Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



32-bit Microcontroller

S6J3360/S6J3370 Series Hardware Manual Traveo™ Family

Document Number: 002-18302 Rev. *F

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2017-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Preface

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

Purpose of this manual and intended readers

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

Note:

- *This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the family.*
  Users should refer to the respective data sheets of devices for device-specific details.

Table of Contents

CHAPTER 1: Overview 18
1. Overview 19
2. Document Definition 20
3. Register Attribute 21
   3.1. Read and Write 21
   3.2. Protection 21
   3.3. Register information 22
4. Vocabulary 23

CHAPTER 2: Function List 25
1. Function List 26
2. Optional Function 29
   2.1. Basic Option 29
   2.2. ID 31

CHAPTER 3: Product Description 32
1. Overview 33
2. Product description 34
3. Note 39
   3.1. Status Flag Clear 39
   3.2. Error Response 39
   3.3. Register Initial Value 39
   3.4. Restriction 41

CHAPTER 4: Block Diagram 43
1. Block Diagram 44
2. Note 45

CHAPTER 5: Clock Configuration 46
1. Overview 47
2. Operation 48
   2.1. Spread Spectrum Clock Generator (SSCG) 50
3. Remark 51

CHAPTER 6: Operation Mode 53
1. Overview 54
2. Configuration 55
3. Registers 56

CHAPTER 7: Memory and Base Address Map 57
1. Overview 58
2. Memory Map 59
3. Base Address Map 61

CHAPTER 8: IRQ Map / NMI Map 69
1. IRQ Map 70

2. NMI Map 76
CHAPTER 9: DMA Channel Activation Factors 77
1. Factors List 78
CHAPTER 10: Port Description 84
1. Port Description List 85
2. Remark 109
CHAPTER 11: Port Configuration 110
1. Overview 111
2. Configuration and Block Diagram 112
3. Operation 113
3.1. Resource Input Configuration 113
3.2. Port Output Function Configuration 199
3.3. The Analog I/O Setting 208
3.4. Input Level Setting 209
3.5. Output Drive Capacity Setting 213
3.6. Port Status 217
3.7. Function Port Group 218
3.8. Key Code Register 223
4. Registers 224
5. Configuration Procedure 225
5.1. Resource I/O Port (Both Direction) 226
5.2. Resource Input 227
5.3. Resource Output 229
5.4. Port Function Input 230
5.5. Port Function Output 231
5.6. Analog Function Input or Output 232
6. Note 233
6.1. Base Timer Port 233
6.2. I2C Port Configuration 233
CHAPTER 12: State Transition 234
1. Overview 235
2. Diagram of State Transition 236
3. Fetching the Operation Mode 238
4. Changes to PSS and RUN 242
CHAPTER 13: Pin Status in Each CPU State 245
1. Pin Status in Each CPU State 246
2. Note 249
CHAPTER 14: Low Voltage Detection 250
1. Overview 251
2. Configuration of Supervised Power Domain 252
3. Operation 253
3.1. LVD Operation 253
3.2. Configurations 254
4. Registers 255
CHAPTER 15: Serial Programming 258
1. Overview 259
2. Memory Map 260
3. Flash Sector Configuration 261

4. Port Configuration 262
5. Operation 263
5.1. Timing Chart 263
5.2. RAM Executing Communication Protocol 264
5.3. Operation Flow 267
6. Note 268
CHAPTER 16: Base Timer 269
1. Overview of the Base Timer 270
2. Block Diagrams of the Base Timer 273
3. Operations of the Base Timer 277
4. 32-Bit Mode Operation 279
5. Debug Mode 281
6. Interrupts from the Base Timer 282
7. Start of DMA Controller (DMAC) 283
8. Registers of the Base Timer 284
9. Notes on Using the Base Timer 290
10. Base Timer Description by Function Mode 292
10.1. PWM Timer Function 293
10.2. PPG Timer Function 320
10.3. Reload Timer Function 344
10.4. PWC Timer Function 364
CHAPTER 17: Base Timer I/O Selection Function 385
1. Overview 386
2. Configuration 387
3. Explanation of Operation 388
4. Registers 392
4.1. I/O Selection Registers (BT_BTSELmn) 397
4.2. Simultaneous Soft Start Register (BT_BTSSSR) 398
4.3. TOUT Read Register (BT_BTTRR) 400
CHAPTER 18: Base Timer Simultaneous Operation 401
1. Overview 402
2. Configuration 403
3. Explanation of Simultaneous Soft Start Operation 404
3.1. Global Timer and Base Timers Operation Waveform 405
3.2. Global Timer Match Starting Channel Buffer Operation by MODE Set 409
4. Overview of the 16-bit Global Timer 413
4.1. Configuration Diagram of the 16-bit Global Timer 414
5. Explanation of the 16-bit Global Timer Operation 415
5.1. Interrupts of the 16-bit Global Timer 418
5.2. Setting Procedure Example of the 16-bit Global Timer 419
5.3. Debug Mode 420
6. Registers of the 16-bit Global Timer 421
6.1. Global Timer Compare Clear Register (GT_CPCLR) 424
6.2. Global Timer Data Register (GT_TCDT) 425
6.3. Global Timer State Control Register (GT_TCCS) 426
6.4. Global Timer State Control Clear register (GT_TCCSC) 430
6.5. Global Timer State Control Set register (GT_TCCSS) 432
6.6. Global Timer Debug Register (GT_DEBUG) 434

6.7. Global Timer Match Starting Control Register (GT_MSCTR) 435
6.8. Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) 438
6.9. Global Timer Match Starting Channel Register (GT_MSCH0~1) 440
7. Precautions for Using This Device 442
CHAPTER 19: 32-Bit Free-Run Timer 443
1. Overview of the 32-Bit Free-Run Timer 444
2. Block Diagram of the 32-Bit Free-Run Timer 445
3. Operation of the 32-Bit Free-Run Timer 446
3.1. Interrupts of the 32-Bit Free-Run Timer 453
3.2. Setting Procedure Example of the 32-Bit Free-Run Timer 454
3.3. Debug Mode 456
4. Registers of the 32-Bit Free-Run Timer 457
4.1. Compare Clear Buffer Register (CPCLRB)/Compare Clear Register (CPCLR) 458
4.2. Timer Data Register (TCDT) 460
4.3. Timer State Control Register (TCCS) 462
4.4. Timer Extended State Control Register (TECCS) 468
4.5. Timer State Control Clear Register (TCCSC) 471
4.6. Timer State Control Set Register (TCCSS) 475
4.7. Free-Run Timer Debug Register(FRT_DEBUG) 478
5. Precautions for Using This Device 479
CHAPTER 20: 32-Bit Input Capture 480
1. Overview of the 32-Bit Input Capture 481
2. Explanation of 32-Bit Input Capture Operation 483
2.1. Interrupt of the 32-Bit Input Capture 484
2.2. Setting Procedure Example 485
3. Registers of the 32-Bit Input Capture 486
3.1. Input Capture Data Registers 0, 1 (IPCP0, IPCP1) 488
3.2. Input Capture State Control Register (ICS) 490
3.3. Input Capture State Control Clear Register (ICSC) 494
3.4. Input Capture State Control Set Register (ICSS) 496
4. Precautions for Using This Device 498
CHAPTER 21: 32-bit Reload Timer 499
1. Overview 500
2. Configuration and Block Diagram 501
3. Operation of the 32-bit Reload Timer 502
3.1. Internal Clock and External Event Counter Operations of 32-bit Reload Timer 503
3.2. Underflow Operation of 32-bit Reload Timer 505
3.3. Output Functions of 32-bit Reload Timer 507
3.4. Counter Operation State 508
3.5. DMA Operation 509
4. Registers 510
4.1. DMA Configuration Register (RLTn_DMACFG) 511
4.2. Timer Control Status Register (RLTn_TMCSR) 512
4.3. 32-bit Reload Register (RLTn_TMRLR) 516
4.4. 32-bit Timer Register (RLTn_TMR) 517
CHAPTER 22: 12/10/8-Bit Analog to Digital Converter 518

1. Overview 519
2. Configuration and Block Diagram 521
3. Operation of A/D Converter 523
3.1. A/D Conversion Flow 523
3.2. Logical Channel Mapping to Analog Input Signals 525
3.3. Logical Channel Data Protection Function 526
3.4. Logical Channel Triggering and Priority 527
3.5. Group Processing 530
3.6. Multiple Conversion Logical Channels 540
3.7. Forced Stop 542
3.8. A/D Converter Calibration 542
3.9. DMA Transfer Function 544
3.10. Range Comparator Function 544
3.11. Pulse Detection Function 548
3.12. Debug Mode 551
4. Setup Procedure Examples 552
4.1. Control of A/D Conversion 552
4.2. Setting of Global A/D Conversion 553
4.3. Setting of Logical Channel 554
4.4. Setting of Range Comparator 555
4.5. Setting of Pulse Detection 556
4.6. A/D Conversion 557
4.7. Range Comparator 559
4.8. Pulse Detection 560
5. Registers 561
5.1. A/D Channel Control Registers (ADC12Bn_CHCTRL0~63) 562
5.2. A/D Channel Status Registers (ADC12Bn_CHSTAT0~63) 566
5.3. A/D Conversion Data Registers (ADC12Bn_CD0~63) 568
5.4. Pulse Counter Control Registers (ADC12Bn_PCCTRL0~63) 570
5.5. A/D Conversion Done Interrupt Flag Registers (ADC12Bn_CDONEIRQ0~1) 572
5.6. A/D Conversion Done Interrupt Enable Registers (ADC12Bn_CDONEIRQE0~1) 575
5.7. A/D Conversion Done Interrupt Clear Registers (ADC12Bn_CDONEIRQC0~1) 578
5.8. Group Interrupted Interrupt Flag Registers (ADC12Bn_GRPIRQ0~1) 581
5.9. Group Interrupted Interrupt Enable Registers (ADC12Bn_GRPIRQE0~1) 584
5.10. Group Interrupted Interrupt Clear Registers (ADC12Bn_GRPIRQC0~1) 586
5.11. Range Comparator Interrupt Flag Registers (ADC12Bn_RCIRQ0~1) 588
5.12. Range Comparator Interrupt Enable Registers (ADC12Bn_RCIRQE0~1) 592
5.13. Range Comparator Interrupt Clear Registers (ADC12Bn_RCIRQC0~1) 594
5.14. Pulse Counter Interrupt Flag Registers (ADC12Bn_PCIRQ0~1) 596
5.15. Pulse Counter Interrupt Enable Registers (ADC12Bn_PCIRQE0~1) 599
5.16. Pulse Counter Interrupt Clear Registers (ADC12Bn_PCIRQC0~1) 601
5.17. A/D Channel Trigger Status Flag Registers (ADC12Bn_TRGST0~1) 604
5.18. A/D Channel Trigger Clear Registers (ADC12Bn_TRGCL0~1) 608
5.19. A/D Channel Trigger Overrun Flag Registers (ADC12Bn_TRGOR0~1) 612
5.20. A/D Channel Trigger Overrun Clear Registers (ADC12Bn_TRGORC0~1) 615
5.21. Range Comparator Over Threshold Flag Registers (ADC12Bn_RCOTF0~1) 617
5.22. Conversion Done DMA Select Registers (ADC12Bn_CDDS0~3) 620
5.23. A/D Converter Comparison Time Setting Register (ADC12Bn_CT) 622

5.24. A/D Converter Resumption Time Setting Register (ADC12Bn_RT) 623
5.25. A/D Converter Sampling Time Setting Registers (ADC12Bn_ST0~3) 624
5.26. A/D Converter Offset Compensation Setting Register (ADC12Bn_OCV) 625
5.27. A/D Converter Gain Compensation Setting Register (ADC12Bn_GCV) 626
5.28. A/D Converter Global Control Register (ADC12Bn_CTRL) 627
5.29. A/D Converter Global Status Register (ADC12Bn_STAT) 630
5.30. Range Comparator Upper Threshold Registers (ADC12Bn_RCOH0~7) 632
5.31. Range Comparator Lower Threshold Registers (ADC12Bn_RCOL0~7) 633
5.32. Full Range Comparator Upper Threshold Registers (ADC12Bn_FRCOH0~7) 634
5.33. Full Range Comparator Lower Threshold Registers (ADC12Bn_FRCOL0~7) 636
5.34. A/D Multiple Conversion Channel Control Registers (ADC12Bn_MCCTRL0~3) 638
5.35. A/D Multiple Conversion Channel Status Registers (ADC12Bn_MCSTAT0~3) 640
CHAPTER 23: Partial Wakeup Control 641
1. Overview 642
2. Configuration 643
3. Explanation of Operation 647
3.1. State Transition Diagram 647
3.2. Flowchart 648
3.3. Timing Charts 664
4. Registers 667
4.1. PWU Control Register (PWU_PWUC) 668
4.2. A/D Conversion Request Trigger Control Register (PWU_ADTC) 669
5. Precautions for Using this Device 671
CHAPTER 24: Programmable CRC 673
1. Overview 674
2. Configuration and Block Diagram 675
3. Operation of the Programmable CRC 676
3.1. CRC Operation Flowcharts 677
3.2. CRC Input Data and Checksum Calculation Flow 680
3.3. CRC Calculation Example 684
4. Registers 685
4.1. CRC Polynomial Register (CRCn_POLY) 686
4.2. CRC Seed Register (CRCn_SEED) 687
4.3. CRC Final XOR Register (CRCn_FXOR) 688
4.4. CRC Configuration Register (CRCn_CFG) 689
4.5. CRC Write Register (CRCn_WR) 693
4.6. CRC Read Register (CRCn_RD) 694
CHAPTER 25: Clock Monitor 695
1. Overview 696
2. Configuration and Block Diagram 697
3. Operation of Clock Monitor 698
4. Registers 699
4.1. Clock Output Function Control Register (CKOTCNTR) 700
5. Precautions 702
CHAPTER 26: DDR High Speed SPI Controller 703
1. Overview 704
2. Configuration 706
3. Serial Interface 708

3.1. Serial Clock Modes and Data Rate Modes ... 708
3.2. SPI Clock Frequency ... 713
3.3. SPI Input Data Sampling Point ... 713
3.4. SPI Data Learning Pattern ... 715
3.5. SPI Data Protocol ... 715
3.6. Shift Direction ... 716
4. Operations of the DDRHSSPI ... 725
4.1. Direct Mode ... 725
4.2. Command Sequencer Mode ... 740
4.3. Address Map of DDRHSSPI ... 750
4.4. General Use Case Guidelines for DDRHSSPI ... 751
5. Registers ... 760
5.1. DDRHSSPI Module Control Register (DDRHSSPIn_MCTRL) ... 762
5.2. DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3) ... 764
5.3. DDRHSSPI TX Interrupt Flag Register (DDRHSSPIn_TXF) ... 767
5.4. DDRHSSPI TX Interrupt Enable Register (DDRHSSPIn_TXE) ... 770
5.5. DDRHSSPI TX Interrupt Clear Register (DDRHSSPIn_TXC) ... 773
5.6. DDRHSSPI RX Interrupt Flag Register (DDRHSSPIn_RXF) ... 775
5.7. DDRHSSPI RX Interrupt Enable Register (DDRHSSPIn_RXE) ... 779
5.8. DDRHSSPI RX Interrupt Clear Register (DDRHSSPIn_RXC) ... 782
5.9. DDRHSSPI Fault Status Flag Register (DDRHSSPIn_FAULTF) ... 785
5.10. DDRHSSPI Fault Status Clear Register (DDRHSSPIn_FAULTC) ... 787
5.11. DDRHSSPI Direct Mode Configuration Register (DDRHSSPIn_DMCFG) ... 789
5.12. DDRHSSPI DMA Enable Register (DDRHSSPIn_DMAEN) ... 790
5.13. DDRHSSPI Direct Mode Start Register (DDRHSSPIn_DMSTART) ... 791
5.14. DDRHSSPI Direct Mode Peripheral Select Register (DDRHSSPIn_DMPSEL) ... 792
5.15. DDRHSSPI Direct Mode Transfer Protocol Register (DDRHSSPIn_DMTRP) ... 793
5.16. DDRHSSPI Byte Count Control Register (DDRHSSPIn_DMBCC) ... 794
5.17. DDRHSSPI Byte Count Status Register (DDRHSSPIn_DMBCS) ... 795
5.18. DDRHSSPI Direct Mode FIFO Status Register (DDRHSSPIn_DMFIFOSTATUS) ... 796
5.19. DDRHSSPI Direct Mode FIFO Configuration Register (DDRHSSPIn_DMFIFOCFG) ... 798
5.20. DDRHSSPI TX-FIFO Registers (DDRHSSPIn_TXFIFO0-23) ... 800
5.21. DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23) ... 802
5.22. DDRHSSPI Read Command Sequencer Data/Control Registers (DDRHSSPIn_RDCSDC0-11) ... 804
5.23. DDRHSSPI Module ID Register (DDRHSSPIn_MID) ... 806
5.24. DDRHSSPI Command Sequencer Prefetch Address Register (DDRHSSPIn_CSPREFETCHADDR) ... 808
5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7) ... 810
5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7) ... 811
5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7) ... 812
5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP) ... 813
5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS) ... 814
5.30. DDRHSSPI Command Sequencer Configuration Register (DDRHSSPIn_CSCFG) ... 820
5.31. DDRHSSPI Command Sequencer Idle Time Register (DDRHSSPIn_CSITIME) ... 823
5.32. DDRHSSPI Command Sequencer Address Extension Register (DDRHSSPIn_CSAEXT) ... 824
5.33. DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register (DDRHSSPIn_CSPBUFFERCFG) ... 825
5.34. DDRHSSPI Command Sequencer Prefetch Buffer Status Register (DDRHSSPIn_CSPBUFFERSTATUS) ... 826
CHAPTER 27: Stepper Motor Controller ... 827
1. Overview ... 828

2. Configuration and Block Diagram 829
3. Operation of the Stepper Motor Controller 831
3.1. Operation of PWM Pulse Generator 832
3.2. Selection of Motor Drive Signals 836
3.3. Synchronization System of Stepper Motor Controller 837
3.4. Operation of the SMC Trigger Delay Function 838
4. Registers 841
4.1. PWM Control Register (PWC) 842
4.2. PWM1 and PWM2 Compare Registers (PWC1, PWC2) 845
4.3. PWM Selection Register (PWS) 849
4.4. PWM Selection Set Register (PWSS) 852
4.5. SMC Trigger Delay Register (PTRGDL) 854
CHAPTER 28: Trigger Configuration of Stepper Motor Controller 855
1. Overview 856
2. Configuration and Block Diagram 857
3. Operation 858
4. Registers 859
4.1. SMC Trigger Selection Register (SMCTGg_PTRGS) 860
4.2. SMC Trigger Register (SMCTGg_PTRG) 863
CHAPTER 29: Sound Generator 865
1. Overview 866
2. Configuration 868
3. Operations 869
3.1. Relation between the Amplitude Data Register (SGAR) and PWM Pulse 871
3.2. Relation between the Frequency Data Register (SGFR), the Extended Frequency Data Register (SGEFR) and Tone Pulse Signal 872
3.3. Relation between the PWM Cycle Data Register (SGPCR) and PWM Cycle 876
3.4. Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings 877
3.5. Sound Generator Operation 883
3.6. Interrupt of Sound Generator 915
4. Registers 916
4.1. DMA Transfer Update Enable Register (SGDER) 917
4.2. Sound Control Register (SGCR) 920
4.3. Amplitude Data Register (SGAR) 924
4.4. Frequency Data Register (SGFR) 925
4.5. Tone Output Number Register (SGNR) 926
4.6. Time Cycle Register (SGTCR) 927
4.7. Increase and Decrease Data Register (SGIDR) 928
4.8. PWM Cycle Data Register (SGPCR) 929
4.9. Extended Frequency Data Register (SGEFR) 930
4.10. DMA Transfer Intermediate Register (SGDMAR) 931
4.11. Interrupt Clear Register (SGCCR) 933
CHAPTER 30: Sound Waveform Generator 934
1. Overview 935
2. Configuration and Block Diagram 937
3. Operation of the Sound Waveform Generator 938
3.1. Sound Source Specifications 938
3.2. Status Indicator 958

3.3. Interrupt 960
3.4. Sound Source Generation Start 962
3.5. Sound Source Generation Stop 963
3.6. Continuous SWFG Sound Source Generation 967
3.7. Output Data Format 970
3.8. Filter 971
3.9. Phase Synchronizer 977
3.10. Smooth Connection 979
3.11. DMA Interface 982
4. Registers 983
4.1. SWFG Channel Enable Register (WGCHEN) 984
4.2. SWFG Channel Start Register (WGCHSTART) 986
4.3. SWFG Channel Busy Register (WGCHBUSY) 988
4.4. SWFG Channel Stop Register (WGCHSTOP) 990
4.5. SWFG Channel DMA Enable Register (WGCHDMAEN) 992
4.6. SWFG Interrupt Enable Register (WGINTREN) 994
4.7. SWFG Interrupt State Register (WGINTRSTATE) 996
4.8. SWFG Interrupt Clear Register (WGINTRCLR) 998
4.9. SWFG AHB Bus Error Register (WGAHBERR) 1000
4.10. SWFG Channel Address0 Register (WGCHADD0) 1002
4.11. SWFG Channel Address1 Register (WGCHADD1) 1004
4.12. SWFG Channel Address2 Register (WGCHADD2) 1006
4.13. SWFG Channel n Filter Coefficient 0 Register (WGCHnFILCOEF0, n=0 to 4) 1007
4.14. SWFG Channel n Filter Coefficient 1 Register (WGCHnFILCOEF1, n=0 to 4) 1008
4.15. SWFG Channel n Filter Coefficient 2 Register (WGCHnFILCOEF2, n=0 to 4) 1009
4.16. SWFG Channel n Filter Coefficient 3 Register (WGCHnFILCOEF3, n=0 to 4) 1010
4.17. SWFG Channel n Filter Coefficient 4 Register (WGCHnFILCOEF4, n=0 to 4) 1011
4.18. SWFG Channel n Filter Coefficient 5 Register (WGCHnFILCOEF5, n=0 to 4) 1012
4.19. SWFG Channel n Filter Coefficient 6 Register (WGCHnFILCOEF6, n=0 to 4) 1013
4.20. SWFG Channel n Filter Coefficient 7 Register (WGCHnFILCOEF7, n=0 to 4) 1014
4.21. SWFG Channel n Control 0 Register (WGCHnCTRL0, n=0 to 4) 1015
4.22. SWFG Channel n Control 1 Register (WGCHnCTRL1, n=0 to 4) 1018
4.23. SWFG Channel n Control 2 Register (WGCHnCTRL2, n=0 to 4) 1021
4.24. SWFG Channel n Control 3 Register (WGCHnCTRL3, n=0 to 4) 1026
4.25. SWFG Channel n Control 4 Register (WGCHnCTRL4, n=0 to 4) 1029
4.26. SWFG Channel n Status Register (WGCHnSTATUS, n=0 to 4) 1032
5. Appendix 1035
5.1. Example of sound source generation 1035
CHAPTER 31: Sound Mixer 1039
1. Overview 1040
2. Configuration and Block Diagram 1041
3. Operation of the Sound Mixer 1042
3.1. Basic Mixing Operation Procedure 1042
3.2. Volume Effects 1044
3.3. Sound Source Input and Internal Mixing Process Mode 1045
3.4. Interrupt 1046
3.5. Data Request Control 1047
3.6. Data Output Control 1048

3.7. AHB Slave Interface 1049
3.8. AHB Master Interface 1050
3.9. Input Buffer (FIFO) 1051
3.10. Sound Source Sampling Rate 1052
4. Registers 1053
4.1. Mixer Channel Register (MXCH) 1054
4.2. Mixer Output Control Register (MXOCTRL) 1056
4.3. Mixer Date Request Control Register (MXDRQCTRL) 1058
4.4. Mixer Input Control Register (MXICTRL) 1060
4.5. Mixer Channel Monaural Register (MXCHMONO) 1062
4.6. Mixer Channel Volume1 Register (MXCHVOL1) 1064
4.7. Mixer Channel Volume2 Register (MXCHVOL2) 1065
4.8. Mixer Channel Volume3 Register (MXCHVOL3) 1066
4.9. Mixer Channel Mute Register (MXCHMUTE) 1067
4.10. Mixer Channel Fade_In/Out1 Register (MXCHFADE1) 1069
4.11. Mixer Channel Fade_In/Out2 Register (MXCHFADE2) 1071
4.12. Mixer Channel Fade_In/Out3 Register (MXCHFADE3) 1073
4.13. Mixer Channel Fade_In/Out4 Register (MXCHFADE4) 1075
4.14. Mixer Channel Fade_in/out5 Register (MXCHFADE5) 1077
4.15. Mixer Mixed Fade_In/Out Register (MXMXDFADE) 1079
4.16. Mixer Channel Fade_In/Out Enable Register (MXCHFADEEN) 1081
4.17. Mixer Insert the PCM Data Control Register (MXPMISIPDC) 1083
4.18. Mixer Buffer Clear Register (MXBUFFCLR) 1085
4.19. Mixer FADE_In/Out Clear Register (MXFADECLR) 1087
4.20. Mixer Interrupt Enable Register (MXINTREN) 1089
4.21. Mixer Interrupt Status Register (MXINTRSTATE) 1092
4.22. Mixer Interrupt Clear Register (MXINTRCLR) 1095
4.23. Mixer Input Buffer Count1 Register (MXINBUFFCNT1) 1098
4.24. Mixer Input Buffer Count2 Register (MXINBUFFCNT2) 1101
4.25. Mixer Channel Buffer Count Register (MXCHBUFFCNT) 1103
4.26. Mixer Output Buffer Count Register (MXOUTBUFFCNT) 1105
4.27. Mixer AHB Bus Error Register (MXAHBERR) 1107
4.28. Mixer WFGn Data Address Register (MXWFGnDADR, n=0 to 4) 1110
4.29. Mixer PMIS n Data Address Register0-15 (MXPMISnDADR0-15, n=0 to 4) 1111
5. Appendix 1112
5.1. Gain for Volume Control 1112
5.2. Fade In/Fade Out Time 1119
5.3. Digital Filter 1120
CHAPTER 32: Inter-IC Sound (I2S) 1123
1. Overview 1124
2. Configuration and Block Diagram 1125
3. Operations of the I2S 1126
3.1. I2S Frame Construction 1126
3.2. I2S Configuration and Operation Modes 1128
3.3. Bit Alignment 1135
3.4. FIFO Construction 1137
3.5. Caution Summary 1140
4. Registers 1141

