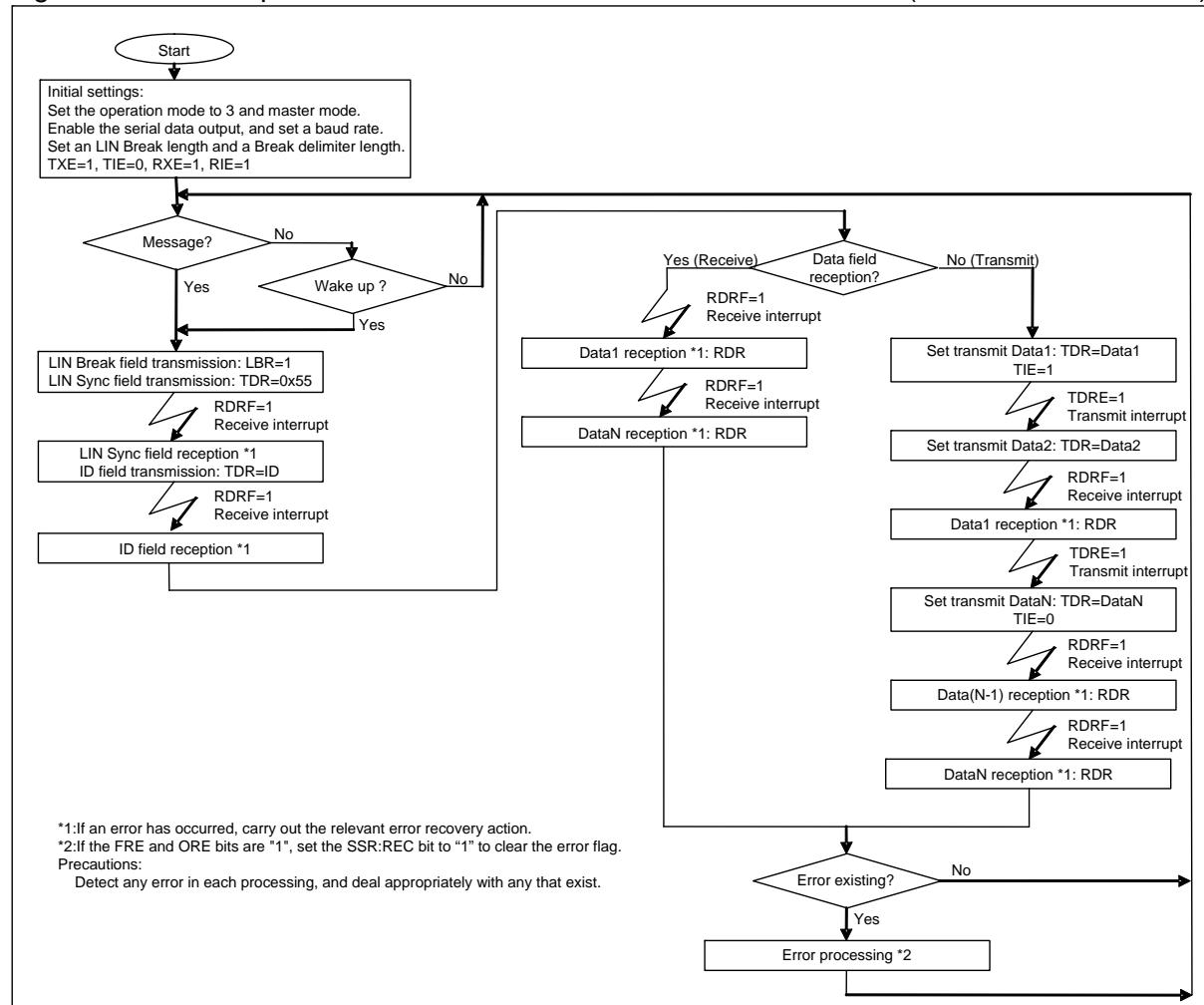
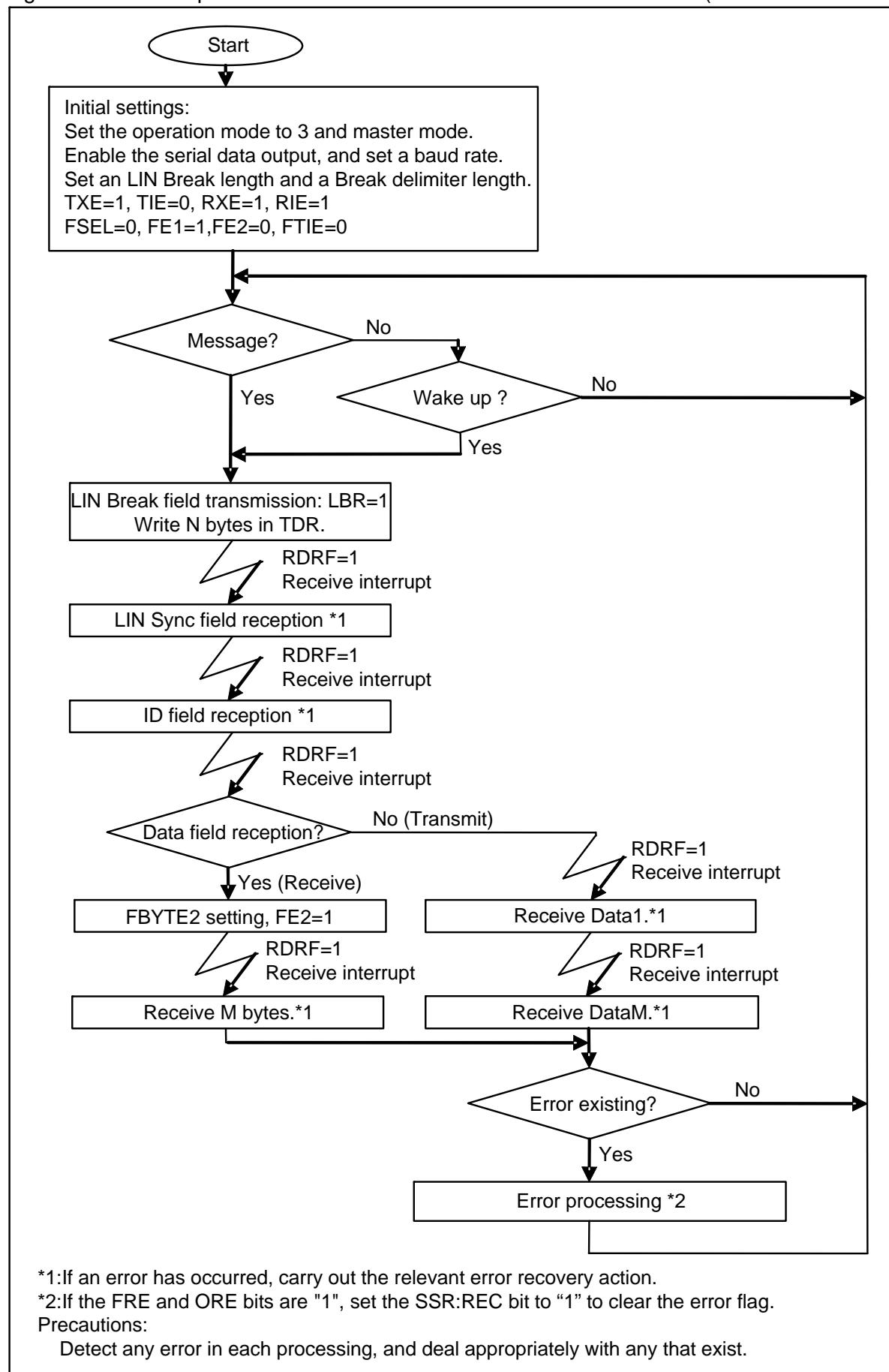


Figure 5-3 An example flowchart of LIN communication in master mode (when FIFO is used)



● Slave mode operations

Figure 5-4 An example flowchart of LIN communication in slave mode (when FIFO is not used)

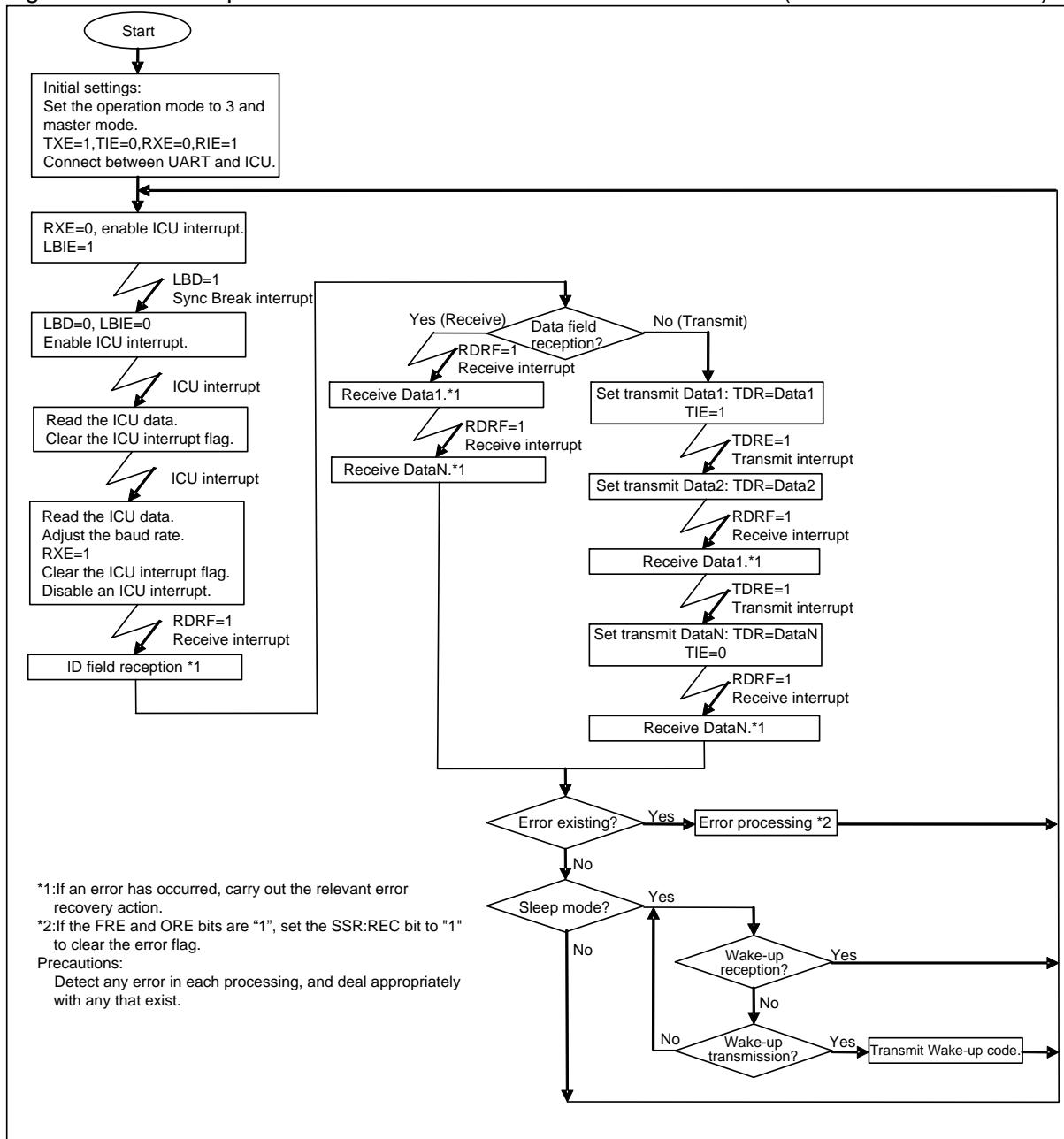
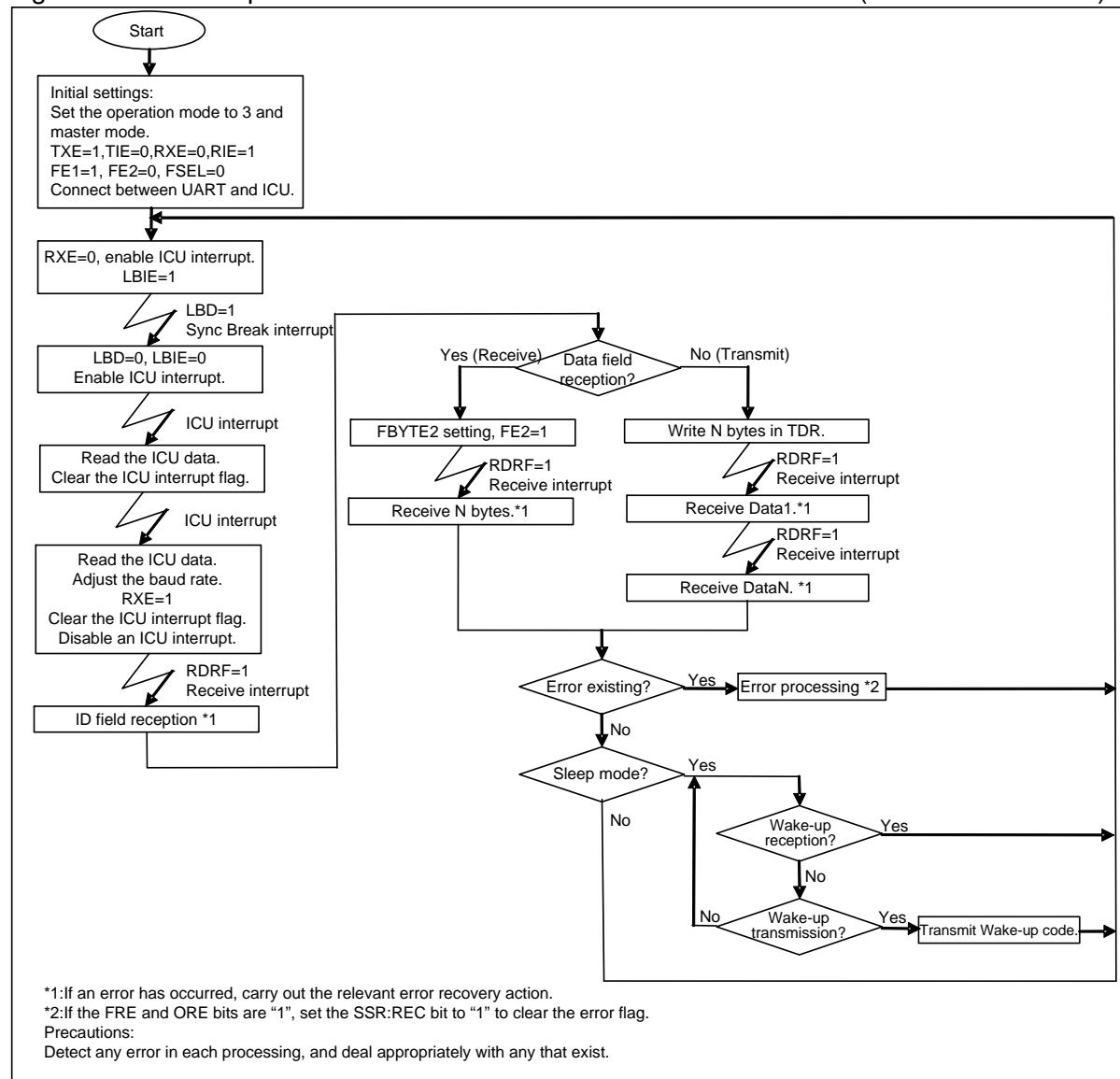


Figure 5-5 An example flowchart of LIN communication in slave mode (when FIFO is used)



6. LIN Interface (ver. 2.1) Registers

The following shows a list of LIN interface (ver. 2.1) registers.

■ List of LIN interface (ver. 2.1) registers

Table 6-1 List of LIN interface (ver. 2.1) registers

	bit 15	bit 8	bit 7	bit 0
LIN interface (ver. 2.1)	SCR (Serial Control Register)		SMR (Serial Mode Register)	
	SSR (Serial Status Register)		ESCR (Extended Communication Control Register)	
	-		RDR/TDR (Transmit/Receive Data Register)	
	BGR1 (Baud Rate Generator Register 1)		BGR0 (Baud Rate Generator Register 0)	
	-		-	
FIFO	FCR1 (FIFO Control Register 1)		FCR0 (FIFO Control Register 0)	
	FBYTE2 (FIFO2 Byte Register)		FBYTE1 (FIFO1 Byte Register)	

Table 6-2 LIN interface (ver. 2.1) bit assignment

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
SCR/ SMR	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SBL	-	-	SOE
SSR/ ESCR	REC	-	LBD	FRE	ORE	RDRF	TDRE	TBI	-	ESBL	-	LBIE	LBL1	LBL0	DEL1	DEL0
TDR/ RDR	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

6.1. Serial Control Register (SCR)

The Serial Control Register (SCR) is used to enable/disable a transmit/receive interrupt, enable/disable a transmit idle interrupt, and enable/disable data transmission and reception. Also, the SCR can be used to generate an LIN Break field and reset the LIN interface (ver. 2.1).

bit	15	14	13	12	11	10	9	8	7	...	0
Field	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	(SMR)		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Initial value	0	-	-	0	0	0	0	0			

[bit 15] UPCL: Programmable clear bit

Initializes the internal state of LIN interface (ver. 2.1).

If set to "1":

- The LIN interface (ver. 2.1) is reset directly (Software reset). However, the current register settings are maintained. The transmit or receive state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/receive interrupt causes (SSR:TDRE, TBI, RDRF, FRE, ORE, LBD) are cleared.

If set to "0":

No effect.

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Programmable clear	

<Notes>

- Disable an interrupt first, and then execute the programmable clear instruction.
- If the FIFO operation is used, disable it (FCR:FE2=0, FE1=0) first and then execute the programmable clear instruction.
- To switch from receive operation to transmit operation continuously, execute the programmable clear instruction after data is received and write transmit data to the Transmit Data Register (TDR).

[bit 14] MS: Master/Slave function select bit

Selects the master or slave mode.

If set to "0": The master mode is selected.

If set to "1": The slave mode is selected.

Bit	Description
0	Master mode
1	Slave mode

[bit 13] LBR: LIN Break Field setting bit (valid in master mode only)

If this bit is set to "1", an LIN Break field (having the length set by the ESCR:LBL1/0 bit) is generated.

Also, an LIN Break delimiter (set by the ESCR:DEL1/0 bit) is generated.

When written:

When "0" is written: No effect.

When "1" is written: An LIN Break field is generated.

When read:

"0" is always read.

Bit	Description
0	No effect.
1	An LIN Break field is generated. "0" is always read.

<Notes>

- This bit setting is valid in the master mode operation only (MS="0").
- Do not set this bit to "1" when an LIN Break field is being generated.

[bit 12] RIE: Receive interrupt enable bit

· This bit enables or disables an output of receive interrupt request to the CPU.

· If the RIE bit and the receive data flag bit (SSR:RDRF) are "1", or if any of the error flag bits (SSR:FRE, ORE) is "1", a receive interrupt request is output.

Bit	Description
0	Disables the receive interrupt.
1	Enables the receive interrupt.

[bit 11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of transmit interrupt request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a transmit interrupt request is output.

Bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit 10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and SSR:TBI bit are "1", a transmit bus idle interrupt request is output.

Bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit 9] RXE: Data receive enable bit

This bit enables or disables a data reception by the LIN interface (ver. 2.1).

- If set to "0": The data frame reception is disabled.
- If set to "1": The data frame reception is enabled.

Bit	Description
0	Disables data reception.
1	Enables data reception.

<Notes>

- Data reception starts only after a falling edge of the start bit is input even if the data reception is enabled (RXE=1).
- When the an LIN Break field is being sent in the master mode operation, no data is received even if data reception is enabled (RXE=1).
- If data reception is disabled (RXE=0), the current data reception is stopped immediately.

[bit 8] TXE: Data transmission enable bit

This bit enables or disables a data transmission by the LIN interface (ver. 2.1).

- If set to "0": The data frame transmission is disabled.
- If set to "1": The data frame transmission is enabled.

Bit	Description
0	Disables the transmission.
1	Enables the transmission.

<Note>

If data transmission is disabled (TXE=0), the current data transmission is stopped immediately.

6.2. Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set an operation mode, to select a transmission direction, data length, and stop bit length, and enable or disable an output of serial data to their pins.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SCR)			MD2	MD1	MD0	WUCR	SBL	-	-	SOE
Attribute				R/W	R/W	R/W	R/W	R/W	-	-	R/W
Initial value				0	0	0	0	0	-	0	0

[bit 7:5] MD2, MD1, MD0: Operation mode setting bits

These bits set an operation mode.

"0b000": Sets operation mode 0 (async normal mode).

"0b001": Sets operation mode 1 (async multiprocessor mode).

"0b010": Sets operation mode 2 (clock sync mode).

"0b011": Sets operation mode 3 (LIN communication mode).

"0b100": Sets operation mode 4 (I²C mode).

This section explains the registers and their operation in operation mode 3 (LIN communication mode).

Bit 7	Bit 6	Bit 5	Description
0	0	0	Operation mode 0 (async normal mode)
0	0	1	Operation mode 1 (async multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I ² C mode)

* This section explains the registers in operation mode 3.

<Notes>

- Any bit setting other than above is inhibited.
- To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.
- After the operation mode has been switched, set each register correctly.

[bit 4] WUCR: Wake-up control bit

Selects a pin to be used for an external interrupt.

If this bit is set to "0": The INT pin is set as an external interrupt pin.

If set to "1": The SIN pin is set as an external interrupt pin.

Bit	Description
0	Disables the Wake-up function.
1	Enables the Wake-up function.

[bit 3] SBL: Stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

If SBL="0" and ESCR:ESBL="0" are set: One (1) stop bit is set.

If SBL="1" and ESCR:ESBL="0" are set: Two (2) stop bits are set.

If SBL="0" and ESCR:ESBL="1" are set: Three (3) stop bits are set.

If SBL="1" and ESCR:ESBL="1" are set: Four (4) stop bits are set.

Bit	Description	
0	ESCR.ESBL=0	1 bit
	ESCR.ESBL=1	3 bits
1	ESCR.ESBL=0	2 bits
	ESCR.ESBL=1	4 bits

<Notes>

- In receive operation, only the first bit of the stop bit data is detected.

- Always set this bit when transmission is disabled (SCR:TXE=0).

[bit 2:1] Reserved bit

This is an undefined bit. The read value is "0". Be sure to write "0".

[bit 0] SOE: Serial data output enable bit

This bit enables or disables a serial data output.

Bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

<Note>

If this bit is used as the SOUT pin, the GPIO must also be set.

6.3. Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmission/reception state, check the Receive Error flag, detect an LIN Break field, and clear the Receive Error flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	REC	-	LBD	FRE	ORE	RDRF	TDRE	TBI		(ESCR)	
Attribute	R/W	-	R/W	R	R	R	R	R			
Initial value	0	-	0	0	0	0	1	1			

[bit 15] REC: Receive Error flag clear bit

This bit clears the FRE and ORE flags of the Serial Status Register (SSR).

- If this bit is set to "1", the error flag is cleared.
- This bit has no effect if set to "0".

"0" is always read during reading.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Clears the Receive Error flag (FRE, ORE).	

[bit 14] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 13] LBD: LIN Break field detection flag bit

This bit shows a detection of LIN Break field.

When 11-bit wide or more of serial input (SIN) are "LOW", the LBD bit is set to "1". If the LIN Break field interrupt enable bit (LBIE) is "1" during this time, a status interrupt occurs.

When read:

If this bit is "1": An LIN Break field has been detected.

If this bit is "0": An LIN Break field has not been detected.

When written:

If this bit is set to "0": The LBD bit is cleared to "0".

If this bit is set to "1": No effect.

Bit	Description	
	During writing	During reading
0	Clears the LBD flag.	A Break field was not detected.
1	No effect.	A Break field was detected.

<Note>

If a read-modify-write instruction is issued, "1" is read.

[bit 12] FRE: Framing error flag bit

- If a framing error occurs during data reception, this bit is set to "1". If the REC bit of Serial Status Register (SSR) is set to "1", this flag is cleared.
- If the FRE and RIE bits are "1", a receive interrupt request is output.
- If this flag is set, data of the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No framing error occurred.
1	A framing error occurred.

[bit 11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". If the REC bit of Serial Status Register (SSR) is set to "1", this flag is cleared.
- If the ORE and RIE bits are "1", a receive interrupt request is output.
- If this flag is set, data in the Receive Data Register (RDR) is invalid.
- If this flag is set when receive FIFO is used, the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

Bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit 10] RDRF: Receive data full flag bit

- This flag shows the state of Receive Data Register (RDR).
- When the receive data is loaded in the RDR, this bit is set to "1". When the Receive Data Register (RDR) is read, this bit is cleared to "0".
- If the RDRF and RIE bits are "1", a receive interrupt request is output.
- If receive FIFO is used, the RDRF bit is set to "1" when the preset amount of data is received in receive FIFO.
- If receive FIFO is used, this bit is cleared to "0" when receive FIFO is emptied.

Bit	Description
0	The Receive Data Register (RDR) is empty.
1	The Receive Data Register (RDR) contains data.

[bit 9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If the transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When the data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not contain the valid data.
- If the TDRE and TIE bits are "1", a transmit interrupt request is output.
- When the UPCL bit of Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/clear timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

Bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register (TDR) is empty.

[bit 8] TBI: Transmit bus idle flag bit

- This bit indicates that the LIN interface (ver. 2.1) is not transmitting data.
- When transmit data is written in the Transmit Data Register (TDR), this bit is set to "0".
- When the LIN Break field is set (SMR:LBR=1), this bit is set to "0".
- If the Transmit Data register (TDR) is empty (TDRE=1) and if no transmission is started, this bit is set to "1".
- If the Transmit Data Register is emptied after the LIN Break field has been transmitted, this bit is set to "1".
- If this bit is "1" and if a transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

Bit	Description
0	Data being transmitted
1	No data transmission

6.4. Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to enable/disable an LIN Break field interrupt, detect an LIN Break field, set an LIN Break field length and a Break delimiter length, and select a stop bit length.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SSR)		-	ESBL	-	LBIE	LBL1	LBL0	DEL1	DEL0	
Attribute			-	R/W	-	R/W	R/W	R/W	R/W	R/W	
Initial value			0	0	-	0	0	0	0	0	

[bit 7] Reserved bit

This is an undefined bit. The read value is "0". Be sure to write "0".

[bit 6] ESBL: Extended stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

If SBL="0" and ESCR:ESBL="0" are set: One (1) stop bit is set.

If SBL="1" and ESCR:ESBL="0" are set: Two (2) stop bits are set.

If SBL="0" and ESCR:ESBL="1" are set: Three (3) stop bits are set.

If SBL="1" and ESCR:ESBL="1" are set: Four (4) stop bits are set.

Bit	Description	
0	SMR.SBL=0	1 bit
	SMR.SBL=1	2 bits
1	SMR.SBL=0	3 bits
	SMR.SBL=1	4 bits

<Notes>

- In receive operation, only the first bit of the stop bit data is detected.

- Always set this bit when transmission is disabled (TXE=0).

[bit 5] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 4] LBIE: LIN Break field detect interrupt enable bit

This bit enables or disables an LIN Break field detect interrupt.

If the LIN Break field detect flag (LBD) is "1", a receive interrupt occurs when an interrupt is enabled (LBIE=1).

Bit	Description
0	Disables an LIN Break field detect interrupt.
1	Enables an LIN Break field detect interrupt.

[bit 3:2] LBL1/0: LIN Break field length select bits (valid in master mode only)

- These bits set an LIN Break field generation time (in number of bits).
- This bit must be set before the LBR bit of Serial Control Register (SCR) is set to "1" (for LIN Break field transmission).
- An LIN Break field is always detected at the 11th bit in the slave mode operation regardless of this bit setting.

Bit 3	Bit 2	Description
0	0	13 bits long
0	1	14 bits long
1	0	15 bits long
1	1	16 bits long

<Note>

This bit setting is valid in the master mode operation only (SMR:MS="0").

[bit 1:0] DEL1/0: LIN Break delimiter length select bits (valid in master mode only)

- These bits set an LIN Break delimiter length (in number of bits).
- These bits must be set before the LBR bit of Serial Control Register (SCR) is set to "1" (for LIN Break field transmission).

Bit 1	Bit 0	Description
0	0	1 bit long
0	1	2 bits long
1	0	3 bits long
1	1	4 bits long

<Note>

This bit setting is valid in the master mode operation only (SMR:MS="0").

6.5. Receive Data Register/Transmit Data Register (RDR/TDR)

The Receive and Transmit Data Registers are allocated at the same address. This register functions as the Receive Data Register when data is read from it. This register functions as the Transmit Data Register when data is written in it.

■ Receive Data Register (RDR)

bit	15	...	8	7	6	5	4	3	2	1	0
Field				D7	D6	D5	D4	D3	D2	D1	D0
Attribute				R	R	R	R	R	R	R	R
Initial value				0	0	0	0	0	0	0	0

The Receive Data Register (RDR) is a data buffer register for serial data reception.

- When serial data signals are sent to the Serial Input pin (SIN pin), they are converted by a shift register and stored in the Receive Data Register (RDR).
- When the receive data is stored in the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is set to "1". If a receive interrupt is enabled (SSR:RIE=1), a receive interrupt request is generated.
- The Receive Data Register (RDR) must be read only when the receive data full flag bit (SSR:RDRF) is "1". When data is read from the Serial Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a receive error occurs (when SSR:ORE or FRE is "1"), data in the Receive Data Register (RDR) becomes invalid.

<Notes>

- If receive FIFO is used and if the preset amount of data is received in receive FIFO, the RDRF bit is set to "1".
- If receive FIFO is used and if this buffer is emptied, the RDRF bit is cleared to "0".
- If a receive error occurs when receive FIFO is used (SSR:ORE or FRE is "1"), the receive FIFO enable bit is cleared and the receive data is not stored in receive FIFO.

■ Transmit Data Register (TDR)

bit	15	...	8	7	6	5	4	3	2	1	0
Field			D7	D6	D5	D4	D3	D2	D1	D0	
Attribute			W	W	W	W	W	W	W	W	W
Initial value			1	1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the transmit shift register. Then, the data is converted into serial data, and output at the serial data output pin (SOUT pin).
- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written in the buffer. If a transmit interrupt is enabled, a transmit interrupt occurs. The next transmit data must be written only after the transmit interrupt has occurred or when the transmit data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and transmit FIFO is disabled or transmit FIFO is full, no transmit data can be written in the Transmit Data Register (TDR).

<Notes>

- The Transmit Data Register is a write-only register. While the Receive Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instructions cannot be used.
- For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

6.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks. Also, an external clock can be selected as the clock source of the reload counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXT	(BGR1)						(BGR0)								
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.
- The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.
- The EXT bit (bit 15) specifies to use the clock source of reload counter as the internal clock or use an external clock. If EXT=0 is set, an internal clock is used. If EXT=1 is set, an external clock is used.

[bit 15] EXT: External clock select bit

Bit	Description
0	Uses the internal clock.
1	Uses an external clock.

[bit 14:8] BGR1: Baud Rate Generator Register 1

Bit 14:8	Description
Write	Write data in reload counter bit 8 to 14.
Read	Reads the BGR1 set value.

[bit 7:0] BGR0: Baud Rate Generator Register 0

Bit 7:0	Description
Write	Write data in reload counter bit 0 to 7.
Read	Reads the BGR0 set value.

<Notes>

- Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) in 16-bit data access mode.
- If the current values of Baud Rate Generator Registers (BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/0 set values and execute the programmable clear (UPCL).
- If the reload value is even, the "LOW" signal width of serial clock is longer than the "HIGH" signal width for a single cycle of bus clock. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.
- Set the reload value to 3 or more. Note that data may not be received normally due to the baud rate error and reload value setting.
- When the baud rate generator is operating and if you need to switch to the external clock (EXT=1), first set the BGR1 and BGR0 bits (baud rate generators 1 and 0) to "0". Then, execute the programmable clear instruction (UPCL) and select the external clock (EXT=1).

6.7. FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select transmit or receive FIFO, enable transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL		(FCR0)	
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W			
Initial value	0	0	-	0	0	1	0	0			

[bit 15:14] FTST1, FTST0: FIFO test bits

They are FIFO Test bits.

They must always be set to "0".

Bit 15:14	Description
0	Disables the FIFO test.
1	Enables the FIFO test.

<Note>

If this bit is set to "1", the FIFO test is executed.

[bit 13] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 12] FLSTE: Re-transmit data lost detect enable bit

This bit enables the FLST bit detection.

If set to "0": The FLST bit detection is disabled.

If set to "1": The FLST bit detection is enabled.

Bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

<Note>

If you wish to set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".

[bit 11] FRIIE: Receive FIFO idle detect enable bit

This bit sets to detect the receive idle state if receive FIFO contains valid data for more than 8-bit hours. If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt is generated when the receive idle state is detected.

If set to "0": The receive idle state detection is disabled.

If set to "1": The receive idle state detection is enabled.

Bit	Description
0	Disables the receive FIFO idle detection.
1	Enables the receive FIFO idle detection.

<Note>

In case of using Receive FIFO, set this bit to "1".

[bit 10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. If the Transmit Interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is cleared when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

Bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

<Notes>

- If the FBYTE (for transmission) is "0", this bit cannot be set to "0".
- If this bit is "0", the FSEL bit state cannot be changed.
- If this bit is set to "1", it has no effect on the operation.
- If a read-modify-write instruction is issued, "1" is read.

[bit 9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

Bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit 8] FSEL: FIFO select bit

This bit selects the transmit or receive FIFO.

If set to "0", transmit FIFO is assigned FIFO1, and receive FIFO is assigned FIFO2.

If set to "1", transmit FIFO is assigned FIFO2, and receive FIFO is assigned FIFO1.

Bit	Description
0	Transmit FIFO:FIFO1; Receive FIFO:FIFO2
1	Transmit FIFO:FIFO2; Receive FIFO:FIFO1

<Notes>

- This bit is not cleared by FIFO reset (FCL2=1, FCL1=1).
- To change this bit state, first disable the FIFO operation (FE2=0, FE1=0).

6.8. FIFO Control Register 0 (FCR0)

FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15	...	8	7	6	5	4	3	2	1	0	
Field	(FCR1)				-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute					-	R	R/W	W	R/W	R/W	R/W	R/W
Initial value					-	0	0	0	0	0	0	0

[bit 7] Unused bit

This bit value is undefined when read.

This bit has no effect when written.

[bit 6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- The FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in FIFO.

The FLST bit is cleared when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten.

Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in FIFO again.

Bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit 5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.

When the re-transmission setting has finished, this bit is set to "0".

Bit	Description
0	Not reloaded
1	Reloaded

<Notes>

- If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.
- When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".
- After you have set the TIE and TBIE bits to "0", set this bit to "1". After you have enabled transmit FIFO, set the TIE and TBIE bits to "1".

[bit 4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer is saved before transmission and if the FLST bit is "0", data can be re-transmitted even when a communication error or others occur.

If set to "1": The current read pointer value is saved.

If set to "0": No effect.

Bit	Description
0	Not saved
1	Saved

<Note>

This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".

[bit 3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST2 bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO2 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO2.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The valid data count of the FBYTE2 register is set to "0".

[bit 2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 state.

If this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST1 bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO1 is reset.	

<Notes>

- Disable the transmission and reception first, and then reset FIFO1.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The valid data count of the FBYTE1 register is set to "0".

[bit 1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the LIN interface (ver. 2.1) is enabled to transmit data (TXE=1). During this time, set both TIE and TBIE bits to "0". Then, set this bit to "1" and set both TIE and TBIE bits to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (TDRE=1).
- If FIFO2 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

Bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit 0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If FIFO1 is set as transmit FIFO and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the LIN interface (ver. 2.1) is enabled to transmit data (TXE=1). During this time, set both TIE and TBIE bits to "0". Then, set this bit to "1" and set both TIE and TBIE bits to "1".
- If receive FIFO is selected by the FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (TDRE=1).
- If FIFO1 is used as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and no valid data exists in receive FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

Bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

6.9. FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a receive interrupt when a certain number of data sets is received in the receive FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count of FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 6-3 Display of data count

FSEL	FIFO selection	Data count display
0	FIFO2:Receive FIFO, FIFO1:Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2:Transmit FIFO, FIFO1:Receive FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
 - Set a data count to flag a receive interrupt for the FBYTE register of receive FIFO. If this transfer data count matches the FBYTE register display, the interrupt flag (RDRF) is set to "1".
 - If both conditions below are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
 - The receive FIFO idle detect enable bit (FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to "0". If data remains in receive FIFO and if receive FIFO is enabled, the data counting is restarted.

FBYTE2, FBYTE1: FIFO2 data count display bit, FIFO1 data count display bit

During writing	Sets the transfer data count.
During reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in FIFO but not transmitted yet
 During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
 During reception: Set the data count to generate a receive interrupt.

<Notes>

- Set "8'h00" in the FBYTE register of transmit FIFO.
 - Set data equal to or greater than "1" in the FBYTE register of receive FIFO.
 - This state can be changed only after the data transmission or reception has been disabled.
 - A read-modify-write instruction cannot be used for this register.
 - Any setting exceeding the FIFO capacity is inhibited.
 - When all the following requirements are met, the receive data full flag bit (SSR:RDRF) is not set to "1" even though the effective data of FBYTE setting number exist in the receive FIFO. If the FBYTE register is set to "2" or greater, this operation will not occur.
 - FBYTE is set to "1".
 - The effective data count is "1", same as the number specified in FBYTE register.
 - When the multi function serial interface macro receives the data, and writes received data in the reception FIFO, the data of the reception FIFO are read at the same time.
- However, after that, the receive data full flag bit (SSR:RDRF) will be set to "1" at any of the following conditions.
- The next data is received.
 - The receive idle state of 8 bits or longer is detected when the receive FIFO idle is enabled (FCR:FRIIE=1)

Chapter: I²C Interface (I²C Communications Control Interface)

This chapter describes the I²C function supported in operation mode 4 of the multifunctional serial interface.

1. Overview of I²C Interface (I²C Communications Control Interface)
2. I²C Interface interrupt
3. Dedicated Baud Rate Generator
4. I²C communication operation flowchart examples
5. I²C Interface Registers

1. Overview of I²C Interface (I²C Communications Control Interface)

The I²C interface (I²C communications control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. It also has transmit/receive FIFO (up to 128 × 9 bits each) ^{*1}installed.

■ Functions of I²C interface (I²C communications control interface)

		Function
1	Data buffer	<ul style="list-style-type: none"> Full duplex double buffer (when FIFO is not used) Transmit/receive FIFO (max 128 × 9 bits each) ^{*1} (when FIFO is used)
2	Serial input	Removes noise up to 2 clocks in the bus clock for serial clock/serial data input.
3	Transfer mode	Synchronous
4	Baud rate	<ul style="list-style-type: none"> Complete with a dedicated baud rate generator (constructed with a 15-bit reload counter) The external clock can be adjusted with the reload counter.
5	Data length	8 bits
6	Signaling system	NRZ (Non Return to Zero)
7	Interrupt request	<ul style="list-style-type: none"> Receive interrupt Transmit interrupt Request of status interrupt/interrupt to ICU Transmit FIFO interrupt (when transmit FIFO is empty) For both transmission and reception, the extended intelligent I/O service (EIIOS) and the DMA support function are available.
8	I ² C	<ul style="list-style-type: none"> Master/slave transmission and reception functions Arbitration function Clock synchronization function Transmission direction detection function Function to generate and detect iteration start condition Bus error detection function General call addressing function 7-bit addressing as master/slave Generation of interrupt enabled during transmission or a bus error The 10-bit addressing function can be programmatically enabled.
9	FIFO	<ul style="list-style-type: none"> Transmit/receive FIFO installed (maximum capacity: 128 × 9 bits for transmit FIFO, 128 × 9 bits for receive FIFO) ^{*1} Transmit FIFO or receive FIFO can be selected. Transmit data can be resent. Receive FIFO interrupt timing can be changed via software. FIFO resetting is supported independently.

*1 : The FIFO capacity size varies from model type to model type.

2. I²C Interface interrupt

I²C interface interrupt request is generated due to the following factors.

- After transmission/reception of the first byte and after data transmission/reception is completed
- Stop condition
- Iteration start condition
- FIFO transmit data request
- FIFO receive data completed

■ I²C Interface Interrupt

Table 2-1 shows the interrupt control bits and interrupt causes for the I²C interface.

Table 2-1 Interrupt control bits and interrupt causes for the I²C interface

Interrupt type	Interrupt request flag bit	Flag register	Interrupt cause	Interrupt cause enable bit	Operation to clear interrupt request flag
Status	INT	IBCR	The first byte has been transmitted/received ^{*1} (except for master operation when SSR:DMA=1)	IBCR:INTE	Setting the interrupt flag bit (IBCR:INT) to "0"
			Data has been transmitted/received ^{*1} (When SSR:DMA=0)		
			Detection of a bus error		
			Detection of arbitration lost		
			Detection of reserved address		
			Reception of NACK		
			Receive FIFO being full during reception as a slave (When SSR:DMA=0)		Setting IBCR:INT to "0" after reading received data until receive FIFO is emptied
SPC	IBSR		Stop condition	IBCR:CNDE	Setting SPC to "0"
			Detection of iteration start		Setting RSC to "0"
Reception	RDRF	SSR	Reception of reserved address	SMR:RIE	Reading from the received data register (RDR)
			Completion of data reception		
			Reception of a data volume matching the value set for FBYTE.		
			Detection of reception idling when FRIIE="1"		
	ORE	SSR	Overrun error		Setting the reception error flag bit (SSR:REC) to "1"
Transmission	TDRE	SSR	The Transmit Data Register is empty.	SMR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) ^{*2}
			Setting the transmit buffer empty flag set bit (SSR:TSET) to "1"		
	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit is set to "0" or transmit FIFO is full.
	TBI (SSR: DMA=1)	SSR	No transmission operation	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) ^{*3}
	Setting the transmit buffer empty flag set bit (SSR:TSET) to "1"				

*1 : If normal data can be transmitted/received and SSR:TDRE is "0", no interrupt is generated. This to support DMA transfers.

To generate an IBCR:INT flag at a time of data transmission/reception, the SSR:TDRE bit needs to be set to "1" before the IBCR:INT flag is set.

*2 : Be sure to check that the SSR:TDRE bit is set to "0" and then set the SMR:TIE bit to "1".

*3 : Be sure to check that the SSR:TBI bit is set to "0" and then set the SSR:TBIE bit to "1".

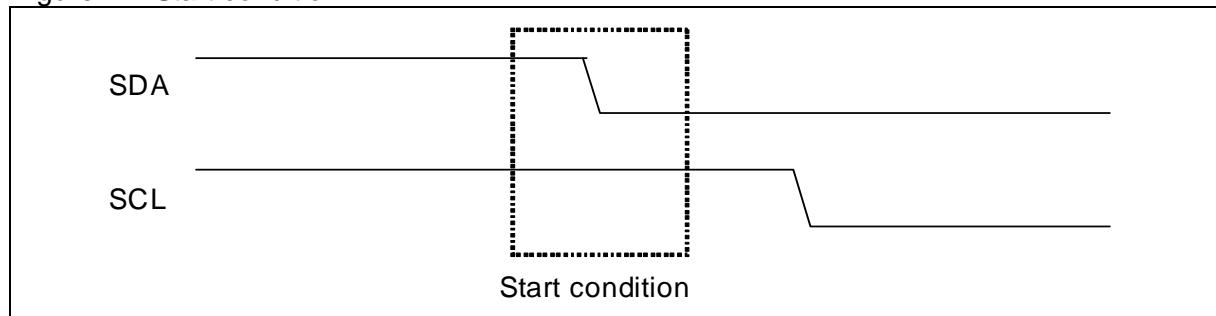
2.1. I²C interface operation

The I²C interface performs communications using two two-way bus lines, a serial data line (SDA) and a serial clock line (SCL).

■ I²C bus start condition

The following shows the I²C bus start condition.

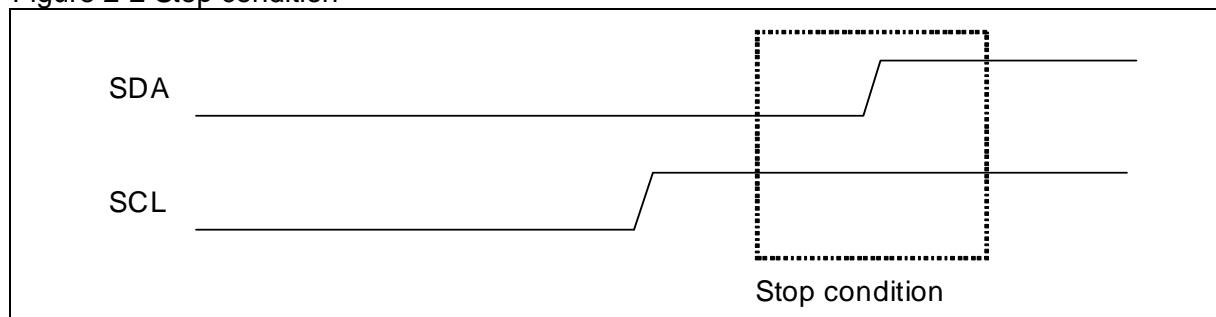
Figure 2-1 Start condition



■ I²C bus stop condition

The following shows the I²C bus stop condition.

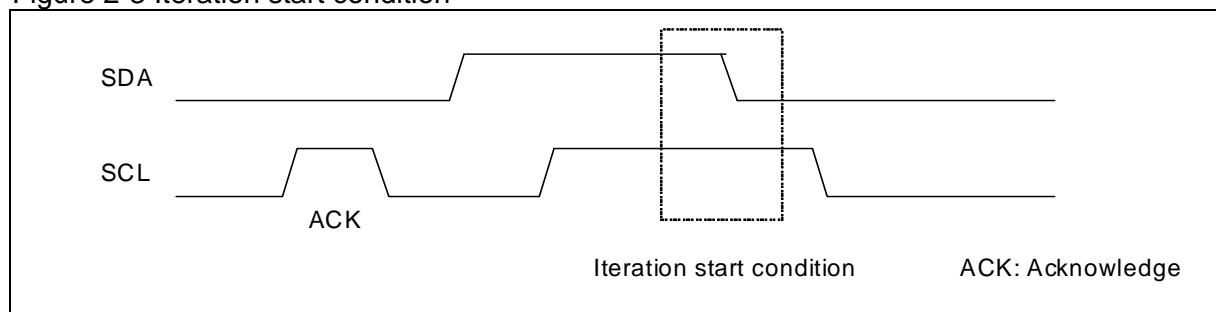
Figure 2-2 Stop condition



■ I²C bus iteration start condition

The following shows the I²C bus iteration start condition.

Figure 2-3 Iteration start condition



2.2. Master mode

Master mode generates the start condition on the I²C bus and outputs clocks to the I²C bus. When the MSS bit in the IBCR register is set to "1" while the I²C bus is in idle state (SCL=HIGH, SDA=HIGH), master mode is activated, causing the ACT bit in the IBCR register to be set to "1".

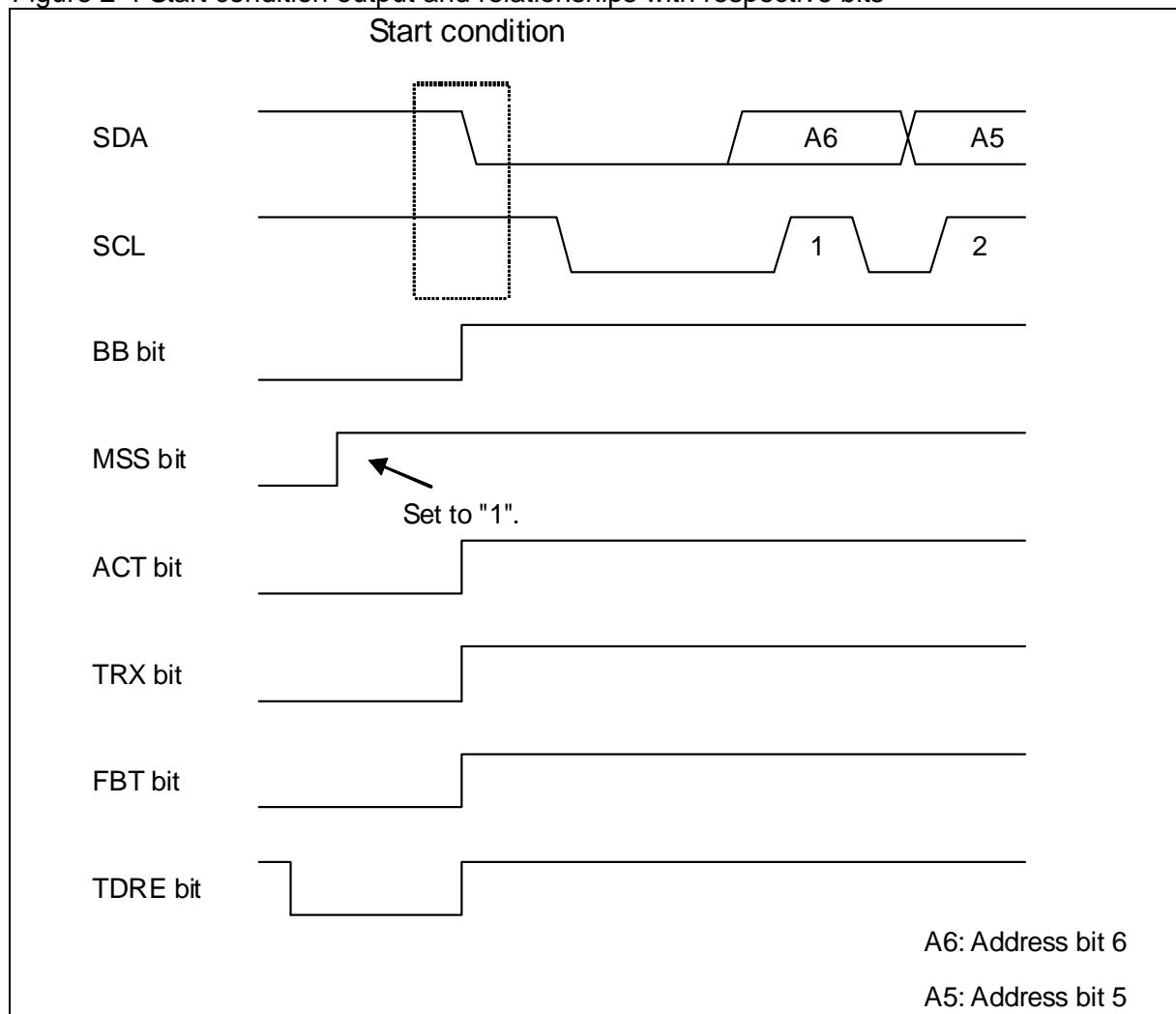
■ Generating start condition

The start condition is generated under the following condition.

- When SDA="H", SCL="H", ISMK:EN="1" and IBSR:BB="0", the IBCR:MSS bit is set to "1".

Outputting the start condition to the I²C bus causes the IBCR:ACT bit to be set to "1". After that, when the start condition is received, the IBSR:BB bit is set to "1" to indicate that the I²C bus is carrying out communications. (See Figure 2-4.)

Figure 2-4 Start condition output and relationships with respective bits



<Note>

In operation mode 4 (I²C mode), the bus clock is used at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.

■ Slave address output

Outputting the start condition causes data that are set in the TDR register to be output as the address, starting with bit 7. When FIFO is enabled, the datum in the TDR register that is written the earliest is output. Bit 0 is used as the data direction bit (R/W). When the data direction bit (R/W) is "0", it indicates that data flow in the write direction (from the master to a slave). Set the address to the TDR register before setting the IBCR:MSS to "1" or IBCR:SCC to "1".

For the output timing of the address and the data output direction, see Figure 2-5, Figure 2-6.

Figure 2-5 Address and data direction (when FIFO is disabled)

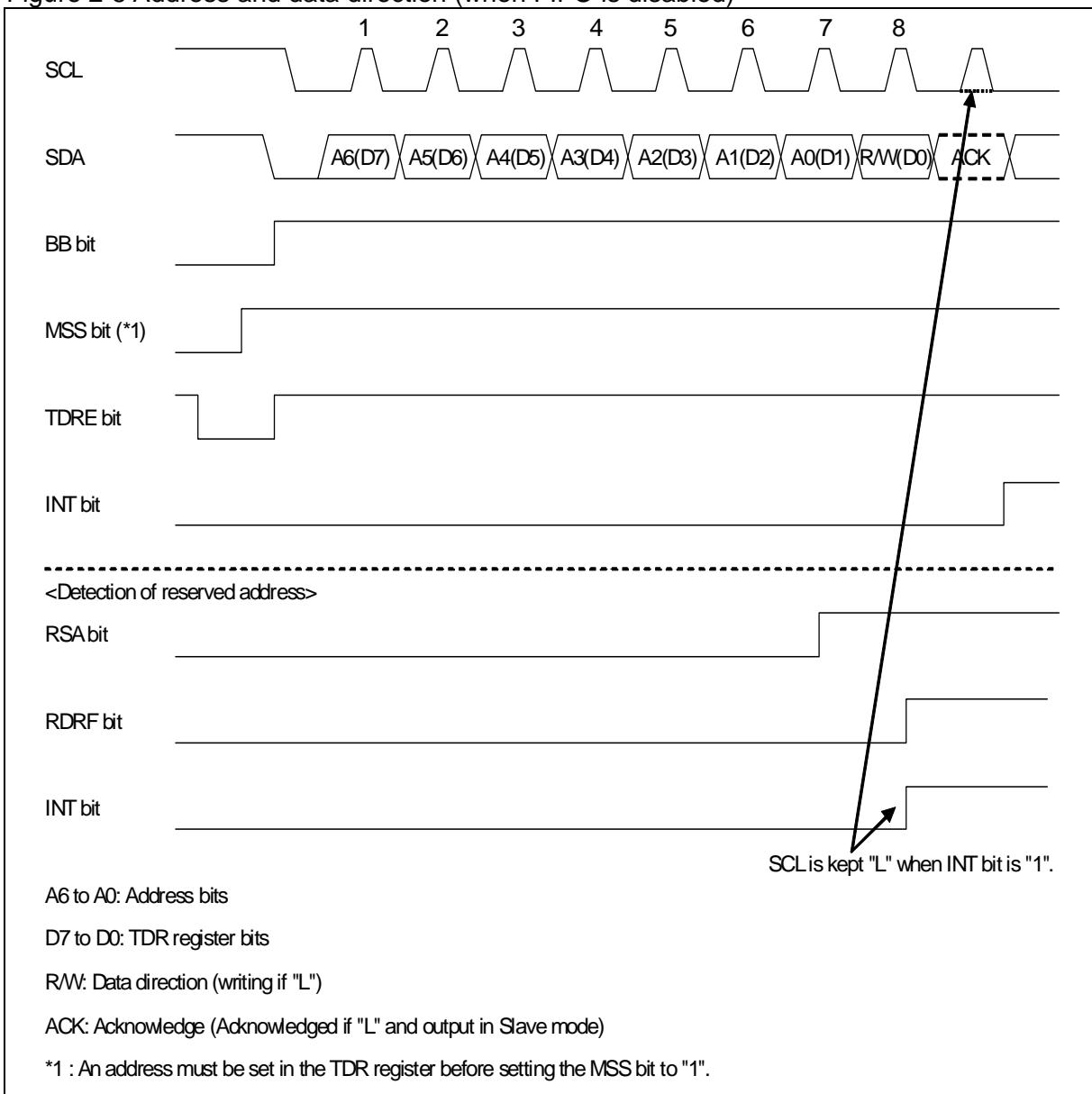
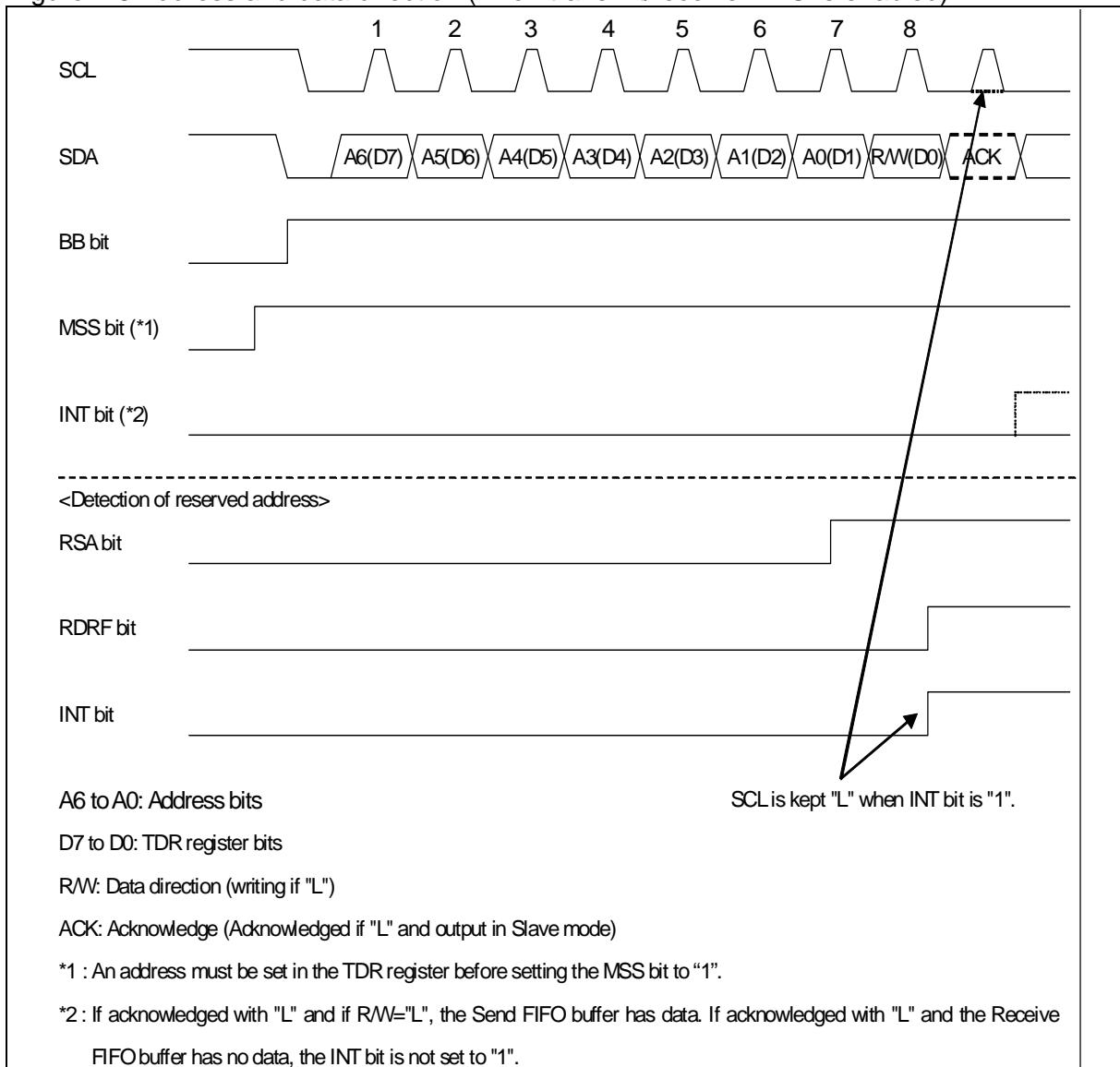


Figure 2-6 Address and data direction (when transmit/receive FIFO is enabled)



■ Acknowledgement reception by first byte transmission

When the data direction bit (R/W) is output, the I²C interface receives acknowledgement from a slave. The following lists operations to enable/disable FIFO.

**Table 2-2 Operations after acknowledgement reception with DMA mode disabled
(IBSR:RSA bit="0", SSR:DMA bit="0")**

Transmit FIFO	Receive FIFO	Transmit FIFO status	Receive FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
Disable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data	-	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	
Enable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
Enable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data	-	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	

**Table 2-3 Operations after acknowledgement reception with DMA mode enabled
(IBSR:RSA bit="0", SSR:DMA bit="1")**

Transmit FIFO	Receive FIFO	Transmit FIFO status	Receive FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
Disable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data	0	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	
Enable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
Enable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data	0	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	

● When DMA mode is disabled (SSR:DMA=0)

To disable FIFO (To disable both transmit FIFO and receive FIFO)

- When the IBSR:RSA bit is set to "0", after receiving acknowledgement, the interface sets the interrupt flag (IBCR:INT) to "1" if the SSR:TDRE bit is set to "1" and waits while maintaining SCL at LOW. Writing "0" to the interrupt flag sets the interrupt flag to "0", which releases wait. If the SSR:TDRE bit is set to "0", the interface generates a clock on SCL upon reception of ACK without setting the interrupt flag to "1".
- When the IBSR:RSA bit is set to "1", after receiving a reserved address (before acknowledgement), the interface sets the interrupt flag (IBCR:INT) to "1" and waits while maintaining SCL at LOW. After reading from the RDR register, setting the IBCR:ACKE bit and transmit data and writing "0" to the interrupt flag causes the interrupt flag to be set to "0", which releases wait.
- The received acknowledgement is set to the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

To enable FIFO

- Before setting "1" to the IBCR:MSS bit, it is needed to set the following for FIFO.
 - When transmitting to a slave (the data direction bit="0"), data including the slave address must be set to transmit FIFO.
 - When receiving data from a slave (the data direction bit="1"), the FIFO Byte Register must be set with the number of data sets to be received, and dummy data must be written to the Transmit Data Register for the slave address, data direction bit and the data volume for the number of bytes to be received.
 - When the IBSR:RSA bit is set to "0", after receiving acknowledgement and if it is ACK, the interface transmits/receives data according to the data direction bit without setting the interrupt flag (IBCR:INT) to "1" (with no wait occurring). If it is NACK, the interface sets the interrupt flag (IBCR:INT) to "1", and waits while maintaining SCL at LOW.
 - The received acknowledgement is stored in the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

● When DMA mode is enabled (SSR:DMA=1)**To disable FIFO (To disable both transmit FIFO and receive FIFO)**

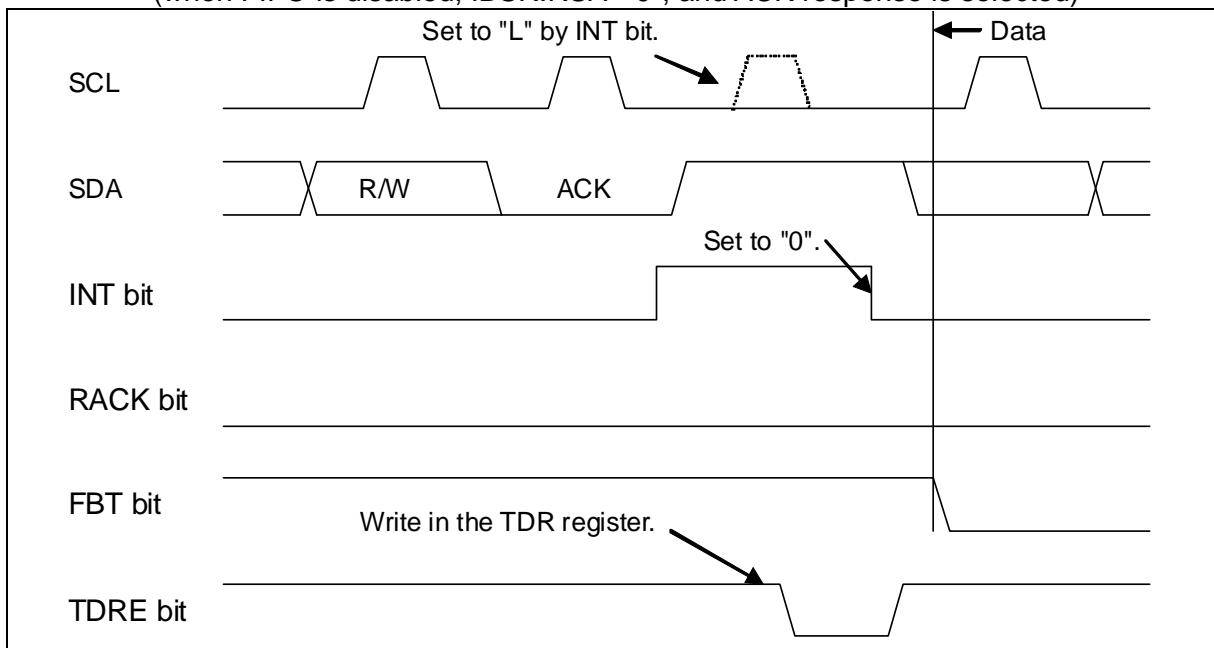
- When the IBSR:RSA bit is set to "0", after receiving acknowledgement, the interface sets the transmit bus idle flag (SSR:TBI) to "1" if the SSR:TDRE bit is set to "1" and waits while maintaining SCL at LOW. Writing data to be transmitted to the TDR register causes the transmit bus idle flag to be set to "0", which releases wait. If the SSR:TDRE bit is set to "0", the interface generates a clock on SCL upon reception of ACK without setting the transmit bus idle flag (SSR:TBI) to "1".
- When the IBSR:RSA bit is set to "1", after receiving a reserved address (before acknowledgement), the interface sets the interrupt flag (IBCR:INT) to "1" and waits while maintaining SCL at LOW. After reading from the RDR register, setting the IBCR:ACKE bit and transmit data and writing "0" to the interrupt flag causes the interrupt flag to be set to "0", which releases wait.
- The received acknowledgement is set to the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

To enable FIFO

- Before setting "1" to the IBCR:MSS bit, it is needed to set the following for FIFO.
 - When transmitting to a slave (the data direction bit="0"), data including the slave address must be set to transmit FIFO.
 - When receiving data from a slave (the data direction bit="1"), the FIFO Byte Register must be set with the number of data sets to be received, and dummy data must be written to the Transmit Data Register for the slave address, data direction bit and the data volume for the number of bytes to be received.
 - When the IBSR:RSA bit is set to "0", after receiving acknowledgement and if it is ACK, the interface transmits/receives data according to the data direction bit without setting the interrupt flag (IBCR:INT) to "1" (with no wait occurring). If it is NACK, the interface sets the interrupt flag (IBCR:INT) to "1", and waits while maintaining SCL at LOW.
 - The received acknowledgement is stored in the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

Figure 2-7 Acknowledgement

(when FIFO is disabled, IBSR:RSA="0", and ACK response is selected)



The following describes the wait timing for an address.

- After receiving acknowledgment if the IBSR:RSA bit is "0".
- Before receiving acknowledgment if the IBSR:RSA bit is "1".

Not dependent on the setting of the IBCR:WSEL.

Figure 2-8 Acknowledgement

(when FIFO is disabled, IBSR:RSA="0", and NACK response is selected)

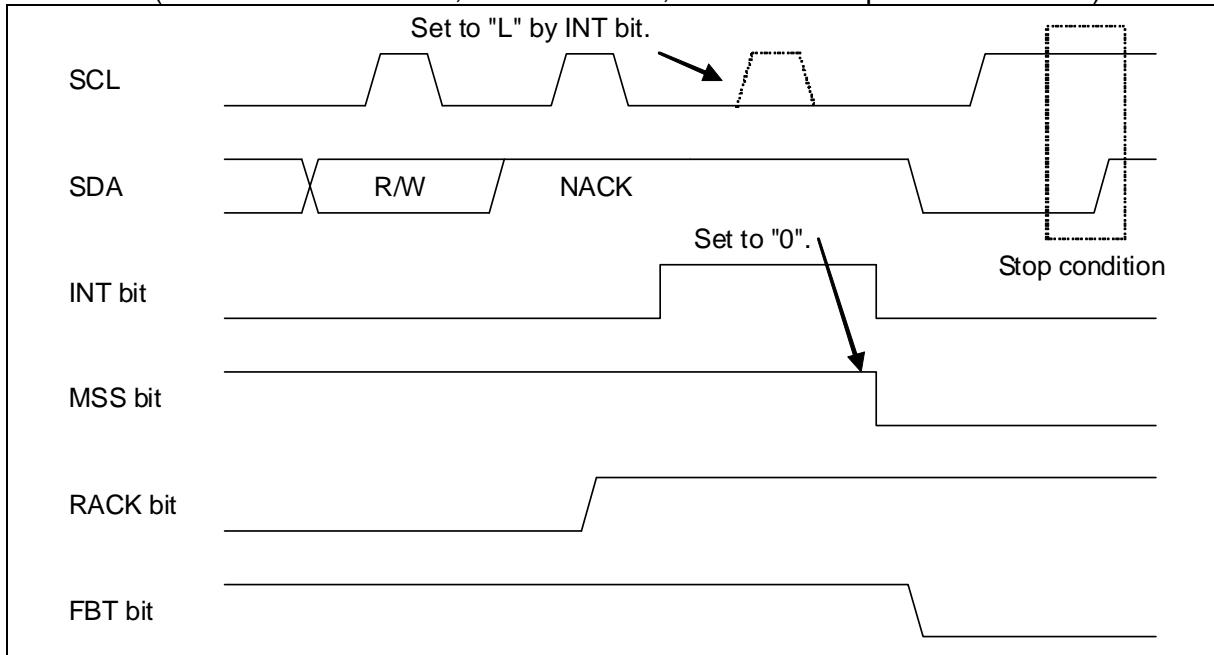


Figure 2-9 Acknowledgement

(when FIFO is disabled, IBSR:RSA="1", and ACK response is selected)

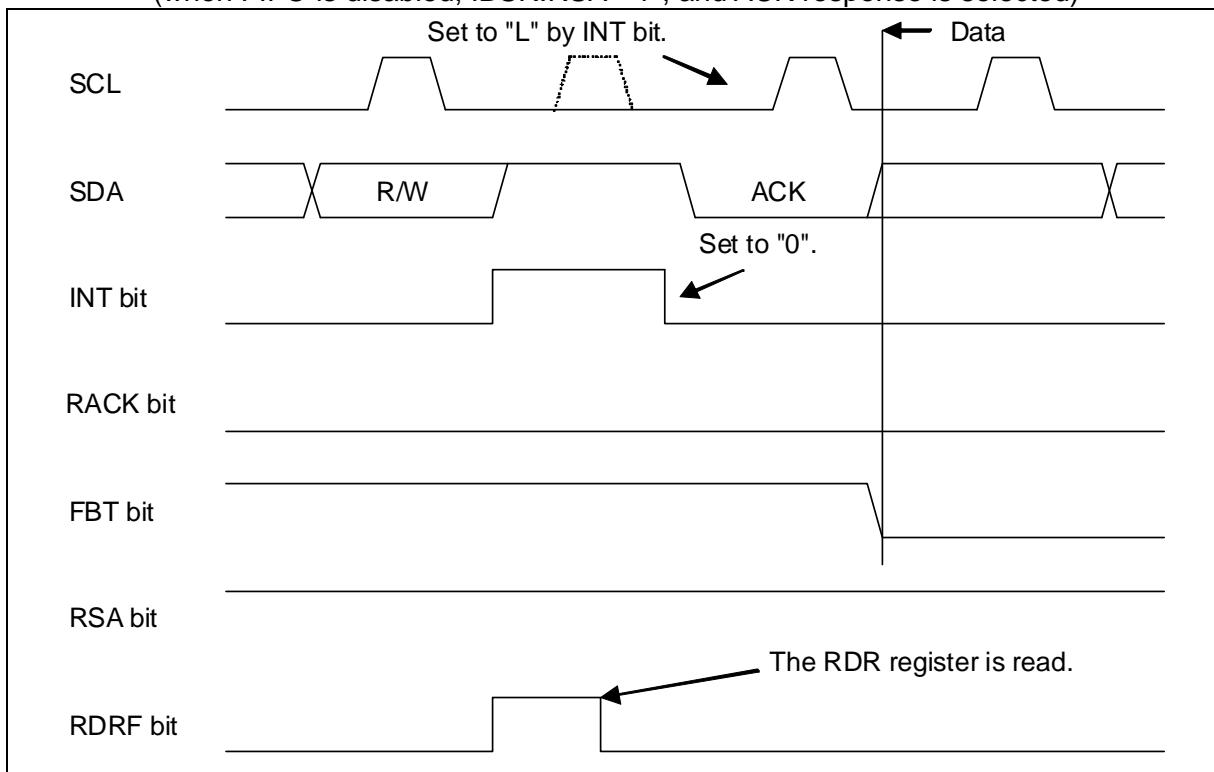


Figure 2-10 Acknowledgement

(when FIFO is disabled, IBSR:RSA="1", and NACK response is selected)

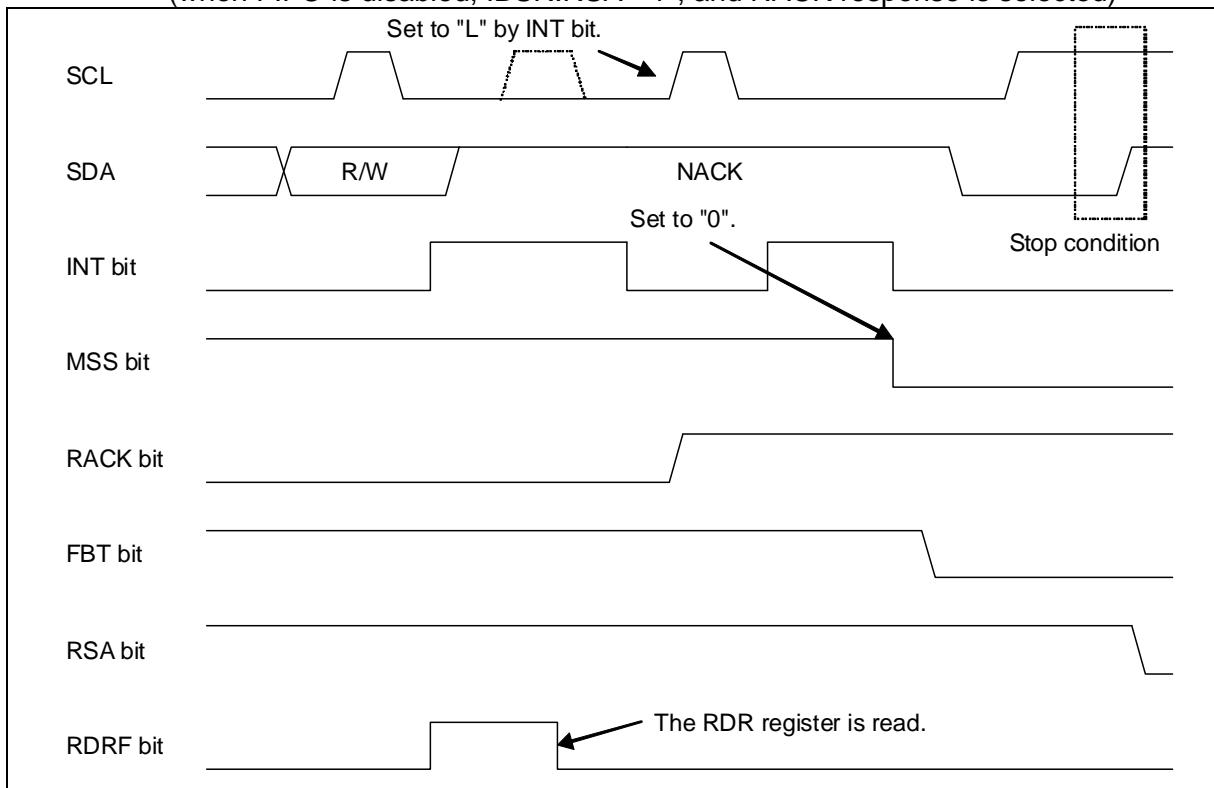
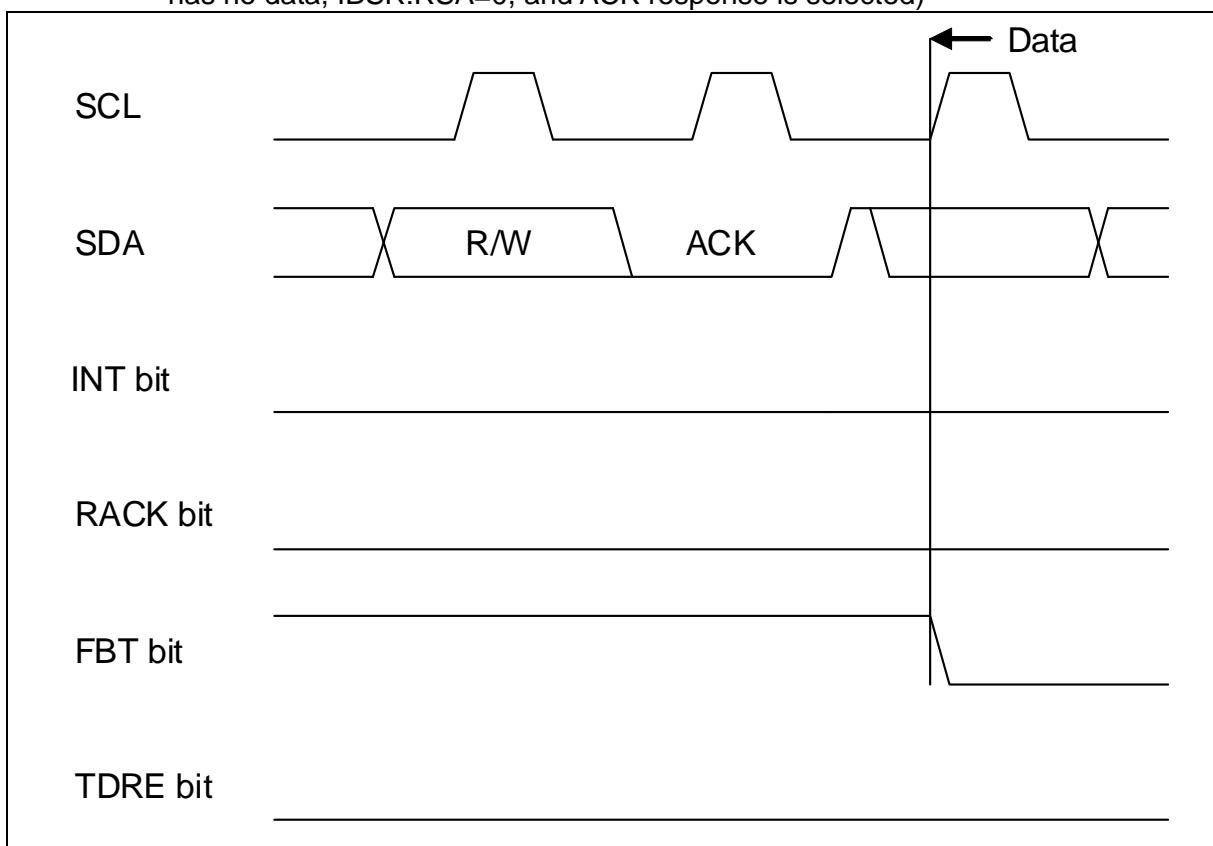


Figure 2-11 Acknowledgement (when FIFO is enabled, transmit FIFO has data, receive FIFO has no data, IBSR:RSA=0, and ACK response is selected)



■ Data transmission by the master

When the data direction bit (R/W) is set to "0", data are transmitted from the master. The slave gives response either with ACK or NACK for each one-byte transmission.

The following shows the wait timing by IBCR:WSEL setting.

Table 2-4 IBCR:WSEL bit status for master data transmission when DMA mode is disabled (SSR:DMA=0)

WSEL bit	Operation
0	<p><When FIFO is not used></p> <p>After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit or upon detection of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.</p> <p><When FIFO is used></p> <p>Starts the wait state by setting the interrupt flag (IBCR:INT) to "1" after acknowledgement upon detection of arbitration lost or when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).</p>
1	<p><When FIFO is not used></p> <p>After the second byte, after the master has transmitted one-byte data with "1" set for the SSR:TDRE bit or upon detection of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.</p> <p><When FIFO is used></p> <p>Starts the wait state by setting the interrupt flag (IBCR:INT) to "1" when data transmission has taken place after detection of arbitration lost or no more valid data in the Transmit Data Register (SSR:TDRE=1).</p>

Table 2-5 IBCR:WSEL bit status for master data transmission when DMA mode is enabled
(SSR:DMA=1)

WSEL bit	Operation
0	<p><When FIFO is not used> After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state after acknowledgement with "1" set for the SSR:TDRE bit.</p> <p><When FIFO is used> Starts the wait state by setting the transmit bus idle flag (SSR:TBI) to "1" after acknowledgment when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).</p>
1	<p><When FIFO is not used> After the second byte, after the master has transmitted one-byte data with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state.</p> <p><When FIFO is used> Starts the wait state by setting the transmit bus idle flag (SSR:TBI) to "1" after the master has transmitted one-byte data when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).</p>

In the following case, however, the interrupt flag (IBCR:INT) is set after acknowledgement, regardless of the IBCR:WSEL setting:

- If NACK is received when the stop condition (IBCR:MSS=0, ACT=1) is not set.

The following shows an example procedure for transmitting data to a slave.

● Data Transmission to slave when DMA mode is disabled (SSR:DMA=0)

1. To transmit data to an address other than the reserved:

- When transmit FIFO is disabled:
 1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
 2. ACK is received after the Slave Address setting is transmitted, and then the interrupt flag (IBCR:INT) is set to "1".
 3. Writes transmit data to the TDR register.
 4. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I²C bus.
 5. After transmitting one byte, the interrupt flag is set to "1", which puts the I²C bus in the wait state after receiving acknowledgement in case of IBCR:WSEL="0", and directly after transmitting one byte in case IBCR:WSEL=1. Repeats steps 3 to 5 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, another interrupt is generated after receiving acknowledgement and the bus enters the wait state.
 6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.
- When transmit FIFO is enabled:
 1. Writes an address for Slave Address (including the data direction bit) and transmit data to the TDR register.
 2. Writes "1" to the IBCR:MSS bit upon setting of the IBCR:WSEL bit.
 3. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I²C bus in the wait state. If ACK responses are received for all bytes, sets the interrupt flag to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I²C bus in the wait state.
 4. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

2. To transmit data to a reserved address:

- When transmit FIFO is disabled:
 1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
 2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
 3. Reads from the RDR register and confirms the reserved address.(*1)
 4. Writes transmit data to the TDR register.
 5. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I²C bus.
 6. After transmitting one byte, the interrupt flag is set to "1", which puts the I²C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL="0", and directly after transmitting one byte in case IBCR:WSEL=1. Repeats steps 4 to 6 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, another interrupt is generated after receiving acknowledgement and the bus enters the wait state.
 7. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

 - When transmit FIFO is enabled:
 1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
 2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
 3. Reads from the RDR register and confirms the reserved address.(*1)
 4. Writes all transmit data to the TDR register (until transmit FIFO becomes full if it is the case).
 5. If NACK is received during transmission, the interrupt flag (IBCR:INT) is set to "1" immediately after that to put the I²C bus in the wait state.
If ACK responses are received for all bytes, sets the interrupt flag to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I²C bus in the wait state.
 6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.
- *1 : When any one of the following conditions is met, the IBCR:ACKE and IBCR:WSEL bits must be set to "1" and to check which is needed for the next data, operation as a master or operation as a slave.
- Multi-master mode is activated and the reserved address is a general call.
 - Arbitration lost has been detected and the interface may operate as a slave.

● Data Transmission to slave when DMA mode is enabled (SSR:DMA=1)**1. To transmit data to an address other than the reserved:**

- When transmit FIFO is disabled:
 1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
 2. ACK is received after the Slave Address setting is transmitted, and then the transmit bus idle flag (SSR:TBI) is set to "1".
 3. Writes data to be transmitted to the TDR register to release the wait state of the I²C bus.
 4. After transmitting one byte, sets the transmit bus idle flag (SSR:TBI) to "1" to put the I²C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL="0", and directly after transmitting one byte in case of IBCR:WSEL=1.
 5. Writes data to be transmitted to the TDR register to release the wait state of the I²C bus.
 6. After transmitting one byte, sets the transmit bus idle flag to "1" to put the I²C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL="0", and directly after transmitting one byte in case of IBCR:WSEL=1. Repeats steps 6 to 7 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, the interrupt flag (IBCR:INT) is set to "1" after receiving acknowledgement and the bus enters the wait state.
 7. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"^{*2} to generate the stop condition or iteration start condition.

- When transmit FIFO is enabled:
 1. Writes an address for Slave Address (including the data direction bit) and transmit data to the TDR register.
 2. Writes "1" to the IBCR:MSS bit upon setting of the IBCR:WSEL bit.
 3. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I²C bus in the wait state. If ACK responses are received for all bytes, sets the transmit bus idle flag (SSR:TBI) to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I²C bus in the wait state.
 4. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"^{*2} to generate the stop condition or iteration start condition.

2. To transmit data to a reserved address:

- When transmit FIFO is disabled:
 1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
 2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
 3. Reads from the RDR register and confirms the reserved address.(^{*1})
 4. Writes transmit data to the TDR register.
 5. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I²C bus.
 6. After transmitting one byte, the interrupt flag is set to "1", which puts the I²C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL="0", and directly after transmitting one byte in case IBCR:WSEL=1.
 7. Writes data to be transmitted to the TDR register to release the wait state of the I²C bus.
 8. After transmitting one byte, sets the transmit bus idle flag to "1" to put the I²C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL="0", and directly after transmitting one byte in case of IBCR:WSEL=1. Repeats steps 7 to 8 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, the interrupt flag (IBCR:INT) is set to "1" after receiving acknowledgement and the bus enters the wait state.
 9. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"^{*2} to generate the stop condition or iteration start condition.

- When transmit FIFO is enabled:

 1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
 2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
 3. Reads from the RDR register and confirms the reserved address.*1)
 4. Writes all transmit data to the TDR register (until transmit FIFO becomes full if it is the case).
 5. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I²C bus in the wait state. If ACK responses are received for all bytes, sets the interrupt flag (IBCR:INT) to "1" according to the setting of IBCR:WSEL after the last byte is transmitted, which puts the I²C bus in the wait state.
 6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"^{*2} to generate the stop condition or iteration start condition.

*1 : When any one of the following conditions is met, the IBCR:ACKE and IBCR:WSEL bits must be set to "1" and to check which is needed for the next data, operation as a master or operation as a slave.

- Multi-master mode is activated and the reserved address is a general call.
- Arbitration lost has been detected and the interface may operate as a slave.

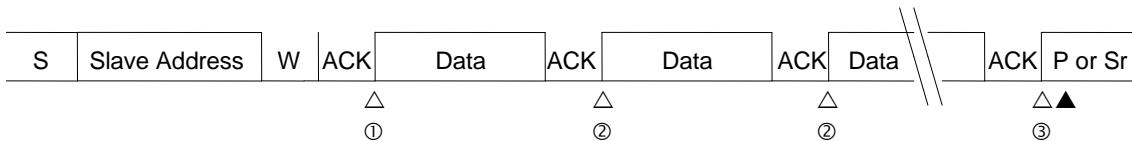
*2 : When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.