4.1. Reception FIFO Data Register (I2Sn_RXFDAT0 to 15) 1144
4.2. Transmission FIFO Data Register (I2Sn_TXFDAT0 to 15) 1146
4.3. Control Register (I2Sn_CNTREG) 1147
4.4. Channel Control Register 0 (I2Sn_MCR0REG) 1152
4.5. Channel Control Register 1 (I2Sn_MCR1REG) 1156
4.6. Channel Control Register 2 (I2Sn_MCR2REG) 1157
4.7. Operation Control Register (I2Sn_OPRREG) 1158
4.8. Software Reset Register (I2Sn_SRST) 1160
4.9. Interrupt Control Register (I2Sn_INTCNT) 1161
4.10. Status Register (I2Sn_STATUS) 1166
4.11. DMA Activate Register (I2Sn_DMAACT) 1170
4.12. Debug Register (I2Sn_DEBUG) 1172
4.13. Module ID Register (I2Sn_MIDREG) 1174
CHAPTER 33: PCMPWM 1175
1. Overview 1176
2. Configuration and Block Diagram 1177
2.1. Block Diagram 1177
2.2. Configuration of the PCMPWM Module 1178
2.3. Output Configuration 1183
3. Operation of the PCMPWM 1184
3.1. Description of the PCM to PWM Conversion Process 1184
3.2. PWM Cycle Time Configuration 1189
3.3. PCM Data Sample Input 1190
3.4. Interrupts 1191
3.5. Dead Timer Operation 1192
4. Registers 1193
4.1. PCMPWM Control Register (PCMPWMi_CONTROL) 1194
4.2. PCMPWM Output Control Register (PCMPWMi_OCTRL) 1197
4.3. PCMPWM Clock Select Register (PCMPWMi_CLKSEL) 1199
4.4. PCMPWM Count Period Register (PCMPWMi_COUNTP) 1200
4.5. PCM Offset Register (PCMPWMi_PCMOFFS) 1201
4.6. PCM Interrupt Enable Register (PCMPWMi_INTREN) 1202
4.7. PCM Interrupt Status Register (PCMPWMi_INTRSTAT) 1204
4.8. PCM Interrupt Clear Register (PCMPWMi_INTRCLR) 1206
4.9. PCM Data Register 0..15 (PCMPWMi_DATA) 1208
CHAPTER 34: LCD Bus Interface (LCDBusIF) 1209
1. Preface 1209
1.1 Purpose and Scope 1209
1.2 Document Structure 1209
2. Function 1209
2.1 Feature Summary 1209
2.2 Block Diagram 1210
2.3 Functional Limitations 1210
2.4 Nomenclature 1211
2.5 Use Cases 1212
2.6 Protocol Engine 1213
2.7 Iris Graphics Core Interface 1217
2.8 Color Lookup Table 1219

2.9 Command Sequencer 1220
2.10 Data Master Interface 1226
2.11 Interrupt Controller 1227
3. Application 1227
3.1 Map Tables 1227
3.2 Getting Started 1228
3.3 Display Setup 1230
3.4 Color Table Setup 1232
3.5 Control Flow 1233
3.6 Fifo Operation 1237
3.7 Error Recovery 1238
3.8 Interrupt Setup 1238
3.9 SW Interface 1238
CHAPTER 35: LCD Controller 1257
1. Overview 1258
2. Features 1259
3. Configuration 1260
4. Operation 1261
4.1. Operation of LCD Controller/Driver (LCDC) 1262
4.2. 1/2 Duty Output Waveform 1264
4.3. 1/3 Duty Output Waveform 1267
4.4. 1/4 Duty Output Waveform 1270
4.5. Static Drive Output Waveform 1273
5. Setting 1276
6. Registers 1278
6.1. LCD Control Register 0: LCR0 1279
6.2. Data Memory for Display: VRAM 1281
6.3. LCDC Control Register 1: LCR1 1283
6.4. Common Pin Switching Register: LCDCMR 1284
6.5. LCDC Static Control Register: LCRS 1285
6.6. Static LCD Display Data Register: LDR 1287
6.7. Key Code Register: LCD_KEYCDR 1289
6.8. Segment Output Register: SEGER 1292
6.9. Common Output V Pin Control Register: COMVER 1293
7. Q&A 1295
7.1. How can I Set Pins to COM Output Pins or SEG Output Pins? 1296
7.2. How to Set VRAM? 1298
7.3. How can I Setting the Frame Cycle? 1299
7.4. How can I Set the Bias? 1300
7.5. How can I Set the Duty? 1301
7.6. How can I Control the LCD Operation Start/Stop? 1302
7.7. How can I Execute/Cancel the Display? 1303
7.8. How can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)? 1304
7.9. How can I Select either Internal or External for the Division Resistor? 1305
7.10. How can I Select Pin of V3 Voltage? 1306
7.11. How can I Select either Internal or External for the Division Resistor? 1307
7.12. How can I Adjust the Brightness When the Internal Division Resistor is Used? 1309
7.13. How can I Block the Current with the External Division Resistor When the LCD Stops? 1310

7.14. How can I Display/Non-display the LCD with Static Drive (ST0 to ST8)? 1311
8. Sample Program 1312
9. Notes 1319
CHAPTER 36: Indicator PWM 1320
1. Overview 1321
2. Configuration and Block Diagram 1322
3. Operation of Indicator PWM 1323
3.1. Indicator PWM Timer 1323
3.2. PWM Timer Count Operation Start 1324
3.3. Timer Count Operation Stop 1325
3.4. Counter Value Updating while PWM Counter Operation is in Progress 1326
3.5. Indicator PWM Interrupts 1327
3.6. Operation Flowchart 1329
4. Registers 1332
4.1. Timer Control Register (ITMCR) 1333
4.2. Count Control Register (ICNTCR) 1336
4.3. Status Control Register (ISTC) 1338
4.4. Status Control Clear Register (ISTCC) 1340
4.5. Status Control Set Register (ISTCS) 1342
4.6. PWM Cycle Setting Register (IPCSR) 1344
4.7. PWM Duty Setting Register (IPDUT) 1346
5. Precautions for Using This Device 1348
5.1. Notes to Observe When Accessing a Register 1348
5.2. Indicator PWM Operation Precautions 1348
CHAPTER 37: Memory Protection Unit for AXI 1349
1. Overview 1350
2. Configuration and Block Diagram 1351
3. Operation of the MPU AXI 1352
4. Registers 1358
4.1. MPU AXI Control Register (MPUXn_CTRL0) 1361
4.2. MPU AXI NMI Enable Register (MPUXn_NMIEN) 1363
4.3. MPU AXI Write Error Control Register (MPUXn_WERRC) 1364
4.4. MPU AXI Write Error Address Register (MPUXn_WERRA) 1365
4.5. MPU AXI Read Error Control Register (MPUXn_RERRC) 1366
4.6. MPU AXI Read Error Address Register (MPUXn_RERRA) 1367
4.7. MPU AXI Region Control Registers (MPUXn_CTRL1 to 8) 1368
4.8. MPU AXI Start Address Registers (MPUXn_SADDR1 to 8) 1369
4.9. MPU AXI End Address Registers (MPUXn_EADDR1 to 8) 1370
4.10. MPU AXI Unlock Register (MPUXn_UNLOCK) 1371
4.11. MPU AXI Module ID Register (MPUXn_MID) 1372
5. Notes on Using MPU AXI 1373
CHAPTER 38: Sound System Configuration 1375
1. Overview 1376
1.1. Input Module 1376
1.2. Sound Mixer 1376
1.3. Output Module 1376
2. Configuration and Block Diagram 1377
3. Operation 1378

3.1. Start Operations 1378
3.2. Addition of Sound Source Inputs 1380
3.3. Stop Operations 1381
3.4. Operation of a Sound Source 1382
3.5. Sound system reset Operations 1384
4. Configuration and limitation of each module 1389
4.1. Interface between SMIX and DAC 1389
4.2. Interface between SMIX and PCM-PWM 1389
4.3. Interface between SMIX and I2S 1390
4.4. Interface between SWFG and SMIX 1391
4.5. Interface between DMAC and SMIX 1391
4.6. Interface between CPU and SMIX 1392
5. Note 1393
CHAPTER 39: Expand PLL 1394
1. Overview 1395
2. Configuration and Block Diagram 1396
3. Operation of the Expand PLL 1397
3.1. Expand PLL Clock 1397
3.2. Clock Source Selector 1397
3.3. Clock Divider 1397
3.4. Clock Stabilization Time 1397
3.5. Expand PLL Clock Supervisor 1398
4. Setting Procedure 1399
5. Registers 1402
5.1. Expand PLLx Clock Supervisor Setting Register 0 (EXCSVPLLxCFGR0) (x=0 to 2) 1403
5.2. Expand PLL0xClock Supervisor Setting Register 1 (EXCSVPLLxCFGR1) (X=0 to 2) 1405
5.3. Expand Clock Supervisor Setting Register (EXCSVCFGR) 1407
5.4. Expand Clock Supervisor Error Register (EXCSVERR) 1409
5.5. Expand Clock Supervisor Error Clear Register (EXCSVERRCLR) 1411
5.6. Expand PLLx Control Register (EXPLLxCNTR) (x=0 to 2) 1413
5.7. Expand PLL Enable Register (EXPLLEN) 1416
5.8. Expand Clock Divider Register x (EXCKDIVRx)(x=0 to 2) 1418
5.9. Expand PLL Select Register (EXPLLSEL) 1421
CHAPTER 40: LCD Subsystem 1423
1. Overview 1424
1.1. Scope 1424
1.2. Protections 1424
2. Configuration and Block Diagram 1425
3. Operation 1426
3.1. Application 1426
CHAPTER 41: APPENDIX: Major Changes 1428
Revision History 1456

CHAPTER 1: Overview

This chapter explains the product overview.

1. Overview
2. Document definition
3. Register attribute
4. Vocabulary

CODE: OVERVIEW-S6J3360/3370-E1

1. Overview

S6J3360/S6J3370 is a microcontroller series which is to be applied to automotive systems representative of a graphical cluster control unit on a dashboard.

2. Document Definition

The related documents of S6J3360/S6J3370 are the followings.

Table 2-1

Document Type	Definition	Primary User	Document Code
Datasheet	The function and its characteristics are specified quantitatively.	Investigator and hardware engineer	002-03359
S6J3360/S6J3370 hardware manual	The function and its operation of S6J3360/S6J3370 series are described.	Software engineer	002-18302
Traveo™ Platform hardware manual	The function and its operation of CPU core platform are described.	Software engineer	002-07884
Application note	The reference software, sample application, the reference board design and so on are explained.	Software and hardware engineer	Under consideration

Notes:

- *Refer all documents for the system development.*
- *"Primary user" is a most likely engineer for whom the document is the most useful.*
- *The description of the datasheet and the S6J3360/S6J3370 hardware manual should precede the duplicated description of Traveo™ platform hardware manual.*
- *Traveo™ platform hardware manual is expected to be used as dictionary of platform specification.*
- *As for the resources specifications of this model, refer the Pin Descriptions chapter in the datasheet.*

3. Register Attribute

3.1. Read and Write

Refer the table with the following definition.

Table 3-1 Register Attribute of Read and Write

Attribute	Definition
R	It can be read.
R0	"0" is to be read.
R1	"1" is to be read.
RX	The read value is undefined.
W	It can be written.
W0	"0" must be written.
W1	"1" must be written.
WX	Writing doesn't affect the operation.
"/" (slash)	The read value is same as the written value.
"," (comma)	The read value is different from the written value.

■ **Example**

R/W : The written value is to be read.

R, W0 : "0" must be written, and the read value is different from the written value.

Notes:

- *The register attribute of a status register which has its clear register and set register is conveniently described to be R/W though the written value cannot be read directly. In this case R/W has a meaning that the controlled value is to be read.*
- *A register attribute does not necessarily describe a prohibited operation of the register. Any prohibited operations should be referred within the line of the register description as well. See every description of the register specification together with every note.*

3.2. Protection

Refer the table with the following definition.

Table 3-2 Register Attribute of Protection

Attribute	Definition
RP	It can be read at the supervisor mode.
WP	It can be written at the supervisor mode.
WS	It can be written after cancelling the sequence protection.
WPS	It can be written after cancelling the sequence protection at the supervisor mode.

3.3. Register information

You can also see register information in a table as bellow.

BIT_OFFSET	The bit offset (Ex. 31,30,29,,,,0)
BIT_NAME	The bit name
ACCESS_TYPE	The allowed access defined 3.1
PROT_TYPE	The allowed access defined 3.2
INITIAL_VALUE	The initial value

4. Vocabulary

A/D converter	Analog digital converter	
ADC	Analog digital converter	
AHB	Advanced high performance bus	
AMBATM	Advanced microcontroller bus architecture	
APB	Advanced peripheral bus	
ATCM	TCM-A port	
AXI	Advanced extensible interface	
B0TCM	TCM B0 port	
B1TCM	TCM B1 port	
BBU	Bit banding unit	
BDR	Boot description record	
CAN	Control are network	
CD	Clock domain	
CPU	Central processing unit	
CR	CR Oscillator	
CRC	Cyclic redundancy check	
CSV	Clock supervisor	
DAP	Debug access port	
DED	Dual error detection	
DMA	Direct memory access	
DMAC	DMA controller	
ECC	Error correction code	
ETM	Embedded trace macro	
EXT-IRC	External interrupt controller	
FIQ	Fast interrupt request	
FPU	Floating point unit	
FRT	Free run timer	
GPIO	General purpose I/O	
HPM	High performance matrix	
HW-WDT	Hardware watchdog timer	
I/O	Input or output	
ICU	Input capture unit	
IPCU	Inter-processor communication unit	
IRC	Interrupt controller	
IRQ	Interrupt request	
ISR	Interrupt service routine	
JTAG	Joint test action group	
LLPP	Low latency peripheral port	
LVD	Low voltage detector	
MCU	Microcontroller unit	
MFS	Multi-function serial interface	
NF	Noise filter	
NMI	Non maskable interrupt	
OCU	Output compare unit	

OSC	Oscillator	
PLL	Phase locked loop	
PONR	Power on reset	
PPC	Port pin configuration	
PSS	Power saving state	
PWM	Pulse width modulation	
RAM	Random access memory	
RIC	Resource input configuration	
ROM	Read only memory	
RTC	Real time clock	
RVD	Low voltage detection and reset for RAM retention	
SCT	Source clock timer	
SEC	Single error correction	
SECDED	Single error correction and dual error detection	
SHE	Secure Hardware Extension	
SRAM	Static RAM	
SW-WDT	Software watchdog timer	
SYSC	System controller	
TCFLASH	FLASH connected to TCM	
TCM	Tightly coupled memory	
TCRAM	RAM connected to TCM	
TPU	Timing protection unit	
UDC	Up-down counter	
VIC	Vectored interrupt controller	
WDR	Watchdog description record	
WDT	Watchdog timer	
WorkFLASH	Work FLASH memory	

CHAPTER 2: Function List

This chapter explains the functions.

1. Function list
2. Optional function

1. Function List

The table shows the functions which are implemented in S6J3360/S6J3370 series.

Table 1-1

Function	S6J33xxJxx 176-pin	S6J33xxHxx 144-pin	S6J33xxGxx 120-pin	S6J33xxFxx 100-pin	Remarks
CPU core	Arm Cortex-R5F				
FPU	Available				
PPU	Available				
MPU	Available				
TPU	Available				
Endian	Little endian				
Core clock frequency	132 MHz				See the datasheet (section 7.4.3)
HPM bus frequency	Option				See the datasheet (section 7.4.3)
LLPM bus frequency	Option				See the datasheet (section 7.4.3)
Resource clock frequency	Option				See the datasheet (section 7.4.3)
Embedded CR oscillation	Slow clock:100 kHz Fast clock: 4 MHz (Center frequency)				
PLL	PLL0, Expand PLL0,1,2				
SSCG PLL	SSCG0				
Clock supervisor	Available				
DMA	16 ch				
Boot-ROM	16 KB				
JTAG	Available				
Data cache	16 KB				
Instruction cache	16 KB				
Program Flash	Option				See 2.1
Work Flash	112 KB				See 2.1
TC-RAM	128KB				See 2.1
System-RAM	128KB (include Backup area)				See 2.1
Backup area in System-RAM	16+8KB				See 2.1
Security (SHE)	Available				
Low latency interrupt	Available				

Function	S6J33xxJxx 176-pin	S6J33xxHxx 144-pin	S6J33xxGxx 120-pin	S6J33xxFxx 100-pin	Remarks
Power domain	3 domains				PD1/ PD2/ PD4 (PD4_1,PD4_2)
Power supply	5.0 V ± 0.5 V or 3.3 V ± 0.3 V				
Embedded LDO power supply for 5.0 V	Available				
Low-voltage detection of external power supply	Available				
Low-voltage detection of Internal LDO output	Available				
Hardware watchdog timer	Available				
Software watchdog timer	Available				
Package	LQFP-176(0.5mm)/ TEQPF-176(0.5mm)	LQFP-144(0.4mm)/ LQFP-144(0.5mm)/ TEQFP-144(0.5mm)	LQFP-120(0.5mm)/ TEQFP-120(0.5mm)	LQFP-100(0.5mm)/ TEQFP-100(0.5mm)	See 2.1 (): pin pich
General-purpose I/O	126 ports (124 ports)	94 ports (92 ports)	70 ports (68 ports)	50 ports (48 ports)	(): When used sub clock
QPRC	2 ch			-	
32-bit reload timer	6 ch (Input: 6pin/ Output: 6pin)	6 ch (Input: 6pin/ Output: 6pin)	6 ch (Input: 4 pin/ Output: 4 pin)	6 ch (Input: 2 pin/ Output: 2 pin)	
Real-time clock	Available				Automatic calibration
Sound waveform generator	1 unit × 5 outputs				
Sound mixer	1 unit × 10 inputs				
PCM-PWM	1 unit (L and R)		-		
Base timer	16 units (32 ch)		15 units (30 ch)	12 unit (25ch)	
Free-run timer	8 ch (Input: 8 pin)			8 ch (Input: 0 pin)	
Input Capture Unit	12 ch		9 ch	4 ch	
Output Compare Unit	12 ch (Output: 12 pin)		12 ch (Output: 8 pin)	12 ch (Output: 5 pin)	
Stepping motor controller (SMC)	6 gauges		4 gauges	3 gauges	
12bit-A/D converter	48 ch	40 ch	29 ch	17 ch	1unit
Partial Wake Up	analog input 8 ch Trigger 1 ch		analog input 5 ch Trigger 1 ch	-	
CRC	4 ch				
Programmable CRC	1 ch				
Source clock timer	4 ch				
NMI	Available				
External interrupt	24 ch			22 ch	
Internal interrupt	256 vectors				
I2S	2 ch		1ch		One only supports an

Function	S6J33xxJxx 176-pin	S6J33xxHxx 144-pin	S6J33xxGxx 120-pin	S6J33xxFxx 100-pin	Remarks
	(ch0: Output only ch1: Input/ Output)		(ch0: Output only)		output as a function of the sound system.
DDR HSSPI	1 ch			-	
Segment LCD controller	4COM × 32 SEG		4COM × 20 SEG	4COM × 15 SEG	
LCD BUS interface	Data: 18 bit	Data: 18 bit	Data: 8 bit	-	
Ext. BUS interface	Address: 22 bit Data: 16 bit	Address: 15 bit Data: 16 bit	-	-	
CAN FD	4 ch		3 ch	1 ch	
CAN FD RAM (ECC supported)	16 KB/ch It is equivalent to a 128-message buffer per channel of the CAN module				
Multi-function serial interface	12 ch	12 ch	8 ch	5 ch	
I2C	10 ch (Fast mode support : 2 ch)	10 ch (Fast mode support : 2 ch)	6 ch (Fast mode support : 0 ch)	4 ch (Fast mode support : 0 ch)	
Sound Generator	5 ch	5 ch	2 ch	2 ch	
Indicator PWM	1 ch				
Clock Output	1 ch				

Notes:

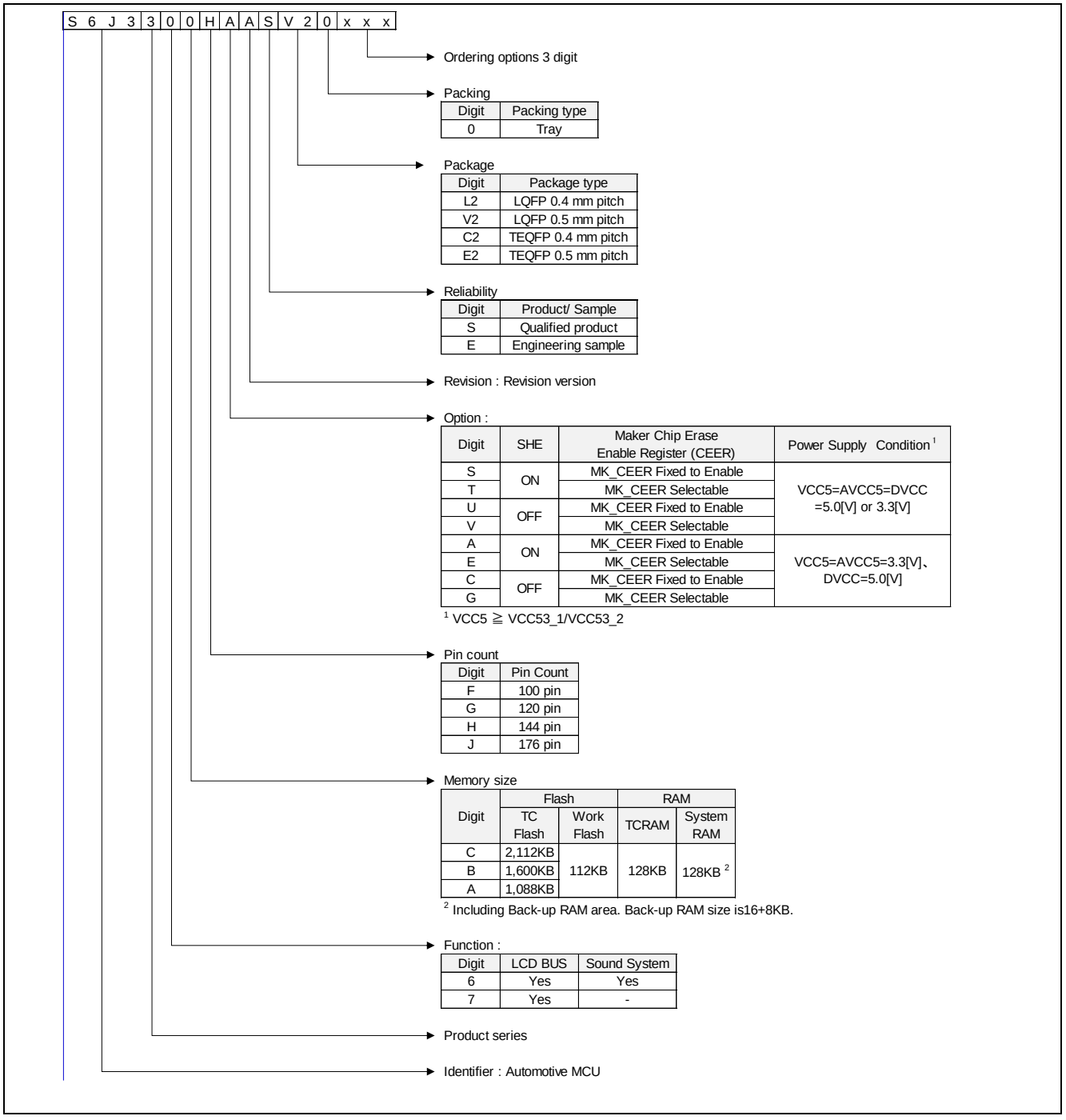
- *The Optional Function are described in Section 2.*
- *The specifications in the table that are related to electrical characteristics only show the typical values. They do not necessarily include the width of characteristics, errors, and so on. They should be seen in the datasheet in detailed.*

2. Optional Function

2.1. Basic Option

The figure shows the optional function and the part number relations of the series.

Figure 2-1



Notes:

- *This table only shows the relation between the optional function and the part numbers; that is, all products are not necessarily available for orders. See the order number on the datasheet and confirm product availability.*
- *Relationship of Package option and Pin count option is as follows*

			Pin count			
			F	G	H	J
			100pin	120pin	144pin	176pin
Package	L2	LQFP 0.4 mm pitch	-	-	○	-
	V2	LQFP 0.5 mm pitch	○	○	○	○
	C2	TEQFP 0.4 mm pitch	-	-	○	-
	E2	TEQFP 0.5 mm pitch	○	○	○	○

○ : Support — : Not support

2.2. ID

ID is specified for each Memory size digit, Option digit and revision.