1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

<Notes>

- When seven-bit slave address detection is enabled (ISBA:SAEN="1"), it is prohibited to specify a seven-bit slave address in master mode.
 - To change the IBCR register during transmission/reception, do so when the interrupt flag (IBCR:INT) is "1".
 - If the IBCR:WSEL bit is changed, the update is used as a condition for generating the transmit bus idle flag (SSR:TBI) when the interrupt flag (IBCR:INT) is enabled and DMA mode is also enabled (SSR:DMA=1) for the next data.
 - The master operates as follows when transmit data are written to the TDR register during data transmission with SSR:TDRE set to "1" and an ACK response is detected.
 - When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) does not attain "1", and the written data are transmitted.
 - When DMA mode is enabled (SSR:DMA=1), the transmit bus idle flag (SSR:TBI) does not attain "1", and the written data are transmitted.
 - The master operates as follows when transmit data are written to the TDR register during data reception with SSR:TDRE set to "1" and an ACK response is detected.
 - When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) does not attain "1" and only SSR:RDRF attains "1" (when receive FIFO is enabled, and the number of bytes set in the FBYTE register have been received).
 - When DMA mode is enabled (SSR:DMA=1), the transmit bus idle flag (SSR:TBI) does not attain "1" and only SSR:RDRF attains "1" (when receive FIFO is enabled, and the number of bytes set in the FBYTE register have been received).
-

Figure 2-12 Master mode interrupt 1 by disabling FIFO
 (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent and an ACK is received.

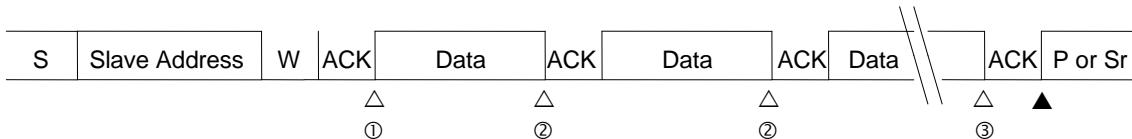
- The send data is written in the TDR register, and the INT bit is set to "0".

③ An interrupt occurs when a single byte is sent and an ACK is received.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-13 Master mode transmit interrupt 2 by disabling FIFO
 (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent.

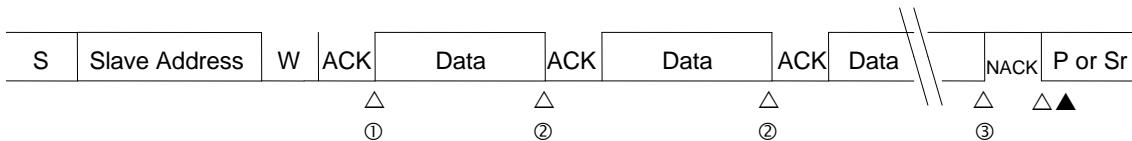
- An interrupt occurs when a single byte is sent.

③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-14 Master mode transmit interrupt 3 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent.

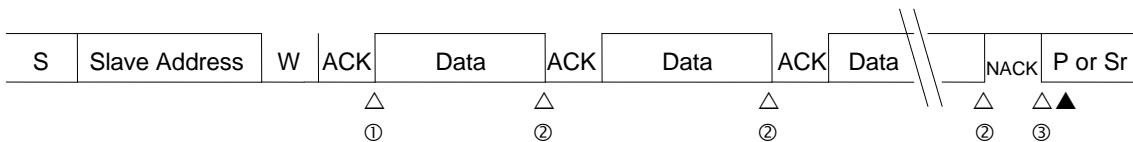
- The send data is written in the TDR register, and the INT bit is set to "0".

③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-15 Master mode transmit interrupt 4 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response during transmission)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

② An interrupt occurs when a single byte is sent.

- The send data is written in the TDR register, and the INT bit is set to "0".

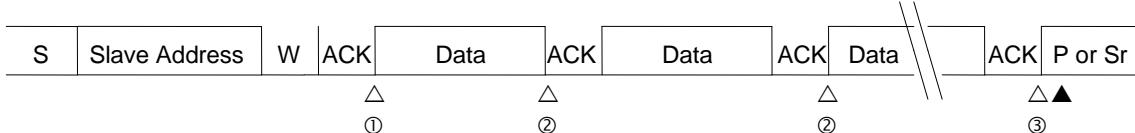
③ An interrupt occurs when a NACK is responded.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-16 Master mode transmit interrupt 5 by disabling FIFO

(SSR:DMA="0", IBCR:WSEL="1" -> "0", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the send buffer, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent.

- The send data is written in the send buffer, and both WSEL and INT bits are set to "0".

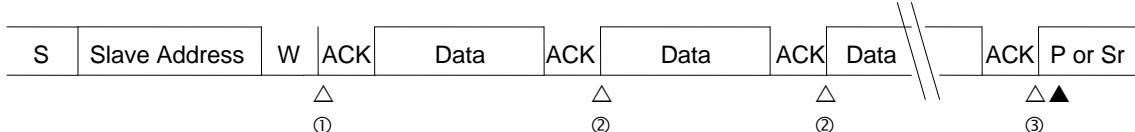
③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-17 Master mode interrupt 6 by disabling FIFO

(SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address (reserved address) is sent, a direction bit is sent, and an ACK is received.

- The send data is written in the TDR register, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent and an ACK is received.

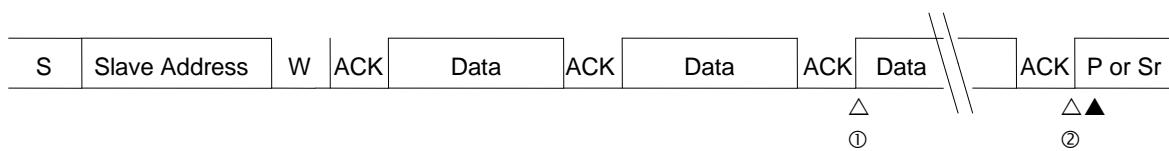
- The send data is written in the TDR register, and the INT bit is set to "0".

③ An interrupt occurs when a single byte is sent and an ACK is received.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-18 Master mode transmit interrupt 7 by enabling FIFO
(SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

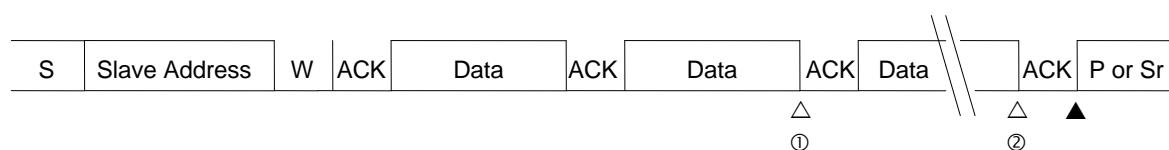
① An interrupt occurs if the Send FIFO buffer is emptied.

- The send data is written in the Send FIFO buffer, and INT bit is set to "0".

② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied) and an ACK is received.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-19 Master mode transmit interrupt 8 by enabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

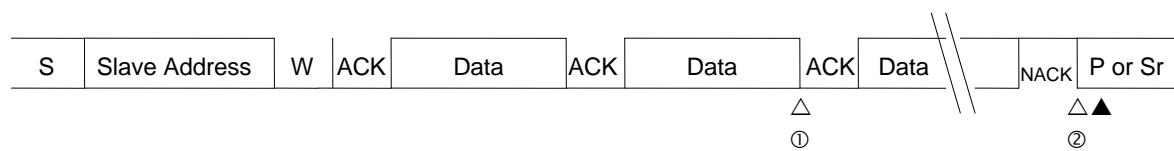
① An interrupt occurs if the Send FIFO buffer is emptied.

- The send data is written in the Send FIFO buffer, and INT bit is set to "0".

② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied).

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-20 Master mode transmit interrupt 9 by enabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

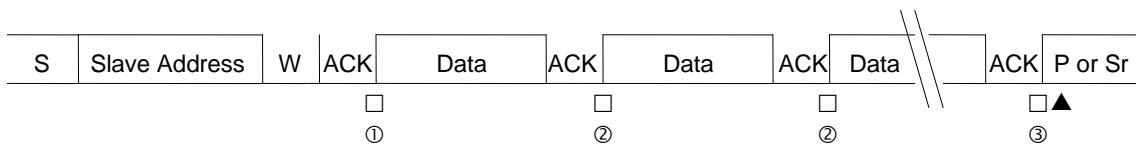
① An interrupt occurs if the Send FIFO buffer is emptied.

- The send data is written in the Send FIFO buffer, and INT bit is set to "0".

② An interrupt occurs when a NACK is responded.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-21 Master mode interrupt 10 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register.

② An interrupt occurs when a single byte is sent and an ACK is received.

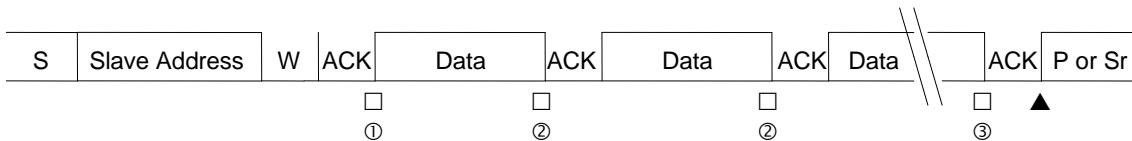
- The send data is written in the TDR register.

③ An interrupt occurs when a single byte is sent and an ACK is received.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (TBI) is set, the TDRE bit is set to "1".

Figure 2-22 Master mode transmit interrupt 11 by disabling FIFO
 (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register.

② An interrupt occurs when a single byte is sent.

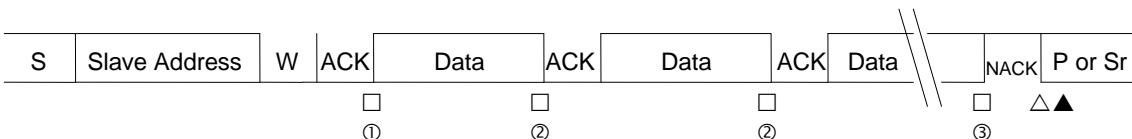
- The send data is written in the TDR register.

③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (TBI) is set, the TDRE bit is set to "1".

Figure 2-23 Master mode transmit interrupt 12 by disabling FIFO
 (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTF="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register.

② An interrupt occurs when a single byte is sent.

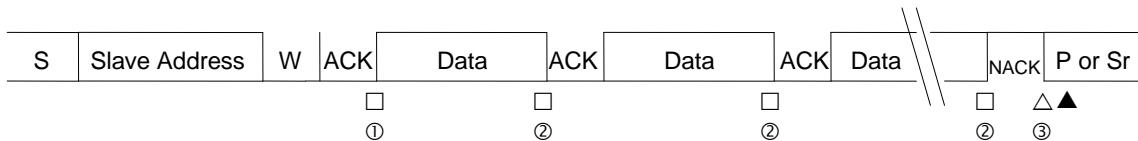
- The send data is written in the TDR register.

③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-24 Master mode transmit interrupt 13 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response during transmission)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the TDR register.

② An interrupt occurs when a single byte is sent.

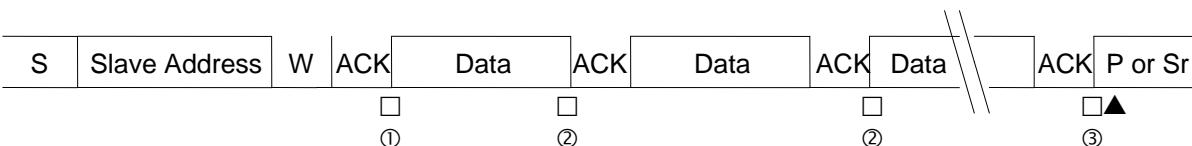
- The send data is written in the TDR register.

③ An interrupt occurs when a NACK is responded.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-25 Master mode transmit interrupt 14 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="1" -> "0", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- The send data is written in the send buffer.

② An interrupt occurs when a single byte is sent.

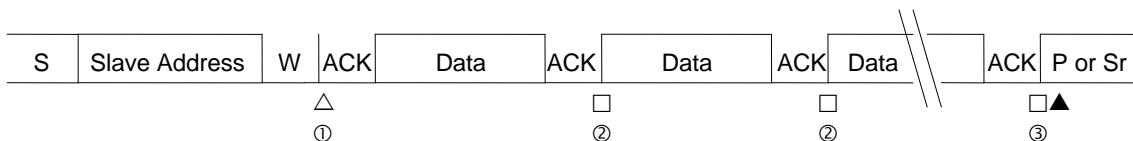
- The WSEL bit is set to "0" and the send data is written in the send buffer.

③ An interrupt occurs when a single byte is sent.

- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (TBIE) is set, the TDRE bit is set to "1".

Figure 2-26 Master mode interrupt 15 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

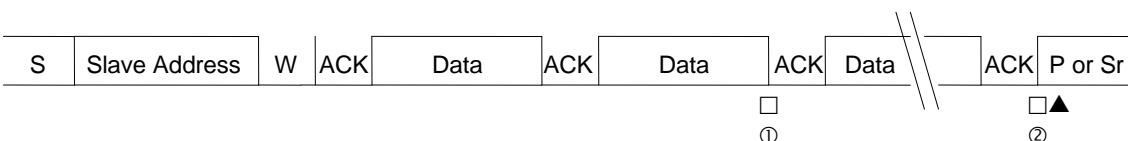
① An interrupt occurs when the slave address (reserved address) is sent, a direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".

② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register.

③ An interrupt occurs when a single byte is sent and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-27 Master mode transmit interrupt 16 by enabling FIFO
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK response)



S: Start condition

W: Data direction bit (writing)

P: Stop condition

Sr: Iteration start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer.

② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied) and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-28 Master mode transmit interrupt 17 by enabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")

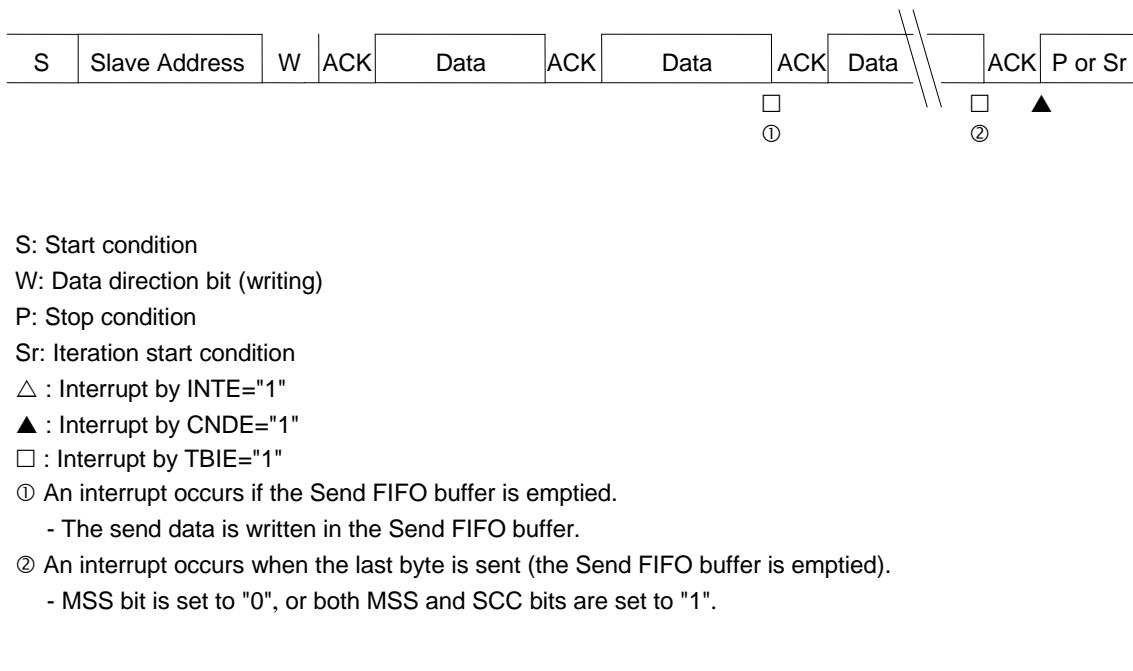
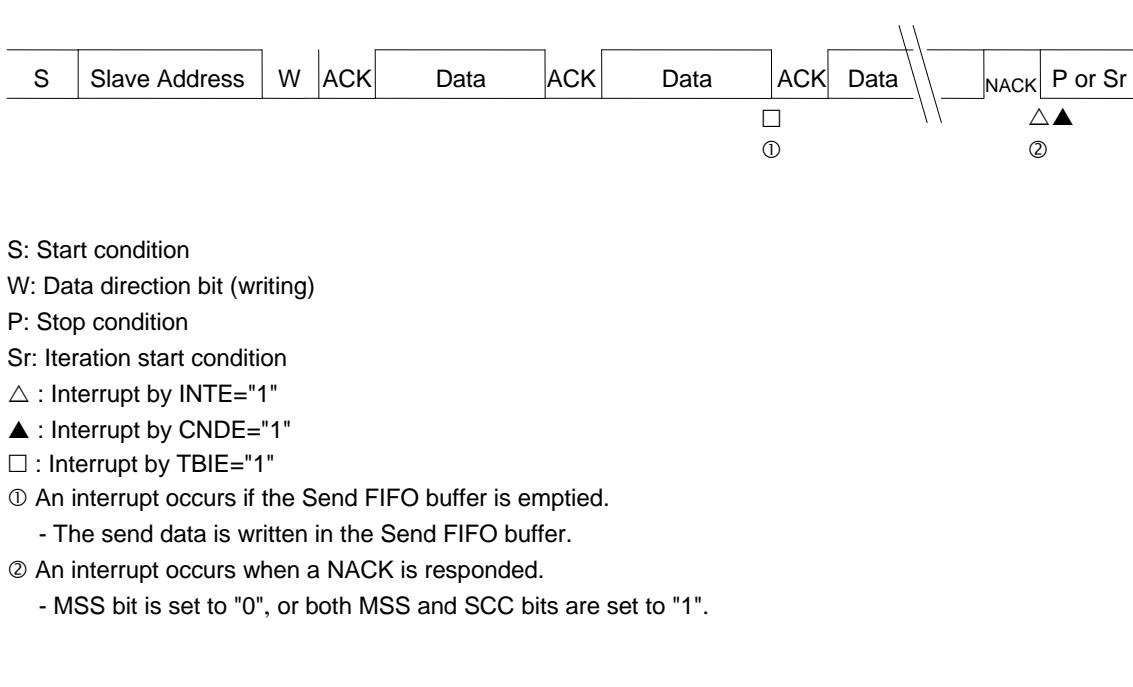


Figure 2-29 Master mode transmit interrupt 18 by enabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



■ Data reception by the master

● When DMA mode is disabled (SSR:DMA=0)

When the data direction bit (R/W) is set to "1", the master receives data transmitted from a slave.

When FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is set to "1", wait is generated (IBCR:INT="1", SSR:RDRF="1") each time one byte is received . At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".
- If the SSR:TDRE bit is set to "0", wait is not generated (IBCR:INT="0") when an ACK response is set for the ACKE bit in the IBCR register while wait is generated when the NACK response is set (IBCR:INT="1").

When FIFO is enabled, the SSR:RDRF bit is set to "1" upon reception of data in the same number of bytes set for the number of bytes to be received. The interrupt flag is set to "1" when the SSR:TDRE bit is "1", which puts the I²C bus in the wait state. At this time, acknowledgement operates as follows. Even if NACK is output, it is stored in receive FIFO as receive data.

- In case of IBCR:WSEL="0", an NACK response is returned when the SSR:TDRE bit is set to "1" if NACK is set for the ACKE bit.
- In case of IBCR:WSEL="1", the interrupt flag is set to "1" after receiving the final byte, which generates wait. During that wait, an ACK or NACK response is returned according to the IBCR:ACKE setting after the IBCR:ACKE bit is set and the interrupt flag is cleared to "0".

For interrupt-generated wait, refer to the following.

Table 2-6 IBCR:WSEL bit status for master data reception when DMA mode is disabled (SSR:DMA=0)

WSEL bit	Operation
0	After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.
1	After the second byte, after the master has received one-byte data with "1" set for the SSR:TDRE bit, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.

The following shows an example procedure for receiving data from a slave.

- When receive FIFO is disabled:
 1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
 2. ACK is received after the Slave Address setting is transmitted, and then the interrupt flag (IBCR:INT) is set to "1".
 3. Writes "0" to the interrupt flag bit (IBCR:INT) upon updating of the IBCR:WSEL bit to release the wait state of the I²C bus.
 4. After receiving one byte, sets the interrupt flag to "1" to set the I²C bus in the wait state after transmitting acknowledgment in case of IBCR:WSEL="0" and directly after receiving one byte in case of IBCR:WSEL=1. Repeats steps 3 to 4 until all the specified number of data sets have been received.
 5. After receiving the last data, outputs NACK and sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

- When transmit/receive FIFO is enabled:

 1. Sets the number of bytes to be received to the FBYTE register.
 2. Writes an address for Slave Address (including the data direction bit) and dummy data in the number of bytes to be received to the TDR register.
 3. Writes "1" to the IBCR:MSS bit.
 4. An ACK response is returned and data reception continues as long as the SSR:TDRE bit stays "0". During that reception operation, SSR:RDRF is set to "1" when the number of bytes set up in FBYTE have been received. When SSR:RDRF is set to "1", starts reading from the RDR register.
 5. When SSR:TDRE bit is "1", sets the interrupt flag to "1" to set the I²C bus in the wait state after outputting NACK if IBCR:WSEL="0", and directly after one-byte reception if IBCR:WSEL="1".
 6. In case of IBCR:WSEL="1", sets the IBCR:ACKE bit to "0". In case of IBCR:WSEL="0", no setting is needed for the IBCR:ACKE bit. Setting the IBCR:MSS bit to "0" or setting the IBCR:SCC bit to "1" generates the stop condition or iteration start condition.

● When DMA mode is enabled (SSR:DMA=1)

When the data direction bit (R/W) is set to "1", the master receives data transmitted from a slave.

When FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is set to "1", wait is generated (SSR:TBI="1", SSR:RDRF="1") each time one byte is received. At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".
- If the SSR:TDRE bit is set to "0", wait is generated (SSR:RDRF="1") each time one byte is received. At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".

When FIFO is enabled, the SSR:RDRF bit is set upon reception of data in the same number of bytes set for the number of bytes to be received. The transmit bus idle flag (SSR:TBI) is set when the SSR:TDRE bit is "1", which puts the I²C bus in the wait state. At this time, acknowledgement operates as follows. Even if NACK is output, it is stored in receive FIFO as receive data.

- In case of IBCR:WSEL="0", an NACK response is returned when the SSR:TDRE bit is set to "1" if NACK is set for the ACKE bit.
- In case of IBCR:WSEL="1", wait is generated (SSR:TBI="1") after receiving the last byte. During that wait, the master sets the IBCR:ACKE bit and returns ACK or NACK response, according to the IBCR:ACKE setting, after clearing the transmit bus idle flag (SSR:TBI).

For interrupt-generated wait, refer to the following.

Table 2-7 IBCR:WSEL bit status for master data reception when DMA mode is enabled (SSR:DMA=1)

WSEL bit	Operation
0	After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. After the second byte, after acknowledgement when receive FIFO is not used, if the receive data full flag (SSR:RDRF) is set to "1", SCL is set to LOW for the wait state.
1	After the second byte, after the master has received one-byte data with "1" set for the SSR:TDRE bit, the interrupt flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. After the second byte, after the receive data full flag (SSR:RDRF) is set to "1" when receive FIFO is not used, SCL is set to LOW for the wait state.

The following shows an example procedure for receiving data from a slave.

- When receive FIFO is disabled:
 1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
 2. ACK is received after the Slave Address setting is transmitted, and then the transmit bus idle flag (SSR:TBI) is set to "1".
 3. Writes data to be transmitted to the TDR register to release the wait state of the I²C bus.
 4. After one byte is received, sets the transmit bus idle flag (SSR:TBI) and the receive data full flag (SSR:RDRF)*2 to "1" under the following conditions to put the I²C bus in the wait state.
 - In case of IBCR:WSEL="0", after transmitting acknowledgement
 - In case of IBCR:WSEL=1, after receiving one byte
 5. Updates the IBCR:WSEL bit, reads from the RDR register and writes dummy data to the TDR register.
 6. After one byte is received, sets the transmit bus idle flag (SSR:TBI) and the receive data full flag (SSR:RDRF)*2 to "1" under the following conditions to put the I²C bus in the wait state.
 - In case of IBCR:WSEL="0", after transmitting acknowledgement
 - In case of IBCR:WSEL=1, after receiving one byte

Repeats steps 5 to 6 until all the specified number of data sets have been received.
- 7. After receiving the last data, outputs NACK and sets the IBCR:MSS bit to "0" or sets the IBCR:SCC*1 bit to "1" to generate the stop condition or iteration start condition.

- When transmit/receive FIFO is enabled:
 1. Sets the number of bytes to be received to the FBYTE register.
 2. Writes an address for Slave Address (including the data direction bit) and dummy data in the number of bytes to be received to the TDR register.
 3. In case of IBCR:WSEL="0", sets NACK for the ACKE bit, and writes "1" to the IBCR:MSS bit.
 4. An ACK response is returned and data reception continues as long as the SSR:TDRE bit stays "0".
During that reception operation, SSR:RDRF is set to "1" when the number of bytes set up in FBYTE have been received. When SSR:RDRF is set to "1", starts reading from the RDR register.
 5. When the SSR:TDRE bit is set to "1", sets the interrupt flag to "1" to set the I²C bus in the wait state after outputting NACK if IBCR:WSEL="0". In case of IBCR:WSEL="1", directly after one byte is received, sets the transmit bus idle flag (SSR:TBI) to "1" to put the I²C bus in the wait state.
 6. In case of IBCR:WSEL="1", sets the IBCR:ACKE bit to "0". In case of IBCR:WSEL="0", no setting is needed for the IBCR:ACKE bit, Set the IBCR:MSS bit to "0" or set the IBCR:SCC*1 bit to "1" to generate the stop condition or iteration start condition.

*1 : When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.

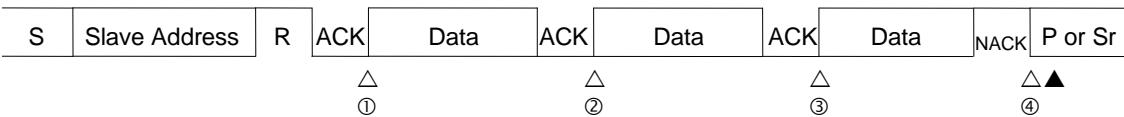
1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

*2 : Directly after receiving one byte, the receive data full flag (SSR:RDRF) is set to "1" regardless of the setting for IBCR:WSEL. When the receive data full flag (SSR:RDRF) is set to "1", put the I²C bus in the wait state after transmitting acknowledgement in case of IBCR:WSEL="0", and directly after receiving one byte in case of IBCR:WSEL=1.

<Notes>

- When seven-bit slave address detection is enabled (ISBA:SAEN="1"), it is prohibited to specify a seven-bit slave address in master mode.
- When SSR:TDRE is "0", even if an overrun error occurs, acknowledgement is output according to the setting for the IBCR:ACKE bit, and then the next process should follow.
- To change the IBCR register during transmission/reception, do so when the interrupt flag (IBCR:INT) is "1" or when the transmit bus idle flag (SSR:TBI) is "1" during DMA mode being enabled (SSR:DMA=1).
- In the master mode reception with DMA disabled (SSR:DMA=0), write dummy data to the TDR register, and then, if the SSR:TDRE bit is "0" when the interrupt flag (IBCR:INT) is turned to "1", receive the next data with the interrupt flag (IBCR:INT) kept at "0".
- In the master mode reception with DMA enabled (SSR:DMA=1), write dummy data to the TDR register, and then, if the SSR:TDRE bit is "0" when the transmit bus idle flag (SSR:TBI) is turned to "1", receive the next data with the transmit bus idle flag (SSR:TBI) kept at "0".
- To receive data when receive FIFO is enabled and IBCR:WSEL="0", the SSR:RDRF bit is set to "1" after receiving the last bit and the interrupt flag (IBCR:INT) is set to "1" after transmitting ACK.

Figure 2-30 Master mode receive interrupt 1 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



\triangle : Interrupt by INTE="1"

\blacktriangle : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- If the INT bit is set to "0", the interrupt flag is cleared to "0".

② An interrupt occurs when a single byte is received and an ACK is sent.

- After the received data has been read, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received and an ACK is sent.

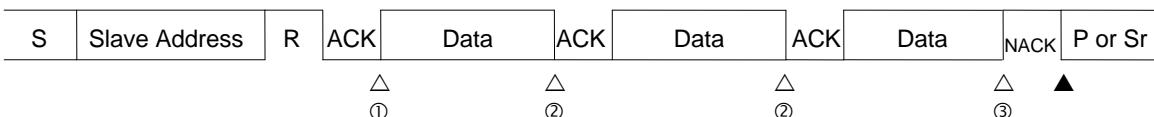
- After the received data has been read, both ACKE and INT bits are set to "0".

④ An interrupt occurs when a single byte is received and an ACK is sent.

- MSS bit is set to "0", or both MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-31 Master mode receive interrupt 2 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



\triangle : Interrupt by INTE="1"

\blacktriangle : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.

- If the INT bit is set to "0", the interrupt flag is cleared to "0".

② An interrupt occurs when a single byte is received.

- After the received data has been read, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received.

- After the received data has been read, ACKE bit are set to "0". MSS bit is set to "0", or both MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-32 Master mode receive interrupt 3 by enabling FIFO

(SSR:DMA="0", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")

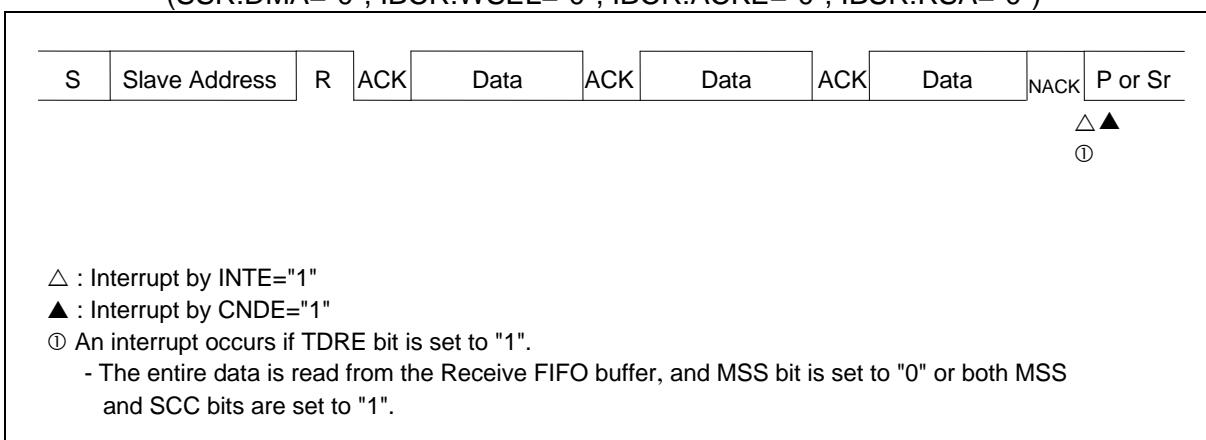


Figure 2-33 Master mode receive interrupt 4 by enabling FIFO

(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")

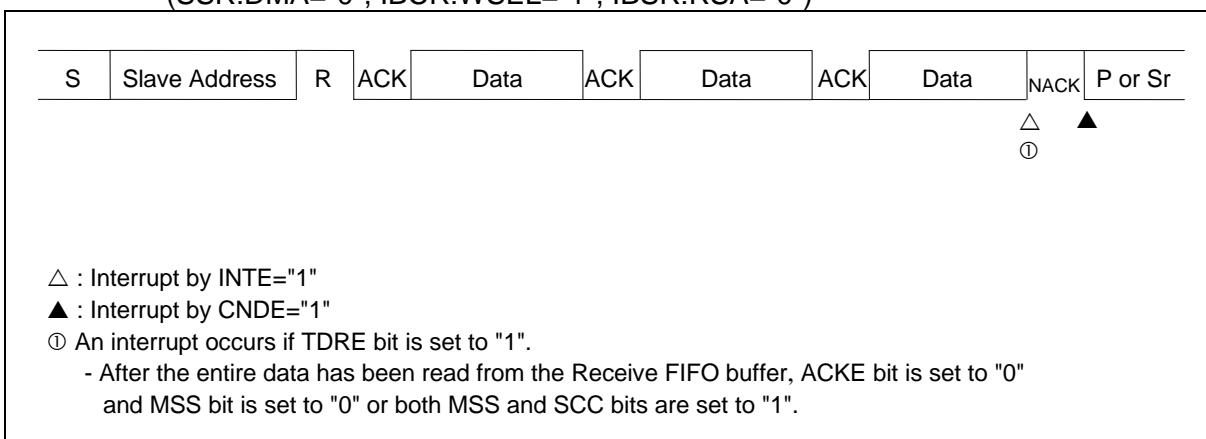


Figure 2-34 Master mode receive interrupt 5 by disabling FIFO

(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")

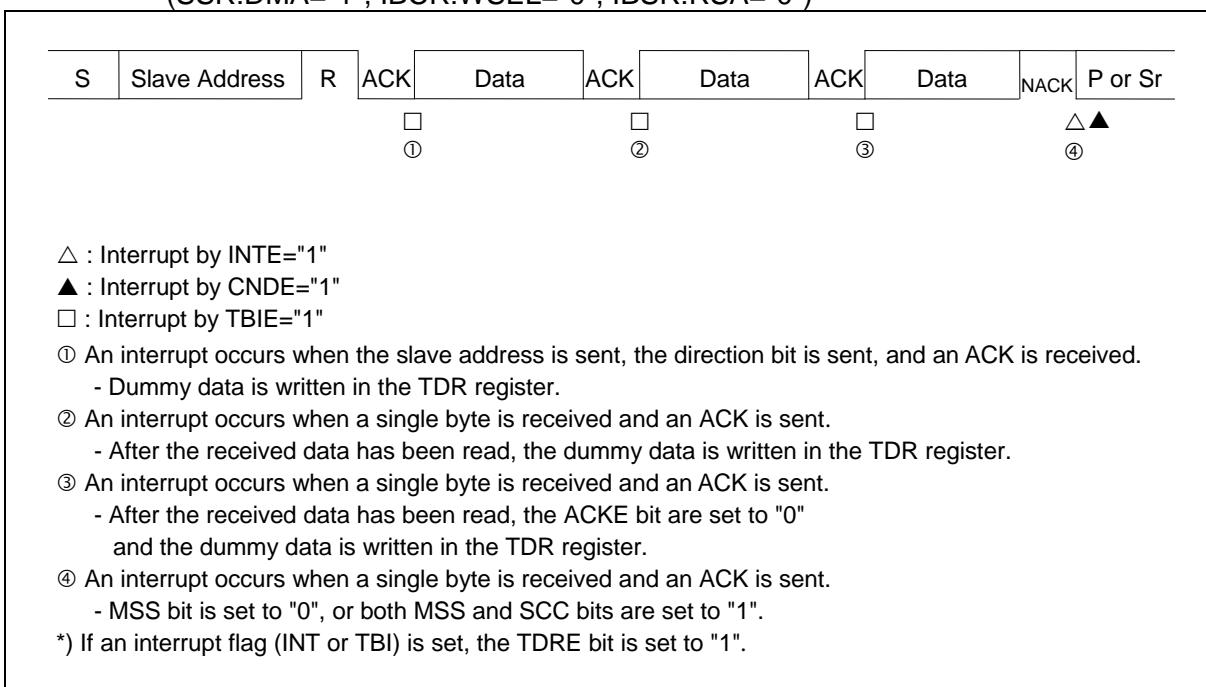
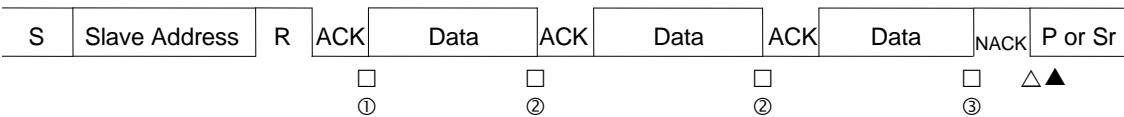


Figure 2-35 Master mode receive interrupt 6 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- Dummy data is written in the TDR register.

② An interrupt occurs when a single byte is received.

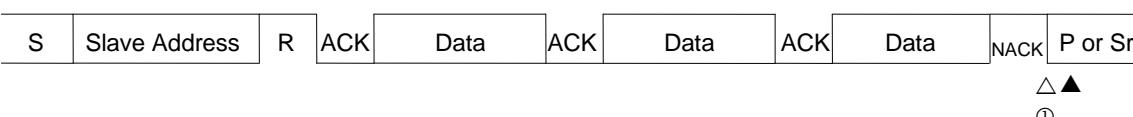
- After the received data has been read, the dummy data is written in the TDR register.

③ An interrupt occurs when a single byte is received.

- After the received data has been read, ACKE bit are set to "0" MSS bit is set to "0",
or both MSS and SCC bits are set to "1".

*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-36 Master mode receive interrupt 7 by enabling FIFO
(SSR:DMA="1", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



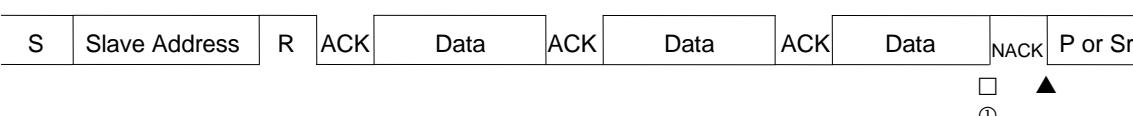
△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs if TDRE bit is set to "1".

- The entire data is read from the Receive FIFO buffer, and MSS bit is set to "0" or both MSS
and SCC bits are set to "1".

Figure 2-37 Master mode receive interrupt 8 by enabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



□ : Interrupt by TBIE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs if TDRE bit is set to "1".

- After the entire data has been read from the Receive FIFO buffer, ACKE bit is set to "0"
and MSS bit is set to "0" or both MSS and SCC bits are set to "1".

■ Arbitration lost

If the master receives the data different from sent data, due to concurrent transmission of data from another master, the master judges the situation as arbitration lost. At this time, the IBCR:MSS bit is set to "0" and the IBSR:AL bit to "1", enabling operation in slave mode.

The IBSR:AL bit can be cleared to "0" under the following conditions:

- The IBCR:MSS bit is set to "1".
- The IBCR:INT bit is set to "0".
- The IBSR:SPC bit is set to "0" when the IBSR:AL bit and IBSR:SPC bit are "1".
- The I²C interface operation is disabled (ISMK:EN bit="0").

Upon an occurrence of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" according to the setting of the IBCR:WSEL bit, and sets SCL of the I²C bus to LOW.

■ Wait state for master mode

When both conditions below are satisfied, master mode is put in the wait state while the IBSR:BB bit stays "1". After the IBSR:BB bit attains "0", start condition is transmitted.

- When the IBCR:MSS is set to "1" while the IBSR:BB bit is "1"
- When the interface is not operating as a slave

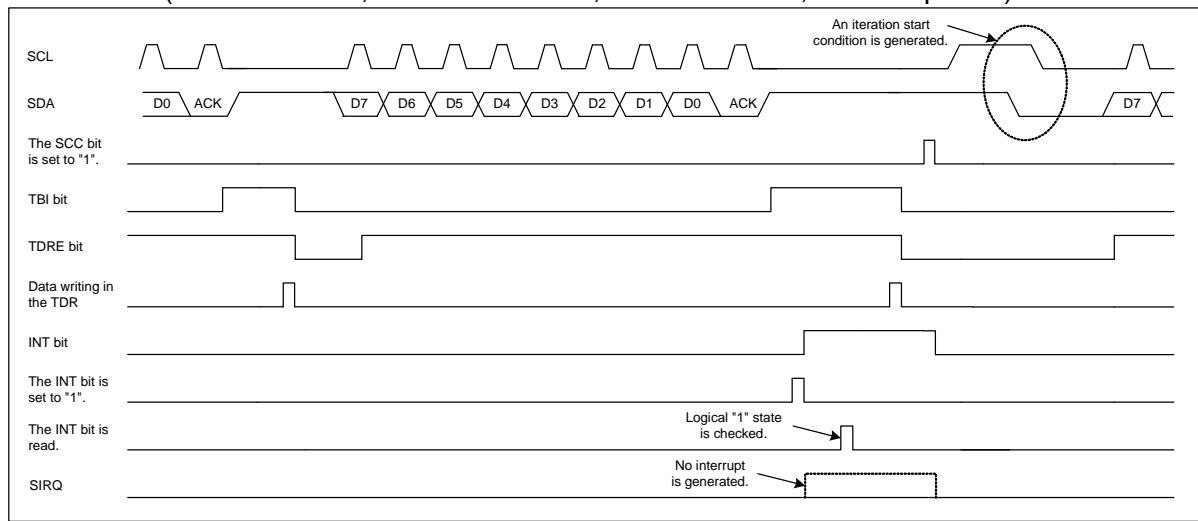
Refer to the IBCR:MSS bit and IBCR:ACT bit to check if master mode is in the wait state or not (in the wait state if the IBCR:MSS="1" and IBCR:ACT="0"). After setting the IBCR:MSS bit to "1" and to operate in slave mode, set the IBSR:AL bit to "1", the IBCR:MSS bit to "0", and the IBCR:ACT bit to "1".

■ Issuing iteration start condition when DMA mode is enabled (SSM:DMA=1)

When writing a slave address to the TDR register while the transmit bus is idle (SSR:TBI="1") and the interrupt flag (IBCR:INT) is "0", transmission starts and the iteration start condition cannot be issued. Therefore, to issue the iteration start condition while the transmit bus is idle (SSR:TBI="1") and the interrupt flag (IBCR:INT) is "0", follow the steps below.

1. Set the IBCR:INT bit to "1". At this time, no SIRQ interrupt is generated.
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Issue the iteration start condition (IBCR:SCC="1").

**Figure 2-38 Issuing iteration condition when DMA mode is enabled
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK response)**



2.3. Slave mode

If the (iteration) start condition is detected and a combination of the ISBA and ISMK registers matches the received address, the interface outputs an ACK response and acts in slave mode.

■ Slave address match detection

After the (iteration) start condition is detected, subsequent seven bits are received as the address. For each of the bits that are set to "1" in the ISMK register, the ISBA register is compared with the received address. If they match, ACK is output.

Table 2-8 Operation immediately after outputting acknowledgement to a slave address

Transmit FIFO	Receive FIFO	Transmit FIFO status	Receive FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
				1		
Disable	Enable	-	Without data	0	Holds the IBCR:INT bit to "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	
Enable	Disable	-	-	0	Sets the IBCR:INT bit to "1" with the wait state if the SSR:TDRE bit is "1", and holds the IBCR:INT bit to "0" without the wait state if the SSR:TDRE bit is "0".	Holds the IBCR:INT bit to "0" without the wait state.
				1		
Enable	Enable	-	Without data	0	Holds the IBCR:INT bit to "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
				1	Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	

- Detection of reserved address

If the first byte matches the reserved address ("0000xxxx" or "1111xxxx"), the value of bit 8 is received regardless of whether or not transmit/receive FIFO is enabled, and the IBCR:INT bit is set to "1", causing the I²C bus to be placed into the wait state. After the receive data has been read, configure the following settings.

- To run the interface as a slave device, set the IBCR:ACKE bit to "1" and check the value of the data direction bit (IBSR:TRX). If the transmitting direction is set, write the transmit data to TDR, and clear the IBCR:INT bit. The interface then acts as a slave device.
- When not running the interface as a slave device, set the IBCR:ACKE bit to "0", and clear the IBCR:INT bit. After acknowledgement has been output, the interface does not act as a slave device.

■ Data direction bit

After receiving the address, the interface receives the data direction bit to determine whether to transmit or receive data. If this bit is "0", it means that data is transmitted from the master device, and the interface receives data as a slave device.

■ Reception in slave mode

If the received data matches the slave address and the data direction bit is "0", it means that data is received in slave mode. The following shows a procedure example to receive data in slave mode.

● When DMA mode is disabled (SSR:DMA=0)

- When receive FIFO is disabled:
 1. After transmitting ACK, set the interrupt flag (IBCR:INT) to "1", and place the I²C bus into the wait state. Based on the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits, judge that the event is an interrupt by a slave address match. Then write "1" to the IBCR:ACKE bit and "0" to the interrupt flag (IBCR:INT), and release the wait state of the I²C bus (see Table 2-8).
 2. After receiving 1-byte data, set the interrupt flag (IBCR:INT) to "1" according to setting of the IBCR:WSEL bit, and place the I²C bus into the wait state.
 3. Read the data received from the RDR register, set the IBCR:ACKE bit, write "0" to the interrupt flag (IBCR:INT), and release the wait state of the I²C bus.
 4. Repeat steps 2 and 3 to detect the stop or iteration start condition.

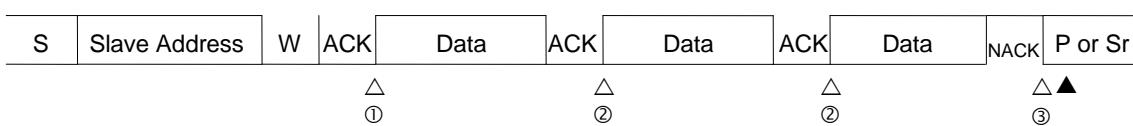
- When receive FIFO is enabled:
 1. If NACK is detected or receive FIFO becomes full, the interrupt flag (IBCR:INT) is set to "1", and the I²C bus is placed into the wait state. If the stop or iteration start condition is detected, the interrupt flag (IBCR:INT) is not set to "1" (the I²C bus is not placed into the wait state) by setting the IBSR:SPC and IBSR:RSC bits to "1". Receive FIFO sets the SSR:RDRF bit to "1" when the set value of the FBYTE register matches the number of data sets received. If the SMR:RIE bit is then "1", a receive interrupt is generated.
 2. When the interrupt flag (IBCR:INT) is set to "1", read the received data from the RDR register. After all data has been read, write "0" to the interrupt flag to release the wait state of the I²C bus. If the stop or iteration start condition is detected, read all the received data from the RDR register, and clear the IBSR:SPC or IBSR:RSC bit to "0".

● When DMA mode is enabled (SSR:DMA=1)

- When receive FIFO is disabled:
 1. After transmitting ACK, set the interrupt flag (IBCR:INT) to "1", and place the I²C bus into the wait state. Based on the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits, judge that the event is an interrupt by a slave address match. Then write "1" to the IBCR:ACKE bit and "0" to the interrupt flag (IBCR:INT), and release the wait state of the I²C bus (see Table 2-8).
 2. Set "1" to the receive data full flag (SSR:RDRF) immediately after receiving 1-byte data. When the receive data full flag (SSR:RDRF) is set to "1", if IBCR:WSEL="0", place the I²C bus into the wait state after transmitting acknowledgement. If IBCR:WSEL=1, place the I²C bus into the wait state immediately after receiving the 1-byte data.
 3. After setting the IBCR:ACKE bit, read the data received from the RDR register, and clear the receive data full flag (SSR:RDRF) to "0" to release the wait state of the I²C bus.
 4. Repeat steps 2 and 3 to detect the stop or iteration start condition.

- When receive FIFO is enabled:
- If NACK is detected, the interrupt flag (IBCR:INT) is set to "1", and the I²C bus is placed into the wait state. When receive FIFO becomes full, place the I²C bus into the wait state. If the stop or iteration start condition is detected, the IBSR:SPC and IBSR:RSC bits are set to "1", and the interrupt flag (IBCR:INT) is not set to "1" (the I²C bus is not placed into the wait state). Receive FIFO sets the SSR:RDRF bit to "1" when the set value of the FBYTE register matches the number of data sets received. If the SMR:RIE bit is then "1", a receive interrupt is generated.
 - When the interrupt flag (IBCR:INT) is set to "1", read the received data from the RDR register. After all data has been read, write "0" to the interrupt flag to release the wait state of the I²C bus. When receive FIFO is full, release the wait state of the I²C bus if the received data is read from the RDR register even once. If the stop or iteration start condition is detected, read all the received data from the RDR register, and clear the IBSR:SPC or IBSR:RSC bit to "0".

Figure 2-39 Slave mode receive interrupt 1 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")

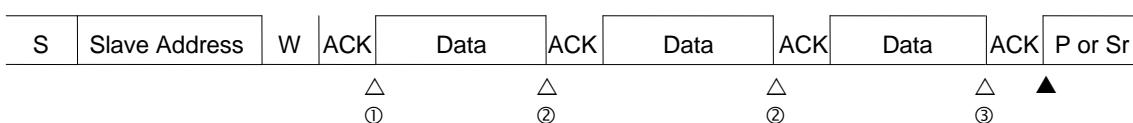


△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

- As the slave address matches , an ACK is output and an interrupt is generated .
 - ACKE bit is set to "1" and INT bit is set to "0".
- An interrupt occurs when a single byte is received and an ACK is responded .
 - After the received data has been read from the receive buffer , the INT bit is set to "0".
- An interrupt occurs when a single byte is received and a NACK is responded .
 - After the received data has been read from the receive buffer , the INT bit is set to "0".

Figure 2-40 Slave mode receive interrupt 2 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

- As the slave address matches , an ACK is output and an interrupt is generated .
 - ACKE bit is set to "1" and INT bit is set to "0".
- An interrupt occurs when a single byte is received .
 - After the received data has been read from the receive buffer , the INT bit is set to "0".
- An interrupt occurs when a single byte is received .
 - After the received data has been read from the receive buffer , the INT bit is set to "0".

Figure 2-41 Slave mode receive interrupt 3 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")

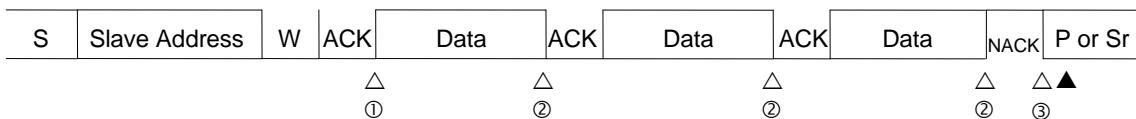


Figure 2-42 Slave mode receive interrupt 4 by enabling receive FIFO
(SSR:DMA="0", IBSR:RSA="0")

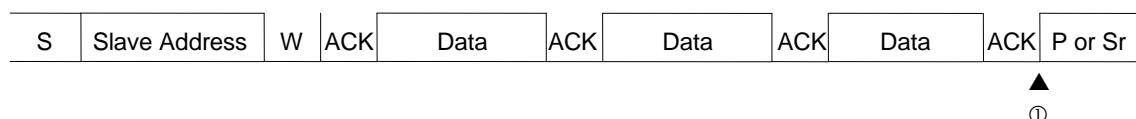


Figure 2-43 Slave mode receive interrupt 5 by enabling receive FIFO
(SSR:DMA="0", IBSR:RSA="0")

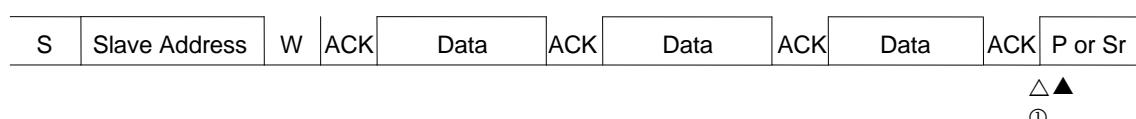
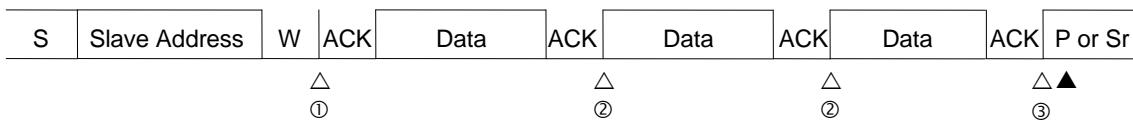


Figure 2-44 Slave mode receive interrupt 6 by disabling FIFO
(SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① An interrupt occurs as the reserved address ("0000xxxx" or "1111xxxx") matches.

- The received data is read , and ACKE bit is set to "1" and INT bit is set to "0".

② An interrupt occurs when a single byte is received and an ACK is output .

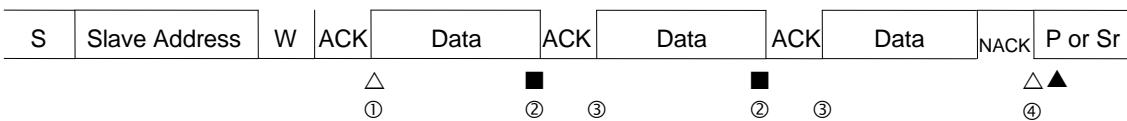
- INT bit is set to "0".

③ An interrupt occurs when a single byte is received and an ACK is output .

- An interrupt occurs if INT bit is set to "0".

Figure 2-45 Slave mode receive interrupt 7 by disabling FIFO

(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

■ :Interrupt by RIE="1"

① As the slave address matches , an ACK is output and an interrupt is generated.

- ACKE bit is set to "1" and INT bit is set to "0".

② An interrupt occurs(but the I₂C bus is not waited) when a single byte is received.

- The received data is read from the receive buffer .

③ The I₂C bus is waited when an ACK is responded.

- The received data is read from the receive buffer .

④ An interrupt occurs when a single byte is received and a NACK is responded.

- After the received data has been read from the receive buffer , the INT bit is set to "0".

Figure 2-46 Slave mode receive interrupt 8 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")

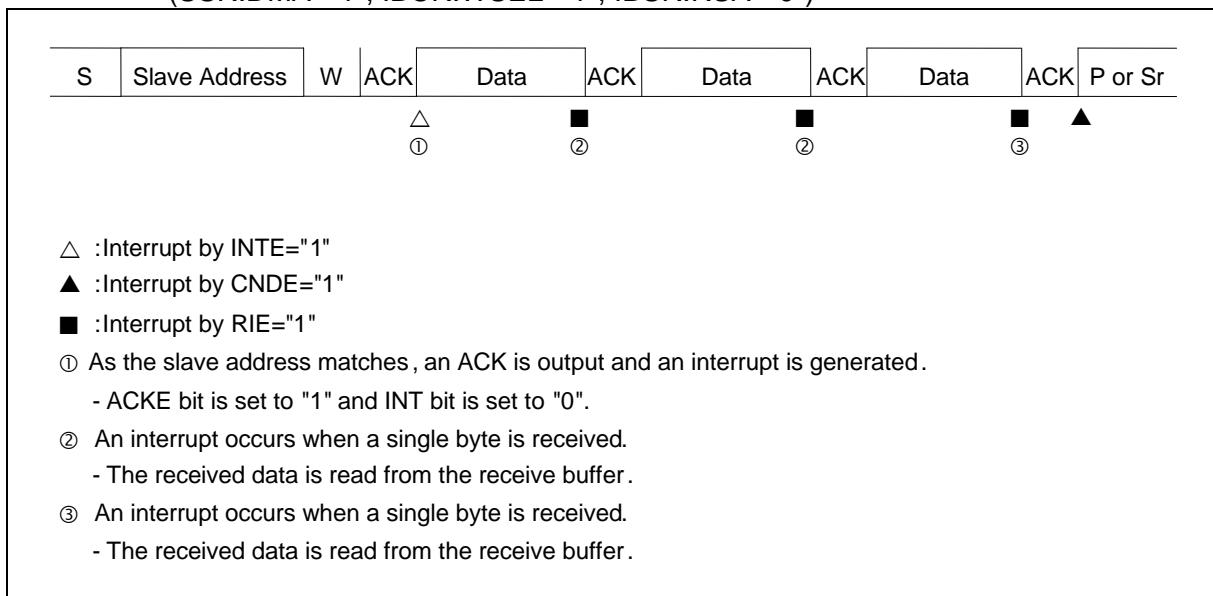


Figure 2-47 Slave mode receive interrupt 9 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")

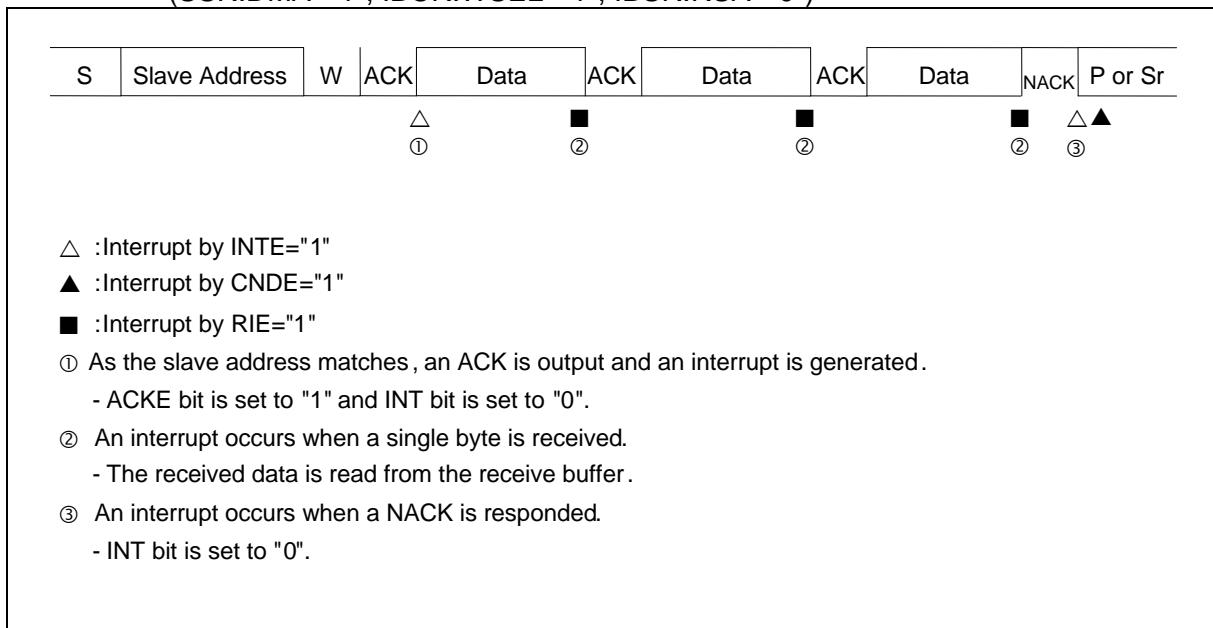


Figure 2-48 Slave mode receive interrupt 10 by enabling receive FIFO
(SSR:DMA="1", IBSR:RSA="0")

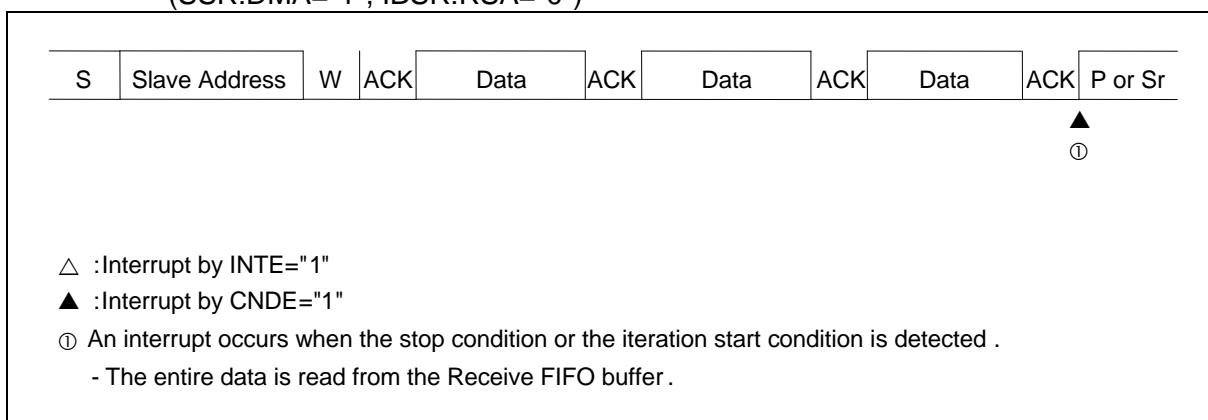


Figure 2-49 Slave mode receive interrupt 11 by enabling receive FIFO
(SSR:DMA="1", IBSR:RSA="0")

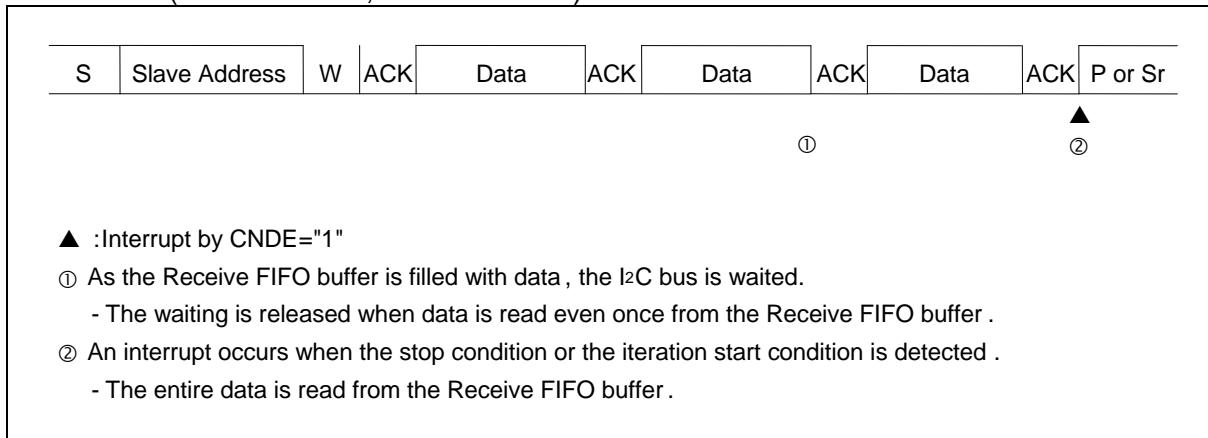
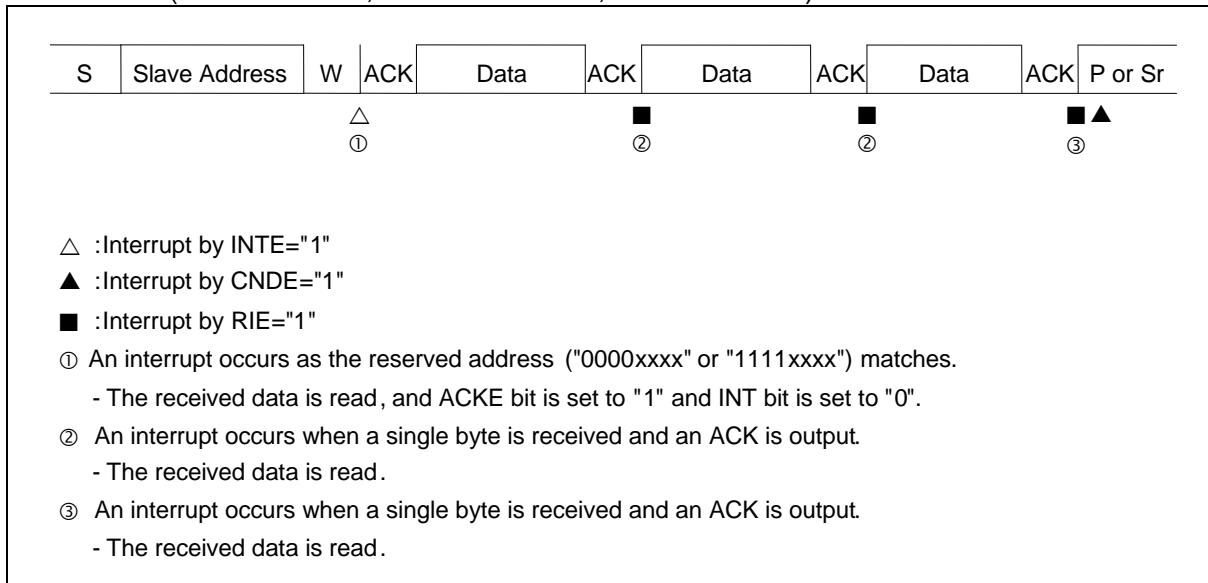


Figure 2-50 Slave mode receive interrupt 12 by disabling FIFO
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



■ Transmission in slave mode

If the received data matches the slave address and the data direction bit is "1", it means that data is transmitted in slave mode. If FIFO is disabled, set the interrupt flag (IBCR:INT) to "1" after transmitting one byte or outputting an acknowledgement response depending on setting of the IBCR:WSEL bit. Then place the I²C bus into the wait state (see Table 2-8).

Using the IBSR:RACK bit, check the acknowledgement output from the master device. If NACK response is returned from the master device, it means that the master device could not receive data correctly or data receiving was ended. If NACK is detected at IBCR:WSEL="1", an interrupt is generated to place the I²C bus into the wait state.

2.4. Bus error

If the stop or (iteration) start condition is detected while transmitting or receiving data on the I²C bus, it is handled as a bus error.

■ Bus error occurrence condition

If a bus error occurs, the IBCR:BER bit is set to "1" in the following conditions.

- The (iteration) start or stop condition is detected while transferring the first byte.
- The (iteration) start condition or stop condition is detected at bit 2 to 9 (acknowledgement) of data.

■ Bus error operation

If the interrupt flag (IBCR:INT) is set to "1" by transmitting or receiving data, check the IBCR:BER bit.

When the IBCR:BER bit is "1", perform error processing. The IBCR:BER bit is cleared by writing "0" to the IBCR:INT bit.

If a bus error occurs, the IBCR:INT bit is set to "1"; however, the I²C bus is not placed into the wait state by setting its SCL to LOW.

3. Dedicated Baud Rate Generator

The dedicated baud rate generator configures the setting of the serial clock frequency.

■ Selecting the baud rate

● Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an internal clock by the set value.

■ Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

The baud rate is obtained in the following formulas.

(1) Reload value

$$V = \phi / b - 1$$

V: Reload value b: Baud rate ϕ : Bus clock frequency or external clock frequency

Note that the preset baud rate may not be generated at a rising edge of signal on I²C bus. In such case, adjust the reload value.

(2) Calculation example

To set the 16MHz bus block and 400K-bps baud rate, set the reload value as follows.

Reload value:

$$V = (16 \times 1000000)/400000 - 1 = 39$$

Therefore, the baud rate is:

$$b = (16 \times 1000000)/(39 + 2) = 400\text{kbps}$$

<Notes>

- Write Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) by 16-bit access operation.
- When the ISMK:EN bit in the ISMK register is "0", set the value of each Baud Rate Generator Register.
- In operation mode 4 (I²C mode), operate the bus clock at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.
- If the reload value is set to "0", the reload counter is stopped.

■ Reload values and baud rates for each bus clock frequency

Table 3-1 Reload values and baud rates

Baud rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32MHz
	Value	Value	Value	Value	Value	Value
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

The numeric values above are available when the SCL rising timing of the I²C bus is 0s. If the SCL rising timing of the I²C bus is late, the baud rate is set to the value later than the numeric values above.

■ Functions of reload counter

Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from internal clocks. The count value of the transmit reload counter can be read from the Baud Rate Generator Registers (BGR1 and BGR0).

■ Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

4. I²C communication operation flowchart examples

This section shows I²C communication operation flowchart examples.

■ I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0)

Figure 4-1 I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0) 1/3

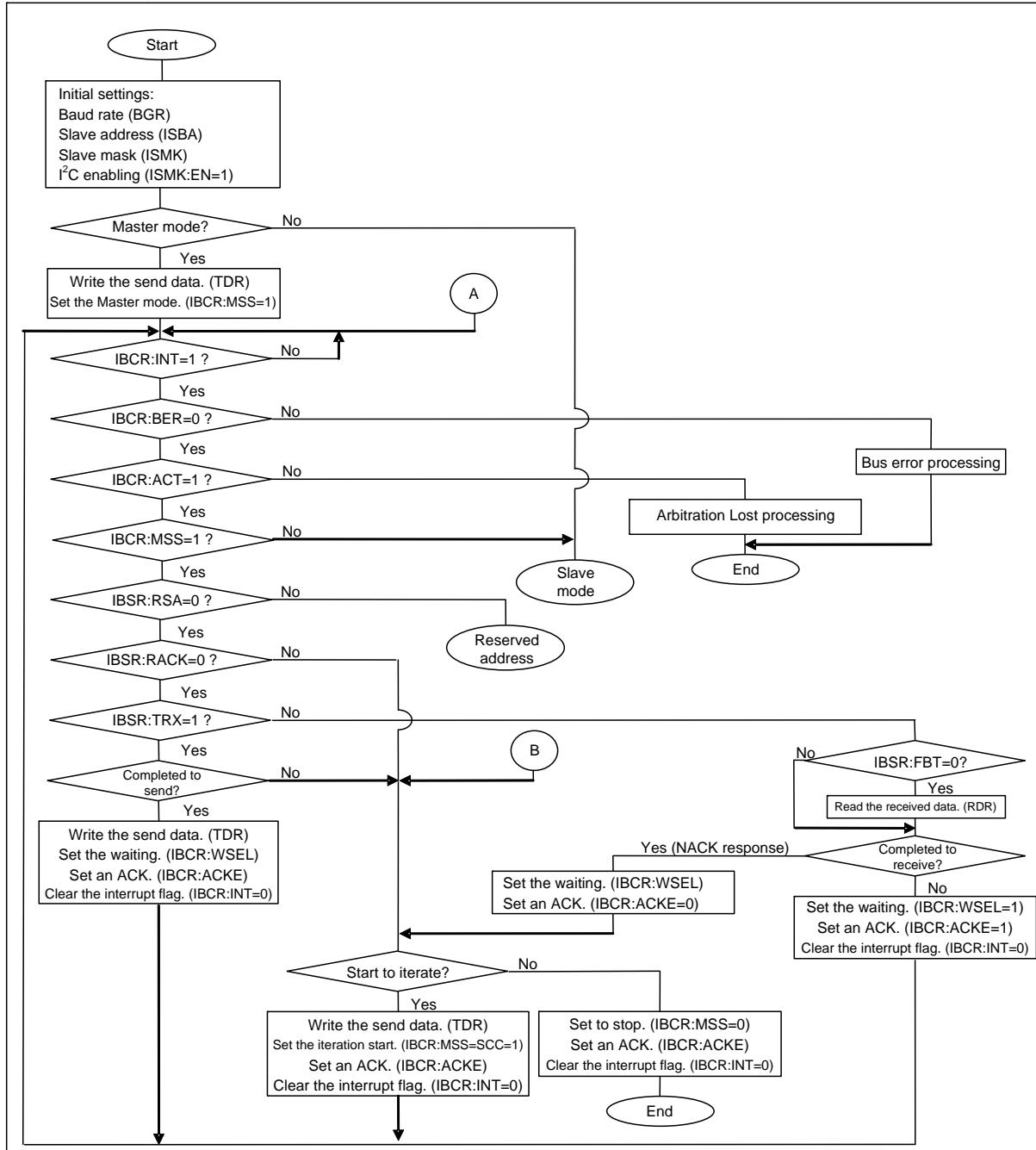


Figure 4-2 I²C flowchart example (FIFO not used) when DMA mode is disabled
(SSR:DMA=0) 2/3

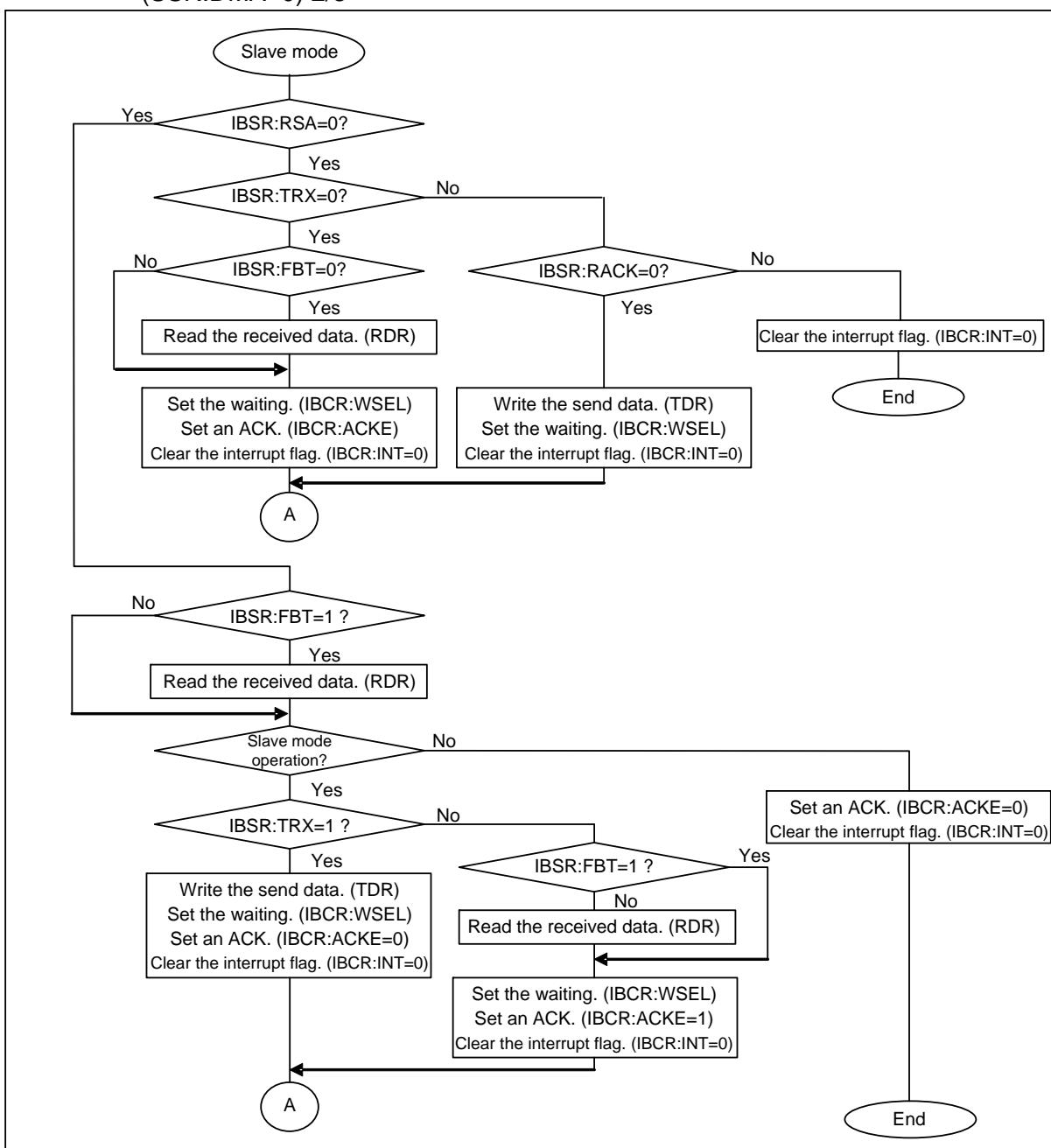
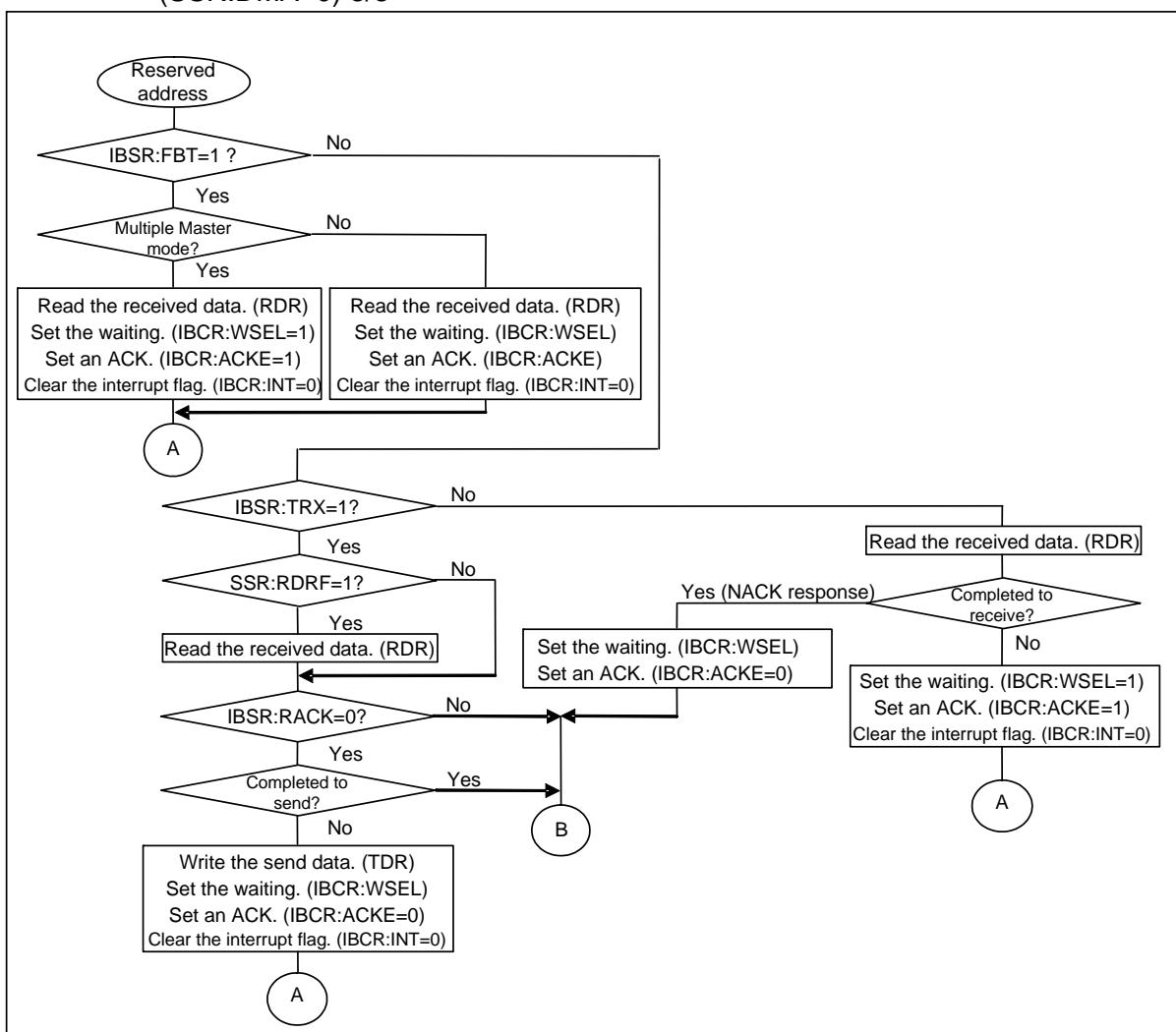


Figure 4-3 I²C flowchart example (FIFO not used) when DMA mode is disabled
(SSR:DMA=0) 3/3



**■ I²C flowchart examples (FIFO not used) when DMA mode is enabled
(SSR:DMA=1)**

Figure 4-4 I²C flowchart example (FIFO not used) when DMA mode is enabled
(SSR:DMA=1) 1/4

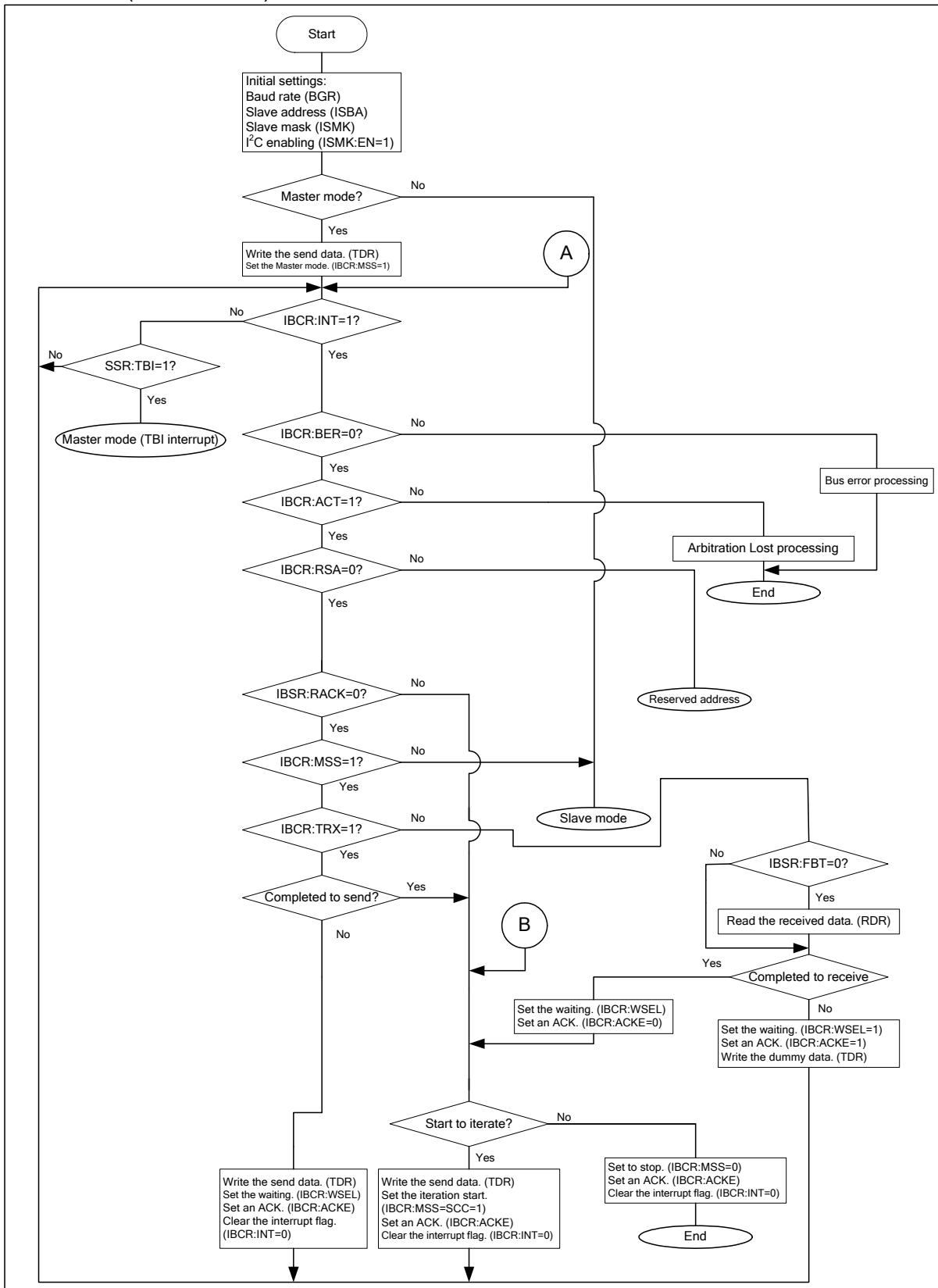


Figure 4-5 I²C flowchart example (FIFO not used) when DMA mode is enabled
(SSR:DMA=1) 2/4

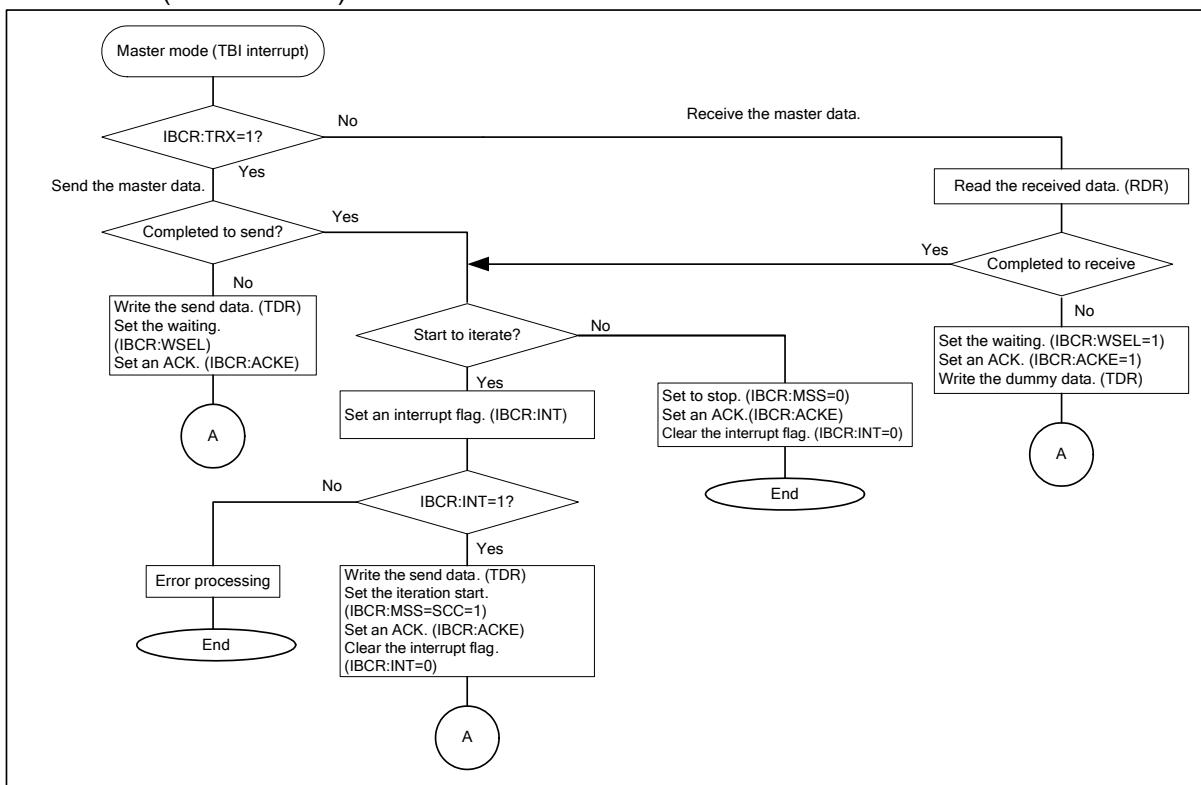


Figure 4-6 I²C flowchart example (FIFO not used) when DMA mode is enabled
(SSR:DMA=1) 3/4

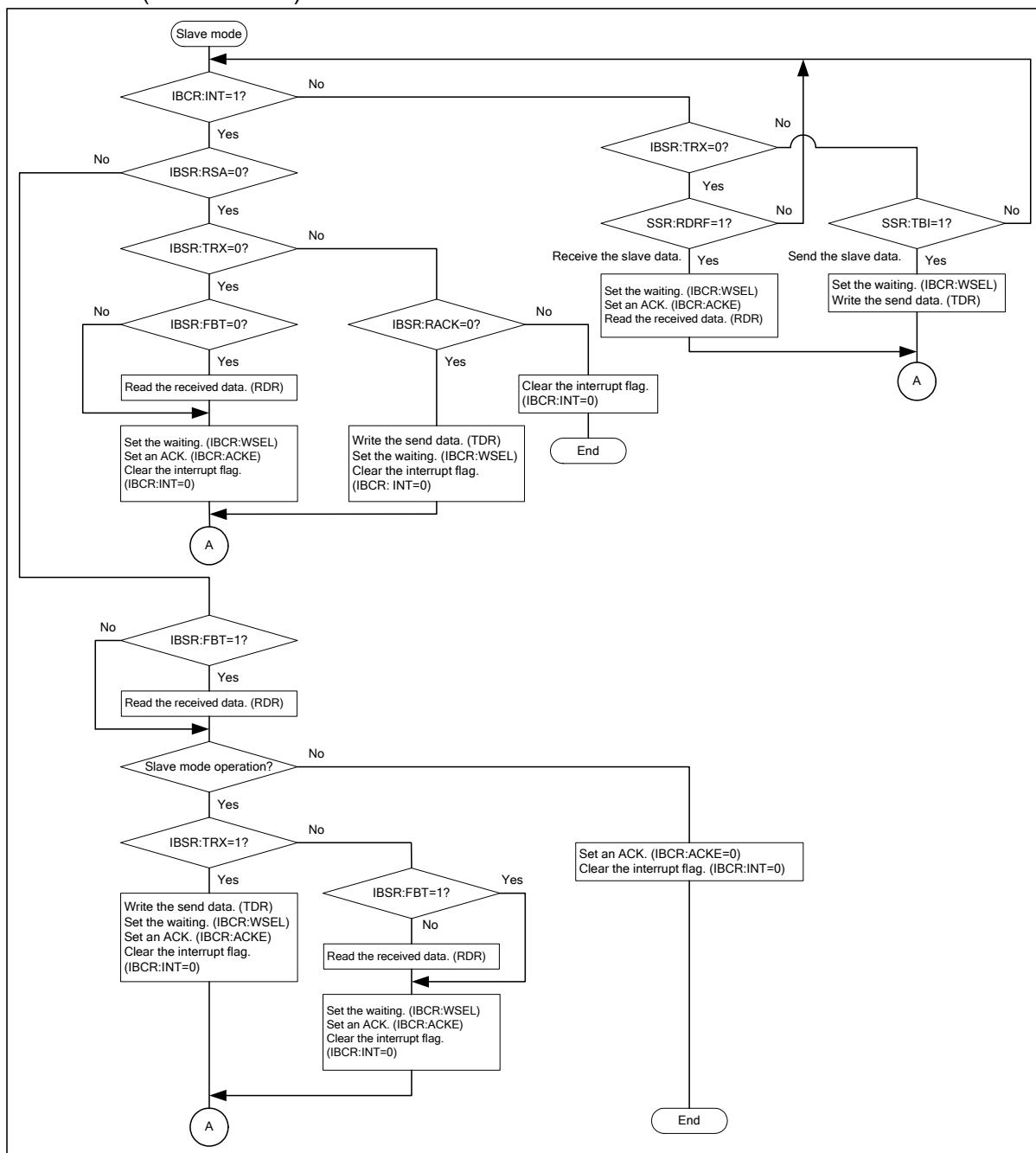
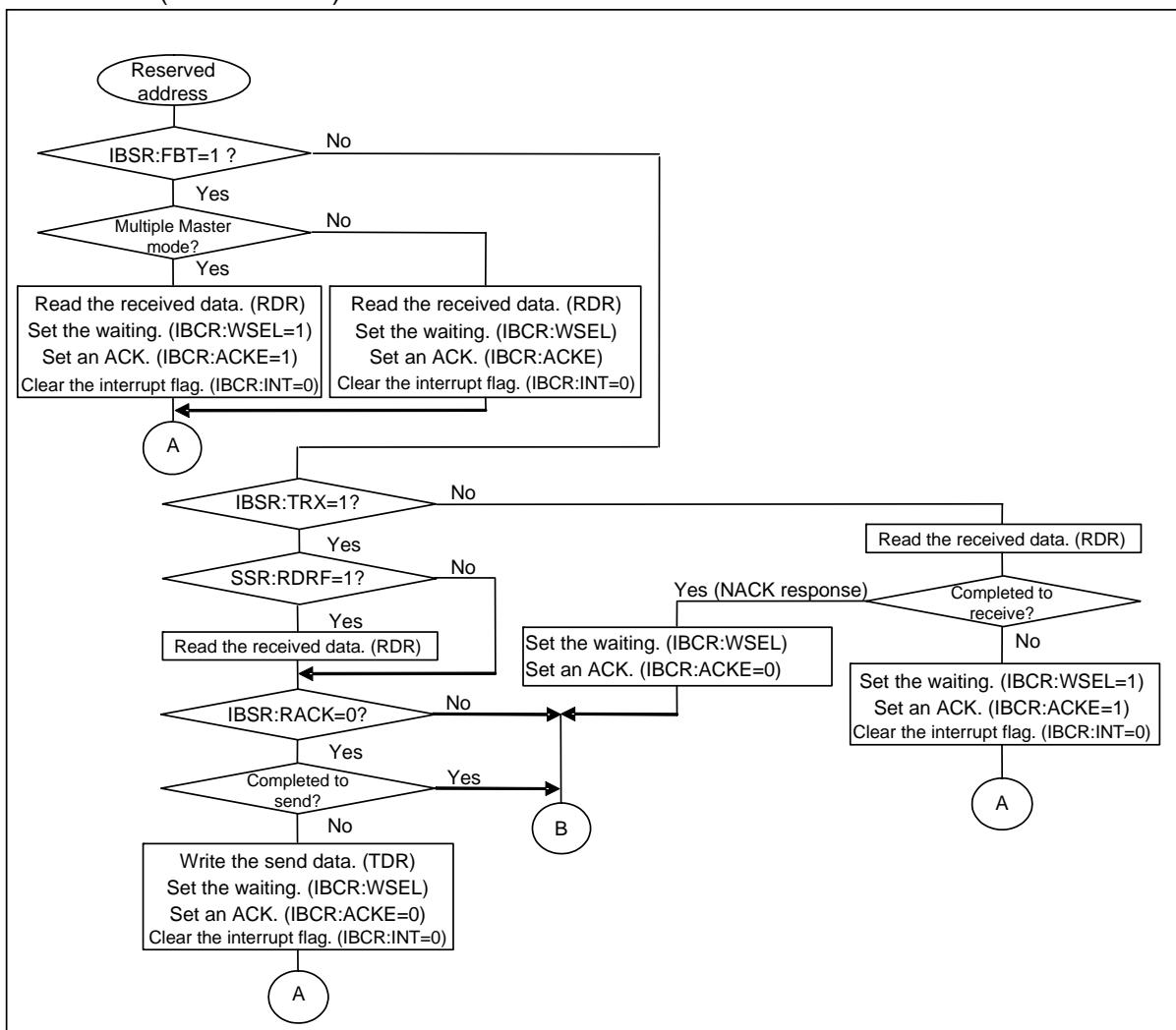


Figure 4-7 I²C flowchart example (FIFO not used) when DMA mode is enabled
(SSR:DMA=1) 4/4



<Note>

The flow shows an outline of operation settings in I²C mode. To perform the appropriate operations, take into account error processing based on applications.

5. I²C Interface Registers

The following lists the I²C interface registers.

■ List of I²C interface registers

Table 5-1 List of I²C interface registers

	bit 15	bit 8	bit 7	bit 0
I ² C	IBCR (I ² C Bus Control Register)		SMR (Serial Mode Register)	
	SSR (Serial Status Register)		IBSR (I ² C Bus Status Register)	
	-		RDR/TDR (Transmit/Receive Data Register)	
	BGR1 (Baud Rate Generator Register 1)		BGR0 (Baud Rate Generator Register 0)	
	ISMK (7-bit Slave Address Mask Register)		ISBA (7-bit Slave Address Register)	
FIFO	FCR1 (FIFO Control Register 1)		FCR0 (FIFO Control Register 0)	
	FBYTE2 (FIFO2 Byte Register)		FBYTE1 (FIFO1 Byte Register)	

Table 5-2 I²C Interface bit assignment

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	WUCR	RIE	TIE	ITST1	ITST0
SSR/ IBSR	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
TDR1/ TDR0	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

5.1. I²C Bus Control Register (IBCR)

The I²C Bus Control Register (IBCR) is used to select master or slave mode, generate an iteration start condition, enable an acknowledgement, enable an interrupt, and display an interrupt flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	(SMR)		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W			
Initial value	0	0	0	0	0	0	0	0			

[bit 15] MSS: Master/slave select bit

- If this bit is set to "1" when the I²C bus is in idle state (ISMK:EN="1", IBSR:BB="0"), master mode is selected.
- If this bit is set to "1" when the BB bit of IBSR register is "1", the occurrence of start condition is waited until the IBSR:BB bit is set to "0". If the slave address matches and the slave operation is started during waiting, this bit is set to "0" and the AL bit of IBSR register is set to "1".
- When master mode is selected (MSS="1", ACT="1") and the interrupt flag (INT) is "1", a stop condition is generated when this bit is set to "0".

The MSS bit is cleared in any of the following conditions.

- When the I²C interface operation is disabled (ISMK:EN bit="0")
- When an arbitration lost occurs
- When a bus error is detected (BER bit="1")
- When the MSS bit is set to "0" if INT="1"
- When DMA mode is enabled (SSR:DMA=1), SSR:TBI="1", and when the MSS bit is set to "0"

The following provides the relation between MSS and ACT bits.

MSS bit	ACT bit	State
0	0	Idle
0	1	The slave address matching or the reserved address is acknowledged (*1), and slave mode is selected.
1	0	The master operation is waited.
1	1	During master mode operation (in master mode)

*1) Acknowledgment: The SDA is LOW on the I²C bus during acknowledgement.

Bit	Description
0	Selects slave mode.
1	Selects master mode.

<Notes>

- If DMA mode is disabled (SSR:DMA=0) and the MSS bit is set to "1", the MSS bit must be set to "0" only when the MSS bit is "1" and the INT bit is "1". If the MSS bit is set to "0" when the ACT bit is "1", the INT bit is also cleared to "0".
- If DMA mode is enabled (SSR:DMA=1) and the MSS bit is set to "1", the MSS bit must be set to "0" only when the MSS bit is "1" and the INT bit is "1", or the SSR:TBI bit is "1". If the MSS bit is set to "0" when the ACT bit is "1", the INT bit is also cleared to "0".
- When master mode is selected, the MSS bit is read to be "1" even when it is set to "0" while the ACT bit is "1".

[bit14] ACT/SCC : Operation flag/iteration start condition generation bit

This bit setting has a different meaning when it is written and read.

During reading	During writing
ACT bit	SCC bit

The ACT bit indicates the current operation in master or slave mode.

The ACT bit is set when:

1. The start condition is output onto the I²C bus (master mode)
2. The slave address matches the address sent from the master device (slave mode)
3. The reserved address is detected and it is acknowledged (If MSS is "0", slave mode is selected.)

The ACT bit is reset when:

<Master mode>

1. The stop condition is detected.
2. An arbitration lost is detected.
3. A bus error is detected
4. The I²C interface operation is disabled (ISMK:EN bit="0")

<Slave mode>

1. The (iteration) start condition is detected
2. The stop condition is detected.
3. The reserved address is detected (IBSR:RSA="1") but not acknowledged
4. The I²C interface operation is disabled (ISMK:EN bit="0")
5. When a bus error is detected (BER bit="1")

If this bit is set to "1" in master mode, the iteration start is executed. This bit is disabled to set to "0".

Bit	Description	
	During writing	During reading
0	No effect	No operation
1	Generates an iteration start condition.	During the I ² C operation

<Notes>

- The SCC bit must be set to "1" during an interrupt of master mode (when MSS="1", ACT="1" and INT="1") only. If the SCC bit is set to "1" when the ACT bit is "1", the INT bit is cleared to "0".
- This bit must not be set to "1" in slave mode (when MSS="0" and ACT="1").
- If the SCC bit is set to "1" and if the MSS bit is set to "0" simultaneously, the MSS bit setting is preceded.
- When data is read by a read-modify-write instruction, the SCC bit is read.
- If both of the following conditions are satisfied, the INT bit is set to "1" and the I²C bus is waited (SCL=LOW). To generate an iteration start condition, clear the INT bit by setting the SCC bit to "1" again.
 - The SCC bit is set to "1" during master mode interrupt at bit 8 (MSS="1", ACT="1", INT="1" and WSEL="1").
 - A negative acknowledgement (NACK) is received at bit 9.
- When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.
 1. Set the IBCR:INT bit to "1".
 2. Check that the IBCR:INT bit is set to "1".
 3. Write the slave address in the TDR.
 4. Set this bit to "1".

[bit 13] ACKE: Data byte acknowledge enable bit

- If this bit is set to "1", LOW is output when acknowledged.
- This bit must be changed if any of the following conditions has occurred:
 - If DMA mode is disabled (SSR:DMA=0), the ACT bit is "1", and the INT bit is "1"
 - If DMA mode is enabled (SSM:DMA=1), the ACT bit is "1", and the SSR:TBI bit is "1"
 - If DMA mode is enabled (SSM:DMA=1), the ACT bit is "1", the slave reception is selected, and the SSM:RDRF is "1"
 - If the ACT bit is "0"

This bit is invalid in the following conditions.

1. During acknowledgement to an address field other than the reserved address (automatic generation)
2. During data transmission (IBSR:RSA="0", IBSR:TRX="1", IBSR:FBT="0")
3. If the receive FIFO is enabled and the slave mode reception is selected (FCR0:FE="1", MSS="0", ACT="1"), an ACK is returned.
4. If the receive FIFO is enabled, the WSEL bit is "0", the master mode reception is selected (FCR0:FE="1", MSS="1", ACT="1", WSEL="0"), and the SSR:TDRE bit is "0", an ACK is always returned. If the SSR:TDRE bit is "1", a NACK is returned.
5. If the receive FIFO is enabled, WSEL="0", the reserved address is detected and the slave transmission is selected (IBSR:RSA="1", IBSR:TRX="1", IBSR:FBT="1"), an ACK is always returned. To respond with a NACK, disable the receive FIFO and set the ACKE bit to "0" during interrupt after detection of the reserved address.
6. The receive FIFO is enabled, the WSEL bit is "1", the master mode reception is selected, and the Transmit Data Register has data (FCR0:FE="1", MSS="1", ACT="1", WSEL="1", SSR:TDRE="0")

Bit	Description
0	Disables acknowledgment.
1	Enables acknowledgement.

[bit 12] WSEL: Wait selection bit

- If DMA mode is disabled (SSR:DMA=0), this bit selects a generation time of interrupt before or after acknowledgement (INT="1") and selects to wait the I²C bus or not.
- If DMA mode is enabled (SSR:DMA=1), this bit selects a generation time of interrupt before or after acknowledgement (INT="1", and SSR:TBI="1" for transmission or SSR:RDRF="1" for reception) and selects to wait the I²C bus or not.
- The WSEL bit is invalid in the following conditions.
 1. An interrupt occurs (INT=1) for the first byte. (*1)
 2. The reserved address is detected (IBSR:FBT="1", IBSR:RSA="1").
 3. The NACK response is detected during FIFO data transmission (FCR0:FE="1", IBSR:RACK="1", ACT="1"). (*2)
 4. The receive FIFO is filled with data during FIFO reception.

*1) The first byte indicates data after the (iteration) start condition.
*2) NACK response: The SDA bit of I²C bus is HIGH during acknowledgement.

Bit	Description
0	Waits (9 bits) after acknowledgement.
1	Waits (8 bits) after data transmission or reception.

[bit 11] CNDE: Condition detection interrupt enable bit

This bit enables an interrupt if a stop condition or an iteration start condition is detected in master or slave mode (ACT="1"). An interrupt occurs if the RSC or SPC bit of IBSR register is "1" and if this bit is set to "1".

Bit	Description
0	Disables an interrupt due to the iteration start or stop condition.
1	Enables an interrupt due to the iteration start or stop condition.

[bit 10] INTE: Interrupt enable bit

This bit enables an interrupt (INT="1") due to a data transmission or bus error in master or slave mode.

Bit	Description
0	Disables an interrupt.
1	Enables an interrupt.

[bit 9] BER: Bus error flag bit

This bit indicates that an error has been detected on the I²C bus.

The BER bit is set when:

1. The start or stop condition is detected during transmission of the first byte. (*1)
2. The (iteration) start condition or the stop condition is detected at bit 2 to 9 (acknowledgement) of data after the 2nd or subsequent byte.

The BER bit is reset when:

1. The INT bit is set to "0" if BER="1".
 2. The I²C interface operation is disabled (ISMK:EN bit="0").
- *1) The first byte indicates data after the (iteration) start condition.

Bit	Description
0	No error
1	An error was detected.

<Note>

Check this bit state if the interrupt flag (INT bit) is "1". If it is "1", the normal data transmission and reception fail. Retransmit the data.

[bit 8] INT: interrupt flag bit

The interrupt flag bit is set to "1" after 8 or 9 bits (ACK) of data have been transmitted or when a bus error has occurred in master or slave mode. During operation other than bus error, if the INT bit is set to "1", the SCL flag is set to LOW. If the INT bit is set to "0", the SCL flag is released from the LOW state.

The INT bit is set when:

<bit 8>

<If DMA mode is not related>

1. The reserved address is detected in the first byte.
2. The WSEL bit is "1" and an arbitration lost is detected in the 2nd or subsequent byte.

<If DMA mode is disabled (SSR:DMA=0)>

1. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", master mode is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
2. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", slave mode is selected, the receive FIFO is disabled, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
3. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
4. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", the receive FIFO is disabled, and the slave mode reception is selected.

<If DMA mode is enabled (SSR:DMA=1)>

1. If DMA mode is enabled (SSR:DMA=1), WSEL bit is "1", master mode is selected, the SSR:TBI bit is "1" in the 2nd or subsequent byte, and the INT bit is set to "1".

<bit 9>

<If DMA mode is not related>

1. An arbitration lost is detected in the first byte.
2. The NACK signal is received during the time other than stop condition output setting (the MSS bit is set to "0" during the master mode operation).
3. The WSEL bit is "0" and an arbitration lost is detected in the 2nd or subsequent byte.
4. The reserved address is not detected in the 1st byte, and data is found in the receive FIFO when the receive FIFO is enabled and data is received in master or slave mode (IBSR:TRX=0).

<If DMA mode is disabled (SSR:DMA=0)>

1. If DMA mode is disabled (SSR:DMA=0), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when data is transmitted (IBSR:TRX=1) in master or slave mode.
2. If DMA mode is disabled (SSR:DMA=0), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when the receive FIFO is disabled for data reception (IBSR:TRX=0) in master or slave mode.
3. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", and the SSR:TDRE bit is "1" in the 2nd or subsequent byte during the master mode operation.
4. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", and the SSR:TDRE bit is "1" in the 2nd or subsequent byte during the slave mode transmission.
5. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", the receive FIFO is disabled, and the slave mode reception is selected. However, if the reserved address is detected in the 1st byte during the slave mode reception, no interrupt is generated by bit 9.
6. If DMA mode is disabled (SSR:DMA=0), the receive FIFO is enabled, data is received in slave mode, and the receive FIFO is filled with data.

<If DMA mode is enabled (SSR:DMA=1)>

1. If DMA mode is enabled (SSR:DMA=1), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when data is transmitted (IBSR:TRX=1) in slave mode.
2. If DMA mode is enabled (SSR:DMA=1), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when the receive FIFO is disabled for data reception (IBSR:TRX=0) in slave mode.
3. If DMA mode is enabled (SSR:DMA=1), WSEL bit is "0", the SSR:TBI bit is "1" in the 2nd or subsequent byte during the master mode operation, and the INT bit is set to "1".

<Others>

1. bus error is detected.

The INT bit is reset when:

1. The INT bit is set to "0".
2. The INT bit is "1" and the ACT bit is "1", the MSS bit is set to "0".
3. The INT bit is "1" and the ACT bit is "1", the SCC bit is set to "1".

If the DMA mode is disabled (SSR:DMA=0), it is invalid to set the INT bit to "1".

Bit	Description	
	During writing	During reading
0	Clears the INT bit.	Does not issue an interrupt request.
1	No effect	Issues an interrupt request.

<Notes>

- When DMA mode is enabled (SSR:DMA=1) and the SSR:TBI bit is "1" in the 2nd or subsequent byte during the master mode operation, a status interrupt (SIRQ="1") is not generated even when the INT bit is set to "1".
- When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.
 1. Set the IBCR:INT bit to "1".
 2. Check that the IBCR:INT bit is set to "1".
 3. Write the slave address in the TDR.
 4. Set the IBCR:SCC bit to "1".
- If the INT flag is changed from "1" to "0", the I²C bus is released from waiting.
- If the ISMK:EN bit is set to "0", the SSR:RDRF and INT bits may be set to "1" in certain receive timing. If so, read the received data and clear the INT bit.
- When a read-modify-write instruction is issued, "1" is read.
- If the receive FIFO is enabled, the INT bit is not set to "1" even when the receive FIFO is filled with data during the master mode reception.
- Set this bit to "1" when the start condition is issued (IBCR:MSS=1).

5.2. Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set an operation mode, and to enable or disable the transmit/receive interrupt.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SCR)		MD2	MD1	MD0	WUCR	RIE	TIE	ITST1	ITST0	
Attribute			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value			0	0	0	0	0	0	0	0	0

[bit 7:5] MD2, MD1, MD0: operation mode set bits

These bits set an operation mode.

"0b000": Sets operation mode 0 (async normal mode).

"0b001": Sets operation mode 1 (async multiprocessor mode).

"0b010": Sets operation mode 2 (clock sync mode).

"0b011": Sets operation mode 3 (LIN communication mode).

"0b100": Sets operation mode 4 (I²C mode).

This section explains the registers and their operation in operation mode 4 (I²C mode).

Bit 7	Bit 6	Bit 5	Description
0	0	0	Operation mode 0 (async normal mode)
0	0	1	Operation mode 1 (async multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I ² C mode)

* This section explains the registers in operation mode 4.

<Notes>

- Any bit setting other than above is inhibited.
- To switch the current operation mode, disable the I²C (ISMK:EN="0") and change the operation mode.
- After the operation mode has been switched, set each register correctly.

[bit 4] WUCR: Wake-up control bit

Selects a pin to be used for an external interrupt.

If this bit is set to "0", the INT pin is used for an external interrupt.

If this bit is set to "1", the SDA or SCL pin is used for an external interrupt.

Bit	Description
0	Disables the Wake-up function.
1	Enables the Wake-up function.

[bit 3] RIE: Receive interrupt enable bit

- This bit enables or disables an output of receive interrupt request to the CPU.
- If the RIE bit and the receive data flag bit (SSR:RDRF) are "1", or if any of error flag bits (SSR:ORE) is "1", a receive interrupt request is output.

Bit	Description
0	Disables the receive interrupt.
1	Enables the receive interrupt.

<Note>

To receive data using the INT bit of I²C Bus Control Register (IBCR) when DMA mode is disabled (SSR:DMA=0), set this bit to "0".

[bit 2] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of Transmit Interrupt Request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a Transmit Interrupt Request is output.

Bit	Description
0	Disables the transmit interrupt.
1	Enables the transmit interrupt.

<Note>

To transmit data using the INT bit of I²C Bus Control Register (IBCR) when DMA mode is disabled (SSR:DMA=0), set this bit to "0".

[bit 1:0] ITST1, ITST0: I²C test bits

They are I²C Test bits.

They must always be set to "0".

Bit 1:0	Description
0	Disables the I ² C test.
1	Enables the I ² C test.

<Note>

If this bit is set to "1", the I²C test is executed.

5.3. I²C Bus Status Register (IBSR)

The I²C Bus Status Register (IBSR) shows the iteration start, acknowledgement, data direction, arbitration lost, stop condition, I²C bus status, and bus error detection.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(SSR)		FBT	RACK	RSA	TRX	AL	RSC	SPC	BB	
Attribute			R	R	R	R	R	R/W	R/W	R	
Initial value			0	0	0	0	0	0	0	0	0

[bit 7] FBT: First byte bit

This bit indicates the first byte.

The FBT bit is set when:

1. The (iteration) start condition is detected.

The FBT bit is cleared when:

1. The second byte is sent or received.
2. The stop condition is detected.
3. The I²C interface operation is disabled (ISMK:EN bit="0").
4. A bus error is detected (IBCR:BER bit="1").

Bit	Description
0	Other than 1st byte
1	The 1st byte is being sent or received.

[bit 6] RACK: Acknowledge flag bit

This bit shows acknowledgement being received in the 1st byte or in master or slave mode.

The RACK bit is updated when:

1. Acknowledged in the 1st byte.
2. Data is acknowledged in master or slave mode.

The RACK bit is cleared (RACK bit="0") when:

1. The (iteration) start condition is detected.
2. The I²C interface operation is disabled (ISMK:EN bit="0").
3. A bus error is detected (IBCR:BER bit="1").

Bit	Description
0	LOW is received.
1	HIGH is received.

[bit 5] RSA: Reserved address detection bit

This bit shows that the reserved address has been detected.

The RSA bit is set (RSA="1") when:

1. The 1st byte is "0000xxxx" or "1111xxxx", where, "x" can be "0" or "1".

The RSA bit is reset (RSA="0") when:

1. The (iteration) start condition is detected.
2. The stop condition is detected.
3. The I²C interface operation is disabled (ISMK:EN bit="0").
4. A bus error is detected (IBCR:BER bit="1").

If the RSA bit is set to "1" in the 1st byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL flag is set to "L" at the falling edge of SCL (bit 8) of the 1st byte regardless of FIFO enable or disable state. To read the received data and start the slave mode operation during this time, set the IBCR:ACKE bit to "1" and clear the interrupt flag (IBCR:INT) to "0". If the TRX bit is "0" after that, data is received in slave mode. To stop the data reception, set the IBCR:ACKE bit to "0". No data is received after that.

Bit	Description
0	The reserved address is not detected.
1	The reserved address is detected.

<Notes>

- If the IBCR:ACKE bit is set to "0" during data transfer, this IBCR:ACKE bit cannot be set to "1" until the stop condition or the iteration start condition is detected.
- If the slave mode transmission is detected during an interrupt by reserved address detection and if the receive FIFO is enabled, an ACK response is returned. In this case, disable the receive FIFO and set the IBCR:ACKE bit to "0".

[bit 4] TRX: Data direction bit

This bit indicates the data transmission direction.

The TRX bit is set when:

1. The (iteration) start condition is sent in master mode.
2. Bit 8 of the 1st byte is "1" in slave mode (in the slave mode transmission direction).

The TRX bit is reset when:

1. An arbitration lost occurs (AL="1").
2. Bit 8 of the 1st byte is "0" in slave mode (in the slave mode reception direction).
3. Bit 8 of the 1st byte is "1" in master mode (in the master mode reception direction).
4. The stop condition is detected.
5. The (iteration) start condition is detected in any mode other than master mode.
6. The I²C interface operation is disabled (ISMK:EN bit="0").
7. A bus error is detected (IBCR:BER bit="1").

Bit	Description
0	Receive direction
1	Transmission direction

[bit3] AL: Arbitration lost bit

This bit indicates an arbitration lost.

The AL bit is set when:

1. The output data does not match the receive data in master mode.
2. The IBCR:MSS bit is set to "1" but the slave mode operation is selected.
3. The iteration start condition is detected by bit 1 of the 2nd or subsequent byte data in master mode.
4. The stop condition is detected by bit 1 of the 2nd or subsequent byte data in master mode.
5. The iteration start condition cannot be generated in master mode.
6. The stop condition cannot be generated in master mode.

The AL bit is reset when:

1. The IBCR:MSS bit is set to "1".
2. The IBCR:INT bit is set to "0".
3. The SPC bit is set to "0" when both AL and SPC bits are "1".
4. The I²C interface operation is disabled (ISMK:EN bit="0").
5. A bus error is detected (IBCR:BER bit="1").

Bit	Description
0	No arbitration lost has occurred.
1	An arbitration lost has occurred.

[bit 2] RSC: Iteration start condition check bit

This bit shows that an iteration start condition is detected in master or slave mode.

The RSC bit is set when:

1. When an iteration start condition is detected after acknowledgement, during the master or slave mode operation.

The RSC bit is reset when:

1. The RSC bit is set to "0".
2. The IBCR:MSS bit is set to "1".
3. The I²C interface operation is disabled (ISMK:EN bit="0").

It is invalid to set this bit to "1".

Bit	Description
0	No iteration start condition has been detected.
1	An iteration start condition has been detected.

<Notes>

- If no acknowledgement response is sent while data is received in slave mode due to the reserved address being detected, slave mode is released. In this case, this bit is not set to "1" even if the next iteration start condition is detected.
- When a read-modify-write instruction is issued, "1" is read.

[bit 1] SPC: Stop condition check bit

This bit shows that a stop condition is detected in master or slave mode.

The SPC bit is set when:

1. The stop condition is detected in the master or slave mode operation.
2. In master mode, the stop condition has occurred and, therefore, an arbitration lost has occurred.

The SPC bit is reset when:

1. This bit is set to "0".
2. The IBCR:MSS bit is set to "1".
3. The I²C interface operation is disabled (ISMK:EN bit="0").

It is invalid to set this bit to "1".

Bit	Description	
0	No stop condition is detected.	
1	Master mode	An arbitration lost has occurred when the stop condition is detected or when it is output.
	Slave mode	The stop condition is detected.

<Notes>

- If no acknowledgement response is sent while data is received in slave mode due to the reserved address being detected, slave mode is released. In this case, this bit is not set to "1" even if the next stop condition is detected.
- When a read-modify-write instruction is issued, "1" is read.

[bit 0] BB: Bus state bit

This bit shows the bus state.

The BB bit is set when:

1. LOW is detected in SDA or SCL of the I²C bus.

The BB bit is reset when:

1. The stop condition is detected.
2. The I²C interface operation is disabled (ISMK:EN bit="0").
3. A bus error is detected (IBCR:BER bit="1").

Bit	Description
0	The bus is in idle state.
1	The bus is in transmission state.

5.4. Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the transmission or reception state.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI		(IBSR)	
Attribute	R/W	R/W	R/W	R/W	R	R	R	R			
Initial value	0	0	0	0	0	0	1	1			

[bit 15] REC: Receive error flag clear bit

This bit clears the ORE bit of Serial Status Register (SSR).

- If this bit is set to "1", the ORE bit is cleared.
- This bit has no effect if set to "0".

When it is read, "0" is always read.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	Clears the Receive Error flag (ORE).	

[bit 14] TSET: Transmit empty flag set bit

This bit sets the TDRE bit of Serial Status Register (SSR).

- If it is set to "1" and if the TDRE bit and DMA mode are enabled (DMA=1), the TBI bit is set.
- This bit has no effect if set to "0".

When it is read, "0" is always read.

Bit	Description	
	Write	Read
0	No effect.	"0" is always read.
1	The TDRE bit is set.	

<Note>

Set this bit to "1" only when the IBCR:INT bit is "1".

[bit 13] DMA: DMA mode enable bit

This bit enables or disables DMA mode.

- If this bit is set to "1", an interrupt condition is generated during DMA transfer.
- If this bit is set to "0", an interrupt condition is generated during normal data transfer.

For details, see Table 2-1.

Bit	Description
0	Disables DMA mode.
1	Enables DMA mode.

<Note>

This bit state can be changed only when the ISMK:EN bit is "0".

[bit 12] TBIE: Transmit bus idle interrupt enable bit (Effective only when DMA mode is enabled)

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If DMA mode is enabled (DMA="1") and both TBIE and TBI bits are "1", a transmit bus idle interrupt request is output.
- If DMA mode is disabled (DMA="0"), this bit is set to "0". In such case, this bit is set to "0". If data is written, this writing is ignored and the "0" is maintained.

Bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit 11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SMR:RIE bits are "1", a receive interrupt request is output.
- If this flag is set, the Receive Data Register (RDR) is invalid.
- If the receive FIFO is used and if this flag is set, the received data is not stored in the receive FIFO.

Bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit 10] RDRF: Receive data full flag bit

- This flag shows the state of Receive Data Register (RDR).
- If the SMR:RIE bit and the receive data flag bit (RDRF) are "1", a receive interrupt request is issued.
- When the receive data is loaded in the RDR, this bit is set to "1". When data is read from the Receive Data Register (RDR), this bit is cleared to "0".
- This bit is set at the falling edge of SCL signal (bit 8 of data).
- This bit is also set even when a NACK is responded. (*1)
- If the receive FIFO is used and if a certain count of data is received by the receive FIFO, the RDRF bit is set to "1".
- If the receive FIFO is used and if this buffer is emptied, this bit is cleared to "0".
- If all of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (SSR:RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FCR:FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
 - The IBCR:BER bit is "0".

If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted.

*1) NACK response: The SDA bit of I²C bus is "H" during acknowledgement.

Bit	Description
0	The Receive Data Register (RDR) is empty.
1	The Receive Data Register (RDR) contains data.

<Notes>

- If all of the following conditions are satisfied, the SCL flag is set to LOW after acknowledgement was transmitted. If the RDRF bit is set to "0", the SCL flag is released from the LOW state.
 - The receive FIFO is not used.
 - DMA mode is enabled (IBCR:DMA=1).
 - Data is received in the 2nd or subsequent byte (IBSR:TRX=0), and the RDRF bit is "1".
 - The IBCR:WSEL bit is "0".
- If all of the following conditions are satisfied, the SCL flag is set to LOW immediately after single-byte data reception. If the RDRF bit is set to "0", the SCL flag is released from the LOW state.
 - The receive FIFO is not used.
 - DMA mode is enabled (IBCR:DMA=1).
 - Data is received in the 2nd or subsequent byte (IBSR:TRX=0), and the RDRF bit is "1".
 - The IBCR:WSEL bit is "1".
- If the receive FIFO is used and DMA mode is enabled for data reception (DMA=1), the SCL flag is set to LOW when the receive FIFO is filled with data. If data is read from the RDR even once, the SCL flag is released from the LOW state.

[bit 9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If the SMR:TIE and TDRE bits are "1", a Transmit Interrupt Request is output.
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to a shift register for transmission and its transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TSET bit of Serial Status Register (SSR) is set to "1", this flag is set. If an arbitration lost or a bus error is detected, use this flag to set the TDRE bit to "1".

Bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register is empty.

[bit 8] TBI: Transmit bus idle flag bit (Effective only when DMA mode is enabled)

This bit shows that no data is sent by the I²C when DMA mode is enabled (DMA=1). If DMA mode is enabled (DMA=1) and the TBI bit is set to "1" in the 2nd or subsequent byte, the SCL flag is set to LOW. If the TBI bit is set to "0", the SCL flag is cleared from the LOW state.

The TBI bit is set when:

<bit 8>

1. The WSEL bit is "1", master mode is selected, and the TDRE bit is "1" in the 2nd or subsequent byte.
2. The WSEL bit is "1", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.

<bit 9>

1. Master mode is selected, the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1".
2. The WSEL bit is "0", master mode is selected, and the TDRE bit is "1" in the 2nd or subsequent byte.
3. The WSEL bit is "0", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.

<Others>

The transmit buffer empty flag set bit (TSET) is set to "1".

The TBI bit is reset when:

1. The transmit data is written in the Transmit Data Register (TDR).

If this bit is "1" and if the transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

- If DMA mode is disabled (DMA="0"), this bit is undefined.

Bit	Description
0	During data transmission
1	No data transmission

5.5. Receive Data Register/Transmit Data Register (RDR/TDR)

The Receive and Transmit Data Registers are allocated at the same address. This register functions as the Receive Data Register when data is read from it. This register operates as the Transmit Data Register when data is written in it.

■ Receive Data Register (RDR)

bit	15	...	8	7	6	5	4	3	2	1	0
Field				D7	D6	D5	D4	D3	D2	D1	D0
Attribute				R	R	R	R	R	R	R	R
Initial value				0	0	0	0	0	0	0	0

The Receive Data Register (RDR) is a data buffer register for serial data reception.

- When a serial data signal is sent to the serial data line (SDA pin), it is converted by a shift register and stored in the Receive Data Register (RDR).
- When the first byte is received, a received address is not stored in the Receive Data Register (RDR). However, when the first byte is a reserved address, a received address is stored in the Receive Data Register (RDR). In this case, the least significant bit (RDR:D0) is the data direction bit. (*1)
- When the received data is stored in the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is set to "1".
- When data is read from the Receive Data Register (RDR), the receive data full flag bit (SSR:RDRF) is cleared to "0" automatically.

*1) The first byte indicates data after the (iteration) start condition.

<Notes>

- If the receive FIFO is used and if a certain count of data is received by the receive FIFO, the SSR:RDRF bit is set to "1".
- If the receive FIFO is used and if this buffer is emptied, the SSR:RDRF bit is cleared to "0".

■ Transmit Data Register (TDR)

bit	15	...	8	7	6	5	4	3	2	1	0
Field				D7	D6	D5	D4	D3	D2	D1	D0
Attribute				W	W	W	W	W	W	W	W
Initial value				1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a data buffer register for serial data transmission.

- Data of the Transmit Data register (TDR) is output to the serial data line (SDA pin) with the MSB first order.
- When the first byte is transmitted, the least significant bit (TDR:D0) indicates the data transmission direction.
- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When data is transferred to a shift register for transmission, the transmit data empty flag (SSR:TDRE) is set to "1".
- If transmit FIFO is disabled and if the data empty flag (SSR:TDRE) is "0", the transmit data cannot be written in the Transmit Data Register (TDR).
- If transmit FIFO is used, the transmit data can be written until this buffer is filled with it even if the data empty flag (SSR:TDRE) is "0".

<Note>

The Transmit Data Register is a write-only register. While the Receive Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instructions cannot be used.

5.6. 7-bit Slave Address Mask Register (ISMK)

The 7-bit Slave Address Mask Register (ISMK) is used to compare or set each bit of the slave address.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0		(ISBA)	
Attribute	R/W										
Initial value	0	1	1	1	1	1	1	1			

[bit 15] EN: I²C interface operation enable bit

This bit enables or disables the I²C interface operation.

If set to "0": The I²C interface operation is disabled.

If set to "1": The I²C interface operation is enabled.

Bit	Description
0	Disable
1	Enable

<Notes>

- This bit is not cleared to "0" even if the BER bit of IBSR register is set to "1".
- The baud rate generator must be set only when this bit is "0".
- When this bit is "0", set both the 7-bit Slave Address Register and the 7-bit Slave Address Mask Register.
- If the I²C interface operation is disabled (EN="0"), data transmission is inhibited immediately.
- If you have set the IBCR:MSS bit to "0" to generate a Stop condition and if you wish to disable the I²C interface operation, make sure that the stop condition has occurred. Then, disable the interfacing (EN="0").
- If the EN bit is set to "0" during data transmission, a pulse may be generated on the SDA/SCL signal of the I²C bus.

[bit 14:8] SM6 to SM0: Slave address mask bits

These bits specify to exclude the 7-bit slave address and the received address from comparison.

If set to "1", the address is compared.

If set to "0", the address matching is assumed.

Bit 14:8	Description
0	Does not compare the bits.
1	Compares the bits.

<Note>

This register must be set only when the EN bit is "0".

5.7. 7-bit Slave Address Register (ISBA)

The 7-bit Slave Address Register (ISBA) is used to set the slave address.

bit	15	...	8	7	6	5	4	3	2	1	0
Field	(ISMK)		SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0	
Attribute			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value			0	0	0	0	0	0	0	0	0

[bit 7] SAEN: Slave address enable bit

This bit enables the slave address detection.

If set to "0": The slave address is not detected.

If set to "1": The ISBA and ISMK settings and the received 1st byte are compared.

Bit	Description
0	Disable
1	Enable

[bit 6:0] SA6 to SA0: 7-bit slave address

- If the slave address detection is enabled (SAEN=1), the 7-bit Slave Address Register (ISBA) compares the 7-bit data, which has been received after detection of (iteration) start condition, with this register value. If all bits match each other, slave mode is selected and an ACK is output. At this time, the received slave address is stored in this register (if SAEN=0, no ACK is output).
- If an address bit is set to "0" in the ISMK register, it is not compared.

Bit 6:0	Description
6-0	7-bit slave address

<Notes>

- The reserved address cannot be set.
- This register must be set only when the EN bit of ISMK register is "0".

5.8. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-							(BGR1)								
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Initial value	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.

The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/0 set value can be read.

When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the Reload counter starts its counting.

[bit 15] Reserved bit

This bit value is undefined during reading.

It has no effect during writing.

[bit 14:8] BGR1: Baud Rate Generator Register 1

Bit 14:8	Description
Write	Writes data in bit 8 to 14 of reload counter.
Read	Reads the BGR1 set value.

[bit 7:0] BGR0: BAUD RATE GENERATOR REGISTER 0

Bit 7:0	Description
Write	Writes data in bit 0 to 7 of reload counter.
Read	Reads the BGR0 set value.

<Notes>

- Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) by 16-bit data accessing.
- The Baud Rate Generator Registers must be set when the EN bit of ISMK register is "0".
- The baud rate must be set regardless of master or slave mode selection.
- In operation mode 4 (I²C mode), operate the bus clock at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.

5.9. FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select the transmit or receive FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL		(FCR0)	
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W			
Initial value	0	0	-	0	-	1	0	0			

[bit 15:14] FTST1, FTST0: FIFO test bits

They are FIFO Test bits.

They must always be set to "0".

Bit 15:14	Description
0	Disables the FIFO test.
1	Enables the FIFO test.

<Note>

If this bit is set to "1", the FIFO test is executed.

[bit 13] Reserved bit

This bit value is undefined during reading.

This bit has no effect during writing.

[bit 12] FLSTE: Re-transmit data lost detection enable bit

This bit enables the FCR0:FLST bit detection.

If set to "0", the FCR0:FLST bit detection is disabled.

If set to "1", the FCR0:FLST bit detection is enabled.

Bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

<Note>

If you wish to set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".

[bit 11] FRIIE: Receive FIFO idle detection enable bit

This bit sets to detect the receive idle state if the receive FIFO contains valid data and if it continues more than 8-bit hours. If the receive interrupt is enabled (SCR:RIE=1), a receive interrupt is generated when the receive idle state is detected.

If set to "0", a detection of receive idle state is disabled.

If set to "1", a detection of receive idle state is enabled.

Bit	Description
0	Disables the receive FIFO idle detection.
1	Enables the receive FIFO idle detection.

<Note>

In case of using Receive FIFO, set this bit to "1".

[bit 10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. If the Transmit Interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

Bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

<Notes>

- If the FBYTE (for transmission) is "0", this bit cannot be set to "0".
- If this bit is "0", the FSEL bit state cannot be changed.
- If this bit is set to "1", it has no effect on the operation.
- If a read-modify-write instruction is issued, "1" is read.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".

[bit 9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

Bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit 8] FSEL: FIFO buffer selection bit

This bit selects the transmit or receive FIFO.

If set to "0", transmit FIFO is assigned FIFO1, and the receive FIFO is assigned FIFO2.

If set to "1", transmit FIFO is assigned FIFO2, and the receive FIFO is assigned FIFO1.

Bit	Description
0	Set transmit FIFO as FIFO1, and the receive FIFO as FIFO2.
1	Set transmit FIFO as FIFO2, and the receive FIFO as FIFO1.

<Notes>

- This bit is not cleared by FIFO reset (FCL2, FCL1=1).
- To change this bit state, first disable the FIFO operation (FCR0:FE2, FE1=0).

5.10. FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15	...	8	7	6	5	4	3	2	1	0	
Field	(FCR1)				-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute					-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value					0	0	0	0	0	0	0	0

[bit 7] Reserved bit

When read, "0" is always read.
When writing, always set to "0".

[bit 6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- If the FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in the FIFO buffer.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data which has been saved by the FSET bit and identified by the read pointer is overwritten. The data re-transmission cannot be set by the FLD bit even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

Bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit 5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.

When the re-transmission setting has finished, this bit is set to "0".

Bit	Description
0	Not reloaded
1	Reloaded

<Notes>

- If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.
- When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".
- Set the SMR:TIE bit to "0" first, and set this bit to "1". Then, enable transmit FIFO and set the SMR:TIE bit to "1".

[bit 4] FSET: FIFO pointer save bit

This bit saves the read pointer value of transmit FIFO.

If the read pointer value is saved before being transmitted and if the FLST bit is "0", the data can be re-transmitted even if a communication error or others have occurred.

If set to "1", the current read pointer value is saved.

If set to "0", it has no effect.

Bit	Description
0	Not saved
1	Saved

<Note>

This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".

[bit 3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 buffer is initialized.

Only the FCR0:FLST bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO2 is reset.	

<Notes>

- Disable the FIFO2 operation first, and then reset the FIFO2 buffer.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The FBYTE2 register has the significant data count of "0".

[bit 2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 value.

If this bit is set to "1", the FIFO1 buffer is initialized.

Only the FCR0:FLST bit is initialized, but the other bits of FCR1/0 registers are kept.

Bit	Description	
	During writing	During reading
0	No effect.	"0" is always read.
1	FIFO1 is reset.	

<Notes>

- Disable the FIFO1 operation first, and then reset FIFO1.

- Set the transmit FIFO interrupt enable bit to "0" before the execution.

- The FBYTE1 register has the significant data count of "0".

[bit 1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If receive FIFO is selected by the FCR1:FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- To use FIFO2 as transmit FIFO, this bit must be set to "1" or "0" when the transmit data is empty (SSR:TDRE=1).
- To use FIFO2 as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and receive FIFO contains no valid data (FBYTE2=0) while the I²C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- To use FIFO2 as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) while the I²C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- The FIFO2 state is held even if the FIFO2 operation is disabled.

Bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

<Notes>

- The enable or disable state must be switched only when the IBSR:BB bit is "0" or when the IBCR:INT bit is "1".
- If receive FIFO is selected and the reserved address is detected, and if you wish to select the slave mode transmission, set this bit to "0" and set IBCR:ACKE bit to "0" with an interrupt of reserved address detection.
- If receive FIFO is selected and if the SSR:RDRF bit of SSR is "1" when this bit is changed from "1" to "0", receive FIFO is not disabled until the bit is set to "0".
- If transmit FIFO is selected, FIFO2 contains data, and you wish to change this bit from "0" to "1", set the SMR:TIE bit to "0" first. Then, set this bit to "1", and set the SMR:TIE bit to "1".

[bit 0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If receive FIFO is selected by the FCR1:FSEL bit and if a receive error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the receive error is cleared.
- To use FIFO1 as transmit FIFO, this bit must be set to "1" or "0" when the transmit data is empty (SSR:TDRE=1).
- To use FIFO1 as receive FIFO, this bit must be set to "0" when the receive buffer is empty (SSR:RDRF=0) and receive FIFO contains no valid data (FBYTE2=0) while the I²C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- To use FIFO1 as receive FIFO, this bit must be set to "1" when the receive buffer is empty (SSR:RDRF=0) while the I²C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- The FIFO1 state is held even if the FIFO1 operation is disabled.

Bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

<Notes>

- The enable or disable state must be switched only when the IBSR:BB bit is "0" or when the IBCR:INT bit is "1".
- If receive FIFO is selected and the reserved address is detected, and if you wish to select the slave mode transmission, set this bit to "0" and set IBCR:ACKE bit to "0" with an interrupt of reserved address detection.
- If receive FIFO is selected and the SSR:RDRF bit is "1" when this bit is changed from "1" to "0", receive FIFO is not disabled until the bit is set to "0".
- If transmit FIFO is selected, FIFO1 contains data, and if you wish to change this bit from "0" to "1" state, set the SMR:TIE bit to "0" first. Then, set this bit to "1", and set the SMR:TIE bit to "1".

5.11. FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a receive interrupt when certain number of data sets are received in the receive FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count in the FIFO buffer. The following table shows the relation between the FCR1:FSEL bit state to FBYTE.

Table 5-3 Display of data count

FSEL	FIFO selection	Data count display
0	FIFO2:Receive FIFO, FIFO1: Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2:Transmit FIFO, FIFO1:Receive FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to flag a receive interrupt for the FBYTE register of receive FIFO. If this transfer data count matches the FBYTE register display, the interrupt flag (SSR:RDRF) is set to "1".
- If both of the following conditions are satisfied and if the receive idle state continues for more than 8 baud rate clocks, the interrupt flag (SSR:RDRF) is set to "1".
 - The receive FIFO idle detection enable bit (FCR:FRIIE) is "1".
 - The number of data sets stored in the receive FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted. If receive FIFO is disabled, this counter is reset to zero (0). If data remains in the receive FIFO and if receive FIFO is enabled, the data counting is restarted.
- To receive data in the master mode operation (master mode reception), set the SMR:TIE bit to "0", set the receive data count for the FBYTE register of transmit FIFO, and set the FCR1:FDRQ bit to "0". The SCL clocks are output for the specified data count, and then IBCR:INT bit is set to "1". The SMR:TIE bit must be set to "1" only after the FCR1:FDRQ bit is set to "1".

FBYTE2, FBYTE1: FIFO2 data count display bit, FIFO1 data count display bit

During writing	Sets the transfer data count.
During reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in the FIFO buffer but not transmitted yet
 During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
 During reception: Set the data count to generate a receive interrupt.

<Notes>

- The FBYTE value of transmit FIFO must be "8'h00" except when data is received in the master mode operation.
- During the master mode data reception, the transmit data count must be set only when transmit FIFO is empty and the SMR:TIE bit is "0".
- When data is being received in the master mode operation, the I²C interface operation can be disabled (ISMK:EN=0) only after transmit/receive FIFO has been disabled.
- The FBYTE bit of receive FIFO must be set to "1" or larger.
- Change this register under one of the following conditions:
 - When the I²C interface operation is disabled (ISMK:EN=0)
 - When IBCR:INT=1 in case of SSR:DMA=0 and master mode reception
 - When SSR:TBI=1 in case of SSR:DMA=1 and master mode reception
- A read-modify-write instruction cannot be used for this register.
- Any setting exceeding the FIFO capacity is inhibited.
- To receive data in the master mode operation (master mode reception), do not write dummy data to the Transmit Data Register (TDR) when setting the SMR:TIE bit to "0" and setting the receive data count for the FBYTE register of transmit FIFO.
- When all the following requirements are met, the receive data full flag bit (SSR:RDRF) is not set to "1" even though the effective data of FBYTE setting number exist in the receive FIFO. If the FBYTE register is set to "2" or greater, this operation will not occur.
 - FBYTE is set to "1".
 - The effective data count is "1", same as the number specified in FBYTE register.
 - When the multi function serial interface macro receives the data, and writes received data in the reception FIFO, the data of the reception FIFO are read at the same time.

However, after that, the receive data full flag bit (SSR:RDRF) will be set to "1" at any of the following conditions.

- The next data is received.
- The receive idle state of 8 bits or longer is detected when the receive FIFO idle is enabled (FCR:FRIIE=1)

Chapter: USB Clock Generation

This chapter explains USB clock generation.

1. Overview
2. Configuration and Block Diagram
3. Explanation of Operation
4. Setup Procedure Example
5. Register List
6. Usage Precautions

1. Overview

This section provides an overview of the USB operating clock.

The USB clock runs at 48 MHz and is used by USB macro for communication.

An external 48 MHz clock can be used for the USB clock, or a 48 MHz clock can be generated with USB PLL.

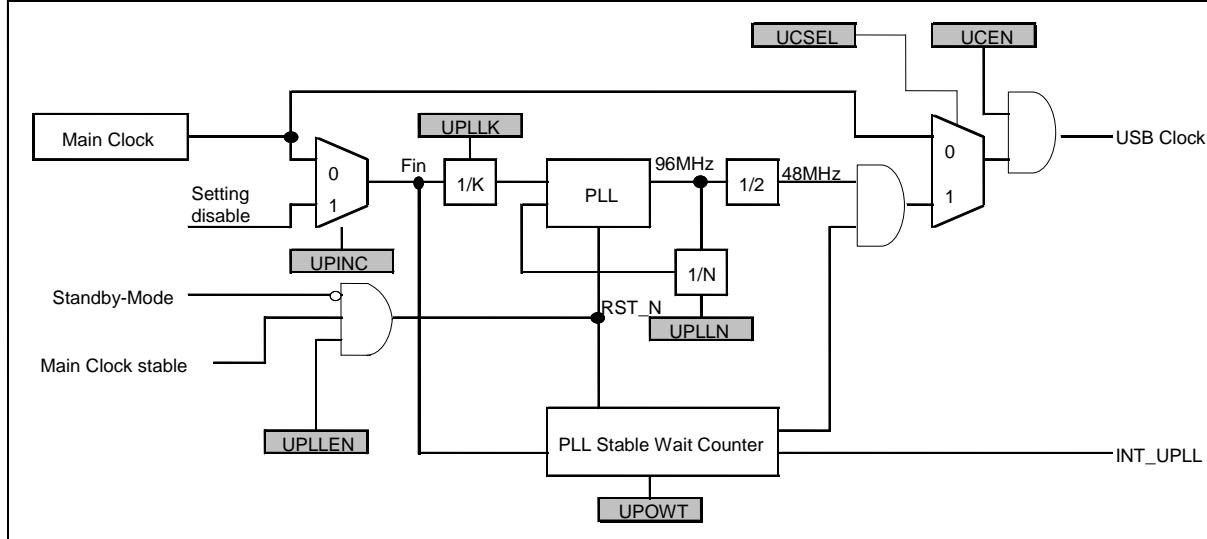
The USB clock generation unit is responsible for the following functions:

- Enables or stops output of the USB clock.
- Selects the USB clock.
- Enables or stops oscillation of PLL macro.
- Selects the input clock of PLL macro.
- Sets the input clock frequency division of PLL macro.
- Sets the input clock multiplication of PLL macro.
- Sets the stabilization wait time of PLL macro.
- Stops the USB clock in standby mode.

2. Configuration and Block Diagram

This section explains the configuration and block diagram of the USB operating clock generation unit.

Figure 2-1 Block diagram of USB operating clock generation unit



■ Operating Clock Control Register (UPLEN)

- The control register can enable oscillation.

■ Input Clock Select Register (UPINC)

- Be sure to select the main clock.

■ USB-PLL

- Frequency division setting (UPLLK, UPLLN)
 - To generate a 48 MHz clock used for USB operation, the PLL clock output must be set to 96 MHz. By configuring the K frequency division and N frequency division, the USB-PLL clock must be set to 48 MHz.
- Oscillation stabilization wait time setting (UPOWT)
 - Oscillation stabilization wait time for USB-PLL can be specified.

■ Output clock

- Output Clock Select Register (UCSEL)
 - Can be selected from the main clock and USB-PLL clock.
- PLL Clock Output Enable Register (UCEN)
 - Can enable the output of the USB-PLL clock.

■ Standby mode setting

- The Standby-Mode signal shown in Figure 2-1 turns to be active in the following modes.
 - The USB clock stops in the following standby modes:
 - Stop mode
 - TIMER mode:
- The Main Clock stable signal shown in Figure 2-1 is an oscillation stabilization signal for each mode.

3. Explanation of Operation

This section explains the operation of the USB operating clock generation unit.

■ Selecting the USB operating clock

The following two types of clocks can be selected for the USB operating clock.

● Selecting the main clock

Select the main clock to use the main oscillation clock (CLKMO) directly as the USB clock. In this case, CLKMO must be input externally at 48 MHz, or must oscillate at 48 MHz. Enable the output of the USB clock after confirming stabilization of the CLKMO oscillation.

● Selecting the USB-PLL clock

Select the USB-PLL clock to use it as the USB clock. The PLL VCO clock output must be set to 96 MHz and divided into two to generate a 48 MHz clock used for USB operation. The following Table 3-1 shows the frequency division ratio settings when the PLL clock output is set to 96 MHz.

Table 3-1 Example PLL frequency division ratio settings

Fin(MHz)	K	N
4	1	24
8	1	12
8	2	24
16	1	6
16	2	12
16	4	24
24	1	4
24	2	8
24	4	16
24	6	24

■ Changing to standby mode

● When changing to standby mode

Before changing to standby mode (Stop mode or TIMER mode), set UCCR.UCEN to "0" to stop the USB clock supply.

1. Set UCCR.UCEN to "0".
2. Read the UCCR Register to check that UCEN is set to "0".
3. Changing to standby mode.

When returning from standby mode, set UCEN to "1". The supply starts when the USB clock oscillation has been stabilized. Take either of the following actions to confirm whether or not the USB clock oscillation has been stabilized.

- a) When PLL macro is used

Check that UPRDY is "1", or use the PLL macro oscillation stable wait interrupt.

- b) When main clock 48 MHz is used

After the main clock oscillation has been stabilized, supply the USB clock.

■ USB-PLL macro oscillation stabilization wait settings

● Oscillation stabilization wait time for USB-PLL can be specified

After the main clock oscillation has been stabilized, the oscillation stabilization wait time for USB-PLL begins to be counted.

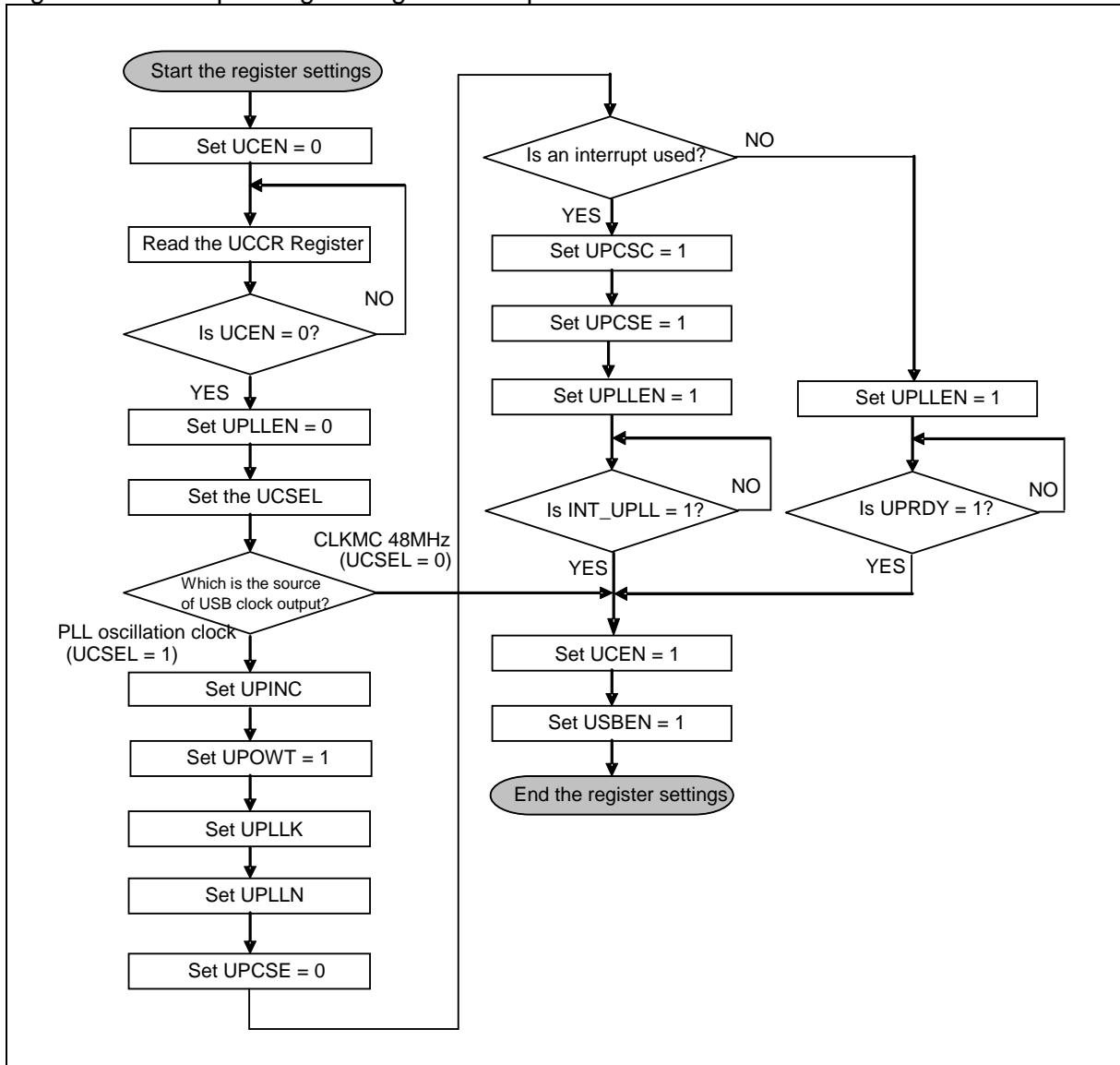
Before enabling the USB-PLL oscillation, configure the oscillation stabilization wait time for USB-PLL and the oscillation stable complete interrupt. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

4. Setup Procedure Example

This section explains an example of setting up the USB operating clock generation unit.

Figure 4-1 shows an example of setting up the USB operating clock.

Figure 4-1 USB operating clock generation procedure



5. Register List

This section explains the register list of the USB operating clock generation unit.

■ The register list of the USB operating clock generation unit.

Abbreviation	Register name	See
UCCR	USB Clock Control Register	5.1
UPCR1	USB-PLL Control Register-1	5.2
UPCR2	USB-PLL Control Register-2	5.3
UPCR3	USB-PLL Control Register-3	5.4
UPCR4	USB-PLL Control Register-4	5.5
UP_STR	USB-PLL Macro Status Register	5.6
UPINT_ENR	USB-PLL Interrupt Enable Register	5.7
UPINT_CLR	USB-PLL Interrupt Clear Register	5.9
UPINT_STR	USB-PLL Interrupt Status Register	5.8
USBEN	USB Enable Request Register	5.10

5.1. USB Clock Setup Register (UCCR)

The UCCR selects the USB clock and enables/disables the USB clock output.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved			UCSEL	UCEN
Initial value				-			1'b0	1'b0
Attribute				-			R/W	R/W

■ Register functions

[bit 7:2] res: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1] UCSEL: USB clock select bit

Bit	Description
0	Main clock [Initial value]
1	PLL macro oscillation clock

[bit 0] UCEN: USB clock output enable bit

Bit	Description
0	Disables the USB clock output [Initial value]
1	Enables the USB clock output

<Notes>

- When selecting the main clock with UCSEL, the 48 MHz frequency must be input from an external main oscillation.
- This register is not initialized by software reset.

5.2. USB-PLL Control Register-1 (UPCR1)

The UPCR1 sets PLL for USB.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved			UPINC	UPLLEN
Attribute				-			R/W	R/W
Initial value				-			1'b0	1'b0

■ Register functions

[bit 7:2] res: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 1] UPINC: PLL macro input clock select bit.

Bit	Description
0	Main clock input [Initial value]
1	Setting disabled

[bit 0] UPLLEN: USB-PLL macro oscillation enable bit

Bit	Description
0	Stops USB-PLL macro [Initial value]
1	Enables the USB-PLL macro oscillation

<Notes>

- Be sure to set UPINC to "0". Operation is not guaranteed when UPINC is set to "1".
- This register is not initialized by software reset.

5.3. USB-PLL Control Register-2 (UPCR2)

The UPCR2 sets the oscillation stability wait time of PLL macro for USB.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field			Reserved				UPOWT	
Attribute			-				R/W	
Initial value			-				3'b000	

■ Register functions

[bit 7:3] res: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit 2:0] UPOWT: USB-PLL macro oscillation stabilization wait setting bit

Bit 2	Bit 1	Bit 0	Description
0	0	0	$2^9/\text{Fin}$: Approx. 128 μs * [Initial value]
0	0	1	$2^{10}/\text{Fin}$: Approx. 256 μs *
0	1	0	$2^{11}/\text{Fin}$: Approx. 512 μs *
0	1	1	$2^{12}/\text{Fin}$: Approx. 1.02 ms *
1	0	0	$2^{13}/\text{Fin}$: Approx. 2.05 ms *
1	0	1	$2^{14}/\text{Fin}$: Approx. 4.10 ms *
1	1	0	$2^{15}/\text{Fin}$: Approx. 8.20 ms *
1	1	1	$2^{16}/\text{Fin}$: Approx. 16.4 ms *

* : When Fin = 4 MHz

<Notes>

- F_{in} is the clock (main oscillation) selected by UPINC.
- This register is not initialized by software reset.

5.4. USB-PLL Control Register 3 (UPCR3)

The UPCR3 sets the frequency division ratio (K) of PLL macro for USB.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field		Reserved				UPLLK		
Attribute		-				R/W		
Initial value		-				5'b00000		

■ Register functions

[bit 7:5] res: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit 4:0] UPLLK: Frequency division ratio (K) setting bit of the USB-PLL clock

Bit 4:0	Description
00000	
00001	
•	Divides the frequency by (UPLLK+1) (Example) UPLLK = "00000" => 1/1 frequency [Initial value]
•	
11111	

<Note>

This register is not initialized by software reset.

5.5. USB-PLL Control Register 4 (UPCR4)

The UPCR4 Register sets the frequency division ratio (N) of PLL macro for USB.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field		Reserved				UPLLN		
Attribute		-				R/W		
Initial value		-				5'b10111		

■ Register functions

[bit 7:5] res: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit 4:0] UPLLN: Frequency division ratio (N) setting bit of the USB-PLL clock

Bit 4:0	Description
00000	This setting is disabled.
00001	
•	Divides the frequency by (UPLLN+1) (Example) UPLLN = "10111" => 1/24 frequency [Initial value]
•	
11111	

<Note>

This register is not initialized by software reset.

5.6. USB-PLL Macro Status Register (UP_STR)

The UP_STR indicates the macro status of USB-PLL.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved				UPRDY
Attribute				-				R
Initial value				-				1'b0

■ Register functions

[bit 7:1] res: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit 0] UPRDY: USB-PLL macro oscillation stable bit

Bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stabilized state

<Note>

This register is not initialized by software reset.

5.7. USB-PLL Interrupt Enable Register (UPINT_ENR)

The UPINT_ENR enables/disables the USB-PLL oscillation stabilization complete interrupt.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved				UPCSE
Attribute				-				R/W
Initial value				-				1'b0

■ Register functions

[bit 7:1] res: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit 0] UPCSE: USB-PLL macro oscillation stabilization complete interrupt enable bit

Bit	Description
0	Disables the interrupt [Initial value]
1	Enables the interrupt

5.8. USB-PLL Interrupt Status Register (UPINT_STR)

The UPINT_STR indicates the status of USB-PLL oscillation stabilization wait interrupts.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved				UPCSI
Attribute				-				R
Initial value				-				1'b0

■ Register functions

[bit 7:1] res: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit 0] UPCSI: USB-PLL interrupt cause status bit

Bit	Description
0	No interrupt has occurred [Initial value]
1	An interrupt has occurred

5.9. USB-PLL Interrupt Clear Register (UPINT_CLR)

The UPINT_CLR is used to clear the USB-PLL interrupt cause.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field				Reserved				UPCSC
Attribute				-				W
Initial value				-				1'b0

■ Register functions

[bit 7:1] res: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit 0] UPCSC: USB-PLL macro oscillation stabilization interrupt cause clear bit

Bit	Description
0	Disabled [Initial value]
1	Clears the PLL macro oscillation stabilization wait interrupt.

<Note>

Writing to this register and clearing the interrupt clears the UPINT_STR Register.

5.10. USB Enable Request Register (USBEN)

The USBEN enables/disables USB controller operation.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				-	-	-	USBEN
Attribute					-	-	-	R/W
Initial value					-	-	-	1'b0

■ Register functions

[bit 7:1] res: Reserved bits

"0b0000010" is read from these bits.

Set these bits to "0b0000010" when writing.

[bit 0] USBEN: USB enable bit

Bit	Description
0	Disables USB operation (Resets the USB controller) [Initial value]
1	Enables the USB operation

<Notes>

- When using USB, set this bit to "1" previously.
- Supply at least five cycles of USB clocks to the USB controller before setting this bit to "1".

6. Usage Precautions

This section explains the precautions for using clock generation unit.

- Setting clock output and stopping PLL macro

Do not disable the USB clock output (UCEN = 0) and select the USB clock (UCSEL) at the same time. Be sure to disable the USB clock output before selecting the USB clock.
- Setting the frequency division ratio of PLL oscillation

When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then enable the PLL oscillation again.
- Selecting the main clock

By writing "0" to the UCSEL bit, the main clock is selected as the USB clock. The main clock should be selected when the main clock oscillates at 48 MHz.
- Setting the PLL oscillation stabilization wait time

Set the oscillation stabilization wait time with the PLL Oscillation Stabilization Wait Time Setup Register, and then enable PLL. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.
- Selecting the PLL macro input clock

By writing "1" to the UCSEL bit, the PLL macro oscillation clock is selected as the USB clock. Write "0" to the UPINC bit of the USB-PLL Control Register 1 (UPCR1), and be sure to select the main clock as the PLL macro input clock.

The following Table 6-1 shows relationship between the USB clock and UPDS/UPLLEN/UPINC.

Table 6-1 USB clock and register settings

		UCSEL	UPLLEN	UPINC
When using the 48 MHz main clock		0	0	-
When using the PLL macro oscillation clock	Main clock oscillation input	1	1	0
	Setting disabled	1	1	1

- Standby mode and the PLL oscillation stabilization wait counter

If the mode changes to TIMER/STOP mode while waiting for the PLL oscillation to stabilize, PLL stops and the stabilization wait counter is cleared.
- Setting the USB enable bit and USB controller

To use the USB controller, enable the USB enable bit. Supply the USB operating clock to the USB controller before enabling the USB enable bit. For details on USB controller settings, see Chapters "USB Function" and "USB Host".

Chapter: USB Function

This chapter explains the USB function.

1. Overview of USB Function
2. Configuration of USB Function
3. Operations of USB Function
4. Examples of USB Function Setting Procedures
5. USB Function Registers

1. Overview of USB Function

The USB function is an interface supporting the USB (Universal Serial Bus) communication protocol. It supports full-speed transfer mode (12 Mbps), and has the following features.

1.1. Features of USB function

- Full-speed (12 Mbps) transfer supported.
- Auto answered device status.
- Automatic generation and check of bit stripping, bit stuffing, CRC5, and CRC16.
- Toggle check by data synchronization bit.
- Auto-answer to standard all commands other than the Get/SetDescriptor and SynchFrame (these three commands can be processed similarly as class vendor commands).
- The class vendor commands can be received as data and responded by firmware.
- Up to 6 Endpoints supported. (Endpoint 0 is fixed to control transfer)
- Each Endpoint includes 2 buffers for data transfer.
(Endpoint 0 includes each buffer exclusively for IN and OUT directions)
- Automatic data transfer via DMA supported (except Endpoint 0 buffers).

<Note>

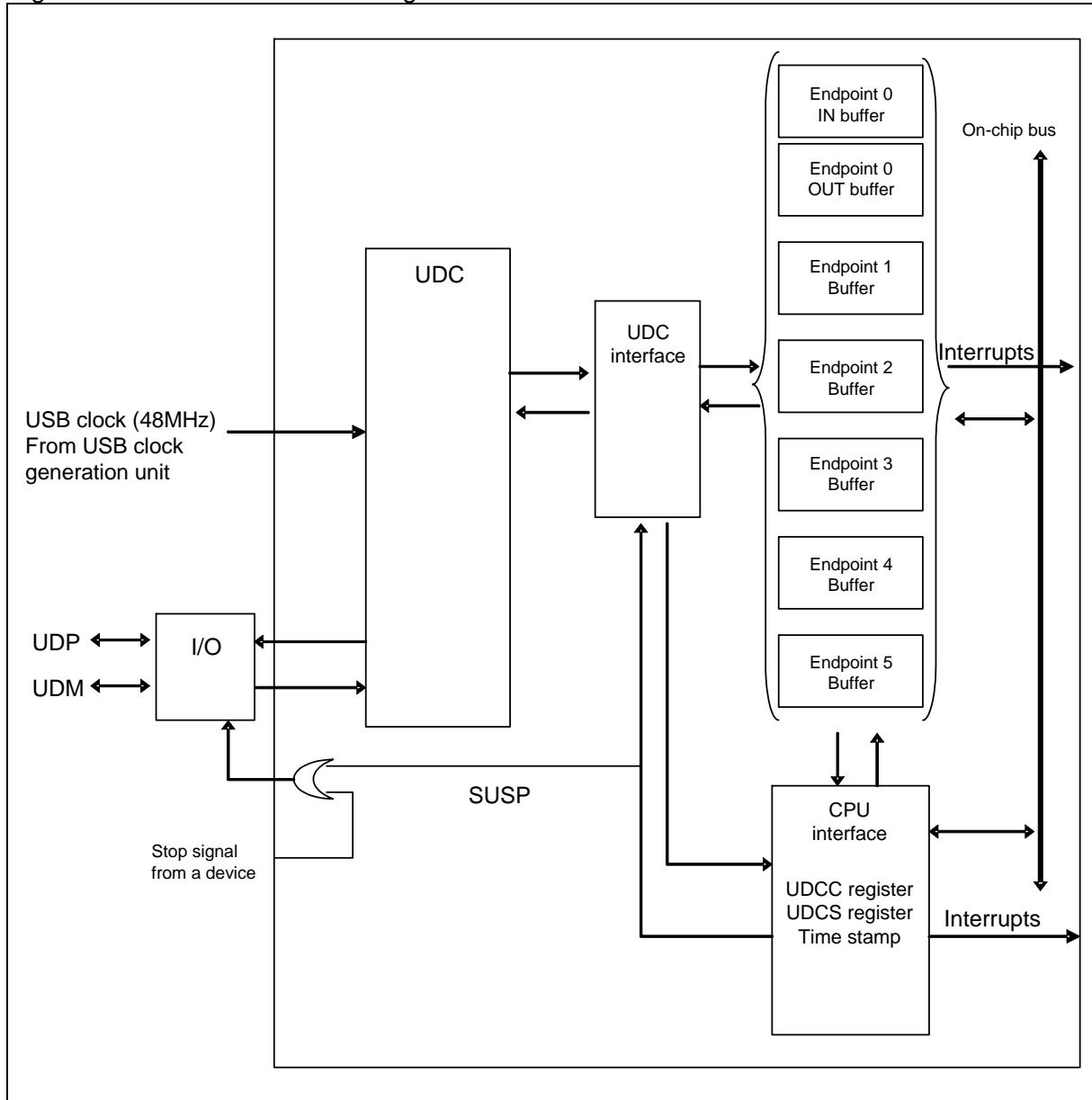
Set the base clock (HCLK) to 13 MHz or higher when using the USB function.

2. Configuration of USB Function

Figure 2-1 shows the block diagram of the USB function

■ USB function block diagram

Figure 2-1 USB function block diagram



3. Operations of USB Function

The USB function supports the USB (Universal Serial Bus) communication protocol. Its hardware supports the basic protocol operation (handshake). Therefore, USB communication can be implemented by processing only transfer data.

- 3.1 USB function operation
- 3.2 Detection of connection and disconnection
- 3.3 Operation of each register in response to a command
- 3.4 Suspend function
- 3.5 Wake-up function
- 3.6 DMA transfer function
- 3.7 NULL transfer function
- 3.8 STALL response/release of endpoint 0
- 3.9 Stall response/release of endpoints 1 to 5

3.1. USB function operation

To use the USB function, take the following steps for setup.

1. Configure the USB clock generation unit while the USB Enable Register (USBEN) disables USB operation (USBEN = 0).
2. Enable the USB clock output.
3. Enable USB operation (USBEN = 1).

The USB function transfers packets bi-directionally to/from a host controller that supports the USB protocol. Connection with the host and devices, and configuration are emulated. Communications are implemented subsequently in different transfer types using device drivers.

The following explains the operation of USB communication between the host and devices by taking an enumeration for example.

Movements of registers and USB packets are shown here to provide details of the entire process.

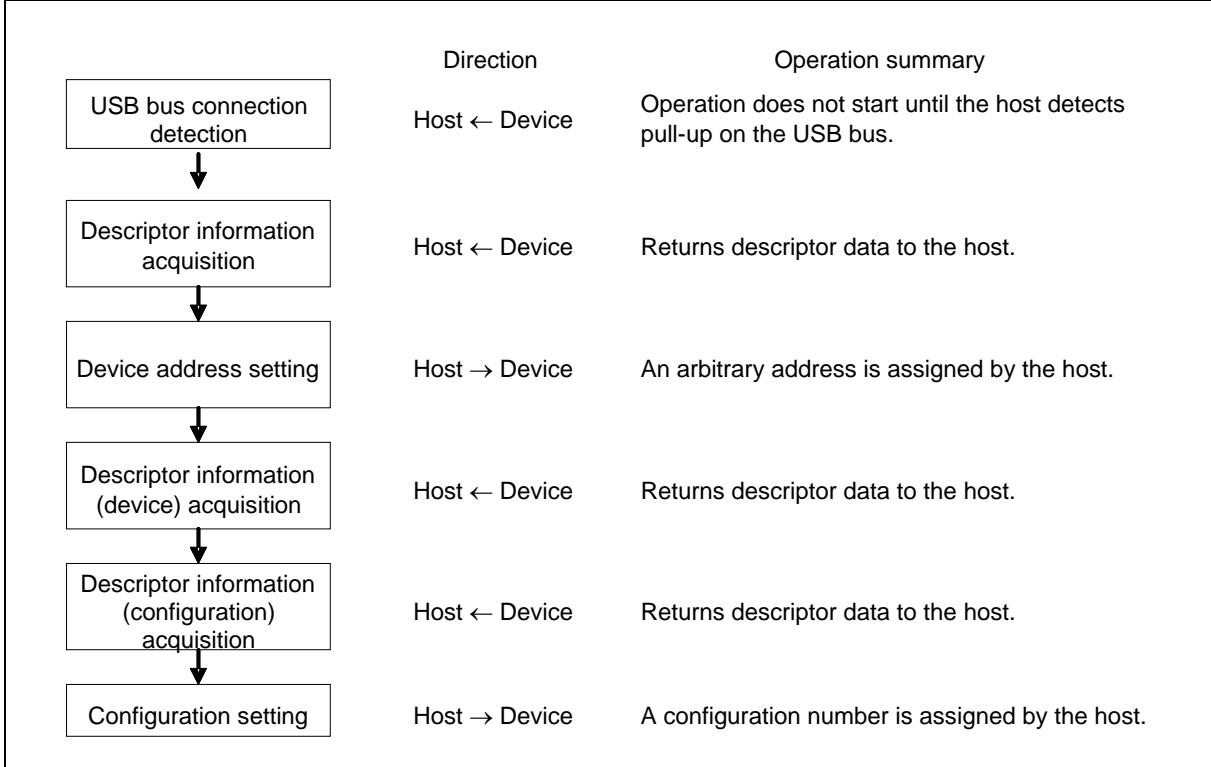
■ Emulation processing

Emulation is the first process for USB to work, establishes connection between the host and devices. The host investigates what types of devices are connected on the USB bus by using USB control transfer (a USB transfer type). (Defined in the USB specification) This process uses EP0 (Endpoint 0) from the six Endpoints (as defined in the USB specification).

To use EP1 to EP5, reception and processing on the USB bus are required in the following order:

1. Resetting the USB bus
2. Setting the address by SET_Address
3. Setting configuration by SET_Config

Figure 3-1 Example USB cable pin connection



- USB bus connection detection

The connection is reported from a device to the host.

The host monitors two signal lines (D+ and D-) on the USB bus, and found the connection of a device if either of the signals turns to HIGH level.

For a detailed procedure explaining how to use the device in self-powered mode, see "3.2 Detection of connection and disconnection". To use the device as bus-powered, follow the procedure given in "Register Initial Setting and Operation Start Procedures".

● Register initial setting and operation start procedures

The following shows an example initial setting procedure of USB function registers.

- Set EP0 configuration (such as packet size) by the EP0C register.
- Set EPEN, DIR, or TYPE of each Endpoint by the EP1C to 5C registers.
- Clear the RST bit in the UDCC register.
- Clear BFINI in the EP0IS, EP0OS, and EP1S to EP5S registers.
- Clear the HCONX bit in the UDCC register.

● USB bus reset

The USB device core is initialized when the host executes a bus reset on the device, but register and buffer states are not initialized.

Take the following steps to process the device. (The process is not required in the initial bus reset after USB connection.)

- Initialize the buffer by the BFINI bit in the EP0I Status Register (EP0IS), the BFINI bit in the EP0O Status Register (EP0OS), and the BFINI bit in the EP1 to EP5 Status Registers (EP1S to EP5S).
- Return firmware control to the state before the emulation.

● **Descriptor acquisition**

When the host requests a device, the device reports data to the host in reply to the request.

The communication is broken up into the following three stages.

Figure 3-2 Communication stages



The setup stage checks whether the device has received the packets from the host successfully. The descriptor information to be returned in the next stage is prepared in the send buffer in this stage. The data stage checks whether the host has sent data successfully. In the status stage, the host sends a packet without data to end the transfer.

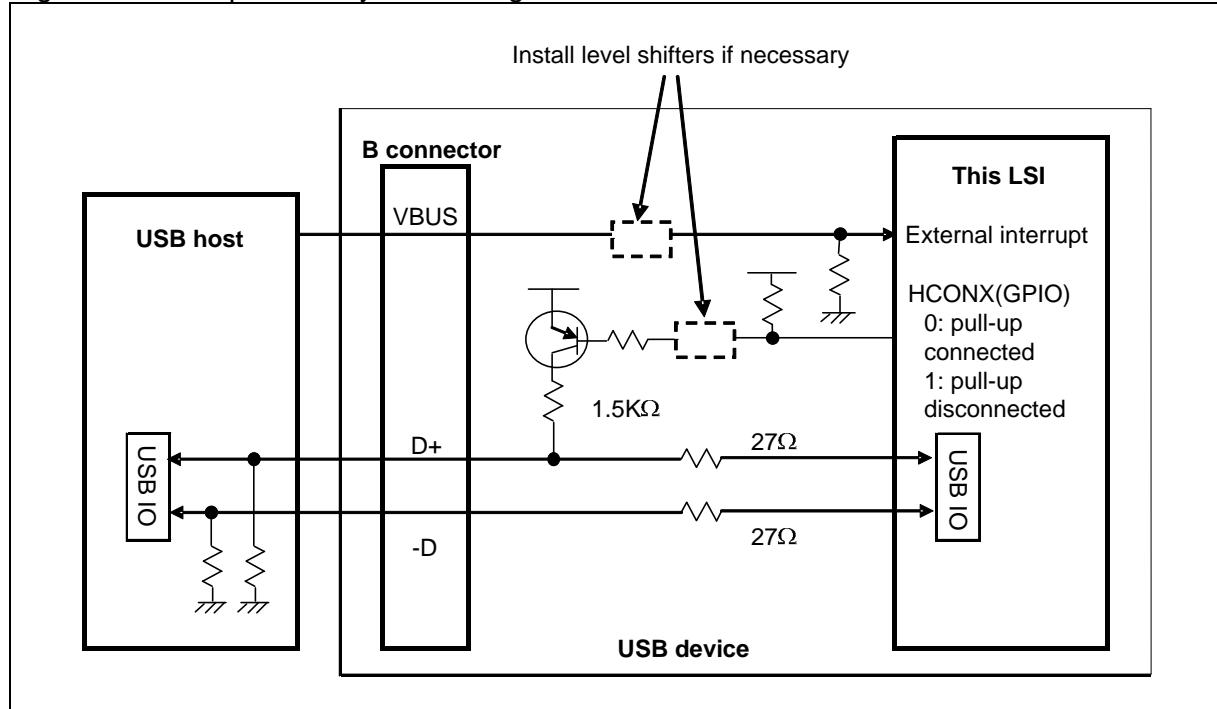
3.2. Detection of connection and disconnection

The following explains about detecting connection and disconnection to/from the USB host.

■ Example of USB system connection

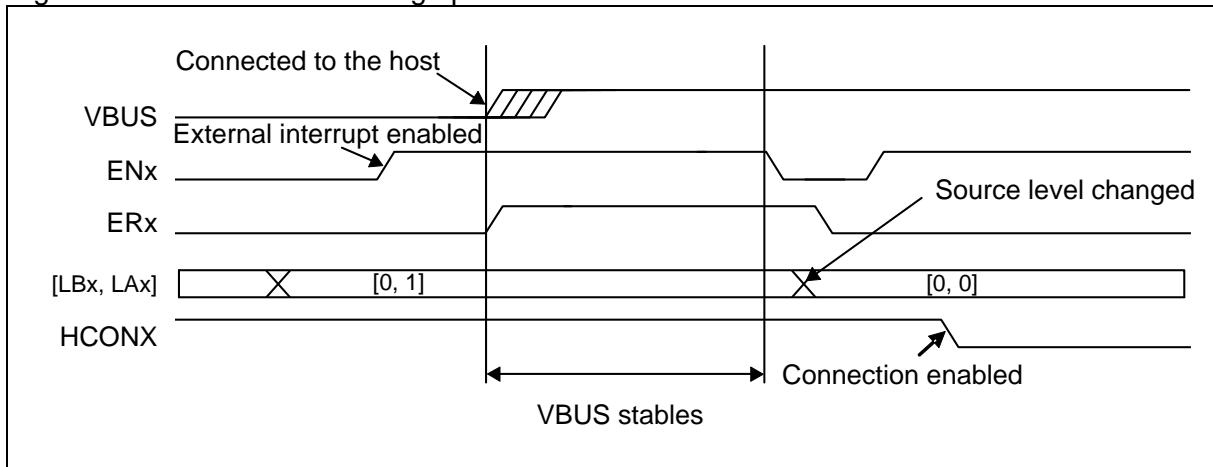
By connecting an external interrupt pin to the VBUS pin of the USB connector, and installing a pull-down resistor onto the VBSU signal, disconnection from the USB host can be detected. Figure 3-3 shows an example connection of USB connector with D+, D- and VBUS.

Figure 3-3 Example USB system configuration



● Connection detection

Figure 3-4 Connection detecting operation



A device finds and processes the connection with the host in the following sequence:

1. The HCONX bit in the UDCC register must be set to "1". (When controlling a pull-up resistor on a general-purpose port, set the port to the pull-up resistor disconnection.)
2. Set the source level of external interrupts connected with VBUS to HIGH level detection to enable interrupts.
3. Find the USB host connection by the detection of HIGH level of the external interrupt pin, and waits for the period the VBUS becomes stable.
4. Disable external interrupts once. Change the external interrupt cause level to LOW to clear the interrupt cause, and enable external interrupts again.
5. Configure the initial settings (Initialize all components including the USB function registers.) See "Register Initial Setting and Operation Start Procedures" in this section.
6. Connect the pull-up resistor to D+ by clearing the HCONX bit in the UDCC register. *1 *2

*1 : When control the pull-up resistor on a general-purpose port, clear the HCONX bit in the UDCC register, and set the pull-up resistor control general-purpose port to the pull-up resistor connection.