Memory size	Option	Revision	Chip ID	JTAG ID	SHE Module ID (SHE_MID) *
C	S	B	0x10160101	0x002505CF	0x000F_0300
	T			0x002515CF	
	U			0x002525CF	
	V			0x002535CF	
	A		0x10168101	0x002505CF	
	E			0x002515CF	
	C			0x002525CF	
	G			0x002535CF	
B	S		0x10160101	0x002545CF	
	T			0x002555CF	
	U			0x002565CF	
	V			0x002575CF	
	A		0x10168101	0x002545CF	
	E			0x002555CF	
	C			0x002565CF	
	G			0x002575CF	
A	S		0x10160101	0x002585CF	
	T			0x002595CF	
	U			0x0025A5CF	
	V			0x0025B5CF	
	A		0x10168101	0x002585CF	
	E			0x002595CF	
	C			0x0025A5CF	
	G			0x0025B5CF	

* See "SHE Module ID Register" of the chapter of "Secure Hardware Extension (SHE)" in Traveo™ Platform hardware manual for details.

CHAPTER 3: Product Description

This chapter explains the function feature.

1. Overview
2. Product description
3. Note

CODE: PRODUCT-S6J3360/3370-E1

1. Overview

This chapter explains the product features of S6J3360/S6J3370 series. The description of this chapter should precede the duplicated description on platform manual.

■ Option Features

See the common information of the option features on the hardware manual of Traveo™ Platform.

In the case of option of default and LP-Type, S6J3360/S6J3370 select the LP-Type.

In the case of options that do not include the LP-Type, S6J3360/S6J3370 select the default.

2. Product description

The table shows features.

Table 2-1

Feature	Description
Technology	40nm CMOS technology with embedded FLASH Fully automotive qualified according to ISO/TS 16949 and AEC-Q100
Functional Safety	The product series has some functional safety features suited for ASIL-B application.
Peripherals	See function list.
Power Domain (PD)	See the platform manual and chapter STATE TRANSITION in detail.
Debug and Trace	See the platform manual in detail. − Standard 5-pin JTAG interface − SWD (Serial Wire Debug) interface − 16kB Embedded Trace Buffer 8-bit trace support.
System Control	See the platform manual in detail. Main and sub oscillator is available. − A wide range of 3.6 - 16MHz is available for main oscillator − 32KHz is available for sub oscillator Sub clock is enable/disable by register settings
Clock	See the platform manual in detail. CLK_CLKO (Clock Output Function) is supported.
Embedded CR oscillation	See the platform manual in detail. Stabilization time is as followings. − 30us for 4MHz (Fast clock) − 20us for 100kHz (Slow clock)
Clock Supervisor	See the platform manual in detail. This product series doesn't support clock supervisor output port. (Related register and internal circuit is implemented.)
Reset	See the platform manual in detail. Following resets are not mounted on this device. − INITX − SRSTX (and nSRST pin)
Hardware watchdog	See the platform manual in detail. Hardware watchdog function stops during PSS mode. In the related register of HWDG_CFG, the bit ALLOWSTOPCLK is always read as 1 (HWDG_CFG.ALLOWSTOPCLK=1). The product series doesn't support Watchdog Counter Monitor Output port. (Related register and internal circuit is implemented.).
Software watchdog	See the platform manual in detail. The product series doesn't support Watchdog Counter Monitor Output port. (Related register and internal circuit is implemented.).
Standby mode	See the platform manual in detail. Standby mode with 5V (or 3.3V) single power supply is available.

Feature	Description
Partial wakeup	Partial wakeup function is mounted. This mode is one of the PSS modes.
PLL / SSCG PLL	See the platform manual in detail. Use case assumption is following. − PLL ➢ Peripherals ➢ Trace clock − SSCG ➢ CPU core ➢ Peripherals
External Interrupts	Sensitive: H level, L level, pos-edge, neg-edge, both-edge Up to 24 channels
NMI	Sensitive: L level 1 NMI pin.
Memory Protection	16 region MPU inside CPU core (see the Cortex™-R5 Revision:r1p2 Technical Reference Manual) 8 region MPU for SHE (see Memory Protection Unit for AXI of SHE in platform manual) 16 region MPU for DMA controller (see MPU16 AHB in platform manual) To configure Lock or Unlock for both MPUXn_UNLOCK and MPUHn_UNLOCK. − Lock: 0x112ABB56 − Unlock: 0xACCABB56
Peripheral Protection	See the platform manual in detail. Protected peripherals are described in the base address map.
Internal Memories System SRAM	See the platform manual in detail. Max memory size is smaller than the above value on some product type. Please see the " Optional Function " of FUNCTION LIST chapter.
Internal Memories TCRAM	See the platform manual in detail. Max memory size is smaller than the above value on some product type. Please see the " Optional Function " of FUNCTION LIST chapter.
Internal Memories Backup area in System SRAM	Up to 64kByte Backup area can only be operated in RUN mode (normal operation mode). In other mode the memory content should be retained, but it cannot be operated. SLEEP control for Buckup area is not supported and cannot be used.

Feature	Description
Embedded Program/Work Flash Memory	Embedded Program Flash can be accessed with 0wait cycle if CPU frequency is 80MHz or less. 0-wait-cycle: 80MHz or less. 1-wait-cycle: 132MHz or less. Work Flash can be accessed with 0-wait-cycle if TCFlash frequency is 12.5MHz or less. 6-wait-cycle: 80MHz or less. 10-wait-cycle: 132MHz or less. The wait-cycle setting see the Traveo™ Platform hardware manual in details. The maximum frequency should be referred in datasheet. Erase suspend is supported. Reading and writing to the other sector are possible when Flash Erase is suspended. Serial Flash programing and Parallel Flash programing are supported. Margin mode is not supported.
Internal Power Domain	PD1: Always ON PD2: Cortex R5F platform/ additional peripherals PD4: Backup RAM in Always On domain * The chapter of the block diagram explains in detailed.
Power Supply	External 5V (or 3.3V) is required. Built in LDO provides internal voltage. There are constraints of power on/off sequence.
Low Voltage Detection	LVD for external voltage is supported. LVD for internal voltage is supported. See the specification of the detected level on the datasheet.
Low voltage detection for RAM retention (RVD)	RVD for RAM retention is effective during the standby mode only. That is, it is only for the retention RAM that the function is available.
Resource inter-connect	The output signal of some resources can be inputted to the other resource.
I/O Ports	5V general purpose I/O Multi input level and multi output drivability Pull-up, pull-down function is available. Resource input and output is multiplexed. +B input is allowed many pins.

Feature	Description
A/D Converter	12bit resolution, 1 unit 48 channels of analog input for TEQFP176 and LQFP176 40 channels of analog input for TEQFP144 and LQFP144 29 channel of analog input for TEQFP120 and LQFP120 17 channel of analog input for TEQFP100 and LQFP100 External trigger and timer trigger are available. ch.0-63 of logical channels are supported. For ch.0-47, HWTRG can be selected from combination specified on PORT CONFIGURATION. For ch.48-63, HWTRG can be assigned on P1_09 (ADC_TRG0_0) only. Relocation or resource combination cannot be selected. The description of the A/D converter function should be referred in the S6J3360/S6J3370 hardware manual.
CRC	4 CRC calculators (CRC32, CCITT)
Programmable CRC	DMA support
Base Timer	16-bit PWM Timer 16-bit PPG Timer 16/32-bit Reload Timer 16/32-bit PWC Timer.
Base Timer Simultaneous Operation	16-bit Global Timer 1ch
Reload Timer	32-bit Reload Timer
I/O Timer	32-bit Freerun Timer 32-bit Input Capture 32-bit Output Compare
QPRC (Quad Precision Counter)	Quad Position & Revolution Counter 16-bit position counter
Multi-Functional Serial (MFS)	UART mode CSIO (SPI) mode LIN mode I^2C mode Only 2 ports of MFS have the dedicated I/O for I^2C. See the datasheet in detailed. CTS/RTS is not mounted (hardware flow control is not supported for this series.) This model does not support Wake Up Function.
CAN-FD	Flexible data rate is supported. 16KB/ch of message RAM is available. The clock output from CAN pre-scaler is supplied to every CAN. ECC error generation function of the message RAM is not supported for this device. Therefore CAN FD ECC Error Insertion Control Register (FDFECR) is not writeable.
Real Time Clock (RTC) with auto-calibration	See the platform manual in detailed.
SHE	See the platform manual in detailed.

Feature	Description
Source Clock Timer	See the platform manual in detailed.
BUS DIAGNOSIS FUNCTION	The bus diagnosis function prevents an LSI malfunction by checking data that was output on a bus during access to each resource.
I2S	If the JTAG debugger reads out I2S1_RXFDAT0 to 15:RXDATA register under debug state (breakpoint), the read pointer will be incremented and the data will be processed.
JTAG boundary scan	MODE=0 + RSTX input is needed to entry boundary scan mode correctly.
Parallel flash programming	Parallel writer mode needs MD=0+RSTX to transit the mode correctly.

Note:

- *The description of the preliminary documentation will be changed without any notification.*

3. Note

3.1. Status Flag Clear

Note that the hardware operation of a write access to a register which has status flags may be later than program operations of software. The delay results from the fact that the write accesses and their signals are operated through multiple buses.

For example, when a software program intends to return from interrupt software routine (ISR) after clearing the interrupt flag, the return instruction may practically be executed before the completion of the write access as for hardware operation. That is, the CPU execution may immediately jump to the ISR again after returning from the ISR because the flag is not cleared.

To avoid such a phenomenon, the execution of Data Memory Barrier (DMB) instruction between the write access and the return instruction is recommended for the program. The return instruction is never executed before the completion of DMB execution.

It should be considered that the method takes a number of execution cycles and that it may cause some influence to application performance. For example, one time of the DMB execution is enough after a number of continuous flag clear operations.

3.2. Error Response

Error response is generated when access occurs to the following register and bit offset.

Function	Register	Bit Offset	Access	Error Type
12/10/8-BIT ANALOG TO DIGITAL CONVERTER	ADC12Bn_CHSTAT0 to 31	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_CD0 to 31	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_CDONEIRQ0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_GRPIRQ0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_RCIRQ0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_PCIRQ0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_TRGST0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_RCOTF0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_TRGOR0	Whole register	Write(B,H,W)	Bus error
	ADC12Bn_MCSTAT0 to 3	Whole register	Write(B,H,W)	Bus error
	(Other reserved area)	(Other reserved area)	Read / Write(B,H,W)	Bus error

3.3. Register Initial Value

The table shows initial value of Register bit which is uniquely specified for the product series.

Register	Bit	Initial Value	Description
SYSC0_APPLVDCFGR	LVDL1S	0	-
SYSC0_APPLVDCFGR	LVDL1V	000	-
SYSC0_APPLVDCFGR	LVDL1E	1	-
SYSC0_APPLVDCFGR	Bit22	0	-
SYSC0_APPLVDCFGR	Bit19 to Bit17	000	-

Register	Bit	Initial Value	Description
SYSC0_APPLVDCFGR	Bit16	0	-
SYSC0_APPLVDCFGR	LVDH1S	0	-
SYSC0_APPLVDCFGR	LVDH1V	111	-
SYSC0_APPLVDCFGR	LVDH1E	1	-
SYSC0_APPLVDCFGR	LVDH2S	0	-
SYSC0_APPLVDCFGR	LVDH2V	000	-
SYSC0_APPLVDCFGR	LVDH2E	0	-
SYSC0_STSLVDCFGR	LVDL1R	1	-
SYSC0_STSLVDCFGR	LVDL1S	0	-
SYSC0_STSLVDCFGR	LVDL1V	000	-
SYSC0_STSLVDCFGR	LVDL1E	1	-
SYSC0_STSLVDCFGR	Bit23	0	-
SYSC0_STSLVDCFGR	Bit22	0	-
SYSC0_STSLVDCFGR	Bit19 to Bit17	000	-
SYSC0_STSLVDCFGR	Bit16	0	-
SYSC0_STSLVDCFGR	LVDH1R	1	-
SYSC0_STSLVDCFGR	LVDH1S	0	-
SYSC0_STSLVDCFGR	LVDH1V	111	-
SYSC0_STSLVDCFGR	LVDH1E	1	-
SYSC0_STSLVDCFGR	LVDH2R	0	-
SYSC0_STSLVDCFGR	LVDH2S	0	-
SYSC0_STSLVDCFGR	LVDH2V	000	-
SYSC0_STSLVDCFGR	LVDH2E	0	-
SYSC0_PSSLVDCFGR	LVDL1S	0	1 should be written only if LVDL1V is changed from initial value.
SYSC0_PSSLVDCFGR	LVDL1V	000	-
SYSC0_PSSLVDCFGR	LVDL1E	1	-
SYSC0_PSSLVDCFGR	Bit22	0	Reserved
SYSC0_PSSLVDCFGR	Bit19 to Bit17	000	Reserved
SYSC0_PSSLVDCFGR	Bit16	0	Reserved
SYSC0_PSSLVDCFGR	LVDH1S	0	1 should be written only if LVDH1V is changed from initial value.
SYSC0_PSSLVDCFGR	LVDH1V	111	-
SYSC0_PSSLVDCFGR	LVDH1E	1	-
SYSC0_PSSLVDCFGR	LVDH2S	0	Reserved
SYSC0_PSSLVDCFGR	LVDH2V	000	Reserved
SYSC0_PSSLVDCFGR	LVDH2E	0	Reserved
SYSC0_RUNLVDCFGR	LVDL1S	0	1 should be written only if LVDL1V is changed from initial value.
SYSC0_RUNLVDCFGR	LVDL1V	000	
SYSC0_RUNLVDCFGR	LVDL1E	1	-
SYSC0_RUNLVDCFGR	Bit22	0	Reserved
SYSC0_RUNLVDCFGR	Bit19 to Bit17	000	Reserved
SYSC0_RUNLVDCFGR	Bit16	0	Reserved
SYSC0_RUNLVDCFGR	LVDH1S	0	1 should be written only if LVDH1V is changed from initial value.
SYSC0_RUNLVDCFGR	LVDH1V	111	-
SYSC0_RUNLVDCFGR	LVDH1E	1	-

Register	Bit	Initial Value	Description
SYSC0_RUNLVDCFGR	LVDH2S	0	Reserved
SYSC0_RUNLVDCFGR	LVDH2V	000	Reserved
SYSC0_RUNLVDCFGR	LVDH2E	0	Reserved
SYSC0_SPESPSWCFGR0	-	0x00000000	Writing doesn't affect anything.
SYSC0_SPESPSWCFGR1	-	0x11000001	Writing doesn't affect anything.
SYSC0_SPEWPSWCFGR0	-	0x00000000	Writing doesn't affect anything.
SYSC0_SPEWPSWCFGR1	-	0x11000001	Writing doesn't affect anything.

Note:

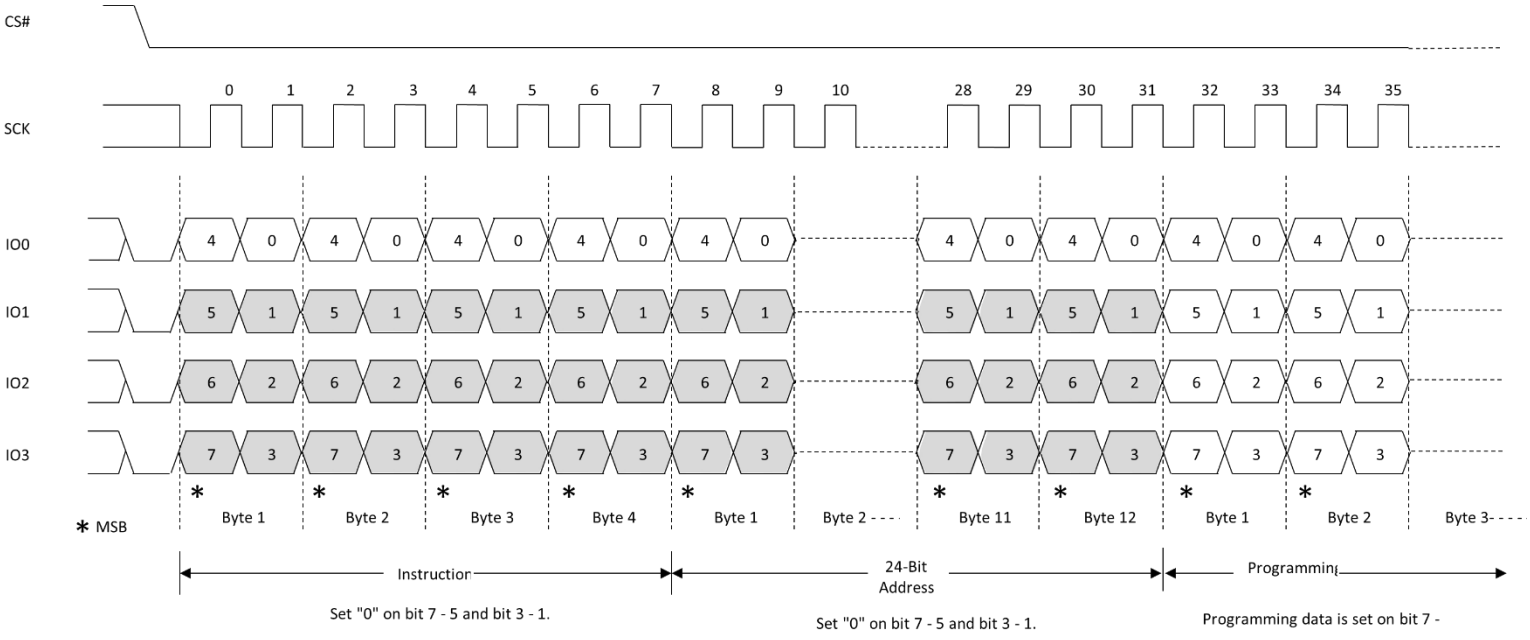
- *SYSC0_SPESPSWCFGR0 and 1, and SYSC0_SPEWPSWCFGR0 and 1 are not software configurable registers but hardware specific registers for the system.*

3.4. Restriction

Function	Related Register and Configuration	Restriction	Remark
Interrupt Controller	IRC0_IRQS8 to IRC0_IRQS15 IRC0_IRQR8 to IRC0_IRQR15 IRC0_IRQSIS8 to IRC0_IRQSIS15 IRC0_IRQCES8 to IRC0_IRQCES15 IRC0_IRQCEC8 to IRC0_IRQCEC15 IRC0_IRQCE8 to IRC0_IRQCE15 IRC0_IRQHS8 to IRC0_IRQHS15 IRC0_IRQRS8 to IRC0_IRQRS15 IRC0_IRQPS8 to IRC0_IRQPS15	Not support these registers because 256 interrupt vectors.	-
Profile update	System Status Register (SYSC0_SYSSTSR) PSSSTS0 bit RUNSTS0 bit PSSDF0 bit RUNDF0 bit	Before profile update, confirm the status is not updating by STS0 bit. After profile update, confirm the profile update completion by DF0 bit and the software go ahead with next handling.	-
Low Power Consumption	MCG_IRSR0 IRQ_PW bit	This model does not support IRQ_PW bit in MCG_IRSR0 register. This bit is always read to 0.	-
Main clock division	SYSC_MOSCCNTR DIV2SEL bit	Set SYSC_MOSCCNTR.DIV2SEL=1. Main clock pulse that is no division in chip deforms easily by noise. It causes running out of control.	-
Main clock fast clock input enable	SYSC_MOSCCNTR FCIMEN bit	Set SYSC_MOSCCNTR.FCIMEN=0. Main clock needs setting to 0 to oscillate.	-

Function	Related Register and Configuration	Restriction	Remark
PPU support of DDRHSSPI	PPU0_SR.VD, PPU0_SR.VP, PPU0_SR.VW, PPU0_SR.VL[4:0]	This series has a product specific LCD subsystem and DDRHSSPI. PPU protects the DDRHSSPI against an illegal access, and then only responds bus error. Please note the status flag bits which are served as Traveo platform function does not support the product specific PPU and DDRHSSPI, that is, the status bits on PPU0_SR are no set when the PPU detects the illegal access.	-
DDRHSSPI	DDRHSSPIn_PCC0-3.SS2CD[1:0] Quad Page Program	DDRHSSPIn_PCC0-3.SS2CD[1:0] must be set 2'b11 Regarding Quad Page Program, the instruction and address must be transferred in direct mode with quad (or dual quad) mode. e.g. The transfer order of Quad Page Program instruction (0x38) are: - Quad mode: 0x00, 0x11, 0x10, 0x00. - Dual quad mode: 0x00, 0x00, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00.	Please refer to the following figure 2-1. Regarding the data shift direction, please refer to the chapter for DDR High Speed SPI Controller.

Fig.2-1: Quad Page Program by Quad Mode



CHAPTER 4: Block Diagram

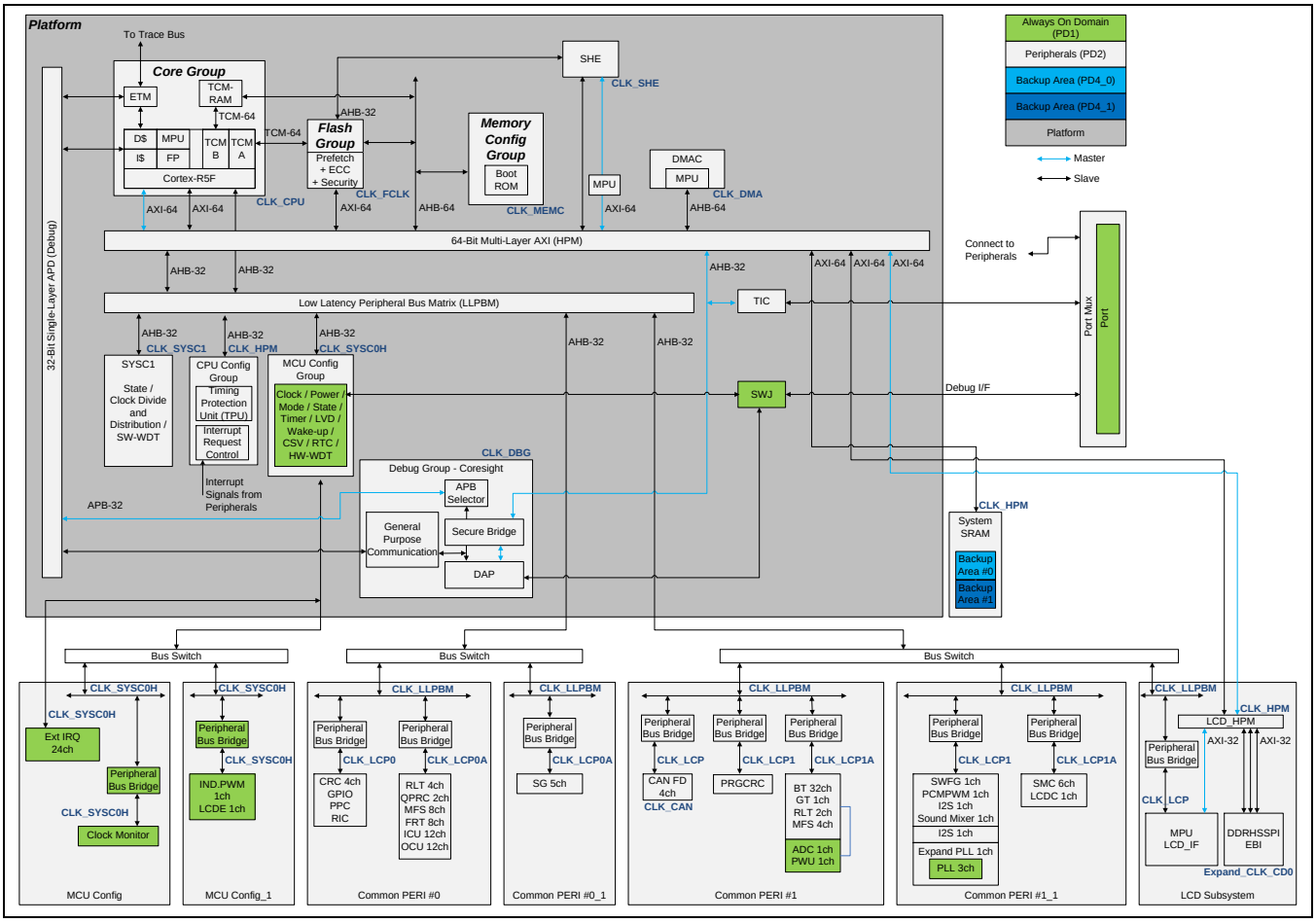
This chapter explains the block diagram.

1. Block diagram
2. Note

CODE: BLOCK_DIAGRAM-S6J3360/3370-E1

1. Block Diagram

Figure 1-1



2. Note

No description.

CHAPTER 5: Clock Configuration

This chapter explains the clock configuration.

1. Overview
2. Operation
3. Remark

CODE: CLOCKCFG-JUPI-E1.1

1. Overview

This chapter describes clock selection and configuration of each function.

See the chapter of the “Clock System” in Traveo™ Platform hardware manual before referring this chapter.

2. Operation

The source clock of each function should be selected as following table.