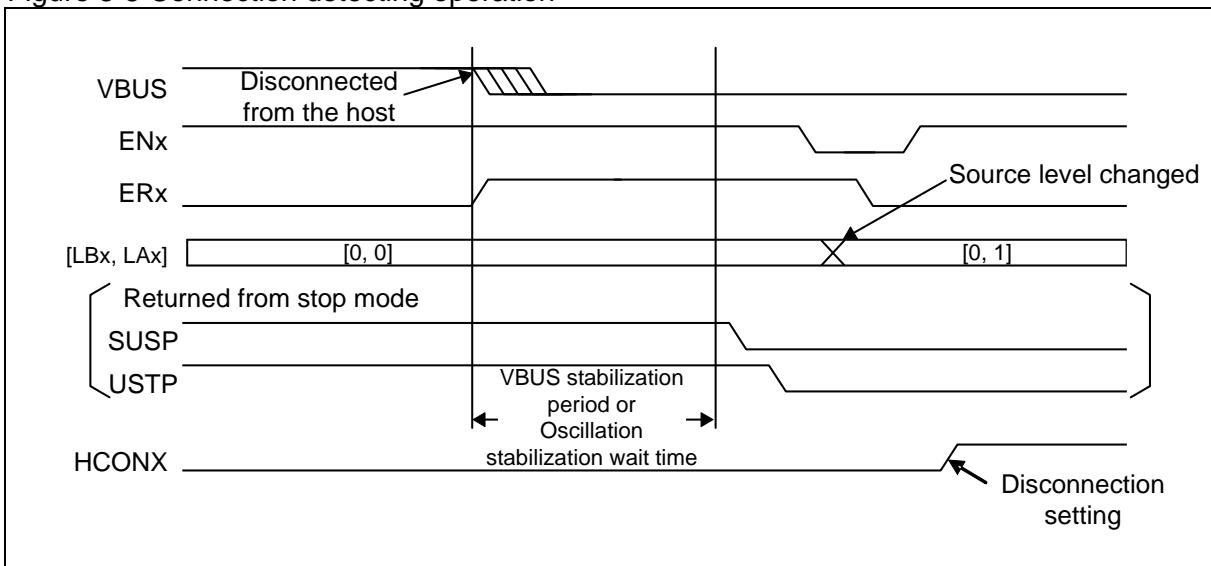
*2 : Clear the HCONX bit even if the pull-up resistor is not controlled.

<Note>

If an external filter is installed on the external interrupt pin, the above VBUS stabilization period does not need to be set by the program.

● Disconnection detection

Figure 3-5 Connection detecting operation



A device finds and processes the disconnection from the host in the following sequence:

- Find the disconnection of the USB host by detecting LOW level of the external interrupt pin connected to VBUS.
- When returned from stop mode or timer mode
 - After the oscillation stabilization wait time, clear in the order of SUSP in the UDCS register and USTP in the UDCC register.
 - In other than stop mode and timer mode wait for the period the VBUS becomes stable.
- Disable external interrupts once. Change the external interrupt cause level to HIGH to clear the interrupt cause, and enable external interrupts again.
- Disconnect the pull-up resistor from D+ by setting the HCONX bit in the UDCC register. *1*2

*1 : When controlling the pull-up resistor on a general-purpose port, set the HCONX bit in the UDCC register, and set the pull-up resistor control general-purpose port to the pull-up resistor connection.

*2 : Set the HCONX bit even if the pull-up resistor is not controlled.

<Note>

If an external filter is installed on the external interrupt pin, the above VBUS stabilization period does not need to be set by the program.

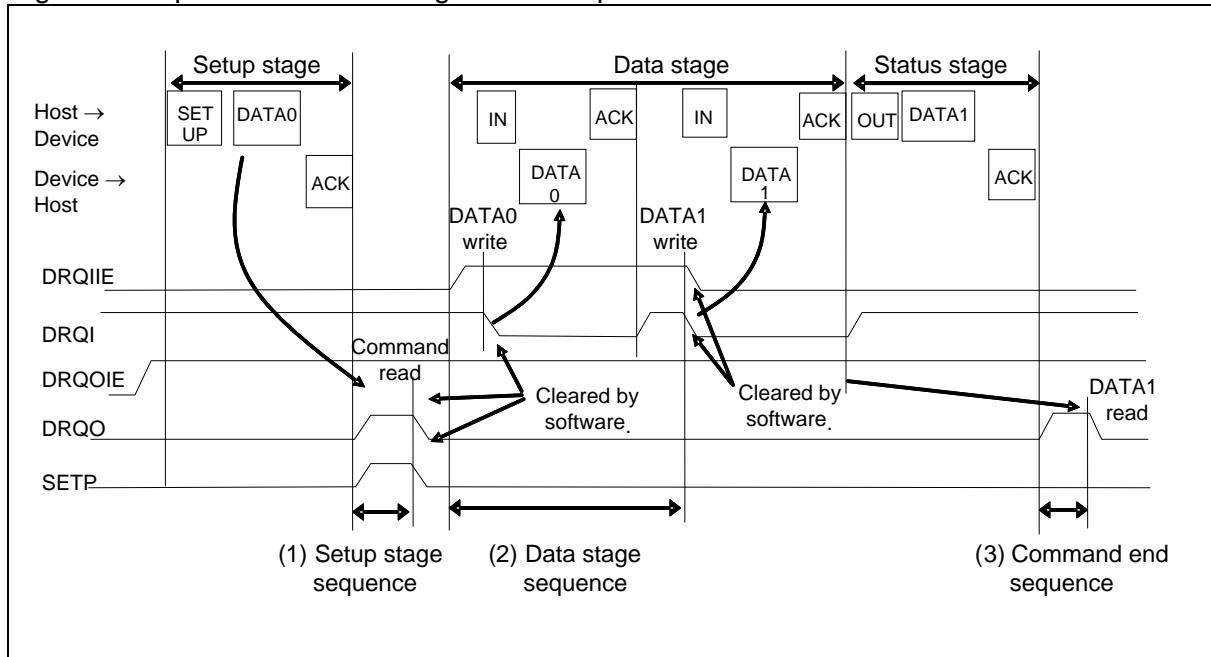
3.3. Operation of each register in response to a command

The following explains the method (architecture) to process USB packets. Responding to CPU interrupts, the firmware sequence is processed for each handshake. This is equivalent to the processing of each packet on the stage basis.

■ Operation of each register in response to a read command

The following explains the case of GetDescriptor, SynchFrame, and class vendor commands.

Figure 3-6 Operation of Each Register in Response to a Read Command



- **Setup sequence**

Upon the receipt of the setup stage, DRQO changes to "1". Immediately when DRQO has changed to "1", enter the CPU interrupt and check the SETP flag. If the flag is "1", read required bits of the command in the receive buffer. (Not necessarily read all the eight bytes.) Subsequently, decode the command, configure required settings, clear the SETP flag and the DRQO interrupt cause, and return.

- **Data stage sequence**

If the command decoding concludes that the data stage is in the IN direction, enable DRQIE,* and transfer outgoing data to the send buffer by the CPU interrupt. When the transfer has finished, clear the DRQI interrupt cause, and return.

*: The DRQI interrupt cause is initially set to "1", and is only used to enable interrupts.

DRQI is set when the data packet to the IN direction has finished. The CPU interrupt is entered immediately when DRQI has been set, and outgoing data is transferred to the send buffer in preparation for the next data packet. When the transfer has finished, interrupt cause DRQI, and return.

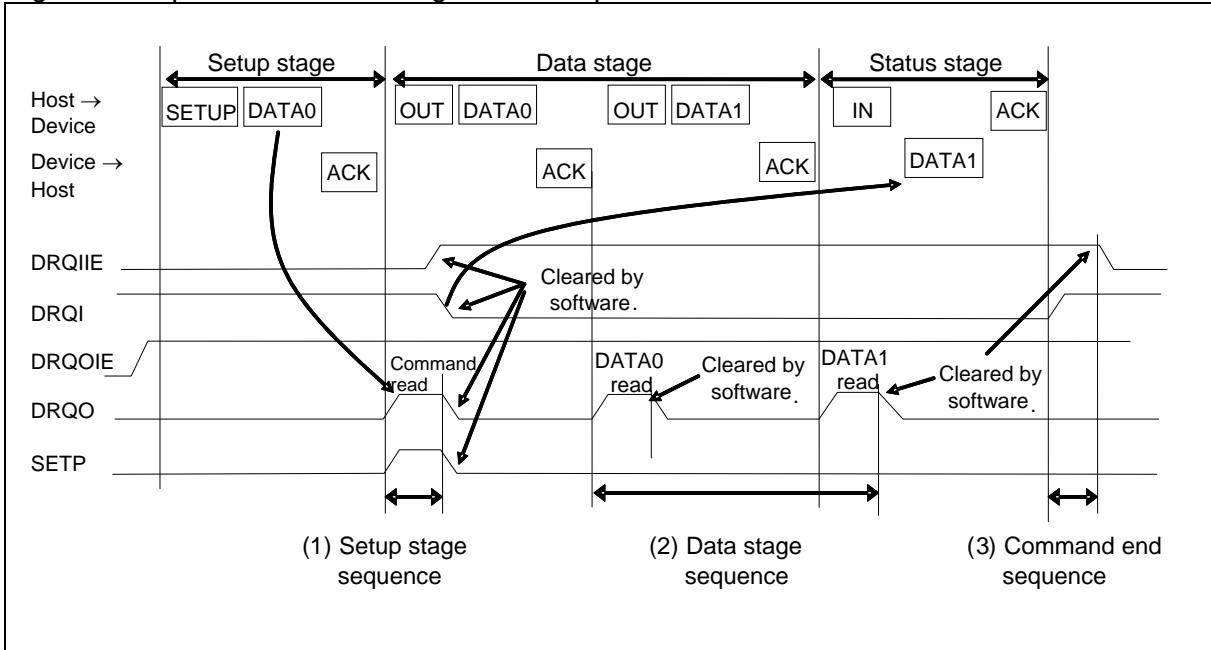
- Command end sequence

DRQO is set when the status stage to OUT direction has finished. Immediately when DRQO is set, enter the CPU interrupt and check that the number of received data units is 0. In preparation for the next setup stage, interrupt cause DRQO, and return.

■ Operation of each register in response to a write command

The following explains the case of SetDescriptor and class vendor commands.

Figure 3-7 Operation of Each Register in Response to a write Command



· **Setup sequence**

Upon the receipt of the setup stage, DRQO changes to "1". Immediately when DRQO has changed to "1", enter the CPU interrupt and check the SETP flag. If the flag is "1", read required bits of the command in the receive buffer. (Not necessarily read all the eight bytes.) Subsequently, decode the command, configure required settings.

In preparation of 0-byte response in the status stage, do not write data to the send buffer, and set DRQI to "0" (as the DRQI interrupt cause is initially set to "1"). Set the DRQIIE to "1" to check a successful completion of the status stage. Clear the SETP flag and the DRQO interrupt cause to return from the interrupt.

· **Data stage sequence**

DRQO is set when the data packed to OUT direction has finished. Immediately when DRQO is set, enter the CPU interrupt and check SIZE in the EP0 Status Register. Use DMA limited to received data, or use CPU read access to read data from the send buffer. Subsequently, clear interrupt cause DRQO to return from the interrupt.

· **Command end sequence**

DRQI is set when the status stage to the IN direction has finished. Immediately when DRQI is set, enter the CPU interrupt and check that the status stage has finished successfully. Subsequently, clear interrupt cause DRQI, and return.

3.4. Suspend function

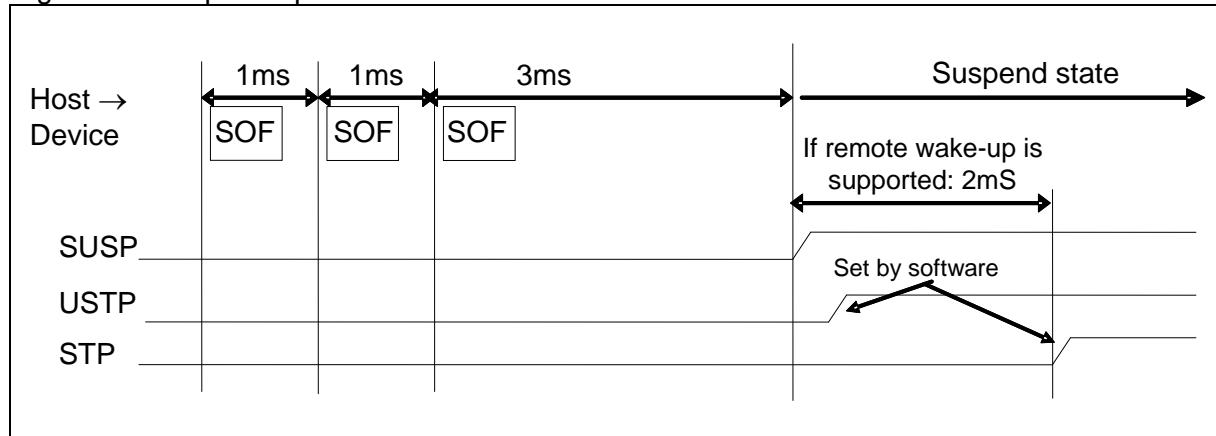
Depending on the bus power configuration, USB devices must drop the power consumption to 500 μ A or less in suspend state. The following explains the sequence a device makes transition to suspend state, and then stop mode or timer mode.

■ Suspend sequence

When the USB device core detects a suspend state, SUSP in the UDCS is enabled.

The following provides an example sequence.

Figure 3-8 Suspend operation



- Suspend sequence

When there is a 3 ms or longer period of inactivity on the USB bus, the USB function detects a suspend state, and sets the SUSP interrupt cause in the UDCS register. For devices supporting remote wake-up, the USB function waits 2 ms more * and sets stop mode or timer mode.

*: This period is required to block remote wake-up.

<Note>

Before stock mode or timer mode is entered, set UDCIE.SUSPIE = 0 and UDCC.USTP = 1 in this order.

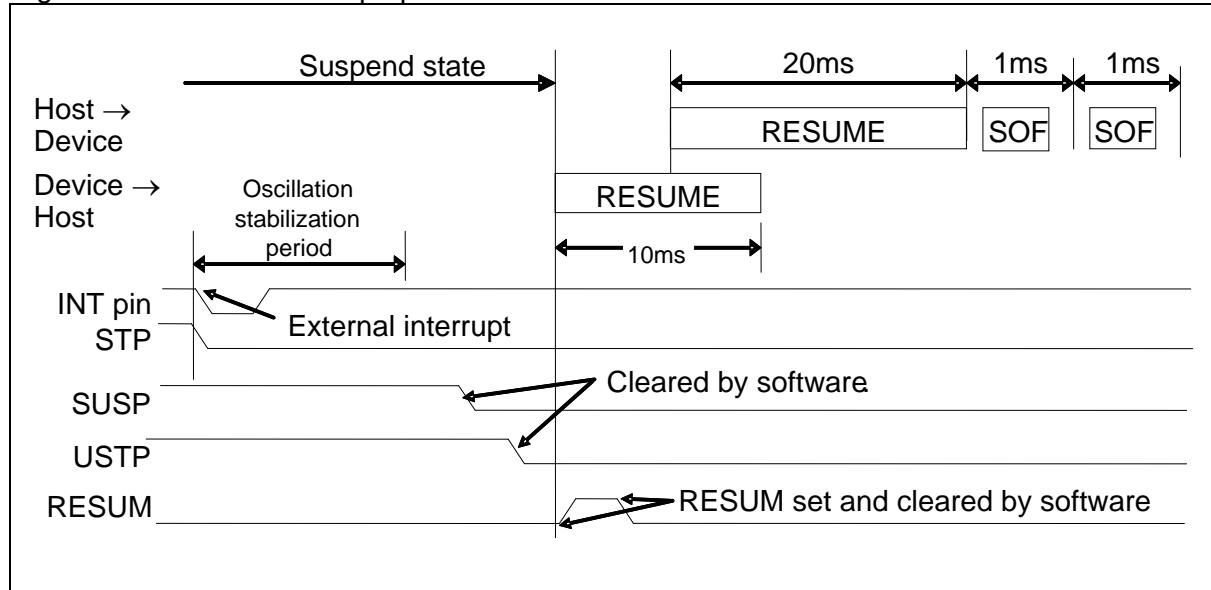
3.5. Wake-up function

To recover a USB device from suspend state to wake-up state, the USB protocol provides two ways.

- Remote wake-up from the device
- Wake-up from the host

■ Remote wake-up

Figure 3-9 Remote wake-up operation

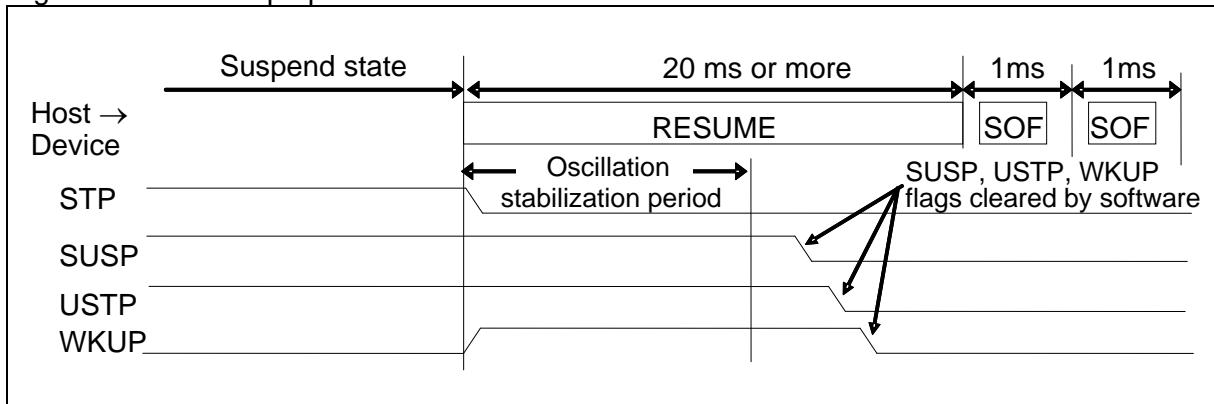


The device must be processed in the following sequence:

1. Recover the device from stop mode or timer mode by an external interrupt.
2. Check that the USB generation clock is stable.
3. Clear the SUSP bit in the UDCC register.
4. Perform a dummy read in the UDCC register.
5. Clear the USTP bit of the UDCC register.
6. Perform a dummy read of the UDCC register.
7. Set the RESUM bit in the UDCC register.
8. Clear the RESUM bit in the UDCC register.

■ Wake-up from the host

Figure 3-10 Wake-up operation from the host



Process the USB device in the following sequence.

1. Set the oscillation stabilization time so that it will not exceed 10 ms.
2. Check that the USB clock is stable.
3. Clear SUSP in the UDCS register, and USTP in the UDCC register in this order.
4. Clear WKUP in the UDCS register.

3.6. DMA transfer function

Data handled by the USB function can be transferred via DMA between the send/receive buffer and internal RAM. The following two modes are available for the DMA transfer.

- Packet transfer mode that transfers each packet according to the configured data size
- Automatic data size transfer mode that transfers the configured data size by a single transfer.

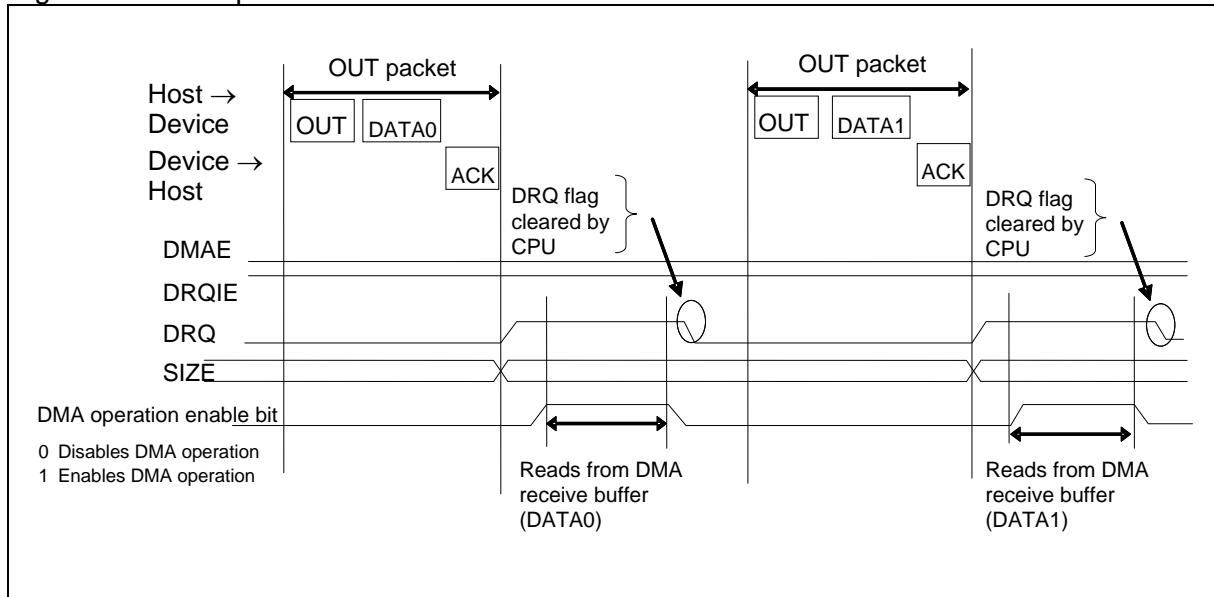
■ Packet transfer mode

The packet transfer mode transfers each packet according to the data size set in DMA and, each time the transfer of a packet finished, clears the interrupt cause for the next packet transfer. This transfer mode can access buffers of Endpoints 1 to 5. Before using DMA, set the interrupt output destination by the DREQ Select Register. (Connect the interrupt output to CPU.NVIC. For details, see Chapter "DMAC".)

The following shows the timing to access buffers in each OUT direction and IN direction.

● Transfer in the OUT direction (Host -> Device)

Figure 3-11 OUT packet transfer

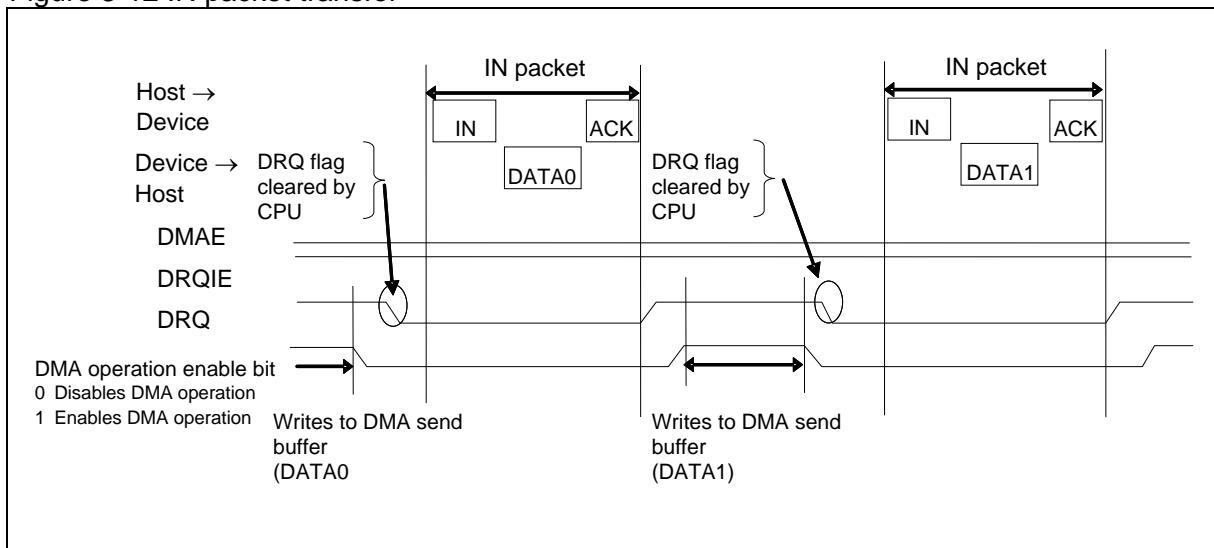


In the OUT direction transfer, the device must be processed in the following sequence:

1. Once the DRQ flag is set and the interrupt handling is entered, check the transfer data size.
2. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the transfer data size, and then enable DMA to start the transfer.
3. After the transfer, clear the pertinent DRQ flag in the EP1S to EP5S registers and the pertinent interrupt source flag in the DMAC status register, and return from the interrupt handling.

● Transfer in the IN direction (Host -> Device)

Figure 3-12 IN packet transfer



In the IN direction transfer, the device must be processed in the following sequence:

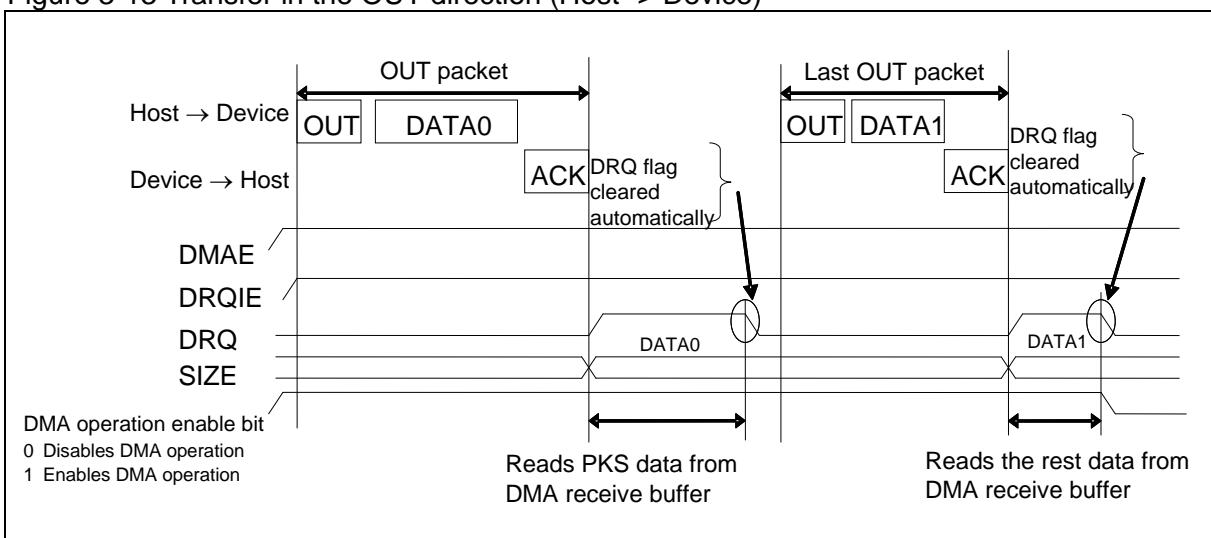
- Once the DRQ flag is set and the interrupt handling is entered, configure the DMA register settings relevant to the number of transfers and block size corresponding to the data size to be transferred in the next IN packet, and then enable DMA to start the transfer.
- After the DMA transfer, clear the pertinent DRQ flag in the EP1S to EP5S registers and the pertinent interrupt cause flag in the DMAC status register, and return from the interrupt handling.

■ Automatic data size transfer mode

This mode can transfer even bytes. To transfer odd bytes, a CPU transfer sequence is required. (See Figure 3-14) Before using DMA, set the interrupt output destination by the DREQ Select Register.(Connect the interrupt output to DMAC.) For details, see Chapter "DMAC".Configure in DMA the total data size to transfer, and also set the transfer enable bit previously.If DRQ is set after transfer from the host while DMAE is enabled, the interrupt cause is automatically cleared when the data size corresponding to PKS in the EP1 to EP5 Control Registers (EPxC) has been transferred. Afterward, the same sequence is repeated after transfer from the host until the transfer data size configured previously in DMA is reached.Meanwhile, configuration by the CPU is not required at all. Thus this mode can transfer data automatically by a single setting. The CPU interrupt is entered after the transfer of the last data. To perform the next transfer, therefore, reconfigure DMAC then to enable DMA and return from the interrupt. The automatic data size transfer mode uses DMAE as "1", buffer access to Endpoints 1 to 5 is only enabled. The following shows the timing to access the buffer in each of the OUT and IN directions.

● Transfer in the OUT direction (Host -> Device)

Figure 3-13 Transfer in the OUT direction (Host -> Device)



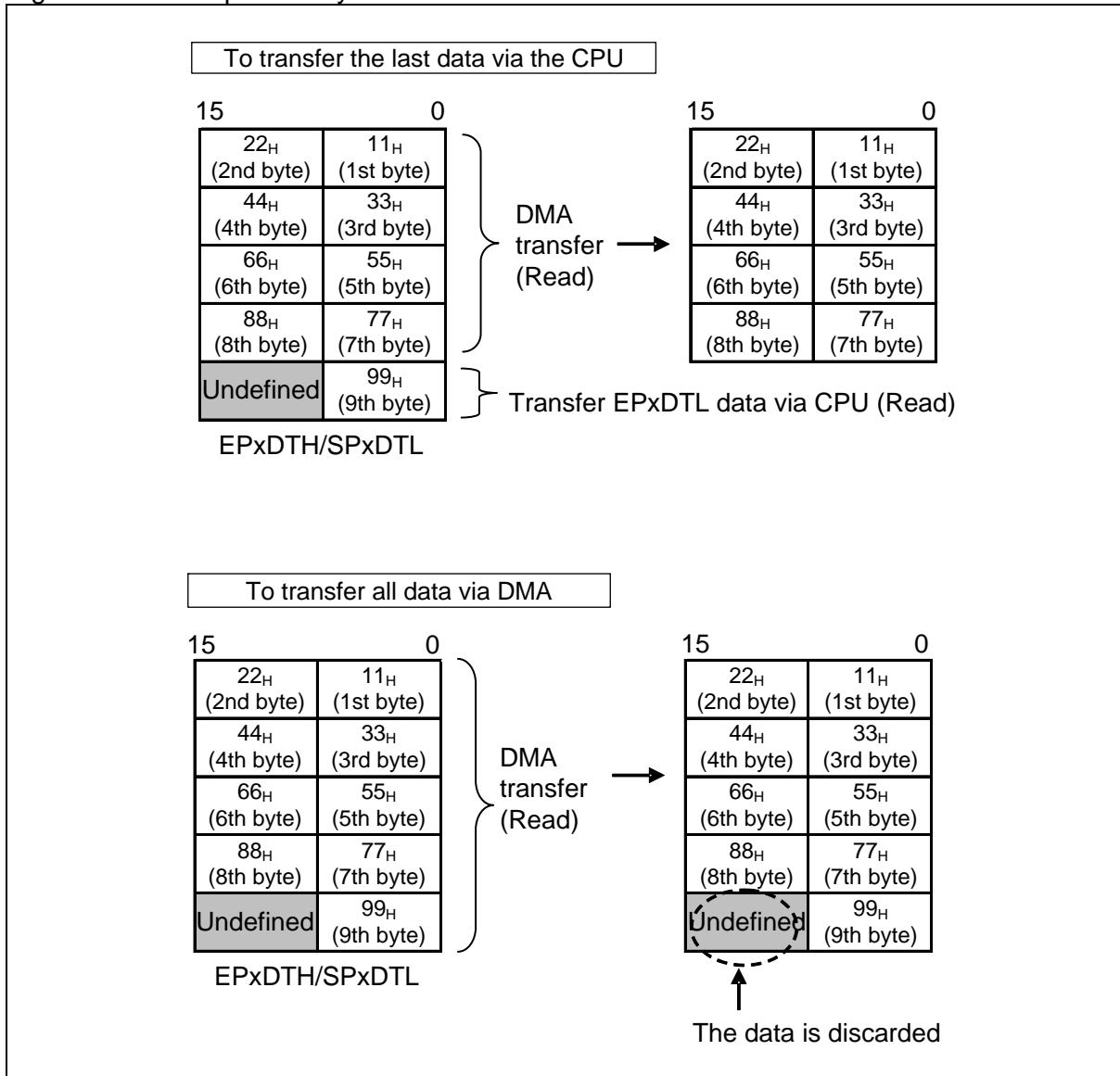
In the OUT direction transfer, the device must be processed in the following sequence:

1. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the total data size, and then enable DMA to start the transfer.
2. Enable DMAE and DRQIE.
3. After the transfer, reconfigure the DMAC using an interrupt generated by the interrupt cause pertinent to the DMAC status register, and clear the flag to return from the interrupt handling.

To transfer the data size corresponding to the odd bytes via DMA, either of the following methods are available:

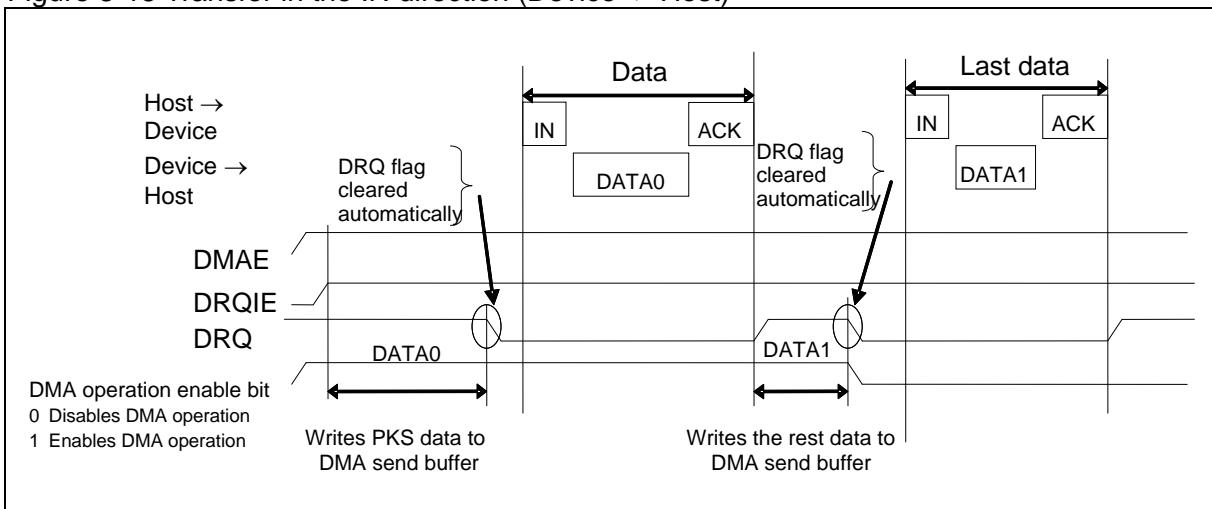
- Use the CPU transfer only for the last data, and read the low-order byte (EPxDTL).
- Transfer all the data + 1 byte via DMA, and discard the last data after an endian conversion.

Figure 3-14 Example odd bytes transfer in the OUT direction



● Transfer in the IN direction (Host -> Device)

Figure 3-15 Transfer in the IN direction (Device -> Host)

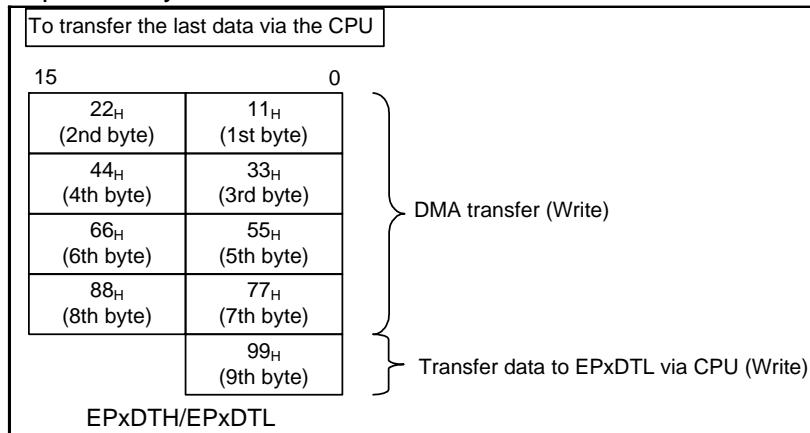


In the IN direction transfer, the device must be processed in the following sequence:

1. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the total data size, and then enable DMA to start the transfer.
2. Enable DMAE and DRQIE.
3. After the transfer, reconfigure the DMAC using an interrupt generated by the interrupt cause pertinent to the DMAC status register, and clear the flag to return from the interrupt handling.

To transfer the data size corresponding to the odd bytes via DMA, use the CPU transfer only for byte writing to the last data.

Figure 3-16 Example odd bytes transfer in the IN direction



3.7. NULL transfer function

If data sent from the USB function is the last packet and satisfies the maximum packet size, then the 0-byte can be automatically transferred via the next packet transfer. DMAE must be enabled to use this function. This function is valid only in IN transfer.

■ NULL transfer mode

NULL transfer mode sends 0-byte in reply to the next host's data request in the IN direction after the last data in the IN direction has been transferred.

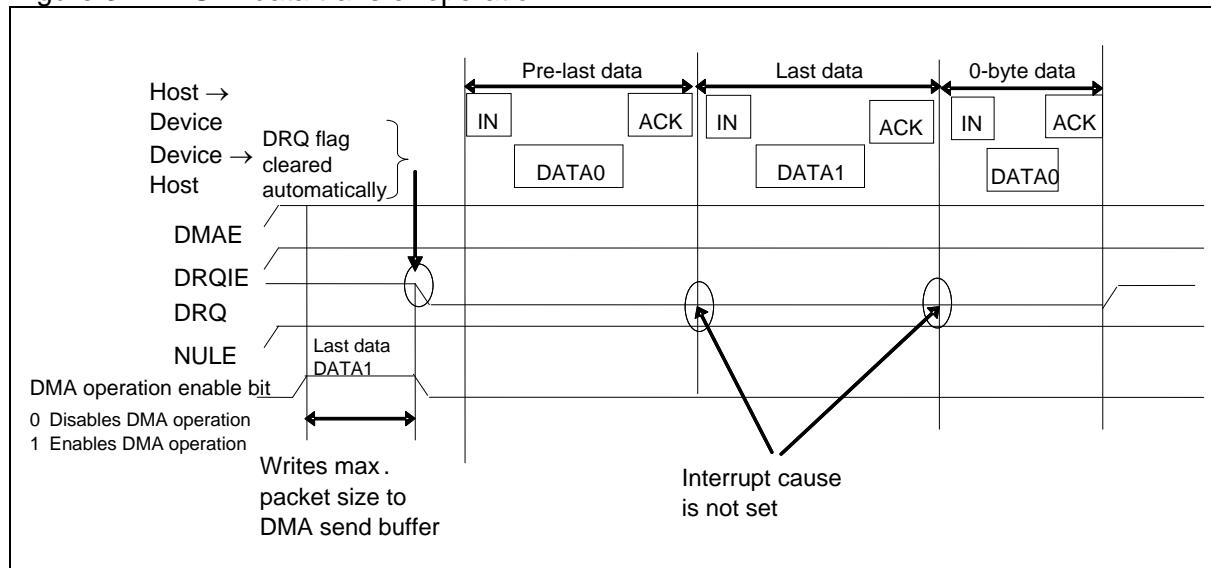
NULL transfer mode works when the following conditions are met:

- Automatic buffer transfer mode is set (DMAE = 1)
- The last data transfer writes the maximum packet size to the DMA buffer
- DMA data units are counted as 0 by writing the last data

After the last data has been written to buffer via DMA, the DRQ interrupt flag is not set until the 0-byte data is read from the host. The following shows the timing to access the buffer.

Only the transfer in the IN direction (Device -> Host) is explained.

Figure 3-17 NULL data transfer operation



The device must be processed as follows:

1. Enable DMAE, DRQIE, and NULE.

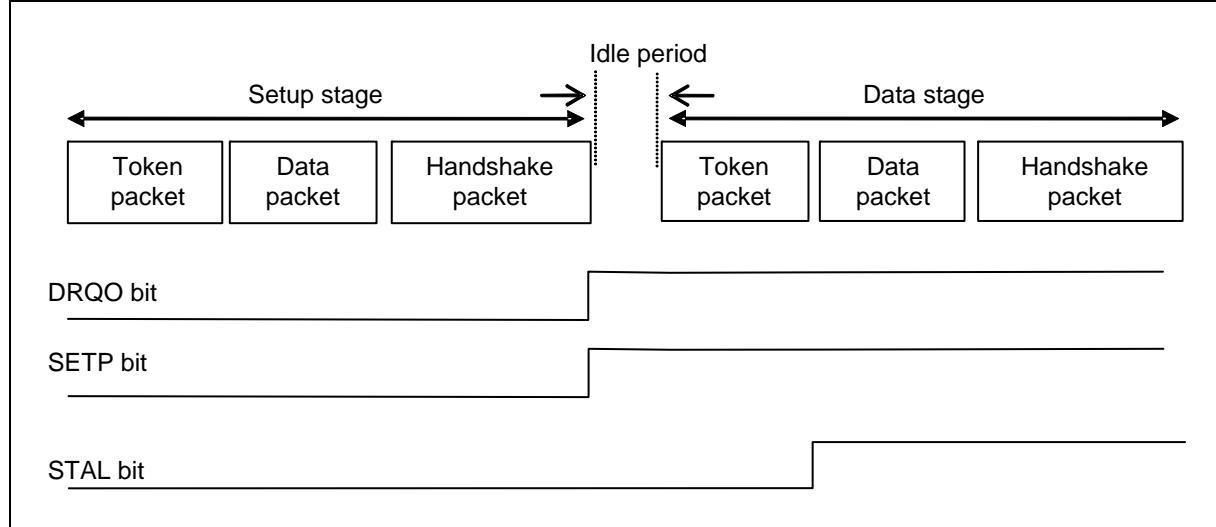
3.8. STALL response/release of endpoint 0

The STAL bit in the EP0 Control Register (EP0C) controls the STALL response and release of Endpoint 0.

■ STAL bit set timing

To perform the STALL response, interpret the command at the setup stage (SETP bit = 1 detection) of control transfer. If the STALL response is required, set the STAL bit. (See Figure 3-18) After setting the STAL bit, clear the interrupt cause (DRQO bit).

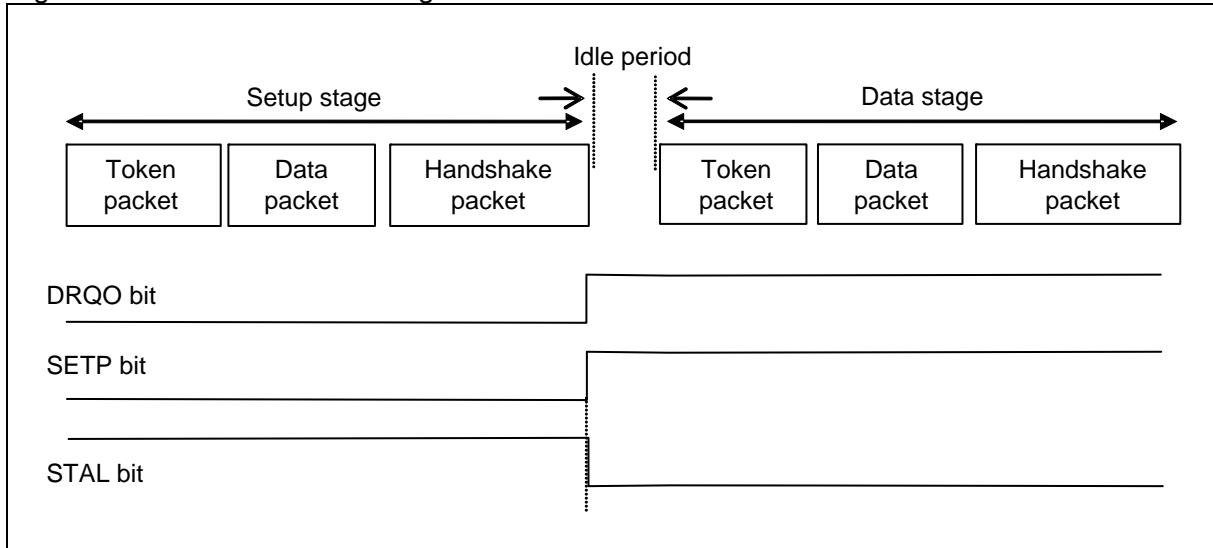
Figure 3-18 STAL bit set timing



■ STAL bit clear timing

Upon the detection of SETP bit = 1, pointing to the setup stage of control transfer, the STAL bit is automatically cleared and the STALL state is released. (See Figure 3-19)

Figure 3-19 STAL bit clear timing



<Note>

Upon the detection of SETP bit = 1 (DRQO bit = 1 interrupt), the STAL bit is cleared to "0". To enable the STALL response again, set the STAL bit to "1".

3.9. Stall response/release of endpoints 1 to 5

The STAL bit and the internal status bit in the EP1 to 5 Control Registers (EP1C to EP5C) controls the STALL response and release of Endpoints 1 to 5.

■ Stall response processed by software

Figure 3-20 and Figure 3-21 shows the procedure to process the STALL response by software. To perform the STALL response, configure the STAL bit of relevant Endpoint by software. The internal status bit does not change then.

When a transaction occurs from the host to the Endpoint to which the STAL bit is set, the hardware automatically sets the internal status bit of the relevant Endpoint to perform the STALL response to the host. Once the internal status bit is set, it remains set even when the STAL bit cleared. As the internal state bit remains set until the host issues the Clear Feature command, the STALL response remains running. While the STALCLREN bit of the UDC Control Register (UDCC) is set to "0", the STALL response also remains running in the following condition:

The STAL bit remains set even after the internal status bit is cleared by the Clear Feature command.

This is because the internal status bit is set each time a transaction occurs to the relevant Endpoint. To release the STALL response, therefore, the STAL bit must be cleared, and the internal status bit must be cleared by the Clear Feature command. If the STALCLREN bit in the UDC Control Register (UDCC) is set to "1", the STAL bit is cleared at the same time the internal status bit is cleared by the Clear Feature command, and the STALL response is not performed for the next transaction.

Figure 3-20 To process the STALL response by software (the STAL bit is cleared by software)
UDCC.STALCLREN=0

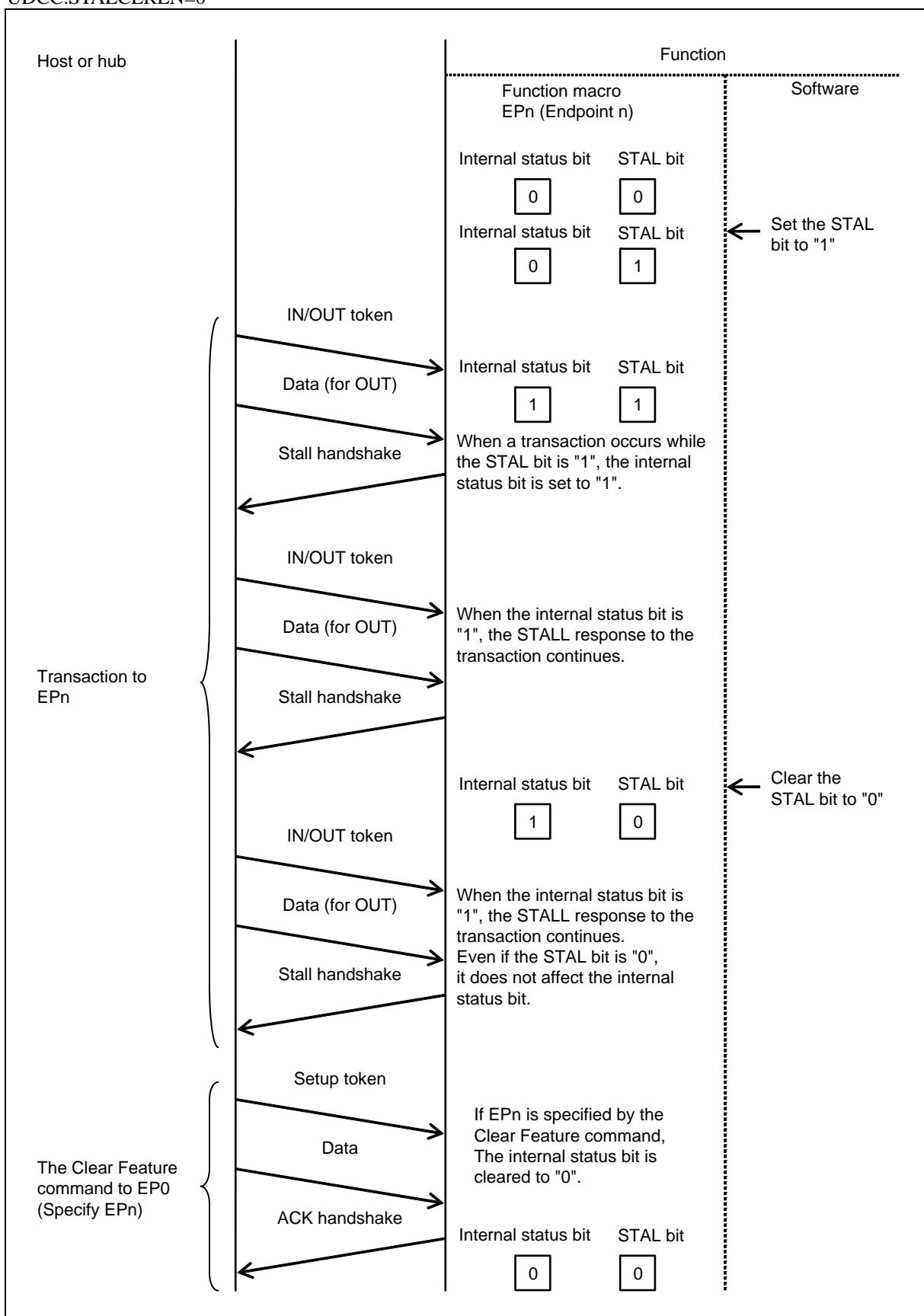
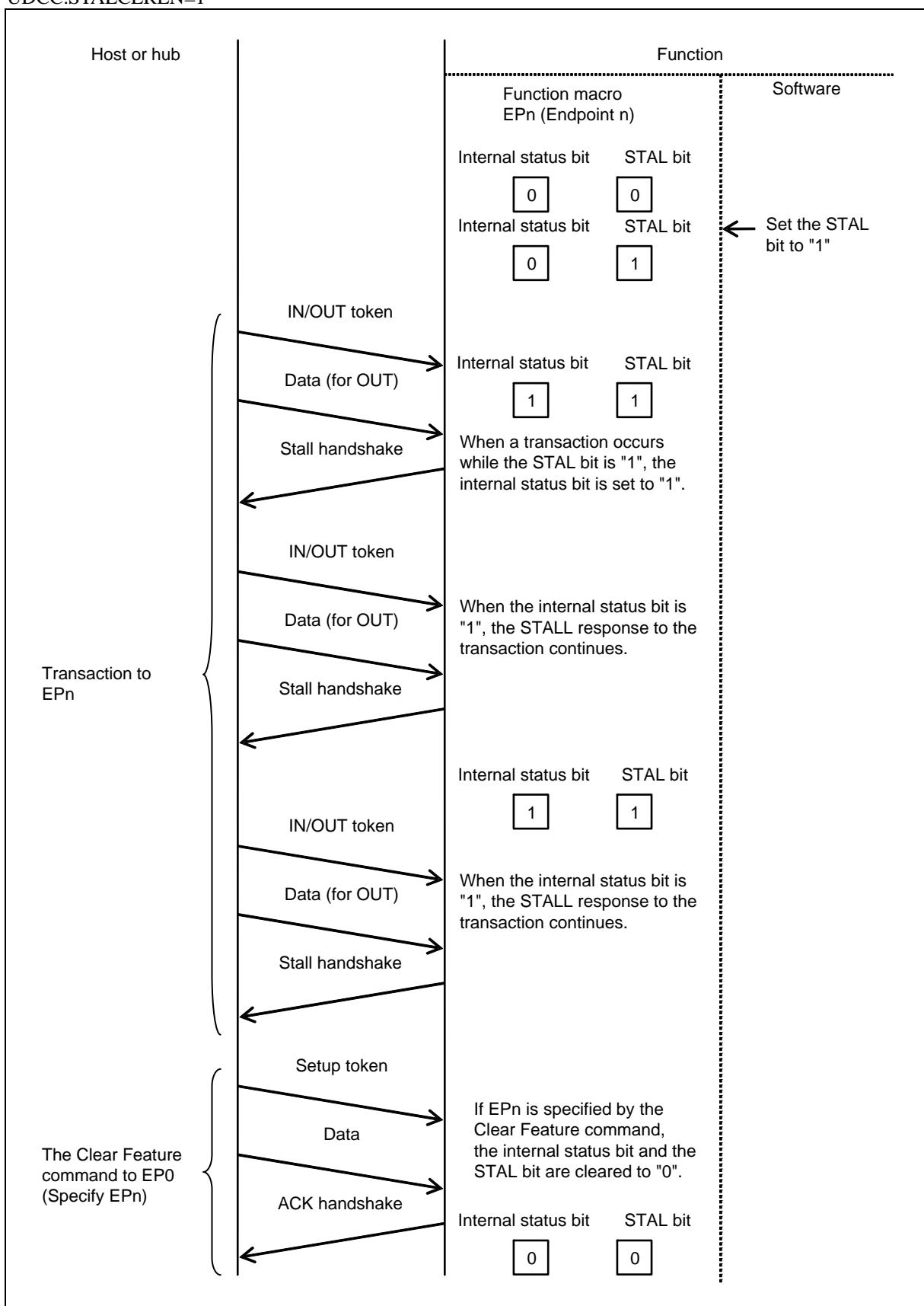


Figure 3-21 To process the STALL response by software (the STAL bit is cleared by hardware)
UDCC.STALCLREN=1



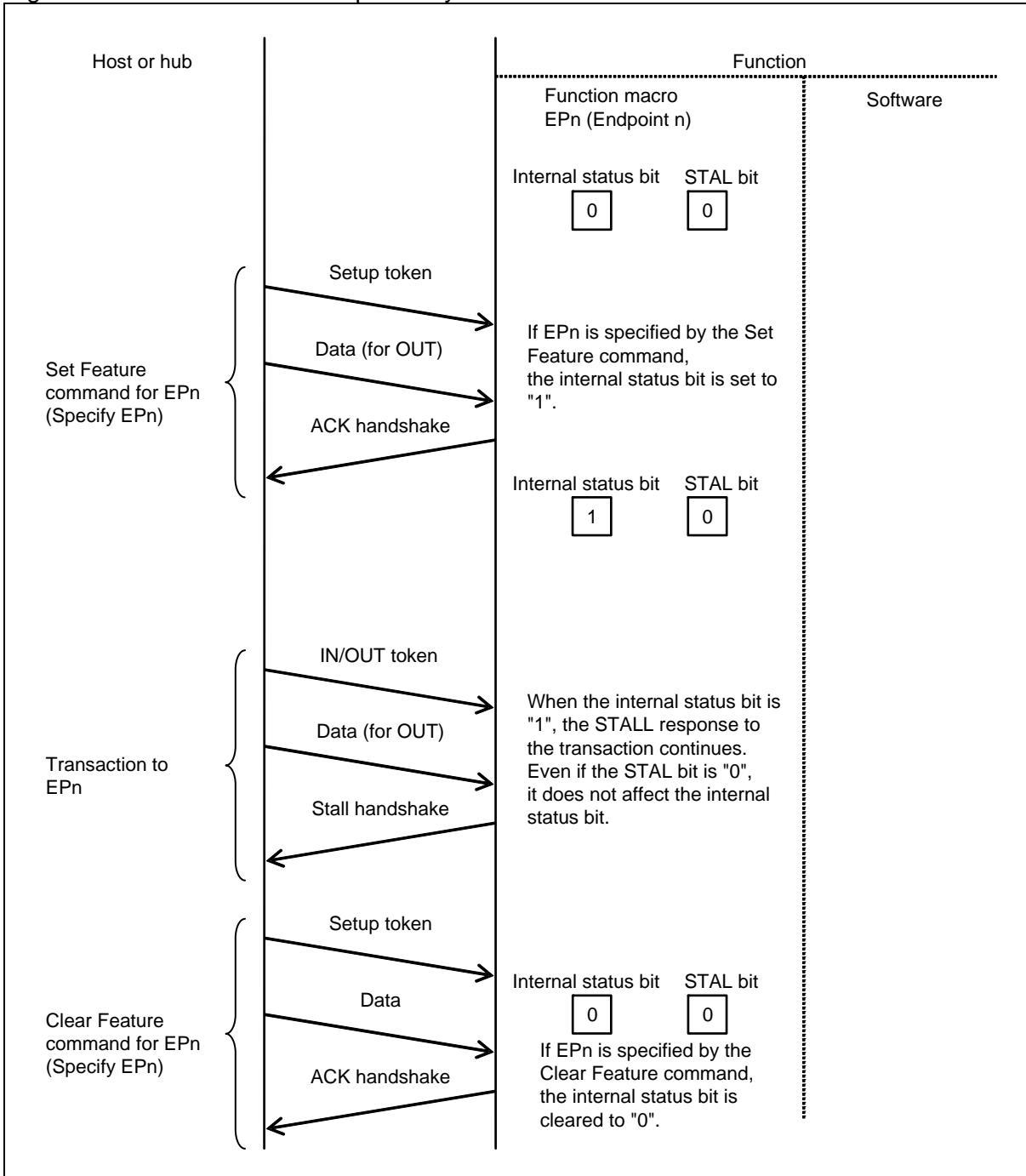
■ Automatic STALL response by hardware

Figure 3-22 shows the procedure for the automatic STALL response by hardware.

When the STALL response is set by the Set Feature command, the hardware automatically set the internal status bit of the relevant Endpoint, irrespective of the STAL bit setting, and perform the STALL response. Once the internal bit is set, the value is retained until cleared by the Clear Feature command from the host irrespective of the STAL bit setting.

The STAL bit is referred to even after the internal status bit is cleared by the Clear Feature command. To release the STALL response, therefore, the internal status bit must be cleared by the Clear Feature command.

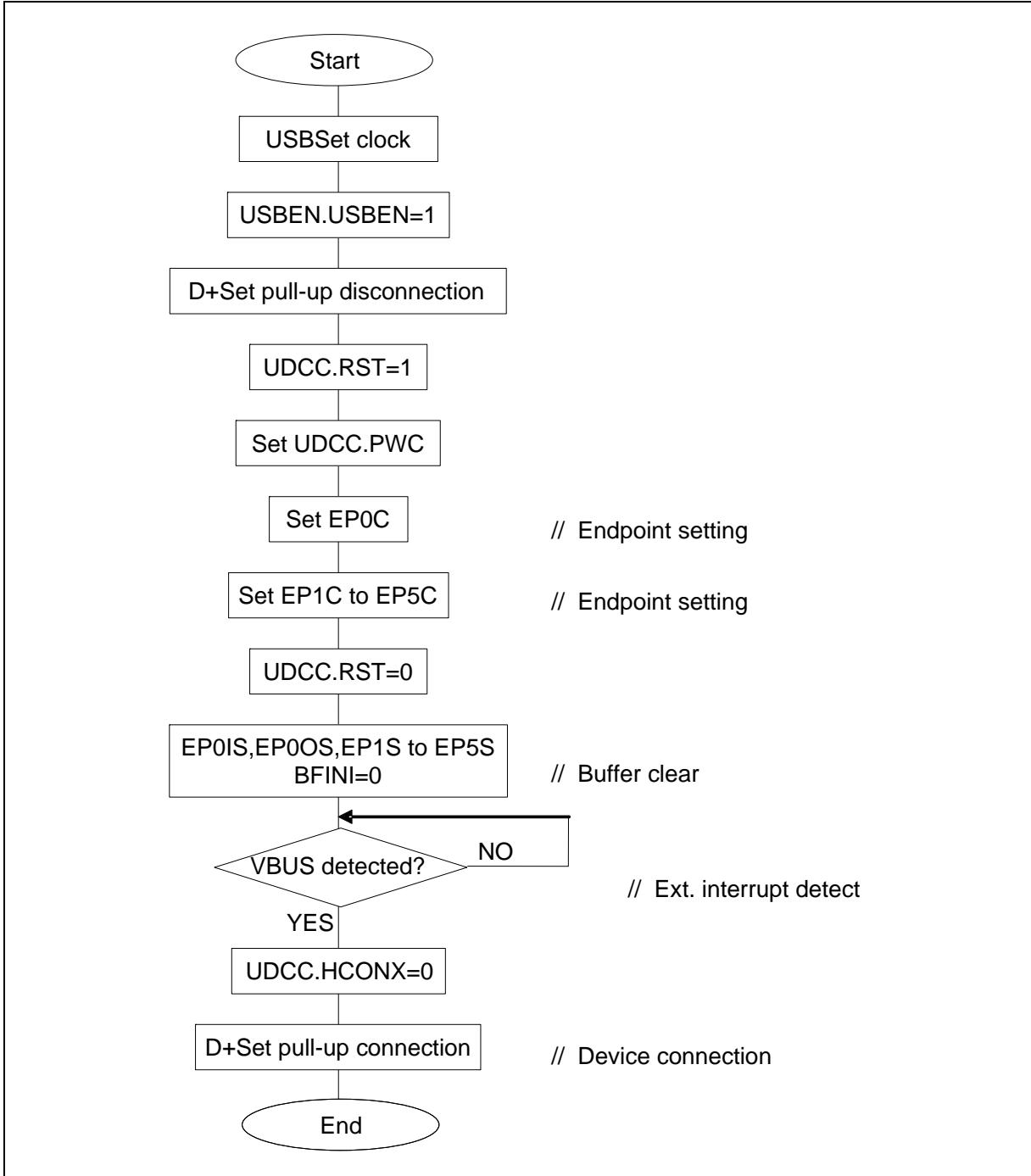
Figure 3-22 Automatic Stall Response by Hardware

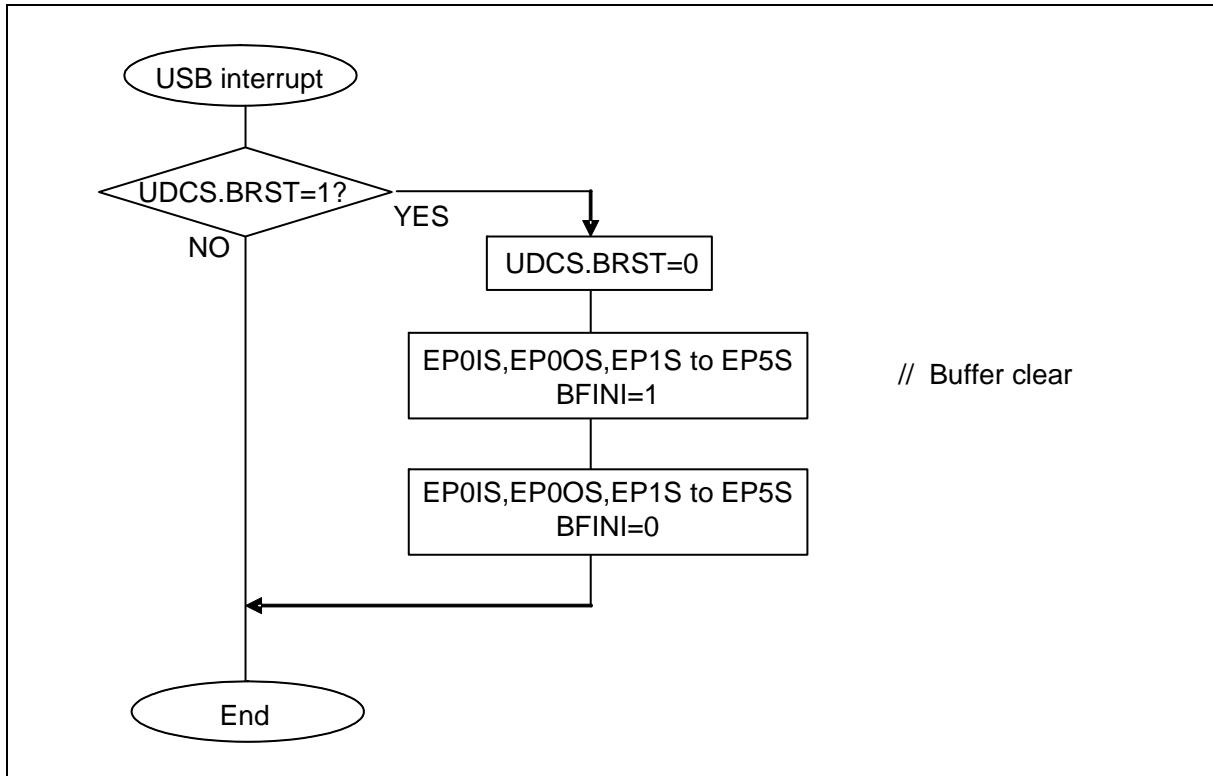


4. Examples of USB Function Setting Procedures

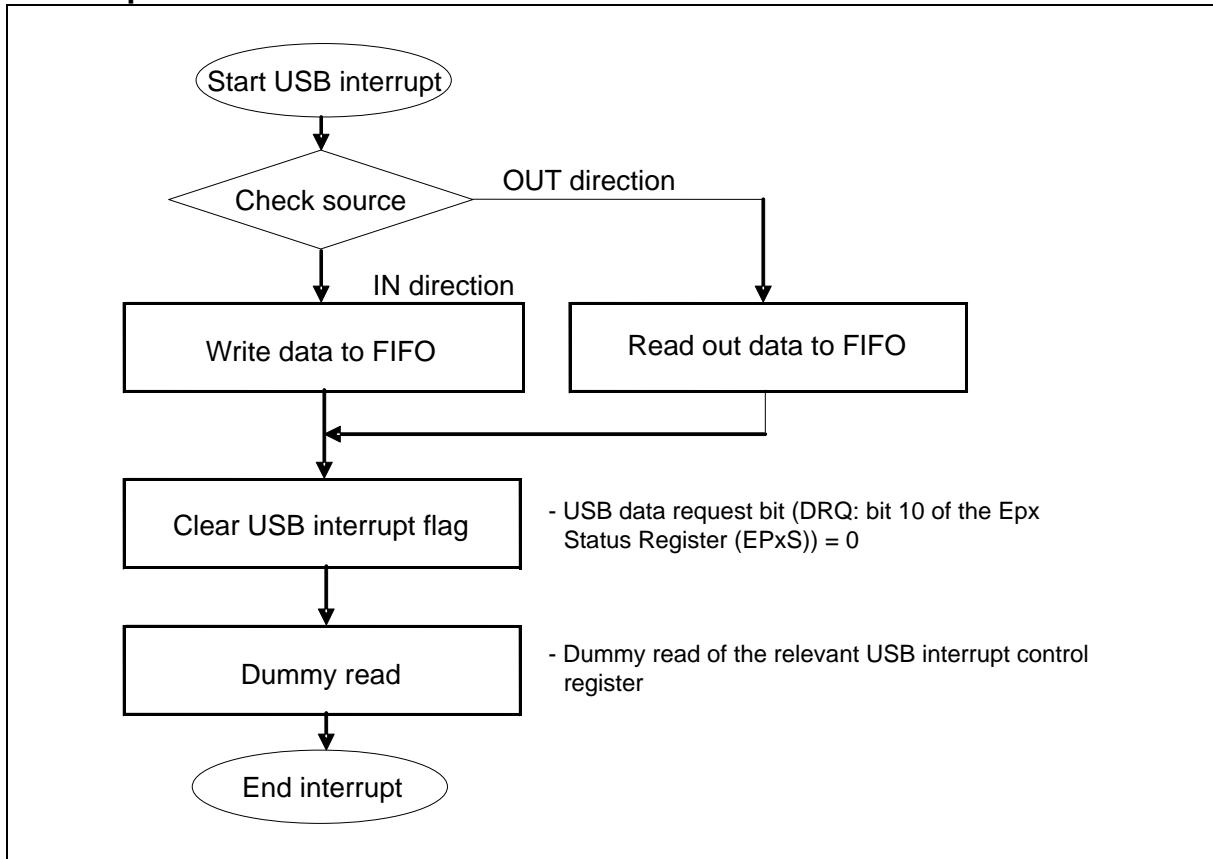
This section provides flowcharts for initialization, bus reset, CPU transfer, packet transfer (IN/OUT) and automatic data size transfer (IN/OUT).

■ Initialization



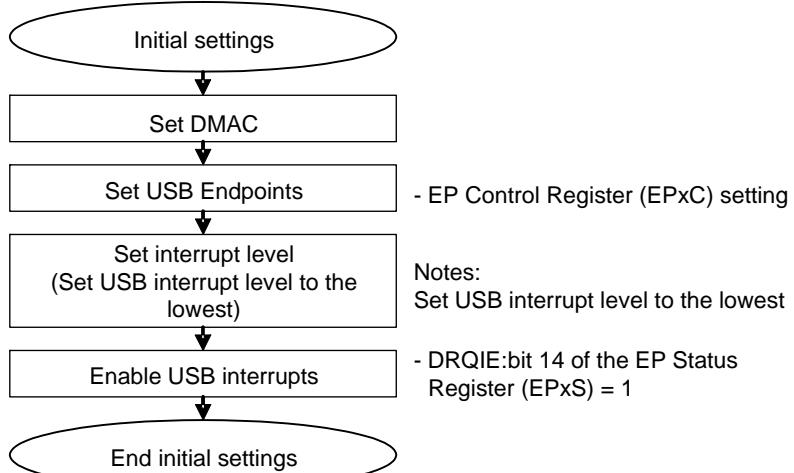
■ Bus reset

■ Example control for CPU transfer

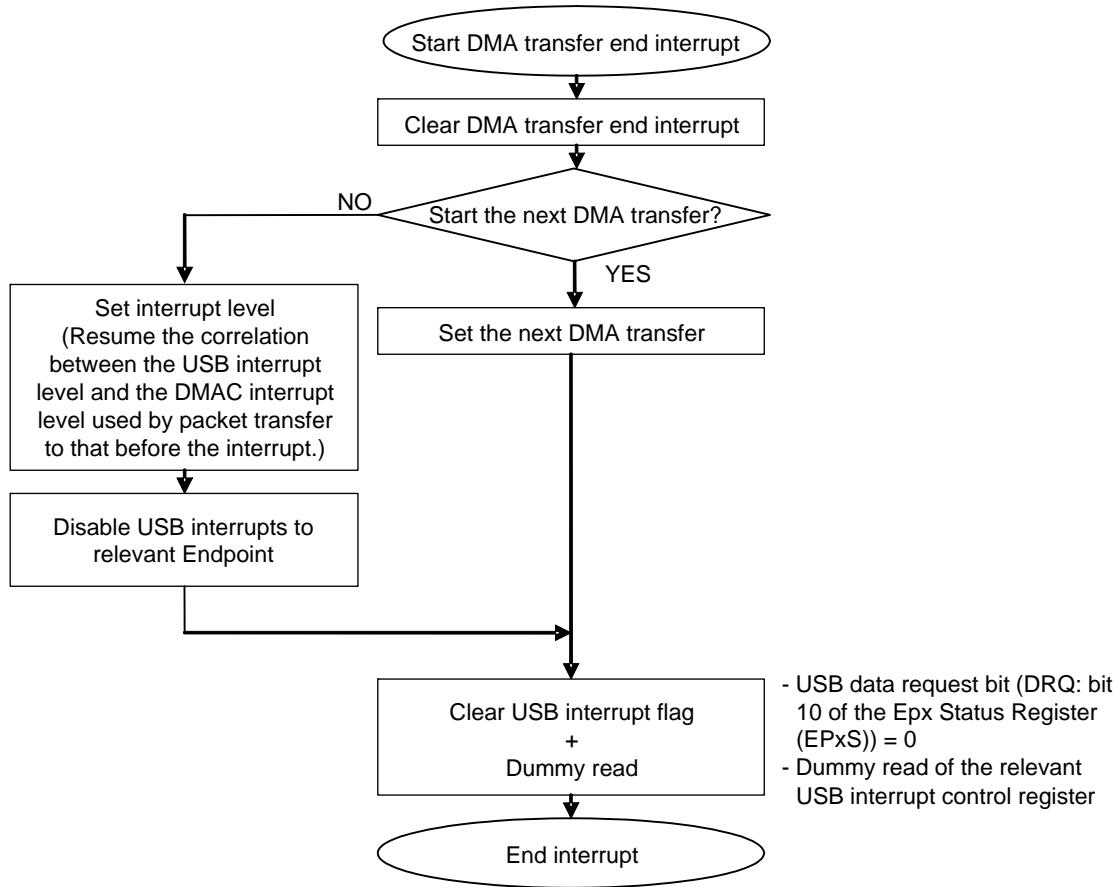


■ Example control for packet transfer in IN direction

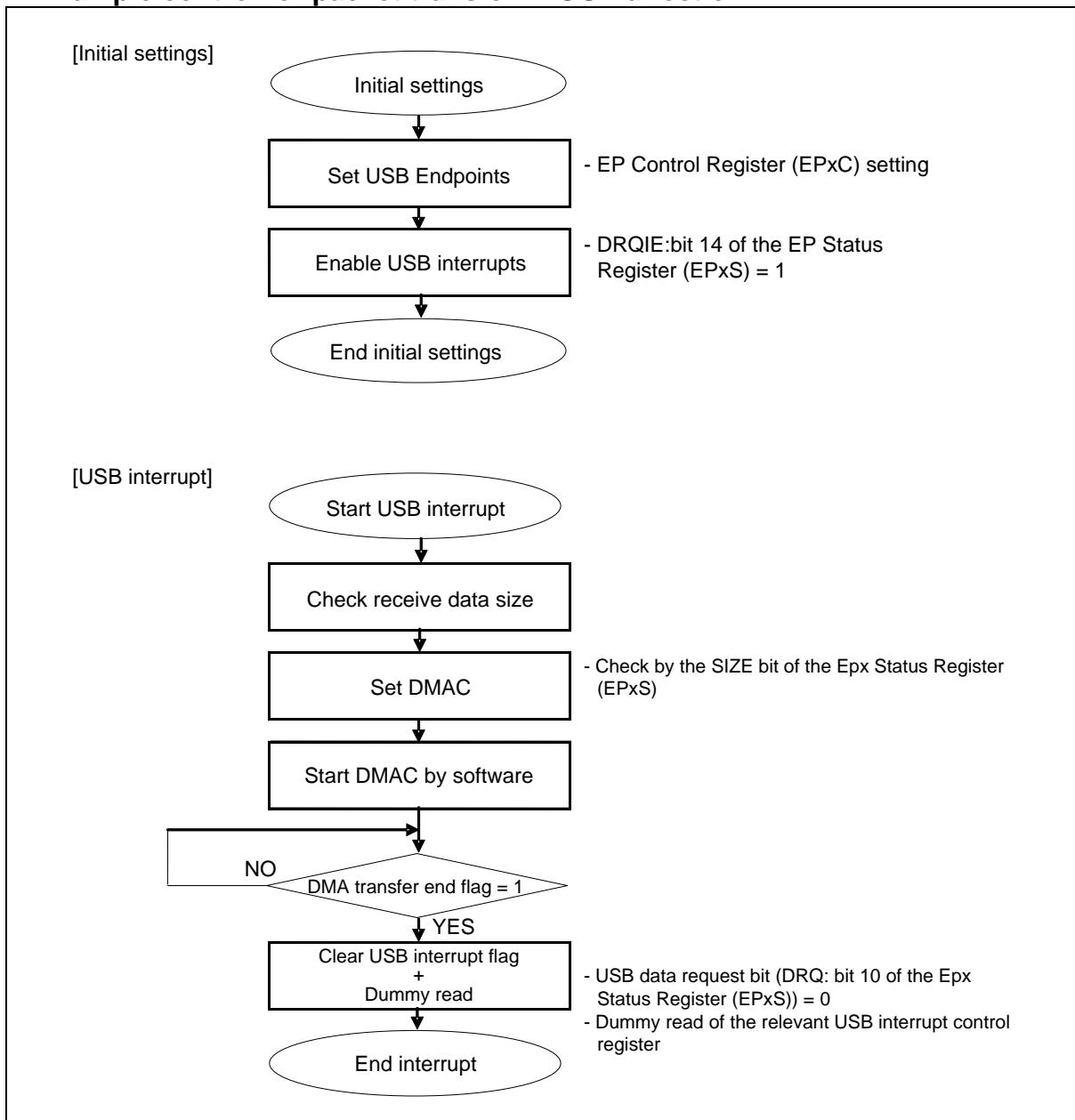
[Initial settings]



[DMA transfer end interrupt]

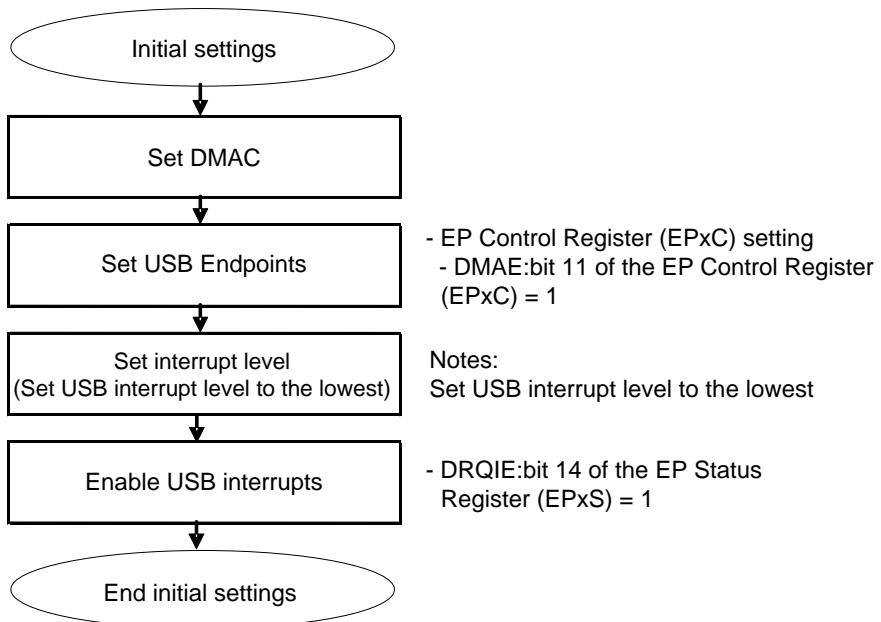


■ Example control for packet transfer in OUT direction

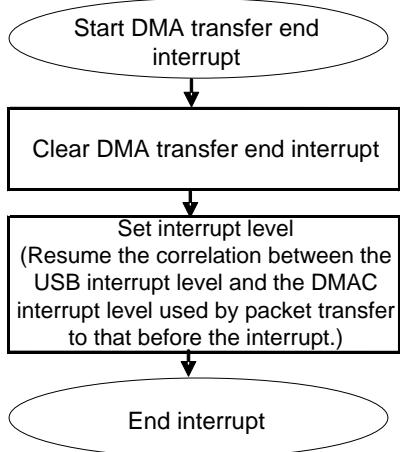


■ Example control for automatic data size transfer in IN direction

[Initial settings]

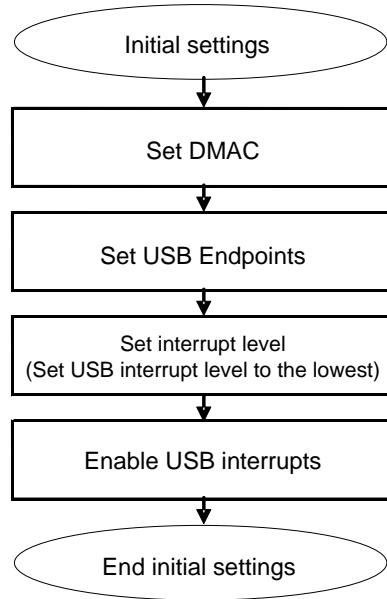


[DMAtransfer end interrupt]



■ Example control for automatic data size transfer in OUT direction

[Initial settings]

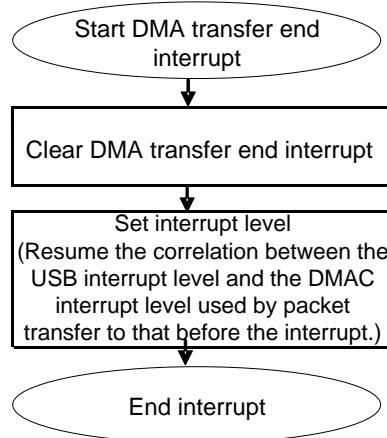


- EP Control Register (EPxC) setting
- DMAE:bit 11 of the EP Control Register (EPxC) = 1

Notes:
Set USB interrupt level to the lowest

- DRQIE:bit 14 of the EP Status Register (EPxS) = 1

[DMAtransfer end interrupt]



5. USB Function Registers

This section explains the configuration and functions of the registers used for the USB function.

■ USB function register list

Abbreviation	Register name	See
UDCC	UDC Control Register	5.1
EP0C	EP0 Control Register	5.2
EP1C	EP1 Control Register	5.3
EP2C	EP2 Control Register	5.3
EP3C	EP3 Control Register	5.3
EP4C	EP4 Control Register	5.3
EP5C	EP5 Control Register	5.3
TMSP	Time Stamp Register	5.4
UDCS	UDC Status Register	5.5
UDCIE	UDC Interrupt Enable Register	5.6
EP0IS	EP0I Status Register	5.7
EP0OS	EP0O Status Register	5.8
EP1S	EP1 Status Register	5.9
EP2S	EP2 Status Register	5.9
EP3S	EP3 Status Register	5.9
EP4S	EP4 Status Register	5.9
EP5S	EP5 Status Register	5.9
EP0DTH	EP0 Data Register high-order	5.10
EP0DTL	EP0 Data Register low-order	5.10
EP1DTH	EP0 Data Register high-order	5.10
EP1DTL	EP0 Data Register low-order	5.10
EP2DTH	EP0 Data Register high-order	5.10
EP2DTL	EP0 Data Register low-order	5.10
EP3DTH	EP0 Data Register high-order	5.10
EP3DTL	EP0 Data Register low-order	5.10
EP4DTH	EP0 Data Register high-order	5.10
EP4DTL	EP0 Data Register low-order	5.10
EP5DTH	EP0 Data Register high-order	5.10
EP5DTL	EP0 Data Register low-order	5.10

■ UDCC.RST dependent register bit update timing list

	Register	Bit
Register bits to be updated when UDCC.RST=1	UDCC	HCONTX, PFBK, PWC
	EP0C	PKS0
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
	EP3C	EPEN, TYPE, DIR, PKS3
	EP4C	EPEN, TYPE, DIR, PKS4
	EP5C	EPEN, TYPE, DIR, PKS5
Register bits initialized when UDCC.RST=1 (Update when UDCC.RST=0)	EPOIS	BFINI, DRQI
	EP0OS	BFINI, DRQ, SPK
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
	EP3S	BFINI, DRQ, SPK
	EP4S	BFINI, DRQ, SPK
	EP5S	BFINI, DRQ, SPK
	TMSP	TMSP
	UDCS	SUSP, SOF, BRST, WKUP, SETP, CONF
	UDCIE	SUSPIE, SOFIE, BRSTIE, WKUPIE, CONFN, CONFIE
Register bits unaffected by UDCC.RST	UDCC	RESUME, USTP
	EP0C	STAL
	EP1C	DMAE, NULE, STAL
	EP2C	DMAE, NULE, STAL
	EP3C	DMAE, NULE, STAL
	EP4C	DMAE, NULE, STAL
	EP5C	DMAE, NULE, STAL
	EP1DTH/L	BFDT
	EP2DTH/L	BFDT
	EP3DTH/L	BFDT
	EP4DTH/L	BFDT
	EP5DTH/L	BFDT

5.1. UDC Control Register (UDCC)

The UDC Control Register (UDCC) controls the UDC core circuit.

The following figure shows the bit configuration of the UDC Control Register (UDCC).

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RST	RESUM	HCONX	USTP	STALCLREN	Reserved	RFBK	PWC
Attribute	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
Initial value	1	0	1	0	0	0	0	0

<Note>

The UDC Control Register (UDCC), except bit 6 RESUM and bit 4 USTP, should be configured while bit 7 RST = 1, and should not be rewritten while USB is running. Bit 6 RESUM must be set or reset in USB suspend mode and while the remote wake-up is enabled by the following command.

Set bit 4 USTP to "1" before stop mode or timer mode is entered.

When those modes have been released, set the SUSP of UDSCS and USTP of UDCC to "0" in this order after confirmation of stabilized USB supply clock.

The following explains the function of each bit in the UDC Control Register (UDCC).

[bit 15:7] Reserved bits

These bits are reserved. Always write "0" to these bits. They are always read as "0".

[bit 7] RST: Function Reset Bit (function ReSeT)

This bit is ORed with the chip system reset to individually resets the USB function. The USB function is reset by the RST bit when connected with the host via cable. As the initial value is "1", reset enabled, write "0" to release the state.

Bit	Description
0	Releases USB Function reset
1	Resets the USB function

<Note>

This bit initializes the relevant bit of the Time Stamp Register, UDC Status Register, Interrupt Enable Register at the same time. It also sets the BFINI of the EP0I, EP0O, and EP1 to 5 Status Register concurrently. After the initial settings, therefore, clear the RST bit (BFINI is not cleared) and clear BFINI of the Endpoints used in this order.

[bit 6] RESUM: Resume Setting Bit (RESUME set)

In suspend state while remote wake-up is enabled *, the resume is started when writing "1" to the RESUM bit. To instruct to resume, set the RESUM bit to "1", and then write "0" to it to clear.

* : The DEVICE_REMOTE_WAKEUP bit is set by the SET_FEATURE command from the host.

Bit	Description
0	Resets the USB resume start instruction bit
1	Instructs to start the USB resume

[bit 5] HCONX: Host Connection Bit (Host CONnection)

This bit controls the switch between an external pull-up resistor and the USB data line to make the connection with the host or hub recognized.

Bit	Description
0	Connected to the host or hub
1	Disconnected from the host or hub

<Note>

Even if the connection is found by the host or hub while the external pull-up resistor is kept ON, the bus reset command on the USB bus is ignored while this bit is "1".

[bit 4] USTP: USB Operating Clock Stop Bit (Udc SToP)

Setting this bit stops the clock for the USB operating unit. When USB is not operated, power consumption can be reduced by configuring this bit.

Bit	Description
0	Normal mode
1	Stops the clock for the USB operating unit

<Note>

If stop mode and timer mode is not set, the USTP bit must be configured after setting RST to "1", and also after 3 cycles at full speed or 43 cycles at low speed (supported only in host mode) so that the reset can be ensured. This bit can be cleared at the same time RST is cleared.

[bit 3] STALCLREN: Endpoint 1 to 5 STAL Bit Clear Select Bit (STALI CLeaR Enable)

This bit selects the method to clear the STAL bit of Endpoint 1 to Endpoint 5 using the Clear Feature command. The STALCLREN bit sets whether to automatically clear the STAL bit to "0" by hardware, a bit of EP1 to EP5 Control Registers (EP1C to EP5C) for Endpoints (1 to 5) specified by the Clear Feature command. This bit selects the method to clear the STAL bit of the Endpoint Control Registers (EP1C to EP5C), either by software or hardware.