Table 2-1

Function	Clock Name	Source Clock System	Configuration
CPU	CLK_CPU	SSCG0	See Traveo™ Platform hardware manual
FLASH	CLK_FCLK	SSCG0	See Traveo™ Platform hardware manual
HPM	CLK_HPM	SSCG0	See Traveo™ Platform hardware manual
DMA	CLK_DMA	SSCG0	See Traveo™ Platform hardware manual
LLPBM	CLK_LLPBM	SSCG0	See Traveo™ Platform hardware manual
MCU Configuration	CLK_SYSC0H CLK_COMH	SSCG0	See Traveo™ Platform hardware manual
CAN	CLK_CAN	PLL0	See Traveo™ Platform hardware manual
CoreSight	CLK_TRC	PLL0	See Traveo™ Platform hardware manual
Hardware watchdog timer	CLK_HWWDT	Internal CR oscillator (High frequency)	See Traveo™ Platform hardware manual
		Internal CR oscillator (Low frequency)	See Traveo™ Platform hardware manual
Software watchdog timer	CLK_SWWDT	Main clock	See Traveo™ Platform hardware manual
		Sub clock	See Traveo™ Platform hardware manual
		Internal CR oscillator (High frequency)	See Traveo™ Platform hardware manual
		Internal CR oscillator (Low frequency)	See Traveo™ Platform hardware manual
Real time clock	CLK_RTC	Main clock	See Traveo™ Platform hardware manual
		Sub clock	See Traveo™ Platform hardware manual
		Internal CR oscillator (Low frequency)	See Traveo™ Platform hardware manual
MCU Configuration_1	CLK_SYSC0H	SSCG0	See Traveo™ Platform hardware manual
CRC, GPIO, PPC, RIC in Common PERI #0	CLK_LCP0	SSCG0	See Traveo™ Platform hardware manual
RLT, QPRC, MFS, FRT, ICU, OCU in Common PERI #0	CLK_LCP0A	SSCG0	See Traveo™ Platform hardware manual
SG in Common PERI #0_1		PLL0	See Traveo™ Platform hardware manual
PRGCRC in Common PERI #1	CLK_LCP1	SSCG0	See Traveo™ Platform hardware manual
I2S1, Expand PLL (Bus) in Common PERI #1_1			

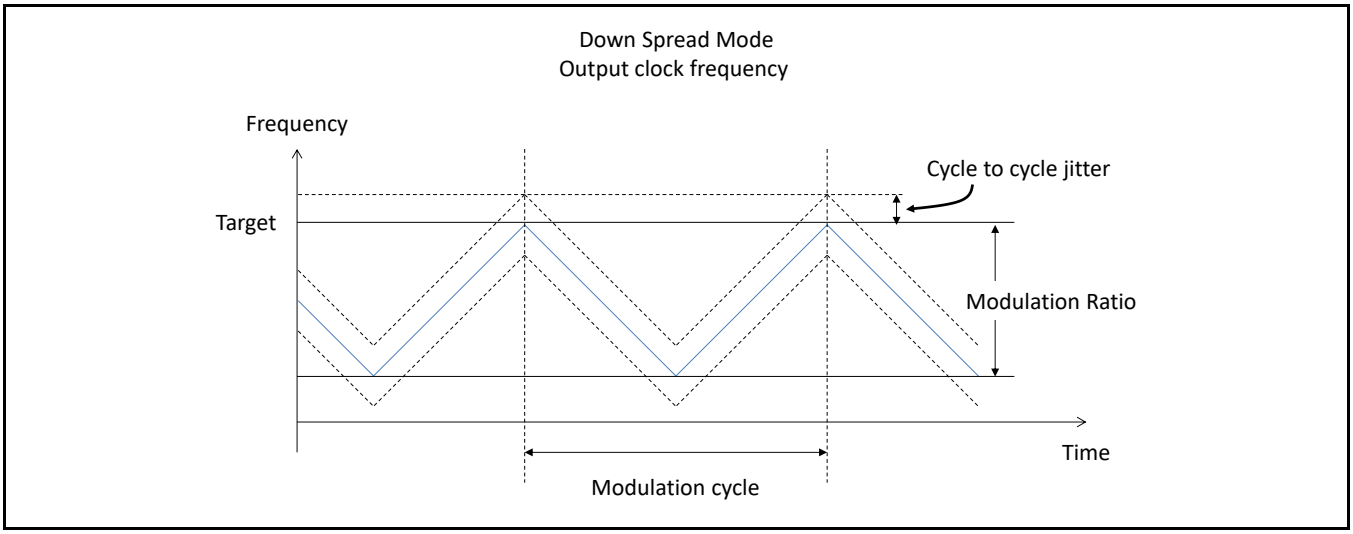
Function	Clock Name	Source Clock System	Configuration
BT, GT, RLT, MFS, ADC in Common PERI #1	CLK_LCP1A	SSCG0	See Traveo™ Platform hardware manual
SMC, LCDC(Bus) in Common PERI #1_1		PLL0	See Traveo™ Platform hardware manual
CAN(Bus) in Common PERI #1	CLK_LCP	SSCG0	See Traveo™ Platform hardware manual
MPU, LCD_IF(Bus) in LCD Subsystem			
DDR HSSPI in LCD Subsystem	Expand_CLK_CD0	Expand PLL0	See “EXPAND PLL” chapter.
SWFG, Sound Mixer (Bus) in Sound System	Expand_CLK_CD1	Expand PLL1	See “EXPAND PLL” chapter.
SWFG, Sound Mixer (Operation) in Sound System	Expand_CLK_CD1A	Expand PLL1	See “EXPAND PLL” chapter.
I2S0 ,PCM-PWM (Bus) in Sound System	Expand_CLK_CD1B	Expand PLL1	See “EXPAND PLL” chapter.
PCM-PWM (Operation) in Sound System	Expand_CLK_CD2	Expand PLL2	See “EXPAND PLL” chapter.
LCD_IF (Operation) in LCD Subsystem	CLK_HPM	SSCG0	See Traveo™ Platform hardware manual
EBI in LCD Subsystem			

2.1. Spread Spectrum Clock Generator (SSCG)

Target frequency of SSCG should be referred in Datasheet.

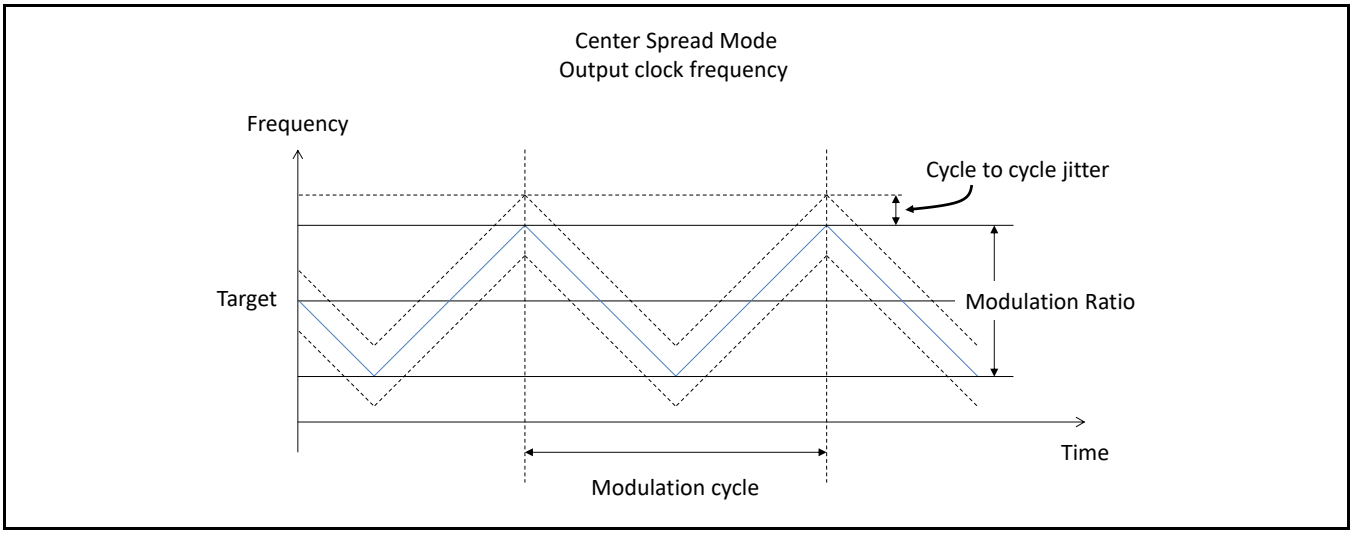
■ Down Spread Mode

Figure 2-1 Down Spread Mode



■ Center Spread Mode

Figure 2-2 Center Spread Mode



3. Remark

See the “Clock System” in Traveo™ Platform hardware manual before referring this chapter.

Notes:

- *If you want to know a combination of a peripheral function and its source clock, you need to see the group name from the table of base address map on this manual at first.*
- *The group name and its clock source are described in the chapter of “Platform Overview” Configuration and “Clock System”. See Traveo™ Platform hardware manual.*
- *Clock frequency can be seen in datasheet of S6J3360 series.*
- *Each clock need to keep the specific ratio for CLK_CPU as following table.*

In case of CLK_LCP0A and CLK_LCP1A with PLL0, not need to keep this specific ratio between CLK_CPU and CLK_LCP0A, CLK_LCP1A.

The table shows the possible setting for CLK_CPU = 132MHz, 80MHz, 40MHz.

- *The frequency of Sound System clocks should satisfy the following conditions.*

Expand_CLK_CD1 : Expand_CLK_CD1A = 1:1

Expand_CLK_CD1 : Expand_CLK_CD1B = 1:1 or 2:1 or 3:1

Expand_CLK_CD1A >= 100MHz

Table 2-2

<table>
<tr><th rowspan="2">CLOCK</th><th colspan="3">CLK_CPU : CLOCK</th></tr>
<tr><th>≦132 MHz</th><th>≦80 MHz</th><th>≦40 MHz</th></tr>
<tr><td>CLK_FCLK
CLK_LCP</td><td rowspan="2">1:2n</td><td>1:n</td><td rowspan="4">1:n</td></tr>
<tr><td>CLK_ATB
CLK_DBG</td><td rowspan="3">1:2n</td></tr>
<tr><td>CLK_HPM/DMA/MEMC/SHE
CLK_SYS1
CLK_LCP0
CLK_LCP0A
CLK_LCP1
CLK_LCP1A
CLK_SYSC0H/COMH</td><td>1:4n</td></tr>
</table>

n : Same divide number

Table 2-3

	CLK_CPU = 132 MHz		CLK_CPU = 80 MHz		CLK_CPU = 40 MHz		
	n=1	n=2	n=1	n=2	n=1	n=2	n=4
CLK_FCLK	66 MHz	33 MHz	80 MHz	40 MHz	40 MHz	20 MHz	10 MHz
CLK_LCP	66 MHz	33 MHz	80 MHz	40 MHz	40 MHz	20 MHz	10 MHz
CLK_ATB	66 MHz	33 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_DBG	66 MHz	33 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_HPM/DMA/MEMC/SHE	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_SYS1	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_LCP0	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_LCP0A	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_LCP1	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_LCP1A	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz
CLK_SYSC0H/COMH	33 MHz	16.5 MHz	40 MHz	20 MHz	40 MHz	20 MHz	10 MHz

CHAPTER 6: Operation Mode

This chapter explains operation mode.

1. Overview
2. Configuration
3. Registers

CODE: MODE-JUPI-E1.1

1. Overview

This section gives a description of operation mode.

The mode controller determines the operation mode of MCU. It supports,

- User mode
- Serial programming mode (with synchronous communication)
- Serial programming mode (with asynchronous communication)
- JTAG boudary scan mode

2. Configuration

Operation Mode	PORT		
	MODE	P1_12(SOT0)	P1_14(SIN0)
User Mode	1	-	-
Serial Programming Mode (Sync)	0	1	0
Serial Programming Mode (Async)	0	1	1
JTAG Boundary Scan Mode	0	0	0

Notes:

- *User mode can be applied to Software debugging using JTAG interface with ICE.*
- *As for serial programming, see the chapter of "Serial Programming" on this manual.*

3. Registers

See Traveo™ Platform hardware manual.

CHAPTER 7: Memory and Base Address Map

This chapter explains the memory map and the base address map of registers.

1. Overview
2. Memory map
3. Base address map

CODE: MEMORYMAP-JUPI-E1.0

1. Overview

When MPU attribute of Cortex-R5F is configured as "Normal", store buffer inside Cortex-R5F can operate and write data can be merged. To avoid influence of this data merger, MPU attribute "Device" or "Strongly Ordered" should be used.

MPU attribute "Device" or "Strongly Ordered" must be used for areas below, to avoid this influence.

- Peripheral area [B000_0000 ~ B7FF_FFFF]
- Peripheral area [B800_0000 ~ BFFF_FFFF]
- Error Config area (ERRCFG) [FFFE_E000 ~ FFFE_FFFF]

MPU attribute "Device" or "Strongly Ordered" is required for accesses to areas below, in particular situation.

- FLASH Memory (when writing commands)
- External Bus Interface (when writing commands to NAND FLASH)

2. Memory Map

START Address	END Address	Group	Part
0000_0000	0001_FFFF	Internal area for CR5 Complex	TCRAM 128KB@Max
0002_0000	007F_FFFF		Reserved
0080_0000	009E_FFFF		Reserved
009F_0000	009F_FFFF		TCFLASH 64KB@Max (small sector 8KB x 8)
00A0_0000	00BF_FFFF		TCFLASH 2MB@Max
00C0_0000	00FF_FFFF		Reserved
0100_0000	019E_FFFF		Reserved
019F_0000	019F_FFFF		AXI_FLASH 64KB @Max (small sector 8KB x 8)
01A0_0000	01BF_FFFF		AXI_FLASH 2MB@Max
01C0_0000	01FF_FFFF		Reserved
0200_0000	0201_FFFF	Shared Flash and memory area	System SRAM 128KB@Max
0202_0000	027F_FFFF		Reserved
0280_0000	0280_0FFF		Reserved
0280_1000	03FF_FFFF		Reserved
0400_0000	05FF_FFFF		AXI_SLAVE_CORE0
0600_0000	07FF_FFFF		Reserved
0800_0000	09FF_FFFF		Reserved
0A00_0000	0BFF_FFFF		Reserved
0C00_0000	0DFF_FFFF		Reserved
0E00_0000	0E01_BFFF		WORK_FLASH 112KB@Max (mirror area 1)
0E01_C000	0E1F_FFFF		Reserved
0E20_0000	0E21_BFFF		WORK_FLASH 112KB@Max (mirror area 3)
0E21_C000	0E2F_FFFF		Reserved
0E30_0000	0E31_BFFF		WORK_FLASH 112KB@Max (mirror area 4)
0E31_C000	0E3F_FFFF		Reserved
0E40_0000	0E7F_FFFF		Reserved
0E80_0000	0E81_FFFF		System SRAM 128KB@Max (mirror)
0E82_0000	0EFF_FFFF		Reserved
0F00_0000	0FFF_FFFF		Reserved
1000_0000	1FFF_FFFF	LCD_subsystem	External Bus Interface(SRAM/FLASH)
2000_0000	2FFF_FFFF	Reserved	Reserved
3000_0000	3FFF_FFFF		
4000_0000	4FFF_FFFF	Reserved	Reserved
5000_0000	501F_FFFF	Reserved	Reserved
5020_0000	5020_03FF	Reserved	Reserved
5020_0400	5020_3FFF	Reserved	Reserved
5020_4000	5020_43FF	Reserved	Reserved
5020_4400	5021_3FFF	Reserved	Reserved
5021_4000	5021_43FF	Reserved	Reserved
5021_4400	5021_47FF	Reserved	Reserved
5021_4800	503F_FFFF	Reserved	Reserved
5040_0000	504F_FFFF	Reserved	Reserved

START Address	END Address	Group	Part
5050_0000	5FFF_FFFF	Reserved	Reserved
6000_0000	7FFF_FFFF	Reserved	Reserved
8000_0000	8FFF_FFFF	LCD_subsystem	HSSPI0_MEMORY
9000_0000	9FFF_FFFF	Reserved	Reserved
A000_0000	A010_7FFF	Reserved	Reserved
A010_8000	A010_80FF	System SRAM	System SRAM registers (mirror)
A010_8100	A041_1FFF	Reserved	Reserved
A041_2000	A041_20FF	MEM Config Group	Wflash Control Status Register (mirror)
A041_2100	A118_FFFF	Reserved	Reserved
A119_0000	A11F_FFFF	Bit RMW alias	Bit RMW alias for MEMC (mirror) (Covers B041_2000 -- B041_FFFF)
A120_0000	AFFF_FFFF	Reserved	Reserved
B000_0000	B7FF_FFFF	Peripheral area	Peripheral area
B800_0000	B802_87FF	Peripheral area	Peripheral area
B802_8800	BFFF_FFFF	Reserved	Reserved
C000_0000	EFFF_FFFF	Reserved	Reserved
F000_0000	FFFE_DFFF	BootROM/ERRCFG area	Reserved
FFFE_E000	FFFE_EFFF		ERRCFG
FFFE_F000	FFFE_FFFF		
FFFF_0000	FFFF_FFFF		BootROM

Memory Map of System SRAM

START Address	END Address	Part
0200_0000	0200_3FFF	Backup area (PD4_0)
0200_4000	0200_5FFF	Backup area (PD4_1)
0200_6000	0201_FFFF	Non backup area (PD2)

Notes:

- *Start Address is same for every memory size variation.*
- *End Address is specified for maximum memory size. It depends on each memory size variation.*
- *See memory size at 2.1 Basic option on Chapter: Function List.*

3. Base Address Map

START Address	END Address	Group	Function	PPU_No
B000_0000	B00F_FFFF	Reserved	Reserved	-
B010_0000	B010_03FF	LCD_subsystem	External Bus Interface registers	0
B010_0400	B010_0FFF	Reserved	Reserved	-
B010_1000	B010_13FF	LCD_subsystem	DDR_HSSPI0	1
B010_1400	B010_7FFF	Reserved	Reserved	-
B010_8000	B010_80FF	System SRAM	System SRAM registers	2
B010_8100	B01F_FFFF	Reserved	Reserved	-
B020_0000	B02F_FFFF	Reserved	Reserved	-
B030_0000	B030_7FFF	SYSC1	System Controller #1	4
B030_8000	B03F_FFFF	SYSC1	SWDT	5
B040_0000	B040_7FFF	CPU Config Group	IRC0	21
B040_8000	B040_FFFF	CPU Config Group	TPU0	19
B041_0000	B041_0FFF	CPU Config Group	TCRAM Control Status Register	16
B041_1000	B041_1FFF	CPU Config Group	TCFlash Control Status Register	17
B041_2000	B041_20FF	MEM Config Group	Wflash Control Status Register	18
B041_2100	B04F_FFFF	Reserved	Reserved	-
B050_0000	B050_0FFF	Debug Group	DAP ROM table	-
B050_1000	B050_1FFF	Debug Group	ETB	-
B050_2000	B050_2FFF	Debug Group	CTI #4	-
B050_3000	B050_3FFF	Debug Group	TPIU	-
B050_4000	B057_FFFF	Debug Group	TRACE FUNNEL	-
B058_0000	B058_FFFF	Debug Group	CR5_RomTable	-
B059_0000	B059_1FFF	Debug Group	CORE0	-
B059_2000	B059_7FFF	Reserved	Reserved	-
B059_8000	B059_8FFF	Debug Group	CTI #0	-
B059_9000	B059_BFFF	Reserved	Reserved	-
B059_C000	B059_CFFF	Debug Group	ETM0	-
B059_D000	B05B_FFFF	Reserved	Reserved	-
B05C_0000		Debug Group	Security Checker	-
	B05F_FFFF	Reserved	Reserved	-
B060_0000	B060_007F	MCU Config Group	Protection register area	51
B060_0080	B060_00FF	MCU Config Group	RUN profile register area	51
B060_0100	B060_017F	MCU Config Group	PSS profile register area	51
B060_0180	B060_01FF	MCU Config Group	APP profile register area	51
B060_0200	B060_027F	MCU Config Group	STS profile register area	51
B060_0280	B060_02FF	MCU Config Group	System register area	51
B060_0300	B060_037F	MCU Config Group	CSV	51
B060_0380	B060_03FF	MCU Config Group	RESET	51
B060_0400	B060_047F	MCU Config Group	SCT (Fast CR)	34
B060_0480	B060_04FF	MCU Config Group	SCT (Slow CR)	33
B060_0500	B060_057F	MCU Config Group	SCT (Main clock)	35

START Address	END Address	Group	Function	PPU_No
B060_0580	B060_05FF	MCU Config Group	SCT (Sub clock)	36
B060_0600	B060_067F	MCU Config Group	Clock System	51
B060_0680	B060_06FF	MCU Config Group	Special register area	51
B060_0700	B060_07FF	MCU Config Group	Debug register area	51
B060_0800	B060_BFFF	MCU Config Group	MODEC	55
B060_C000	B060_FFFF	MCU Config Group	HWDT	52
B061_0000	B061_7FFF	Reserved	Reserved	-
B061_8000	B061_FFFF	MCU Config Group	RTC	32
B062_0000	B063_FFFF	MCU Config Group	Ext-IRQ	53
B064_0000	B064_0FFF	MCU Config Group	Indicator PWM	364
B064_1000	B064_1FFF	MCU Config Group	LCDE	365
B064_2000	B064_2FFF	Reserved	Reserved	-
B064_3000	B064_3FFF	MCU Config Group	Clock Monitor	-
B064_4000	B065_FFFF	Reserved	Reserved	-
B066_0000	B067_FFFF	Reserved	Reserved	-
B068_0000	B068_7FFF	Reserved	Reserved	-
B068_8000	B068_83FF	Reserved	Reserved	-
B068_8400	B068_87FF	MCU Config Group	CR_CALIBRATION	38
B068_8800	B068_8BFF	MCU Config Group	MCG_IRS	42
B068_8C00	B068_FFFF	MCU Config Group	CAN_PRESCALER	43
B069_0000	B06A_7FFF	Reserved	Reserved	-
B06A_8000	B06B_FFFF	Reserved	Reserved	-
B06C_0000	B06F_FFFF	Reserved	Reserved	-
B070_0000	B07F_FFFF	Reserved	Reserved	-
B080_0000	B0FF_FFFF	Bit RMW alias	Bit RMW alias for MCU Config (Covers B060_0000 -- B06F_FFFF)	-
B100_0000	B10F_FFFF	Bit RMW alias	Bit RMW alias for SYSC1 (Covers B030_0000 -- B031_FFFF)	-
B110_0000	B118_FFFF	Bit RMW alias	Bit RMW alias for CPU Config (Covers B040_0000 -- B041_1FFF)	-
B119_0000	B11F_FFFF	Bit RMW alias	Bit RMW alias for MEMC (Covers B041_2000 -- B041_FFFF)	-
B120_0000	B1FF_FFFF	Reserved	Reserved	-
B200_0000	B20F_FFFF	SHE Group	SHE configuration registers (1MB)	63
B210_0000	B3FF_FFFF	Reserved	Reserved	-
B400_0000	B46F_FFFF	Reserved	Reserved	-
B470_0000	B470_3FFF	CPU Config Group	DMAC #0 registers	64
B470_4000	B470_7FFF	Reserved	Reserved	-
B470_8000	B470_FFFF	Reserved	Reserved	-
B471_0000	B471_0FFF	CPU Config Group	MPU for DMAC #0	66
B471_1000	B471_1FFF	Reserved	Reserved	-
B471_2000	B471_3FFF	Reserved	Reserved	-
B471_4000	B471_4FFF	CPU Config Group	DMA Complex #0 registers (Additional registers, RLTs)	68

START Address	END Address	Group	Function	PPU_No
B471_5000	B471_5FFF	Reserved	Reserved	-
B471_6000	B471_7FFF	Reserved	Reserved	-
B471_8000	B471_83FF	Common PERI #0	CRC #0	70
B471_8400	B471_87FF	Common PERI #0	CRC #1	71
B471_8800	B471_8BFF	Common PERI #0	CRC #2	72
B471_8C00	B471_8FFF	Common PERI #0	CRC #3	73
B471_9000	B471_9FFF	Reserved	Reserved	-
B471_A000	B473_7FFF	Reserved	Reserved	-
B473_8000	B473_FFFF	Common PERI #0	GPIO	74
B474_0000	B474_7FFF	Common PERI #0	PPC	75
B474_8000	B474_FFFF	Common PERI #0	RIC	76
B475_0000	B475_7FFF	CPU Config Group	PPU	-
B475_8000	B478_FFFF	Reserved	Reserved	-
B479_0000	B47F_FFFF	Reserved	Reserved	-
B480_0000	B480_03FF	Common PERI #0	M.F.Serial ch.0	176
B480_0400	B480_07FF	Common PERI #0	M.F.Serial ch.1	177
B480_0800	B480_0BFF	Common PERI #0	M.F.Serial ch.2	178
B480_0C00	B480_0FFF	Common PERI #0	M.F.Serial ch.3	179
B480_1000	B480_13FF	Common PERI #0	M.F.Serial ch.4	180
B480_1400	B480_17FF	Common PERI #0	M.F.Serial ch.5	181
B480_1800	B480_1BFF	Common PERI #0	M.F.Serial ch.6	182
B480_1C00	B480_1FFF	Common PERI #0	M.F.Serial ch.7	183
B480_2000	B480_7FFF	Reserved	Reserved	-
B480_8000	B480_83FF	Common PERI #1	Base Timer ch.0	88
B480_8400	B480_87FF	Common PERI #1	Base Timer ch.1	89
B480_8800	B480_8BFF	Common PERI #1	Base Timer ch.2	90
B480_8C00	B480_8FFF	Common PERI #1	Base Timer ch.3	91
B480_9000	B480_93FF	Common PERI #1	Base Timer ch.4	92
B480_9400	B480_97FF	Common PERI #1	Base Timer ch.5	93
B480_9800	B480_9BFF	Common PERI #1	Base Timer ch.6	94
B480_9C00	B480_9FFF	Common PERI #1	Base Timer ch.7	95
B480_A000	B480_A3FF	Common PERI #1	Base Timer ch.8	96
B480_A400	B480_A7FF	Common PERI #1	Base Timer ch.9	97
B480_A800	B480_ABFF	Common PERI #1	Base Timer ch.10	98
B480_AC00	B480_AFFF	Common PERI #1	Base Timer ch.11	99
B480_B000	B480_FFFF	Reserved	Reserved	-
B481_0000	B481_03FF	Common PERI #0	Reload Timer ch.0	128
B481_0400	B481_07FF	Common PERI #0	Reload Timer ch.1	129
B481_0800	B481_0BFF	Common PERI #0	Reload Timer ch.2	130
B481_0C00	B481_0FFF	Common PERI #0	Reload Timer ch.3	131
B481_1000	B481_7FFF	Reserved	Reserved	-
B481_8000	B481_FFFF	Reserved	Reserved	-
B482_0000	B482_03FF	Common PERI #0	FRT ch.0	208

START Address	END Address	Group	Function	PPU_No
B482_0400	B482_07FF	Common PERI #0	FRT ch.1	209
B482_0800	B482_0BFF	Common PERI #0	FRT ch.2	210
B482_0C00	B482_0FFF	Common PERI #0	FRT ch.3	211
B482_1000	B482_13FF	Common PERI #0	FRT ch.4	212
B482_1400	B482_7FFF	Reserved	Reserved	-
B482_8000	B482_83FF	Common PERI #0	Input Capture ch.0 / ch.1	224
B482_8400	B482_87FF	Common PERI #0	Input Capture ch.2 / ch.3	225
B482_8800	B482_8BFF	Common PERI #0	Input Capture ch.4 / ch.5	226
B482_8C00	B482_FFFF	Reserved	Reserved	-
B483_0000	B483_03FF	Common PERI #0	Output Compare ch.0 / ch.1	240
B483_0400	B483_07FF	Common PERI #0	Output Compare ch.2 / ch.3	241
B483_0800	B483_0BFF	Common PERI #0	Output Compare ch.4 / ch.5	242
B483_0C00	B483_FBFF	Reserved	Reserved	-
B483_FC00	B483_FFFF	Common PERI #0	Reload Timer Simultaneous Soft Start for ch.0/1/2/3	80
B484_0000	B484_03FF	Common PERI #0	SG ch.0	266
B484_0400	B484_07FF	Common PERI #0	SG ch.1	267
B484_0800	B484_0BFF	Common PERI #0	SG ch.2	268
B484_0C00	B484_0FFF	Common PERI #0	SG ch.3	269
B484_1000	B484_13FF	Common PERI #0	SG ch.4	270
B484_1400	B484_5FFF	Reserved	Reserved	-
B484_6000	B484_63FF	Common PERI #1	Base Timer ch.24	290
B484_6400	B484_67FF	Common PERI #1	Base Timer ch.25	291
B484_6800	B484_6BFF	Common PERI #1	Base Timer ch.26	292
B484_6C00	B484_6FFF	Common PERI #1	Base Timer ch.27	293
B484_7000	B484_73FF	Common PERI #1	Base Timer ch.28	294
B484_7400	B484_77FF	Common PERI #1	Base Timer ch.29	295
B484_7800	B484_7BFF	Common PERI #1	Base Timer ch.30	296
B484_7C00	B484_7FFF	Common PERI #1	Base Timer ch.31	297
B484_8000	B484_83FF	Reserved	Reserved	-
B484_8400	B484_87FF	Reserved	Reserved	-
B484_8800	B484_8BFF	Reserved	Reserved	-
B484_8C00	B484_8FFF	Reserved	Reserved	-
B484_9000	B484_93FF	Reserved	Reserved	-
B484_9400	B484_97FF	Reserved	Reserved	-
B484_9800	B484_9BFF	Reserved	Reserved	-
B484_9C00	B484_9FFF	Reserved	Reserved	-
B484_A000	B484_A3FF	Reserved	Reserved	-
B484_A400	B484_A7FF	Reserved	Reserved	-
B484_A800	B484_ABFF	Reserved	Reserved	-
B484_AC00	B484_AFFF	Reserved	Reserved	-
B484_B000	B484_B3FF	Reserved	Reserved	-
B484_B400	B484_B7FF	Reserved	Reserved	-
B484_B800	B484_BBFF	Reserved	Reserved	-

START Address	END Address	Group	Function	PPU_No
B484_BC00	B484_BFFF	Reserved	Reserved	-
B484_C000	B484_C3FF	Reserved	Reserved	-
B484_C400	B484_C7FF	Reserved	Reserved	-
B484_C800	B484_CBFF	Reserved	Reserved	-
B484_CC00	B484_CFFF	Reserved	Reserved	-
B484_D000	B484_D3FF	Reserved	Reserved	-
B484_D400	B484_D7FF	Reserved	Reserved	-
B484_D800	B484_DBFF	Reserved	Reserved	-
B484_DC00	B484_DFFF	Reserved	Reserved	-
B484_E000	B484_E3FF	Reserved	Reserved	-
B484_E400	B484_E7FF	Reserved	Reserved	-
B484_E800	B484_EBFF	Reserved	Reserved	-
B484_EC00	B484_EFFF	Reserved	Reserved	-
B484_F000	B484_F3FF	Reserved	Reserved	-
B484_F400	B484_F7FF	Reserved	Reserved	-
B484_F800	B484_FBFF	Reserved	Reserved	-
B484_FC00	B484_FFFF	Reserved	Reserved	-
B485_0000	B485_03FF	Reserved	Reserved	-
B485_0400	B485_07FF	Reserved	Reserved	-
B485_0800	B487_FFFF	Reserved	Reserved	-
B488_0000	B488_03FF	Common PERI #1	M.F.Serial ch.8	184
B488_0400	B488_07FF	Common PERI #1	M.F.Serial ch.9	185
B488_0800	B488_0BFF	Common PERI #1	M.F.Serial ch.10	186
B488_0C00	B488_0FFF	Common PERI #1	M.F.Serial ch.11	187
B488_1000	B488_13FF	Reserved	Reserved	-
B488_1400	B488_17FF	Reserved	Reserved	-
B488_1800	B488_7FFF	Reserved	Reserved	-
B488_8000	B488_83FF	Common PERI #1	Base Timer ch.12	100
B488_8400	B488_87FF	Common PERI #1	Base Timer ch.13	101
B488_8800	B488_8BFF	Common PERI #1	Base Timer ch.14	102
B488_8C00	B488_8FFF	Common PERI #1	Base Timer ch.15	103
B488_9000	B488_93FF	Common PERI #1	Base Timer ch.16	104
B488_9400	B488_97FF	Common PERI #1	Base Timer ch.17	105
B488_9800	B488_9BFF	Common PERI #1	Base Timer ch.18	106
B488_9C00	B488_9FFF	Common PERI #1	Base Timer ch.19	107
B488_A000	B488_A3FF	Common PERI #1	Base Timer ch.20	108
B488_A400	B488_A7FF	Common PERI #1	Base Timer ch.21	109
B488_A800	B488_ABFF	Common PERI #1	Base Timer ch.22	110
B488_AC00	B488_AFFF	Common PERI #1	Base Timer ch.23	111
B488_B000	B488_B3FF	Common PERI #1	Global Timer	334
B488_B400	B488_FFFF	Reserved	Reserved	-
B489_0000	B489_03FF	Common PERI #1	Reload Timer ch.16	144
B489_0400	B489_07FF	Common PERI #1	Reload Timer ch.17	145

START Address	END Address	Group	Function	PPU_No
B489_0800	B489_7FFF	Reserved	Reserved	-
B489_8000	B489_83FF	Common PERI #0	QPRC ch.8	200
B489_8400	B489_87FF	Common PERI #0	QPRC ch.9	201
B489_8800	B489_FFFF	Reserved	Reserved	-
B48A_0000	B48A_03FF	Common PERI #0	FRT ch.8	216
B48A_0400	B48A_07FF	Common PERI #0	FRT ch.9	217
B48A_0800	B48A_0BFF	Common PERI #0	FRT ch.10	218
B48A_0C00	B48A_7FFF	Reserved	Reserved	-
B48A_8000	B48A_83FF	Common PERI #0	Input Capture ch.16 / ch.17	232
B48A_8400	B48A_87FF	Common PERI #0	Input Capture ch.18 / ch.19	233
B48A_8800	B48A_8BFF	Common PERI #0	Input Capture ch.20 / ch.21	234
B48A_8C00	B48A_FFFF	Reserved	Reserved	-
B48B_0000	B48B_03FF	Common PERI #0	Output Compare ch.16 / ch.17	248
B48B_0400	B48B_07FF	Common PERI #0	Output Compare ch.18 / ch.19	249
B48B_0800	B48B_0BFF	Common PERI #0	Output Compare ch.20 / ch.21	250
B48B_0C00	B48B_FBFF	Reserved	Reserved	-
B48B_FC00	B48B_FFFF	Common PERI #1	Reload Timer Simultaneous Soft Start for ch.16/17	81
B48C_0000	B48C_03FF	Reserved	Reserved	-
B48C_0400	B48C_07FF	Common PERI #1	ADC12B0	331
B48C_0800	B48C_0BFF	Common PERI #1	Partial Wake Up	332
B48C_0C00	B48C_3FFF	Reserved	Reserved	-
B48C_4000	B48C_43FF	Common PERI #1	SMC ch.0	346
B48C_4400	B48C_47FF	Common PERI #1	SMC ch.1	347
B48C_4800	B48C_4BFF	Common PERI #1	SMC ch.2	348
B48C_4C00	B48C_4FFF	Common PERI #1	SMC ch.3	349
B48C_5000	B48C_53FF	Common PERI #1	SMC ch.4	350
B48C_5400	B48C_57FF	Common PERI #1	SMC ch.5	351
B48C_5800	B48C_5BFF	Common PERI #1	SMC_Trg_Reg	352
B48C_5C00	B48C_5FFF	Common PERI #1	LCDC	353
B48C_6000	B48C_FFFF	Reserved	Reserved	-
B48D_0000	B48D_FFFF	Reserved	Reserved	-
B48E_0000	B48F_FFFF	Reserved	Reserved	-
B490_0000	B490_FFFF	Common PERI #1	CAN_FD ch.0	256
B491_0000	B491_FFFF	Common PERI #1	CAN_FD ch.1	257
B492_0000	B492_FFFF	Common PERI #1	CAN_FD ch.2	258
B493_0000	B493_FFFF	Common PERI #1	CAN_FD ch.3	259
B494_0000	B494_FFFF	Reserved	Reserved	-
B495_0000	B495_FFFF	Reserved	Reserved	-
B496_0000	B497_FFFF	Reserved	Reserved	-
B498_0000	B4BF_FFFF	Reserved	Reserved	-
B4C0_0000	B4FF_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #1 (Covers B490_0000 -- B497_FFFF)	-
B500_0000	B5FF_FFFF	Reserved	Reserved	-

START Address	END Address	Group	Function	PPU_No
B600_0000	B6FF_FFFF	Reserved	Reserved	-
B700_0000	B70B_FFFF	Bit RMW alias	Bit RMW alias for CPU Config Group (Covers B470_0000 -- B471_7FFF)	-
B70C_0000	B727_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B471_8000 -- B474_FFFF)	-
B728_0000	B747_FFFF	Bit RMW alias	Bit RMW alias for CPU Config Group (Covers B475_0000 -- B478_FFFF)	-
B748_0000	B77F_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B479_0000 -- B47F_FFFF)	-
B780_0000	B783_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B480_0000 -- B480_7FFF)	-
B784_0000	B787_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #1 (Covers B480_8000 -- B480_FFFF)	-
B788_0000	B7A2_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B481_0000 -- B484_5FFF)	-
B7A3_0000	B7A8_1FFF	Bit RMW alias	Bit RMW alias for Common PERI #1 (Covers B484_6000 -- B485_03FF)	-
B7A8_2000	B7BF_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B485_0400 -- B487_FFFF)	-
B7C0_0000	B7CB_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #1 (Covers B488_0000 -- B489_7FFF)	-
B7CC_0000	B7DF_DFFF	Bit RMW alias	Bit RMW alias for Common PERI #0 (Covers B489_8000 -- B48B_FBFF)	-
B7DF_E000	B7FF_FFFF	Bit RMW alias	Bit RMW alias for Common PERI #1 (Covers B48B_FC00 -- B48F_FFFF)	-
B800_0000	B801_7FFF	Reserved	Reserved	-
B801_8000	B801_83FF	Common PERI #1	PRGCRC	356
B801_8400	B801_FFFF	Reserved	Reserved	-
B802_0000	B802_03FF	Common PERI #1	SMIX (Sound System)	361
B802_0400	B802_07FF	Reserved	Reserved	-
B802_0800	B802_0BFF	Common PERI #1	PCM-PWM (Sound System)	-
B802_0C00	B802_0FFF	Common PERI #1	I2S0 (Sound System)	-
B802_1000	B802_13FF	Common PERI #1	I2S1	358
B802_1400	B802_7FFF	Reserved	Reserved	-
B802_8000	B802_83FF	Common PERI #1	Wave Form Generator (Sound System)	357
B802_8400	B802_87FF	Reserved	Reserved	-
B802_8800	B802_FFFF	Reserved	Reserved	-
B803_0000	B803_FFFF	Reserved	Reserved	-
B804_0000	B804_07FF	Common PERI #1	LCD Bus Interface	336
B804_0800	B804_0BFF	Common PERI #1	MPU_AXI (For LCD Bus)	335
B804_0C00	B804_FFFF	Reserved	Reserved	-
B805_0000	B805_03FF	Common PERI #1	Expand PLL	337
B805_0400	BFFF_FFFF	Reserved	Reserved	-
C000_0000	FFFE_DFFF	Reserved	Reserved	-
FFFE_E000	FFFE_E3FF	ERRCFG	IRC0 (IRC0_NMIVASBR)	-

START Address	END Address	Group	Function	PPU_No
FFFE_E400	FFFE_F7FF	Reserved	Reserved	-
FFFE_F800	FFFE_FBFF	ERRCFG	IRC (IRC_NMIVASBR) Mirror *	-
FFFE_FC00	FFFE_FFFF	ERRCFG	BootROM I/F	20

CHAPTER 8: IRQ Map / NMI Map

This chapter explains IRQ MAP and NMI MAP.

1. IRQ MAP
2. NMI MAP

CODE: IRQMAP-JUPI-E1.1

1. IRQ Map

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
0	Reserved		
1	System Control Status	IRC0_IRQPL0: IRQPL1	IRC0_IRQVA1
2	HW-WDT Pre-warning	IRC0_IRQPL0: IRQPL2	IRC0_IRQVA2
3	SW-WDT Pre-warning	IRC0_IRQPL0: IRQPL3	IRC0_IRQVA3
4 to 7	Reserved		
8	TCFLASH Single Bit Error	IRC0_IRQPL2: IRQPL8	IRC0_IRQVA8
9	Reserved		
10	Work FLASH Single Bit Error	IRC0_IRQPL2: IRQPL10	IRC0_IRQVA10
11 to 13	Reserved		
14	System SRAM Single Bit Error (include backup area)	IRC0_IRQPL3: IRQPL14	IRC0_IRQVA14
15	CAN FD RAM (ch.0 to ch.3) Single Bit Error	IRC0_IRQPL3: IRQPL15	IRC0_IRQVA15
16	IRC Vector Address RAM Single Bit Error	IRC0_IRQPL4: IRQPL16	IRC0_IRQVA16
17 to 19	Reserved		
20	Work FLASH RDY Interrupt Request / Work FLASH Write Completion	IRC0_IRQPL5: IRQPL20	IRC0_IRQVA20
21 to 23	Reserved		
24	External Interrupt Request ch.0 / ch.16	IRC0_IRQPL6: IRQPL24	IRC0_IRQVA24
25	External Interrupt Request ch.1 / ch.17	IRC0_IRQPL6: IRQPL25	IRC0_IRQVA25
26	External Interrupt Request ch.2 / ch.18	IRC0_IRQPL6: IRQPL26	IRC0_IRQVA26
27	External Interrupt Request ch.3 / ch.19	IRC0_IRQPL6: IRQPL27	IRC0_IRQVA27
28	External Interrupt Request ch.4 / ch.20	IRC0_IRQPL7: IRQPL28	IRC0_IRQVA28
29	External Interrupt Request ch.5 / ch.21	IRC0_IRQPL7: IRQPL29	IRC0_IRQVA29
30	External Interrupt Request ch.6 / ch.22	IRC0_IRQPL7: IRQPL30	IRC0_IRQVA30
31	External Interrupt Request ch.7 / ch.23	IRC0_IRQPL7: IRQPL31	IRC0_IRQVA31
32	External Interrupt Request ch.8	IRC0_IRQPL8: IRQPL32	IRC0_IRQVA32
33	External Interrupt Request ch.9	IRC0_IRQPL8: IRQPL33	IRC0_IRQVA33
34	External Interrupt Request ch.10	IRC0_IRQPL8: IRQPL34	IRC0_IRQVA34
35	External Interrupt Request ch.11	IRC0_IRQPL8: IRQPL35	IRC0_IRQVA35
36	External Interrupt Request ch.12	IRC0_IRQPL9: IRQPL36	IRC0_IRQVA36
37	External Interrupt Request ch.13	IRC0_IRQPL9: IRQPL37	IRC0_IRQVA37
38	External Interrupt Request ch.14	IRC0_IRQPL9: IRQPL38	IRC0_IRQVA38
39	External Interrupt Request ch.15	IRC0_IRQPL9: IRQPL39	IRC0_IRQVA39
40	CAN FD ch.0 (OR-ed of all factors)	IRC0_IRQPL10: IRQPL40	IRC0_IRQVA40
41	CAN FD ch.1 (OR-ed of all factors)	IRC0_IRQPL10: IRQPL41	IRC0_IRQVA41
42	CAN FD ch.2 (OR-ed of all factors)	IRC0_IRQPL10: IRQPL42	IRC0_IRQVA42
43	CAN FD ch.3 (OR-ed of all factors)	IRC0_IRQPL10: IRQPL43	IRC0_IRQVA43
44	Reserved	IRC0_IRQPL11: IRQPL44	IRC0_IRQVA44
45	Reserved	IRC0_IRQPL11: IRQPL45	IRC0_IRQVA45
46	MFS RX ch.0	IRC0_IRQPL11: IRQPL46	IRC0_IRQVA46

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
47	MFS TX ch.0	IRC0_IRQPL11: IRQPL47	IRC0_IRQVA47
48	MFS RX ch.1	IRC0_IRQPL12: IRQPL48	IRC0_IRQVA48
49	MFS TX ch.1	IRC0_IRQPL12: IRQPL49	IRC0_IRQVA49
50	MFS RX ch.2	IRC0_IRQPL12: IRQPL50	IRC0_IRQVA50
51	MFS TX ch.2	IRC0_IRQPL12: IRQPL51	IRC0_IRQVA51
52	MFS RX ch.3	IRC0_IRQPL13: IRQPL52	IRC0_IRQVA52
53	MFS TX ch.3	IRC0_IRQPL13: IRQPL53	IRC0_IRQVA53
54	MFS RX ch.4	IRC0_IRQPL13: IRQPL54	IRC0_IRQVA54
55	MFS TX ch.4	IRC0_IRQPL13: IRQPL55	IRC0_IRQVA55
56	MFS RX ch.5	IRC0_IRQPL14: IRQPL56	IRC0_IRQVA56
57	MFS TX ch.5	IRC0_IRQPL14: IRQPL57	IRC0_IRQVA57
58	MFS RX ch.6	IRC0_IRQPL14: IRQPL58	IRC0_IRQVA58
59	MFS TX ch.6	IRC0_IRQPL14: IRQPL59	IRC0_IRQVA59
60	MFS RX ch.7	IRC0_IRQPL15: IRQPL60	IRC0_IRQVA60
61	MFS TX ch.7	IRC0_IRQPL15: IRQPL61	IRC0_IRQVA61
62	MFS RX ch.8	IRC0_IRQPL15: IRQPL62	IRC0_IRQVA62
63	MFS TX ch.8	IRC0_IRQPL15: IRQPL63	IRC0_IRQVA63
64	MFS RX ch.9	IRC0_IRQPL16: IRQPL64	IRC0_IRQVA64
65	MFS TX ch.9	IRC0_IRQPL16: IRQPL65	IRC0_IRQVA65
66	MFS RX ch.10	IRC0_IRQPL16: IRQPL66	IRC0_IRQVA66
67	MFS TX ch.10	IRC0_IRQPL16: IRQPL67	IRC0_IRQVA67
68	MFS RX ch.11	IRC0_IRQPL17: IRQPL68	IRC0_IRQVA68
69	MFS TX ch.11	IRC0_IRQPL17: IRQPL69	IRC0_IRQVA69
70 to 73	Reserved		
74	SMC ch.0	IRC0_IRQPL18: IRQPL74	IRC0_IRQVA74
75	SMC ch.1	IRC0_IRQPL18: IRQPL75	IRC0_IRQVA75
76	SMC ch.2	IRC0_IRQPL19: IRQPL76	IRC0_IRQVA76
77	SMC ch.3	IRC0_IRQPL19: IRQPL77	IRC0_IRQVA77
78	SMC ch.4	IRC0_IRQPL19: IRQPL78	IRC0_IRQVA78
79	SMC.ch.5	IRC0_IRQPL19: IRQPL79	IRC0_IRQVA79
80	SG ch.0	IRC0_IRQPL20: IRQPL80	IRC0_IRQVA80
81	SG ch.1	IRC0_IRQPL20: IRQPL81	IRC0_IRQVA81
82	SG ch.2	IRC0_IRQPL20: IRQPL82	IRC0_IRQVA82
83	SG ch.3	IRC0_IRQPL20: IRQPL83	IRC0_IRQVA83
84	SG ch.4	IRC0_IRQPL21: IRQPL84	IRC0_IRQVA84
85	LCD Bus Interface Sequencer Sync Interrupt / Sequencer Error Interrupt / Tearing Effect Interrupt	IRC0_IRQPL21: IRQPL85	IRC0_IRQVA85
86	LCD Bus interface InstrFifointerrupt	IRC0_IRQPL21: IRQPL86	IRC0_IRQVA86
87	LCD Bus interface RxFifoInterrupt	IRC0_IRQPL21: IRQPL87	IRC0_IRQVA87
88	LCD Bus interface ReadChannelDone	IRC0_IRQPL22: IRQPL88	IRC0_IRQVA88

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
89	LCD Bus interface WriteChannelDone	IRC0_IRQPL22: IRQPL89	IRC0_IRQVA89
90	Indicator PWM	IRC0_IRQPL22: IRQPL90	IRC0_IRQVA90
91	SHE Error	RC0_IRQPL22: IRQPL91	IRC0_IRQVA91
92	SHE	RC0_IRQPL23: IRQPL92	IRC0_IRQVA92
93	DDR HSSPI RX	RC0_IRQPL23: IRQPL93	IRC0_IRQVA93
94	DDR HSSPI TX	RC0_IRQPL23: IRQPL94	IRC0_IRQVA94
95	TCRAM diag	RC0_IRQPL23: IRQPL95	IRC0_IRQVA95
96	Reserved		
97	Global Timer (Compare Clear Interrupt)	RC0_IRQPL24: IRQPL97	IRC0_IRQVA97
98	RTC	RC0_IRQPL24: IRQPL98	IRC0_IRQVA98
99	CR CARIBRATION	RC0_IRQPL24: IRQPL99	IRC0_IRQVA99
100	Base Timer ch.0/8/9/10/11	IRC0_IRQPL25: IRQPL100	IRC0_IRQVA100
101	Base Timer ch.1	IRC0_IRQPL25: IRQPL101	IRC0_IRQVA101
102	Base Timer ch.2	IRC0_IRQPL25: IRQPL102	IRC0_IRQVA102
103	Base Timer ch.3	IRC0_IRQPL25: IRQPL103	IRC0_IRQVA103
104	Base Timer ch.4	IRC0_IRQPL26: IRQPL104	IRC0_IRQVA104
105	Base Timer ch.5	IRC0_IRQPL26: IRQPL105	IRC0_IRQVA105
106	Base Timer ch.6	IRC0_IRQPL26: IRQPL106	IRC0_IRQVA106
107	Base Timer ch.7	IRC0_IRQPL26: IRQPL107	IRC0_IRQVA107
108	Base Timer ch.12/20/21/22/23	IRC0_IRQPL27: IRQPL108	IRC0_IRQVA108
109	Base Timer ch.13	IRC0_IRQPL27: IRQPL109	IRC0_IRQVA109
110	Base Timer ch.14	IRC0_IRQPL27: IRQPL110	IRC0_IRQVA110
111	Base Timer ch.15	IRC0_IRQPL27: IRQPL111	IRC0_IRQVA111
112	Base Timer ch.16	IRC0_IRQPL28: IRQPL112	IRC0_IRQVA112
113	Base Timer ch.17	IRC0_IRQPL28: IRQPL113	IRC0_IRQVA113
114	Base Timer ch.18	IRC0_IRQPL28: IRQPL114	IRC0_IRQVA114
115	Base Timer ch.19	IRC0_IRQPL28: IRQPL115	IRC0_IRQVA115
116	Base Timer ch.24	IRC0_IRQPL29: IRQPL116	IRC0_IRQVA116
117	Base Timer ch.25	IRC0_IRQPL29: IRQPL117	IRC0_IRQVA117
118	Base Timer ch.26	IRC0_IRQPL29: IRQPL118	IRC0_IRQVA118
119	Base Timer ch.27	IRC0_IRQPL29: IRQPL119	IRC0_IRQVA119
120	Base Timer ch.28	IRC0_IRQPL30: IRQPL120	IRC0_IRQVA120
121	Base Timer ch.29	IRC0_IRQPL30: IRQPL121	IRC0_IRQVA121
122	Base Timer ch.30	IRC0_IRQPL30: IRQPL122	IRC0_IRQVA122
123	Base Timer ch.31	IRC0_IRQPL30: IRQPL123	IRC0_IRQVA123
124 to 143	Reserved		
144	Reload Timer ch.0	IRC0_IRQPL36: IRQPL144	IRC0_IRQVA144
145	Reload Timer ch.1	IRC0_IRQPL36: IRQPL145	IRC0_IRQVA145
146	Reload Timer ch.2	IRC0_IRQPL36: IRQPL146	IRC0_IRQVA146
147	Reload Timer ch.3	IRC0_IRQPL36: IRQPL147	IRC0_IRQVA147
148	Reload Timer ch.16	IRC0_IRQPL37: IRQPL148	IRC0_IRQVA148