Bit	Description
0	Clears the STAL bit of the EP1 to EP5 Control Registers (EP1C to EP5C) by software.
1	Automatically clears the STAL bit of the EP1 to EP5 Control Registers (EP1C to EP5C) by hardware.

<Note>

The STALCLREN bit should be configured while the RST of the UDC Control Register (UDCC) is "1", and should not be rewritten while USB is running.

[bit 2] Reserved bit

This bit is reserved. Always write "0" to this bit. It is always read as "0".

[bit 1] RFBK: Data Toggle Mode Select Bit (Rate Feed BacK mode)

This bit selects the data toggle mode for USB interrupt transfer.

Bit	Description
0	Selects the alternating data toggle mode. Toggles data PID when the transfer has finished successfully.
1	Selects the data toggle mode. Unconditionally toggles data PID.

[bit 0] PWC: Power Control Bit (PoWer Control)

This bit specifies the operating power mode (self power or bus power) of the USB function.
(Configuration of this bit applies to standard command GetStatus.)

Bit	Description
0	Bus power
1	Self power

5.2. EP0 Control Register (EP0C)

The EP0 Control Register (EP0C) controls Endpoint 0.

The following figure shows the bit configuration of the EP0 Control Register (EP0C).

bit	15	14	13	12	11	10	9	8
Field	-	-	-	-	Reserved	Reserved	STAL	Reserved
Attribute	-	-	-	-	-	-	R/W	-
Initial value	X	X	X	X	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	PKS0						
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	0	0	0	0	0	0

<Note>

Except bit 9 STAL, the EP0 Control Register (EP0C) must be configured while both of the bit 7 RST in the UDC Control Register (UDCC) and bit 7 BFINI in the EP0I/O Status Register (EP0I/OS) are "1". It must not be rewritten while USB is running.

The following explains the function of each bit in the EP0 Control Register (EP0C).

[bit 15:12] Undefined bits

The written value has no effect. The read value is undefined.

[bit 11:10] Reserved bits

These bits are reserved. Always write "0" to these bits.

They are always read as "0".

[bit 9] STAL: Endpoint 0 Stall Setting Bit (STALI ep0 set)

This bit can set Endpoint 0 to the STALL state (STALL response).

This bit is cleared by hardware. If a SETUP packet is received by Endpoint 0 after the STALL response to Endpoint 0 is performed, this bit is cleared to "0". For the timing to clear this bit, see " STAL bit clear timing" of "3.8 STALL response/release of endpoint 0".

Bit	Description
0	Ignored
1	Sets the STALL state (STALL response)

<Notes>

- If the STALCLREN bit of the USB Enable Register (USBEN) is "0", the STALL response remains operating to the host while the STAL bit is set to "1". Upon the receipt of a normal SETUP packet after STAL bit reset, Endpoint 0 resumed from the STALL state.
- A read-modify-write instruction read this bit as "0".

[bit 8:7] Reserved bits

These bits are reserved. Write value should always be "0".

They are always read as "0".

[bit 6:0] PKS0: Packet Size Endpoint 0 Setting Bits (PacKet Size ep0 set)

These bits specify the maximum number of bytes transferred by one packet. For Endpoint 0, the maximum number of bytes is 64, and the set value is valid both for IN and OUT directions.

Example: "0x08" => 8 bytes, "0x40" => 64 bytes (maximum value)

<Notes>

- These bits must be configured when both of the RST bit in the UDC Control Register (UDCC) and the BFINI bit in the EP0I/O Status Register (EP0I/OS) are "1". Do not rewrite while USB is running.
- A value exceeding the maximum number of transferable bytes (0x40), and "0x00" must not be written.

5.3. EP1 to 5 Control Registers (EP1C to EP5C)

The EP1 to 5 Control Registers (EP1C to EP5C) control Endpoints 1 to 5.

The following figure shows the bit configuration of the EP1 to 5 Control Registers (EP1C to EP5C).

■ EP1 Control Register (EP1C)

bit	15	14	13	12	11	10	9	8
Field	EPEN	TYPE		DIR	DMAE	NULE	STAL	PSK1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	11		0	0	0	0	1
bit	7	6	5	4	3	2	1	0
Field	PSK1							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ EP2 to 5 Control Registers (EP2C to EP5C)

bit	15	14	13	12	11	10	9	8
Field	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Initial value	0	11		0	0	0	0	0
bit	7	6	5	4	3	2	1	0
Field	Reserved	PKS5 to 2						
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	0	0	0	0	0	0

<Note>

Except DMAE, NULE, and STAL, the EP1 to 5 Control Registers (EP1C to EP5C) must be configured while both of the bit 7 RST in the UDC Control Register (UDCC) and bit 15 BFINI in the EP0 to 5 Status Registers (EP1S to EP5S) are "1". They must not be rewritten while USB is running.

The following explains the function of each bit in the EP1 to 5 Control Registers (EP1C to EP5C).

[bit 15] EPEN: Endpoints 1 to 5 Enable Bits (EndPoint1 to 5 ENable)

This bit enables the Endpoint. Based on the EPEN bit setting, the Endpoint is configured by the host as those used by the function. TYPE, DIR and PKS in the EP1 to EP5 Control Registers are valid as the configuration information.

Bit	Description
0	Disables the Endpoint
1	Enables the Endpoint

[bit 14:13] TYPE: Endpoint Transfer Type Select Bits (endpoint TYPE)

These bits specify the transfer type the Endpoint support.

Bit 14:13	Description
00	Setting disabled
01	Setting disabled
10	Bulk transfer
11	Interrupt transfer

[bit 12] DIR: Endpoint Transfer Direction Select Bit (endpoint DIRection)

This bit specifies the transfer direction the Endpoint support.

Bit	Function operating mode	Host operating mode (EP1 and EP2 only)
0	OUT Endpoint	IN Endpoint
1	IN Endpoint	OUT Endpoint

[bit 11] DMAE: DMA Automatic Transfer Enable Bit (DMA Enable)

This bit sets a mode that uses DMA for writing or reading transfer data to/from send/receive buffer, and automatically transfers the send/receive data synchronized with an data request in the IN or OUT direction by the host. Until the data size set in the DMA is reached, the data is transferred.

Bit	Description
0	Releases the automatic buffer transfer mode
1	Sets the automatic buffer transfer mode

<Note>

The CPU must not access the send/receive buffer while the DMAE bit is set to "1". For data transfer in the OUT direction, set the DMA transfer size to the multiples of that set in PKS in the EP1 to 5 Control Registers (EP1C to EP5C).

[bit 10] NULE: Null Automatic Transfer Enable Bit (NULI Enable set)

When a data transfer request in IN the direction is received while automatic buffer transfer mode is set (DMAE = 1), this bit sets a mode that transfers 0-byte data automatically upon the detection of the last packet transfer.

Bit	Description
0	Releases the NULL automatic transfer mode
1	Sets the NULL automatic transfer mode

<Note>

For data transfer in the OUT direction or when automatic buffer transfer mode is not set, the NULL bit configuration does not affect communication.

[bit 9] STAL: Endpoints 1 to 5 Stall Setting Bit (STALI set)

This bit can set Endpoint to the STALL state (STALL response).

- When the STALCLREN bit of the UDC Control Register (UDCC) is "0"
This bit is not cleared to "0" by the Clear Feature command. This bit must be cleared by software.
For the timing to clear this bit, see " Stall response processed by software" of "3.9 Stall response/release of endpoints 1 to 5".

Bit	Description
0	Release the STALL state
1	Sets the STALL state (STALL response)

- When the STALCLREN bit of the UDC Control Register (UDCC) is "1"
This bit is cleared by hardware. It is cleared to "0" for the Endpoint specified by the Clear Feature command. For the timing to clear this bit, see " Stall response processed by software" of "3.9 Stall response/release of endpoints 1 to 5".

Bit	Description
0	Ignored
1	Sets the STALL state (STALL response)

<Notes>

- If the STALCLREN bit of the UDC Control Register (UDCC) is "0", the STALL response remains operating to the host while the STAL bit is set to "1". Restoration from the STALL state is possible by the Clear Feature command after resetting the STAL bit.
- The value read by a read-modify-write instruction differs depending on the value set in STALCLREN.
 - When STALCLREN = 0, the value at that time is read.
 - When STALCLREN = 1, "0" is read.

[EP2 to EP5: bit 8:7] EP2 to EP5 reserved bits

In EP2 to EP5, these bits are reserved. Write value should always be "0". They are always read as "0".

[(EP1: bit 8:7) bits 6:0] PKS: Packet Size Setting Bits (PackEt Size ep1 set)

This bit specifies the maximum size transferred by one packet. The following shows the maximum packet size that can be specified for Endpoints 1 to 5.

EndPoint	Maximum transfer size	Configurable range
1	256 bytes (Odd numbers allowed)	0x001 to 0x100
2 to 5	64 bytes (Odd numbers allowed)	0x01 to 0x40

<Note>

A value exceeding the maximum number of transferable bytes (0x100 or 0x40), and "0x00" must not be written. For Endpoints 2 to 5, write "00" to bit 8 to 7, Also when automatic buffer transfer mode (DMAE = 1) is used, 0 to 2 must not be written to the relevant Endpoint.

5.4. Time Stamp Register (TMSP)

The Time Stamp Register (TMSP) indicates the frame number upon the receipt of SOF packets.

The following figure shows the bit configuration of the Time Stamp Register (TMSP).

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved		Reserved		TMSP		
Attribute	-	-		-		R	R	R
Initial value	X	X		XXX		0	0	0
RST reset	0	0		Irrelevant			0	0
bit	7	6	5	4	3	2	1	0
Field	TMSP							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0
RST reset	0	0	0	0	0	0	0	0

The following explains the function of each bit in the Time Stamp Register (TMSP).

[bit 15:11] Reserved bits

The written value has no effect. The read value is undefined.

[bit 10:0] TMSP: Time Stamp Bits (TiMe StamP)

These bits indicate the frame number of a received SOF packet. The frame number is updated upon the receipt of a SOF packet.

5.5. UDC Status Register (UDCS)

The UDC Status Register (UDCS) indicates the bus status during USB communication or the reception of specific commands. Each bit except the SETP bit is an interrupt cause, and so can generate an interrupt to the CPU if the correspondent interrupt enable bit is enabled.

The following figure shows the bit configuration of the UDC Status Register (UDCS).

bit	7	6	5	4	3	2	1	0
Field	-	-	SUSP	SOF	BRST	WKUP	SETP	CONF
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	0	0	0	0	0	0
RST reset	X	X	0	0	0	0	0	0

The following explains the function of each bit in the UDC Status Register (UDCS).

[bit 7:6] Undefined bits

The written value has no effect. The read value is undefined.

[bit 5] SUSP: Suspend detection bit (SUSPend)

This bit indicates that the USB function makes transition to suspend state. The SUSP bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Suspend undetected. Clears the interrupt cause.
1	Suspend detected

[bit 4] SOF: SOF Detection Bit (Start Of Frame)

This bit indicates that a SOF packet has been received, and then the Time Stamp Register value is updated. The SOF bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	SOF unreceived. Clears the interrupt cause.
1	SOF packet received

[bit 3] BRST: Bus Reset Detection Bit (Bus ReSeT)

This bit indicates the detection of a USB bus reset. The BRST bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	USB bus reset undetected. Clears the interrupt cause.
1	USB bus reset detected

<Note>

When this bit is detected, initialize the buffer by the BFINI bit in the EP0I Status Register (EP0IS), the BFINI bit in the EP0O Status Register (EP0OS), and the BFINI bit in the EP1 to EP5 Status Registers (EP1S to EP5S).

[bit 2] WKUP: Wake-up Detection Bit (WaKe UP)

This bit indicates that the USB function has resumed from suspend state. Remote wake-up caused by the RESUM bit setting, and wake-up caused by a request from the host are the resume sources, but the WKUP bit is automatically set only by a resume request by the host. The WKUP bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Host caused resume undetected. Clears the interrupt cause.
1	Host caused resume detected

<Note>

Even when wake-up caused by a host request occurs, this bit is not set if the RESUM bit in the UDCC register has been set.

[bit 1] SETP: Setup Stage Detection Bit (SETuP)

This bit indicates that the received data is the setup stage of USB control transfer. Writing "1" to this bit is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	SETUP stage unreceived. Clears the source.
1	Setup stage of control transfer received

<Note>

The SETP bit is not set during standard command automatic response. This bit is not an interrupt cause.

[bit 0] CONF: Configuration Detection Bit (CONFигuration)

This bit indicates that the USB function has been configured. The CONF bit is set when SetConfig of a USB command is received successfully. The CONF bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	SetConfig undetected. Clears the interrupt cause.
1	SetConfig detected

5.6. UDC Interrupt Enable Register (UDCIE)

The UDC Interrupt Enable Register (UDCIE) enables each bit (except the CONFN bit), interrupts generated by each interrupt cause of the UDC Status Register.

The following figure shows the bit configuration of the UDC Interrupt Enable Register (UDCIE).

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE
Attribute	-	-	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0
RST reset	0	Irrelevant	0	0	0	0	0	0

The following explains the function of each bit in the UDC Interrupt Enable Register (UDCIE).

[bit 15:14] Reserved bits

These bits are reserved. Always write "0" to these bits. They are always read as "0".

[bit 13] SUSPIE: Suspend Interrupt Enable Bit (SUSP Interrupt Enable)

This bit enables interrupts generated by the "SUSP" interrupt cause of the UDC Status Register.

Bit	Description
0	Disables interrupts generated by the SUSP cause
1	Enables interrupts generated by the SUSP cause

[bit 12] SOFIE: SOF Reception Interrupt Enable Bit (SOF Interrupt Enable)

This bit enables interrupts generated by the "SOF" interrupt cause of the UDC Status Register.

Bit	Description
0	Disables interrupts generated by the SOF cause
1	Enables interrupts generated by the SOF cause

[bit 11] BRSTIE: Bus Reset Enable Bit (BRST Interrupt Enable)

This bit enables interrupts generated by the "BRST" interrupt cause of the UDC Status Register.

Bit	Description
0	Disables interrupts generated by the BRST cause
1	Enables interrupts generated by the BRST cause

[bit 10] WKUPIE: Wake-up Interrupt Enable Bit (WKUP Interrupt Enable)

This bit enables interrupts generated by the "WKUP" interrupt cause of the UDC Status Register.

Bit	Description
0	Disables interrupts generated by the WKUP cause
1	Enables interrupts generated by the WKUP cause

[bit 9] CONFN: Configuration Number Indication Bit (CONFiguration Number)

This bit indicates the configuration number. The information is updated when the CONF interrupt cause of the UDC Status Register is set.

Bit	Description
0	CONFIG number 0
1	CONFIG number 1

[bit 8] CONFIE: Configuration Interrupt Enable Bit (CONFiguration)

This bit enables interrupts generated by the "CONF" interrupt cause of the UDC Status Register.

Bit	Description
0	Disables interrupts generated by the CONF cause.
1	Enables interrupts generated by the CONF cause.

5.7. EP0I Status Register (EP0IS)

The EP0I Status Register (EP0IS) indicates the status of the Endpoint 0 transfer in the IN direction.

The following figure shows the bit configuration of the EP0I Status Register (EP0IS).

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIIE	-	-	-	DRQI	-	-
Attribute	R/W	R/W	-	-	-	R/W	-	-
Initial value	1	0	X	X	X	1	X	X
BFINI reset	1	Irrelevant	X	X	X	1	X	X

bit	7	6	5	4	3	2	1	0
Field	-	-	-	-	-	-	-	-
Attribute	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X
BFINI reset	X	X	X	X	X	X	X	X

The following explains the function of each bit in the EP0I Status Register (EP0IS).

[bit 15] BFINI: Send Buffer Initialization Bit (BuFfer INItial)

This bit initializes the send buffer of transfer data. In addition, this bit is automatically set to "1" when the RST bit in the UDC Control Register (UDCC) is set to "1". If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing this bit.

Bit	Description
0	Clears the initialization
1	Initializes the send buffer

<Note>

Initialization by the BFINI bit initializes the buffer and the DRQI bit. Before initializing the buffer, make sure that the DRQI or DRQO bit is set, and there is no access from the host, and then configure the STAL bit if necessary.

[bit 14] DRQIIE: Send Data Interrupt Enable Bit (Data ReQuest In Interrupt Enable)

This bit enables interrupts generated by the "DRQI" interrupt cause of the EP0I Status Register.

Bit	Description
0	Disables interrupts generated by the DRQI cause.
1	Enables interrupts generated by the DRQI cause.

[bit 13:11] Undefined bits

The written value has no effect. The read value is undefined.

[bit 10] DRQI: Send/Receive Data Interrupt Request Bit (Data ReQuest In)

This bit indicates that the IN packet transfer from the EP0 host normally ended and data was read out from the send buffer, so that the next send data can be written. The DRQI bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Clears the interrupt cause
1	Send data can be written to the send buffer

<Note>

This bit must be cleared after data has been written to the send buffer. Also while this bit is not set, "0" must not be written.

Data can be written to the send buffer when DRQI is "1". Also when the DRQI bit is cleared, data has been set to the send buffer. When an IN packet request is received while the DRQI bit is "1", therefore, NAK is sent automatically to the host.

[bit 9:0] Undefined bits

The written value has no effect. The read value is undefined.

5.8. EP0O Status Register (EP0OS)

The EP0O Status Register (EP0OS) indicates the status of the Endpoint 0 transfer in the OUT direction.

The following figure shows the bit configuration of the EP0O Status Register (EP0OS).

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQOIE	SPKIE	-	-	DRQO	SPK	Reserved
Attribute	R/W	R/W	R/W	-	-	R/W	R/W	-
Initial value	1	0	0	X	X	0	0	0
BFINI reset	1	Irrelevant	Irrelevant	X	X	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved					SIZE		
Attribute	-	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X
BFINI reset	X	X	X	X	X	X	X	X

The following explains the function of each bit in the EP0O Status Register (EP0OS).

[bit 15] BFINI: Receive Buffer Initialization Bit (BuFFer INItial)

This bit initializes the receive buffer for transfer data. This bit is also automatically set by setting the RST bit of the UDC Control Register (UDCC). If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing this bit.

Bit	Description
0	Clears the initialization
1	Initializes the receive buffer

<Note>

Initialization by the BFINI bit initializes the DRQO and SPK bits. Before initializing the buffer, make sure that the DRQI or DRQO bit is set, and there is no access from the host, and then configure the STAL bit if necessary.

[bit 14] DRQOIE: Receive Data Interrupt Enable Bit (Data ReQuest Out Interrupt Enable)

This bit enables interrupts generated by the "DRQO" interrupt cause of the EP0O Status Register.

Bit	Description
0	Disables interrupts generated by the DRQO cause
1	Enables interrupts generated by the DRQO cause

[bit 13] SPKIE: Short Packet Interrupt Enable Bit (SPK Interrupt Enable)

This bit enables interrupts generated by the "SPK" interrupt cause of the EP0O Status Register.

Bit	Description
0	Disables interrupts generated by the SPK cause
1	Enables interrupts generated by the SPK cause

[bit 12:11] Undefined bits

The written value has no effect. The read value is undefined.

[bit 10] DRQO: Receive Data Interrupt Request Bit (Data ReQuest Out)

This bit indicates that the OUT packet transfer from the EP0 host normally ended, and data has been written to the receive buffer, which can be read out. This bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Clears the interrupt cause
1	Received data can be read from the receive buffer

<Note>

This bit must be cleared after data has been read from the receive buffer. Also while this bit is not set, "0" must not be written.

The receive buffer is not updated when DRQO is "1". The update is allowed when DRQO is cleared. When an OUT packet request is received while the DRQO bit is "1", therefore, NAK is sent automatically to the host.

[bit 9] SPK: Short Packet Interrupt Request Bit (Short PackEt)

This bit indicates that the data size transferred from the host does not satisfy the maximum packet size (including 0-byte) set by PKS in the EP0 Control Register (EP0C) when the data has been received successfully. This bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Received data size satisfies the maximum packet size
1	Received data size does not satisfy the maximum packet size

[bit 8:7] Reserved bits

These bits are reserved. The written value has no effect. They are always read as "0".

[bit 6:0] SIZE: Packet Size Indication Bit (packet SIZE)

This bit indicates the number of data bytes written to the receive buffer after EP0's OUT packet transfer has finished. The SIZE bit is updated to a valid value when the DRQO interrupt cause of the EP0O Status Register (EP0OS) has been set.

Example: 8 bytes => "0x08", 64 bytes => "0x40" (maximum value)

5.9. EP1 to 5 Status Registers (EP1S to EP5S)

The EP1 to 5 Status Registers (EP1S to EP5S) indicate the status of the Endpoints 1 to 5.

The following figure shows the bit configuration of the EP1 to 5 Status Registers (EP1S to EP5S).

■ EP1 Status Register (EP1S)

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	SIZE1
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	R
Initial value	1	0	0	X	0	0	0	X
bit	7	6	5	4	3	2	1	0
Field	SIZE1							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

■ EP2 to 5 Status Registers (EP2S to EP5S)

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	-
Initial value	1	0	0	X	0	0	0	X
bit	7	6	5	4	3	2	1	0
Field	Reserved	SIZE2 to SIZE5						
Attribute	-	R	R	R	R	R	R	R
Initial value	0	X	X	X	X	X	X	X

The following explains the function of each bit in the EP1 to 5 Control Registers (EP1S to EP5S).

[bit 15] BFINI: Send/Receive Buffer Initialization Bit (BuFFer INItial)

This bit initializes the send/receive buffer of transfer data. The BFINI bit is also automatically set by setting the RST bit of the UDC Control Register (UDCC). If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing the BFINI bit.

Bit	Description
0	Clears the initialization
1	Initializes the send/receive buffer

<Note>

The EP1 to EP5 send/receive buffer has a double-buffer configuration. The BFINI bit initialization initializes the double buffers concurrently and also initializes the DRQ and SPK bits. Before initializing the buffer, make sure that the DRQ bit is set, and check the BUSY bit to make sure that there is no access from the host, and then configure the STAL bit.

[bit 14] DRQIE: Packet Transfer Interrupt Enable Bit (Data ReQuest Interrupt Enable)

This bit enables interrupts generated by the "DRQ" interrupt cause of the EP1 to EP5 Status Register.

Bit	Description
0	Disables interrupts generated by the DRQ cause
1	Enables interrupts generated by the DRQ cause

<Note>

To use the automatic buffer transfer mode (DMAE = 1), set DMA and enables transfer before enabling the DRQIE bit.

[bit 13] SPKIE: Short Packet Interrupt Enable Bit (SPK Interrupt Enable)

This bit enables interrupts generated by the "SPK" interrupt cause of the EP1 to EP5 Status Register.

Bit	Description
0	Disables interrupts generated by the SPK cause
1	Enables interrupts generated by the SPK cause

[bit 12] Reserved bit

This bit is reserved. The written value has no effect. The read value is undefined.

[bit 11] BUSY: Busy Flag Bit (BUSY flag)

This bit indicates that the host is currently gaining write or read access to the send/receive buffer. The BUSY bit is automatically set or reset.

Bit	Description
0	No access from the host
1	Write or read access from the host is in process

<Note>

If the BUSY bit is set while the DRQ bit is set, it indicates that the host is currently accessing either of the double buffers that is not accessed by the CPU or via DMA.

Usually, control using the BUSY bit is not required. To initialize the buffer by setting BFINI, however, take the following steps previously.

1. Make sure that the DRQ bit has been set, and check the BUSY bit to make sure that there is no access from the host.
2. Set the STAL bit.

[bit 10] DRQ: Packet Transfer Interrupt Request Bit (Data ReQuest)

This bit indicates that the EP1 to EP5 packet transfer has normally ended, and processing of the data is required. The DRQ bit is an interrupt cause, and writing "1" is ignored. Clear the DRQ bit by writing "0" while it is "1". A read-modify-write access reads the bit as "1".

Bit	Description
0	Clears the interrupt cause
1	Packet transfer normally ended

<Note>

If automatic buffer transfer mode (DMAE = 1) is not used, "0" must be written to the DRQ bit after data has been written or read to/from the send/receive buffer. Switch the access buffers once the DRQ bit is cleared. That DRQ = 0 may not be read after the DRQ bit is cleared. If the transfer direction is set to IN, and the DRQ bit is cleared without writing buffer data while the DRQ bit is "1", it implies that 0-byte data is set. If DIR of the EP1 to EP5 Control Registers (EP1C to EP5C) is set to "1" at initial settings, the DRQ bit of corresponding Endpoint is set at the same time. Also while the DRQ bit is not set, "0" must not be written.

[bit 9] SPK: Short Packet Interrupt Request Bit (Short PackEt)

This bit indicates that the data size transferred from the host does not satisfy the maximum packet size (including 0-byte) set by PKS in the EP1 to EP5 Control Registers (EP1C to EP5C) when the data has been received successfully. This bit is an interrupt cause, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

Bit	Description
0	Received data size satisfies the maximum packet size
1	Received data size does not satisfy the maximum packet size

<Note>

The SPK bit is not set during data transfer in the IN direction.

[EP2 to EP5: bit 8:7] Reserved bits

In EP2 to EP5, these bits are reserved. The written value has no effect. They are always read as "0".

[(EP1: bit 8:7) bit 6:0] SIZE: packet SIZE

These bits indicate the number of data bytes written to the receive buffer when OUT packet transfer of EP1 to EP5 has finished. The SIZE bit is updated to a valid value when the DRQ interrupt cause of the EP1 to EP5 Status Registers (EP1S to EP5S) has been set.

The maximum transfer data size of Endpoints 1 to 5 is as follows:

EndPoint	Maximum transfer size	Indication range
1	256 bytes	0x000 to 0x100
2 to 5	64 bytes	0x00 to 0x40

<Note>

These bits are set to the data size transferred from the host in the OUT direction and written to the buffer. Therefore, a value read during transfer in the IN direction has no effect.

5.10. EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

The EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) control writing or reading transfer data to/from the send/receive buffer for Endpoints 0 to 5.

The following figure shows the bit configuration of the EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

■ EP0DTH to EP5DTH

bit	15	14	13	12	11	10	9	8
Field	BFDT							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

■ EP0DTL to RP5DTL

bit	7	6	5	4	3	2	1	0
Field	BFDT							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

The following explains the function of each bit in the EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

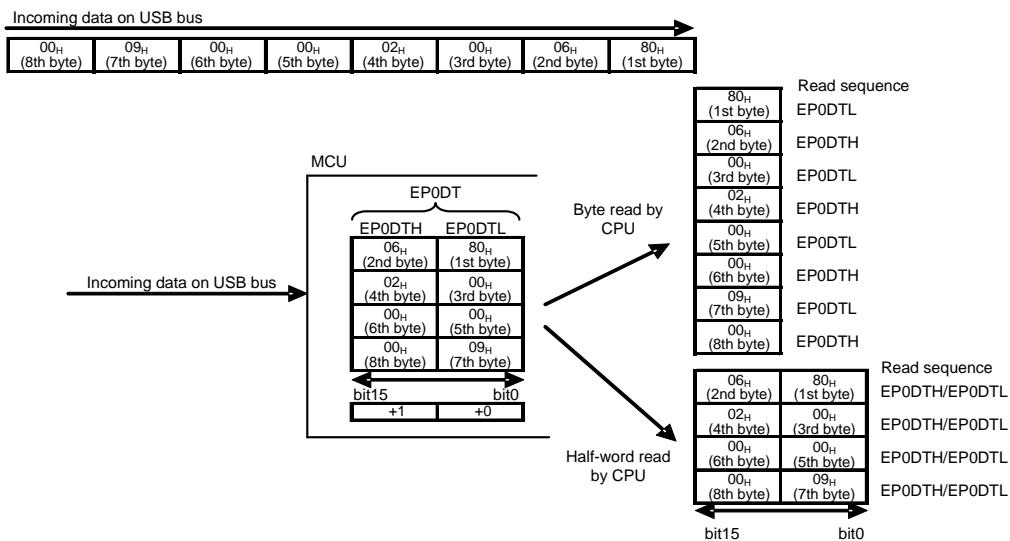
[bit 15:0] BFDT: Endpoint Send/Receive Buffer Data Bits (BuFfer DaTa)

A register used for data write/read to/from the send/received buffer for each end point.

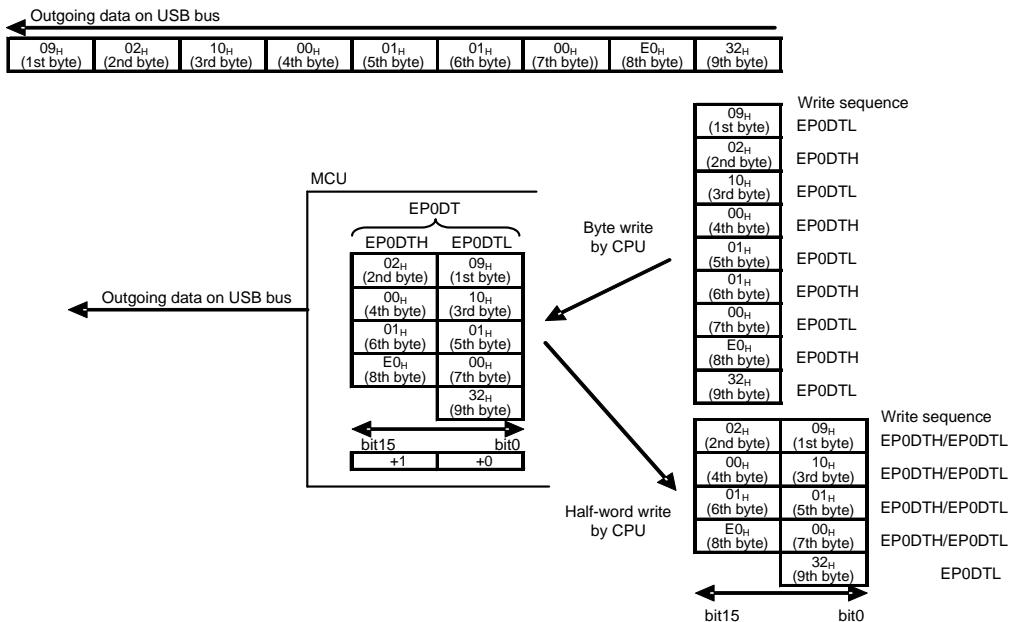
<Notes>

- The CPU can access the EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) either by the byte or by the half-word.
- Byte access
First access low-order (EPxDTL) and then high-order (EPxDTH). Subsequently, access low-order (EPxDTL) and high-order (EPxDTH) alternately.
- This register must not be accessed by the bit operation instruction.

Example: Setup transfer by the Get descriptor



Example: IN transfer by the Get descriptor



The DMA transfer can only access the EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) by the half-word. (See "Automatic data size transfer mode" of "3.6 DMA transfer function")

Chapter: USB Host

This chapter explains the functions and operations of the USB host.

1. Overview of USB host
2. USB host configuration
3. USB host operations
4. USB host setting procedure examples
5. USB host registers

1. Overview of USB host

This section explains the functions and operations of the USB host.

■ Features of USB host

The USB host has the following features:

- Automatic detection of full-speed or low-speed transfer
- Support of full-speed or low-speed transfer
- Automatic detection of device connection or disconnection
- Support of USB bus reset sending function
- Support of IN, OUT, SETUP, and SOF tokens
- Automatic sending of handshake packet for IN token (excluding STALL)
- Automatic detection of handshake packet for OUT token
- Support of maximum packet length of up to 256 bytes
- Support of actions against errors (CRC error, toggle error, and timeout)
- Support of Wake-up function
- Support of FUJITSU's original USB host functions Can also be operated as USB functions by switching the operation mode. (For restrictions in the USB host specifications, see "Table 1-1".)

<Note>

Set the base clock to 13 MHz or higher when using the USB host.

Table 1-1 Restrictions in USB host specifications

		Host
Hub support		o ^{*1}
Transfer functions	Bulk transfer	o
	Control transfer	o
	Interrupt transfer	o
	Isochronous transfer	o
Transfer speed modes	Low Speed	o
	Full Speed	o
PRE packet support		x
SOF packet support		o
Error types	CRC error	o
	Toggle error	o
	Timeout	o
	Max. packet < Received data	o
Detection of device connection or disconnection		o
Detection of transfer speed		o

o: Supported.

x: Not supported.

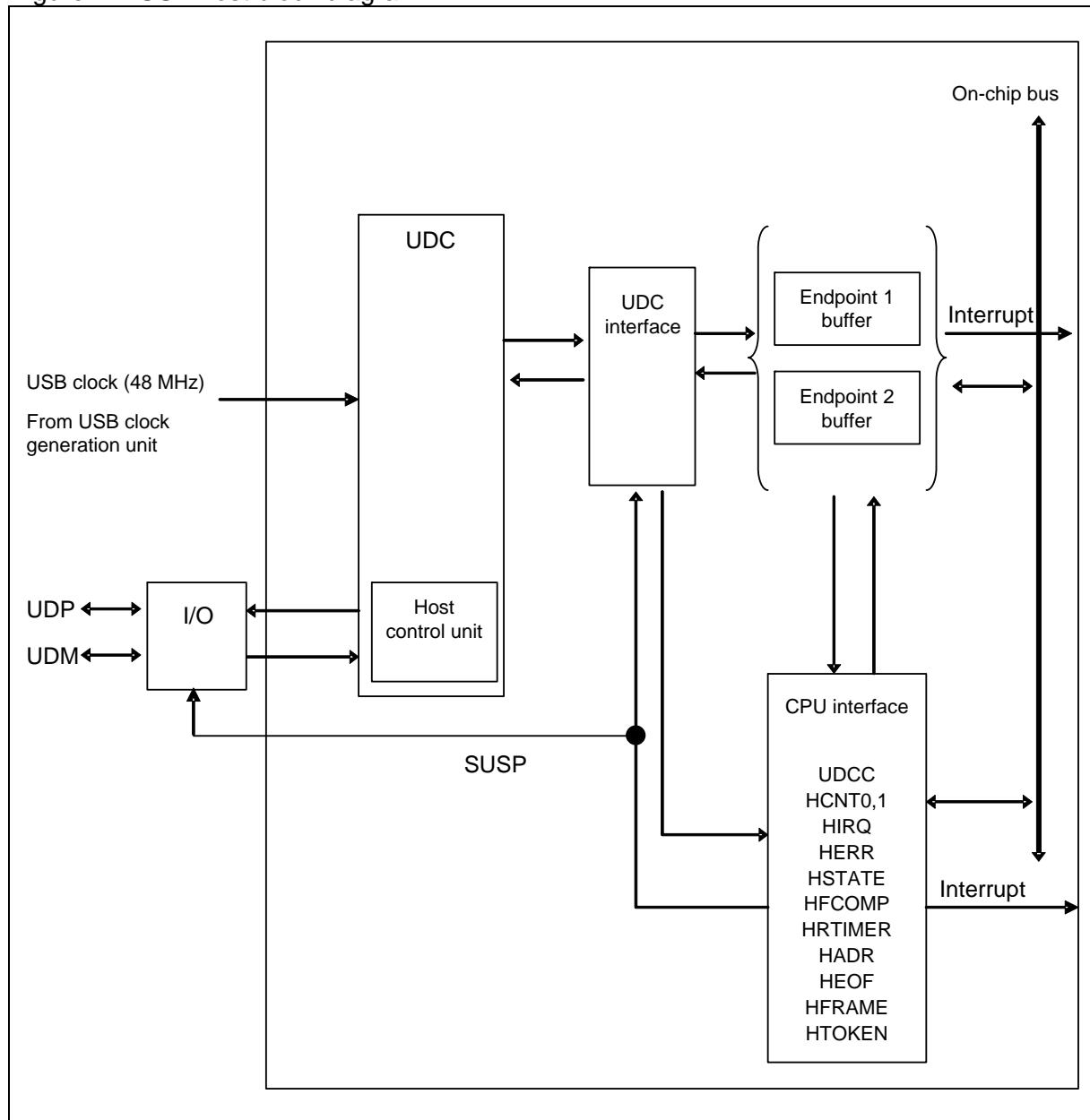
*1 : Supports a hub of up to one stage in only the full-speed mode.

2. USB host configuration

Figure 2-1 shows the USB host block diagram.

■ USB host block diagram

Figure 2-1 USB host block diagram



3. USB host operations

This section explains the operations of the USB host.

- 3.1 Device connection
- 3.2 USB bus resetting
- 3.3 Token packet
- 3.4 Data packet
- 3.5 Handshake packet
- 3.6 Retry function
- 3.7 SOF interrupt
- 3.8 Error status
- 3.9 End of packet
- 3.10 Suspend and resume operations
- 3.11 Device disconnection

3.1. Device connection

This section shows how to detect that an external USB device is connected using software.

■ Host function setting

To carry out USB operation, configure the setting of the USB clock generation unit and enable the USB clock output while the USBEN bit of the USB Enable Register (USBEN) is "0" (USB operation disabled). Next, set the USBEN bit to "1" (USB operation enabled). Then, to operate the USB as a host, set "1" to the HOST bit of Host Control Register 0 (HCNT0).

■ When an external USB device is not connected or connected

When an external USB device is not connected, both of host pins D+ and D- are set to "LOW" by the pull-down resistor. In this case, the CSTAT bit of the Host Status Register (HSTATE) is "0" and the TMODE bit is undefined. When an external USB device is connected, the CSTAT bit of the Host Status Register (HSTATE) is changed to "1".

■ Detection of external USB device connection

When a connection of an external USB device is detected, the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If "1" is set to the CNNIRE bit of Host Control Register 0 (HCNT0), a device connection interrupt occurs. To clear this interrupt, write "0" to the CNNIRQ bit of the Host Interrupt Register (HIRQ). When detecting a device connection by polling, instead of an interrupt, use the following steps to create a program.

1. Set the CNNIRE bit of Host Control Register 0 (HCNT0) to "0".
2. Check that the CNNIRQ bit of the Host Interrupt Register (HIRQ) changes to "1".

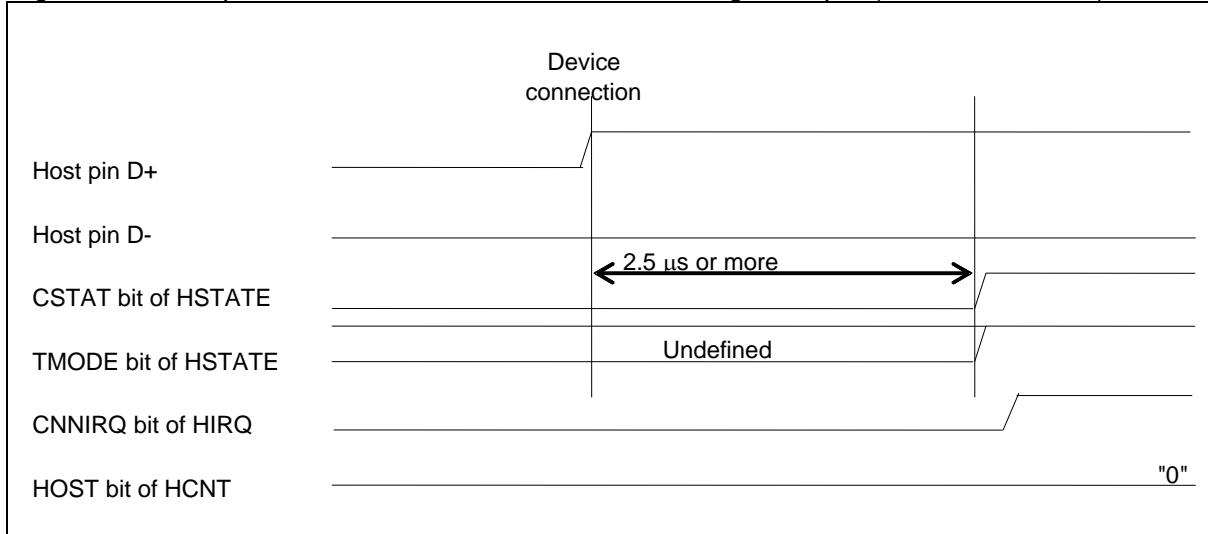
■ Obtaining the transfer speed of the remote USB device and selecting clocks

To obtain the possible transfer speed of the remote USB device after detecting a connection, check the value of the TMODE bit of the Host Status Register (HSTATE). The following shows the relationships between the transfer speed and the value of the TMODE bit of the Host Status Register (HSTATE).

- The destination is a device in the full-speed mode. -> TMODE="1"
- The destination is a device in the low-speed mode. -> TMODE="0"

If the RST bit of the UDC control register (UDCC) is "1" after obtaining the transfer speed of an external USB device, update the CLKSEL bit of the Host Status Register (HSTATE) according to the obtained transfer speed.

Figure 3-1 Full-speed device connection detection timing example (HCNT0 bit 0="0")



<Notes>

- When $2.5\mu s$ lapsed after an external USB device was connected, the CSTAT bit of the Host Status Register (HSTATE) is changed to "1".
- The TMODE and CSTAT bits of the Host Status Register (HSTATE) are updated regardless of the setting of the HOST bit of Host Control Register 0 (HCNT0). The CNNIRQ and DIRQ bits of the Host Interrupt Register (HIRQ) are set to "1" if conditions are satisfied.

3.2. USB bus resetting

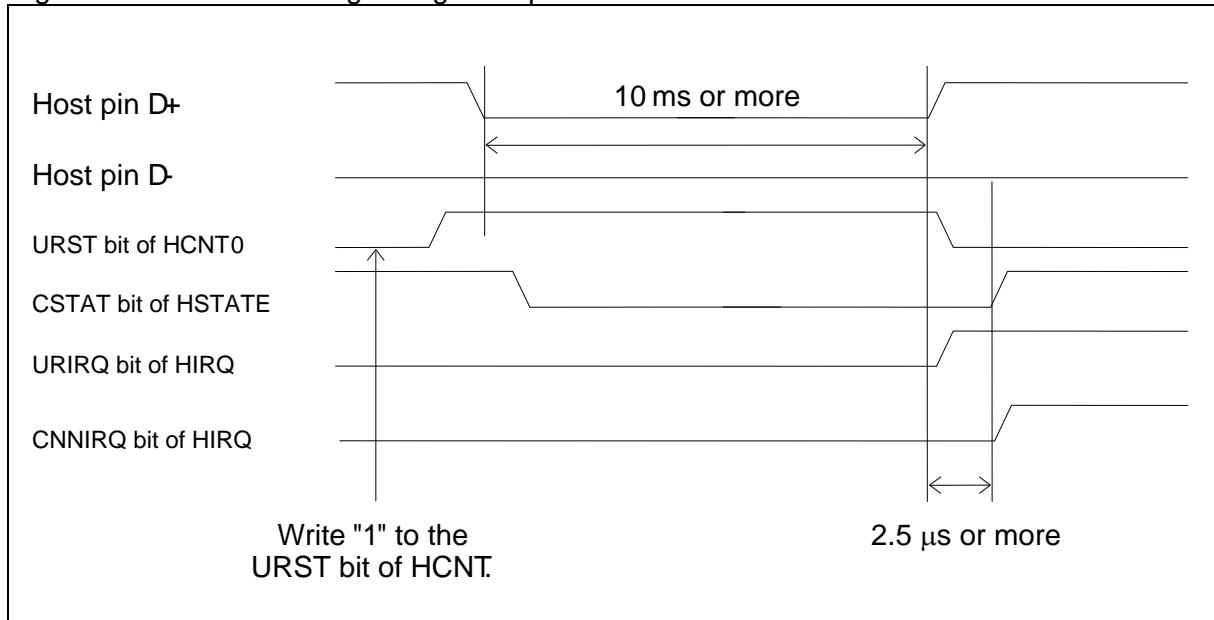
The USB bus is reset by sending SE0 for 10 ms or more if the URST bit of Host Control Register 0 (HCNT0) is set to "1" in the host mode. After USB bus resetting has been completed, the URST bit of Host Control Register 0 (HCNT0) is set to "0", and the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If the URIRE bit of Host Control Register 0 (HCNT0) is then set to "1", an interrupt occurs. To clear this interrupt, write "0" to the URIRQ bit of the Host Interrupt Register (HIRQ).

■ Notes on before and after resetting the USB bus

Note on the following points when resetting the USB bus.

1. To check that the device is connected before resetting the USB bus, make sure that the CSTAT bit of the Host Status Register (HSTATE) is set to "1".
2. Resetting the USB bus changes the CSTAT bit of the Host Status Register (HSTATE) to "0", resulting in the USB device being disconnected. At this time, the DIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".
3. After USB bus resetting has been completed, compare the value of the CLKSEL bit with that of the TMODE bit in the Host Status Register (HSTATE). If they do not match, update the CLKSEL bit to make a match. Update the CLKSEL bit when the RST bit of the UDC Control Register (UDCC) is "1".
4. After USB bus resetting has been completed, check that the USB device is connected using one of the bits shown below, and execute token processing.
 - CNNIRQ bit of Host Interrupt Register (HIRQ)
 - CSTAT bit of Host Status Register (HSTATE)

Figure 3-2 Device resetting timing example



<Note>

No token is issued if a connection of the USB device is not detected after USB bus resetting has been completed.

3.3. Token packet

When issuing an IN, OUT, or SETUP token in the host mode, use the following setting steps to send a token packet.

1. Set the Host Address Register (HADR).
2. Set the DIR and PKS bits of the EP1 Control Register (EP1C) or EP2 Control Register (EP2C).
3. Write the required data to the Host Token Endpoint Register (HTOKEN).

When issuing an SOF token, set the Frame Setup Register (HFRAME) and EOF Setup Register (HEOF), and write the required data to the Host Token Endpoint Register (HTOKEN). The setting above is not required if no change is made in the HADR, EP1C, EP2C, HFRAME, and HEOF registers.

■ Token packet setting

In the host mode, use endpoint 1 and endpoint 2 buffers to send and receive data.

When issuing an IN, OUT, or SETUP token, specify the target address in the Host Address Register (HADR). Then, specify the maximum number of bytes for each packet in the PKS bit and the transfer direction of each packet in the DIR bit of the EP1 Control Register (EP1C) or EP2 Control Register (EP2C) respectively.

If the DIR bit of the EP1 Control Register (EP1C) is "1", the endpoint 1 buffer is used as an OUT buffer. The endpoint 2 buffer is used as an IN buffer. Then set "0" to the DIR bit of the EP2 Control Register (EP2C).

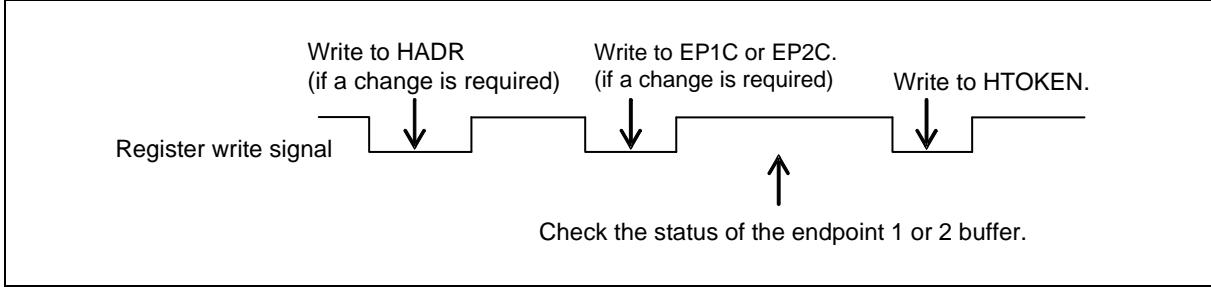
If the DIR bit of the EP1 Control Register (EP1C) is "0", the endpoint 1 buffer is used as an IN buffer. The endpoint 2 buffer is used as an OUT buffer. Then set "1" to the DIR bit of the EP2 Control Register (EP2C).

Take the following steps to execute token processing.

1. Specify the DIR and PKS bits of the EP1 Control Register (EP1C) and EP2 Control Register (EP2C).
2. If the target endpoint n (n: 1 or 2) is set to the OUT direction, write send data to the endpoint n (n: 1 or 2) buffer. Also set "0" to the DRQ bit of the EPn Status Register (EPnS: n = 1 or 2).
If the IN direction is selected, read the DRQ bit of the EPn Status Register (EPnS: n = 1 or 2), and check that its value is "0".
3. Specify the target endpoint, token, and toggle data in the Host Token Endpoint Register (HTOKEN).

The USB circuit sends a token packet in the order of Sync, token, address, endpoint, CRC5, and EOP based on the specified token; however, Sync, CRC5, and EOP are sent automatically. After one packet has been sent, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is set to "0b000" (see "3.7 SOF interrupt"). At this time, if the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. To clear this interrupt, write "0" to the CMPIRQ bit of the Host Interrupt Register (HIRQ).

Figure 3-3 Example of register setting to issue an IN, OUT, or SETUP token



When issuing an SOF token, specify the EOF time in the EOF Setup Register (HEOF) and the frame number in the Frame Setup Register (HFRAME) respectively. Then specify an SOF token code in the TKNEN bit of the Host Token Endpoint Register (HTOKEN). After this, send Sync, SOF token, frame number, CRC5, and EOP are sent, the SOFBUSY bit of the Host Status Register (HSTATE) is set to "1", and the Frame Setup Register (HFRAME) is incremented by one. The CMPIRQ bit of the Host Interrupt Register (HIRQ) is also set to "1", causing the TKNEN bit of the Host Token Endpoint Register (HTOKEN) to be cleared to "(000)b". If the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. When SOF is sent automatically, an interrupt by CMPIRQ does not occur. To clear a token completion interrupt, write "0" to the CMPIRQ bit of the Host Interrupt Register (HIRQ).

SOF is automatically sent every 1 ms while the SOFBUSY bit of the Host Status Register (HSTATE) is "1". The following shows the conditions (SOF stop conditions) to set the SOFBUSY bit of the Host Status Register (HSTATE) to "0".

- Write "0" to the SOFBUSY bit of the Host Status Register (HSTATE).
- Reset the USB bus (write "1" to the URST bit of HCNT).
- Write "1" to the SUSP bit of the Host Status Register (HSTATE).
- Disconnect the USB device (when the CSTAT bit of HSTATE is "0").

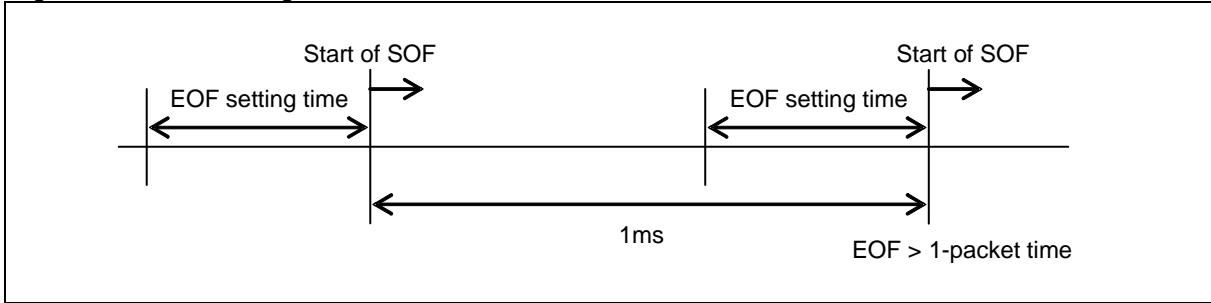
Take the following steps to change the USB from the host mode to the function mode.

1. Set "0" to the SOFBUSY bit of the Host Status Register (HSTATE).
2. Check the following conditions.
 - The SOFBUSY bit of the Host Status Register (HSTATE) is cleared to "0".
 - The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is set to "000".
 - The SUSP bit of the Host Status Register (HSTATE) is set to "0".
3. Set "1" to the RST bit of the UDC Control Register (UDCC).
4. Change the operation mode from the host mode to the function mode.

To set the SOFBUSY bit of the Host Status Register (HSTATE) to "1" again, send an SOF token once more.

The EOF Setup Register is used to prevent SOF from being sent simultaneously with other tokens. If the TKNEN bit of the Host Token Endpoint Register (HTOKEN) is written in the period from the EOF setting time to the SOF starting time, the specified token is placed into the wait state. After SOF has been sent, the token in the wait state is issued. The EOF Setup Register specifies a 1-bit time as the time unit. For example, if "(10)h" is specified in the EOF Setup Register, the time is set to $16*1/12\text{MHz}=1333.3\text{ns}$ in the full-speed mode and $16*1/1.5\text{MHz}=10666.6\text{ns}$ in the low-speed mode. When the EOF setting time is shorter than the 1-packet time, SOF may be sent doubly during execution of other token. In this case, the LSTSOF bit of the Host Error Status Register (HERR) is set to "1", and SOF is not sent. If "1" is set to the LSTSOF bit of the Host Error Status Register (HERR), the value of the EOF Setting Register must be increased (see the explanation of the EOF Setup Register).

Figure 3-4 SOF timing



3.4. Data packet

When sending a data packet after a token packet, transfer toggle data based on the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN). Further, send endpoint 1 or 2 buffer data, CRC16 data, and EOP depending on the value of the DIR bit of the EP1 Control Register (EP1C).

When receiving a data packet, compare the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN) with the received toggle data. If they match, the received data is distributed to the endpoint 1 or 2 buffer depending on the value of the DIR bit of the EP1 Control Register (EP1C) to verify whether or not a CRC16 error occurs.

■ Data packet

Take the following steps to send or receive a data packet after sending a token packet.

1. For sending
 - Automatically send Sync.
 - If the TGGL bit of the Host Token Endpoint Register (HTOKEN) is "0", send DATA0. If the TGGL bit is "1", send DATA1.
 - If the DIR bit of the EP1 Control Register (EP1C) is "1", select the endpoint 1 buffer. If the DIR bit of the EP1 Control Register (EP1C) is "0", select the endpoint 2 buffer. Then, send all the target data.
 - Send a 16-bit CRC.
 - Send a 2-bit EOP.
 - Send a 1-bit J State.
2. For receiving
 - Receive Sync.
 - Receive toggle data, and compare it with the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN).
 - If the toggle data matches the value of the TGGL bit, check the DIR bit of the EP1 Control Register (EP1C). If the DIR bit is "1", select the endpoint 2 buffer. If the DIR bit of the EP1 Control Register (EP1C) is "0", select the endpoint 1 buffer. Then, distribute the received data to the respective buffers.
 - Verify the 16-bit CRC when EOF is received.

When the HOST bit of Host Control Register 0 (HCNT0) is "1", set the inverted value to the respective DIR bits of the EP1 Control Register (EP1C) and EP2 Control Register (EP2C). For example, if "0" is set to the DIR bit of the EP1 Control Register (EP1C), set "1" to the DIR bit of the EP2 Control Register (EP2C).

3.5. Handshake packet

A handshake packet is used to notify the remote device of the status of the local device.

■ Handshake packet

A handshake packet sends either one of ACK, NAK, and STALL from the receiving side when it is judged that the receiving side is ready to receive data normally. If the USB circuit receives a handshake packet, the type of the received handshake packet is set to the HS bit of the Host Error Status Register (HERR). If the USB circuit sends a handshake packet, the type of the sent handshake packet is set to the HS bit of the Host Error Status Register (HERR).

3.6. Retry function

When a NAK or CRC error occurs at the end of a packet, if "1" is set to the RETRY bit of Host Control Register 1 (HCNT1), processing is retried repeatedly for the period specified in the Retry Timer Setting Register (HRTIMER).

■ Retry function

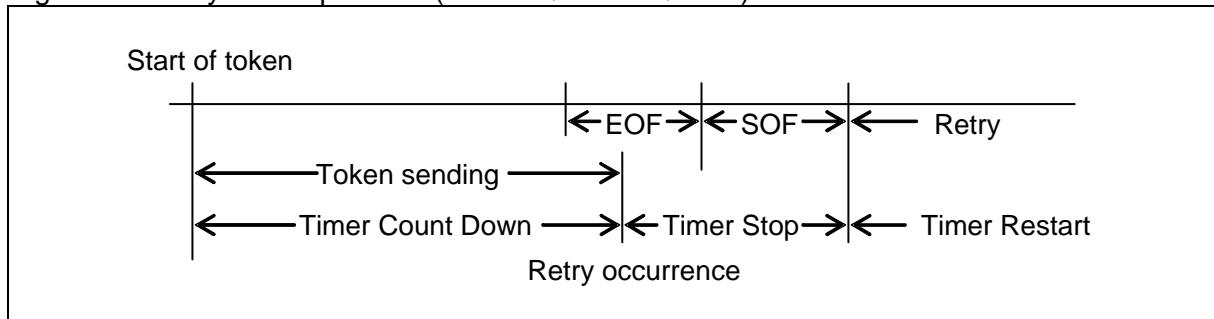
When an error* other than STALL or device disconnection occurs, the target token is retried if the RETRY bit of Host Control Register 1 (HCNT1) is "1". The following shows the conditions to end retry processing.

* : HERR.HS="01", HERR.RERR="1", HERR.TOUT="1", HERR.TOUT="1", HERR.CRC="1",
HERR.STUFF="1"

- The RETRY bit of Host Control Register 1 (HCNT1) is set to "0".
- "0" is detected in the retry timer.
- The interrupt flag is generated by SOF (SOFIRQ of HIRQ = "1").
- ACK is detected.
- A device disconnection is detected.

The retry timer is activated at start of a token, and counted down by a 1-bit transfer clock. If retry occurs in the EOF area, counting stops. If a SOF token is ended while the SOFIRQ bit of HIRQ is "0", counting restarts from the timer value that is set when counting stopped. When the retry timer is set to "0", a packet is ended, and the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Figure 3-5 Retry timer operation (SOFIRQ of HIRQ = "0")



When retry processing is ended, end information of the EOP is set to each register.

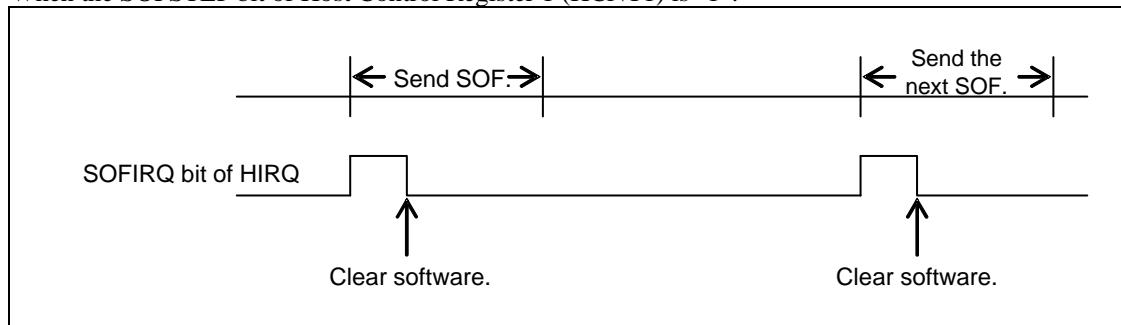
3.7. SOF interrupt

The SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1" at start of SOF depending on the setting of the SOFSTEP bit of Host Control Register 1 (HCNT1) and SOF Interrupt Frame Compare Register (HFCOMP). If the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. When SOF processing is executed using the Host Token Endpoint Register (HTOKEN), the SOFIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".

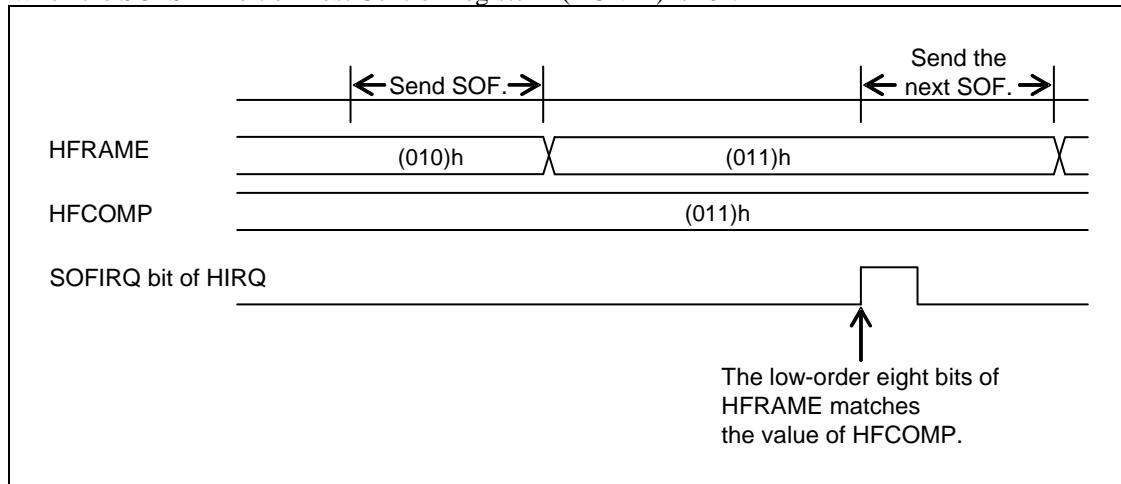
■ SOF interrupt

When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "0", the value of the SOF Interrupt Frame Compare Register (HFCOMP) is compared with the low-order eight bits of the frame number for SOF token. If they match, "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ) when sending SOF. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "1", "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ) each time SOF is sent.

1. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "1":



2. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "0":



If "1" is set to the CANSEL bit of Host Control Register 1 (HCNT1), the target token is not sent when it is set at the following timing.

- A token other than SOF is set to the Host Token Endpoint Register (HTOKEN) in the EOF area.
- If a token is set at this timing, the following operations are carried out.
- If the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1" when the next SOF is sent, the TKNEN bit of the Host Token Endpoint Register (HTOKEN) is cleared to "0b000". In this case, that token is not sent.

The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is cleared at the following timing.

At this timing, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1". When the SOFIRQ bit is set to "1", the TCAN bit of the Host Interrupt Register (HIRQ) indicates that a token is canceled. When retrying to send a token, write "0" to the TCAN bit of the Host Interrupt Register (HIRQ). Then write a token to be sent to the TKNEN bit of the Host Token Endpoint Register (HTOKEN).

If "0" is set to the CANCEL bit of Host Control Register 1 (HCNT1), the token specified in the Host Token Endpoint Register (HTOKEN) is sent following SOF.

Figure 3-6 Token cancellation example at CANCEL bit of HCNT1 = "1"

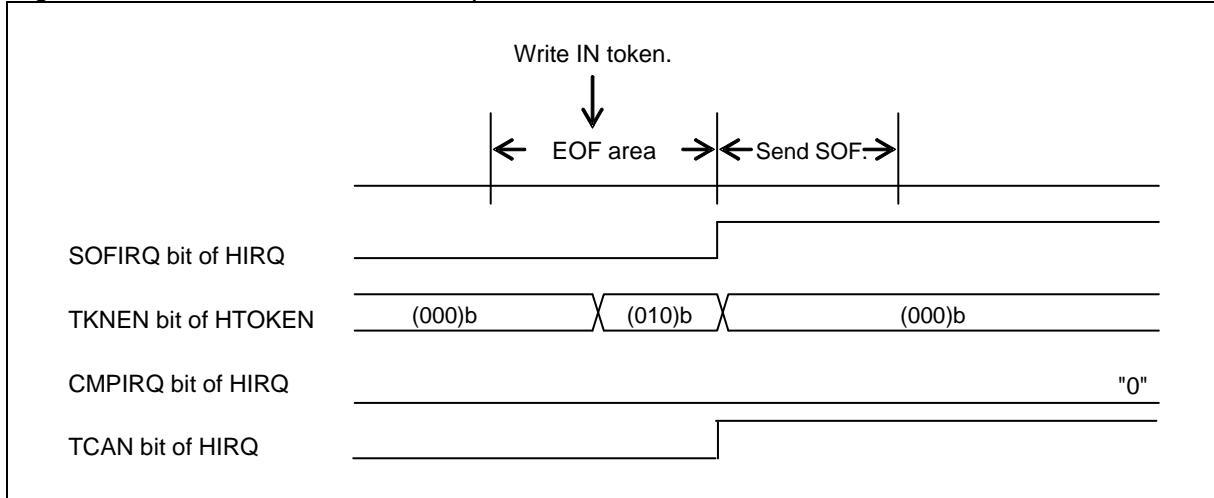
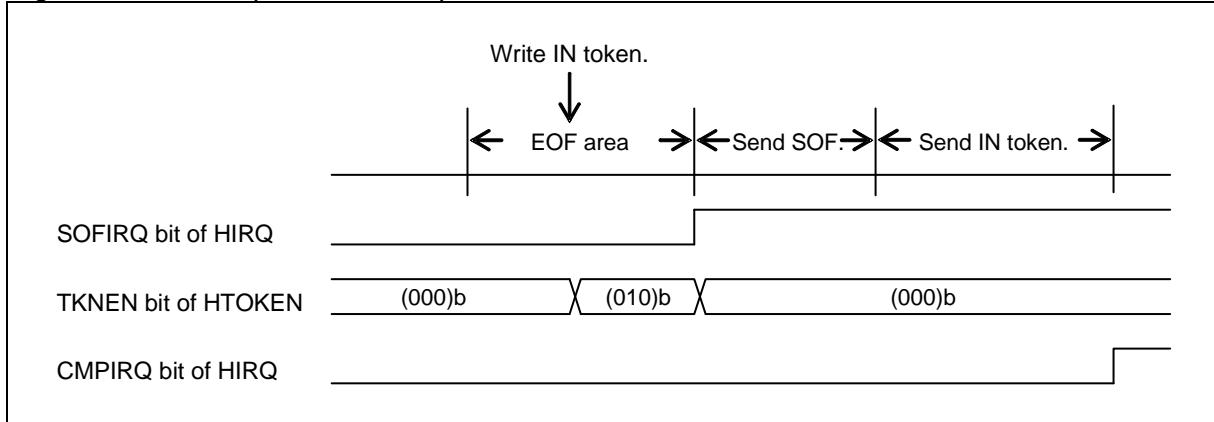


Figure 3-7 Token operation example at CANCEL bit of HCNT1 = "0"



3.8. Error status

The USB host supports error information.

■ Error status

1. Stuffing Error

If "1" is successively set to six bits, "0" is inserted into one bit. If "1" is successively detected in seven bits, it is judged to be Stuffing Error, and the STUFF bit of the Host Error Status Register (HERR) is set to "1". To clear this status, write "0" to the STUFF bit. If the next token is sent without clearing the STUFF bit, a cause is reflected on the STUFF bit when the next token is ended.

2. Toggle Error

When sending an IN token, the toggle data of a data packet is compared with the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN). If they do not match, the TGERR bit of the Host Error Register (HERR) is set to "1". To clear the TGERR bit, write "0" to the TGERR bit of the Host Error Register (HERR). If the next token is sent without clearing the TGERR bit, a cause is reflected on the TGERR bit when the next token is ended.

3. CRC Error

When receiving an IN token, data and CRC of the received data packet are obtained with the CRC polynomial " $G(X) = X^{16} + X^{15} + X^2 + 1$ ". If the remainder is not "(800d)h", it means that CRC Error occurs, and the CRC bit of the Host Error Register (HERR) is set to "1". To clear the CRC bit, write "0" to the CRC bit of the Host Error Register (HERR). If the next token is sent without clearing the CRC bit, a cause is reflected on the CRC bit when the next token is ended.

4. Time Out Error

"1" is set to the TOUT bit of the Host Error Status Register (HERR) when:

- A data packet or handshake packet has not been input in the specified time;
- SE0 has been detected during data receiving; or
- Stuffing Error has been detected.

To clear the TOUT bit, write "0" to the TOUT bit of the Host Error Register (HERR). If the next token is sent without clearing the TOUT bit, a cause is reflected on the TOUT bit when the next token is ended.

5. Receive Error

If EP1 is used as a receive buffer, the value of the PKS bit of the EP1 Control Register (EP1C) is used as the receive packet size. If EP2 is used as a receive buffer, the value of the PKS bit of the EP2 Control Register (EP2C) is used as the receive packet size. When the received data exceeds the specified receive packet size, the RERR bit of the Host Error Status Register (HERR) is set to "1". To clear the RERR bit, write "0" to the RERR bit of the Host Error Register (HERR). If the next token is sent without clearing the RERR bit, a cause is reflected on the RERR bit when the next token is ended.

3.9. End of packet

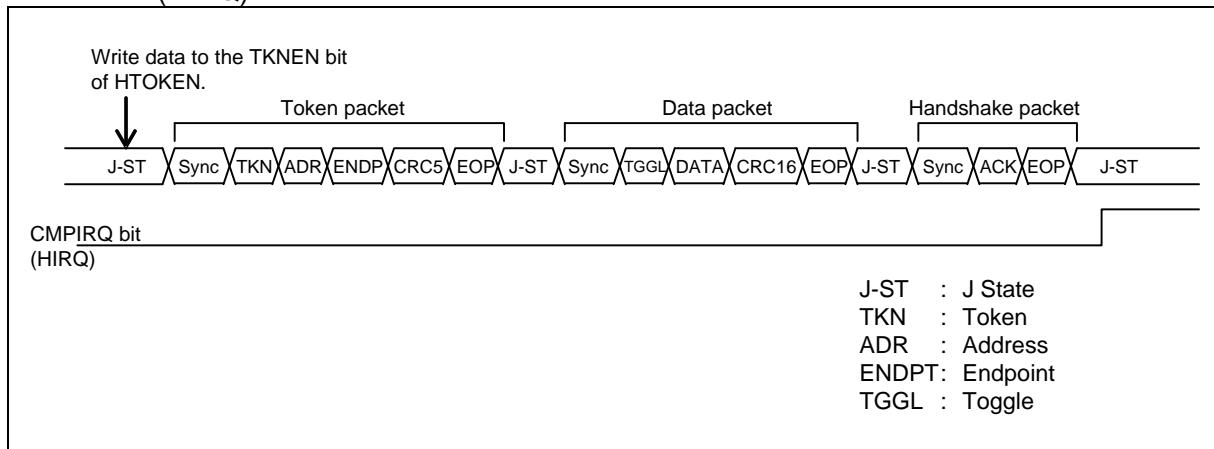
If one packet is ended in the USB host, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". At this time, if the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

■ Packet end timing

When one packet ends, the interrupt flag is generated when:

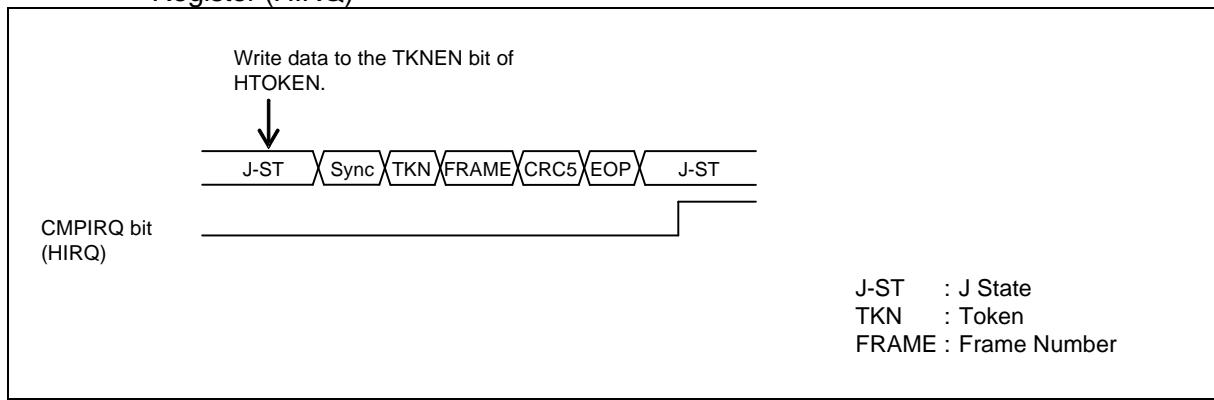
- The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is "(001)b", "(010)b", or "(011)b" (SETUP token, IN token, or OUT token).

Figure 3-8 Timing example 1 when setting the CMPIRQ bit of the Host Interrupt Register (HIRQ)



- The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is "(100)b" (SOF token).

Figure 3-9 Timing example 2 (SOF token) when setting the CMPIRQ bit of the Host Interrupt Register (HIRQ)



3.10. Suspend and resume operations

The USB host supports suspend and resume operations.

■ Suspend operation

If "1" is set to the SUSP bit of the Host Status Register (HSTATE), the procedure below is performed, and the USB circuit is placed into the suspend state.

- The USB bus is placed in the high-impedance state.
- A circuit block with no clock required is stopped.

If the USB circuit is placed in the suspend state, the SUSP bit of the Host Status Register (HSTATE) is set to "1".

However, the following operations are prohibited while resetting the USB bus.

- "1" is set to the SOFBUSY bit of the Host Status Register (HSTATE) or the USB circuit is placed into the suspend state during data transfer.
- Clocks supplied to the USB are stopped in the suspend state.

Take the following steps to stop clocks.

1. Change to the stop or timer mode.
2. Set the UCEN bit of the USB Clock Setup Register (UCCR) to "0".

■ Resume operation

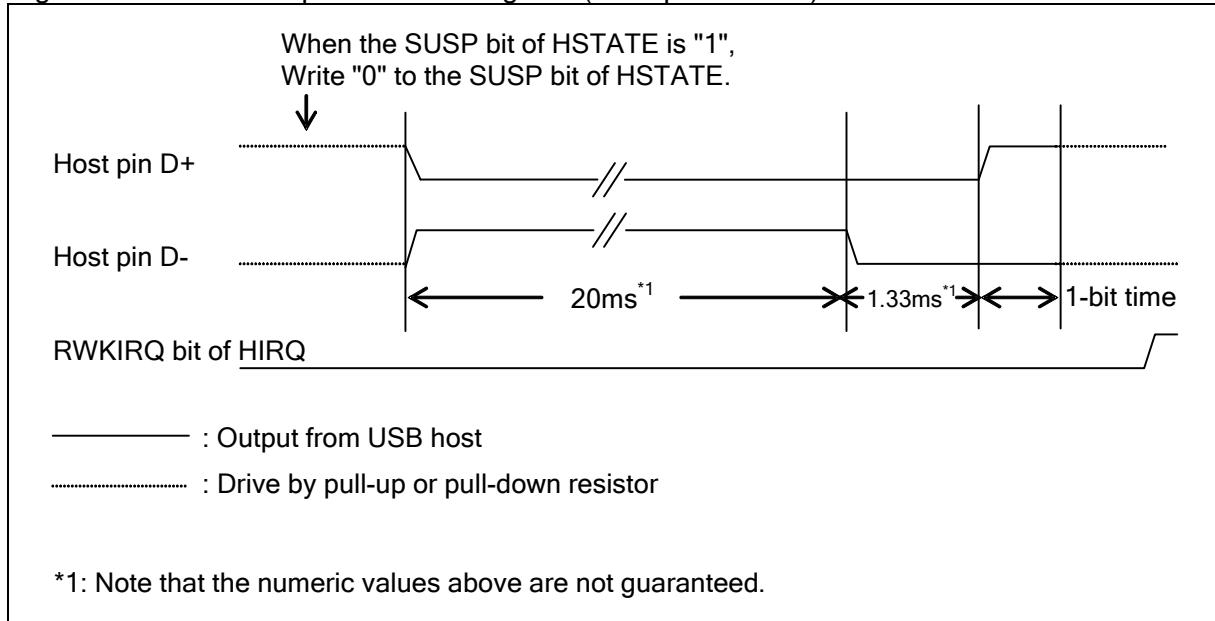
The USB bus changes from the suspend state to the resume state to resume processing when one of the following conditions is satisfied.

- "0" is set to the SUSP bit of the Host Status Register (HSTATE).
- The host pin D+ or D- is placed in the K-state mode.
- A device disconnection is detected.
- A device connection is detected.

After the RWKIRQ bit of the Host Interrupt Register (HRQ) has been set to "1", a token can be issued. The following shows the operation timing for each condition.

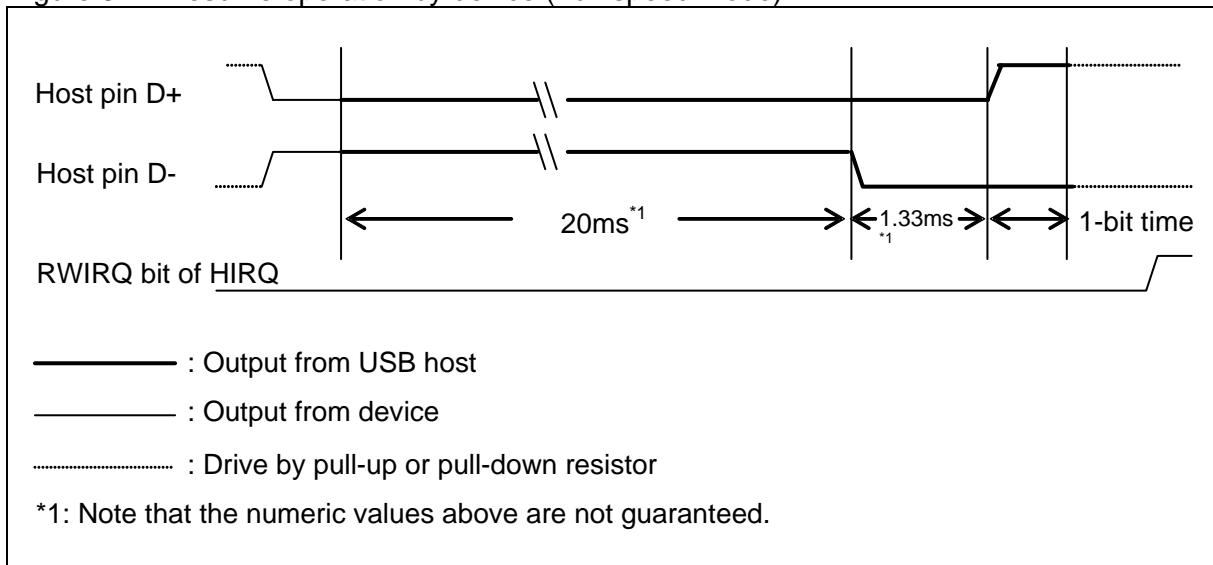
- "0" is set to the SUSP bit of the Host Status Register (HSTATE).

Figure 3-10 Resume operation with register (Full-speed mode)



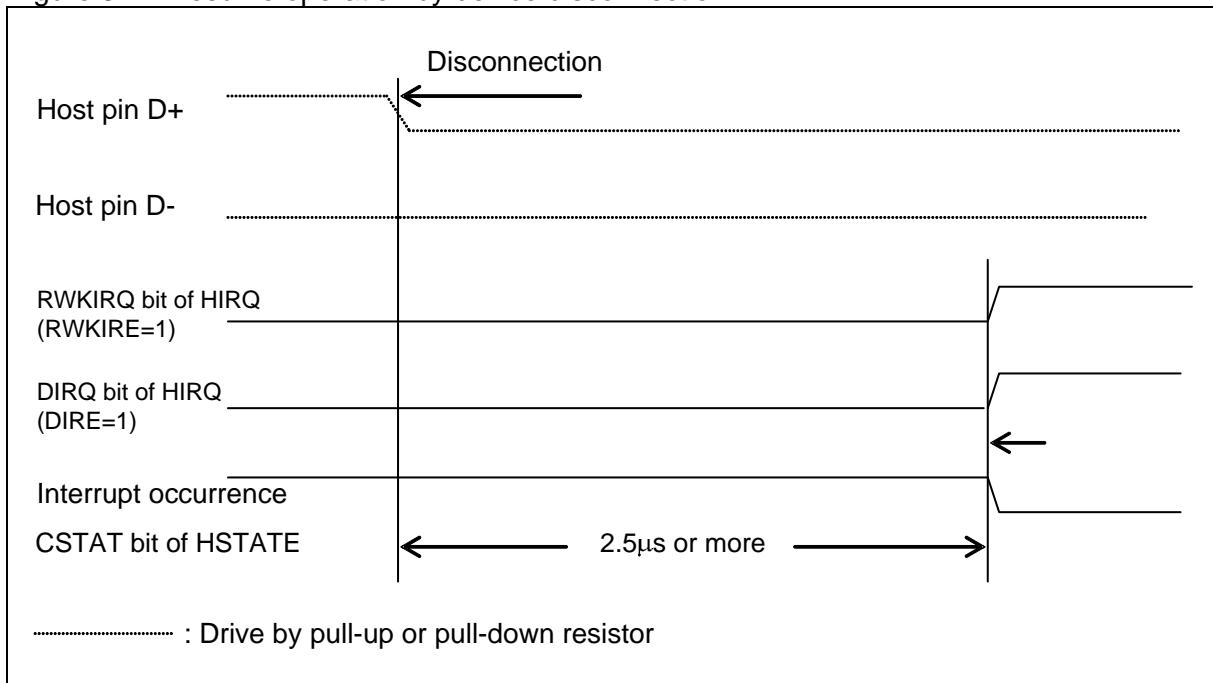
- The host pin D+ or D- is placed in the K-state mode.

Figure 3-11 Resume operation by device (Full-speed mode)



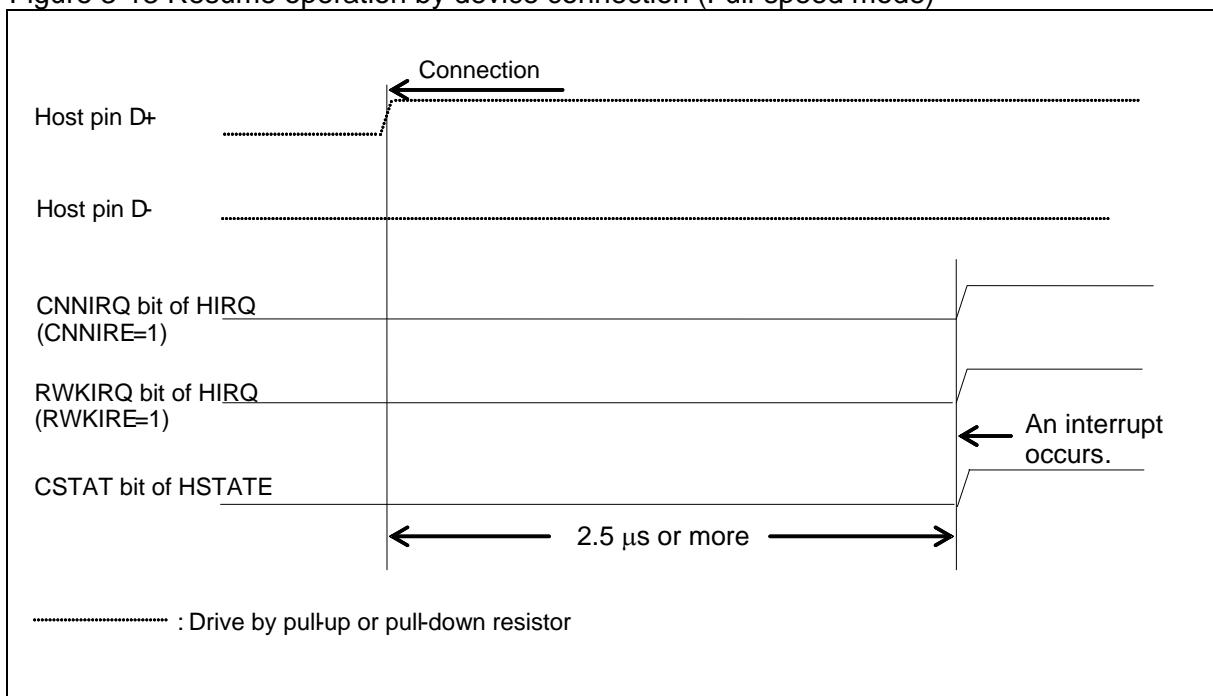
- A device disconnection is detected.

Figure 3-12 Resume operation by device disconnection



- A device connection is detected.

Figure 3-13 Resume operation by device connection (Full-speed mode)



3.11. Device disconnection

The device disconnection timer starts when both the host pins D+ and D- are set to "LOW". If "LOW" is detected for 2.5μs or more, the CSTAT bit of the Host Status Register (HSTATE) is set to "0".

■ Device disconnection

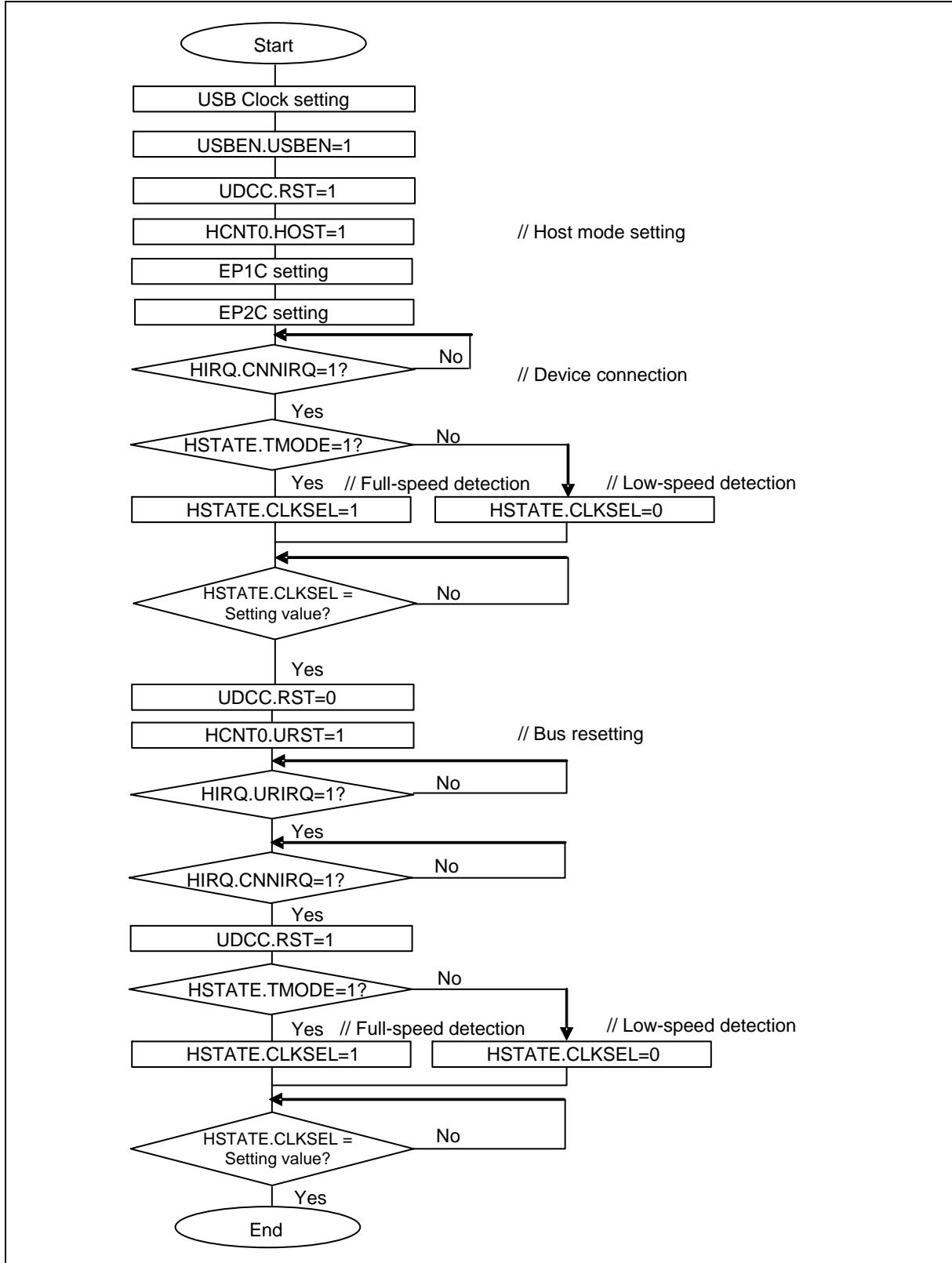
If both the host pins D+ and D- remain set to "LOW" for 2.5μs or more regardless of the host or function mode, it is judged that the device has been disconnected. This then sets "0" to the CSTAT bit of the Host Status Register (HSTATE) and "1" to the DIRQ bit of the Host Interrupt Register (HIRQ). At this time, if the DIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. To clear this interrupt, write "0" to the DIRQ bit of the Host Interrupt Register (HIRQ).