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
149	Reload Timer ch.17	IRC0_IRQPL37: IRQPL149	IRC0_IRQVA149
150	FRT ch.0	IRC0_IRQPL37: IRQPL150	IRC0_IRQVA150
151	FRT ch.1	IRC0_IRQPL37: IRQPL151	IRC0_IRQVA151
152	FRT ch.2	IRC0_IRQPL38: IRQPL152	IRC0_IRQVA152
153	FRT ch.3	IRC0_IRQPL38: IRQPL153	IRC0_IRQVA153
154	FRT ch.4	IRC0_IRQPL38: IRQPL154	IRC0_IRQVA154
155	Reserved		
156	FRT ch.8	IRC0_IRQPL39: IRQPL156	IRC0_IRQVA156
157	FRT ch.9	IRC0_IRQPL39: IRQPL157	IRC0_IRQVA157
158	FRT ch.10	IRC0_IRQPL39: IRQPL158	IRC0_IRQVA158
159	PCMPWM_OVFL / PCMPWM_UDRN, / PCMPWM_DMAE	IRC0_IRQPL39: IRQPL159	IRC0_IRQVA159
160	PCMPWM_DREQ	IRC0_IRQPL40: IRQPL160	IRC0_IRQVA160
161	Reserved		
162	IRQ0 of Input Capture 0 (ch.0)	IRC0_IRQPL40: IRQPL162	IRC0_IRQVA162
163	IRQ0 of Input Capture 1 (ch.2)	IRC0_IRQPL40: IRQPL163	IRC0_IRQVA163
164	IRQ0 of Input Capture 2 (ch.4)	IRC0_IRQPL41: IRQPL164	IRC0_IRQVA164
165	IRQ0 of Input Capture 8 (ch.16)	IRC0_IRQPL41: IRQPL165	IRC0_IRQVA165
166	IRQ0 of Input Capture 9 (ch.18)	IRC0_IRQPL41: IRQPL166	IRC0_IRQVA166
167	IRQ0 of Input Capture 10 (ch.20)	IRC0_IRQPL41: IRQPL167	IRC0_IRQVA167
168	IRQ1 of Input Capture 0 (ch.1)	IRC0_IRQPL42: IRQPL168	IRC0_IRQVA168
169	IRQ1 of Input Capture 1 (ch.3)	IRC0_IRQPL42: IRQPL169	IRC0_IRQVA169
170	IRQ1 of Input Capture 2 (ch.5)	IRC0_IRQPL42: IRQPL170	IRC0_IRQVA170
171	IRQ1 of Input Capture 8 (ch.17)	IRC0_IRQPL42: IRQPL171	IRC0_IRQVA171
172	IRQ1 of Input Capture 9 (ch.19)	IRC0_IRQPL43: IRQPL172	IRC0_IRQVA172
173	IRQ1 of Input Capture 10 (ch.21)	IRC0_IRQPL43: IRQPL173	IRC0_IRQVA173
174	IRQ0 of Output Compare 0 (ch.0)	IRC0_IRQPL43: IRQPL174	IRC0_IRQVA174
175	IRQ0 of Output Compare 1 (ch.2)	IRC0_IRQPL43: IRQPL175	IRC0_IRQVA175
176	IRQ0 of Output Compare 2 (ch.4)	IRC0_IRQPL44: IRQPL176	IRC0_IRQVA176
177	Reserved		
178	IRQ0 of Output Compare 8 (ch.16)	IRC0_IRQPL44: IRQPL178	IRC0_IRQVA178
179	IRQ0 of Output Compare 9 (ch.18)	IRC0_IRQPL44: IRQPL179	IRC0_IRQVA179
180	IRQ0 of Output Compare 10 (ch.20)	IRC0_IRQPL45: IRQPL180	IRC0_IRQVA180
181	IRQ1 of Output Compare 0 (ch.1)	IRC0_IRQPL45: IRQPL181	IRC0_IRQVA181
182	IRQ1 of Output Compare 1 (ch.3)	IRC0_IRQPL45: IRQPL182	IRC0_IRQVA182
183	IRQ1 of Output Compare 2 (ch.5)	IRC0_IRQPL45: IRQPL183	IRC0_IRQVA183
184	Reserved		
185	IRQ1 of Output Compare 8 (ch.17)	IRC0_IRQPL46: IRQPL185	IRC0_IRQVA185
186	IRQ1 of Output Compare 9 (ch.19)	IRC0_IRQPL46: IRQPL186	IRC0_IRQVA186
187	IRQ1 of Output Compare 10 (ch.21)	IRC0_IRQPL46: IRQPL187	IRC0_IRQVA187

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
188	QPRC ch.8	IRC0_IRQPL47: IRQPL188	IRC0_IRQVA188
189	QPRC ch.9	IRC0_IRQPL47: IRQPL189	IRC0_IRQVA189
190 to 193	Reserved		
194	ADC12B0 Conversion Done	IRC0_IRQPL48: IRQPL194	IRC0_IRQVA194
195	ADC12B0 Group interrupt	IRC0_IRQPL48: IRQPL195	IRC0_IRQVA195
196	ADC12B0 pulse detection function	IRC0_IRQPL49: IRQPL196	IRC0_IRQVA196
197	ADC12B0 RCO	IRC0_IRQPL49: IRQPL197	IRC0_IRQVA197
198	DMA Error	IRC0_IRQPL49: IRQPL198	IRC0_IRQVA198
199	DMAC Completion ch.0	IRC0_IRQPL49: IRQPL199	IRC0_IRQVA199
200	DMAC Completion ch.1	IRC0_IRQPL50: IRQPL200	IRC0_IRQVA200
201	DMAC Completion ch.2	IRC0_IRQPL50: IRQPL201	IRC0_IRQVA201
202	DMAC Completion ch.3	IRC0_IRQPL50: IRQPL202	IRC0_IRQVA202
203	DMAC Completion ch.4	IRC0_IRQPL50: IRQPL203	IRC0_IRQVA203
204	DMAC Completion ch.5	IRC0_IRQPL51: IRQPL204	IRC0_IRQVA204
205	DMAC Completion ch.6	IRC0_IRQPL51: IRQPL205	IRC0_IRQVA205
206	DMAC Completion ch.7	IRC0_IRQPL51: IRQPL206	IRC0_IRQVA206
207	DMAC Completion ch.8	IRC0_IRQPL51: IRQPL207	IRC0_IRQVA207
208	DMAC Completion ch.9	IRC0_IRQPL52: IRQPL208	IRC0_IRQVA208
209	DMAC Completion ch.10	IRC0_IRQPL52: IRQPL209	IRC0_IRQVA209
210	DMAC Completion ch.11	IRC0_IRQPL52: IRQPL210	IRC0_IRQVA210
211	DMAC Completion ch.12	IRC0_IRQPL52: IRQPL211	IRC0_IRQVA211
212	DMAC Completion ch.13	IRC0_IRQPL53: IRQPL212	IRC0_IRQVA212
213	DMAC Completion ch.14	IRC0_IRQPL53: IRQPL213	IRC0_IRQVA213
214	DMAC Completion ch.15	IRC0_IRQPL53: IRQPL214	IRC0_IRQVA214
215	DMAC RLT (ch.0,1,2,3 OR-ed)	IRC0_IRQPL53: IRQPL215	IRC0_IRQVA215
216	SCT RC IRQ	IRC0_IRQPL54: IRQPL216	IRC0_IRQVA216
217	SCT SRC IRQ	IRC0_IRQPL54: IRQPL217	IRC0_IRQVA217
218	SCT Main OSC IRQ	IRC0_IRQPL54: IRQPL218	IRC0_IRQVA218
219	SCT Sub OSC IRQ	IRC0_IRQPL54: IRQPL219	IRC0_IRQVA219
220	CR5 Performance Monitor Unit IRQ	IRC0_IRQPL55: IRQPL220	IRC0_IRQVA220
221	PRGCRC	IRC0_IRQPL55: IRQPL221	IRC0_IRQVA221
222	MFS ch.0 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL55: IRQPL222	IRC0_IRQVA222
223	MFS ch.1 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL55: IRQPL223	IRC0_IRQVA223
224	MFS ch.2 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL56: IRQPL224	IRC0_IRQVA224
225	MFS ch.3 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL56: IRQPL225	IRC0_IRQVA225
226	MFS ch.4 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL56: IRQPL226	IRC0_IRQVA226
227	MFS ch.5 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL56: IRQPL227	IRC0_IRQVA227
228	MFS ch.6 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL57: IRQPL228	IRC0_IRQVA228
229	MFS ch.7 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL57: IRQPL229	IRC0_IRQVA229
230	MFS ch.8 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL57: IRQPL230	IRC0_IRQVA230
231	MFS ch.9 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL57: IRQPL231	IRC0_IRQVA231

IRQ No.	Detail	IRQ Priority Register	Vector Address Register
232	MFS ch.10 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL58: IRQPL232	IRC0_IRQVA232
233	MFS ch.11 Error (Tx/Rx error, Status OR-ed)	IRC0_IRQPL58: IRQPL233	IRC0_IRQVA233
234	Reserved	IRC0_IRQPL58: IRQPL234	IRC0_IRQVA234
235	Reserved	IRC0_IRQPL58: IRQPL235	IRC0_IRQVA235
236	I2S0_IRQ	IRC0_IRQPL59: IRQPL236	IRC0_IRQVA236
237	I2S1_IRQ	IRC0_IRQPL59: IRQPL237	IRC0_IRQVA237
238	WG_AHB_ERR_IRQ	IRC0_IRQPL59: IRQPL238	IRC0_IRQVA238
239	WG_END_IRQ0	IRC0_IRQPL59: IRQPL239	IRC0_IRQVA239
240	WG_END_IRQ1	IRC0_IRQPL60: IRQPL240	IRC0_IRQVA240
241	WG_END_IRQ2	IRC0_IRQPL60: IRQPL241	IRC0_IRQVA241
242	WG_END_IRQ3	IRC0_IRQPL60: IRQPL242	IRC0_IRQVA242
243	WG_END_IRQ4	IRC0_IRQPL60: IRQPL243	IRC0_IRQVA243
244	MX_AHB_ERR_IRQ	IRC0_IRQPL61: IRQPL244	IRC0_IRQVA244
245	MX_OVFL_IRQ0 / MX_DMA_ERR_IRQ0	IRC0_IRQPL61: IRQPL245	IRC0_IRQVA245
246	MX_OVFL_IRQ1 / MX_DMA_ERR_IRQ1	IRC0_IRQPL61: IRQPL246	IRC0_IRQVA246
247	MX_OVFL_IRQ2 / MX_DMA_ERR_IRQ2	IRC0_IRQPL61: IRQPL247	IRC0_IRQVA247
248	MX_OVFL_IRQ3 / MX_DMA_ERR_IRQ3	IRC0_IRQPL62: IRQPL248	IRC0_IRQVA248
249	MX_OVFL_IRQ4 / MX_DMA_ERR_IRQ4	IRC0_IRQPL62: IRQPL249	IRC0_IRQVA249
250	MX_DATA_REQ_IRQ0	IRC0_IRQPL62: IRQPL250	IRC0_IRQVA250
251	MX_DATA_REQ_IRQ1	IRC0_IRQPL62: IRQPL251	IRC0_IRQVA251
252	MX_DATA_REQ_IRQ2	IRC0_IRQPL63: IRQPL252	IRC0_IRQVA252
253	MX_DATA_REQ_IRQ3	IRC0_IRQPL63: IRQPL253	IRC0_IRQVA253
254	MX_DATA_REQ_IRQ4	IRC0_IRQPL63: IRQPL254	IRC0_IRQVA254
255	Reserved		

2. NMI Map

NMI No.	Detail	Priority Level	IRQ/NMI Vector Address
0	NMIX pin(Ext-IRC)	IRC0_NMIPL0: NMIPL0	IRC0_NMIVA0
1 to 3	Reserved	-	-
4	LVDs IRQ	IRC0_NMIPL1: NMIPL4	IRC0_NMIVA4
5	CSV, Profile	IRC0_NMIPL1: NMIPL5	IRC0_NMIVA5
6	HW-WDT	IRC0_NMIPL1: NMIPL6	IRC0_NMIVA6
7	SW-WDT	IRC0_NMIPL1: NMIPL7	IRC0_NMIVA7
8	IRC 2-bit ECC err detection	IRC0_NMIPL2: NMIPL8	IRC0_NMIVA8
9	Expand PLL CSV(OR-ed of all factors)	IRC0_NMIPL2: NMIPL9	IRC0_NMIVA9
10 to 11	Reserved	-	-
12	CAN-FD RAMs 2-bit ECC error detection (OR-ed of all factors)	IRC0_NMIPL3: NMIPL12	IRC0_NMIVA12
13	DMAC MPU #0 protection violation	IRC0_NMIPL3: NMIPL13	IRC0_NMIVA13
14	Reserved	-	-
15	SHE MPU	IRC0_NMIPL3: NMIPL15	IRC0_NMIVA15
16 to 17	Reserved	-	-
18	TPU protection violation	IRC0_NMIPL4: NMIPL18	IRC0_NMIVA18
19	Reserved	-	-
20	LCDBUS MPU	IRC0_NMIPL5: NMIPL20	IRC0_NMIVA20
21 to 31	Reserved	-	-

CHAPTER 9: DMA Channel Activation Factors

This chapter explains the DMA channel activation factors.

1.Factors list

CODE: DMAFACT-JUPI-E1

1. Factors List

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
0 to 8	Reserved						-	
9	WORK FLASH	-	-	-	-	-	Unnecessary	*2
10	Ext IRQ 0	Ext IRQ 8	Ext IRQ 16	-	-	-	Unnecessary	*1
11	Ext IRQ 1	Ext IRQ 9	Ext IRQ 17	-	-	-	Unnecessary	*1
12	Ext IRQ 2	Ext IRQ 10	Ext IRQ 18	-	-	-	Unnecessary	*1
13	Ext IRQ 3	Ext IRQ 11	Ext IRQ 19	-	-	-	Unnecessary	*1
14	Ext IRQ 4	Ext IRQ 12	Ext IRQ 20	-	-	-	Unnecessary	*1
15	Ext IRQ 5	Ext IRQ 13	Ext IRQ 21	-	-	-	Unnecessary	*1
16	Ext IRQ 6	Ext IRQ 14	Ext IRQ 22	-	-	-	Unnecessary	*1
17	Ext IRQ 7	Ext IRQ 15	Ext IRQ 23	-	-	-	Unnecessary	*1
18	MFS ch.0 RX	-	-	-	-	-	Necessary	*4
19	MFS ch.0 TX	-	-	-	-	-	Necessary	*4
20	MFS ch.1 RX	-	-	-	-	-	Necessary	*4
21	MFS ch.1 TX	-	-	-	-	-	Necessary	*4
22	MFS ch.2 RX	-	-	-	-	-	Necessary	*4
23	MFS ch.2 TX	-	-	-	-	-	Necessary	*4
24	MFS ch.3 RX	-	-	-	-	-	Necessary	*4
25	MFS ch.3 TX	-	-	-	-	-	Necessary	*4
26	MFS ch.4 RX	-	-	-	-	-	Necessary	*4
27	MFS ch.4 TX	-	-	-	-	-	Necessary	*4
28	MFS ch.5 RX	-	-	-	-	-	Necessary	*4
29	MFS ch.5 TX	-	-	-	-	-	Necessary	*4
30	MFS ch.6 RX	-	-	-	-	-	Necessary	*4
31	MFS ch.6 TX	-	-	-	-	-	Necessary	*4

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
32	MFS ch.7 RX	-	-	-	-	-	Necessary	*4
33	MFS ch.7 TX	-	-	-	-	-	Necessary	*4
34	MFS ch.8 RX	-	-	-	-	-	Necessary	*4
35	MFS ch.8 TX	-	-	-	-	-	Necessary	*4
36	MFS ch.9 RX	-	-	-	-	-	Necessary	*4
37	MFS ch.9 TX	-	-	-	-	-	Necessary	*4
38	MFS ch.10 RX	-	-	-	-	-	Necessary	*4
39	MFS ch.10 TX	-	-	-	-	-	Necessary	*4
40	MFS ch.11 RX	-	-	-	-	-	Necessary	*4
41	MFS ch.11 TX	-	-	-	-	-	Necessary	*4
42 to 55	Reserved						-	
56	Base Timer ch.0-0	Base Timer ch.0-1	Base Timer ch.24-0	Base Timer ch.24-1	-	-	Necessary	*1, *3
57	Base Timer ch.1-0	Base Timer ch.1-1	Base Timer ch.25-0	Base Timer ch.25-1	-	-	Necessary	*1, *3
58	Base Timer ch.2-0	Base Timer ch.2-1	Base Timer ch.26-0	Base Timer ch.26-1	-	-	Necessary	*1, *3
59	Base Timer ch.3-0	Base Timer ch.3-1	Base Timer ch.27-0	Base Timer ch.27-1	-	-	Necessary	*1, *3
60	Base Timer ch.4-0	Base Timer ch.4-1	Base Timer ch.28-0	Base Timer ch.28-1	-	-	Necessary	*1, *3
61	Base Timer ch.5-0	Base Timer ch.5-1	Base Timer ch.29-0	Base Timer ch.29-1	-	-	Necessary	*1, *3
62	Base Timer ch.6-0	Base Timer ch.6-1	Base Timer ch.30-0	Base Timer ch.30-1	-	-	Necessary	*1, *3

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
63	Base Timer ch.7-0	Base Timer ch.7-1	Base Timer ch.31-0	Base Timer ch.31-1	-	-	Necessary	*1, *3
64	Base Timer ch.8-0	Base Timer ch.8-1	-	-	-	-	Necessary	*1, *3
65	Base Timer ch.9-0	Base Timer ch.9-1	-	-	-	-	Necessary	*1, *3
66	Base Timer ch.10-0	Base Timer ch.10-1	-	-	-	-	Necessary	*1, *3
67	Base Timer ch.11-0	Base Timer ch.11-1	-	-	-	-	Necessary	*1, *3
68	Base Timer ch.12-0	Base Timer ch.12-1	-	-	-	-	Necessary	*1, *3
69	Base Timer ch.13-0	Base Timer ch.13-1	-	-	-	-	Necessary	*1, *3
70	Base Timer ch.14-0	Base Timer ch.14-1	-	-	-	-	Necessary	*1, *3
71	Base Timer ch.15-0	Base Timer ch.15-1	-	-	-	-	Necessary	*1, *3
72	Base Timer ch.16-0	Base Timer ch.16-1	-	-	-	-	Necessary	*1, *3
73	Base Timer ch.17-0	Base Timer ch.17-1	-	-	-	-	Necessary	*1, *3
74	Base Timer ch.18-0	Base Timer ch.18-1	-	-	-	-	Necessary	*1, *3
75	Base Timer ch.19-0	Base Timer ch.19-1	-	-	-	-	Necessary	*1, *3
76	Base Timer ch.20-0	Base Timer ch.20-1	-	-	-	-	Necessary	*1, *3

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
77	Base Timer ch.21-0	Base Timer ch.21-1	-	-	-	-	Necessary	*1, *3
78	Base Timer ch.22-0	Base Timer ch.22-1	-	-	-	-	Necessary	*1, *3
79	Base Timer ch.23-0	Base Timer ch.23-1	-	-	-	-	Necessary	*1, *3
80	FRT ch.0 Match	FRT ch.0 Zero	FRT ch.2 Match	FRT ch.2 Zero	FRT ch.4 Match	FRT ch.4 Zero	Necessary	*1, *3
81	FRT ch.1 Match	FRT ch.1 Zero	FRT ch.3 Match	FRT ch.3 Zero	-	-	Necessary	*1, *3
82	FRT ch.8 Match	FRT ch.8 Zero	FRT ch.10 Match	FRT ch.10 Zero	-	-	Necessary	*1, *3
83	FRT ch.9 Match	FRT ch.9 Zero	-	-	-	-	Necessary	*1, *3
84	IRQ0 of ICU pair0 (ch.0)	IRQ0 of ICU pair1 (ch.2)	IRQ0 of ICU pair2 (ch.4)	-	-	-	Necessary	*1, *3
85	IRQ0 of ICU pair8 (ch.16)	IRQ0 of ICU pair9 (ch.18)	IRQ0 of ICU pair10 (ch.20)	-	-	-	Necessary	*1, *3
86	IRQ1 of ICU pair0 (ch.1)	IRQ1 of ICU pair1 (ch.3)	IRQ1 of ICU pair2 (ch.5)	-	-	-	Necessary	*1, *3
87	IRQ1 of ICU pair8 (ch.17)	IRQ1 of ICU pair9 (ch.19)	IRQ1 of ICU pair10 (ch.21)	-	-	-	Necessary	*1, *3
88	IRQ0 of OCU pair0 (ch.0)	IRQ0 of OCU pair1 (ch.2)	IRQ0 of OCU pair2 (ch.4)	-	-	-	Necessary	*1, *3
89	IRQ0 of OCU pair8 (ch.16)	IRQ0 of OCU pair9 (ch.18)	IRQ0 of OCU pair10 (ch.20)	-	-	-	Necessary	*1, *3

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
90	IRQ1 of OCU pair0 (ch.1)	IRQ1 of OCU pair1 (ch.3)	IRQ1 of OCU pair2 (ch.5)	-	-	-	Necessary	*1, *3
91	IRQ1 of OCU pair8 (ch.17)	IRQ1 of OCU pair9 (ch.19)	IRQ1 of OCU pair10 (ch.21)	-	-	-	Necessary	*1, *3
92	Reload Timer ch.0	Reload Timer ch.2	Reload Timer ch.16	-	-	-	Unnecessary	*1
93	Reload Timer ch.1	Reload Time ch.3	Reload Timer ch.17	-	-	-	Unnecessary	*1
94	DMAC #0 Reload Timer 0	DMAC #0 Reload Timer 1	DMAC #0 Reload Timer 2	DMAC #0 Reload Timer 3	-	-	Unnecessary	*1
95 to 98	Reserved						-	
99	ADC12B1-0	-	-	-	-	-	Unnecessary	
100	ADC12B1-1	-	-	-	-	-	Unnecessary	
101	ADC12B1-2	-	-	-	-	-	Unnecessary	
102	ADC12B1-3	-	-	-	-	-	Unnecessary	
103	PRGCRC	-	-	-	-	-	Unnecessary	
104 to 105	Reserved						-	
106	DDRHSSPI RX	-	-	-	-	-	Unnecessary	
107	DDRHSSPI TX	-	-	-	-	-	Unnecessary	
108	Sound Generator 0	-	-	-	-	-	Necessary	*3
109	Sound Generator 1	-	-	-	-	-	Necessary	*3
110	Sound Generator 2	-	-	-	-	-	Necessary	*3
111	Sound Generator 3	-	-	-	-	-	Necessary	*3
112	Sound Generator 4	-	-	-	-	-	Necessary	*3
113	Reserved						-	

Client Number	Peripheral Functions Combination						Enable Setting of IRQ Request	Remarks
	0	1	2	3	4	5		
114	Mixer [0]	-	-	-	-	-	Unnecessary	
115	Mixer [1]	-	-	-	-	-	Unnecessary	
116	Mixer [2]	-	-	-	-	-	Unnecessary	
117	Mixer [3]	-	-	-	-	-	Unnecessary	
118	Mixer [4]	-	-	-	-	-	Unnecessary	
119	WFG [0]	-	-	-	-	-	Unnecessary	
120	WFG [1]	-	-	-	-	-	Unnecessary	
121	WFG [2]	-	-	-	-	-	Unnecessary	
122	WFG [3]	-	-	-	-	-	Unnecessary	
123	WFG [4]	-	-	-	-	-	Unnecessary	
124	I2S ch.1 RX	-	-	-	-	-	Unnecessary	
125	I2S ch.1 TX	-	-	-	-	-	Unnecessary	
126 to 128	Reserved						-	

*1: For assignment of source, see "RESOURCE INPUT SELECTION"

*2: DSTP_ACK function is set by DMAi_CMCICm:BEHSTPACK.

*3: The interrupt factor flags of peripheral function are cleared automatically by acceptance of DMA transfer request

*4: The interrupt factor flags of peripheral function are cleared automatically after DMA transfer is started.

CHAPTER 10: Port Description

This chapter explains port functions.

1. Port description list
2. Remark

CODE: PORT_DESCRIPTION-JUPI-E2

1. Port Description List

The table shows the port function of description which is supported. The port function which is not described in the table is not supported for the product.