If the USB bus is reset, it is judged that the device has been disconnected. In this case, the CSTAT bit of the Host Status Register (HSTATE) is set to "0", but the DIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".

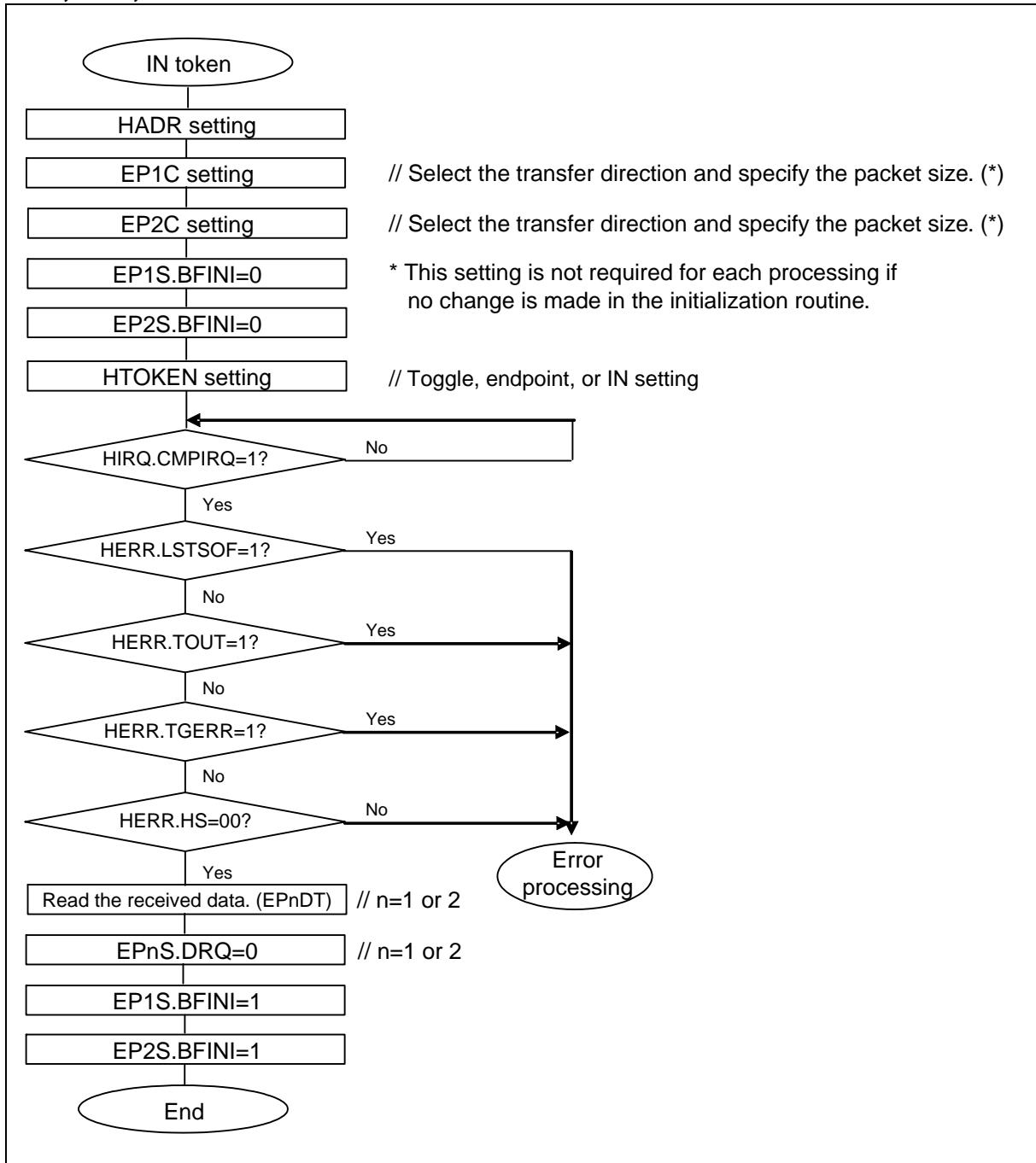
4. USB host setting procedure examples

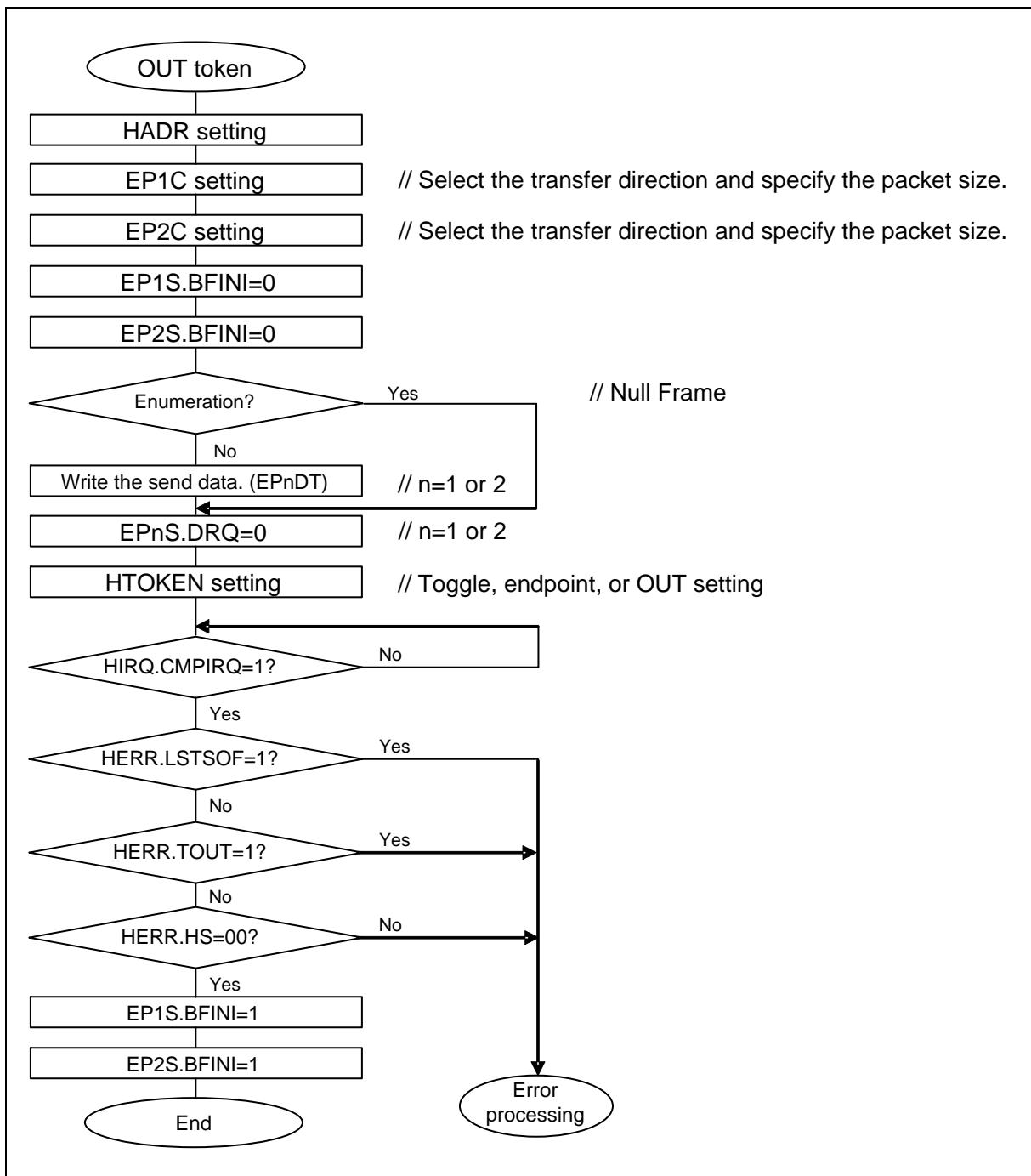
The following shows the flowchart for each token in the USB host.

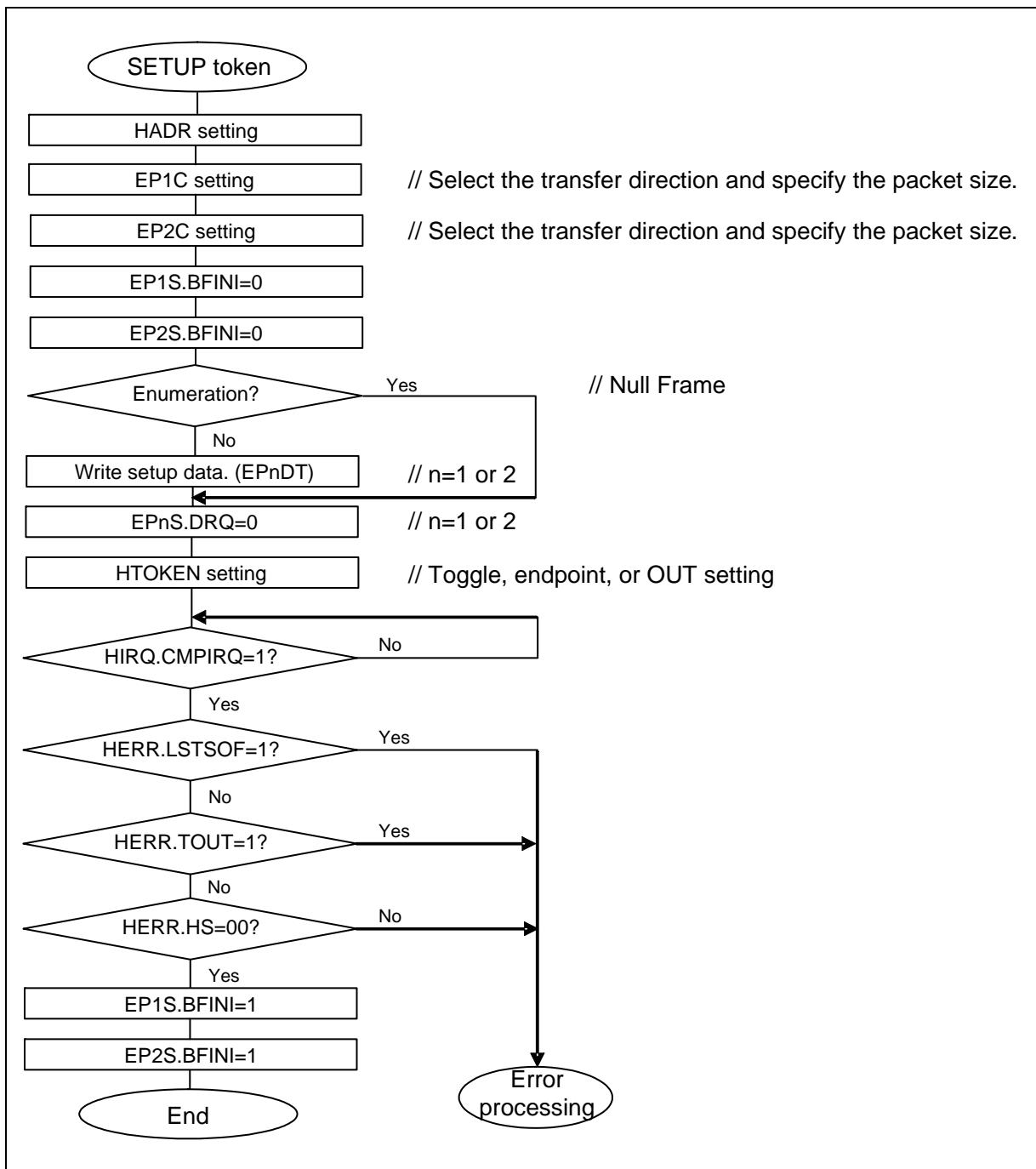
■ Initialization and device detection



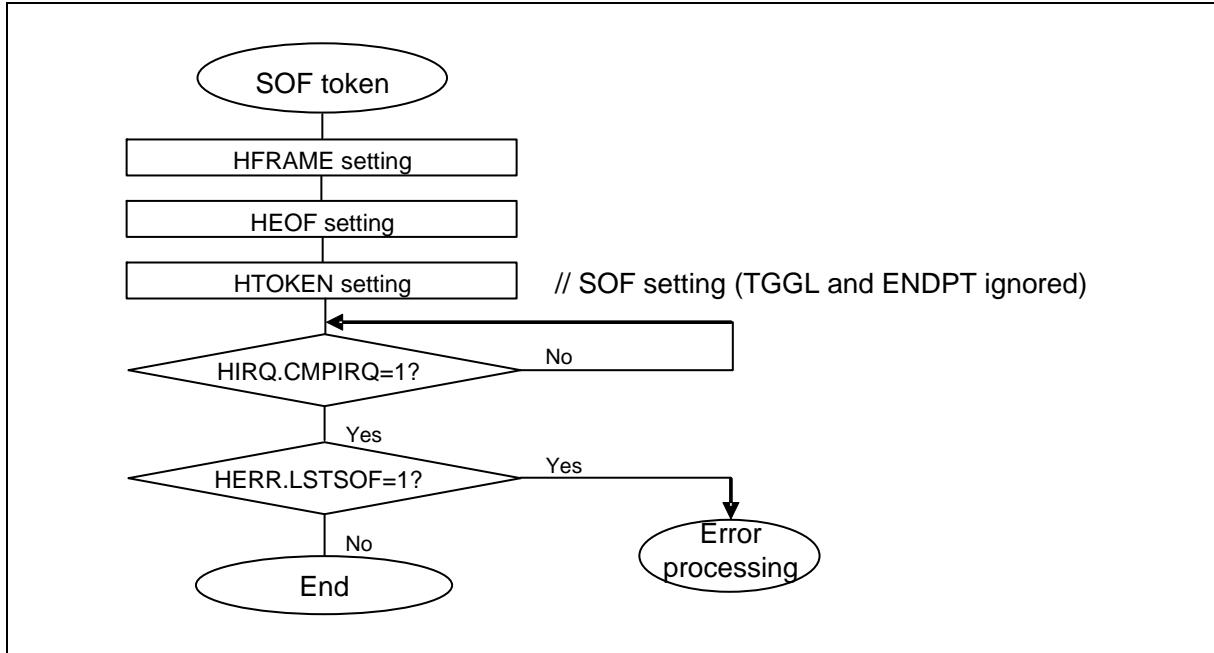
■ IN, OUT, or SETUP token







■ SOF token



5. USB host registers

This section explains the configuration and functions of the registers used for the USB host.

■ List of USB host registers

Abbreviation	Register name	See
UDCC	UDC Control Register	*
EP1C	EP1 Control Register	*
EP2C	EP2 Control Register	*
EP1S	EP1 Status Register	*
EP2S	EP2 Status Register	*
EP1DTH	EP0 Data Register high-order	*
EP1DTL	EP0 Data Register low-order	*
EP2DTH	EP0 Data Register high-order	*
EP2DTL	EP0 Data Register low-order	*
HCNT0	Host Control Register 0	5.1
HCNT1	Host Control Register 1	5.1
HIRQ	Host Interrupt Register	5.2
HERR	Host Error Status Register	5.3
HSTATE	Host Status Register	5.4
HFCOMP	SOF Interrupt Frame Compare Register	5.5
HRTIMER	Retry Timer Setup Register	5.6
HADR	Host Address Register	5.7
HEOF	EOF Setup Register	5.8
HFRAME	Frame Setup Register	5.9
HTOKEN	Host Token Endpoint Register	5.10

* : See "USB Functions".

■ UDCC.RST dependent register bit update timing list

	Register	Bit
Register bits to be updated when UDCC.RST=1	HCNT0	HOST
	HSTATE	CLKSEL
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
Register bits initialized when UDCC.RST=1 (Update when UDCC.RST=0)	HCNT0	URST
	HIRQ	TCAN, RWKIRQ, URIRQ, CMPIRQ, CNNIRQ, DIRQ, SOFIRQ
	HERR (All bits)	LSTSOF, RERR, TOUT, CRC, TGERR, STUFF, HS
	HSTATE	SOFBUSY, SUSP
	HFRAME	FRAME0, FRAME1
	HTOKEN (All bits)	TGGL, TKNEN, ENDPT
	EP1S	BFINI, DRQ, SPK
Register bits unaffected by UDCC.RST	EP2S	BFINI, DRQ, SPK
	HCNT0	RWKIRE, URIRE, CMPIRE, CNNIRE, DIRE, SOFIRE
	HCNT1	SOFSTEP, CANSEL, RETRY
	HIRQ	CNNIRQ, DIRQ
	HFCOMP	HFRAMECOMP
	HSTATE	TMODE, CSTAT
	HRTIMERO, 1, 2	RTIMER0, 1, 2
	HADR	Address
	HEOF	EOF0, 1

5.1. Host Control Registers 0 and 1 (HCNT0 and HCNT1)

Host Control Registers 0 and 1 (HCNT0 and HCNT1) are used to specify the USB operation mode and interrupt.

■ Host Control Register 1 (HCNT1)

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	SOFSTEP	CANCEL	RETRY
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1
Reset enabled or not*	x	x	x	x	x	x	x	x

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

■ Host Control Register 0 (HCNT0)

bit	7	6	5	4	3	2	1	0
Field	RWKIRE	URIRE	CMPIRE	CNNIRE	DIRE	SOFIRE	URST	HOST
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Reset enabled or not*	x	x	x	x	x	x	x	x

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 15:11] Reserved bits

These are reserved bits. Always set it to "0".

[bit 10] SOFSTEP (SOF STEP)

This is a SOF interrupt occurrence selection bit

If this bit is set to "1", the SOF interrupt flag (HIRQ.SOFIRQ) is set to "1" each time SOF is sent.

If this bit is set to "0", the set value of the SOF Interrupt Frame Compare Register (HFCOMP) is compared with the low-order eight bits of the SOF frame number. If they match, the SOF interrupt flag (HIRQ.SOFIRQ) is set to "1".

Bit	Description
0	An interrupt occurred due to the HFCOMP setting.
1	An interrupt occurred.

<Notes>

- If a SOF token (TKNEN="001") is sent by the setting of the Host Token Endpoint Register (HTOKEN), the SOF interrupt flag (HIRQ.SOFIRQ) is not set to "1" regardless of the setting of this bit.
- This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 9] CANCEL (token CANCEL enable)

This is a token cancellation enable bit.

When "1" is set to this bit, if the target token is written to the Host Token Endpoint Register (HTOKEN) in the EOF area (specified in the EOF Setting Register), its sending is canceled. When "0" is set to this bit, token sending is not canceled even if the target token is written to the register. The cancellation of token sending is detected by reading the TCAN bit of the Host Interrupt Register (HIRQ).

Bit	Description
0	Continues a token.
1	Cancels a token.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 8] RETRY (RETRY enable)

this is a retry enable bit.

If this bit is set to "1", the target token is retried if a NAK or error* occurs. Retry processing is performed during the time that is specified in the Retry Timer Setting Register (HRTIMER).

* : HERR.RERR="1", HERR.TOUT="1", HERR.CRC="1", HERR.TGERR="1", HERR.STUFF="1"

Bit	Description
0	Does not retry token sending.
1	Retries token sending.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 7] RWKIRE (Remove WaKe up Interrupt Request Enable)

This is a resume interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the RWKIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the RWIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt after restarting.
1	Enables an interrupt after restarting.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 6] URIRE (Usb bus Rest Interrupt Request Enable)

This is a bus reset interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt after resetting the USB bus.
1	Enables an interrupt after resetting the USB bus.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 5] CMPIRE (CoMPletion Interrupt Request Enable)

This is a token completion interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt at completion.
1	Enables an interrupt at completion.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 4] CNNIRE (CoNNection Interrupt Request Enable)

This is a device connection detection interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt at device connection.
1	Enables an interrupt at device connection.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 3] DIRE (Disconnection Interrupt Request Enable)

This is a device disconnection detection interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the DIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the DIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt at device disconnection.
1	Enables an interrupt at device disconnection.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 2] SOFIRE (Start Of Frame Interrupt Request Enable)

This is a SOF interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Bit	Description
0	Disables an interrupt when sending SOF.
1	Enables an interrupt when sending SOF.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 1] URST (Usb bus ReSeT)

This is a bus reset bit.

When "1" is set to this bit, the USB bus is reset. This bit continues set to "1" during USB bus resetting, and changes to "0" when USB bus resetting is ended. If "0" is set to this bit, no processing is performed.

Bit	Description
0	Holds the status of the USB bus.
1	Resets the USB bus.

<Notes>

- No processing is performed even if this bit is set to "1" while the RST bit of the UDC Control Register (UDCC) is "1".
- This bit cannot be set to "1" while the SUSP bit of the Host Status Register (HSTATE) is "1" or during token sending.
- The Host Control Register (HCNT0 or HCNT1) cannot be written while this bit is "1".

[bit 0] HOST (HOST mode)

This is a host mode bit.

When "1" is set to this bit, the USB acts as a host. When "0" is set to this bit, the USB acts as a function.

Bit	Description
0	Function mode
1	Host mode

<Notes>

- This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).
- Change the value of this bit while the RST bit of the UDC Control Register (UDCC) is "1".
- The operation mode does not transition to the required one immediately after it was changed using this bit. Read this bit to check that the operation mode has changed.
- Before changing from the host mode to the function mode, check that the following conditions are satisfied and also set "1" to the RST bit of the UDC Control Register (UDCC).
 - The SOFBUSY bit of the Host Status Register (HSTATE) is set to "0".
 - The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is set to "000".
 - The SUSP bit of the Host Status Register (HSTATE) is set to "0".
- Before changing from the function mode to the host mode, set "1" to the HCONX bit of the UDC Control Register (UDCC), and disconnect the host or hub.

5.2. Host Interrupt Register (HIRQ)

The Host Interrupt Register (HIRQ) indicates the USB host interrupt request flag. A host interrupt can occur by setting the interrupt enable bit of the Host Control Register (HCNT0 or HCNT1), excluding the TCAN bit.

bit	7	6	5	4	3	2	1	0
Field	TCAN	Reserved	RWKIRQ	URIRQ	CMPIRQ	CNNIRQ	DIRQ	SOFIRQ
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Reset enabled or not*						x	x	

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 7] TCAN (Token CANcel flag)

This is a token cancellation flag.

If this bit is set to "1", it means that token sending is canceled based on the setting of the CANCEL bit of Host Control Register 1 (HCNT1). When this bit is "0", it means that token sending is not canceled. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	Does not cancel token sending.
1	Cancels token sending.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- No interrupt occurs even if this bit is set. To carry out interrupt processing, check that token sending is canceled during SOF interrupt processing.

[bit 6] Reserved bit

This is a reserved bit. Always set it to "0".

[bit 5] RWKIRQ (Remove WaKe up Interrupt ReQuest)

This is a remote Wake-up end flag.

If this bit is set to "1", it means that remote Wake-up is ended. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the RWKIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Issues no interrupt request by restart.
1	Issues an interrupt request by restart.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 4] URIRQ (Usb bus Reset Interrupt ReQuest)

This is a bus reset end flag.

If this bit is set to "1", it means that USB bus resetting is ended. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the URIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Issues no interrupt request by USB bus resetting.
1	Issues an interrupt request by USB bus resetting.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 3] CMPIRQ (CoMPIletion Interrupt ReQuest)

This is a token completion flag.

If this bit is set to "1", it means that a token is completed. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Issues no interrupt request by token completion.
1	Issues an interrupt request by token completion.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- This bit is not set to "1" even if the TCAN bit of the Host Interrupt Register (HIRQ) changes to "1".

[bit 2] CNNIRQ (CoNNection Interrupt ReQuest)

This is a device connection detection flag.

If this bit is set to "1", it means that a device connection is detected. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the CNNIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Issues no interrupt request by detecting a device connection.
1	Issues an interrupt request by detecting a device connection.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- A device connection is also detected in the function mode.

[bit 1] DIRQ (Disconnection Interrupt ReQuest)

This is a device disconnection detection flag.

If this bit is set to "1", it means that a device disconnection is detected. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the DIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Issues no interrupt request by detecting a device disconnection.
1	Issues an interrupt request by detecting a device disconnection.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- A device disconnection is also detected in the function mode.

[bit 0] SOFIRQ (Start Of Frame Interrupt ReQuest)

This is a SOF starting flag.

If this bit is set to "1", it means that SOF token sending is started. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

Bit	Description
0	Does not issue an interrupt request by starting a SOF token.
1	Issues an interrupt request by starting a SOF token.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

5.3. Host Error Status Register (HERR)

The Host Error Status Register (HERR) indicates whether or not an error occurs while sending or receiving data in the host mode.

bit	15	14	13	12	11	10	9	8
Field	LSTSOF	RERR	TOUT	CRC	TGERR	STUFF		HS
Attribute	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Initial value	0	0	0	0	0	0		11
Reset enabled or not*								

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. . : To be reset.

[bit 15] LSTSOF (LoST SOF)

This is a lost SOF flag.

If this bit is set to "1", it means that the SOF token cannot be sent in the host mode because other token is in process. When this bit is "0", it means that no lost SOF error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	Sends SOF.
1	SOF sending error

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 14] RERR (Receive Error)

This is a receive error flag.

When this bit is set to "1", it means that the received data exceeds the specified maximum number of packets in the host mode. If a receive error is detected, bit 5 (Timeout) of this register is also set to "1". When this bit is "0", it means that no error occurs. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	No receive error
1	Maximum packet receive error

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 13] TOUT (Time OUT)

This is a timeout flag.

If this bit is set to "1", it means that no response is returned from the device within the specified time after a token has been sent in the host mode. When this bit is "0", it means that no timeout is detected. When this bit is "0", it means that no error occurs. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	No timeout
1	Timeout occurs.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 12] CRC (CRC error)

This is a CRC error flag.

If this bit is set to "1", it means that a CRC error is detected in the host mode. When this bit is "0", it means that no CRC error is detected. If a CRC error is detected, bit 5 (Timeout) of this register is also set to "1". When this bit is "0", it means that no CRC error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	No CRC error
1	CRC error occurs.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 11] TGERR (ToGgle ERRor)

This is a toggle error flag.

If this bit is set to "1", it means that the data of this bit does not match the value of the received toggle data. When this bit is "0", it means that no toggle error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	No toggle error.
1	Toggle error occurs.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 10] STUFF (STUFFing error)

This is a stuffing error flag.

If this bit is set to "1", it means that a bit stuffing error is detected. When this bit is "0", it means that no stuffing error is detected. If a stuffing error is detected, bit 5 (Timeout) of this register is also set to "1". If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

Bit	Description
0	No stuffing error.
1	Stuffing error occurs.

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

[bit 9:8] HS (Hand Shake status)

These are handshake status flags.

These flags indicate the status of a handshake packet to be sent or received.

These flags are set to "NULL" when no handshake occurs due to an error or when a SOF token has been ended with the TKNEN bit of the Host Token Endpoint Register (HTOKEN).

These bits are updated when sending or receiving has been ended.

Table 5-1 Handshake

Bit 9	Bit 8	Handshake
0	0	ACK
0	1	NAK
1	0	STALL
1	1	NULL

<Note>

This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).

5.4. Host Status Register (HSTATE)

The Host Status Register (HSTATE) indicates the state of the USB circuit such as a device connection or transfer mode. Note that the setting of the CLKSEL bit is also effective in the function mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	ALIVE	CLKSEL	SOFBUSY	SUSP	TMODE	CSTAT
Attribute	-		R/W	R/W	R/W	R/W	R	R
Initial value	X		0	1	0	0	1	0
Reset enabled or not*	-		x	x			x	x

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 7:6] Reserved bits

These are reserved bits. These bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit 5] ALIVE (keep-ALIVE)

This bit is used to specify the keep-alive function in the low-speed mode. If this bit is set to "1" while the CLKSEL bit of the Host Status Register (HSTATE) is "0", SE0 is output instead of SOF. This bit is effective when the CLKSEL bit of the Host Status Register (HSTATE) is "0". If the CLKSEL bit is "1", SOF is output regardless of the setting of the ALIVE bit.

Bit	Description
0	SOF output
1	SE0 output (Keep-alive)

[bit 4] CLKSEL (CLocK SELect)

This is a USB operation clock selection bit.

Bit	Description
0	Low-speed clock
1	Full-speed clock

<Notes>

- This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).
- Change the value of this bit while the RST bit of the UDC Control Register (UDCC) is "1".
- This bit must always be set to "1". It must not be set to "0".
- Use the on-chip bus (HCLK) clock with 13 MHz or more.

[bit 3] SOFBUSY (SOF BUSY)

This is a SOF busy flag.

When a SOF token is sent using the Host Token Endpoint Register (HTOKEN), this bit is set to "1", which means that the SOF timer is active. When this bit is "0", it means that the SOF timer is under suspension. To stop the active SOF timer, write "0" to this bit. However, if this bit is written with "1", its value is ignored.

Bit	Description
0	The SOF timer is stopped.
1	The SOF timer is active.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- The SOF timer does not stop immediately after "0" has been set to this bit to stop the SOF timer. To check whether or not the SOF timer is stopped, read this bit.

[bit 2] SUSP (SUSPend)

This is a suspend setting bit.

If this bit is set to "1", the USB circuit is placed into the suspend state. If this bit is set to "0" while it is "1" or the USB bus is placed into the k-state mode, the suspend state is released, and the RWIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Table 5-2 Suspend setting

Bit 10	Operation
Set to "1".	Suspend
Set "0" while this bit is "1".	Resume
Others	Holds the status.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- Do not set this bit to "1" while the USB is active (during USB bus resetting, data transfer, or SOF timer running).
- USB clock must not be stopped in the suspend state.
- If the value of this bit is changed, it is not immediately reflected on the state of the USB bus. To check whether or not the state is updated, read this bit.

[bit 1] TMODE (Transmission MODE)

This is a transmission mode flag.

If this bit is "1", it means that the device is connected in the full-speed mode. When this bit is "0", it means that the device is connected in the low-speed mode. This bit is valid when the CSTAT bit of the Host Status Register (HSTATE) is "1".

Bit	Description
0	Low Speed
1	Full Speed

<Notes>

- This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).
 - Use the base clock (HCLK) with 13 MHz or more.
-

[bit 0] CSTAT (Connect STATus)

This is a connection status flag.

When this bit is "1", it means that the device is connected. When this bit is "0", it means that the device is disconnected.

Bit	Description
0	Device is disconnected.
1	Device is connected.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

5.5. SOF Interrupt Frame Compare Register (HFCOMP)

The SOF Interrupt Frame Compare Register (HFCOMP) is used to specify the data to be compared with the low-order eight bits of a frame number when sending a SOF token. When the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0", the value of this register is compared with that of the low-order eight bits of a frame number. If they match, the SOFIRQ bit of the Host interrupt Register (HIRQ) is set to "1" when starting SOF sending. When the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

bit	15	14	13	12	11	10	9	8
Field					FRAMECOMP			
Attribute					R/W			
Initial value					00000000			
Reset enabled or not*					x			

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. o: To be reset.

[bit 15:8] FRAMECOMP

These are frame compare data.

These bits are used to specify the data to be compared with the low-order eight bits of a frame number when sending a SOF token.

If the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0", the frame number of SOF is compared with the value of this register when sending a SOF token. If they match, "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ).

The setting of this register is invalid when the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0".

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

5.6. Retry Timer Setup Register (HRTIMER)

The Retry Timer Setup Register (HRTIMER) is used to specify the token retry time.

bit	15	14	13	12	11	10	9	8
Field					RTIMER1			
Attribute					R/W			
Initial value					00000000			
Reset enabled or not*					x			

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

bit	7	6	5	4	3	2	1	0
Field					RTIMER0			
Attribute					R/W			
Initial value					00000000			
Reset enabled or not*					x			

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

bit	7(23)	6(22)	5(21)	4(20)	3(19)	2(18)	1(17)	0(16)
Field				Reserved			RTIMER2	
Attribute				-			R/W	
Initial value				X			00	
Reset enabled or not*				-			x	

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 23:18] Reserved bits

These are reserved bits. These bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit 17:0] HRTIMER0, 1, 2

These are retry timer setting bits.

These bits are used to specify the retry time in this register. The retry timer is activated when token sending starts while the RETRY bit of Host Control Register 1 (HCNT1) is "1". The retry time is then decremented by one when a 1-bit transfer clock (12 MHz in the full-speed mode) is output. When the retry timer reaches "0", the target token is sent, and processing is ended.

If a token retry occurs in the EOF area, the retry timer is stopped until SOF sending is ended. After SOF sending has been completed, the retry timer restarts with the value that is set when the timer stopped.

<Notes>

- This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC). If data is written while the RST bit of the UDC Control Register (UDCC) is "1", the written data is ignored.
- Write this register in the host mode. Bit 15 to 0 of this register are set to "0" in the function mode. Even if data is written to bits 15 to 0 of this register, it is ignored.

5.7. Host Address Register (HADR)

The Host Address Register (HADR) is used as an address field to send a token.

bit	15	14	13	12	11	10	9	8
Field	Reserved				Address			
Attribute	-				R/W			
Initial value	X				0000000			
Reset enabled or not*	-				x			

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. o: To be reset.

[bit 15] Reserved bit

This is a reserved bit. This bits is undefined in read mode. Even if "0" or "1" is written to this bit, it has no effect on LSI operations.

[bit 14:8] Address

These are address bits.

These bits are used to specify a token address.

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

5.8. EOF Setup Register (HEOF)

The EOF Setup Register (HEOF) is used to specify the token disable time before sending a SOF token. If both the following conditions are satisfied, a request token is sent after a SOF token has been transferred.

- When the value of the SOF timer is compared with that of this register, it is less than the value of this register.
- An IN, OUT, or SETUP token sending request has been issued.

This is a function to prevent a SOF token generated by hardware from being sent together with other tokens. The time unit of this register is the 1-bit transfer time.

bit	15	14	13	12	11	10	9	8
Field	Reserved				EOF1			
Attribute	-				R/W			
Initial value	X				000000			
Reset enabled or not*	-				x			

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

bit	7	6	5	4	3	2	1	0
Field				EOF0				
Attribute				R/W				
Initial value				0000000				
Reset enabled or not*				x				

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 15:14] Reserved bits

These are reserved bits. These bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit 13:0] EOF1, EOF0 (End Of Frame)

These are EOF bits.

These bits are used to specify the time to disable token sending before transferring SOF. Specify the time with a margin, which is longer than the one-packet length. The time unit is the 1-bit transfer time.

Setting example: MAXPKT = 64 bytes, full-speed mode

$$\begin{aligned} & (\text{Token_length} + \text{packet_length} + \text{header} + \text{CRC}) * 7/6 + \text{Turn_around_time} \\ & = (34 \text{ bit} + 546 \text{ bit}) * 7/6 + 36 \text{ bit} = 712.7 \text{ bit} \end{aligned}$$

Therefore, set "(2C9)h".

<Note>

This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).

5.9. Frame Setup Register (HFRAME)

The Frame Setup Register (HFRAME) is used to specify a frame number when sending a SOF token. If SOF sending is set to the TKNEN bit of the Host Token Endpoint Register (HTOKEN), the SOF timer is activated. After this, SOF is sent automatically every 1 ms. The Frame Setup Register is automatically incremented by one each time SOF is ended.

bit	15	14	13	12	11	10	9	8
Field	Reserved					FRAME1		
Attribute	-					R/W		
Initial value	X					000		
Reset enabled or not*	-							

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

bit	7	6	5	4	3	2	1	0
Field	FRAME0							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	-							

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 15:11] Reserved bits

These are reserved bits. These bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit 10:0] FRAME1, FRAME0

These are frame setting bits.

These bits are used to specify a frame number of SOF.

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- Specify a frame number in this register before setting SOF in the TKNEN bit of the Host Token Endpoint Register (HTOKEN).
- This register cannot be written while the SOFBUSY bit of the Host Status Register (HSTATE) is "1" and a SOF token is in process.

5.10. Host Token Endpoint Register (HTOKEN)

The Host Token Endpoint Register (HTOKEN) is used to specify toggle, endpoint, and token.

bit	7	6	5	4	3	2	1	0
Field	TGGL		TKNEN			ENDPT		
Attribute	R/W		R/W			R/W		
Initial value	0		000			0000		
Reset enabled or not*								

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. : To be reset.

[bit 7] TGGL (ToGGLE)

This is a toggle bit.

This bit is used to set toggle data. Toggle data is sent depending on the setting of this bit. When receiving toggle data, received toggle data is compared with the toggle data of this bit to verify whether or not an error occurs.

Bit	Description
0	DATA0
1	DATA1

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- Set this bit when the TKNEN bit of the Host Token Endpoint Register (HTOKEN) is "000".

[bit 6:4] TKNEN (ToKeN ENable)

These are token enable bits.

These bits send a token according to the settings. After operation has been ended, the TKNEN bit is set to "000", and the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

The settings of the TGGL and ENDPT bits are ignored when sending a SOF token.

Table 5-3 Token setting

Bit 6	Bit 5	Bit 4	Operation
0	0	0	Sends no data.
0	0	1	Sends SETUP token.
0	1	0	Sends IN token.
0	1	1	Sends OUT token.
1	0	0	Sends SOF token.
1	0	1	Sends Isochronous IN.
1	1	0	Sends Isochronous OUT.
1	1	1	Reserved (Setting disabled)

<Notes>

- This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).
- The PRE packet is not supported.
- Do not set "100" to the TKNEN bit when the SOFBUSY bit of the Host Status Register (HSTATE) is "1".
- Change the USB to the host mode before writing data to this bit.
- When issuing a token again after the token interrupt flag (CMPIRQ) has been set to "1", wait for 3 cycles or more after a USB transfer clock (12 MHz in the full-speed mode, 1.5 MHz in the low-speed mode) was output, then write data to this bit.
- When the device is disconnected (CSTAT of HSTATE = "0"), token sending is not performed even if data is written to this bit.

[bit 3:0] ENDPT (ENDPoinT)

These are endpoint bits.

These bits are used to specify an endpoint to send or receive data to or from the device.

<Note>

This bit is initialized when "1" is set to the RST bit of the UDC Control Register (UDCC).

Chapter: CAN Prescaler

This chapter describes the CAN prescaler.

1. Overview and configuration
2. CAN Prescaler Register

CODE: 9BFCANPRE-E01.2

1. Overview and configuration

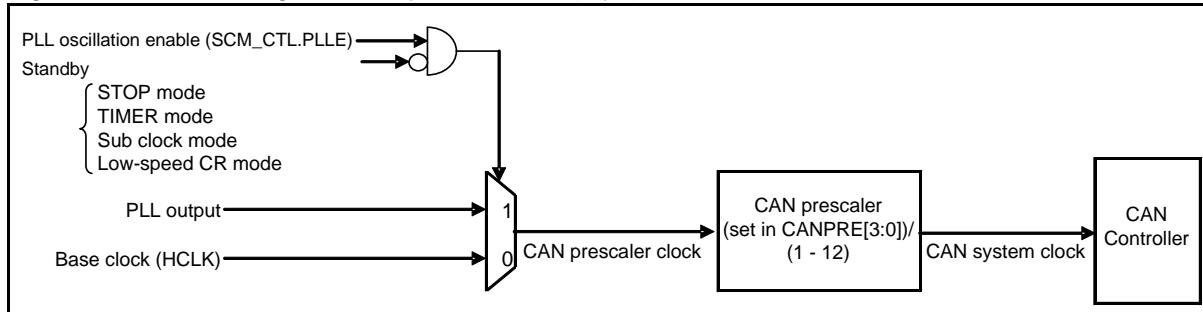
The CAN prescaler generates a CAN system clock (f_{sys}) and supplies it to the CAN.

The CAN prescaler divides a CAN prescaler clock by 1 to 12 frequency, and supplies it to the CAN as a CAN system clock (f_{sys}).

Figure 1-1 shows the block diagram of the CAN prescaler.

■ CAN block diagram

Figure 1-1 Generating a CAN system clock (f_{sys})



■ Explanation of Operations

The CAN prescaler selects the following as a CAN prescaler clock, and supplies it to the CAN after dividing cycles.

- For PLL: PLL output
- For others (including Standby in Figure 1-1): Base clock (HCLK)

2. CAN Prescaler Register

This chapter describes the CAN Prescaler Register.

Abbreviation	Register name	See
CANPRE	CAN Prescaler Register	2.1

2.1. CAN Prescaler Register (CANPRE)

The CAN Prescaler Register is used to configure the CAN system clock (f_{sys}) generation prescaler.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		Reserved		CANPRE			
Attribute	R/W0	-	-	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	0	1	1

■ Register functions

[Bit7] Reserved bit

Be sure to write "0".

[Bit6:4] Reserved bits

Logical 0 is always read. In the write mode, set "0".

[Bit3:0] CANPRE: CAN prescaler setting bits

Bit 3:0	Function	CAN system clock frequency	
		At input of 80 MHz prescaler clock	At input of 60 MHz prescaler clock
0000	The 1/1 cycle of the CAN prescaler clock is selected as a CAN system clock. (Initial value: CANPRE[3:0]=0000)	80 MHz	60 MHz
0001	The 1/2 cycle of the CAN prescaler clock is selected as a CAN system clock.	40 MHz	30 MHz
001x	The 1/4 cycle of the CAN prescaler clock is selected as a CAN system clock.	20 MHz	15 MHz
01xx	The 1/8 cycle of the CAN prescaler clock is selected as a CAN system clock.	10 MHz	7.5 MHz
1000	The 2/3 cycle of the CAN prescaler clock is selected as a CAN system clock. The clock duty is 67%.	53.3 MHz	40 MHz
1001	The 1/3 cycle of the CAN prescaler clock is selected as a CAN system clock.	26.7 MHz	20 MHz
1010	The 1/6 cycle of the CAN prescaler clock is selected as a CAN system clock.	13.3 MHz	10 MHz
1011	The 1/12 cycle of the CAN prescaler clock is selected as a CAN system clock.	6.7 MHz	5 MHz
110x	The 1/5 cycle of the CAN prescaler clock is selected as a CAN system clock.	16 MHz	12 MHz
111x	The 1/10 cycle of the CAN prescaler clock is selected as a CAN system clock.	8 MHz	6 MHz

<Notes>

- Before changing the value of the CAN prescaler setting bit, set the initialization bit of the CAN Control Register (CTRLR) to "1", and stop all bus operations.
- To use the PLL output as a CAN prescaler clock, set the initialization bit of the CAN Control Register (CTRLR) to "0" after PLL oscillation has been stabilized.
- Make sure that the CAN system clock output by the CAN prescaler is 16 MHz or less.

CHAPTER: CAN Controller

This chapter explains CAN.

1. Overview
2. Configuration
3. CAN Controller Operations
4. CAN Registers

1. Overview

The CAN controller complies with CAN protocol version 2.0A/B, a standard protocol for serial communication. CAN is widely used in various industrial fields such as automobile and factory automation.

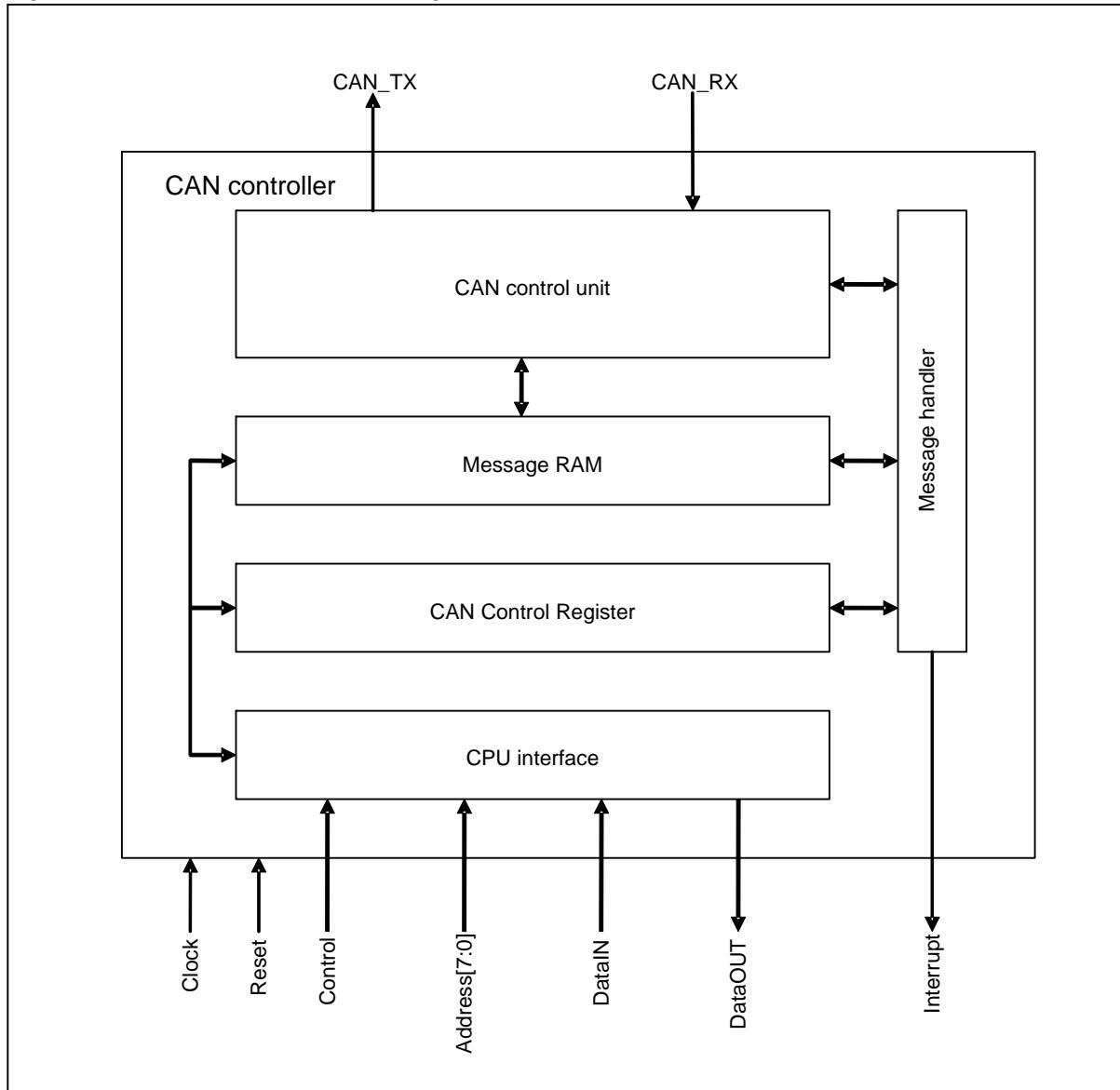
The CAN controller has the following features:

- Supports CAN protocol version 2.0A/B
- Supports a bit rate up to 1 Mbits/s
- Identifier mask for each message object
- Supports programmable FIFO mode
- Maskable interrupt
- Supports 32 message buffers
- Supports programmable loop-back mode for self-test operation
- Read and write from/to the message buffer using interface registers

2. Configuration

Figure 2-1 shows the block diagram of the CAN controller.

Figure 2-1 CAN controller block diagram



- CAN control unit
Controls the CAN protocol and the serial registers for serial/parallel conversion to transfer send/received messages.
- Message RAM
Stores message objects
- Registers
All registers used by CAN.
- Message handler
Controls the message RAM and CAN control unit.
- CPU interface
Controls the internal bus interface.

3. CAN Controller Operations

This section explains the operations and functions of the CAN controller.

Following functions are included:

- Message objects
- Message transmission
- Message reception
- FIFO buffer function
- Interrupt function
- Bit timing
- Test mode
- Software initialization

3.1. Message objects

The following explains message objects and the interface of the message RAM.

■ Message objects

The configuration of message objects in the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits) is not initialized by a hardware reset. Initialize the message objects by the CPU, or set the MsgVal bit to disable (MsgVal = "0"). Configure the CAN Bit Timing Register while the Init bit in the CAN Control Register is "1".

A message object must be configured by programming message interface registers (the IFx Mask Register, IFx Arbitration Register, IFx Message Control Register, and IFx Data Register), and then writing a message number to the corresponding IFx Command Request Register. By writing the message number, the interface register data will be transferred to the addressed message object.

When the Init bit in the CAN Control Register is cleared to "0", the CAN controller starts operation. The received data that have passed acceptance filtering are stored into the message RAM. Messages with pending transmission requests are transferred from the message RAM to the shift register in the CAN controller, and then sent to the CAN bus.

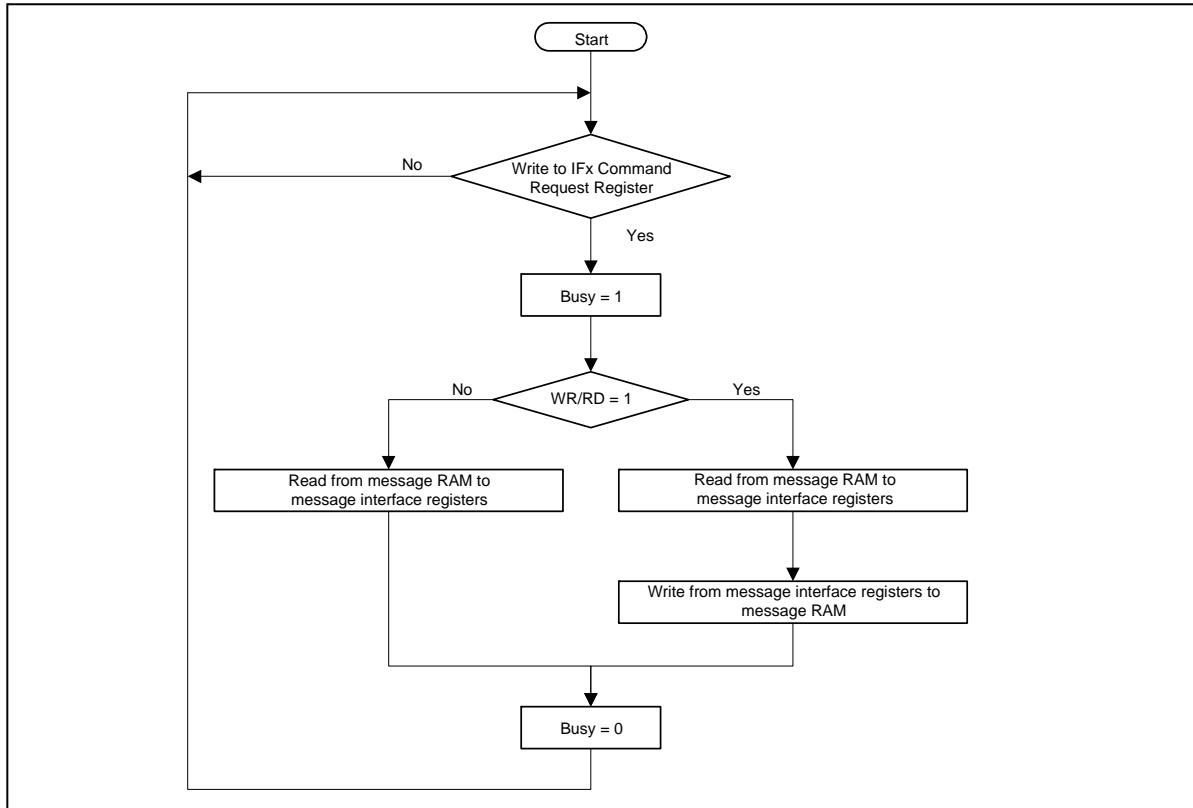
The CPU reads the received messages and updates outgoing messages via message interface registers. The CPU is interrupted according to the configuration of the CAN Control Register and IFx Message Control Register (message object).

■ Data transfer from/to message RAM

When data transfer starts between the message interface registers and message RAM, the Busy bit in the IFx Command Request Register is set to "1". After the transfer has finished, the Busy bit is cleared to "0". (See Figure 3-1)

The IFx Command Register selects whether to transfer complete data or only partial data of one message object. The structure of the message RAM does not allow the writing of single bits/bytes of one message object. The complete data of one message object is always written to the message RAM. Therefore, the data from the message interface registers to the message RAM is transferred in a read-modify-write cycle.

Figure 3-1 Data transfer between the message interface registers and message RAM



3.2. Message transmission

The following explains how to configure the send message objects, and about the transmission.

■ Sending messages

If there is no data transfer between the message interface registers and message RAM, the MsgVal bit in the CAN Message Valid Register and the TxRqst bit in the CAN Transmit Request Register are evaluated. A valid message object with the highest priority of pending transmission requests is transferred to the shift register for transmission. Then the NewDat bit of the message object is reset to "0".

When the transmission has finished successfully, and if there is no new data in the message object (NewDat = "0"), the TxRqst bit is reset to "0". If TxIE is set to "1", then the IntPnd bit is set to "1" after a successful transmission. If the CAN controller lost the arbitration on the CAN bus, or if an error occurred during transmission, the message is resent immediately when the CAN bus becomes idle.

■ Transmission priority

The transmission priority of the message objects is determined by the message number. Message object 1 has the highest priority, while message object 32 (the largest number of the installed message objects) has the lowest priority. If two or more transmission requests are pending, they are transferred in the order of corresponding message number from smallest to largest.

<Notes>

- In one of the following conditions, the messages may not be sent until any of the events described below occurs.
 - Conditions : (1) A message buffer with the lowest priority is used for transmission.
(2) The TxRqst bit was previously set to "1", but is set to "0" to abort transmission.
(3) The TxRqst bit is set to "1" again at the timing of (2).
 - Events :
 - A valid message flows on the CAN bus.
 - A transmission request is issued to another message buffer.
 - CAN is initialized by the Init bit.

If canceling the transmission is required to suit system operations, execute the following steps.

1. Execute one of the following steps.
 - Do not use a message buffer with the lowest priority as a send message buffer.
 - After aborting the transmission, generate any of the above events.
 2. Set the TxRqst bit to "1" again.
- If the message objects of ID28-0, DLC3-0, Xtd, and Data7-0 are changed while the TxRqst bit is "1", message objects before and after the change are mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".

■ Configuring a send message object

Table 3-1 shows how a send object should be initialized.

Table 3-1 Initialization of a send message object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The IFx Arbitration Register (ID28-0 and Xtd bit), given by the application, defines the ID and the type of the outgoing message.

If the standard frame (11-bit ID) is set, then ID28 to ID18 are used, and ID17 to ID0 are ignored. If the extended frame (29-bit ID) is set, then ID28 to ID0 are used.

If TxIE bit is set to "1", then the IntPnd bit is set to "1" after a successful transmission of the message object.

If the RmtEn bit is set to "1", the TxRqst bit is set to "1" after receiving the corresponding remote frame, and a data frame is sent automatically.

The data register (DLC3-0, Data0-7) settings are given by the application.

When Umask is set to "1", the IFx Mask Register (Msk28-0, UMask, MXtd, and MDir bits) is used to receive remote frames with the IDs grouped by the mask setting, and then enable the transmission (by setting the TxRqst bit to "1"). For details, see Remote Frame in "3.3 Message reception".

<Note>

The Dir bit in the IFx Mask Register must not be mask-enabled.

■ Updating a send message object

The CPU can update the data of a send message object via the message interface registers.

The send message object data is written by four bytes of the corresponding IFx data register (in the unit of IFx data register A or IFx data register B). Therefore, the send message object cannot be changed by a single byte.

To update 8-byte data, write 0x0087 to the IFx Command Mask Register, and the message number to the IFx Command Request Register. This concurrently updates the send message object data (of 8-byte) and write "1" to the TxRqst bit.

If both the NewDat and TxRqst bits are set to "1", the NewDat bit is reset to "0" once the transmission is started.

<Notes>

- To update data, update it by four bytes of the IFx Data Register A or IFx Data Register B.
 - If the message objects of ID28-0, DLC3-0, Xtd, and Data7-0 are changed while the TxRqst bit is "1", message objects before and after the change are mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".
-

3.3. Message reception

The following explains how to configure the receive message object and about the reception.

■ Acceptance filtering for received messages

When the arbitration and control field (ID + IDE + RTR + DLC) of a message is completely shifted into the shift register of the CAN controller, scanning of the message RAM is started to compare matching with a valid message object.

Then the arbitration field and mask data (including MsgVal, UMask, NewDat, and EoB) are loaded from a message object in the message RAM, and the message object is compared with the arbitration field of the shift register including mask data.

This operation is repeated "until a matching is detected between a message object and the arbitration field of the shift register", or "until the last word of the message RAM is reached." When a matching is detected, scanning of the message RAM is stopped, and the CAN controller processes data depending of the type of the received frame (data frame or remote frame).

■ Reception priority

The reception priority of the message objects is determined by the message number. Message object 1 has the highest priority, while message object 32 (the largest number of the installed message objects) has the lowest priority. If two or more objects are matched in the acceptance filtering, therefore, the object with the smallest message number becomes the receive message object.

■ Data frame reception

The CAN controller transfers the received message from the shift register into the message RAM of the message object matched in the acceptance filtering. The stored data includes all arbitration fields and the data length code as well as data bytes. This is implemented (to keep the ID and the data bytes) even if the IFx Mask Register is used for masking.

The NewDat bit is set to "1" upon the reception of new data. When the CPU reads the message object, reset the NewDat bit to "0". If the NewDat bit has already been set to "1" upon the reception of a message, the MsgLst is set to "1" indicating that the previous data was lost.

If the RxIE bit has been set to "1", reception of a message buffer causes the IntPnd bit in the CAN Interrupt Pending Register to be set to "1". Then the TxRqst bit of the message object is reset to "0". This is implemented to prevent transmission of a remote frame when the requested data frame is received during the transmission.

■ Remote frame

One of the following three operations is selected when a remote frame is received. The selection depends on how the matching message object is configured.

1. Dir = "1" (Direction = Send), RmtEn = "1", UMask = "1" or "0"
Receives the matched remote frame, sets only the TxRqst of this message object to "1", and automatically replies (sends) data frame to the remote frame. (Other than the TxRqst bit, the message object remains unchanged.)
2. Dir = "1" (Direction = Send), RmtEn = "0", UMask = "0"
Does not receive an incoming remote frame, even if it matches the message object, and disables the remote frame. (The TxRqst bit of the message object remains unchanged.)
3. Dir = "1" (Direction = Send), RmtEn = "0", UMask = "1"
If an incoming remote frame matches the message object, the TxRqst bit of the message object is set to "0", and the remote frame is handled as if it were a received data frame. The received arbitration field and control field (ID + IDE + RTR + DLC) are stored into the message object in the message RAM, and the NewDat bit of this message object is set to "1", The data field of the message object remains unchanged.

■ Configuring a receive message object

Table 3-2 shows how a receive message object should be initialized.

Table 3-2 Initialization of a receive message object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The IFx Arbitration Register (ID28-0 and Xtd bit) is given by the application. The register defines the ID and the type of a received message, used for the acceptance filtering.

If the standard frame (11-bit ID) is set, then ID28 to ID18 are used, and ID17 to ID0 are ignored. When a standard frame is received, ID17 to ID0 are reset to "0". If the extended frame (29-bit ID) is set, then ID28 to ID0 are used.

When the RxIE has been set to "1", and when a received data frame is stored into the message object, then the IntPnd bit is set to "1".

The data length code (DLC3-0) is given by the application. When the CAN controller stores the received data frame into the message object, it stores the received data length code and eight bytes data. If the data length code is less than eight, unspecified data is written to the remaining bytes of the message object.

When Umask is set to "1", the IFx Mask Register (Msk28-0, UMask, MXtd, and MDir bits) is used to allow the reception of data frames with the IDs grouped by the mask setting. For details, see Data Frame Reception in "3.3 Message reception".

<Note>

The Dir bit in the IFx Mask Register must not be mask-enabled.

■ Handling a received message

The CPU can read a received message any time via the message interface registers.

The following shows an example of handling a received message. Write "0x007F" to the IFx Command Register, and a message number of the message object to the IFx Command Request Register. This procedure transfers a received message of the specified message number from the message RAM to the message interface registers. Then the NewDat bit and IntPnd bit of the message object can be cleared to "0" according to the configuration of the IFx Command Mask Register.

An incoming message is received if it is matched in the acceptance filtering. If the message object uses a mask for acceptance filtering, the masked data is excluded from the acceptance filtering to determine whether or not the message should be received.

The NewDat bit indicates whether a new message has been received since the last time the message object was read.

The MsgLst bit indicates that the previous received data was lost because the next data is received before the previous data is read from the message object. The MsgLst bit is not automatically reset.

During transmission of a remote frame, if a data frame matched in the acceptance filtering is received, the TxRqst bit is automatically reset to "0".

3.4. FIFO buffer function

The following explains the configuration of a FIFO buffer of the message object and its operations in handling received messages.

■ Configuration of a FIFO buffer

The configuration of the receive message object belonging to a FIFO buffer is the same as that of a receive message object except the EoB bit. (See Configuring a Receive Message Object in "3.3 Message reception".)

A FIFO buffer is used by concatenating two or more receive message objects. To store received messages into this FIFO buffer, the ID and the mask settings of the receive message objects must be matched when they are used.

The first receive message object of the FIFO buffer has the lowest message number, i.e., the highest priority. In the last receive message object of the FIFO buffer, set "1" to the EoB bit to indicate that the object is the end of the FIFO buffer block. (Except in the last message object, the EoB bit in each message object that uses the FIFO buffer configuration must be set to "0".)

<Notes>

- Be sure to configure the same settings for the ID and the masks of message objects used in the FIFO buffer.
- When the FIFO buffer is not used, be sure to set the EoB bit to "1".

■ Receiving messages using FIFO buffers

A received message, when it matches the FIFO buffer ID, is stored into the receive message object in the FIFO buffer with the lowest message number.

When a message is stored into the receive message object in the FIFO buffer, the NewDat bit of this receive message object is set to "1". When the NewDat bit is set in receive message object while the EoB bit is set to "0", the receive message object is protected until the last receive message object (with EoB bit = "1") is reached. Meanwhile, the CAN controller does not write to the FIFO buffer.

When both of the following conditions are met, the next incoming message is written to the last message object and therefore overwrites the previous message.

- Valid data is stored into the last FIFO buffer
- The NewDat bit of the receive message object is not written by "0" (to release the write protect)

If "0" is not written to the NewDat bit (to release the write protect) of the receive message object while valid data is stored into the last FIFO buffer, the next incoming message is written to the last message object and overwrites the previous message.

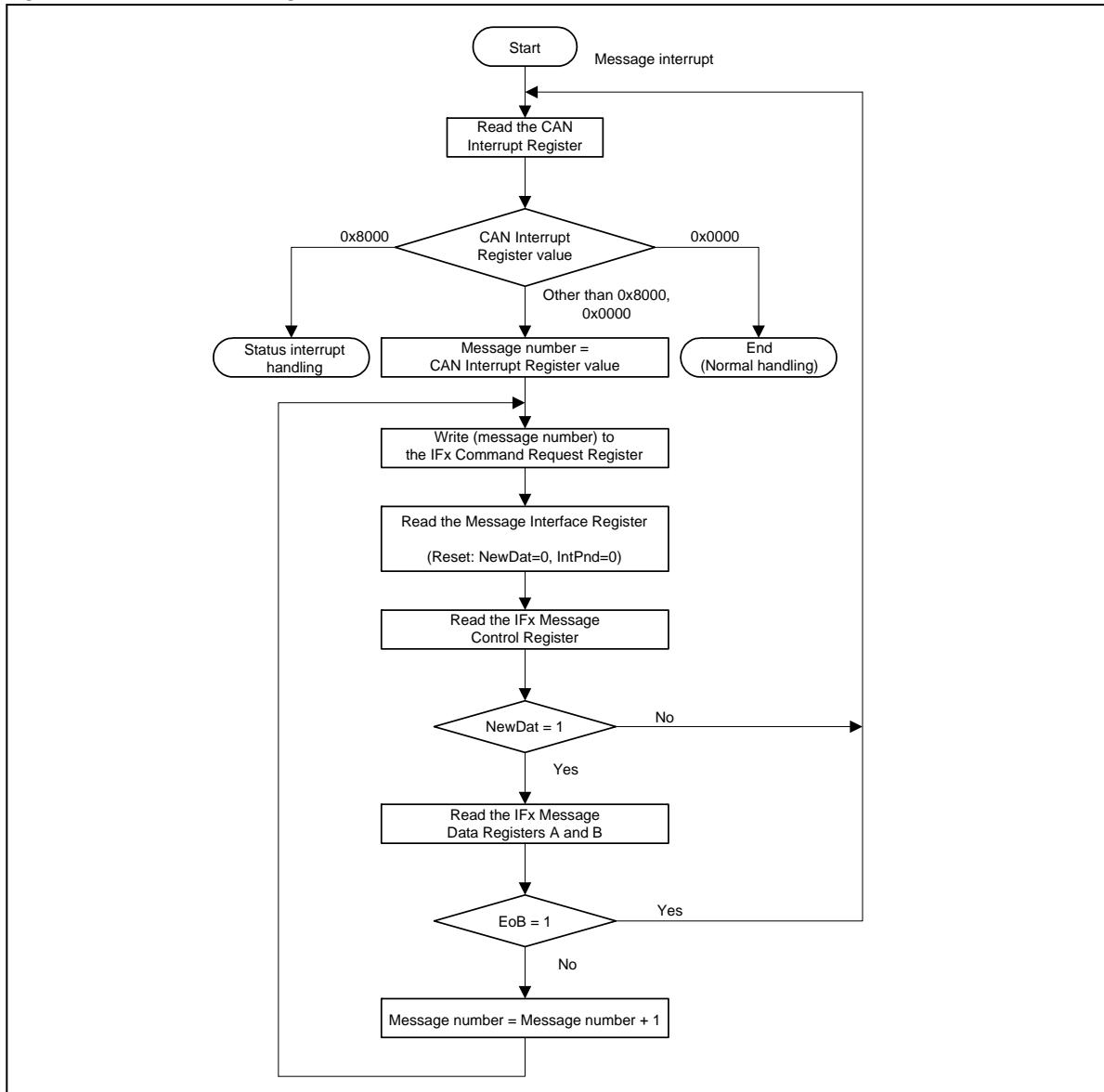
■ Reading from a FIFO buffer

To read the contents of a receive message object, the CPU transfers the object to the Message Interface Register by writing the received message number to the IFx Command Request Register. Then, set WR/RD in the IFx Command Mask Register to "0" (read), set TxRqst/NewDat = 1, ClrIntPnd = 1, and set the NewDat bit and IntPnd bit to "0".

To assure the correct FIFO buffer function, be sure to first read a receive message object in the FIFO buffer with the lowest message number, and then other objects in ascending order.

Figure 3-2 shows how the CPU handles the message objects the FIFO buffer concatenates.

Figure 3-2 CPU handling of FIFO buffer



3.5. Interrupt function

The following explains the interrupt handing using the status interrupt (IntId = 0x8000) and message interrupt (IntId = Message number).

If two or more interrupts are pending, the CAN Interrupt Register points to a pending interrupt code with the highest priority. The chronological order of the interrupt codes are neglected, and the interrupt code with the highest priority is always shown. The interrupt code is retained until the CPU clears it.

The status interrupt (0x8000 of the IntId bit) has the highest priority.

Priority of message interrupts is determined by the message number. A smaller number has a higher priority while the larger the lower.

A message interrupt is cleared by clearing the IntPnd bit of the message object. A status interrupt is cleared by reading the CAN Status Register.

The IntPnd bit in the CAN interrupt Pending Register indicates whether an interrupt has been caused. When no interrupts are pending, the IntPnd bit retains "0".

While the IE bit in the CAN Control Register, and the TxIE bit and RxIE bit in the IFx Message Control Register are set to "1", if the IntPnd bit turns to "1", then the interrupt line to the CPU becomes active. The interrupt line remains active until the CAN Interrupt Pending Register is cleared to "0" (the interrupt cause is reset) or the IE bit in the CAN Control Register is reset to "0".

The 0x8000 value of the CAN Interrupt Register indicates that the CAN Status Register has been updated by the CAN controller. This interrupt has the highest priority. The interrupt by updating the CAN Status Register can enable or disable the setting of the CAN Interrupt Register using the EIE bit and SIE bit in the CAN Control Register. The interrupt line to the CPU can be controlled by the IE bit in the CAN Control Register.

A write access from the CPU can update (reset) the RxOk bit, TxOk bit, and LEC bit in the CAN Status Register. However, the write access cannot generate or reset an interrupt.

Except the 0x8000 and 0x0000 values, the CAN Interrupt Register indicates that a message interrupt is pending, and that the interrupt has the highest priority.

The CAN Interrupt Register is updated even when IE is reset.

The source of a message interrupt to the CPU can be checked from the CAN Interrupt Register or CAN Interrupt Pending Register. (See "4.5 Message handler registers") When clearing a message interrupt, the message data can be read concurrently. If a message interrupt indicated by the CAN Interrupt Register is cleared, the CAN Interrupt Register sets another interrupt with the next higher priority. This waits for the next interrupt handling. If no interrupts are pending, the CAN Interrupt Register shows the 0x0000 value.

<Notes>

- A status interrupt (IntId = 0x8000) is cleared by a read access to the CAN Status Register.
- A write access to the CAN Status Register will not generate a status interrupt (IntId = 0x8000).

3.6. Bit timing

The following provides the overview of the bit timing and explains about the bit timing in the CAN controller.

Each CAN node in the CAN network has its own clock generator (usually a quartz oscillator). The time parameter of the bit time can be configured individually for each CAN node. Even if each CAN node's oscillator has a different period (f_{osc}), a common bit rate can be generated.

The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range (Δf) of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

The bit time can be divided into four segments according to the CAN specifications (see Figure 3-3), into the synchronization segment (Sync_Seg), the propagation time segment (Prop_Seg), the phase buffer segment 1 (Phase_Seg1), and the phase buffer segment 2 (Phase_Seg2). Each segment consists of the programmable number of time quanta (See Table 3-3). The basic unit of the time quantum (tg) is defined by CAN controller's system clock "fsys" and the baud rate prescaler (BRP).

$$tg = BRP / f_{sys}$$

CAN's system clock "fsys" is the frequency of its clock input (See Figure 2-1). Synchronization segment Sync_Seg is a timing in the bit time where edges of the CAN bus level are expected to occur. Propagation time segment Prop_Seg compensates for the physical delay times within the CAN network. Phase buffer segments Phase_Seg1 and Phase_Seg2 must specify the sampling points. Resynchronization jump width (SJW) must define the width within which resynchronization can move the sampling point to compensate for edge phase errors.

Figure 3-3 Bit timing

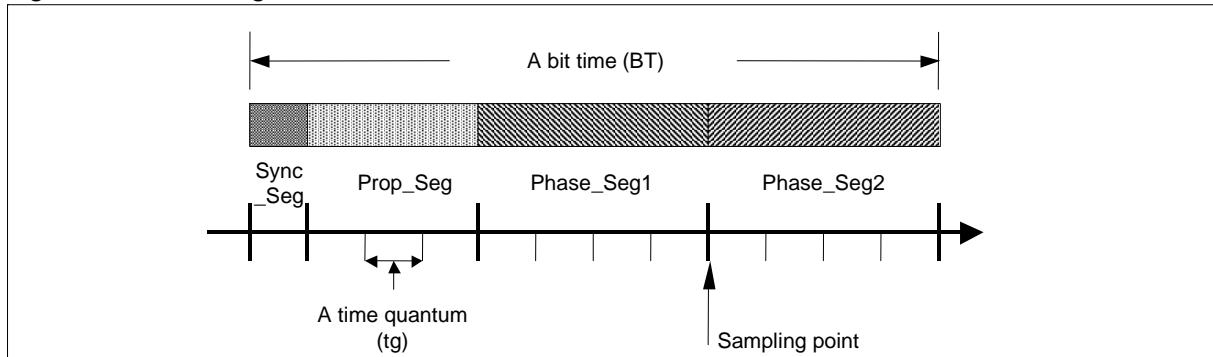


Table 3-3 CAN bit time parameters

Parameter	Range	Function
BRP	[1-32]	Defines the length of time quantum tq.
Sync_Seg	1 tq	Fixed length. Synchronization to system clock.
Prop_Seg	[1-8] tq	Compensates for the physical delay times.
Phase_Seg1	[1-8] tq	Assures edge phase errors before the sampling point. May be prolonged temporarily by synchronization.
Phase_Seg2	[1-8] tq	Assures edge phase errors after the sampling point. May be shortened temporarily by synchronization.
SJW	[1-4] tq	Resynchronization jump width. Will not be longer than either of the phase buffer segments.

The following shows the bit timing in the CAN controller.

Figure 3-4 The bit timing in the CAN controller

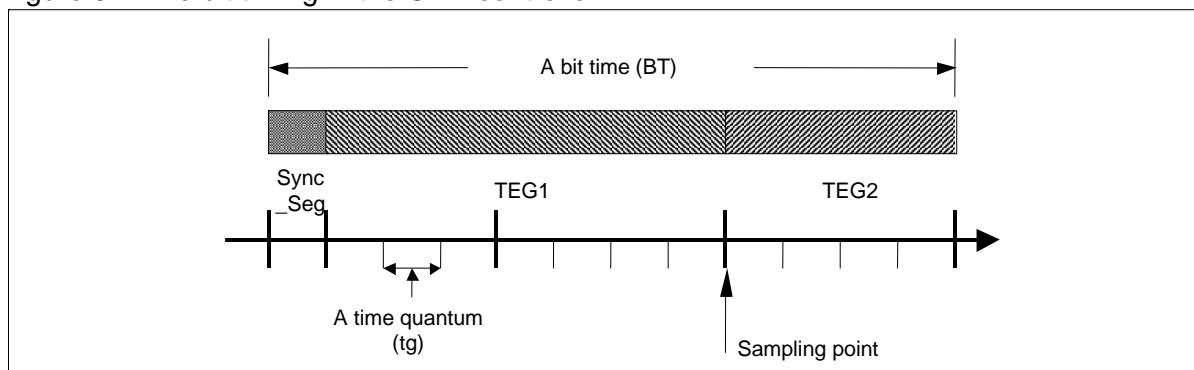


Table 3-4 CAN controller parameters

Parameter	Range	Function
BRPE, BRP	[0-1023]	Defines the length of time quantum tq. Can extend the prescaler by up to 1024 by the Bit Timing Register and the Prescaler Extension Register.
Sync_Seg	1 tq	Synchronization to system clock. Fixed length.
TSeg1	[1-15] tq	A time segment before the sampling point. Equivalent to Prop_Seg and Phase_Seg1. Can be controlled by the Bit Timing Register.
TSeg2	[0-7] tq	A time segment after the sampling point. Equivalent to Phase_Seg2. Can be controlled by the Bit Timing Register.
SJW	[0-3] tq	Resynchronization jump width. Can be controlled by the Bit Timing Register.

The following shows the relations among the parameters:

$$\begin{aligned}
 tq &= ([BRPE, BRP]+1) / f_{sys} \\
 BT &= SYNC_SEG + TEG1 + TEG2 \\
 &= (1 + (TSeg1 + 1) + (TSeg2 + 1)) \times tq \\
 &= (3 + TSeg1 + TSeg2) \times tq
 \end{aligned}$$

3.7. Test mode

The following explains how to configure test mode, and about its operations.

■ Test mode setting

Test mode is entered by setting the Test bit in the CAN Control Register to "1". In test mode, the Tx1, Tx0, LBack, Silent, and Basic bits in the CAN Test Register are enabled.

When the Test bit in the CAN Control Register is set to "0", all test register functions are disabled.

■ Silent mode

The CAN controller can be set in silent mode by programming the Silent bit in the CAN Test Register to "1".

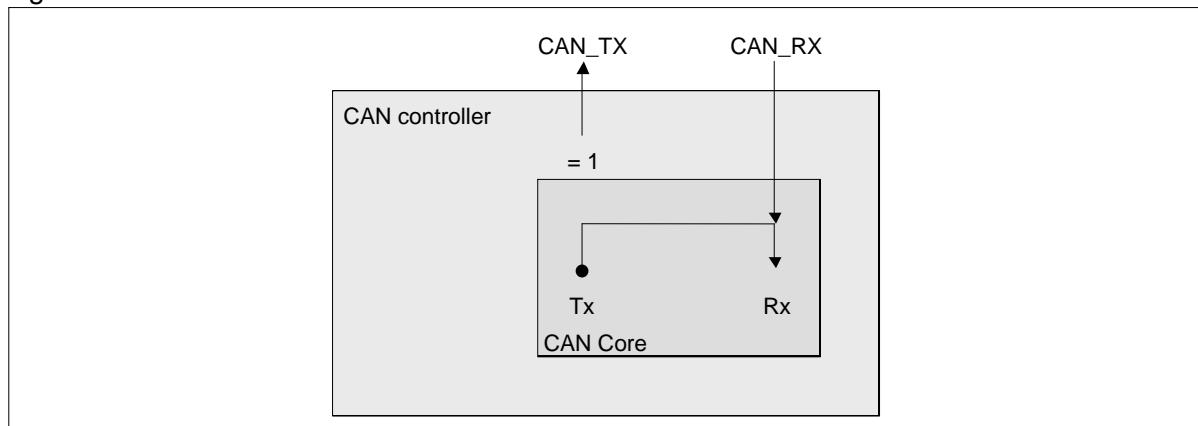
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs recessive bits onto the CAN bus and does not send messages and ACK.

When the CAN controller is required to send dominant bits (ACK bits, overload flags, active error flags), the CAN controller uses the internal rerouting circuit to send them to the RX side. In this operation, the RX side can receive dominant bits rerouted inside the CAN controller even when the CAN bus remains in a recessive state.

In silent mode, the analysis of CAN bus traffic is possible without being affected by transmission of the dominant bits (ACK bits, error flags).

Figure 3-5 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller in silent mode.

Figure 3-5 CAN controller in silent mode



■ Loop back mode

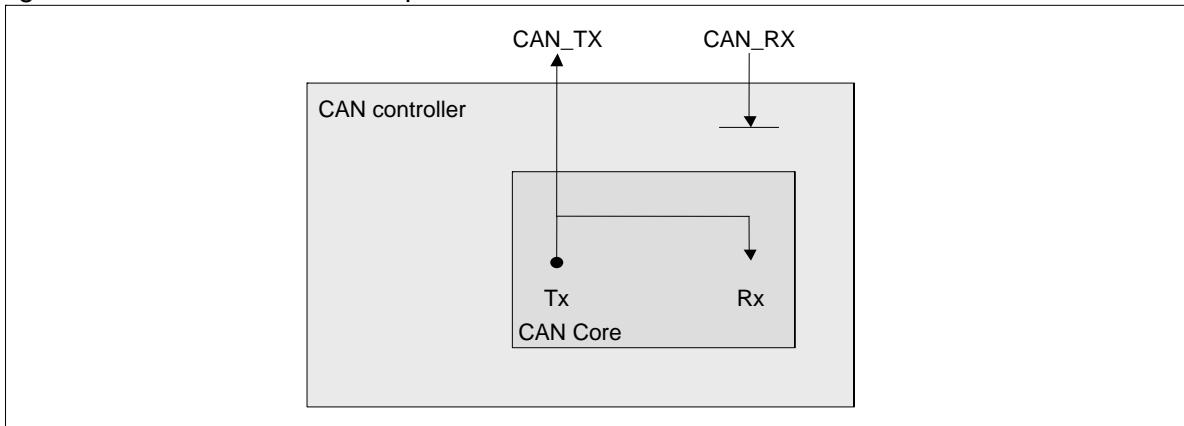
The CAN controller can be set in loop back mode by programming the LBack bit in the CAN Test Register to "1".

Loop back mode can be used for self-diagnostic functions.

In loop back mode, TX is connected with RX inside the CAN controller. The CAN controller treats the transmitted messages as messages received by RX, and stores the messages passed acceptance filtering into the receive buffer.

Figure 3-6 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller in loop back mode.

Figure 3-6 CAN controller in loop back mode



<Note>

Being independent of external signals, the CAN controller does not sample dominant bits in the acknowledgement slot of a data/remote frame. This usually causes the CAN controller to generate acknowledgement errors. In this test mode, however, the errors are not cased.

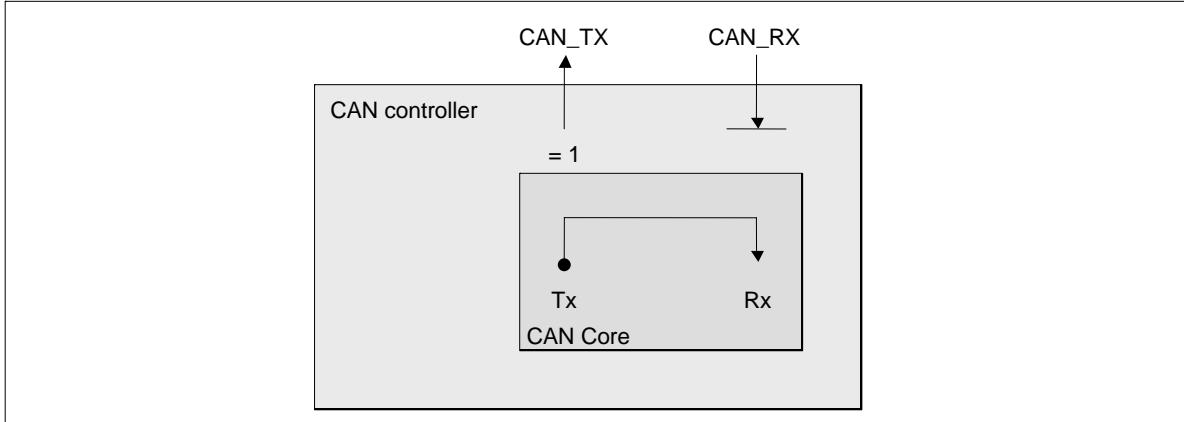
■ Combination of silent mode and loop back mode

Loop back mode and silent mode can be combined by setting the LBack bit and Silent bit in the CAN Test Register to "1" at the same time.

This mode can be used for "Hot self-test". The "Hot self-test" means that the CAN controller can be tested in loop back mode without affecting operation of the CAN system, because a constant recessive value is output from the CAN_TX pin and the input to the CAN_RX pin is ignored.

Figure 3-7 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller when silent mode and loop back mode are combined.

Figure 3-7 CAN controller in combined silent and loop back modes



■ Basic mode

The CAN controller can be set in basic mode by programming the Basic bit in the CAN Test Register to "1".

In basic mode the CAN controller runs without using the message RAM.

The IF1 Message Interface Register is used to control transmission.

First when sending a message, the contents of transmission are configured in the IF1 Message Register. Then the Busy bit in the IF1 Command Request Register is set to "1" to request transmission. While the Busy bit is set to "1", the IF1 Message Interface Register is locked or the transmission is pending.

When the Busy bit is set to "1", the CAN controller performs the following operation:

Immediately when the CAN bus becomes idle, the CAN controller loads the contents of the IF1 Message Interface Register to the send shift register to start transmission. When the transmission has finished successfully, the Busy bit is reset to "0", and the locked IF1 Message Interface Register is released.

While pending, the transmission can be aborted by resetting the Busy bit in the IF1 Command Request Register to "0". If the Busy bit is reset to "0" during the transmission, a possible retransmission in case of lost arbitration or error detection is disabled.

The IF2 Message Interface Register is used to control reception.

All contents of the message are received without using acceptance filtering. The contents of the received message can be read by setting the Busy bit in the IF2 Command Request Register to "1".

When the Busy bit is set to "1", the CAN controller performs the following operation:

- Stores the received message (the contents of the receive shift register) into the IF2 Message Interface Register without any acceptance filtering.

If a new message is stored into the IF2 Message Interface Register, the CAN controller sets the NewDat bit to "1". When an additional message is received while the NewDat bit is "1", then CAN controller sets MsgLst to "1".

<Notes>

- In basic mode, all the message objects related to control and status bits are ignored as well as the control mode setting of the IFx Command Mask Register.
 - The message number of the command request register is ignored.
 - The NewDat bit and MsgLst bit in the IF2 Message Control Register retain their usual function, DLC3-0 indicates the received DLC, and other control bits are read as "0".
-

■ Software control of the CAN_TX pin

CAN_TX is a CAN send pin and has four output functions:

- Outputs serial data (Usual output)
- Outputs CAN sampling point signals to monitor the bit timing of the CAN controller
- Outputs a constant dominant value
- Outputs a constant recessive value

The output of constant dominant and recessive values, combined with CAN_RX monitoring function of the CAN receive pin, can be used to check the CAN bus physical layer.

The output mode of the CAN_TX pin can be controlled by the Tx1 and Tx0 bits in the CAN Test Register.

<Note>

When using CAN message transmission or any of the loop back, silent, or basic modes, the CAN_TX must be set to the serial data output.

3.8. Software initialization

The following explains about initialization using software.

The sources of software initialization are as follows:

- Hardware reset
- Setting the Init bit in the CAN Control Register
- Shift to a busoff state

A hardware reset resets all other than the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits). The message RAM must be initialized, after the hardware reset, by the CPU or by setting the MsgVal in the message RAM to "0". The Bit Timing Register must be configured before clearing the Init bit in the CAN Control Register to "0".

The Init bit in the CAN Control Register is set to "1" in the following conditions:

- Writing "1" from the CPU
- Hardware reset
- In a busoff state

When the Init bit is set to "1", all message transfer from/to the CAN bus is stopped, and the CAN_TX pin in the CAN bus output is in a recessive state (excluding CAN_TX test mode).

Setting the Init bit to "1" does not change the error counter and any register.

When the Init bit and CCE bit in the CAN Control Register are set to "1", the Bit Timing Register for baud rate control and Prescaler Extension Register can be configured.

The software initialization is completed by resetting the Init bit to "0".

By waiting for the occurrence of a consecutive 11 recessive bits (i.e., bus idle) after the Init bit is reset to "0", the message is transferred after synchronization with data transfer on the CAN bus.

Before changing message object masks ID, XTD, EoB, and RmtEn during normal operation, the MsgVal must be disabled.

4. CAN Registers

The following registers are provided for CAN.