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MODE	Mode pin	29	34	40	48	
RSTX	External reset input pin	30	35	41	49	
NMIX	Non-maskable interrupt input pin	39	48	55	69	
X0	Main clock oscillation input pin	28	33	39	47	
X1	Main clock oscillation output pin	27	32	38	46	
X0A	Sub clock oscillation input pin	40	49	56	70	
X1A	Sub clock oscillation output pin	41	50	57	71	
EINT0_0	External interrupt ch.0 input pin (0)	97	112	130	160	
EINT0_1	External interrupt ch.0 input pin (1)	-	117	137	167	
EINT0_2	External interrupt ch.0 input pin (2)	-	-	3	3	
EINT0_4	External interrupt ch.0 input pin (4)	-	-	-	90	
EINT0_5	External interrupt ch.0 input pin (5)	-	84	97	121	
EINT0_6	External interrupt ch.0 input pin (6)	-	-	131	161	
EINT1_0	External interrupt ch.1 input pin (0)	-	119	139	169	
EINT1_1	External interrupt ch.1 input pin (1)	-	-	-	9	
EINT1_2	External interrupt ch.1 input pin (2)	-	-	4	4	
EINT1_3	External interrupt ch.1 input pin (3)	13	13	19	27	
EINT1_4	External interrupt ch.1 input pin (4)	-	-	-	82	
EINT1_6	External interrupt ch.1 input pin (6)	-	-	132	162	
EINT2_0	External interrupt ch.2 input pin (0)	15	15	21	29	
EINT2_1	External interrupt ch.2 input pin (1)	79	94	112	142	
EINT2_3	External interrupt ch.2 input pin (3)	14	14	20	28	
EINT2_4	External interrupt ch.2 input pin (4)	-	-	-	92	
EINT2_5	External interrupt ch.2 input pin (5)	-	87	100	124	
EINT2_6	External interrupt ch.2 input pin (6)	-	113	133	163	
EINT3_0	External interrupt ch.3 input pin (0)	-	25	31	39	
EINT3_1	External interrupt ch.3 input pin (1)	88	103	121	151	
EINT3_3	External interrupt ch.3 input pin (3)	16	16	22	30	
EINT3_4	External interrupt ch.3 input pin (4)	-	-	-	93	
EINT3_5	External interrupt ch.3 input pin (5)	-	88	101	125	
EINT3_6	External interrupt ch.3 input pin (6)	-	116	136	166	
EINT4_0	External interrupt ch.4 input pin (0)	-	44	50	64	
EINT4_1	External interrupt ch.4 input pin (1)	-	118	138	168	
EINT4_3	External interrupt ch.4 input pin (3)	17	17	23	31	
EINT4_4	External interrupt ch.4 input pin (4)	-	-	-	95	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
EINT4_5	External interrupt ch.4 input pin (5)	-	-	102	126	
EINT5_0	External interrupt ch.5 input pin (0)	-	45	51	65	
EINT5_1	External interrupt ch.5 input pin (1)	-	-	140	170	
EINT5_2	External interrupt ch.5 input pin (2)	-	-	5	5	
EINT5_3	External interrupt ch.5 input pin (3)	-	20	26	34	
EINT5_4	External interrupt ch.5 input pin (4)	-	-	-	97	
EINT6_0	External interrupt ch.6 input pin (0)	-	46	52	66	
EINT6_1	External interrupt ch.6 input pin (1)	-	-	-	59	
EINT6_2	External interrupt ch.6 input pin (2)	-	-	6	6	
EINT6_3	External interrupt ch.6 input pin (3)	-	22	28	36	
EINT6_4	External interrupt ch.6 input pin (4)	-	-	-	98	
EINT6_5	External interrupt ch.6 input pin (5)	-	-	104	128	
EINT7_0	External interrupt ch.7 input pin (0)	-	47	53	67	
EINT7_1	External interrupt ch.7 input pin (1)	-	-	-	171	
EINT7_2	External interrupt ch.7 input pin (2)	-	-	7	7	
EINT7_3	External interrupt ch.7 input pin (3)	21	23	29	37	
EINT7_5	External interrupt ch.7 input pin (5)	74	89	105	129	
EINT8_0	External interrupt ch.8 input pin (0)	40	49	56	70	
EINT8_1	External interrupt ch.8 input pin (1)	-	-	-	61	
EINT8_2	External interrupt ch.8 input pin (2)	2	2	8	8	
EINT8_3	External interrupt ch.8 input pin (3)	-	24	30	38	
EINT9_0	External interrupt ch.9 input pin (0)	41	50	57	71	
EINT9_1	External interrupt ch.9 input pin (1)	-	-	-	172	
EINT9_2	External interrupt ch.9 input pin (2)	-	-	-	10	
EINT9_3	External interrupt ch.9 input pin (3)	22	26	32	40	
EINT9_4	External interrupt ch.9 input pin (4)	54	64	77	101	
EINT9_5	External interrupt ch.9 input pin (5)	-	-	107	131	
EINT10_0	External interrupt ch.10 input pin (0)	44	53	60	74	
EINT10_2	External interrupt ch.10 input pin (2)	-	-	-	11	
EINT10_3	External interrupt ch.10 input pin (3)	-	29	35	43	
EINT10_5	External interrupt ch.10 input pin (5)	80	95	113	143	
EINT11_0	External interrupt ch.11 input pin (0)	45	54	61	75	
EINT11_1	External interrupt ch.11 input pin (1)	-	-	141	173	
EINT11_2	External interrupt ch.11 input pin (2)	-	-	-	14	
EINT11_4	External interrupt ch.11 input pin (4)	58	68	81	105	
EINT11_5	External interrupt ch.11 input pin (5)	81	96	114	144	
EINT12_0	External interrupt ch.12 input pin (0)	-	55	62	76	
EINT12_1	External interrupt ch.12 input pin (1)	-	-	54	68	
EINT12_2	External interrupt ch.12 input pin (2)	-	-	-	15	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
EINT12_4	External interrupt ch.12 input pin (4)	60	70	83	107	
EINT12_5	External interrupt ch.12 input pin (3)	82	97	115	145	
EINT13_0	External interrupt ch.13 input pin (0)	-	-	64	78	
EINT13_1	External interrupt ch.13 input pin (1)	-	-	142	174	
EINT13_4	External interrupt ch.13 input pin (4)	62	72	85	109	
EINT13_5	External interrupt ch.13 input pin (5)	83	98	116	146	
EINT14_0	External interrupt ch.14 input pin (0)	-	-	65	79	
EINT14_1	External interrupt ch.14 input pin (1)	-	-	66	80	
EINT14_4	External interrupt ch.14 input pin (4)	63	73	86	110	
EINT14_5	External interrupt ch.14 input pin (5)	84	99	117	147	
EINT15_0	External interrupt ch.15 input pin (0)	-	-	67	81	
EINT15_1	External interrupt ch.15 input pin (1)	-	-	143	175	
EINT15_2	External interrupt ch.15 input pin (2)	-	-	-	16	
EINT15_4	External interrupt ch.15 input pin (4)	64	74	87	111	
EINT15_5	External interrupt ch.15 input pin (5)	85	100	118	148	
EINT16_0	External interrupt ch.16 input pin (0)	-	-	74	96	
EINT16_2	External interrupt ch.16 input pin (2)	-	-	-	17	
EINT16_5	External interrupt ch.16 input pin (5)	89	104	122	152	
EINT17_0	External interrupt ch.17 input pin (0)	57	67	80	104	
EINT17_1	External interrupt ch.17 input pin (1)	-	-	-	91	
EINT17_2	External interrupt ch.17 input pin (2)	-	-	-	18	
EINT17_3	External interrupt ch.17 input pin (3)	-	-	-	60	
EINT17_4	External interrupt ch.17 input pin (4)	67	77	90	114	
EINT17_5	External interrupt ch.17 input pin (5)	90	105	123	153	
EINT18_0	External interrupt ch.18 input pin (0)	59	69	82	106	
EINT18_1	External interrupt ch.18 input pin (1)	-	-	-	58	
EINT18_2	External interrupt ch.18 input pin (2)	5	5	11	19	
EINT18_3	External interrupt ch.18 input pin (3)	-	-	-	63	
EINT18_4	External interrupt ch.18 input pin (4)	68	78	91	115	
EINT18_5	External interrupt ch.18 input pin (5)	91	106	124	154	
EINT19_0	External interrupt ch.19 input pin (0)	61	71	84	108	
EINT19_1	External interrupt ch.19 input pin (1)	-	-	-	94	
EINT19_2	External interrupt ch.19 input pin (2)	6	6	12	20	
EINT19_3	External interrupt ch.19 input pin (3)	-	-	63	77	
EINT19_5	External interrupt ch.19 input pin (5)	92	107	125	155	
EINT20_0	External interrupt ch.20 input pin (0)	69	79	92	116	
EINT20_1	External interrupt ch.20 input pin (1)	-	-	-	83	
EINT20_2	External interrupt ch.20 input pin (2)	7	7	13	21	
EINT20_4	External interrupt ch.20 input pin (4)	70	80	93	117	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
EINT20_5	External interrupt ch.20 input pin (5)	93	108	126	156	
EINT21_0	External interrupt ch.21 input pin (0)	-	82	95	119	
EINT21_1	External interrupt ch.21 input pin (1)	-	-	-	62	
EINT21_2	External interrupt ch.21 input pin (2)	8	8	14	22	
EINT21_4	External interrupt ch.21 input pin (4)	71	81	94	118	
EINT21_5	External interrupt ch.21 input pin (5)	94	109	127	157	
EINT22_0	External interrupt ch.22 input pin (0)	-	-	103	127	
EINT22_2	External interrupt ch.22 input pin (2)	9	9	15	23	
EINT22_4	External interrupt ch.22 input pin (4)	-	83	96	120	
EINT22_5	External interrupt ch.22 input pin (5)	95	110	128	158	
EINT23_0	External interrupt ch.23 input pin (0)	-	-	106	130	
EINT23_1	External interrupt ch.23 input pin (1)	-	-	2	2	
EINT23_2	External interrupt ch.23 input pin (2)	10	10	16	24	
EINT23_5	External interrupt ch.23 input pin (5)	96	111	129	159	
ADC0_TRG0_0	ADC unit0 external trigger input pin (0)	54	64	77	101	
ADC0_TRG0_1	ADC unit0 external trigger input pin (1)	-	-	74	96	
ADC0_AN0	ADC unit0 ch.0 analog input pin	-	25	31	39	
ADC0_AN1	ADC unit0 ch.1 analog input pin	22	26	32	40	
ADC0_AN2	ADC unit0 ch.2 analog input pin	-	29	35	43	
ADC0_AN3	ADC unit0 ch.3 analog input pin	-	-	-	61	
ADC0_AN4	ADC unit0 ch.4 analog input pin	-	-	-	62	
ADC0_AN5	ADC unit0 ch.5 analog input pin	-	-	-	63	
ADC0_AN6	ADC unit0 ch.6 analog input pin	-	44	50	64	
ADC0_AN7	ADC unit0 ch.7 analog input pin	-	45	51	65	
ADC0_AN8	ADC unit0 ch.8 analog input pin	-	46	52	66	
ADC0_AN9	ADC unit0 ch.9 analog input pin	-	47	53	67	
ADC0_AN10	ADC unit0 ch.10 analog input pin	-	-	54	68	
ADC0_AN11	ADC unit0 ch.11 analog input pin	44	53	60	74	
ADC0_AN12	ADC unit0 ch.12 analog input pin	45	54	61	75	
ADC0_AN13	ADC unit0 ch.13 analog input pin	-	55	62	76	
ADC0_AN14	ADC unit0 ch.14 analog input pin	-	-	63	77	
ADC0_AN15	ADC unit0 ch.15 analog input pin	-	-	64	78	
ADC0_AN16	ADC unit0 ch.16 analog input pin	-	-	65	79	
ADC0_AN17	ADC unit0 ch.17 analog input pin	-	-	66	80	
ADC0_AN18	ADC unit0 ch.18 analog input pin	-	-	67	81	
ADC0_AN19	ADC unit0 ch.19 analog input pin	-	-	-	82	
ADC0_AN20	ADC unit0 ch.20 analog input pin	-	-	-	83	
ADC0_AN21	ADC unit0 ch.21 analog input pin	-	-	-	95	
ADC0_AN22	ADC unit0 ch.22 analog input pin	-	-	-	97	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
ADC0_AN23	ADC unit0 ch.23 analog input pin	-	-	-	98	
ADC0_AN24	ADC unit0 ch.24 analog input pin	57	67	80	104	
ADC0_AN25	ADC unit0 ch.25 analog input pin	58	68	81	105	
ADC0_AN26	ADC unit0 ch.26 analog input pin	59	69	82	106	
ADC0_AN27	ADC unit0 ch.27 analog input pin	60	70	83	107	
ADC0_AN28	ADC unit0 ch.28 analog input pin	61	71	84	108	
ADC0_AN29	ADC unit0 ch.29 analog input pin	62	72	85	109	
ADC0_AN30	ADC unit0 ch.30 analog input pin	63	73	86	110	
ADC0_AN31	ADC unit0 ch.31 analog input pin	64	74	87	111	
ADC0_AN32	ADC unit0 ch.32 analog input pin	67	77	90	114	
ADC0_AN33	ADC unit0 ch.33 analog input pin	68	78	91	115	
ADC0_AN34	ADC unit0 ch.34 analog input pin	69	79	92	116	
ADC0_AN35	ADC unit0 ch.35 analog input pin	70	80	93	117	
ADC0_AN36	ADC unit0 ch.36 analog input pin	71	81	94	118	
ADC0_AN37	ADC unit0 ch.37 analog input pin	-	82	95	119	
ADC0_AN38	ADC unit0 ch.38 analog input pin	-	83	96	120	
ADC0_AN39	ADC unit0 ch.39 analog input pin	-	84	97	121	
ADC0_AN40	ADC unit0 ch.40 analog input pin	-	87	100	124	
ADC0_AN41	ADC unit0 ch.41 analog input pin	-	88	101	125	
ADC0_AN42	ADC unit0 ch.42 analog input pin	-	-	102	126	
ADC0_AN43	ADC unit0 ch.43 analog input pin	-	-	103	127	
ADC0_AN44	ADC unit0 ch.44 analog input pin	-	-	104	128	
ADC0_AN45	ADC unit0 ch.45 analog input pin	74	89	105	129	
ADC0_AN46	ADC unit0 ch.46 analog input pin	-	-	106	130	
ADC0_AN47	ADC unit0 ch.47 analog input pin	-	-	107	131	
PWU_AN0	Partial wakeup ADC analog 0 input pin	-	44	50	64	
PWU_AN1	Partial wakeup ADC analog 1 input pin	-	45	51	65	
PWU_AN2	Partial wakeup ADC analog 2 input pin	-	46	52	66	
PWU_AN3	Partial wakeup ADC analog 3 input pin	45	54	61	75	
PWU_AN4	Partial wakeup ADC analog 4 input pin	-	55	62	76	
PWU_AN5	Partial wakeup ADC analog 5 input pin	-	-	63	77	
PWU_AN6	Partial wakeup ADC analog 6 input pin	-	-	64	78	
PWU_AN7	Partial wakeup ADC analog 7 input pin	-	-	65	79	
PWUTRG_0	Partial wakeup trigger output pin (0)	-	47	53	67	
PWUTRG_1	Partial wakeup trigger output pin (1)	-	-	-	95	
CAN0_RX_0	CAN ch.0 reception data input pin (0)	-	44	50	64	
CAN0_TX_0	CAN ch.0 transmission data output pin (0)	-	45	51	65	
CAN1_RX_0	CAN ch.1 reception data input pin (0)	-	46	52	66	
CAN1_TX_0	CAN ch.1 transmission data output pin (0)	-	47	53	67	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
CAN2_RX_0	CAN ch.2 reception data input pin (0)	44	53	60	74	
CAN2_TX_0	CAN ch.2 transmission data output pin (0)	45	54	61	75	
CAN3_RX_0	CAN ch.3 reception data input pin (0)	-	-	65	79	
CAN3_TX_0	CAN ch.3 transmission data output pin (0)	-	-	66	80	
CAN0_RX_1	CAN ch.0 reception data input pin (1)	-	-	-	59	
CAN0_TX_1	CAN ch.0 transmission data output pin (1)	-	-	-	60	
CAN1_RX_1	CAN ch.1 reception data input pin (1)	-	-	-	61	
CAN1_TX_1	CAN ch.1 transmission data output pin (1)	-	-	-	62	
CAN2_RX_1	CAN ch.2 reception data input pin (1)	-	-	-	82	
CAN2_TX_1	CAN ch.2 transmission data output pin (1)	-	-	-	83	
CAN3_RX_1	CAN ch.3 reception data input pin (1)	-	-	-	91	
CAN3_TX_1	CAN ch.3 transmission data output pin (1)	-	-	-	92	
CAN0_RX_2	CAN ch.0 reception data input pin (2)	-	-	-	171	
CAN0_TX_2	CAN ch.0 transmission data output pin (2)	-	-	-	172	
MFS0_CS0_0	Multi-function serial ch.0 serial chip select 0 I/O pin (0)	62	72	85	109	
MFS0_SCK_0	Multi-function serial ch.0 clock I/O pin (0)	60	70	83	107	
MFS0_SIN_0	Multi-function serial ch.0 serial data input pin (0)	61	71	84	108	
MFS0_SOT_0	Multi-function serial ch.0 serial data output pin (0)	59	69	82	106	
MFS1_CS0_0	Multi-function serial ch.1 serial chip select 0 I/O pin (0)	-	-	140	170	
MFS1_CS1_0	Multi-function serial ch.1 serial chip select 1 output pin (0)	-	-	141	173	
MFS1_CS2_0	Multi-function serial ch.1 serial chip select 2 output pin (0)	-	-	142	174	
MFS1_CS3_0	Multi-function serial ch.1 serial chip select 3 output pin (0)	-	-	143	175	
MFS1_SCK_0	Multi-function serial ch.1 clock I/O pin (0)	-	118	138	168	
MFS1_SIN_0	Multi-function serial ch.1 serial data input pin (0)	-	119	139	169	
MFS1_SOT_0	Multi-function serial ch.1 serial data output pin (0)	-	117	137	167	
MFS2_CS0_0	Multi-function serial ch.2 serial chip select 0 I/O pin (0)	16	16	22	30	
MFS2_CS1_0	Multi-function serial ch.2 serial chip select 1 output pin (0)	17	17	23	31	
MFS2_CS2_0	Multi-function serial ch.2 serial chip select 2 output pin (0)	-	20	26	34	
MFS2_CS3_0	Multi-function serial ch.2 serial chip select 3 output pin (0)	-	22	28	36	
MFS2_SCK_0	Multi-function serial ch.2 clock I/O pin (0)	14	14	20	28	
MFS2_SIN_0	Multi-function serial ch.2 serial data input pin (0)	15	15	21	29	
MFS2_SOT_0	Multi-function serial ch.2 serial data output pin (0)	13	13	19	27	
MFS3_CS0_0	Multi-function serial ch.3 serial chip select 0 I/O pin (0)	-	26	32	40	
MFS3_CS1_0	Multi-function serial ch.3 serial chip select 1 output pin (0)	-	29	35	43	
MFS3_SCK_0	Multi-function serial ch.3 clock I/O pin (0)	-	24	30	38	
MFS3_SIN_0	Multi-function serial ch.3 serial data input pin (0)	-	25	31	39	
MFS3_SOT_0	Multi-function serial ch.3 serial data output pin (0)	-	23	29	37	
MFS4_CS0_0	Multi-function serial ch.4 serial chip select 0 I/O pin (0)	-	-	131	161	
MFS4_CS1_0	Multi-function serial ch.4 serial chip select 1 output pin (0)	-	-	132	162	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MFS4_CS2_0	Multi-function serial ch.4 serial chip select 2 output pin (0)	-	113	133	163	
MFS4_CS3_0	Multi-function serial ch.4 serial chip select 3 output pin (0)	-	116	136	166	
MFS4_SCK_0	Multi-function serial ch.4 clock I/O pin (0)	96	111	129	159	
MFS4_SIN_0	Multi-function serial ch.4 serial data input pin (0)	97	112	130	160	
MFS4_SOT_0	Multi-function serial ch.4 serial data output pin (0)	95	110	128	158	
MFS5_CS0_0	Multi-function serial ch.5 serial chip select 0 I/O pin (0)	-	-	107	131	
MFS5_SCK_0	Multi-function serial ch.5 clock I/O pin (0)	-	-	105	129	
MFS5_SIN_0	Multi-function serial ch.5 serial data input pin (0)	-	-	106	130	
MFS5_SOT_0	Multi-function serial ch.5 serial data output pin (0)	-	-	104	128	
MFS6_CS0_0	Multi-function serial ch.6 serial chip select 0 I/O pin (0)	-	-	56	70	
MFS6_CS1_0	Multi-function serial ch.6 serial chip select 1 output pin (0)	-	-	57	71	
MFS6_SCK_0	Multi-function serial ch.6 clock I/O pin (0)	-	-	63	77	
MFS6_SIN_0	Multi-function serial ch.6 serial data input pin (0)	-	-	64	78	
MFS6_SOT_0	Multi-function serial ch.6 serial data output pin (0)	-	-	62	76	
MFS7_CS0_0	Multi-function serial ch.7 serial chip select 0 I/O pin (0)	-	-	74	96	
MFS7_CS1_0	Multi-function serial ch.7 serial chip select 1 output pin (0)	-	-	77	101	
MFS7_SCK_0	Multi-function serial ch.7 clock I/O pin (0)	-	-	66	80	
MFS7_SIN_0	Multi-function serial ch.7 serial data input pin (0)	-	-	67	81	
MFS7_SOT_0	Multi-function serial ch.7 serial data output pin (0)	-	-	65	79	
MFS8_CS0_0	Multi-function serial ch.8 serial chip select 0 I/O pin (0)	-	47	53	67	
MFS8_SCK_0	Multi-function serial ch.8 clock I/O pin (0)	-	45	51	65	
MFS8_SIN_0	Multi-function serial ch.8 serial data input pin (0)	-	46	52	66	
MFS8_SOT_0	Multi-function serial ch.8 serial data output pin (0)	-	44	50	64	
MFS9_CS0_0	Multi-function serial ch.9 serial chip select 0 I/O pin (0)	63	73	86	110	
MFS9_CS1_0	Multi-function serial ch.9 serial chip select 0 output pin (0)	64	74	87	111	
MFS9_SCK_0	Multi-function serial ch.9 clock I/O pin (0)	68	78	91	115	
MFS9_SIN_0	Multi-function serial ch.9 serial data input pin (0)	69	79	92	116	
MFS9_SOT_0	Multi-function serial ch.9 serial data output pin (0)	67	77	90	114	
MFS10_CS0_0	Multi-function serial ch.10 serial chip select 0 I/O pin (0)	-	83	96	120	
MFS10_SCK_0	Multi-function serial ch.10 clock I/O pin (0)	-	81	94	118	
MFS10_SIN_0	Multi-function serial ch.10 serial data input pin (0)	-	82	95	119	
MFS10_SOT_0	Multi-function serial ch.10 serial data output pin (0)	-	80	93	117	
MFS11_CS0_0	Multi-function serial ch.11 serial chip select 0 I/O pin (0)	-	-	97	121	
MFS11_CS1_0	Multi-function serial ch.11 serial chip select 1 output pin (0)	-	-	100	124	
MFS11_SCK_0	Multi-function serial ch.11 clock I/O pin (0)	-	-	102	126	
MFS11_SIN_0	Multi-function serial ch.11 serial data input pin (0)	-	-	103	127	
MFS11_SOT_0	Multi-function serial ch.11 serial data output pin (0)	-	-	101	125	
MFS0_CS0_1	Multi-function serial ch.0 serial chip select 0 I/O pin (1)	-	-	140	170	
MFS0_SCK_1	Multi-function serial ch.0 clock I/O pin (1)	-	-	142	174	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MFS0_SIN_1	Multi-function serial ch.0 serial data input pin (1)	-	-	143	175	
MFS0_SOT_1	Multi-function serial ch.0 serial data output pin (1)	-	-	141	173	
MFS1_CS0_1	Multi-function serial ch.1 serial chip select 0 I/O pin (1)	-	-	8	8	
MFS1_CS1_1	Multi-function serial ch.1 serial chip select 1 output pin (1)	-	-	-	9	