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)
- IFx Command Request Register (IFxCREQ)
- IFx Command Mask Register (IFxCMSK)
- IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx Arbitration 1, 2 (IFxARB1, IFxARB2)
- IFx Message Control Register (IFxMCTR)
- IFx Data Register A1, A2, B1, B2 (IFxDA1, IFxDA2, IFxDTB1, IFxDTB2)
- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

■ Total control register list

Table 4-1 Total control register list

Abbreviation	Register name	See
CTRLR	CAN Control Register	4.2.1
STATR	CAN Status Register	4.2.2
ERRCNT	CAN Error Counter	4.2.3
BTR	CAN Bit Timing Register	4.2.4
INTR	CAN Interrupt Register	4.2.5
TESTR	CAN Test Register	4.2.6
BRPER	CAN Prescaler Extension Register	4.2.7

■ Message interface register list

Table 4-2 Message interface register list

Abbreviation	Register name	See
IF1CREQ	IF1 Command Request Register	4.3.1
IF1CMSK	IF1 Command Mask Register	4.3.2
IF1MSK1	IF1 Mask Register 1	4.3.3
IF1MSK2	IF1 Mask Register 2	4.3.3
IF1ARB1	IF1 Arbitration Register 1	4.3.4
IF1ARB2	IF1 Arbitration Register 2	4.3.4
IF1MCTR	IF1 Message Control Register	4.3.5
IF1DTA1	IF1 Data A Register 1 (Little endian)	4.3.6
IF1DTA2	IF1 Data A Register 2 (Little endian)	4.3.6
IF1DTB1	IF1 Data B Register 1 (Little endian)	4.3.6
IF1DTB2	IF1 Data B Register 2 (Little endian)	4.3.6
IF1DTA2	IF1 Data A Register 2 (Big endian)	4.3.6
IF1DTA1	IF1 Data A Register 1 (Big endian)	4.3.6
IF1DTB2	IF1 Data B Register 2 (Big endian)	4.3.6
IF1DTB1	IF1 Data B Register 1 (Big endian)	4.3.6
IF2CREQ	IF2 Command Request Register	4.3.1
IF2CMSK	IF2 Command Mask Register	4.3.2
IF2MSK1	IF2 Mask Register 1	4.3.3
IF2MSK2	IF2 Mask Register 2	4.3.3
IF2ARB1	IF2 Arbitration Register 1	4.3.4
IF2ARB2	IF2 Arbitration Register 2	4.3.4
IF2MCTR	IF2 Message Control Register	4.3.5
IF2DTA1	IF2 Data A Register 1 (Little endian)	4.3.6
IF2DTA2	IF2 Data A Register 2 (Little endian)	4.3.6
IF2DTB1	IF2 Data B Register 1 (Little endian)	4.3.6
IF2DTB2	IF2 Data B Register 2 (Little endian)	4.3.6
IF2DTA2	IF2 Data A Register 2 (Big endian)	4.3.6
IF2DTA1	IF2 Data A Register 1 (Big endian)	4.3.6
IF2DTB2	IF2 Data B Register 2 (Big endian)	4.3.6
IF2DTB1	IF2 Data B Register 1 (Big endian)	4.3.6

■ Message handler register list

Table 4-3 Message handler register list

Abbreviation	Register name	See
TREQ1	CAN Transmit Request Register 1	4.5.1
TREQ2	CAN Transmit Request Register 2	4.5.1
NEWDT1	CAN New Data Register 1	4.5.2
NEWDT2	CAN New Data Register 2	4.5.2
INTPND1	CAN Interrupt Pending Register 1	4.5.3
INTPND2	CAN Interrupt Pending Register 2	4.5.3
MSGVAL1	CAN Message Valid Register 1	4.5.4
MSGVAL2	CAN Message Valid Register 2	4.5.4

4.1. CAN register functions

An address space of 256 bytes is allocated the CAN register. The CPU gains access to the message RAM via the message interface registers.

This section lists CAN registers, and describes the detailed function of each register.

■ Total control registers

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)

■ Message interface registers

- IFx Command Request Register (IFxCREQ)
- IFx Command Mask Register (IFxCMSK)
- IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx Arbitration 1, 2 (IFxARB1, IFxARB2)
- IFx Message Control Register (IFxMCTR)
- IFx Data Register A1, A2, B1, B2 (IFxDA1, IFxDA2, IFxDB1, IFxDB2)

■ Message handler registers

- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

4.2. Total control registers

Total control registers control the CAN protocol and operating modes, and provide status information.

■ Total control registers

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)

4.2.1. CAN Control Register (CTRLR)

The CAN Control Register controls the operating modes of the CAN controller.

■ Register configuration

- CAN Control Register (high-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R0,W0							
Initial value	0	0	0	0	0	0	0	0

- CAN Control Register (low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Test	CCE	DAR	Reserved	EIE	SIE	IE	Init
Attribute	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1

■ Register functions

[bit 15:8] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 7] Test: Test mode enable bit

Bit	Function	
0	Normal operation	[Initial value]
1	Test mode	

<Note>

The Test bit can be set to "1" only while the Init bit is "1".

[bit 6] CCE: Bit Timing Register write enable bit

Bit	Function	
0	Disables write access to the CAN Bit Timing Register and CAN Prescaler Extension Register.	[Initial value]
1	Enables write access to the CAN Bit Timing Register and CAN Prescaler Extension Register. This setting is valid while the Init bit is "1".	

[bit 5] DAR: Automatic retransmission disable bit

Bit	Function
0	Enables automatic retransmission when arbitration is lost or an error is detected. [Initial value]
1	Disables automatic retransmission.

Based on the CAN specification (ISO11898. See 6.3.3 Recovery Sequence), the CAN controller automatically resends frames when arbitration is lost or an error is detected during transfer. To allow the automatic retransmission, set the DAR bit to "0". To operate CAN in Time Triggered CAN (TTCAN. See ISO11898-1) environments, set the DAR bit to "1".

<Notes>

- In the mode where the DAR bit is set to "1", the TxRqst bit and the NewDat bit of a message object behave differently. (For message objects, see "4.4 Message objects")
 - When frame transmission has started, the TxRqst bit of the message object is reset to "0" while NewDat remains set.
 - When frame transmission has finished successfully, the NewDat bit is reset to "0".
- If arbitration is lost or an error is detected during transmission, the NewDat bit remains set. To restart the transmission, the CPU must set the TxRqst to "1".
- If the DAR bit in the CAN Control Register (CTRLR) is changed from "0" to "1" during frame transmission (TxRqst = "1"), a frame being transmitted will be transmitted again. Therefore, change the DAR bit only while the Init bit is "1".
- A transmission using two or more message buffers while the DAR bit is set to "1" assumes the following operations:
 - If the TxRqst in other message buffer is set to "1" before or during frame transmission (TxRqst bits in multiple message buffers are set to "1"), all the set TxRqst bits are reset to "0" upon the start of frame transmission, and data in the message buffer with the highest priority will be sent.

When frame transmission has finished successfully, the NewDat bit of the sent message buffer is reset to "0" and, if TxIE of the message buffer is "1" then, IntPnd of the message object is set to "1".

Data in other message buffers will not be sent because their TxRqst bits have been reset to "0" upon the start of frame transmission.

Check the message buffer sent by NewDat and IntPnd, and then set TxRqst and NewDat to "1" again for another message buffer to be sent.

[bit 4] Reserved bit

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 3] EIE: Error interrupt code enable bit

Bit	Function
0	A change of the BOff or EWarn bit in the CAN Status Register disables the setting of interrupt code in the CAN Interrupt Register. [Initial value]
1	A change of the BOff or EWarn bit in the CAN Status Register enables the setting of status interrupt code in the CAN Interrupt Register.

[bit 2] SIE: Status interrupt code enable bit

Bit	Function
0	A change of the TxOk, RxOk, or LEC bit in the CAN Status Register disables the setting of interrupt code in the CAN Interrupt Register. [Initial value]
1	A change of the TxOk, RxOk, or LEC bit in the CAN Status Register enables the setting of status interrupt code in the CAN Interrupt Register. A change of TxOk, RxOk, or LEC bit caused by write access from the CPU is not set in the CAN Interrupt Register.

[bit 1] IE: Interrupt enable bit

Bit	Function
0	Disables interrupt generation. [Initial value]
1	Enables interrupt generation.

[bit 0] Init: Initialization bit

Bit	Function
0	CAN controller can operate.
1	Initialization [Initial value]

<Notes>

- The busoff recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting the Init bit. If the device enters busoff state, the CAN controller itself sets the Init bit to "1", stopping all bus operations. If the Init bit is cleared to "0" from the busoff state, the bus operation remains stopped until 129 bus idle sequences (one bus idle sequence consists of 11 recessive bits) occur consecutively. When the bus recovery sequence has completed, the error counter is reset.
- If the Init bit is set to "1" and then reset to "0" during the busoff recovery sequence, the busoff recovery sequence restarts from the beginning (sends a set of 11 recessive bits 129 times).
- To write to the CAN Bit Timing Register, set the Init and CCE bits to "1".
- Setting the Init bit to "1" during transfer stops data reception immediately.
- Before making transition to low consumption mode (stop mode or clock mode), and before changing clock supply, the Init bit must be set to "1" to initialize the CAN controller.
- To change the division ratio of clock supplied to the CAN interface by using the following registers, set the Init bit to "1" to stop the CAN controller previously.
 - CAN Bit Timing Register (BTR)
 - CAN Prescaler Extension Register (BRPER)
 - CAN Prescaler (CANPRE)

4.2.2. CAN Status Register (STATR)

The CAN Status Register indicates the CAN status and a CAN bus state.

■ Register configuration

- CAN Status Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R0,W0							
Initial value	0	0	0	0	0	0	0	0

- CAN Status Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	BOff	EWarn	EPass	RxOk	TxOk		LEC	
Attribute	R,WX	R,WX	R,WX	R,W	R,W	R,W	R,W	R,W
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15:8] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 7] BOff: Busoff bit

Bit	Function
0	CAN bus is not in busoff state. [Initial value]
1	CAN bus is in busoff state.

[bit 6] EWarn: Warning bit

Bit	Function
0	Both the send and receive counters are below 96. [Initial value]
1	Send or receive counter has reached or exceeded 96.

[bit 5] EPass: Error passive bit

Bit	Function
0	Both the send and receive counters are below 128 (error active state). [Initial value]
1	The RP bit of the receive counter is "1", or the send counter is between 128 and 255 (error passive state).

[bit 4] RxOk: Successful message reception bit

Bit	Function
0	No message has been transferred successfully on the CAN bus, or the bus is in idle state. [Initial value]
1	A messages has been transferred successfully on the CAN bus.

[bit 3] TxOk: Successful message transmission bit

Bit	Function
0	The bus is in idle state, or no message has been sent successfully. [Initial value]
1	A messages has been sent successfully.

<Note>

The RxOk and TxOk bits can be reset only by the CPU.

[bit 2:0] LEC: Last error code bits

Bit 2:0	State	Function
0	Normal	Successful transmission or reception. [Initial value]
1	Stuff error	Six or more dominant or recessive bits have been detected consecutively in a message.
2	Form error	A wrong fixed format part of a received frame has been detected.
3	Ack error	A sent message was not acknowledged by another node.
4	Bit 1 error	In the sent message data excluding the arbitration field, bits that have been sent as recessive data is detected as dominant data.
5	Bit 0 error	In the sent message data excluding the arbitration field, bits that have been sent as dominant data is detected as recessive data. This bit is set each time 11 recessive bits are detected during bus recovery. The bus recovery sequence can be monitored by reading this bit.
6	CRC error	The CRC data in a received message did not match with the calculated CRC value.
7	Undetected	If the CPU wrote "7" to the LEC bit, and the LEC value is read as "7" afterward, it indicates that no bus event has been detected since the CPU wrote the value. (The bus is in idle state)

The LEC bit holds a code that indicates the last error occurred on the CAN bus. When a message has been transferred (sent or received) without error, this bit is cleared to "0". The undetected code "7" is written by the CPU to check for code updates.

<Notes>

- If the BOff and EWarn bits change while the EIE bit is "1", or if the RxOk, TxOk, and LEC bits change while the SIE bit is "1", the status interrupt code (0x8000) is written to the CAN Interrupt Register.
- Writing from the CPU updates the RxOk and TxOk bits, and this erases the RxOk and TxOk bits set by the CAN controller. If the RxOk and TxOk bits are used, clear the RxOk and TxOk bits within the time (45 x BT) after they are set to "1". BT indicates one bit time.
- If a change of the LEC bit causes an interrupt while the SIE bit is "1", do not write to the CAN Status Register.
- No interrupt is caused by a change of the EPass bit, or writing to the RxOk, TxOk, and LEC bits from the CPU.
- When the BOff bit has turned to "1", the EPass bit and EWarn bit are "1". When the EPass bit has turned to "1", the EWarn bit is "1".
- The status interrupt (0x8000) of the CAN Interrupt Register is cleared by reading this register.

4.2.3. CAN Error Counter (ERRCNT)

The CAN Error Counter indicates the receive error passive, the receive error counter, and the send error counter.

■ Register configuration

- CAN Error Counter (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	RP				REC6-0			
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Error Counter (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field					TEC7-0			
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15] RP: Receive error passive indication

Bit	Function
0	The receive error counter is below the error passive level. [Initial value]
1	The receive error counter has reached the error passive level defined in the CAN specification.

[bit 14:8] REC6-0: Receive error counter

A receive error counter value. The range of the receive error counter value is between 0 and 127.

If the receive error counter reaches or exceeds 128, the RP bit is set to "1", and the counter is not refreshed.

Example: If a receive error adds 8 to REC6-0 = 127 with RP = 0,
then REC6-0 = 127 with RP = 1.

If a receive error adds 8 to REC6-0 = 126 with RP = 0,
then REC6-0 = 126 with RP = 1.

If a receive error adds 8 to REC6-0 = 119 with RP = 0,
then REC6-0 = 127 with RP = 0.

If reception is successful when REC6-0 = 126 and RP = 1,
then REC6-0 = 125 and RP = 0.

[bit 7:0] TEC7-0: Send error counter

A send error counter value. The range of the send error counter value is between 0 and 255.

If the send error counter reaches or exceeds 256, the Init bit of the CAN Control Register is set to "1", and the counter is not refreshed.

Example: If a send error adds 8 to TEC7-0 = 255 with Init = 0,
then TEC7-0 = 255 with Init = 1.

If a send error adds 8 to TEC7-0 = 254 with Init = 0,
then TEC7-0 = 254 with Init = 1.

If a receive error adds 8 to TEC7-0 = 247 with Init = 0,
then TEC7-0 = 255 with Init = 0.

4.2.4. CAN Bit Timing Register (BTR)

The CAN Bit Timing Register configures the prescaler and the bit timing.

■ Register configuration

- CAN Bit Timing Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved	TSeg2			TSeg1			
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	1	1

- CAN Bit Timing Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	SJW		BRP					
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	1

■ Register functions

[bit 15] Reserved bit

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 14:12] TSeg2: Time segment 2 setting bits

Valid programmed values are 0 to 7. The TSeg2 + 1 value is the time segment 2.

The time segment 2 is equivalent to the Phase Buffer Segment (PHASE_SEG2) in the CAN specification.

[bit 11:8] TSeg1: Time segment 1 setting bits

Valid programmed values are 1 to 15. The 0 value must not be used. The TSeg1 + 1 value is the time segment 1.

The time segment 1 is equivalent to the Propagation Segment (PROP_SEG) + Phase Buffer Segment 1 (PHASE_SEG1) in the CAN specification.

[bit 7:6] SJW: Resynchronization jump width setting bits

Valid programmed values are 0 to 3. The SJW + 1 value is the resynchronization jump width.

[bit 5:0] BRP: Baud rate prescaler setting bits

Valid programmed values are 0 to 63. The BRP + 1 value is the baud rate prescaler.

It determines the basic unit of time quantum (tq) for the CAN controller by dividing the system clock (fsys).

<Note>

The CAN Bit Timing Register and CAN Prescaler Extension Register must be configured while the Init bit and CCE bit in the CAN Control Register are set to "1".

4.2.5. CAN Interrupt Register (INTR)

The CAN Interrupt Register indicates message interrupt code and status interrupt code.

■ Register configuration

- CAN Interrupt Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntId15-8							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Interrupt Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntId7-0							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

Bit 15:0	Function
0x0000	No interrupt
0x0001 - 0x0020	An interrupt cause indicates a message object number. (Message interrupt code)
0x0021 - 0x7FFF	Unused.
0x8000	Indicates an interrupt by a change in the CAN Status Register. (Status interrupt code)
0x8001 - 0xFFFF	Unused.

If two or more interrupts are pending, the CAN Interrupt Register indicates a high-priority interrupt code. If a high-priority interrupt code is generated while an interrupt code is set to the CAN Interrupt Register, the CAN Interrupt Register is updated to the high-priority interrupt code.

High-priority interrupt codes are arranged in the order of status interrupt code (0x8000), message interrupt codes (0x0001, 0x0002, 0x0003,, 0x0020).

When the IE bit of the CAN Control Register is set to "1" while the IntId bit is not 0x0000, a CPU interrupt signal becomes active. When the IntId bit is set to 0x0000 (an interrupt cause is reset) or the IE bit of the CAN Control Register is reset to "0", an interrupt signal becomes inactive.

To clear a message interrupt code, reset the IntPnd bit of the target message object (see "4.4 Message objects" for the message object) to "0".

A status interrupt code is cleared by reading the CAN Status Register.

<Note>

To read the CAN Interrupt Register, access it in halfword or word mode.

4.2.6. CAN Test Register (TESTR)

The CAN Test Register is used to monitor the setting of test mode and RX pin. For operations, see "3.7 Test mode".

■ Register configuration

- CAN Test Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R0,W0							
Initial value	0	0	0	0	0	0	0	0

- CAN Test Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	Reserved
Attribute	R,WX	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0
Initial value	r	0	0	0	0	0	0	0

The initial value "r" of Rx in bit 7 indicates the level on the CAN bus.

■ Register functions

[bit 15:8] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 7] Rx: Rx pin monitor bit

Bit	Function	
0	Indicates that the CAN bus is in the dominant state.	
1	Indicates that the CAN bus is in the recessive state.	

[bit 6:5] Tx1-0: TX pin control bit

Bit 6	Bit 5	Function
0	0	Normal operation. [Initial value]
0	1	Outputs a sampling point to the Tx pin.
1	0	Outputs a dominant to the TX pin.
1	1	Outputs a recessive to the TX pin.

[bit 4] LBack: Loop back mode

Bit	Function	
0	Disables loop back mode. [Initial value]	
1	Enables loop back mode.	

[bit 3] Silent: Silent mode

Bit	Function
0	Disables silent mode. [Initial value]
1	Enables silent mode.

[bit 2] Basic: Basic mode

Bit	Function
0	Disables basic mode. [Initial value]
1	Enables basic mode. The IF1 register is used for a sent message, and the IF2 register for a received message.

[bit 1:0] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

<Notes>

- After setting "1" to the Test bit of the CAN Control Register, write data to this register. When the Test bit of the CAN Control Register is set to "1", test mode becomes valid. If the Test bit of the CAN Control Register is set to "0" during processing, test mode changes to normal mode.
 - If the Tx bit is set to a value other than "00", no message can be sent.
-

4.2.7. CAN Prescaler Extension Register (BRPER)

The CAN Prescaler Extension Register is used to extend the prescaler used in the CAN controller by combining it with the prescaler specified at a CAN bit timing.

■ Register configuration

- CAN Prescaler Extension Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R0,W0							
Initial value	0	0	0	0	0	0	0	0

- CAN Prescaler Extension Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	BRPE			
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15:4] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 3:0] BRPE: Baud rate prescaler extension bit

This bit is used to extend the baud rate prescaler up to 1023 by combining BRP and BRPE in the CAN Bit Timing Register.

The value "{BRPE (MSB: 4 bits), BRP (LSB: 6 bits)} + 1" is set as the prescaler value of the CAN controller.

4.3. Message interface registers

The CAN controller provides two message interface registers to control an access from the CPU to the message RAM.

The CAN controller provides two message interface registers to control an access from the CPU to the message RAM. These two registers are used to avoid a conflict between an access from the CPU to the message RAM and an access from the CAN controller to the message RAM by buffering the data (message object) transferred or to be transferred. A message object (see "4.4 Message objects" for message object) is used to collectively transfer data between the message interface registers and message RAM.

Two message interface registers have the same functions, excluding basic test mode, and can be operated independently. For example, the IF1 Message Interface Register can be used to write data to the message RAM while the IF2 Message Interface Register is being used to read data from the message RAM. Table 4-2 shows two message interface registers.

Each Message Interface Register consists of two components: (1) Command Register (Command Request and Command Mask Registers) and (2) Message Buffer Register (Mask, Arbitration, Message Control, and Data Registers) controlled with the Command Register. The Command Mask Register indicates the data transfer direction and also which part in a message object is to be transferred. The Command Request Register is used to select a message number and perform the operation specified in the Command Mask Register.

4.3.1. IFx Command Request Register (IFxCREQ)

The IFx Command Request Register is used to select a message number of the message RAM and transfer data between the message RAM and Message Buffer Register. In basic test mode, IF1 is used to control sending and IF2 to control receiving.

■ Register configuration

- IFx Command Request Register (High-order byte)

Field	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Attribute	BUSY	Reserved						
Initial value	R/W	R0,W0						
	0	0	0	0	0	0	0	0

- IFx Command Request Register (Low-order byte)

Field	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Attribute	R/W							
Initial value	0	0	0	0	0	0	0	1

■ Register functions

A message transfer starts between the message RAM and Message Buffer Register (Mask, Arbitration, Message Control, and Data Registers) immediately after a message number has been written to the IFx Command Request Register. This write operation sets the BUSY bit to "1" and continues transfer processing while the BUSY bit is "1". When transfer processing is ended, the BUSY bit is reset to "0".

If the CPU accesses the Message Interface Register while the BUSY bit is "1", the CPU waits until the BUSY bit is set to "0" (for 3 to 6 clock cycles after data has been written to the Command Request Register).

The method for using the BUSY bit is different in basic test mode. The IF1 Command Request Register, which is used as a send message, starts message sending when the BUSY bit is set to "1". When message transfer has finished successfully, the BUSY bit is reset to "0". Resetting the BUSY bit to "0" enables canceling message transfer at any time.

The IF2 Command Request Register, which is used for receiving message, stores the received message in the IF2 Message Interface Register when the BUSY bit is set to "1".

[bit 15] BUSY: Busy flag bit

- Other than basic test mode

Bit	Function
0	Indicates that data transfer is not performed between the Message Interface Register and message RAM. [Initial value]
1	Indicates that data transfer is being performed between the Message Interface Register and message RAM.

- Basic test mode

- IF1 Command Request Register

Bit	Function
0	Disables message sending.
1	Enables message sending.

- IF2 Command Request Register

Bit	Function
0	Disables message receiving.
1	Enables message receiving.

[bit 14:8] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 7:0] Message Number: Message number (32 message buffers)

Bit 7:0	Function
0x00, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0	Setting disabled. If specified, it is interpreted as 0x20, causing 0x20 to be read.
0x01 - 0x20	Specifies a message number to perform processing.
0x21-0x3F, 0x41-0x5F, 0x61-0x7F, 0x81-0x9F, 0xA1-0xBF, 0xC1-0xDF, 0xE1-0xFF	Setting disabled. If specified, it is interpreted as one of 0x01 to 0x1F, causing the interpreted value to be read.

<Note>

The BUSY bit can be read and written. Therefore, writing any data to this bit does not affect operations, excluding in basic test mode (see "3.7 Test mode" for basic test mode).

4.3.2. IFx Command Mask Register (IFxCMSK)

The IFx Command Mask Register is used to control the transfer direction between the Message Interface Register and message RAM and specify which data is to be updated. This register is invalid in basic test mode.

■ Register configuration

- IFx Command Mask Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R0,W0							
Initial value	0	0	0	0	0	0	0	0

- IFx Command Mask Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	WR/RD	Mask	Arb	Control	CIP	TxRqst/ NewDat	Data A	Data B
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register functions

[bit 15:8] Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit 7] WR/RD: Writing or reading control bit

Bit	Function
0	Indicates that data is read from the message RAM. Reading from the message RAM is performed by writing data to the IFx Command Request Register. What data is to be read from the message RAM depends on the setting of the Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, or Data B bit. [Initial value]
1	Indicates that data is written to the message RAM. Writing to the message RAM is performed by writing data to the IFx Command Request Register. What data is to be written to the message RAM depends on the setting of the Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, or Data B bit.

<Note>

After resetting, data of the message RAM is undefined. The message RAM cannot be read while its data is undefined.

Bit 6 to 0 in the IFx Command Mask Register are set to different values depending on the transfer direction specified with the WR or RD bit.

● When the transfer direction is "writing" (WR/RD="1")

[bit 6] Mask: Mask data update bit

Bit	Function
0	Indicates that mask data (ID mask + MDir + MXtd) of a message object*1 is not updated. [Initial value]
1	Indicates that mask data (ID mask + MDir + MXtd) of a message object*1 is updated.

*1: See "4.4 Message objects".

[bit 5] Arb: Arbitration data update bit

Bit	Function
0	Indicates that arbitration data (ID + Dir + Xtd + MsgVal) of a message object*1 is not updated. [Initial value]
1	Indicates that arbitration data (ID + Dir + Xtd + MsgVal) of a message object*1 is updated.

*1: See "4.4 Message objects".

[bit 4] Control: Control data update bit

Bit	Function
0	Indicates that control data (IFx Message Control Register) of a message object*1 is not updated. [Initial value]
1	Indicates that control data (IFx Message Control Register) of a message object*1 is updated.

*1: See "4.4 Message objects".

[bit 3] CIP: Interrupt clear bit

If this bit is set to "0" or "1", it does not affect CAN controller operations.

[bit 2] TxRqst/NewDat: Message transmission request bit

Bit	Function
0	Indicates that the TxRqst bits of the message object*1 and CAN Transmit Request Register are not changed. [Initial value]
1	Indicates that the TxRqst bits of the message object*1 and CAN Transmit Request Register are set to "1" (transmission requested).

*1: See "4.4 Message objects".

[bit 1] Data A: Data 0-3 update bit

Bit	Function
0	Indicates that data 0-3 of a message object*1 is not updated. [Initial value]
1	Indicates that data 0-3 of a message object*1 is updated.

*1: See "4.4 Message objects".

[bit 0] Data B: Data 4-7 update bit

Bit	Function
0	Indicates that data 4-7 of a message object*1 is not updated. [Initial value]
1	Indicates that data 4-7 of a message object*1 is updated.

*1: See "4.4 Message objects".

<Notes>

- When the TxRqst or NewDat bit of the IFx Command Mask Register is set to "1", the setting of the TxRqst bit in the IFx Message Control Register becomes invalid.
 - This register is invalid in basic test mode.
-

● When the transfer direction is "reading" (WR/RD="0")

[bit 6] Mask: Mask data update bit

Bit	Function
0	Indicates that data (ID mask + MDir + MXtd) is not transferred from a message object*1 to IFx Master Register 1 or 2. [Initial value]
1	Indicates that data (ID mask + MDir + MXtd) is transferred from a message object*1 to IFx Master Register 1 or 2.

*1: See "4.4 Message objects".

[bit 5] Arb: Arbitration data update bit

Bit	Function
0	Indicates that data (ID + Dir + Xtd + MsgVal) is not transferred from a message object*1 to IFx Arbitration Register 1 or 2. [Initial value]
1	Indicates that data (ID + Dir + Xtd + MsgVal) is transferred from a message object*1 to IFx Arbitration Register 1 or 2.

*1: See "4.4 Message objects".

[bit 4] Control: Control data update bit

Bit	Function
0	Indicates that data is not transferred from a message object*1 to the IFx Message Control Register. [Initial value]
1	Indicates that data is transferred from a message object*1 to the IFx Message Control Register.

*1: See "4.4 Message objects".

[bit 3] CIP: Interrupt clear bit

Bit	Function
0	Indicates that the IntPnd bits of the message object*1 and CAN Interrupt Pending Register are held. [Initial value]
1	Indicates that the IntPnd bits of the message object*1 and CAN Interrupt Pending Register are cleared to "0".

*1: See "4.4 Message objects".

[bit 2] TxRqst/NewDat: Data update bit

Bit	Function
0	Indicates that the NewDat bits of the message object*1 and CAN New Data Register are held. [Initial value]
1	Indicates that the NewDat bits of the message object*1 and CAN New Data Register are cleared to "0".

*1: See "4.4 Message objects".

[bit 1] Data A: Data 0-3 update bit

Bit	Function
0	Indicates that data of the message object*1 and CAN Data Register A1 or A2 is held. [Initial value]
1	Indicates that data of the message object*1 and CAN Data Register A1 or A2 is updated.

*1: See "4.4 Message objects".

[bit 0] Data B: Data 4-7 update bit

Bit	Function
0	Indicates that data of the message object*1 and CAN Data Register B1 or B2 is held. [Initial value]
1	Indicates that data of the message object*1 and CAN Data Register B1 or B2 is updated.

*1: See "4.4 Message objects".

<Notes>

- The IntPnd and NewDat bits can be reset to "0" by reading a message object. However, the value before reset by reading is set to the IntPnd and NewDat bits of the IFx Message Control Register.
- This register is invalid in basic test mode.

4.3.3. IFx Mask Registers 1, 2 (IFxMSK1 ,IFxMSK2)

The IFx Mask Registers 1 and 2 are used to write or read message object mask data of the message RAM. The specified mask data is invalid in basic test mode.

For the function of each bit, see "4.4 Message objects".

■ Register configuration

- IFx Mask Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MXtd	MDir	Reserved	Msk28-24				
Attribute	R/W	R/W	R1,W1	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

- IFx Mask Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Msk23-16							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

- IFx Mask Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Msk15-8							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

- IFx Mask Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Msk7-0							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

For the explanation of each bit in this register, see "4.4 Message objects".

Read "1" in the reserved bit (bit 13 of IFx Mask Register 2), and set "1" in write mode.

4.3.4. IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2)

The IFx Arbitration Registers 1 and 2 are used to write or read message object arbitration data of the message RAM. This register is invalid in basic test mode.

For the function of each bit, see "4.4 Message objects".

■ Register configuration

- IFx Arbitration Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal	Xtd	Dir	ID28-24				
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- IFx Arbitration Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	ID23-16							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- IFx Arbitration Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	ID15-8							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- IFx Arbitration Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	ID7-0							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

For the explanation of each bit in this register, see "4.4 Message objects".

<Note>

If the MsgVal bit of a message object is cleared to "0" during transmission, the TxOk bit of the CAN Status Register is set to "1" when transmission has been completed. However, the TxRqst bits of the message object and CAN Transmit Request Register are not cleared to "0". Use the Message Interface Register to clear the TxRqst bit to "0".

4.3.5. IFx Message Control Register (IFxMCTR)

The IFx Message Control Register is used to write or read message object control data of the message RAM. This register is invalid in basic test mode. The NewDat and MsgLst bits of the IF2 Message Control Register are used to perform normal operations. The DLC bits indicate the DLC of the received message. The other control bits are invalid ("0").

For the function of each bit, see "4.4 Message objects".

■ Register configuration

- IFx Message Control Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NewDat	MsgLst	IntPnd	UMask	TxE	RxE	RmtEn	TxRqst
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- IFx Message Control Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	EoB	Reserved	Reserved	Reserved	DLC3-0			
Attribute	R/W	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

For the explanation of each bit in this register, see "4.4 Message objects".

<Note>

The values of the TxRqst, NewDat, and IntPnd bits are set as shown below depending on the setting of the WR or RD bit in the IFx Command Mask Register.

- When the transfer direction is "writing" (IFx Command Mask Register: WR/RD="1")
 - The TxRqst bit of this register is valid only when the TxRqst or NewDat bit of the IFx Command Mask Register is set to "0".
- When the transfer direction is "reading" (IFx Command Mask Register: WR/RD="0")
 - If the IntPnd bits of the message object and CAN Interrupt Pending Register are reset by setting the CIP bit of the IFx Command Mask Register to "1" and writing data to the IFx Command Request Register, the value of the IntPnd bit that is specified before reset is stored in this register.
 - If the NewDat bits of the message object and CAN New Data Register are reset by setting the TxRqst or NewDat bit of the IFx Command Mask Register to "1" and writing data to the IFx Command Request Register, the value of the NewDat bit that is specified before reset is stored in this register.

4.3.6. IFx Data Registers A1, A2, B1, and B2 (IFxDTA1, IFxDTA2, IFxDTB1, and IFxDTB2)

The IFx Data Registers A1, A2, B1, and B2 are used to write or read message object sending or receiving data to or from the message RAM. Those registers are used only to send or receive a data frame, and not to send or receive a remote frame.

■ Register configuration

	addr+3	addr+2	addr+1	addr+0
IFx Data A Register 1 (Little endian)			Data(1)	Data(0)
IFx Data A Register 2 (Little endian)	Data(3)	Data(2)		
IFx Data B Register 1 (Little endian)			Data(5)	Data(4)
IFx Data B Register 2 (Little endian)	Data(7)	Data(6)		
IFx Data A Register 2 (Big endian)			Data(2)	Data(3)
IFx Data A Register 1 (Big endian)	Data(0)	Data(1)		
IFx Data B Register 2 (Big endian)			Data(6)	Data(7)
IFx Data B Register 1 (Big endian)	Data(4)	Data(5)		

- IFx Data Register

bit	15	14	13	12	11	10	9	8
Field	Data							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register functions

- Send message data setting
The set data is sent in the order of Data(0), Data(1), ..., Data(7), beginning with the MSB (bit 7 or bit 15).
- Received message data
The received message data is stored in the order of Data(0), Data(1), ..., Data(7), beginning with the MSB (bit 7 or bit 15).

<Notes>

- If the received message data is lower than eight bytes in length, undefined data is written to the remaining bytes of the Data Register.
- To transfer data to a message object, it is processed every four bytes in the Data A or Data B Register; therefore, it is impossible to update only a part of 4-byte data.

4.4. Message objects

The message RAM provides 32 message objects. To avoid a conflict when simultaneously accessing the message RAM from the CPU and the CAN controller, the CPU cannot directly access message objects. The message RAM is accessed via the IFx Message Interface Register.

This section explains the configuration and functions of a message object.

■ Configuration of message object

Message object

UMask	Msk28-0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28-0	Xtd	Dir	DLC3-0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

<Note>

A message object is not initialized using the Init bit of the CAN Control Register or the hardware reset function. For the hardware reset function, release the hardware reset function, and initialize the message RAM using the CPU or set MsgVal of the message RAM to "0".

■ Functions of message object

The ID28-0, Xtd, and Dir bits are used to indicate the ID and message type when sending a message. They are used in the acceptance filter together with the Msk28-0, MXtd, and MDir bits when receiving a message.

ID, IDE, RTR, DLC, and DATA in a data or remote frame that passed through the acceptance filter are respectively stored in ID28-ID0, Xtd, Dir, DLC3-DLC0, and Data7-Data0 of a message object. Xtd indicates whether the received frame is an extension or standard frame. If Xtd is "1", a 29-bit ID (extension frame) is received. If Xtd is "0", a 11-bit ID (standard frame) is received.

When the received data or remote frame matches one or more message objects, it is stored in the message object with the lowest message number. For details, see Acceptance Filter for Received Messages in "3.3 Message reception".

MsgVal : Valid message bit

Bit	Function
0	Message objects are invalid. Disables message sending/receiving.
1	Message objects are valid. Enables message sending/receiving.

<Notes>

- Reset the MsgVal bit of an unused message object to "0" before clearing the Init bit of the CAN Control Register to "0".
- Be sure to reset the MsgVal bit of a message object to "0" before changing the value of ID28-0, Xtd, Dir, or DLC3-0.
- If the MsgVal bit of a message object is cleared to "0" during transmission, the TxOk bit of the CAN Status Register is set to "1" when transmission has been completed. However, the TxRqst bits of the message object and CAN Transmit Request Register are not cleared to "0". Use the Message Interface Register to clear the TxRqst bit to "0".

UMask : Acceptance mask enable bit

Bit	Function
0	Does not use Msk28-0, MXtd, or MDir.
1	Uses Msk28-0, MXtd, or MDir.

<Notes>

- Change the value of the UMask bit when the Init bit of the CAN Control Register is "1" or the MsgVal bit is "0".
- When the Dir bit is "1" and the RmtEn bit is "0", operations vary depending on the setting of the UMask bit.
 - If the UMask bit is "1", reset the TxRqst bit to "0" when a remote frame has been received through the acceptance filter. The received ID, IDE, RTR, and DLC are stored in a message object, and the NewDat bit is set to "1" while data remains unchanged (data is handled as a data frame).
 - If the UMask bit is "0", the TxRqst bit is held and a remote frame is ignored even if it has been received.

ID28-0 : Message ID

	Function
ID28 - ID0	Specifies a 29-bit ID (extension frame).
ID28 - ID18	Specifies a 11-bit ID (standard frame).

Msk28-0: ID mask

Bit	Function
0	Masks the bit that corresponds to the ID of a message object.
1	Does not mask the bit that corresponds to the ID of a message object.

Xtd: Extension ID enable bit

Bit	Function
0	Uses the 11-bit ID (standard frame) for message object.
1	Uses the 29-bit ID (extension frame) for message object.

MXtd : Extension ID mask bit

Bit	Function
0	Does not compare the set value of the Xtd bit in a message object with that of the IDE bit of a received frame. Determine whether to perform the comparison as the ID of a standard frame or extension frame based on the IDE bit of a received frame.
1	Compares the set value of the Xtd bit in a message object with that of the IDE bit of a received frame.

<Note>

When a 11-bit ID (standard frame) is set to a message object, the ID of a received data frame is written to ID28 to ID18. Msk28 to Msk18 are used to mask the ID.

Dir: Message direction bit

Bit	Function
0	Indicates the receiving direction. When the TxRqst bit is set to "1", a remote frame is sent. When the TxRqst bit is set to "0", a data frame that passed through the acceptance filter is received.
1	Indicates the transmission direction. When the TxRqst bit is set to "1", a data frame is sent. When the TxRqst is "0" and the RmtEn bit is "1", the CAN controller sets the TxRqst bit to "1" if a data frame that passed through the acceptance filter is received.

MDir : Message direction mask bit

Bit	Function
0	Masks the message direction bit (Dir) through the acceptance filter.
1	Does not mask the message direction bit (Dir) through the acceptance filter.

<Note>

Always set the Mdir bit to "1".

EoB: End of buffer bit (For details, see "3.4 FIFO buffer function".)

Bit	Function
0	Indicates that a message object is used as a FIFO buffer, not the last message.
1	Indicates a single message object or the last message object in the FIFO buffer.

<Notes>

- The EoB bit is used to configure a FIFO buffer for message objects 2 to 32.
- When processing a single message object without using a FIFO buffer, be sure to set the EoB bit to "1".

NewDat: Data update bit

Bit	Function
0	Indicates that no valid data resides.
1	Indicates that valid data resides.

MsgLst : Message lost

Bit	Function
0	Message lost does not occur.
1	Message lost occurs.

<Note>

The MsgLst bit is valid only when the Dir bit is "0" (receiving direction).

RxE: Receiving interrupt flag enable bit

Bit	Function
0	Does not change the value of the IntPnd bit after frame receiving has succeeded.
1	Changes the IntPnd bit to "1" after frame receiving has succeeded.

TxE: Transmission interrupt flag enable bit

Bit	Function
0	Does not change the value of the IntPnd bit after frame transmission has succeeded.
1	Changes the IntPnd bit to "1" after frame transmission has succeeded.

IntPnd: Interrupt pending bit

Bit	Function
0	No interrupt cause is detected.
1	An interrupt cause is detected. If other high-priority interrupt is not found, the IntId bit of the CAN Interrupt Register indicates this message object.

RmtEn: Remote enable

Bit	Function
0	Does not change the value of the TxRqst bit when a remote frame has been received.
1	Sets the TxRqst bit to "1" when a remote frame is received while the Dir bit is "1".

<Note>

When the Dir bit is "1" and the RmtEn bit is "0", operations vary depending on the setting of the UMask bit.

- If the UMask bit is "1", reset the TxRqst bit to "0" when a remote frame has been received through the acceptance filter. The received ID, IDE, RTR, and DLC are stored in a message object. The NewDat bit is set to "1" while data remains unchanged (data is handled as a data frame).
- If the UMask bit is "0", the TxRqst bit is held and a remote frame is ignored even if it has been received.

TxRqst : Transmission request bit

Bit	Function
0	Indicates the sending idle state (neither the sending state nor the sending wait state).
1	Indicates the sending or sending wait state.

DLC3-0: Data length code

Bit	Function
0-8	The data frame length is 0 to 8 bytes.
9-15	Setting disabled. 8-byte length if specified.

<Note>

The received DLC is stored in the DLC bit if a data frame is received.

Data 0-7: Data 0 to 7

	Function
Data 0	First data byte in CAN data frame
Data 1	2nd data byte in CAN data frame
Data 2	3rd data byte in CAN data frame
Data 3	4th data byte in CAN data frame
Data 4	5th data byte in CAN data frame
Data 5	6th data byte in CAN data frame
Data 6	7th data byte in CAN data frame
Data 7	8th data byte in CAN data frame

<Notes>

- Serial data is output from the MSB (bit 7 or bit 15) to the CAN bus.
 - If the received message data is lower than eight bytes in length, undefined data is written to the remaining bytes of the Data Register.
 - To transfer data to a message object, it is processed every four bytes in the Data A or Data B Register; therefore, it is impossible to update only a part of 4-byte data.
-

4.5. Message handler registers

Message handler registers are all in read only mode. The TxRqst, NewDat, IntPnd, and MsgVal bits of a message object and the IntId bit indicate the status.

■ Message handler registers

- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

4.5.1. CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)

The CAN Transmit Request Register indicates the TxRqst bit of all message objects. This register checks which message object transmission request is pending by reading the TxRqst bit.

■ Register configuration

- CAN Transmit Request Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	TxRqst32-25							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Transmit Request Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	TxRqst24-17							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Transmit Request Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	TxRqst16-9							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Transmit Request Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	TxRqst8-1							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

TxRqst32-1: Transmission request bit

Bit	Function
0	Indicates the sending idle state (neither the sending state nor the sending wait state).
1	Indicates the sending or sending wait state.

The following shows conditions to set or reset the TxRqst bit.

- Setting conditions
 - Set "1" to the WR/RD bit of the IFx Command Mask Register and "1" to the TxRqst bit, and write data to the IFx Command Request Register to set the TxRqst bit to a specific message object.
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, "0" to the TxRqst bit, and "1" to the Control bit, and "1" to the TxRqst bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the TxRqst bit to a specific message object.
 - If the Dir bit is "1" and the RmtEn bit is "1", the TxRqst bit is set by receiving a remote frame that passed through the acceptance filter.

- Resetting conditions
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, "0" to the TxRqst bit, and "1" to the Control, and "0" to the TxRqst bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the TxRqst bit of a specific message object.
 - The TxRqst bit is reset when frame transmission has finished successfully.
 - If the Dir bit is "1", the RmtEN bit is "0", and the UMask bit is "1", the TxRqst bit is reset by receiving a remote frame that passed through the acceptance filter.

<Notes>

- In one of the following conditions, the messages may not be sent until any of the events described below occurs.

Conditions : (1) A message buffer with the lowest priority is used for transmission.

- (2) The TxRqst bit was previously set to "1", but is set to "0" to abort transmission.
- (3) The TxRqst bit is set to "1" again at the timing of (2).

Events : - A valid message flows on the CAN bus.

- A transmission request is issued to another message buffer.
- CAN is initialized by the Init bit.

If canceling the transmission is required to suit system operations, execute the following steps.

1. Execute one of the following steps.
 - Do not use a message buffer with the lowest priority as a send message buffer.
 - After aborting the transmission, generate any of the above events.
 2. Set the TxRqst bit to "1" again.
- If the message objects of ID28-0, DLC3-0, Xtd, and Data7-0 are changed while the TxRqst bit is "1", message objects before and after the change are mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".
-

4.5.2. CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)

The CAN New Data Register indicates the NewDat bit of all message objects. This register checks which message object data is updated by reading the NewDat bit.

■ Register configuration

- CAN New Data Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NewDat32-25							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN New Data Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	NewDat24-17							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN New Data Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NewDat16-9							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN New Data Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	NewDat8-1							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

NewDat32-1: Data update bit

Bit	Function
0	Indicates that no valid data resides.
1	Indicates that valid data resides.

The following shows conditions to set or reset the NewDat bit.

- Setting conditions
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "1" to the NewDat bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the NewDat bit to a specific message object.
 - The NewDat bit is set by receiving a data frame that passed through the acceptance filter.
 - If the Dir bit is "1", the RmtEN bit is "0", and the UMask bit is "1", the NewDat bit is set by receiving a remote frame that passed through the acceptance filter.

- Resetting conditions
 - Set "0" to the WR/RD bit of the IFx Command Mask Register and "1" to the NewDat bit, and write data to the IFx Command Request Register to reset the NewDat bit of a specific message object.
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "0" to the NewDat bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the NewDat bit of a specific message object.
 - The NewDat bit is reset after data has been transferred to the transmission shift register (internal register).

4.5.3. CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)

The CAN Interrupt Pending Register indicates the IntPnd bit of all message objects. This register checks which message object is pending for interrupt by reading the IntPnd bit.

■ Register configuration

- CAN Interrupt Pending Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntPnd32-25							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Interrupt Pending Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntPnd24-17							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Interrupt Pending Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntPnd16-9							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Interrupt Pending Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntPnd8-1							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

IntPnd32-1: Interrupt pending bit

Bit	Function
0	No interrupt cause is detected.
1	An interrupt cause is detected.

The following shows conditions to set or reset the IntPnd bit.

- Setting conditions
 - If the TxIE bit is set to "1", the IntPnd bit is set when frame transmission has been completed normally.
 - If the RxIE bit is set to "1", the IntPnd bit is set when a frame that passed through the acceptance filter was received normally.
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "1" to the IntPnd bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the IntPnd bit of a specific message object.

- Resetting conditions
 - Set "0" to the WR/RD bit of the IFx Command Mask Register and "1" to the CIP bit, and write data to the IFx Command Request Register to reset the IntPnd bit of a specific message object.
 - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "0" to the IntPnd bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the IntPnd bit of a specific message object.

4.5.4. CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

The CAN Message Valid Register indicates the MsgVal bit of all message objects. This register checks which message object is valid by reading the MsgVal bit.

■ Register configuration

- CAN Message Valid Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal32-25							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Message Valid Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	MsgVal24-17							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Message Valid Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal16-9							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

- CAN Message Valid Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	MsgVal8-1							
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Initial value	0	0	0	0	0	0	0	0

■ Register functions

MsgVal32-1: Message valid bit

Bit	Function
0	Message objects are invalid. Disables message sending/receiving.
1	Message objects are valid. Enables message sending/receiving.

The following shows conditions to set or reset the MsgVal bit.

- Setting conditions

Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Arb bit, and "1" to the MsgVal bit of the IFx Arbitration Register 2. Then write data to the IFx Command Request Register to set the MsgVal bit of a specific message object.

- Resetting conditions

Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Arb bit, and "0" to the MsgVal bit of the IFx Arbitration Register 2. Then write data to the IFx Command Request Register to reset the MsgVal bit of a specific message object.

Chapter: CRC (Cyclic Redundancy Check)

This chapter explains the CRC functions.

1. CRC Overview
2. CRC Operations
3. CRC Registers

CODE: FS15-E02.1

1. CRC Overview

The CRC (Cyclic Redundancy Check) is an error detection system. The CRC code is a remainder after an input data string is divided by the pre-defined generator polynomial, assuming the input data string is a high order polynomial. Ordinarily, a data string is suffixed by a CRC code when being sent, and the received data is divided by a generator polynomial as described above. If the received data is dividable, it is judged to be correct.

■ CRC functions

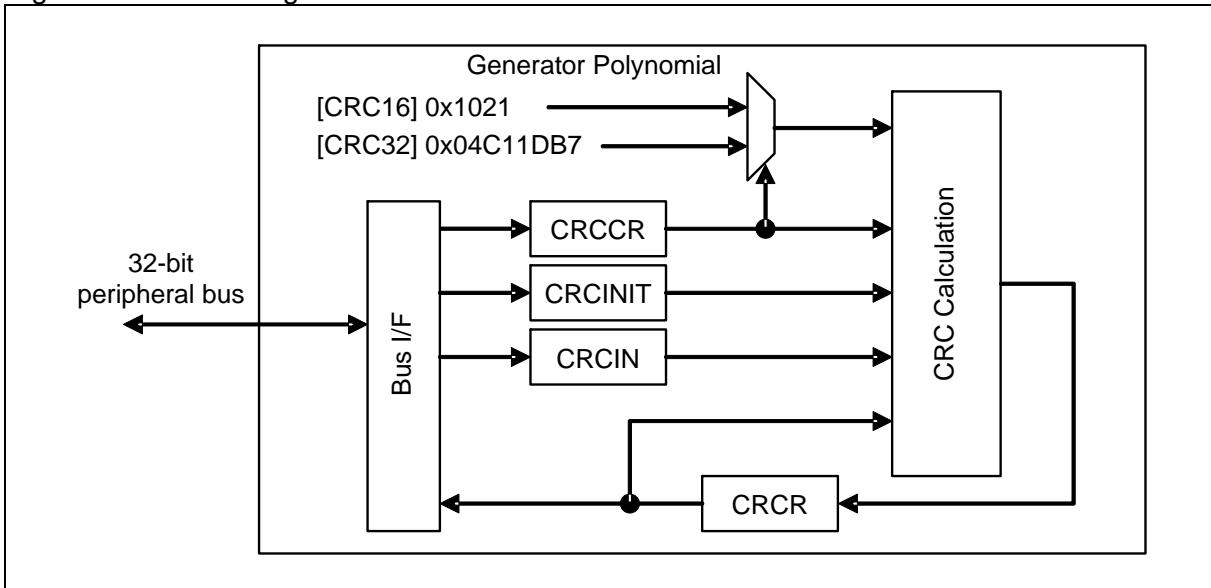
This module enables the calculation in both CCITT CRC16 and IEEE-802.3 CRC32. In this module, the generator polynomial is fixed to the numeric values for those two modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

- CCITT CRC16 generator polynomial: 0x1021
- IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7

■ CRC configuration

Figure 1-1 shows the CRC configuration.

Figure 1-1 CRC configuration



- CRCCR (CRC Control Register)
Used to control CRC calculation.
- CRCINIT (CRC Initial Value Register)
Used to specify the initial values for CRC calculation.
- CRCIN (Input Data Register)
Used to set input data for CRC calculation.
- CRCR (CRC Register)
Used to output the CRC calculation result.
- CRC Calculation
A circuit to perform CRC calculation.

2. CRC Operations

This section provides an overview of CRC operations.

■ CRC definition

[CCITT CRC16 Standard]

Generator polynomial	0x1021	(CRCCR. CRC32=0)
Initial value	0xFFFF	
Final XOR value	0x0000	(CRCCR. FXOR=0)
Bit order	MSB First	(CRCCR. LSBFST=0)
Output bit order	MSB First	(CRCCR. CRCLSF=0)

(The input-output byte order can be specified arbitrarily.)

[IEEE-802.3 CRC32 Ethernet Standard]

Generator polynomial	0x04C11DB7	(CRCCR. CRC32=1)
Initial value	0xFFFFFFFF	
Final XOR value	0xFFFFFFFF	(CRCCR. FXOR=1)
Bit order	LSB First	(CRCCR. LSBFST=1)
Output bit order	LSB First	(CRCCR. CRCLSF=1)

(The input-output byte order can be specified arbitrarily.)

■ Reset operations

When resetting, the Initial Value Register (CRCINIT) and CRC Register (CRCCR) are set to 0xFFFFFFFF. Other registers are cleared to "0".

■ Initialization

Initializing with CRCCR_INIT loads the value of the Initial Value Register to the CRC Register (CRCCR).

■ Processing byte and bit orders

The following shows how to process byte and bit orders, using examples.

Input the following one word to the CRC computing unit.

133.82.171.1 = 10000101 01010010 10101011 00000001

If the byte order is set to big endian (CRCCR_LTLEND=0), the sending sequence in bytes is configured as shown below.

10000101 01010010 10101011 00000001
(1st) (2nd) (3rd) (4th)

If the bit order is set to LSB First (CRCCR_LSBFST=1), the sending sequence in bits is configured as shown below.

10100001 01001010 11010101 10000000
(Head) (End)

<Note>

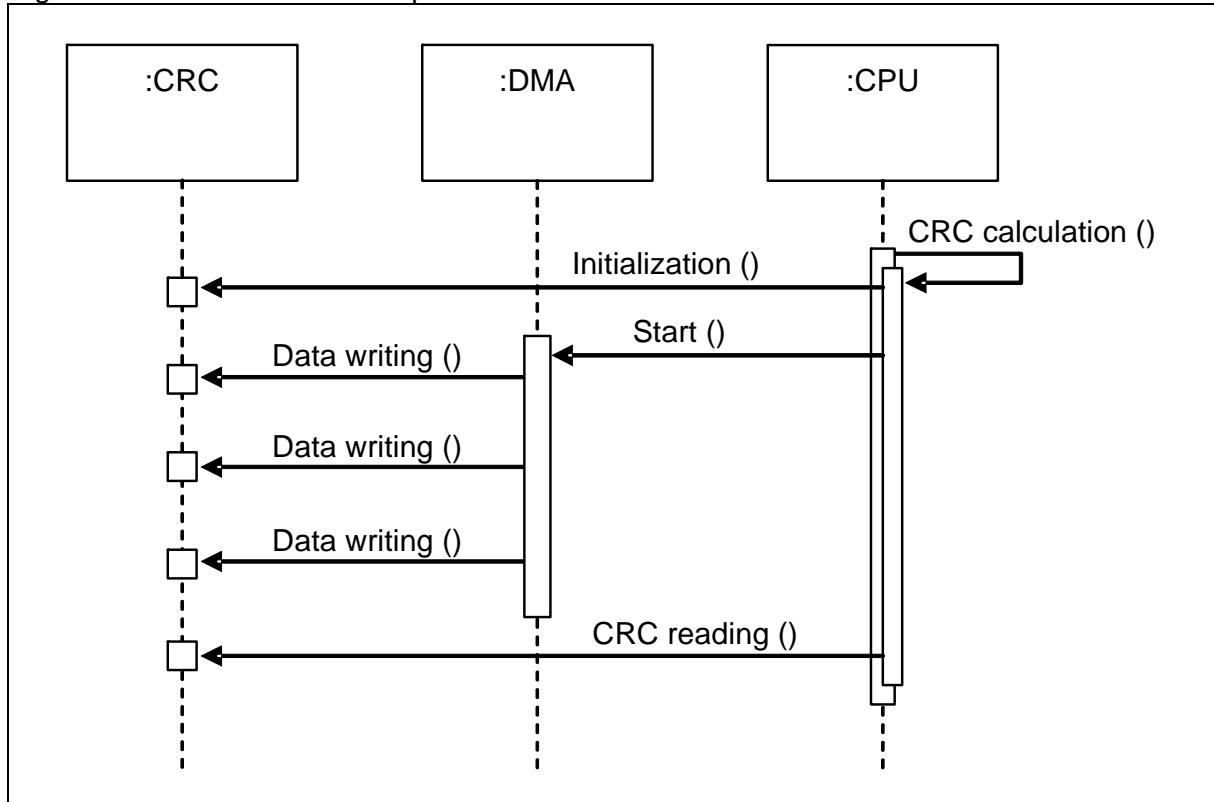
At CRCCR.CRCLTE=1, the CRC result is rearranged in bytes with the 32-bit width in both CRC16 and CRC32.

In particular, in CRC16 mode, note that data is output to bit 31 to bit 16.

2.1. CRC calculation sequence

Figure 2-1 shows the CRC calculation sequence. In this section, it is assumed that the Initial Value Register (CRCINIT) setting, CRC16 or CRC32 mode selection (CRCCR.CRC32), and byte- or bit-order setting (CRCCR.LTLEND, CRCCR.LSBFST) have already been configured. (If the initial value can be set to ALL "H", the Initial Value Register (CRCINIT) setting can be omitted.)

Figure 2-1 CRC calculation sequence



- To perform initialization, write "1" to the initial value bit (CRCCR.INIT). The value of the Initial Value Register is loaded to the CRC Register (CRCR).
- To write input data, write to the Input Data Register (CRCIN). This then starts CRC calculation. If necessary, input data can be written continuously. Furthermore, different bit widths can be used in a sequence to write input data.
- To obtain a CRC code, read the CRC Register (CRCR).

2.2. CRC use examples

Figure 2-2 to Figure 2-5 show CRC use examples.

■ Use example 1 CRC16, Byte input fixed

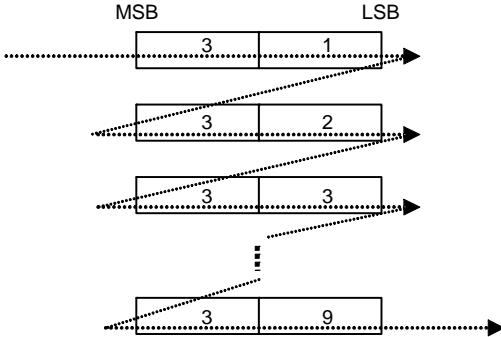
Figure 2-2 Use example 1 (CRC16, byte input fixed, core byte order : Big endian)

```
*****  
// CRC16 (CRC ITU-T)  
// polynomial:      0x1021  
// initial value:   0xFFFF  
// CRCCR.CRC32:    0 //CRC16  
// CRCCR.LTLEND:   0 //big endian  
// CRCCR.LSBFST:   0 //MSB First  
// CRCCR.CRCLTE:   0 //CRC big endian  
// CRCCR.CRCLSF:   0 //CRC MSB First  
// CRCCR.FXOR:     0 //CRC Final XOR off  
*****  
  
// Example 1-1 (Byte-base writing)  
  
// Initialization  
B_WRITE (CRCCR, 0x01);  
  
// data write "123456789"  
B_WRITE (CRCIN, 0x31);  
B_WRITE (CRCIN, 0x32);  
B_WRITE (CRCIN, 0x33);  
B_WRITE (CRCIN, 0x34);  
B_WRITE (CRCIN, 0x35);  
B_WRITE (CRCIN, 0x36);  
B_WRITE (CRCIN, 0x37);  
B_WRITE (CRCIN, 0x38);  
B_WRITE (CRCIN, 0x39);  
  
// read result  
H_READ (CRCR+2, data);  
  
// check result  
assert (data == 0x29B1);  
  
// Example 1-2 (CRC check)  
  
// Initialization  
B_WRITE (CRCCR, 0x01);  
  
// data write "123456789" + CRC  
B_WRITE (CRCIN, 0x31);  
B_WRITE (CRCIN, 0x32);  
B_WRITE (CRCIN, 0x33);  
B_WRITE (CRCIN, 0x34);  
B_WRITE (CRCIN, 0x35);  
B_WRITE (CRCIN, 0x36);  
B_WRITE (CRCIN, 0x37);  
B_WRITE (CRCIN, 0x38);  
B_WRITE (CRCIN, 0x39);  
B_WRITE (CRCIN, 0x29); // <- CRC  
B_WRITE (CRCIN, 0xB1); // <- CRC  
  
// read result  
H_READ (CRCR+2, data);  
  
// check result  
assert (data == 0x0000);
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address

CRC computing unit input sequence image



- The byte and half-word writing positions are arbitrary. In this example, data is written continuously at position +0.
- Table 2-1 shows the CPU, CRC result byte order, CRCR (CRC Register) output position, and read address in CRC16 mode.

Table 2-1 CPU, CRC result byte order, and CRCR read address

Core byte order	CRC result byte order	Output position to CRCR	CRCR H_READ address
Big endian	Big endian	bit 15 to bit 0	CRCR +2
Big endian	Little endian	bit 31 to bit 16	CRCR +0
Little endian	Big endian	bit 15 to bit 0	CRCR +0
Little endian	Little endian	bit 31 to bit 16	CRCR +2

■ Use example 2 CRC16, different input bit widths mixed

Figure 2-3 Use example 2 (CRC16, different input bit widths mixed, core byte order: Big endian)

```

// *****
// CRC16 (CRC ITU-T)
// polynomial:          0x1021
// initial value:       0xFFFF
// CRCCR.CRC32:         0 //CRC16
// CRCCR.LTLEND:        0 //big endian
// CRCCR.LSBFST:        0 //MSB First
// CRCCR.CRCLTE:        0 //CRC big endian
// CRCCR.CRCLSF:        0 //CRC MSB First
// CRCCR.FXOR:           0 //CRC Final XOR off
// *****

//
// Example 2-1 (Writing widths mixed)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789"
W_WRITE (CRCIN, 0x31323334);
H_WRITE (CRCIN, 0x3556);
H_WRITE (CRCIN+2, 0x3738);
B_WRITE (CRCIN+3, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 2-2 (CRC check)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
W_WRITE (CRCIN, 0x31313334);
W_WRITE (CRCIN, 0x35363738);
H_WRITE (CRCIN, 0x3929); // <- CRC(0x29)
B_WRITE (CRCIN, 0xB1); // <- CRC(0xB1)

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);

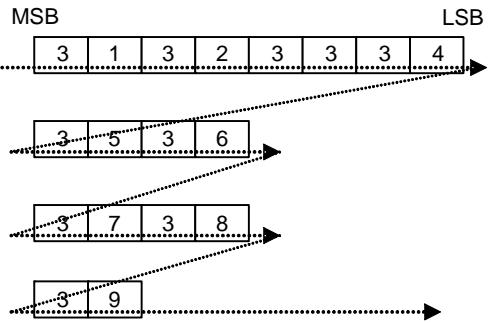
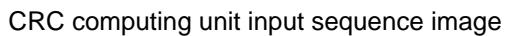
```

(Assumed as follows.)

B_WRITE -- Byte writing
 H_WRITE -- Half-word writing
 W_WRITE -- Word writing

B_READ -- Byte reading
H_READ -- Half-word reading
W_READ -- Word reading

CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



- If the byte or bit order setting is correct and the bit input sequence to the CRC computing unit is the same, the writing width can be specified arbitrarily.
For example, if a 1-, 2-, or 3-byte fraction is obtained in the word-base writing mode, both byte and half-word writings may be enabled.

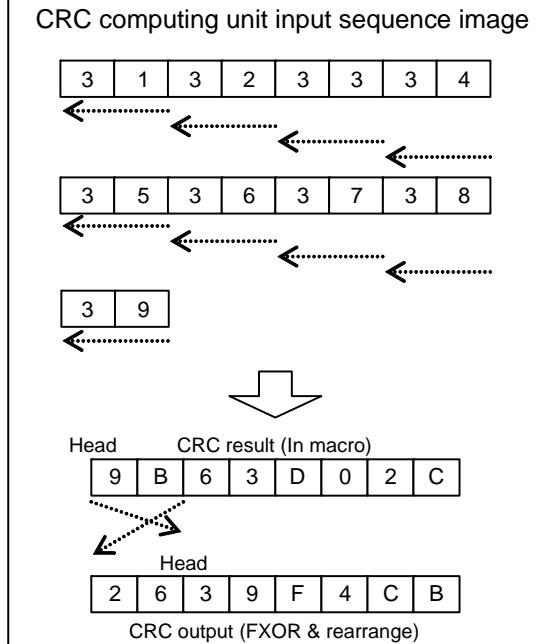
■ Use example 3 CRC32, byte order: Big endian

Figure 2-4 Use example 3 (CRC32, byte order: Big endian)

```
*****  
// CRC32 (IEEE-802.3)  
// polynomial: 0x04C11DB7  
// initial value: 0xFFFF_FFFF  
// CRCCR.CRC32 1 // CRC32  
// CRCCR.LTLEND: 0 // big endian  
// CRCCR.LSBFST: 1 // LSB First  
// CRCCR.CRCLTE: 0 // CRC big endian  
// CRCCR.CRCLSF: 1 // CRC LSB First  
// CRCCR.FXOR: 1 // CRC Final XOR on  
*****  
  
// Example 3-1 (CRC32)  
//  
// Initialization  
B_WRITE (CRCCR, 0x6B);  
  
// data write "123456789"  
W_WRITE (CRCIN, 0x31323334);  
W_WRITE (CRCIN, 0x35363738);  
B_WRITE (CRCIN, 0x39);  
  
// read result  
W_READ (CRCR, data);  
  
// check CRC result  
assert (data == 0x2639F4CB); // <- big endian & LSB First
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



- In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. The figure above shows an example for big endian.

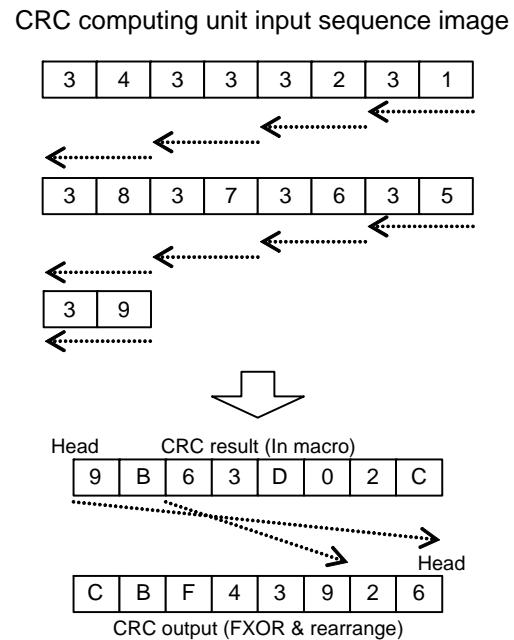
■ Use example 4 CRC32, byte order: Little endian

Figure 2-5 Use example 4 (CRC32, byte order: Little endian)

```
*****  
// CRC32 (IEEE-802.3)  
// polynomial: 0x04C11DB7  
// initial value: 0xFFFF_FFFF  
// CRCCR.CRC32 1 // CRC32  
// CRCCR.LTLEND: 1 // big endian  
// CRCCR.LSBFST: 1 // LSB First  
// CRCCR.CRCLTE: 1 // CRC big endian  
// CRCCR.CRCLSF: 1 // CRC LSB First  
// CRCCR.FXOR: 1 // CRC Final XOR on  
*****  
  
//  
// Example 4-1 (CRC32)  
//  
  
// Initialization  
B_WRITE (CRCCR, 0x7F);  
  
// data write "123456789"  
W_WRITE (CRCIN, 0x34333231);  
W_WRITE (CRCIN, 0x38373635);  
B_WRITE (CRCIN, 0x39);  
  
// read result  
W_READ (CRCCR, data);  
  
// check result  
assert (data == 0xCB43926); // <- little endian & LSB First
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



- In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. The figure above shows an example for little endian.
- If bit inversion is not required for the CRC result, perform either one of the following processes to release the bit inversion for the current result.
 - Initialize with 0x3F before calculation.
 - After data was input, set the CRCCR.FXOR bit to "0" (for example, CRCCR=0x3E).

3. CRC Registers

This section provides a list of CRC registers.

■ CRC registers

Table 3-1 CRC register list

Abbreviation	Register name	See
CRCCR	CRC Control Register	3.1
CRCINIT	Initial Value Register	3.2
CRCIN	Input Data Register	3.3
CRCR	CRC Register	3.4

3.1. CRC Control Register (CRCCR)

The CRC Control Register (CRCCR) is used to control CRC calculation.

bit	7	6	5	4	3	2	1	0
Field	res	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit 7] RES: Reserved bit

The read value is "0".

Be sure to write "0" to this bit.

[bit 6] FXOR: Final XOR control bit

This bit is used to output the CRC result as the XOR value or XOR.

The OR value is set to ALL "H". This bit is inverted at FXOR=1.

This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

Bit	Description
0	None
1	Yes

[bit 5] CRCLSF: CRC result bit-order setting bit

This is a bit-order setting bit for CRC result.

This bit is used to rearrange bits within each byte. Set "0" to specify MSB First and set "1" to specify LSB First.

This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

Bit	Description
0	MSB First
1	LSB First

[bit 4] CRCLTE: CRC result byte-order setting bit

This is a byte-order setting bit for CRC result.

This bit is used to rearrange the byte order in each word. Set "0" to specify big endian and set "1" to specify little endian.

This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

If this bit is set to "1" in CRC16 mode, data is output to bit 31 to bit 16.

Bit	Description
0	Big endian
1	Little endian

[bit 3] LSBFST: Bit-order setting bit

This is a bit-order setting bit.

This bit is used to specify the head bit of a byte (8 bits). Set "0" to specify MSB First and set "1" to specify LSB First.

Four types of processing orders can be specified when this bit is combined with the LTLEND setting.

Bit	Description
0	MSB First
1	LSB First

[bit 2] LTLEND: Byte-order setting bit

This is a byte-order setting bit.

This bit is used to specify the byte order with the write width. Set "0" to specify big endian and set "1" to specify little endian.

Bit	Description
0	Big endian
1	Little endian

[bit 1] CRC32: CRC mode selection bit

This bit is used to select the CRC16 or CRC32 mode.

Bit	Description
0	CRC16
1	CRC32

[bit 0] INIT: Initialization bit

This is an initialization bit. Writing "1" initializes data. This bit does not have a value, and always returns "0" at reading.

At initialization, the value of the Initial Value Register is loaded to the CRC Register.
Initialization must be performed once at the start of CRC calculation.

Bit	Description	
	Write	Read
0	Invalid	Always reads "0".
1	Initialization	

3.2. Initial Value Register (CRCINIT)

The Initial Value Register (CRCINIT) is used to save the initial values for CRC calculation.

bit	31	0
Field	D31-D0	
Attribute	R/W	
Initial value	0xFFFFFFFF	

[bit 31:0] D31 to D0: Initial value bit

This bit is used to save the initial values for CRC calculation.

Write the initial values for CRC calculation to this register.

(0xFFFFFFFF at resetting)

In CRC16 mode, D15 to D0 are used while D31 to D16 are ignored.

3.3. Input Data Register (CRCIN)

The Input Data Register (CRCIN) is used to set input data for CRC calculation.

bit	31	0
Field	D31-D0	
Attribute	R/W	
Initial value	0x00000000	

[bit 31:0] D31 to D0: Input data bit

This bit is used to set input data for CRC calculation.

Write input data for CRC calculation to this register. There are three types of bit widths: 8, 16, and 32, which can be specified together.

The byte and half-word writing positions are arbitrary. The available address positions are as follows.

- Byte writing : +0, +1, +2, +3
- Half-word writing : +0, +2

3.4. CRC Register (CRCR)

The CRC Register (CRCR) is used to output the CRC calculation result. This register must be initialized before start calculating.

bit	31	0
Field		D31-D0
Attribute		R
Initial value		0xFFFFFFFF

[bit 31:0] D31 to D0: CRC bit

This bit is used to read the CRC calculation result. If "1" is written to the initialization bit (CRCCR.INIT), the value of the Initial Value Register (CRCINIT) is loaded to this register.

If input data for CRC calculation is written to the Input Data Register (CRCIN), the CRC calculation result is set to this register after one machine clock cycle has elapsed. When all input data writing has been completed, this register holds the final CRC code.

In CRC16 mode, when the byte order is set to big endian (CRCLTE=0), the result is output to D15 to D0. When the byte order is set to little endian (CRCLTE=1), the result is output to D31 to D16.

Chapter: External Bus Interface

This chapter explains the functions and operations of the external bus interface.

1. External Bus Interface Features
2. Block Diagram
3. Operation
4. Example waveforms of external memory access
5. Endianness and Valid Byte Lanes
6. Connection Examples
7. Registers

1. External Bus Interface Features

This section explains features of the external bus interface.

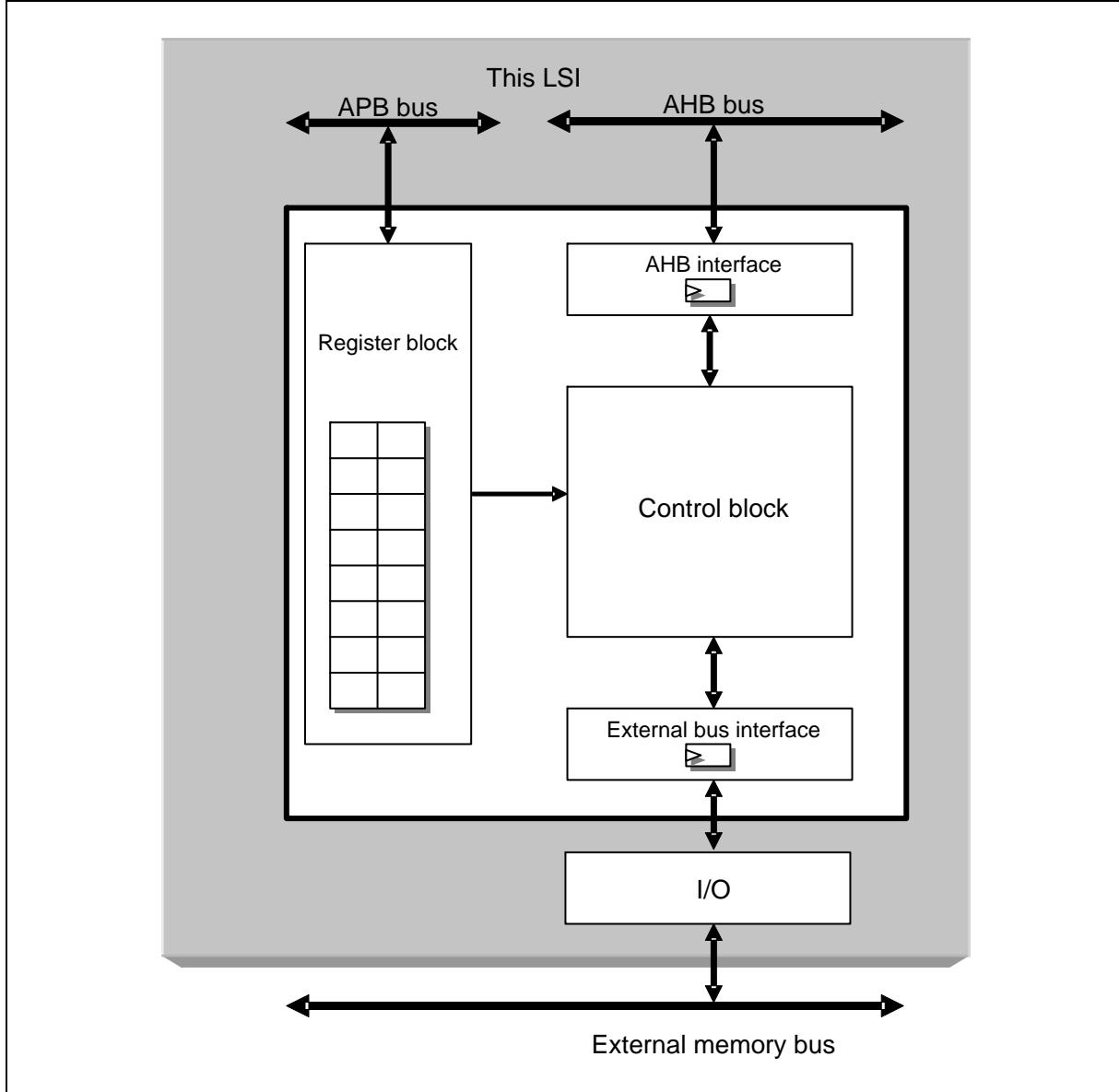
■ External bus interface features

- Supports 8-bit/16-bit wide SRAM/flash memories.
- Provides eight chip select areas for an SRAM/flash memory.
- Can configure parameters individually for each chip select area for an SRAM/flash memory.
- Can access the device during NAND flash memory access. (Exclusive access control is not required.)
- Supports NOR flash memory page access.
- Supports little endian.

2. Block Diagram

This section explains the block diagram of the external bus interface.

Figure 2-1 External bus interface block diagram



3. Operation

The section explains the operations of the external bus interface.

The external bus interface can connect with SRAM and flash memories.

The external bus interface has eight chip select signals.

3.1 Fundamental SRAM and flash memory accesses

3.2 NAND flash memory access

3.1. Fundamental SRAM and flash memory accesses

The following explains fundamental SRAM and flash memory accesses.

The external bus interface has a 256 MB address space. Each address can be configured without restrictions. (The actual maximum address size is 32 MB if the external output address width is taken into consideration.)

A different timing can be specified for each chip select signal. NAND and NOR flash memories can be connected. The NOR flash memory can be accessed as is the case with usual access to SRAM.

A dedicated pin is assigned to a NAND flash memory. With this, an access to the device sharing a data pin with a NAND flash memory is enabled. (See "3.2 NAND flash memory access".)

In SRAM accesses, MCSX0-7 is selected within a single access. If an access uses a bid width larger than the target bit width, the access is converted into a serial access in which only the address is changed while MCSX stays LOW. For example, when an 8-bit wide device receives 16-bit read access from an internal bus, the address is changed from 0 to 1 while MCSX stays LOW, and the data is output serially from MDATA[7:0] with transition timing. (See waveforms of access example.) The data matched to endian is output to the internal bus.

When an access is performed with a width smaller than the target bit width (for example, a byte access to a 16-bit wide internal bus), the byte access is controlled by the MDQM signal (byte mask) during the write operation. (The SRAM/flash memory controller only outputs required data.)

For a device without an input mask, the MDQM signal is used as a write enable.

If the target device has a mask signal, the MDQM control can be used to make the device only output required data. Those processes help reduce power consumption during access.

3.2. NAND flash memory access

The following explains NAND flash memory access.

An access to a NAND flash memory requires different processes from usual SRAM accesses.

A NAND flash memory has a (512+16) byte internal register (an 8-bit/16-bit NAND flash memory has a (1024+32) byte register).

Those registers are used to set the basic data access. For the read operation, several μ s to up to several tens of μ s is required until data is loaded into the internal register. For the write operation, several hundred μ s are required for writing data to memory cells. To erase blocks, several ms are required.

The chip select signal must stay LOW when data is being loaded to the internal register. Therefore, the read/write signal for the NAND flash memory cannot be controlled.

The external bus interface that has a separated enable signal can gain access to the NAND flash memory while the chip select signal connected to the NAND flash memory is kept LOW. (This chip select signal can be reset to HIGH.) This allows the interface to access another chip that shares data signals.

A write access to +0x2000 is converted into the issue of an address for the NAND flash memory (MNALE is asserted).

A write access to +0x1000 is converted into the issue of a command for the NAND flash memory (MNCLE is asserted).

A write/read access to +0x0000 is converted into a data access to the NAND flash memory (MNALE and MNCLE are not asserted) based on the base address within the area set to NAND mode.

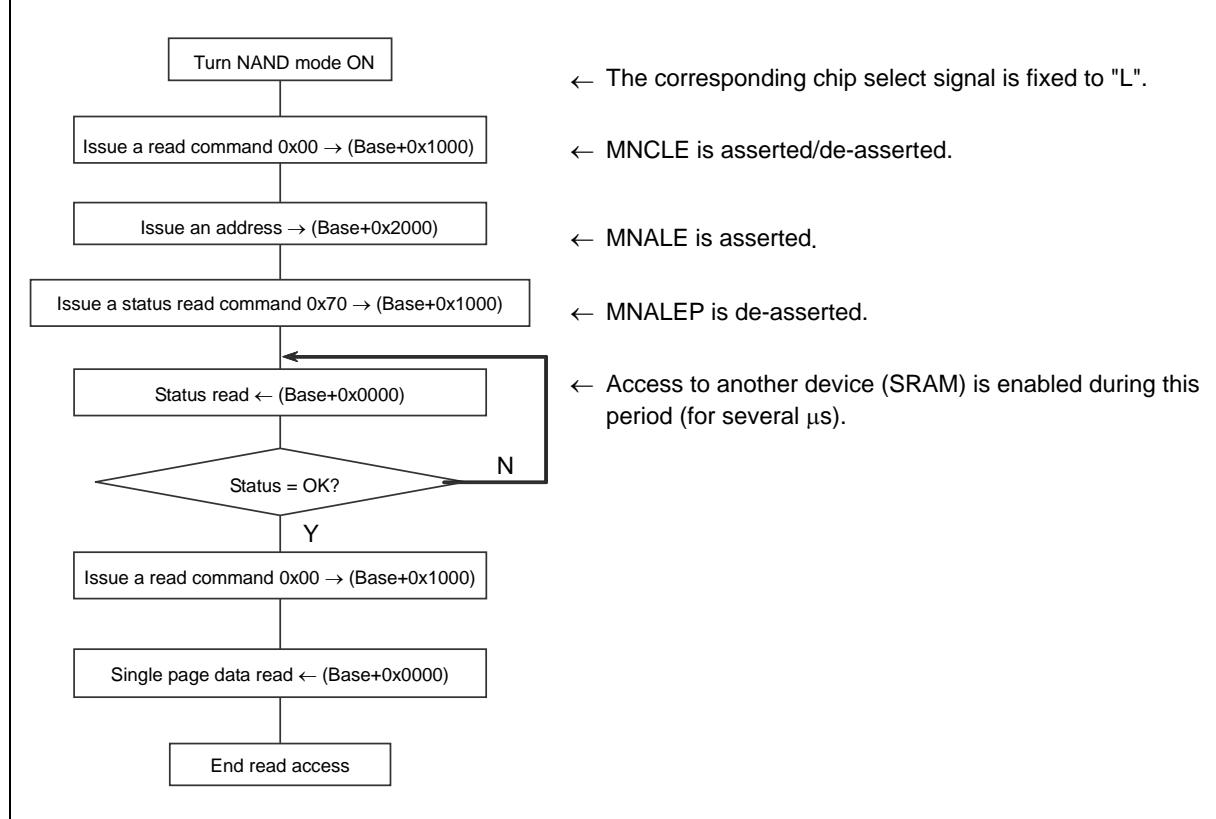
The setting for access timing is the same as that used for SRAM access. MNCLE is output at the same timing as address output during the access.

MNALE is in the asserted state until a write access to +0x3000 after the issue of an address, or any other write accesses (data or command) than address issues. This is because NAND flash memory cannot de-assert ALE within multiple write accesses for the issue of addresses. Addresses up to +0x3000 only assert MNALE and do not allow an access. Figure 3-1 shows the process of NAND flash memory access. (For details about the commands, see the specification of NAND flash memory connected to this LSI)

3.2.1. Read access to NAND flash memory

Figure 3-1 shows the flowchart of read access to NAND flash memory.

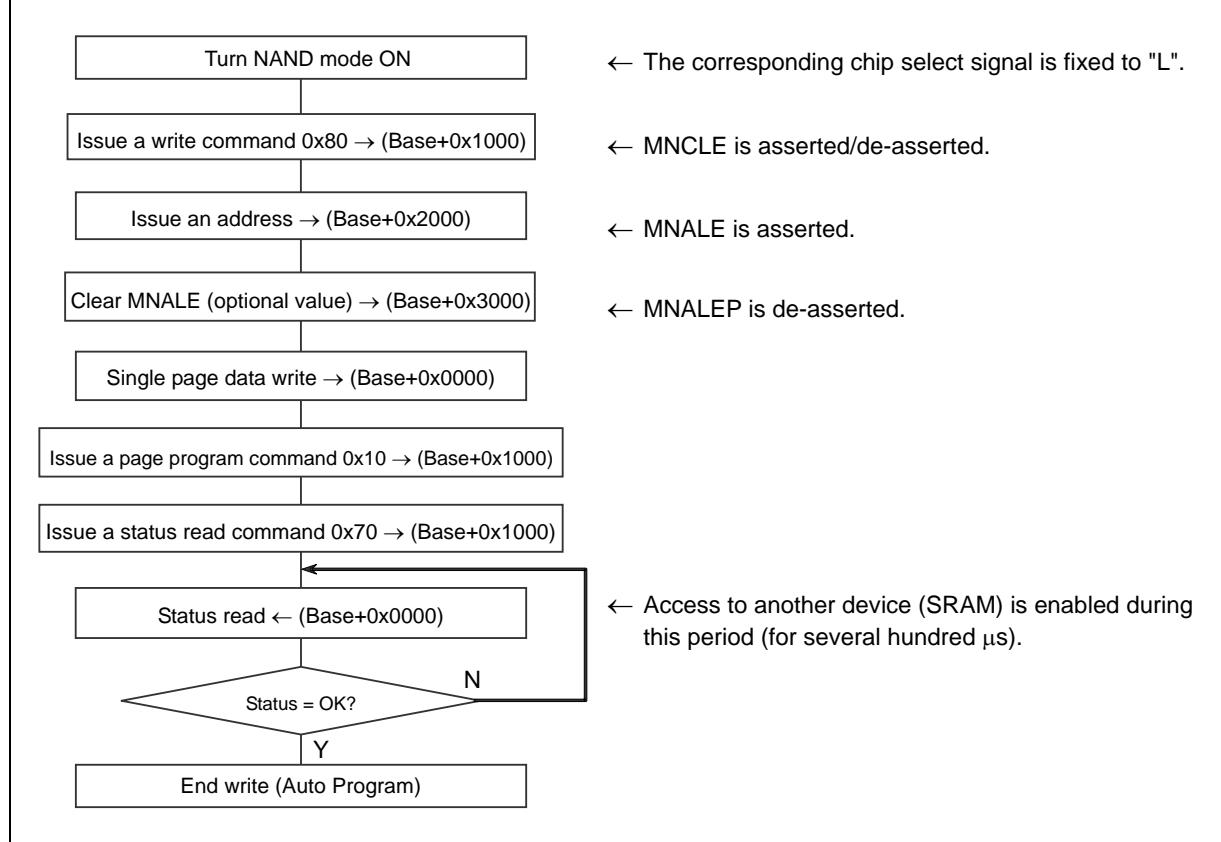
Figure 3-1 Flowchart of read access to NAND flash memory.



3.2.2. Write (auto program) access

Figure 3-2 shows the flowchart of the write (auto program) access.

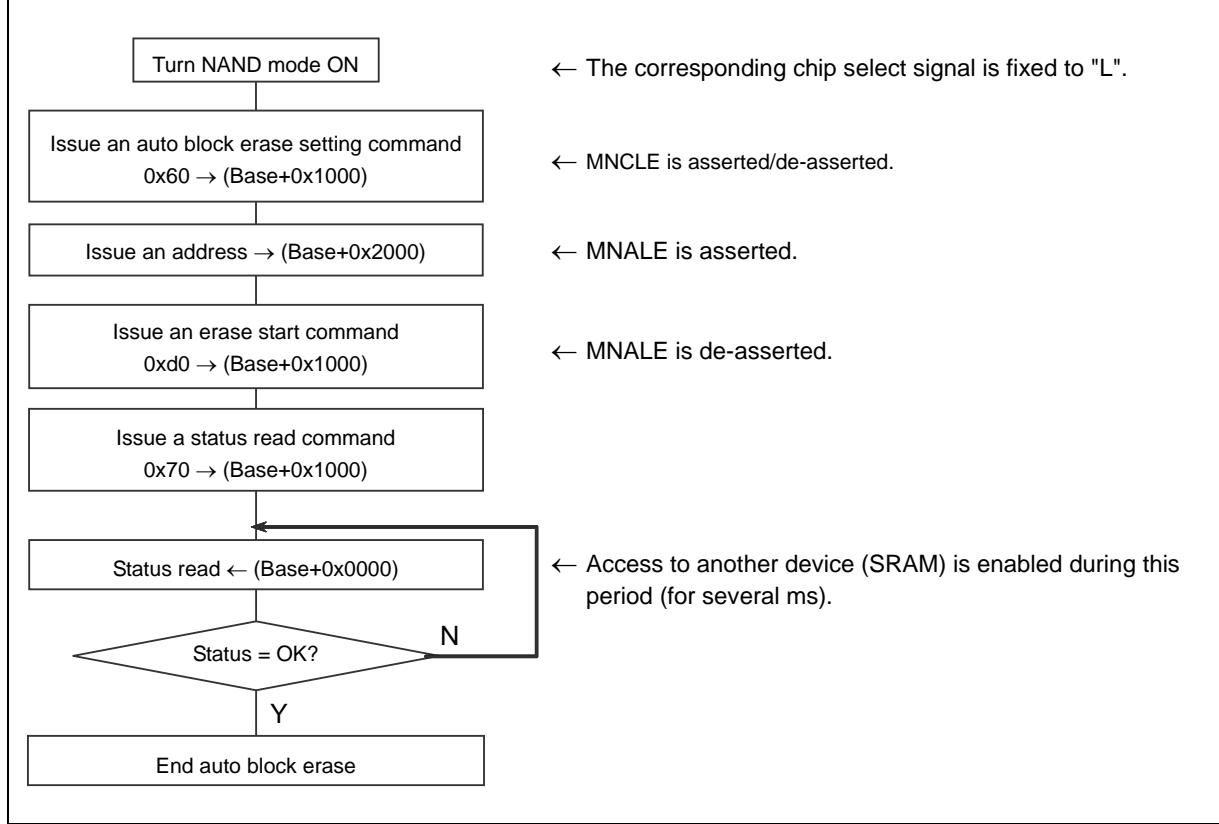
Figure 3-2 Write (auto program) access flowchart



3.2.3. Auto block erase access

Figure 3-3 shows the flowchart of the auto block erase access.

Figure 3-3 Auto block erase access



As shown in the above flowchart, access to another memory device is possible even in the stage where the process of accessing NAND flash memory has not finished. Because DMA can also be used for reading and writing data, the processor can access the NAND flash memory with minimum instructions.

3.2.4. Error reply

The following explains the error reply.

When an access to an invalid address occurs, the external bus interface outputs an error reply (by setting HRESP[1:0] to "01"). When this error occurs during a burst transfer, operation of the external bus interface is not guaranteed.

4. Example waveforms of external memory access

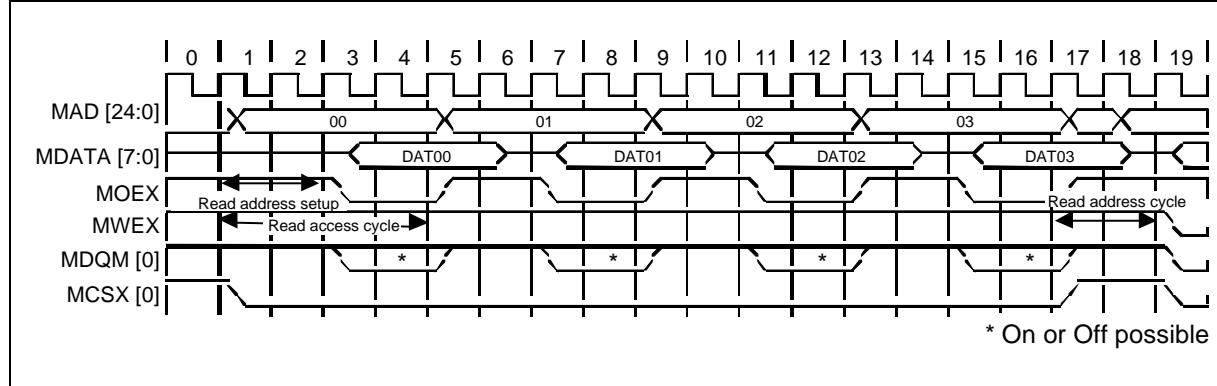
This section provides waveforms of external memory access by the external bus interface.

- 4.1 Word read access to 8-bit wide SRAM
- 4.2 Serial word write/read access to 16-bit wide SRAM
- 4.3 Issue of an 8-bit NAND flash memory read/write command
- 4.4 8-bit NAND flash memory status read
- 4.5 8-bit NAND flash memory data write
- 4.6 16-bit NOR flash memory page read

4.1. Word read access to 8-bit wide SRAM

Figure 4-1 shows waveforms of a word read access to 8-bit wide SRAM.

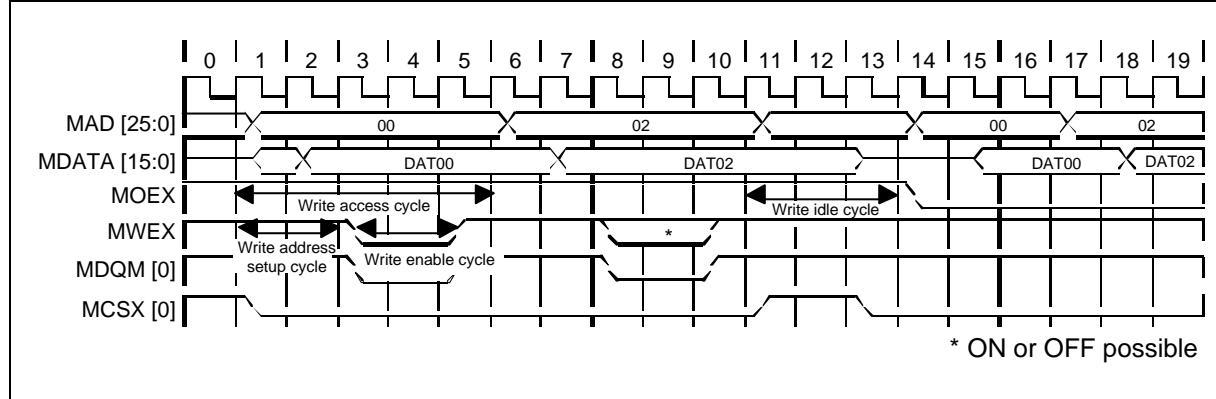
Figure 4-1 Waveforms of a word read access to 8-bit wide SRAM



4.2. Serial word write/read access to 16-bit wide SRAM

Figure 4-2 shows waveforms of a serial word write/read access to 16-bit wide SRAM.

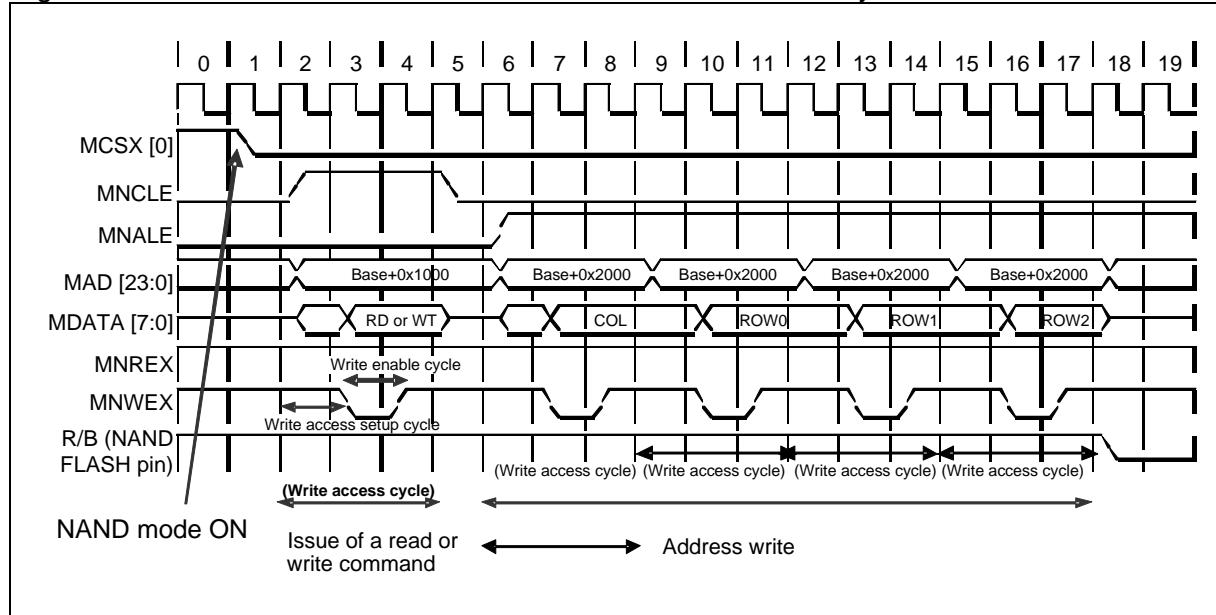
Figure 4-2 Waveforms of a serial word write/read access to 16-bit wide SRAM



4.3. Issue of an 8-bit NAND flash memory read/write command

Figure 4-3 shows waveforms of the issue of an 8-bit wide NAND flash memory read/write command (byte access).

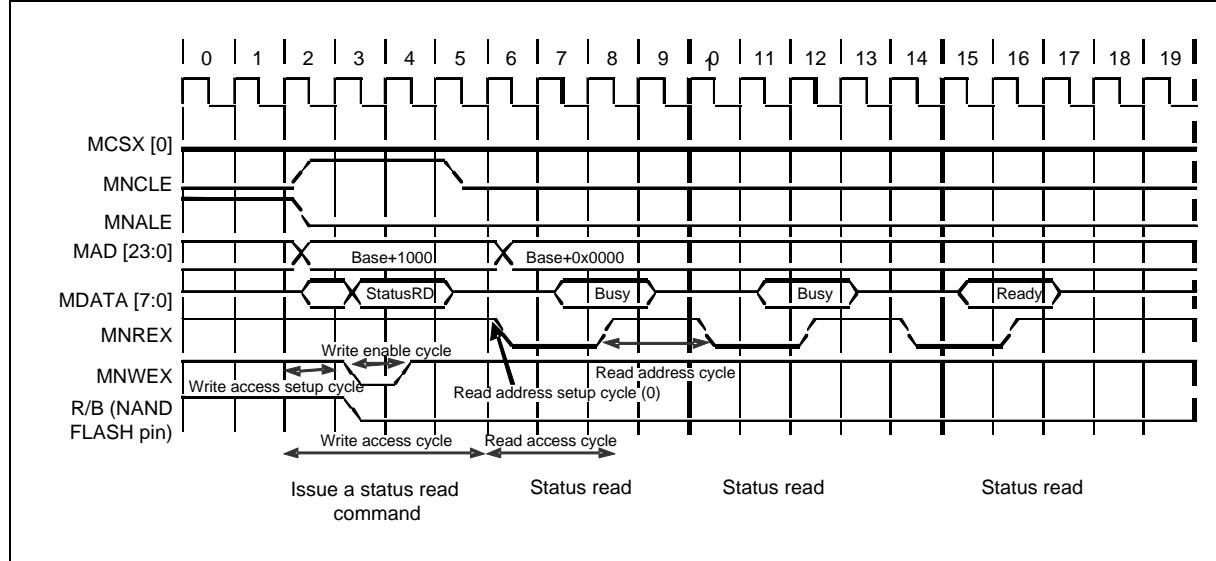
Figure 4-3 Waveforms of the issue of an 8-bit NAND flash memory read/write command



4.4. 8-bit NAND flash memory status read

Figure 4-4 shows waveforms of an 8-bit NAND flash memory status read (byte access).

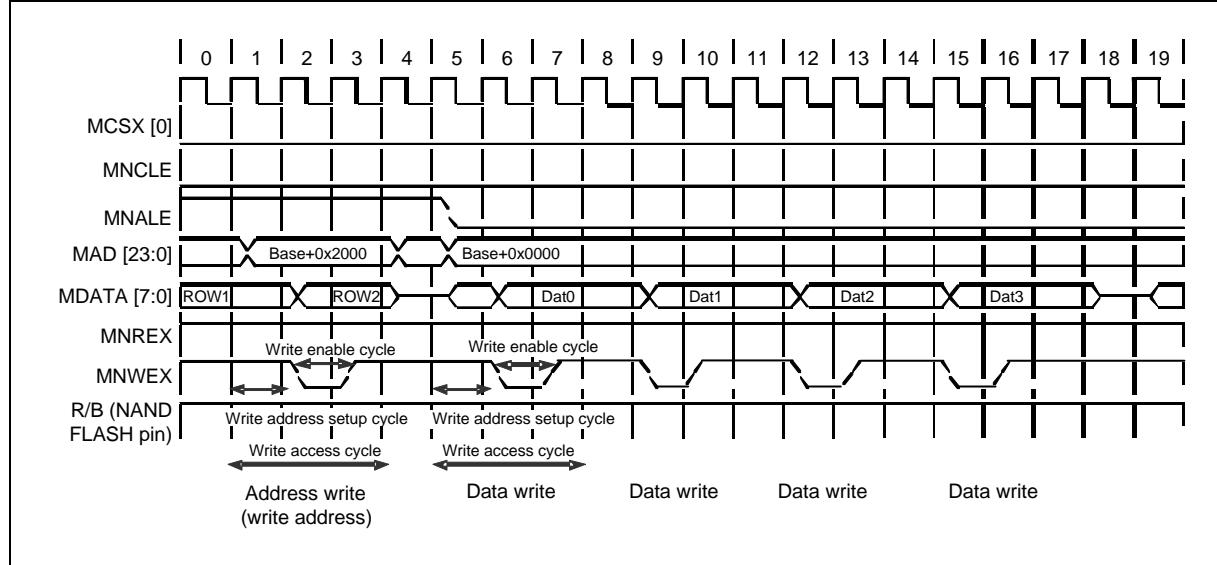
Figure 4-4 Waveforms of an 8-bit NAND flash memory status read



4.5. 8-bit NAND flash memory data write

Figure 4-5 shows waveforms of an 8-bit NAND flash memory data write.

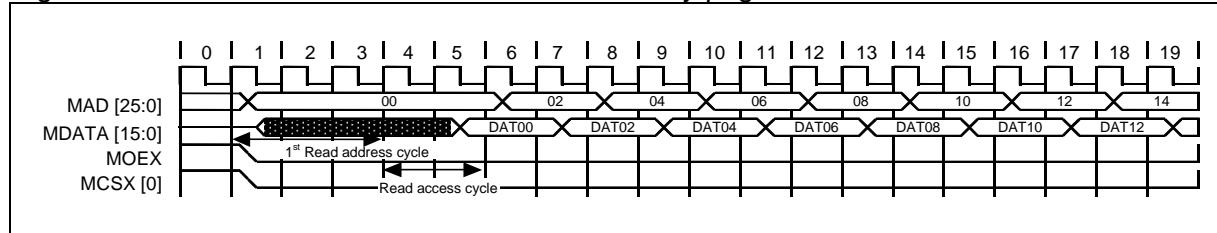
Figure 4-5 8-bit NAND flash memory data write



4.6. 16-bit NOR flash memory page read

Figure 4-6 shows waveforms of a 16-bit NOR flash memory page read.

Figure 4-6 Waveforms of a 16-bit NOR flash memory page read



5. Endianness and Valid Byte Lanes

The external bus interface supports little endian.

Table 5-1 shows correspondence between endian and valid byte lanes.

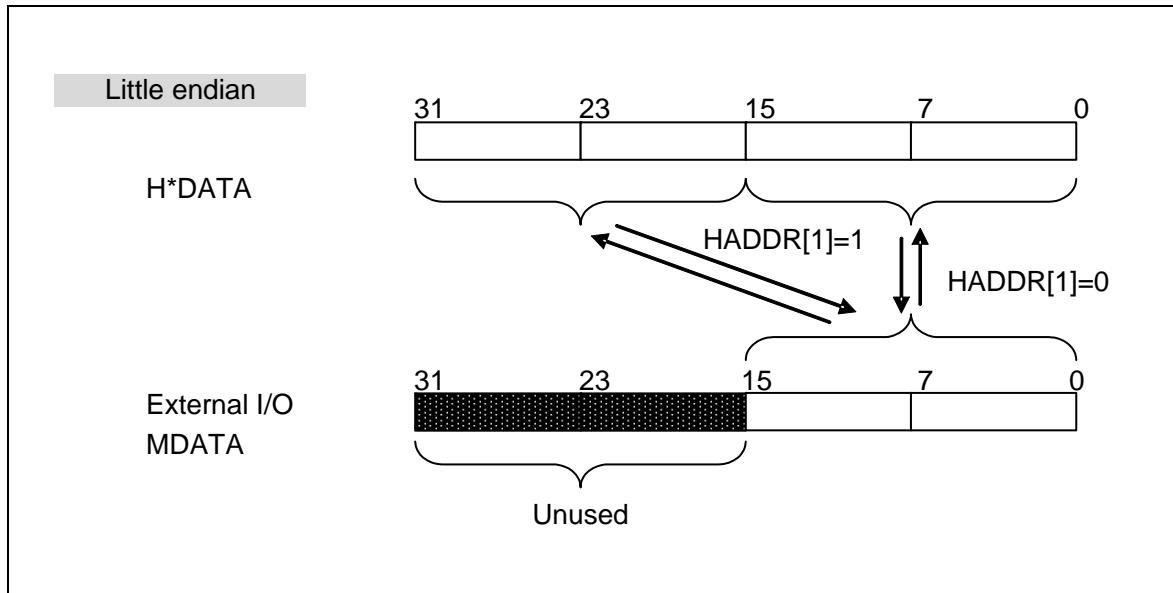
Table 5-1 Endianness and valid byte lanes

Endianness	Access size	Target width	Internal bus address	Valid byte lane	Correspondent internal bus data	MDQM [1:0]	MAD [1:0]	
Little endian	Word	8 bits	0	MDATA[7:0]	1 st : H*DATA[7:0]	0b10	0b00	
					2 nd : H*DATA[15:8]		0b01	
					3 rd : H*DATA[23:16]		0b10	
					4 th : H*DATA[31:24]		0b11	
	Half word	16 bits		MDATA[15:0]	1 st : H*DATA[15:0]	0b00	0b00	
				MDATA[15:0]	2 nd : H*DATA[31:16]		0b10	
		8 bits	0	MDATA[7:0]	1 st : H*DATA[7:0]	0b10	0b00	
					2 nd : H*DATA[15:8]		0b01	
	Byte	16 bits	2	MDATA[7:0]	1 st : H*DATA[23:16]	0b10	0b10	
					2 nd : H*DATA[31:24]		0b11	
		8 bits	0	MDATA[15:0]	H*DATA[15:0]	0b00	0b00	
			2	MDATA[15:0]	H*DATA[31:16]	0b00	0b10	

H*DATA: AHB I/O data

Example:

When a byte access is performed to a 16-bit wide target, only 2 bytes are valid according to endianness shown in the above table. Data in the internal bus is assigned to 16 bits according to endianness, and input or output.

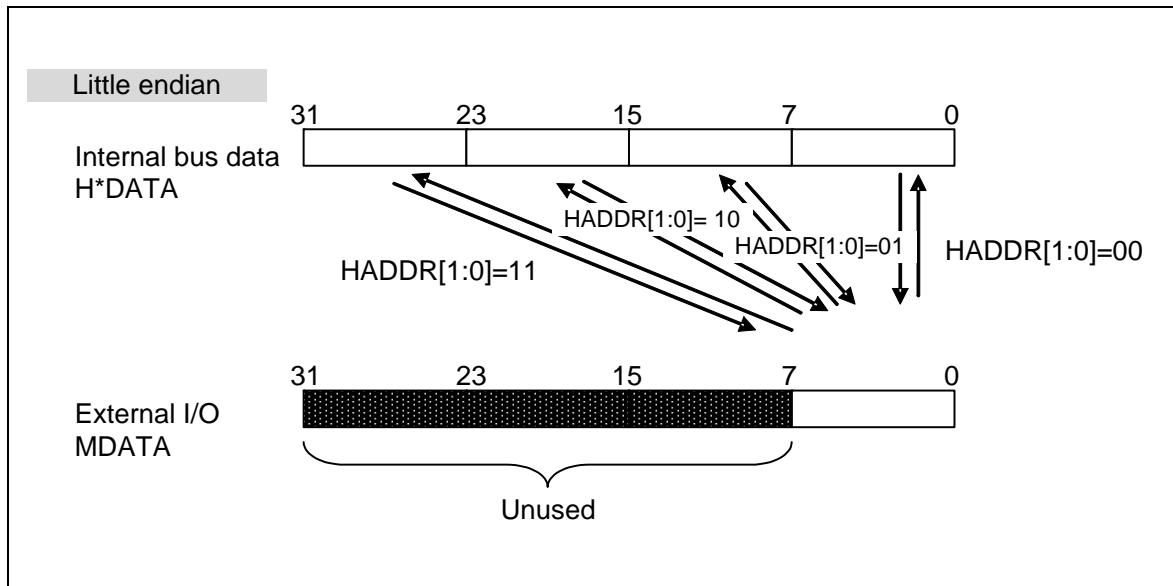


HADDR: AHB address input

As shown in the above figure, only HADDR[1] is involved in data assignment. This is also the case with during a half word access. During a word access, the external bus interface generates addresses automatically, and executes accesses twice corresponding to the HADDR[1] value. The dual access is a serial access in which the address changes while the chip select signal is LOW.

The following figure shows a correlation between internal data and external I/O when the 8-bit wide target is accessed.

Only a single byte is valid according to endianness.



For an 8-bit target, the HADDR[1:0] value determines the I/O data.

When a word or half word access is performed, the external bus interface automatically generates addresses, and operates at least a single access. This applies to every type of access independent of settings.

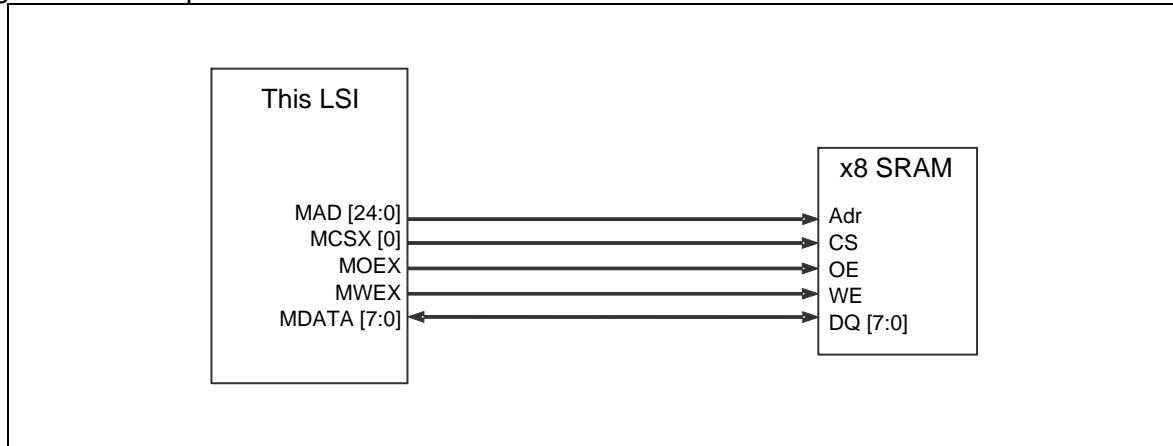
6. Connection Examples

This section provides an example of connections with external devices.

■ 8-bit SRAM

Figure 6-1 shows an example of connecting an 8-bit SRAM.

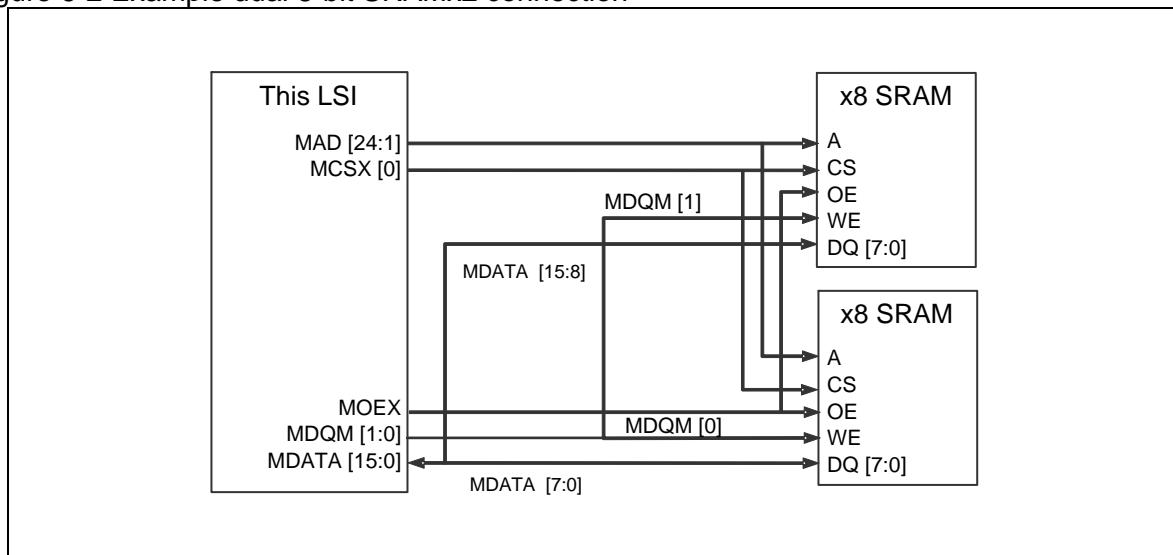
Figure 6-1 Example 8-bit SRAM connection



■ 8-bit SRAM × 2

Figure 6-2 shows an example of connecting two 8-bit SRAMs.

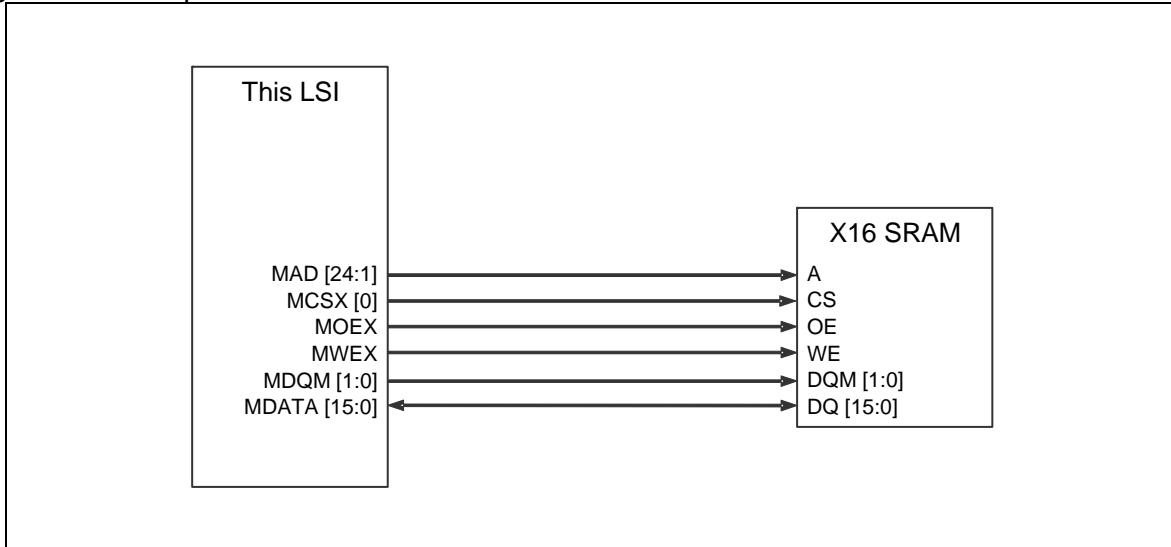
Figure 6-2 Example dual 8-bit SRAMx2 connection



■ 16-bit SRAM

Figure 6-3 shows an example of connecting a 16-bit SRAM.

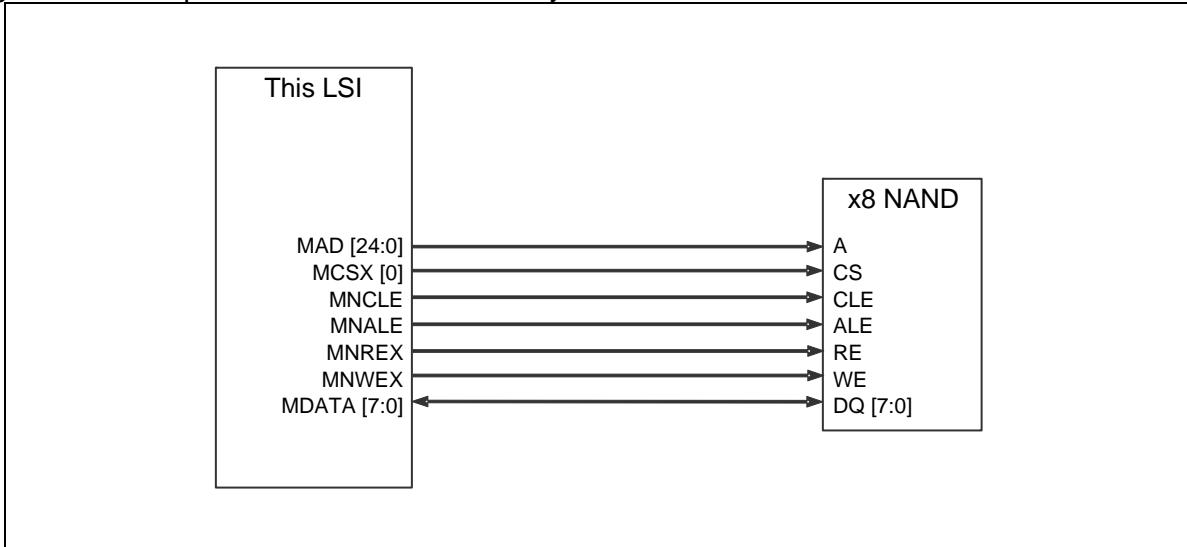
Figure 6-3 Example 16-bit SRAM connection



■ 8-bit NAND

Figure 6-4 shows an example of connecting an 8-bit NAND flash memory.

Figure 6-4 Example 8-bit NAND flash memory connection



7. Registers

This section explains the configuration and functions of registers used for the external bus interface.

The following explains the registers used for the external bus interface. The bit width of every register is 32. Each register can be accessed by the APB interface with 32-bit width (word). Write 0 to reserved areas.

A rewritten register value is not immediately applied to the operation. When you rewrite a register value during access to an external memory, the value is actually rewritten and reflected on the operation after the accessing process of the external memory is finished and then a time period of a few PCLK cycles passes. Read the register value to check that the rewriting has actually been applied to the operation.

Table 7-1 lists the registers.

Table 7-1 Register list

Abbreviation	Register name	See
MODE0 to MODE7	Mode Register 0 to Mode Register 7	7.1
TIM0 to TIM7	Timing Register 0 to Timing Register 7	7.2
AREA0 to AREA7	Area Register 0 to Area Register 7	7.3

7.1. Mode Register 0 to Mode Register 7

The following shows the configuration of the Mode Register (0 to 7).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-															
Initial value	-															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved							TEST	PAGE	NAND	WEOFF	RBMON	WDTH			
Attribute	-							R/W		R/W		R/W				
Initial value	-							0	0	0	0	0	0*1			

[bit 6] TEST

Always set this bit to "0". It must not be set to "1".

The read value is "0".

[bit 5] PAGE (PAGE access mode): NOR flash memory page access mode

This bit controls the mode of NOR flash memory page access.

In NOR flash memory page access mode, the first read access cycle (FRADC) setting can generate the first address cycle. Subsequently, the read access cycle (RACC) setting can continue the access until it reaches the 16-byte boundary.

When you select NOR flash memory page access mode, set the RBMON bit to "0" and the read access cycle (RADC) to "0".

Bit	Description
0	NOR flash memory page access mode is turned OFF (Initial value)
1	NOR flash memory page access mode is turned ON

[bit 4] NAND: NAND flash memory mode

This bit controls the mode used to connect with a NAND flash memory.

To enable the access to a NAND flash memory, set this bit to "1".

In NAND flash memory mode, the corresponding MCSX is fixed to LOW and, subsequently, the pin dedicated to the NAND flash memory is used during the access. If this bit is set to "0" while the NAND flash memory is unused, then MCSX is fixed to HIGH, enabling the NAND flash memory to maintain a low power consumption state.

Bit	Description
0	NAND flash memory page mode is turned OFF (Initial value)
1	NAND flash memory mode is turned ON

[bit 3] WEOFF (WEX OFF): Write Enable OFF

This bit can disable the write enable signal (MWEX) operation.

When the byte mask signal (MDQM) is used as a device write enable signal, disabling unnecessary MWEX operation can reduce current consumption. When this bit is set to disable, MWEX is fixed to HIGH.

Bit	Description
0	Enable [Initial value]
1	Disable

[bit 2] RBMON: Read Byte Mask ON

This bit can enable the byte mask signal (MDQM) for read access.

The setting controls the output of unnecessary data from a device for which the byte mask signal is enabled. This is helpful to reduce power consumption.

Bit	Description
0	Disable [Initial value]
1	Enable

[bit 1:0] WDTH: Data Width

These bits specify the data bit width of a device to be connected.

Bit 1	Bit 0	Description
0	0	8 bits [Initial value]
0	1	16 bits
1	0	Setting disabled
1	1	Setting disabled

<Note>

If you write a disabled value to a TEST or WDTH bit, operation of the external bus interface is not guaranteed.

7.2. Timing Register 0 to Timing Register 7

The following shows the configuration of the Timing Register (0 to 7).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field					WIDLC			WWEC			WADC			WACC		
Attribute																
Initial value					0000			0101			0101			1111		
R/W																
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field					RIDLC			FRADC			RADC			RACC		
Attribute																
Initial value					1111			0000			0000			1111		
R/W																

[bit 31:28] WIDLC: Write Idle Cycle

These bits set the number of idle cycles after write access.

Bit 31	Bit 30	Bit 29	Bit 28	Description
0	0	0	0	1 cycle [Initial value]
⋮				⋮
1	1	1	1	16 cycles

[bit 27:24] WWEC: Write Enable Cycle

These bits set the number of assert cycles of write enable.

The setting of these bits affects the byte mask signal (MDQM).

Bit 27	Bit 26	Bit 25	Bit 24	Description
0	0	0	0	1 cycle
⋮				⋮
0	1	0	1	6 cycles [Initial value]
⋮				⋮
1	1	1	0	15 cycles
1	1	1	1	Setting disabled

[bit 23:20] WADC: Write Address Setup cycle

These bits set the number of setup cycles of write address.

The address is output during the cycle set by these bits, but a write enable signal is not asserted until the set cycle starts.

Bit 23	Bit 22	Bit 21	Bit 20	Description
0	0	0	0	1 cycle
		⋮		⋮
0	1	0	1	6 cycles [Initial value]
		⋮		⋮
1	1	1	0	15 cycles
1	1	1	1	Setting disabled

[bit 19:16] WACC: Write Access Cycle

These bits set the number of cycles required for write access.

The address remains unchanged during the cycle set by these bits.

The number of cycles set by these bits must be equal to or more than the sum of the address setup cycle (WADC) and the write enable cycle (WWEC).

Bit 19	Bit 18	Bit 17	Bit 16	Description
0	0	0	0	Setting disabled
0	0	0	1	Setting disabled
0	0	1	0	3 cycles
		⋮		⋮
1	1	1	1	16 cycles [Initial value]

[bit 15:12] RIDLC: Read Idle Cycle

These bits set the number of idle cycles after read access.

They are used to avoid data collision caused by a write access occurring immediately after a read access.

Bit 15	Bit 14	Bit 13	Bit 12	Description
0	0	0	0	1 cycle
		⋮		⋮
1	1	1	1	16 cycles [Initial value]

[bit 11:8] FRADC: First Read Address Cycle

These bits exclusively set a NOR flash memory that supports page mode access.

They set the initial wait time of the address during read access to flash memory.

The address for the set cycle is retained only during the first cycle. After the first access, the access is performed according to the number of cycles set by RACC.

In page mode access, MCSX and MOEX are asserted simultaneously.

If values other than "0" is set to these bits, set the read access setup cycle (RADC) to "0".

Bit 11	Bit 10	Bit 9	Bit 8	Description
0	0	0	0	0 cycles [Initial value]
:				:
1	1	1	1	15 cycles

[bit 7:4] RADC: Read Address Setup cycle

These bits set the number of setup cycles of read address.

Within the read address setup cycle, MCSX and address are asserted but MOEX is not asserted. If "0" is set to any of these bits, MOEX and MCSX are always asserted.

The set value must be less than the number of read access cycles. (RADC < RACC).

When using NOR flash memory page access mode, set these bits to 0b0000.

If the access size is more than the target width, or if a device such as NAND flash memory needs to switch HIGH and LOW of read enable (MOEX or MNREX), set these bits to 0b0001 or a higher value.

Bit 7	Bit 6	Bit 5	Bit 4	Description
0	0	0	0	0 cycles [Initial value]
:				:
1	1	1	1	15 cycles

[bit 3:0] RACC: Read Access Cycle

These bits set the number of cycles required for read access.

The address remains unchanged during the cycle specified by these bits, and the data is captured at the last cycle.

Bit 3	Bit 2	Bit 1	Bit 0	Description
0	0	0	0	1 cycle
:				:
1	1	1	1	16 cycles [Initial value]

<Notes>

- If you write a disabled value to a WWEC, WADC or WACC bit, operation of the external bus interface is not guaranteed.
 - In NAND flash memory mode, the MNWEX and MNREX timings are set by the timing registers as is the case with MWEX and MOEX.
-

7.3. Area Register 0 to Area Register 7

The following shows the configuration of the Area Register (0 to 7).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved												MASK			
Attribute	-												R/W			
Initial value	-												001111 (16MB width)			
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								ADDR							
Attribute	-								R/W							
Initial value	-								(from MCSX[0])							
	00000000, 00010000, 00100000,								00110000, 01000000, 01010000,							
	01100000, 01110000 *1															

[bit 22:16] MASK: address mask

These bits set the value to mask the value set in ADDR.

If "1" is set as a mask value, the external bus interface masks each of the internal bus and ADDR according to the value, and compares the masking results. If the results are matched, the external bus interface accesses the MCSX signal.

Example:

ADDR = 0b00001000

MASK = 0b 0000011

- An example where a device is selected
Internal bus address (External interface address) AD = 0x10900000

Masking

ADDR & (!MASK) =0b00001000 }
AD [27:20] & (!MASK) =0b00001000 } Matched. The device is selected

- An example where a device is not selected
Internal bus address (External interface address) AD = 0x10C00000

Masking

ADDR & (!MASK) =0b00001000 }
AD [27:20] & (!MASK) =0b00001100 } Not matched. The device is not selected

An area size is selected for masking. In the example, 0x10800000~0x10B00000 (4 MB) is selected.

A bit set to "1" for masking is lost during masking process. The bit is disabled even if it is set in ADDR. In the example, if LSB is set to "1" (ADDR=0b00001001), it is disabled during masking and the same address area is selected. The following example indicates the relationship between the mask settings and the address area size.

0b0000000 → 1MB	0b0000111 → 8MB
0b0000001 → 2MB	0b0001111 → 16MB
0b0000011 → 4MB	0b0011111 → 32MB

[bit 7:0] ADDR: Address

These bits specify the address to set the corresponding MCSX area.

The address is in the fixed 256 MB area assigned to the SRAM/flash memory interface.

The address specified by bits 7:0 corresponds to the internal address [27:20].

<Note>

No address areas must overlap other areas.

If unused MCSX is accessed, operation of the external bus interface is also not guaranteed.

Chapter: Debug Interface

This chapter explains the function and operation of the debug interface.

1. Overview
2. Pin Description

1. Overview

This series contains a Serial Wire JTAG Debug Port (SWJ-DP).

Connecting an ICE to the SWJ-DP allows system debugging.

This series also contains an Embedded Trace Macro Cell (ETM) for tracing instructions and a Trace Port Interface Unit (TPIU) that controls trace data.

This section describes the functions of the pins to be used for debugging.

For details on the SWJ-DP, ETM, TPIU and system debug, see "Cortex-M3 Technical Reference Manual".

■ Features

Five pins are assigned to the SWJ-DP.

These five pins are initially dedicated to the JTAG. It is possible to change their functions to the serial wire debug mode.

It is possible to output instruction trace by selecting it from 4-bit trace data (TRACED0-3) and asynchronous trace data (SWO).

2. Pin Description

This section describes pins.

- 2.1 Pins for Debug Purposes
- 2.2 ETM Pins
- 2.3 Functions Initially Assigned to Pins
- 2.4 Internal Pull-Ups of JTAG Pins

2.1. Pins for Debug Purposes

Five pins (TRSTX, TCK, TMS, TDI, and TDO) are assigned to the JTAG and two pins (SWCLK and SWDIO) are assigned to the serial wire. In addition, a Serial Wire Viewer signal (SWO) that outputs trace data is assigned.

TMS is shared with SWDIO, TCK is shared with SWCLK, and TDO is shared with SWO.

The following provides a list of pin functions in each debug mode.

Table 2-1 SWJ-DP pin functions in debug mode

Pin	JTAG	Serial Wire/Trace
TCK/SWCLK	TCK (JTAG Clock signal)	SWCLK (Serial Wire Clock signal)
TMS/SWDIO	TMS (JTAG State Mode signal)	SWDIO (Serial Wire Data Input/Output signal)
TDI	TDI (JTAG Data Input signal)	-
TDO/SWO	TDO (JTAG Data Output signal)	SWO (Serial Wire Viewer Output signal)
TRSTX	TRSTX (active-LOW JTAG Reset signal)	-

2.2. ETM Pins

The ETM is assigned four trace outputs (TRACED0, TRACED1, TRACED2, and TRACED3) and one clock (TRACECLK).

The following provides a list of pin functions in each debug mode.

Table 2-2 Trace pin functions in debug mode

Pin	Trace
TRACED0	Synchronous Trace Data Output signal
TRACED1	Synchronous Trace Data Output signal
TRACED2	Synchronous Trace Data Output signal
TRACED3	Synchronous Trace Data Output signal
TRACECLK	Trace Clock signal

2.3. Functions Initially Assigned to Pins

The 10 pins - five JTAG pins and five ETM trace pins - are used also as GPIO. Five JTAG pins (TRSTX, TCK, TMS, TDI, and TDO) are initially dedicated to debug function, whereas five ETM pins (TRACED0, TRACED1, TRACED2, TRACED3, and TRACECLK) are not initially dedicated to that.

When using this series, please configure these ETM pins to provide the debug function.

Note: For details on how to set the debug function, see Chapter "I/O Ports"

The following table provides initial states after resets are cleared and the functions that can be changed by setting PFRs (Port function registers).

Note: For details on the PFRs, see Chapter "I/O Ports".

Table 2-3 Functions initially assigned to pins for debugging purposes and change of function

		Initially assigned pin function	Change of function by setting the PFR
JTAG pins	TCK/SWCLK	TCK	GPIO
	TMS/SWDIO	TMS	GPIO
	TDI	TDI	GPIO
	TDO/SWO	TDO	GPIO
	TRSTX	TRSTX	GPIO
ETM pins	TRACED0	GPIO	TRACED0
	TRACED1	GPIO	TRACED1
	TRACED2	GPIO	TRACED2
	TRACED3	GPIO	TRACED3
	TRACECLK	GPIO	TRACECLK

2.4. Internal Pull-Ups of JTAG Pins

As specified by the IEEE Standard, this series provides the JTAG pins that have internal pull-ups.

If the functions of the pins are changed from JTAG to GPIO, the user can control pull-ups by setting the appropriate registers in the GPIO.

Table 2-4 Enabled or disabled state of internal pull-ups of JTAG pins

	Pull-up with JTAG pins enabled *1
TCK/SWCLK	Enabled
TMS/SWDIO	Enabled
TDI	Enabled
TDO/SWO	Enabled *2
TRSTX	Enabled

*1 : Pull-up is enabled on reset.

*2 : Pull-up is disabled on output.

APPENDIXES

This chapter shows the register map.

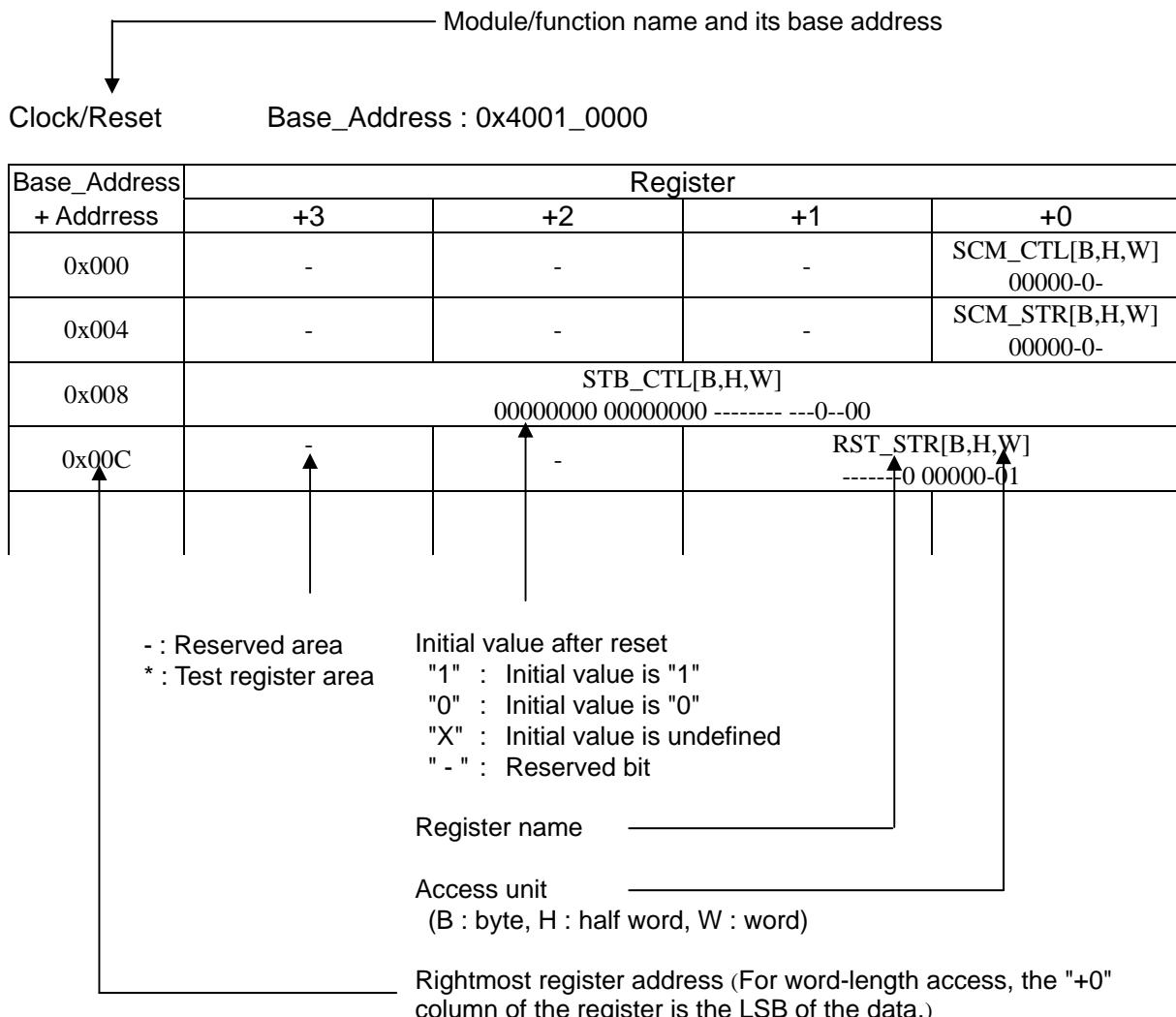
1. Register Map

CODE: 9BFREGMAP-E01.2

1. Register Map

Register map is shown on the table every module/function.

[How to read the each table]



Module/function name and its base address

Clock/Reset Base_Address : 0x4001_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[B,H,W] 00000-0-
0x004	-	-	-	SCM_STR[B,H,W] 00000-0-
0x008	STB_CTL[B,H,W] 00000000 00000000 ----- ---0--0			
0x00C	-	-	-	RST_STR[B,H,W] -----0 00000-01

- : Reserved area Initial value after reset
 * : Test register area "1" : Initial value is "1"
 "0" : Initial value is "0"
 "X" : Initial value is undefined
 "- -" : Reserved bit

Register name _____

Access unit _____
(B : byte, H : half word, W : word)

Rightmost register address (For word-length access, the "+0" column of the register is the LSB of the data.)

<Notes>

- The register table is represented in the little-endian.
- When performing a data access, the addresses should be as below.
 - Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")
 - Half word access : Address should be multiples of 2 (least significant bit should be "0x0")
 - Byte access : -
- Do not access the test register area.

FLASH_IF Base_Address : 0x4000_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000		FASZR[B,H,W] -----10		
0x004		FRWTR[B,H,W] -----10		
0x008		FSTR[B,H,W] -----0X		
0x00C		*		
0x010		FSYNDN[B,H,W] -----000		
0x014 - 0x0FF	-	-	-	-
0x100		CRTRMM[B,H,W] -----XX XXXXXXXX		
0x104-0xFFFF	-	-	-	-

Clock/Reset Base_Address : 0x4001_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W] 00000-0-
0x004	-	-	-	SCM_STR[W] 00000-0-
0x008	STB_CTL[W] 00000000 00000000 -----0--0			
0x00C	-	-	RST_STR[W] -----0 0000-01	
0x010	-	-	-	BSC_PSR[W] ----000
0x014	-	-	-	APBC0_PSR[W] ----00
0x018	-	-	-	APBC1_PSR[W] 1--0--00
0x01C	-	-	-	APBC2_PSR[W] 1--0--00
0x020	-	-	-	SWC_PSR[W] X----00
0x024 - 0x027	-	-	-	-
0x028	-	-	-	TTC_PSR[W] ----0
0x02C - 0x02F	-	-	-	-
0x030	-	-	-	CSW_TMR[W] -000000
0x034	-	-	-	PSW_TMR[W] ---0-000
0x038	-	-	-	PLL_CTL1[W] 00000000
0x03C	-	-	-	PLL_CTL2[W] ---00000
0x040	-	-	CSV_CTL[W] -111--00 -----11	
0x044	-	-	-	CSV_STR[W] -----00
0x048	-	-	FCSWH_CTL[W] 11111111 11111111	
0x04C	-	-	FCSWL_CTL[W] 00000000 00000000	
0x050	-	-	FCSWD_CTL[W] 00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W] 0-0----
0x058 - 0x05F	-	-	-	-
0x060	-	-	-	INT_ENR[W] --0--000
0x064	-	-	-	INT_STR[W] --0--000
0x068	-	-	-	INT_CLR[W] --0--000
0x06C - 0xFFFF	-	-	-	-

HW WDT Base_Address : 0x4001_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WDG_LDR[W] 00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	WDG_CTL[W] - - - -----11			
0x00C	WDG_ICL[W] - - - XXXXXXXX			
0x010	WDG_RIS[W] - - - -----0			
0x014 - 0xBFF	-	-	-	-
0xC00	WDG_LCK[W] 00000000 00000000 00000000 00000001			
0xC04 - 0xFFFF	-	-	-	-

SW WDT Base_Address : 0x4001_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WdogLoad[W] 11111111 11111111 11111111 11111111			
0x004	WdogValue[W] 11111111 11111111 11111111 11111111			
0x008	WdogControl[W] - - - -----00			
0x00C	WdogIntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	WdogRIS[W] - - - -----0			
0x014 - 0xBFF	-	-	-	-
0xC00	WdogLock[W] 00000000 00000000 00000000 00000000			
0xC04 - 0xFFFF	-	-	-	-

Dual_Timer Base_Address : 0x4001_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	Timer1Load[W] 00000000 00000000 00000000 00000000			
0x004	Timer1Value[W] 11111111 11111111 11111111 11111111			
0x008	Timer1Control[W] ----- 00100000			
0x00C	Timer1IntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W] ----- 0			
0x014	Timer1MIS[W] ----- 0			
0x018	Timer1BGLoad[W] 00000000 00000000 00000000 00000000			
0x020	Timer2Load[W] 00000000 00000000 00000000 00000000			
0x024	Timer2Value[W] 11111111 11111111 11111111 11111111			
0x028	Timer2Control[W] ----- 00100000			
0x02C	Timer2IntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W] ----- 0			
0x034	Timer2MIS[W] ----- 0			
0x038	Timer2BGLoad[W] 00000000 00000000 00000000 00000000			
0x040 - 0xFFFF	-	-	-	-

MFT unit0 Base_Address : 0x4002_0000
 MFT unit1 Base_Address : 0x4002_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	OCCP0[H,W] 00000000 00000000	
0x004	-	-	OCCP1[H,W] 00000000 00000000	
0x008	-	-	OCCP2[H,W] 00000000 00000000	
0x00C	-	-	OCCP3[H,W] 00000000 00000000	
0x010	-	-	OCCP4[H,W] 00000000 00000000	
0x014	-	-	OCCP5[H,W] 00000000 00000000	
0x018	-	-	OCSB10[B,H,W] -110--00 00001100	OCSA10[B,H,W]
0x01C	-	-	OCSB32[B,H,W] -110--00	OCSA32[B,H,W] 00001100
0x020	-	-	OCSB54[B,H,W] -110--00	OCSA54[B,H,W] 00001100
0x024	-	-	OCSC[B,H,W] --000000	-
0x028	-	-	TCCP0[H,W] 11111111 11111111	
0x02C	-	-	TCDT0[H,W] 00000000 00000000	
0x030	-	-	TCSA0[B,H,W] 000---00 01000000	
0x034	-	-	TCSB0[B,H,W] -----000	
0x038	-	-	TCCP1[H,W] 11111111 11111111	
0x03C	-	-	TCDT1[H,W] 00000000 00000000	
0x040	-	-	TCSA1[B,H,W] 000---00 01000000	
0x044	-	-	TCSB1[B,H,W] -----000	
0x048	-	-	TCCP2[H,W] 11111111 11111111	
0x04C	-	-	TCDT2[H,W] 00000000 00000000	
0x050	-	-	TCSA2[B,H,W] 000---00 01000000	
0x054	-	-	TCSB2[B,H,W] -----000	
0x058	-	-	OCFS32[B,H,W] 00000000	OCFS10[B,H,W] 00000000
0x05C	-	-	-	OCFS54[B,H,W] 00000000
0x060	-	-	ICFS32[B,H,W] 00000000	ICFS10[B,H,W] 00000000
0x064	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x068	-	-	ICCP0[H,W] XXXXXXXX XXXXXXXX	
0x06C	-	-	ICCP1[H,W] XXXXXXXX XXXXXXXX	
0x070	-	-	ICCP2[H,W] XXXXXXXX XXXXXXXX	
0x074	-	-	ICCP3[H,W] XXXXXXXX XXXXXXXX	
0x078	-	-	ICSB10[B,H,W] -----00	ICSA10[B,H,W] 00000000
0x07C	-	-	ICSB32[B,H,W] -----00	ICSA32[B,H,W] 00000000
0x080	-	-	WFTM10[H,W] 00000000 00000000	
0x084	-	-	WFTM32[H,W] 00000000 00000000	
0x088	-	-	WFTM54[H,W] 00000000 00000000	
0x08C	-	-	WFSA10[H,W] ---00000 000000	
0x090	-	-	WFSA32[H,W] ---00000 000000	
0x094	-	-	WFSA54[H,W] ---00000 000000	
0x098	-	-	WFIR[H,W] 00000000 0000--00	
0x09C	-	-	NZCL[H,W] ----- --0000	
0x0A0	-	-	ACCP0[H,W] 00000000 00000000	
0x0A4	-	-	ACCPDN0[H,W] 00000000 00000000	
0x0A8	-	-	ACCP1[H,W] 00000000 00000000	
0x0AC	-	-	ACCPDN1[H,W] 00000000 00000000	
0x0B0	-	-	ACCP2[H,W] 00000000 00000000	
0x0B4	-	-	ACCPDN2[H,W] 00000000 00000000	
0x0B8	-	-	-	ACSB[B,H,W] -000-111
0x0BC	-	-	-	ACSA[B,H,W] --000000 --000000
0x0C0	-	-	-	ATSA[H,W] --000000 --000000
0x0C4 - 0xFFFF	-	-	-	-

PPG

Base_Address : 0x4002_4000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	TTCR0 [B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0 [B,H,W] 00000000	-
0x00C	-	-	-	COMP2 [B,H,W] 00000000
0x010	-	-	COMP4 [B,H,W] 00000000	-
0x014	-	-	-	COMP6 [B,H,W] 00000000
0x018 - 0x01F	-	-	-	-
0x020	-	-	TTCR1 [B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1 [B,H,W] 00000000	-
0x02C	-	-	-	COMP3 [B,H,W] 00000000
0x030	-	-	COMP5 [B,H,W] 00000000	-
0x034	-	-	-	COMP7 [B,H,W] 00000000
0x038 - 0x0FF	-	-	-	-
0x100	-	-	TRG [B,H,W] 00000000 00000000	
0x104	-	-	REVC [B,H,W] 00000000 00000000	
0x108 - 0x1FF	-	-	-	-
0x200	-	-	PPGC0 [B,H,W] 0000000	PPGC1 [B,H,W] 0000000
0x204	-	-	PPGC2 [B,H,W] 0000000	PPGC3 [B,H,W] 0000000
0x208	-	-	PRLH0 [B,H,W] XXXXXXXX	PRLL0 [B,H,W] XXXXXXXX
0x20C	-	-	PRLH1 [B,H,W] XXXXXXXX	PRLL1 [B,H,W] XXXXXXXX
0x210	-	-	PRLH2 [B,H,W] XXXXXXXX	PRLL2 [B,H,W] XXXXXXXX
0x214	-	-	PRLH3 [B,H,W] XXXXXXXX	PRLL3 [B,H,W] XXXXXXXX
0x218	-	-	-	GATEC0 [B,H,W] --00---00
0x21C - 0x23F	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x240	-	-	PPGC4 [B,H,W] 0000000	PPGC5 [B,H,W] 0000000
0x244	-	-	PPGC6 [B,H,W] 0000000	PPGC7 [B,H,W] 0000000
0x248	-	-	PRLH4 [B,H,W] XXXXXXXXX	PRLL4 [B,H,W] XXXXXXXXX
0x24C	-	-	PRLH5 [B,H,W] XXXXXXXXX	PRLL5 [B,H,W] XXXXXXXXX
0x250	-	-	PRLH6 [B,H,W] XXXXXXXXX	PRLL6 [B,H,W] XXXXXXXXX
0x254	-	-	PRLH7 [B,H,W] XXXXXXXXX	PRLL7 [B,H,W] XXXXXXXXX
0x258	-	-	-	GATEC4 [B,H,W] ----00
0x25C - 0x27F	-	-	-	-
0x280	-	-	PPGC8 [B,H,W] 0000000	PPGC9 [B,H,W] 0000000
0x284	-	-	PPGC10 [B,H,W] 0000000	PPGC11 [B,H,W] 0000000
0x288	-	-	PRLH8 [B,H,W] XXXXXXXXX	PRLL8 [B,H,W] XXXXXXXXX
0x28C	-	-	PRLH9 [B,H,W] XXXXXXXXX	PRLL9 [B,H,W] XXXXXXXXX
0x290	-	-	PRLH10 [B,H,W] XXXXXXXXX	PRLL10 [B,H,W] XXXXXXXXX
0x294	-	-	PRLH11 [B,H,W] XXXXXXXXX	PRLL11 [B,H,W] XXXXXXXXX
0x298	-	-	-	GATEC8 [B,H,W] --00--00
0x29C - 0x2BF	-	-	-	-
0x2C0	-	-	PPGC12 [B,H,W] 0000000	PPGC13 [B,H,W] 0000000
0x2C4	-	-	PPGC14 [B,H,W] 0000000	PPGC15 [B,H,W] 0000000
0x2C8	-	-	PRLH12 [B,H,W] XXXXXXXXX	PRLL12 [B,H,W] XXXXXXXXX
0x2CC	-	-	PRLH13 [B,H,W] XXXXXXXXX	PRLL13 [B,H,W] XXXXXXXXX
0x2D0	-	-	PRLH14 [B,H,W] XXXXXXXXX	PRLL14 [B,H,W] XXXXXXXXX
0x2D4	-	-	PRLH15 [B,H,W] XXXXXXXXX	PRLL15 [B,H,W] XXXXXXXXX
0x2D8	-	-	-	GATEC12 [B,H,W] ----00
0x2DC - 0xFFFF	-	-	-	-

Base Timer ch0	Base Address : 0x4002_5000
Base Timer ch1	Base Address : 0x4002_5040
Base Timer ch2	Base Address : 0x4002_5080
Base Timer ch3	Base Address : 0x4002_50C0
Base Timer ch4	Base Address : 0x4002_5200
Base Timer ch5	Base Address : 0x4002_5240
Base Timer ch6	Base Address : 0x4002_5280
Base Timer ch7	Base Address : 0x4002_52C0

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL [H,W] XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF [H,W] XXXXXXXX XXXXXXXX	
0x008	-	-	TMR [H,W] 00000000 00000000	
0x00C	-	-	TMCR [B,H,W] -0000000 00000000	
0x010	-	-	TMCR2 [B,H,W] -----0	STC [B,H,W] 0000-000
0x014 - 0x03F	-	-	-	-

IO Selector for ch0-ch3 (Base Timer)

Base Address : 0x4002_5100

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSEL0123 [B,H,W] 00000000	-
0x004-0xFFFF	-	-	-	-

IO Selector for ch4-ch7(Base Timer)

Base Address : 0x4002_5300

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSEL4567 [B,H,W] 00000000	-
0x004-0xFFFF	-	-	-	-

Software-based Simulation Starup(Base Timer)

Base Addrsee : 0x4002_5400

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000-0xBFB	-	-	-	-
0xBFC	-	-	BTSSSR [B,H,W] XXXXXXXX XXXXXXXX	

QPRC ch0 Base Address : 0x4002_6000
 QPRC ch1 Base Address : 0x4002_6040

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	QPCR [H,W] 00000000 00000000	
0x0004	-	-	QRCR [H,W] 00000000 00000000	
0x0008	-	-	QPCCR [H,W] 00000000 00000000	
0x000C	-	-	QPRCR [H,W] 00000000 00000000	
0x0010	-	-	QMPPR [H,W] 11111111 11111111	
0x0014	-	-	QICRH [B,H,W] --000000	QICRL [B,H,W] --000000
0x0018	-	-	QCRH [B,H,W] 00000000	QCRL [B,H,W] 00000000
0x001C	-	-	QEGR [B,H,W] -----000	
0x0020 - 0x003F	-	-	-	-

10bit A/DC unit0 Base_Address : 0x4002_7000
 10bit A/DC unit1 Base_Address : 0x4002_7100
 10bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W] 000-0000	ADSR[B,H,W] 00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W] 1000-000	SFNS[B,H,W] ---0000
0x00C	-	-	SCFD[B,H,W] XXXXXXXX XX-XXXXX	
0x010	-	-	SCIS3[B,H,W] 00000000	SCIS2[B,H,W] 00000000
0x014	-	-	SCIS1[B,H,W] 00000000	SCIS0[B,H,W] 00000000
0x018	-	-	PCCR[B,H,W] 1000-000	PFNS[B,H,W] --XX--00
0x01C	-	-	PCFD[B,H,W] XXXXXXXX XXXXXXXX	
0x020	-	-	-	PCIS[B,H,W] 00000000
0x024	-	-	CMPD[B,H,W] 00000000	CMPCR[B,H,W] 00000000
0x028	-	-	ADSS3[B,H,W] 00000000	ADSS2[B,H,W] 00000000
0x02C	-	-	ADSS1[B,H,W] 00000000	ADSS0[B,H,W] 00000000
0x030	-	-	ADST0[B,H,W] 00000000	ADST1[B,H,W] 00000000
0x034	-	-	-	ADCT[B,H,W] ----111
0x038	-	-	SCTSL[B,H,W] ----0000	PRTSL[B,H,W] ----0000
0x03C - 0xFF	-	-	-	-

12bit A/DC unit0 Base_Address : 0x4002_7000
 12bit A/DC unit1 Base_Address : 0x4002_7100
 12bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W] 000-0000	ADSR[B,H,W] 00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W] 1000-000	SFNS[B,H,W] ---0000
0x00C	SCFD[B,H,W] XXXXXXXX XXXX----0001--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W] 00000000	SCIS2[B,H,W] 00000000
0x014	-	-	SCIS1[B,H,W] 00000000	SCIS0[B,H,W] 00000000
0x018	-	-	PCCR[B,H,W] 1000-000	PFNS[B,H,W] --XX--00
0x01C	PCFD[B,H,W] XXXXXXXX XXXX----1-XXX ---XXXX			
0x020	-	-	-	PCIS[B,H,W] 00000000
0x024	CMPD[B,H,W] 00000000 00-----		-	CMPCR[B,H,W] 00000000
0x028	-	-	ADSS3[B,H,W] 00000000	ADSS2[B,H,W] 00000000
0x02C	-	-	ADSS1[B,H,W] 00000000	ADSS0[B,H,W] 00000000
0x030	-	-	ADST0[B,H,W] 00010000	ADST1[B,H,W] 00010000
0x034	-	-	-	ADCT[B,H,W] ----111
0x038	-	-	SCTSL[B,H,W] ----0000	PRTSL[B,H,W] ---0000
0x03C	-	-	-	ADCEN[B,H,W] -----00
0x040 - 0xFF	-	-	-	-

CR Trim Base_Address : 0x4002_E000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W] -----01
0x004	-	-	MCR_FTRM[B,H,W] -----01 10000000	
0x008	-	-	-	*
0x00C	MCR_RLR[W] 00000000 00000000 00000000 00000001			
0x010 - 0xFFFF	-	-	-	-

EXTI Base_Address : 0x4003_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	ENIR[B,H,W] 00000000 00000000	
0x0004	-	-	EIRR[B,H,W] XXXXXXXX XXXXXXXX	
0x0008	-	-	EICL[B,H,W] 11111111 11111111	
0x000C	ELVR[R/W] 00000000 00000000 00000000 00000000			
0x0010	-	-	-	-
0x0014	-	-	-	NMIRR[B,H,W] -----0
0x0018	-	-	-	NMICL[B,H,W] -----1
0x001C	-	-	-	-
0x020 - 0xFFFF	-	-	-	-

INT-Req. READ Base_Address : 0x4003_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	DRQSEL[B,H,W] 00000000 00000000 00000000 00000000			
0x004 - 0x00C	-	-	-	-
0x010	EXC02MON[B,H,W] -----00			
0x014	IRQ00MON[B,H,W] -----0			
0x018	IRQ01MON[B,H,W] -----0			
0x01C	IRQ02MON[B,H,W] -----0			
0x020	IRQ03MON[B,H,W] -----00000000			
0x024	IRQ04MON[B,H,W] -----00000000			
0x028	IRQ05MON[B,H,W] -----00000000			
0x02C	IRQ06MON[B,H,W] -----000000 00000000			
0x030	IRQ07MON[B,H,W] -----0			
0x034	IRQ08MON[B,H,W] -----00			
0x038	IRQ09MON[B,H,W] -----0			
0x03C	IRQ10MON[B,H,W] -----00			
0x040	IRQ11MON[B,H,W] -----0			
0x044	IRQ12MON[B,H,W] -----00			
0x048	IRQ13MON[B,H,W] -----0			
0x04C	IRQ14MON[B,H,W] -----00			
0x050	IRQ15MON[B,H,W] -----0			
0x054	IRQ16MON[B,H,W] -----00			
0x058	IRQ17MON[B,H,W] -----0			
0x05C	IRQ18MON[B,H,W] -----00			
0x060	IRQ19MON[B,H,W] -----0			
0x064	IRQ20MON[B,H,W] -----00			
0x068	IRQ21MON[B,H,W] -----0			
0x06C	IRQ22MON[B,H,W] -----00			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x070		IRQ23MON[B,H,W] -----000000		
0x074		IRQ24MON[B,H,W] -----00000		
0x078		IRQ25MON[B,H,W] -----0000		
0x07C		IRQ26MON[B,H,W] -----0000		
0x080		IRQ27MON[B,H,W] -----0000		
0x084		IRQ28MON[B,H,W] -----0000 00000000		
0x088		IRQ29MON[B,H,W] -----00000000		
0x08C		IRQ30MON[B,H,W] -----0000 00000000		
0x090		IRQ31MON[B,H,W] -----00000000 00000000		
0x094		IRQ32MON[B,H,W] -----0		
0x098		IRQ33MON[B,H,W] -----0		
0x09C		IRQ34MON[B,H,W] -----00000		
0x0A0		IRQ35MON[B,H,W] -----00000		
0x0A4		IRQ36MON[B,H,W] -----		
0x0A8		IRQ37MON[B,H,W] -----		
0x0AC		IRQ38MON[B,H,W] -----0		
0x0B0		IRQ39MON[B,H,W] -----0		
0x0B4		IRQ40MON[B,H,W] -----0		
0x0B8		IRQ41MON[B,H,W] -----0		
0x0BC		IRQ42MON[B,H,W] -----0		
0x0C0		IRQ43MON[B,H,W] -----0		
0x0C4		IRQ44MON[B,H,W] -----0		
0x0C8		IRQ45MON[B,H,W] -----0		
0x0CC		IRQ46MON[B,H,W] -----		
0x0D0		IRQ47MON[B,H,W] -----		
0x0D4 - 0xFFFF	-	-	-	-

GPIO Base_Address : 0x4003_3000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000		PFR0[B,H,W] ----- 0000 0000 0001 1111		
0x004		PFR1[B,H,W] ----- 0000 0000 0000 0000		
0x008		PFR2[B,H,W] ----- 0000 0000 0000 0000		
0x00C		PFR3[B,H,W] ----- 0000 0000 0000 0000		
0x010		PFR4[B,H,W] ----- 0000 0000 0000 0000		
0x014		PFR5[B,H,W] ----- 0000 0000 0000 0000		
0x018		PFR6[B,H,W] ----- 0000 0000 0000 0000		
0x01C		PFR7[B,H,W] ----- 0000 0000 0000 0000		
0x020		PFR8[B,H,W] ----- 0000 0000 0000 0000		
0x024 - 0x0FF	-	-	-	-
0x100		PCR0[B,H,W] ----- 0000 0000 0001 1111		
0x104		PCR1[B,H,W] ----- 0000 0000 0000 0000		
0x108		PCR2[B,H,W] ----- 0000 0000 0000 0000		
0x10C		PCR3[B,H,W] ----- 0000 0000 0000 0000		
0x110		PCR4[B,H,W] ----- 0000 0000 0000 0000		
0x114		PCR5[B,H,W] ----- 0000 0000 0000 0000		
0x118		PCR6[B,H,W] ----- 0000 0000 0000 0000		
0x11C		PCR7[B,H,W] ----- 0000 0000 0000 0000		
0x120 - 0x1FF	-	-	-	-
0x200		DDR0[B,H,W] ----- 0000 0000 0000 0000		
0x204		DDR1[B,H,W] ----- 0000 0000 0000 0000		
0x208		DDR2[B,H,W] ----- 0000 0000 0000 0000		
0x20C		DDR3[B,H,W] ----- 0000 0000 0000 0000		
0x210		DDR4[B,H,W] ----- 0000 0000 0000 0000		
0x214		DDR5[B,H,W] ----- 0000 0000 0000 0000		
0x218		DDR6[B,H,W] ----- 0000 0000 0000 0000		
0x21C		DDR7[B,H,W] ----- 0000 0000 0000 0000		

Base_Address + Address	Register			
	+3	+2	+1	+0
0x220	DDR8[B,H,W] -----0000 0000 0000 0000			
0x224 - 0x2FF	-	-	-	-
0x300	PDIR0[B,H,W] -----0000 0000 0000 0000			
0x304	PDIR1[B,H,W] -----0000 0000 0000 0000			
0x308	PDIR2[B,H,W] -----0000 0000 0000 0000			
0x30C	PDIR3[B,H,W] -----0000 0000 0000 0000			
0x310	PDIR4[B,H,W] -----0000 0000 0000 0000			
0x314	PDIR5[B,H,W] -----0000 0000 0000 0000			
0x318	PDIR6[B,H,W] -----0000 0000 0000 0000			
0x31C	PDIR7[B,H,W] -----0000 0000 0000 0000			
0x320	PDIR8[B,H,W] -----0000 0000 0000 0000			
0x324 - 0x3FF	-	-	-	-
0x400	PDOR0[B,H,W] -----0000 0000 0000 0000			
0x404	PDOR1[B,H,W] -----0000 0000 0000 0000			
0x408	PDOR2[B,H,W] -----0000 0000 0000 0000			
0x40C	PDOR3[B,H,W] -----0000 0000 0000 0000			
0x410	PDOR4[B,H,W] -----0000 0000 0000 0000			
0x414	PDOR5[B,H,W] -----0000 0000 0000 0000			
0x418	PDOR6[B,H,W] -----0000 0000 0000 0000			
0x41C	PDOR7[B,H,W] -----0000 0000 0000 0000			
0x420	PDOR8[B,H,W] -----0000 0000 0000 0000			
0x424 - 0x4FF	-	-	-	-
0x500	ADE[B,H,W] -----1111 1111 1111 1111			
0x504 - 0x57F	-	-	-	-
0x580	SPSR[B,H,W] -----0 ---1			
0x584-0x5FF	-	-	-	-
0x600	EPFR00[B,H,W] ----00 ----11 ----0- ----00			
0x604	EPFR01[B,H,W] 0000 0000 0000 0000 --0 0000 0000 0000			
0x608	EPFR02[B,H,W] 0000 0000 0000 0000 --0 0000 0000 0000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x60C	*			
0x610	EPFR04[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x614	EPFR05[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W] ---- 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W] ---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x628	EPFR10[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x62C - 0x7FF	-	-	-	-
0x800	*			
0x804	*			
0x808 - 0xFFFF	-	-	-	-

LVD

Base_Address : 0x4003_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	LVD_CTL[B,H,W] 010000--
0x004	-	-	-	LVD_STR[B,H,W] 0-----
0x008	-	-	-	LVD_CLR[B,H,W] 1-----
0x00C	LVD_RLR[W] 00000000 00000000 00000000 00000001			
0x010	-	-	-	LVD_STR2 01-----
0x010 - 0xFFFF	-	-	-	-

USB Clock Base_Address : 0x4003_6000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	UCCR[B,H,W] -----00
0x004	-	-	-	UPCR1[B,H,W] -----00
0x008	-	-	-	UPCR2[B,H,W] ----000
0x00C	-	-	-	UPCR3[B,H,W] ---00000
0x010	-	-	-	UPCR4[B,H,W] ---10111
0x014	-	-	-	UP_STR[B,H,W] -----0
0x018	-	-	-	UPINT_ENR[B,H,W] -----0
0x01C	-	-	-	UPINT_CLR[B,H,W] -----0
0x020	-	-	-	UPIN_STR[B,H,W] -----0
0x024 - 0x02F	-	-	-	-
0x030	-	-	-	USBEN[B,H,W] -----0
0x034 - 0xFFFF	-	-	-	-

CAN_Prescaler Base_Address : 0x4003_7000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	CANPRE[B,H,W] ----1011
0x004 - 0xFFFF	-	-	-	-

MFS ch0 Base_Address : 0x4003_8000

MFS ch1 Base_Address : 0x4003_8100

MFS ch2 Base_Address : 0x4003_8200

MFS ch3 Base_Address : 0x4003_8300

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	SCR/ IBCR[B,H,W] 0-0000	SMR[B,H,W] 0000000-0
0x004	-	-	SSR[B,H,W] 0-00011	ESCR/ IBSR[B,H,W] 0000000
0x008	-	-	RDR/TDR[H,W] ----0 00000000	
0x00C	-	-	BGR1[B,H,W] 00000000	BGR0[B,H,W] 00000000
0x010	-	-	ISMK[B,H,W] -----	ISBA[B,H,W] -----
0x014 - 0xFFFF	-	-	-	-

APPENDIXES

1. Register Map

MB9Axxx/MB9Bxxx Series

MFS ch4 Base_Address : 0x4003_8400
 MFS ch5 Base_Address : 0x4003_8500
 MFS ch6 Base_Address : 0x4003_8600
 MFS ch7 Base_Address : 0x4003_8700

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	SCR / IBCR[B,H,W] 0--0000	SMR[B,H,W] 000000-0
0x0004	-	-	SSR[B,H,W] 0-00011	ESCR / IBSR[B,H,W] 0000000
0x0008	-	-	RDR/TDR[H,W] ----0 0000000	
0x000C	-	-	BGR1[B,H,W] 00000000	BGR0[B,H,W] 00000000
0x0010	-	-	ISMK[B,H,W] -----	ISBA[B,H,W] -----
0x0014	-	-	FCR1[B,H,W] ---00100	FCR0[B,H,W] -0000000
0x018	-	-	FBYTE2[B,H,W] 00000000	FBYTE1[B,H,W] 00000000
0x01C - 0xFF	-	-	-	-

CRC Base_Address : 0x4003_9000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	-	CRCCR[B,H,W] -0000000
0x0004			CRCINIT[B,H,W] 11111111 11111111 11111111 11111111	
0x0008			CRCIN[B,H,W] 00000000 00000000 00000000 00000000	
0x000C			CRCR[B,H,W] 11111111 11111111 11111111 11111111	
0x010 - 0xFFFF	-	-	-	-

Watch Counter Base_Address : 0x4003_A000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	WCCR[B,H,W] 00--0000	WCRL[B,H,W] --000000	WCRD[B,H,W] --000000
0x004 - 0x00F	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W] ----000 -----0	
0x014	-	-	-	CLK_EN[B,H,W] -----00
0x018 - 0xFFFF	-	-	-	-

EXT-Bus I/F Base_Address : 0x4003_F000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000			MODE0[W] ----- -0000000	
0x0004			MODE1[W] ----- -0000000	
0x0008			MODE2[W] ----- -0000000	
0x000C			MODE3[W] ----- -0000000	
0x0010			MODE4[W] ----- -0000000	
0x0014			MODE5[W] ----- -0000000	
0x0018			MODE6[W] ----- -0000000	
0x001C			MODE7[W] ----- -0000000	
0x0020		TIMO[W] 00000101 01011111 11110000 00001111		
0x0024		TIM1[W] 00000101 01011111 11110000 00001111		
0x0028		TIM2[W] 00000101 01011111 11110000 00001111		
0x002C		TIM3[W] 00000101 01011111 11110000 00001111		
0x0030		TIM4[W] 00000101 01011111 11110000 00001111		
0x0034		TIM5[W] 00000101 01011111 11110000 00001111		
0x0038		TIM6[W] 00000101 01011111 11110000 00001111		
0x003C		TIM7[W] 00000101 01011111 11110000 00001111		
0x0040		AREA0[W] ----- -0001111 ----- 00000000		
0x0044		AREA1[W] ----- -0001111 ----- 00010000		
0x0048		AREA2[W] ----- -0001111 ----- 00100000		
0x004C		AREA3[W] ----- -0001111 ----- 00110000		
0x0050		AREA4[W] ----- -0001111 ----- 01000000		
0x0054		AREA5[W] ----- -0001111 ----- 01010000		
0x0058		AREA6[W] ----- -0001111 ----- 01100000		
0x005C		AREA7[W] ----- -0001111 ----- 01110000		
0x0060 - 0x0FFF	-	-	-	-

USB ch0 Base_Address : 0x4004_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -00000000	HRTIMER(2)[B,H,W] -----00
0x2114	-	-	HEOF(1/0)[B,H,W] --000000 00000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----000 00000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] ----- 10100-00	
0x2124	-	-	EPOC[H,W] -----0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EPOIS[H,W] 10---1-- -----	
0x2148	-	-	EP0OS[H,W] 100-00- -XXXXXXXX	
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2160	-	-	EP0DTH[B,H,W] XXXXXXX	EP0DTL[B,H,W] XXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXX	EP1DTL[B,H,W] XXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXX	EP2DTL[B,H,W] XXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXX	EP3DTL[B,H,W] XXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXX	EP4DTL[B,H,W] XXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXX	EP5DTL[B,H,W] XXXXXXX
0x2178 - 0xFFFF	-	-	-	-

DMAC Base_Address : 0x4006_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000		DMACR[B,H,W] 00-00000-----		
0x0010		DMACA0[B,H,W] 00000000 0---0000 00000000 00000000		
0x0014		DMACB0[B,H,W] --000000 00000000 00000000 -----0		
0x0018		DMACSA0[B,H,W] 00000000 00000000 00000000 00000000		
0x001C		DMACDA0[B,H,W] 00000000 00000000 00000000 00000000		
0x0020		DMACA1[B,H,W] 00000000 0---0000 00000000 00000000		
0x0024		DMACB1[B,H,W] --000000 00000000 00000000 -----0		
0x0028		DMACSA1[B,H,W] 00000000 00000000 00000000 00000000		
0x002C		DMACDA1[B,H,W] 00000000 00000000 00000000 00000000		
0x0030		DMACA2[B,H,W] 00000000 0---0000 00000000 00000000		
0x0034		DMACB2[B,H,W] --000000 00000000 00000000 -----0		
0x0038		DMACSA2[B,H,W] 00000000 00000000 00000000 00000000		
0x003C		DMACDA2[B,H,W] 00000000 00000000 00000000 00000000		
0x0040		DMACA3[B,H,W] 00000000 0---0000 00000000 00000000		
0x0044		DMACB3[B,H,W] --000000 00000000 00000000 -----0		
0x0048		DMACSA3[B,H,W] 00000000 00000000 00000000 00000000		
0x004C		DMACDA3[B,H,W] 00000000 00000000 00000000 00000000		
0x0050		DMACA4[B,H,W] 00000000 0---0000 00000000 00000000		
0x0054		DMACB4[B,H,W] --000000 00000000 00000000 -----0		
0x0058		DMACSA4[B,H,W] 00000000 00000000 00000000 00000000		
0x005C		DMACDA4[B,H,W] 00000000 00000000 00000000 00000000		
0x0060		DMACA5[B,H,W] 00000000 0---0000 00000000 00000000		
0x0064		DMACB5[B,H,W] --000000 00000000 00000000 -----0		
0x0068		DMACSA5[B,H,W] 00000000 00000000 00000000 00000000		
0x006C		DMACDA5[B,H,W] 00000000 00000000 00000000 00000000		

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0070		DMACA6[B,H,W] 00000000 0---0000 00000000 00000000		
0x0074		DMACB6[B,H,W] --000000 00000000 00000000 -----0		
0x0078		DMACSA6[B,H,W] 00000000 00000000 00000000 00000000		
0x007C		DMACDA6[B,H,W] 00000000 00000000 00000000 00000000		
0x0080		DMACA7[B,H,W] 00000000 0---0000 00000000 00000000		
0x0084		DMACB7[B,H,W] --000000 00000000 00000000 -----0		
0x0088		DMACSA7[B,H,W] 00000000 00000000 00000000 00000000		
0x008C		DMACDA7[B,H,W] 00000000 00000000 00000000 00000000		
0x0090 - 0xFFFF	-	-	-	-

CAN ch0 Base_Address : 0x4006_2000
 CAN ch1 Base_Address : 0x4006_3000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	STATR[B,H,W] ----- 00000000		CTRLR[B,H,W] ----- 000-0001	
0x0004	BTR[B,H,W] -0100011 00000001		ERRCNT[B,H,W] 00000000 00000000	
0x0008	TESTR[B,H,W] ----- X00000--		INTR[B,H,W] 00000000 00000000	
0x000C	-	-	BRPER[B,H,W] ----- 0000	
0x0010	IF1CMSK[B,H,W] ----- 00000000		IF1CREQ[B,H,W] 0----- 00000001	
0x0014	IF1MSK2[B,H,W] 11-11111 11111111		IF1MSK1[B,H,W] 11111111 11111111	
0x0018	IF1ARB2[B,H,W] 00000000 00000000		IF1ARB1[B,H,W] 00000000 00000000	
0x001C	-	-	IF1MCTR[B,H,W] 00000000 0--0000	
0x0020	IF1DTA2[B,H,W] 00000000 00000000		IF1DTA1[B,H,W] 00000000 00000000	
0x0024	IF1DTB2[B,H,W] 00000000 00000000		IF1DTB1[B,H,W] 00000000 00000000	
0x0028 - 0x002F	-	-	-	-
0x0030	IF1DTA1[B,H,W] 00000000 00000000		IF1DTA2[B,H,W] 00000000 00000000	
0x0034	IF1DTB1[B,H,W] 00000000 00000000		IF1DTB2[B,H,W] 00000000 00000000	
0x0038 - 0x003F	-	-	-	-
0x0040	IF2CMSK[B,H,W] ----- 00000000		IF2CREQ[B,H,W] 0----- 00000001	
0x0044	IF2MSK2[B,H,W] 11-11111 11111111		IF2MSK1 11111111 11111111	
0x0048	IF2ARB2[B,H,W] 00000000 00000000		IF2ARB1[B,H,W] 00000000 00000000	
0x004C	-	-	IF2MCTR[B,H,W] 00000000 0--0000	
0x0050	IF2DTA2[B,H,W] 00000000 00000000		IF2DTA1[B,H,W] 00000000 00000000	
0x0054	IF2DTB2[B,H,W] 00000000 00000000		IF2DTB1[B,H,W] 00000000 00000000	
0x0058 - 0x005F	-	-	-	-
0x0060	IF2DTA1[B,H,W] 00000000 00000000		IF2DTA2[B,H,W] 00000000 00000000	
0x0064	IF2DTB1[B,H,W] 00000000 00000000		IF2DTB2[B,H,W] 00000000 00000000	
0x0068 - 0x007F	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0080	TREQR2[B,H,W] 00000000 00000000		TREQR1[B,H,W] 00000000 00000000	
0x0084 - 0x008F	-	-	-	-
0x0090	NEWDT2[B,H,W] 00000000 00000000		NEWDT1[B,H,W] 00000000 00000000	
0x0094 - 0x009F	-	-	-	-
0x00A0	INTPND2[B,H,W] 00000000 00000000		INTPND1[B,H,W] 00000000 00000000	
0x00A4 - 0x00AF	-	-	-	-
0x00B0	MSGVAL2[B,H,W] 00000000 00000000		MSGVAL1[B,H,W] 00000000 00000000	
0x00B4 - 0xFFFF	-	-	-	-

MN706-00002-1v0-E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL
FM3 Family
32-BIT MICROCONTROLLER
MB9Axxx / MB9Bxxx Series
PERIPHERAL MANUAL

November 2010 the first edition

Published **FUJITSU SEMICONDUCTOR LIMITED**
Edited Sales Promotion Dept.