MFS1_CS2_1	Multi-function serial ch.1 serial chip select 2 output pin (1)	-	-	-	10	
MFS1_CS3_1	Multi-function serial ch.1 serial chip select 3 output pin (1)	-	-	-	11	
MFS1_SCK_1	Multi-function serial ch.1 clock I/O pin (1)	-	-	6	6	
MFS1_SIN_1	Multi-function serial ch.1 serial data input pin (1)	-	-	7	7	
MFS1_SOT_1	Multi-function serial ch.1 serial data output pin (1)	-	-	5	5	
MFS2_CS0_1	Multi-function serial ch.2 serial chip select 0 I/O pin (1)	84	99	117	147	
MFS2_CS1_1	Multi-function serial ch.2 serial chip select 1 output pin (1)	85	100	118	148	
MFS2_CS2_1	Multi-function serial ch.2 serial chip select 2 output pin (1)	-	-	-	140	
MFS2_CS3_1	Multi-function serial ch.2 serial chip select 3 output pin (1)	-	-	-	141	
MFS2_SCK_1	Multi-function serial ch.2 clock I/O pin (1)	82	97	115	145	
MFS2_SIN_1	Multi-function serial ch.2 serial data input pin (1)	83	98	116	146	
MFS2_SOT_1	Multi-function serial ch.2 serial data output pin (1)	81	96	114	144	
MFS3_CS0_1	Multi-function serial ch.3 serial chip select 0 I/O pin (1)	91	106	124	154	
MFS3_CS1_1	Multi-function serial ch.3 serial chip select 1 output pin (1)	92	107	125	155	
MFS3_CS2_1	Multi-function serial ch.3 serial chip select 2 output pin (1)	93	108	126	156	
MFS3_CS3_1	Multi-function serial ch.3 serial chip select 3 output pin (1)	94	109	127	157	
MFS3_SCK_1	Multi-function serial ch.3 clock I/O pin (1)	89	104	122	152	
MFS3_SIN_1	Multi-function serial ch.3 serial data input pin (1)	90	105	123	153	
MFS3_SOT_1	Multi-function serial ch.3 serial data output pin (1)	88	103	121	151	
MFS4_CS0_1	Multi-function serial ch.4 serial chip select 0 I/O pin (1)	8	8	14	22	
MFS4_CS1_1	Multi-function serial ch.4 serial chip select 1 output pin (1)	9	9	15	23	
MFS4_CS2_1	Multi-function serial ch.4 serial chip select 2 output pin (1)	10	10	16	24	
MFS4_CS3_1	Multi-function serial ch.4 serial chip select 3 output pin (1)	-	-	-	18	
MFS4_SCK_1	Multi-function serial ch.4 clock I/O pin (1)	6	6	12	20	
MFS4_SIN_1	Multi-function serial ch.4 serial data input pin (1)	7	7	13	21	
MFS4_SOT_1	Multi-function serial ch.4 serial data output pin (1)	5	5	11	19	
MFS5_CS0_1	Multi-function serial ch.5 serial chip select 0 I/O pin (1)	-	-	-	98	
MFS5_SCK_1	Multi-function serial ch.5 clock I/O pin (1)	-	-	-	96	
MFS5_SIN_1	Multi-function serial ch.5 serial data input pin (1)	-	-	-	97	
MFS5_SOT_1	Multi-function serial ch.5 serial data output pin (1)	-	-	-	95	
MFS7_SCK_1	Multi-function serial ch.7 clock I/O pin (1)	-	-	3	3	
MFS7_SIN_1	Multi-function serial ch.7 serial data input pin (1)	-	-	4	4	
MFS7_SOT_1	Multi-function serial ch.7 serial data output pin (1)	-	-	2	2	
MFS8_SCK_1	Multi-function serial ch.8 clock I/O pin (1)	-	-	-	59	
MFS8_SIN_1	Multi-function serial ch.8 serial data input pin (1)	-	-	-	60	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MFS8_SOT_1	Multi-function serial ch.8 serial data output pin (1)	-	-	-	58	
MFS9_CS0_1	Multi-function serial ch.9 serial chip select 0 I/O pin (1)	-	-	-	93	
MFS9_CS1_1	Multi-function serial ch.9 serial chip select 0 output pin (1)	-	-	-	94	
MFS9_SCK_1	Multi-function serial ch.9 clock I/O pin (1)	-	-	-	91	
MFS9_SIN_1	Multi-function serial ch.9 serial data input pin (1)	-	-	-	92	
MFS9_SOT_1	Multi-function serial ch.9 serial data output pin (1)	-	-	-	90	
MFS10_CS0_1	Multi-function serial ch.10 serial chip select 0 I/O pin (1)	-	-	-	68	
MFS10_SCK_1	Multi-function serial ch.10 clock I/O pin (1)	-	-	-	62	
MFS10_SIN_1	Multi-function serial ch.10 serial data input pin (1)	-	-	-	63	
MFS10_SOT_1	Multi-function serial ch.10 serial data output pin (1)	-	-	-	61	
MFS8_CS0_2	Multi-function serial ch.8 serial chip select 0 I/O pin (2)	-	25	31	39	
MFS8_SCK_2	Multi-function serial ch.8 clock I/O pin (2)	-	23	29	37	
MFS8_SIN_2	Multi-function serial ch.8 serial data input pin (2)	-	24	30	38	
MFS8_SOT_2	Multi-function serial ch.8 serial data output pin (2)	-	22	28	36	
MFS9_CS0_2	Multi-function serial ch.9 serial chip select 0 I/O pin (2)	16	16	22	30	
MFS9_CS1_2	Multi-function serial ch.9 serial chip select 0 output pin (2)	17	17	23	31	
MFS9_SCK_2	Multi-function serial ch.9 clock I/O pin (2)	14	14	20	28	
MFS9_SIN_2	Multi-function serial ch.9 serial data input pin (2)	15	15	21	29	
MFS9_SOT_2	Multi-function serial ch.9 serial data output pin (2)	13	13	19	27	
MFS0_SCL	I2C ch.0 clock I/O pin	-	45	51	65	
MFS0_SDA	I2C ch.0 serial data I/O pin	-	44	50	64	
MFS1_SCL	I2C ch.1 clock I/O pin	-	118	138	168	
MFS1_SDA	I2C ch.1 serial data I/O pin	-	117	137	167	
MFS4_SCL	I2C ch.4 clock I/O pin	96	111	129	159	
MFS4_SDA	I2C ch.4 serial data I/O pin	95	110	128	158	
MFS5_SCL	I2C ch.5 clock I/O pin	-	-	105	129	
MFS5_SDA	I2C ch.5 serial data I/O pin	-	-	104	128	
MFS6_SCL	I2C ch.6 clock I/O pin	-	-	63	77	
MFS6_SDA	I2C ch.6 serial data I/O pin	-	-	62	76	
MFS7_SCL	I2C ch.7 clock I/O pin	-	-	66	80	
MFS7_SDA	I2C ch.7 serial data I/O pin	-	-	65	79	
MFS8_SCL	I2C ch.8 clock I/O pin	60	70	83	107	
MFS8_SDA	I2C ch.8 serial data I/O pin	59	69	82	106	
MFS9_SCL	I2C ch.9 clock I/O pin	68	78	91	115	
MFS9_SDA	I2C ch.9 serial data I/O pin	67	77	90	114	
MFS10_SCL	I2C ch.10 clock I/O pin	71	81	94	118	
MFS10_SDA	I2C ch.10 serial data I/O pin	70	80	93	117	
MFS11_SCL	I2C ch.11 clock I/O pin	-	-	102	126	
MFS11_SDA	I2C ch.11 serial data I/O pin	-	-	101	125	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
FRT0_TEXT	Free-run timer ch.0 clock input pin	-	44	50	64	
FRT1_TEXT	Free-run timer ch.1 clock input pin	-	44	50	64	
FRT2_TEXT	Free-run timer ch.2 clock input pin	-	44	50	64	
FRT3_TEXT	Free-run timer ch.3 clock input pin	-	44	50	64	
FRT4_TEXT	Free-run timer ch.4 clock input pin	-	45	51	65	
FRT8_TEXT	Free-run timer ch.8 clock input pin	-	45	51	65	
FRT9_TEXT	Free-run timer ch.9 clock input pin	-	45	51	65	
FRT10_TEXT	Free-run timer ch.10 clock input pin	-	45	51	65	
ICU0_IN0_0	Input capture ch.0 input pin (0)	-	44	50	64	
ICU0_IN1_0	Input capture ch.1 input pin (0)	-	45	51	65	
ICU1_IN0_0	Input capture ch.2 input pin (0)	-	46	52	66	
ICU1_IN1_0	Input capture ch.3 input pin (0)	-	47	53	67	
ICU2_IN0_0	Input capture ch.4 input pin (0)	44	53	60	74	
ICU2_IN1_0	Input capture ch.5 input pin (0)	45	54	61	75	
ICU8_IN0_0	Input capture ch.6 input pin (0)	-	55	62	76	
ICU8_IN1_0	Input capture ch.7 input pin (0)	-	-	63	77	
ICU9_IN0_0	Input capture ch.8 input pin (0)	40	49	56	70	
ICU9_IN1_0	Input capture ch.9 input pin (0)	41	50	57	71	
ICU10_IN0_0	Input capture ch.10 input pin (0)	-	-	67	81	
ICU10_IN1_0	Input capture ch.11 input pin (0)	-	-	74	96	
ICU0_IN0_1	Input capture ch.0 input pin (1)	-	-	-	58	
ICU0_IN1_1	Input capture ch.1 input pin (1)	-	-	-	59	
ICU1_IN0_1	Input capture ch.2 input pin (1)	-	-	-	60	
ICU1_IN1_1	Input capture ch.3 input pin (1)	-	-	-	61	
ICU2_IN0_1	Input capture ch.4 input pin (1)	-	-	-	62	
ICU2_IN1_1	Input capture ch.5 input pin (1)	-	-	-	63	
ICU8_IN0_1	Input capture ch.6 input pin (1)	-	-	54	68	
ICU8_IN1_1	Input capture ch.7 input pin (1)	-	-	64	78	
ICU9_IN0_1	Input capture ch.8 input pin (1)	-	-	65	79	
ICU9_IN1_1	Input capture ch.9 input pin (1)	-	-	66	80	
ICU10_IN0_1	Input capture ch.10 input pin (1)	-	-	-	83	
ICU10_IN1_1	Input capture ch.11 input pin (1)	-	-	-	91	
OCU0_OUT0_0	Output compare ch.0 output pin (0)	-	44	50	64	
OCU0_OUT1_0	Output compare ch.1 output pin (0)	-	45	51	65	
OCU1_OUT0_0	Output compare ch.2 output pin (0)	44	53	60	74	
OCU1_OUT1_0	Output compare ch.3 output pin (0)	45	54	61	75	
OCU2_OUT0_0	Output compare ch.4 output pin (0)	-	55	62	76	
OCU2_OUT1_0	Output compare ch.5 output pin (0)	-	-	63	77	
OCU8_OUT0_0	Output compare ch.6 output pin (0)	40	49	56	70	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
OCU8_OUT1_0	Output compare ch.7 output pin (0)	41	50	57	71	
OCU9_OUT0_0	Output compare ch.8 output pin (0)	-	-	67	81	
OCU9_OUT1_0	Output compare ch.9 output pin (0)	-	-	-	90	
OCU10_OUT0_0	Output compare ch.10 output pin (0)	-	-	74	96	
OCU10_OUT1_0	Output compare ch.11 output pin (0)	54	64	77	101	
OCU0_OUT0_1	Output compare ch.0 output pin (1)	-	-	-	58	
OCU0_OUT1_1	Output compare ch.1 output pin (1)	-	-	-	59	
OCU1_OUT0_1	Output compare ch.2 output pin (1)	-	-	-	60	
OCU1_OUT1_1	Output compare ch.3 output pin (1)	-	-	-	61	
OCU2_OUT0_1	Output compare ch.4 output pin (1)	-	-	-	62	
OCU2_OUT1_1	Output compare ch.5 output pin (1)	-	-	-	63	
OCU8_OUT0_1	Output compare ch.6 output pin (1)	-	-	54	68	
OCU8_OUT1_1	Output compare ch.7 output pin (1)	-	-	64	78	
OCU9_OUT0_1	Output compare ch.8 output pin (1)	-	-	65	79	
OCU9_OUT1_1	Output compare ch.9 output pin (1)	-	-	66	80	
OCU10_OUT0_1	Output compare ch.10 output pin (1)	-	-	-	83	
OCU10_OUT1_1	Output compare ch.11 output pin (1)	-	-	-	91	
BT0_TIOA0_0	Base timer ch.0 TIOA output pin (0)	-	44	50	64	
BT0_TIOA1_0	Base timer ch.1 TIOA output pin (0)	-	45	51	65	
BT1_TIOA2_0	Base timer ch.2 TIOA output pin (0)	-	46	52	66	
BT1_TIOA3_0	Base timer ch.3 TIOA output pin (0)	-	47	53	67	
BT2_TIOA4_0	Base timer ch.4 TIOA output pin (0)	44	53	60	74	
BT2_TIOA5_0	Base timer ch.5 TIOA output pin (0)	45	54	61	75	
BT3_TIOA6_0	Base timer ch.6 TIOA output pin (0)	-	55	62	76	
BT3_TIOA7_0	Base timer ch.7 TIOA output pin (0)	-	-	63	77	
BT4_TIOA8_0	Base timer ch.8 TIOA output pin (0)	40	49	56	70	
BT4_TIOA9_0	Base timer ch.9 TIOA output pin (0)	41	50	57	71	
BT5_TIOA10_0	Base timer ch.10 TIOA output pin (0)	-	-	67	81	
BT5_TIOA11_0	Base timer ch.11 TIOA output pin (0)	-	-	74	96	
BT6_TIOA12_0	Base timer ch.12 TIOA output pin (0)	57	67	80	104	
BT6_TIOA13_0	Base timer ch.13 TIOA output pin (0)	58	68	81	105	
BT7_TIOA14_0	Base timer ch.14 TIOA output pin (0)	59	69	82	106	
BT7_TIOA15_0	Base timer ch.15 TIOA output pin (0)	60	70	83	107	
BT8_TIOA16_0	Base timer ch.16 TIOA output pin (0)	61	71	84	108	
BT8_TIOA17_0	Base timer ch.17 TIOA output pin (0)	62	72	85	109	
BT9_TIOA18_0	Base timer ch.18 TIOA output pin (0)	63	73	86	110	
BT9_TIOA19_0	Base timer ch.19 TIOA output pin (0)	64	74	87	111	
BT10_TIOA20_0	Base timer ch.20 TIOA output pin (0)	67	77	90	114	
BT10_TIOA21_0	Base timer ch.21 TIOA output pin (0)	68	78	91	115	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
BT11_TIOA22_0	Base timer ch.22 TIOA output pin (0)	69	79	92	116	
BT11_TIOA23_0	Base timer ch.23 TIOA output pin (0)	70	80	93	117	
BT12_TIOA24_0	Base timer ch.24 TIOA output pin (0)	-	82	95	119	
BT12_TIOA25_0	Base timer ch.25 TIOA output pin (0)	-	83	96	120	
BT13_TIOA26_0	Base timer ch.26 TIOA output pin (0)	-	84	97	121	
BT13_TIOA27_0	Base timer ch.27 TIOA output pin (0)	-	87	100	124	
BT14_TIOA28_0	Base timer ch.28 TIOA output pin (0)	-	88	101	125	
BT14_TIOA29_0	Base timer ch.29 TIOA output pin (0)	-	-	102	126	
BT15_TIOA30_0	Base timer ch.30 TIOA output pin (0)	-	-	103	127	
BT15_TIOA31_0	Base timer ch.31 TIOA output pin (0)	-	-	104	128	
BT0_TIOB0_0	Base timer ch.0 TIOB input pin (0)	54	64	77	101	
BT1_TIOB2_0	Base timer ch.2 TIOB input pin (0)	54	64	77	101	
BT2_TIOB4_0	Base timer ch.4 TIOB input pin (0)	54	64	77	101	
BT3_TIOB6_0	Base timer ch.6 TIOB input pin (0)	54	64	77	101	
BT4_TIOB8_0	Base timer ch.8 TIOB input pin (0)	54	64	77	101	
BT5_TIOB10_0	Base timer ch.10 TIOB input pin (0)	54	64	77	101	
BT6_TIOB12_0	Base timer ch.12 TIOB input pin (0)	71	81	94	118	
BT7_TIOB14_0	Base timer ch.14 TIOB input pin (0)	71	81	94	118	
BT8_TIOB16_0	Base timer ch.16 TIOB input pin (0)	71	81	94	118	
BT9_TIOB18_0	Base timer ch.18 TIOB input pin (0)	71	81	94	118	
BT10_TIOB20_0	Base timer ch.20 TIOB input pin (0)	71	81	94	118	
BT11_TIOB22_0	Base timer ch.22 TIOB input pin (0)	71	81	94	118	
BT12_TIOB24_0	Base timer ch.24 TIOB input pin (0)	74	89	105	129	
BT13_TIOB26_0	Base timer ch.26 TIOB input pin (0)	74	89	105	129	
BT14_TIOB28_0	Base timer ch.28 TIOB input pin (0)	74	89	105	129	
BT15_TIOB30_0	Base timer ch.30 TIOB input pin (0)	74	89	105	129	
BT0_TIOA0_1	Base timer ch.0 TIOA output pin (1)	-	-	143	175	
BT0_TIOA1_1	Base timer ch.1 TIOA output pin (1)	-	-	2	2	
BT1_TIOA2_1	Base timer ch.2 TIOA output pin (1)	-	-	3	3	
BT1_TIOA3_1	Base timer ch.3TIOA output pin (1)	-	-	4	4	
BT2_TIOA4_1	Base timer ch.4 TIOA output pin (1)	-	-	5	5	
BT2_TIOA5_1	Base timer ch.5 TIOA output pin (1)	-	-	6	6	
BT3_TIOA6_1	Base timer ch.6 TIOA output pin (1)	-	-	7	7	
BT3_TIOA7_1	Base timer ch.7 TIOA output pin (1)	2	2	8	8	
BT4_TIOA8_1	Base timer ch.8 TIOA output pin (1)	-	-	-	9	
BT4_TIOA9_1	Base timer ch.9 TIOA output pin (1)	-	-	-	10	
BT5_TIOA10_1	Base timer ch.10 TIOA output pin (1)	-	-	-	11	
BT5_TIOA11_1	Base timer ch.11 TIOA output pin (1)	-	-	-	14	
BT6_TIOA12_1	Base timer ch.12 TIOA output pin (1)	-	-	-	58	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
BT6_TIOA13_1	Base timer ch.13 TIOA output pin (1)	-	-	-	59	
BT7_TIOA14_1	Base timer ch.14 TIOA output pin (1)	-	-	-	60	
BT7_TIOA15_1	Base timer ch.15 TIOA output pin (1)	-	-	-	61	
BT8_TIOA16_1	Base timer ch.16 TIOA output pin (1)	-	-	-	62	
BT8_TIOA17_1	Base timer ch.17 TIOA output pin (1)	-	-	-	63	
BT9_TIOA18_1	Base timer ch.18 TIOA output pin (1)	-	-	54	68	
BT9_TIOA19_1	Base timer ch.19 TIOA output pin (1)	-	-	64	78	
BT10_TIOA20_1	Base timer ch.20 TIOA output pin (1)	-	-	-	15	
BT10_TIOA21_1	Base timer ch.21 TIOA output pin (1)	-	-	-	92	
BT11_TIOA22_1	Base timer ch.22 TIOA output pin (1)	-	-	-	83	
BT11_TIOA23_1	Base timer ch.23 TIOA output pin (1)	-	-	-	91	
BT12_TIOA24_1	Base timer ch.24 TIOA output pin (1)	81	96	114	144	
BT12_TIOA25_1	Base timer ch.25 TIOA output pin (1)	82	97	115	145	
BT13_TIOA26_1	Base timer ch.26 TIOA output pin (1)	83	98	116	146	
BT13_TIOA27_1	Base timer ch.27 TIOA output pin (1)	84	99	117	147	
BT14_TIOA28_1	Base timer ch.28 TIOA output pin (1)	85	100	118	148	
BT14_TIOA29_1	Base timer ch.29 TIOA output pin (1)	88	103	121	151	
BT15_TIOA30_1	Base timer ch.30 TIOA output pin (1)	89	104	122	152	
BT15_TIOA31_1	Base timer ch.31 TIOA output pin (1)	90	105	123	153	
BT0_TIOB0_1	Base timer ch.0 TIOB input pin (1)	-	-	65	79	
BT1_TIOB2_1	Base timer ch.2 TIOB input pin (1)	-	-	65	79	
BT2_TIOB4_1	Base timer ch.4 TIOB input pin (1)	-	-	65	79	
BT3_TIOB6_1	Base timer ch.6 TIOB input pin (1)	-	-	65	79	
BT4_TIOB8_1	Base timer ch.8 TIOB input pin (1)	-	-	65	79	
BT5_TIOB10_1	Base timer ch.10 TIOB input pin (1)	-	-	65	79	
BT6_TIOB12_1	Base timer ch.12 TIOB input pin (1)	-	-	66	80	
BT7_TIOB14_1	Base timer ch.14 TIOB input pin (1)	-	-	66	80	
BT8_TIOB16_1	Base timer ch.16 TIOB input pin (1)	-	-	66	80	
BT9_TIOB18_1	Base timer ch.18 TIOB input pin (1)	-	-	66	80	
BT10_TIOB20_1	Base timer ch.20 TIOB input pin (1)	-	-	66	80	
BT11_TIOB22_1	Base timer ch.22 TIOB input pin (1)	-	-	66	80	
BT12_TIOB24_1	Base timer ch.24 TIOB input pin (1)	91	106	124	154	
BT13_TIOB26_1	Base timer ch.27 TIOB input pin (1)	91	106	124	154	
BT14_TIOB28_1	Base timer ch.28 TIOB input pin (1)	91	106	124	154	
BT15_TIOB30_1	Base timer ch.30 TIOB input pin (1)	91	106	124	154	
QPRC8_AIN	Quad Position & Revolution Counter ch.8 AIN input pin	-	44	50	64	
QPRC8_BIN	Quad Position & Revolution Counter ch.8 BIN input pin	-	45	51	65	
QPRC8_ZIN	Quad Position & Revolution Counter ch.8 ZIN input pin	-	46	52	66	
QPRC9_AIN	Quad Position & Revolution Counter ch.9 AIN input pin	-	47	53	67	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
QPRC9_BIN	Quad Position & Revolution Counter ch.9 BIN input pin	-	53	60	74	
QPRC9_ZIN	Quad Position & Revolution Counter ch.9 ZIN input pin	-	54	61	75	
RLT0_TIN_0	Reload timer ch.0 event input pin (0)	-	44	50	64	
RLT0_TOT_0	Reload timer ch.0 output pin (0)	-	45	51	65	
RLT1_TIN_0	Reload timer ch.1 event input pin (0)	-	46	52	66	
RLT1_TOT_0	Reload timer ch.1 output pin (0)	-	47	53	67	
RLT2_TIN_0	Reload timer ch.2 event input pin (0)	-	-	-	90	
RLT2_TOT_0	Reload timer ch.2 output pin (0)	-	-	-	81	
RLT3_TIN_0	Reload timer ch.3 event input pin (0)	-	-	74	96	
RLT3_TOT_0	Reload timer ch.3 output pin (0)	-	-	77	101	
RLT16_TIN_0	Reload timer ch.4 event input pin (0)	44	53	60	74	
RLT16_TOT_0	Reload timer ch.4 output pin (0)	45	54	61	75	
RLT17_TIN_0	Reload timer ch.5 event input pin (0)	40	49	56	70	
RLT17_TOT_0	Reload timer ch.5 output pin (0)	41	50	57	71	
RLT0_TIN_1	Reload timer ch.0 event input pin (1)	-	-	-	58	
RLT0_TOT_1	Reload timer ch.0 output pin (1)	-	-	-	59	
RLT1_TIN_1	Reload timer ch.1 event input pin (1)	-	-	-	60	
RLT1_TOT_1	Reload timer ch.1 output pin (1)	-	-	-	61	
RLT2_TIN_1	Reload timer ch.2 event input pin (1)	-	-	65	79	
RLT2_TOT_1	Reload timer ch.2 output pin (1)	-	-	66	80	
RLT3_TIN_1	Reload timer ch.3 event input pin (1)	-	-	-	92	
RLT3_TOT_1	Reload timer ch.3 output pin (1)	-	-	-	93	
RLT16_TIN_1	Reload timer ch.4 event input pin (1)	-	-	-	62	
RLT16_TOT_1	Reload timer ch.4 output pin (1)	-	-	-	63	
RLT17_TIN_1	Reload timer ch.5 event input pin (1)	-	-	54	68	
RLT17_TOT_1	Reload timer ch.5 output pin (1)	-	-	64	78	
RTC0_WOT	RTC ch.0 overflow output pin	54	64	77	101	
SYSC0_CLK_0	Clock monitor output pin (0)	54	64	77	101	
SYSC0_CLK_1	Clock monitor output pin (1)	-	-	67	81	
TRACE0_0	Trace data 0 output pin (0)	-	-	60	74	
TRACE1_0	Trace data 1 output pin (0)	-	-	61	75	
TRACE2_0	Trace data 2 output pin (0)	-	-	62	76	
TRACE3_0	Trace data 3 output pin (0)	-	-	63	77	
TRACE_CTL_0	Trace control output pin (0)	-	-	65	79	
TRACE_CLK_0	Trace clock output pin (0)	-	-	64	78	
TRACE0_1	Trace data 0 output pin (1)	-	-	-	58	
TRACE1_1	Trace data 1 output pin (1)	-	-	-	59	
TRACE2_1	Trace data 2 output pin (1)	-	-	-	60	
TRACE3_1	Trace data 3 output pin (1)	-	-	-	61	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
TRACE_CTL_1	Trace control output pin (1)	-	-	-	63	
TRACE_CLK_1	Trace clock output pin (1)	-	-	-	62	
PCMP0_BL_0	PCM PWM ch.0 output pin (0)	-	-	50	64	
PCMP0_BH_0	PCM PWM ch.0 output pin (0)	-	-	51	65	
PCMP0_AL_0	PCM PWM ch.0 output pin (0)	-	-	52	66	
PCMP0_AH_0	PCM PWM ch.0 output pin (0)	-	-	53	67	
PCMP1_BL_0	PCM PWM ch.1 output pin (0)	-	-	60	74	
PCMP1_BH_0	PCM PWM ch.1 output pin (0)	-	-	61	75	
PCMP1_AL_0	PCM PWM ch.1 output pin (0)	-	-	62	76	
PCMP1_AH_0	PCM PWM ch.1 output pin (0)	-	-	63	77	
PCMP0_BL_1	PCM PWM ch.0 output pin (1)	-	-	5	5	
PCMP0_BH_1	PCM PWM ch.0 output pin (1)	-	-	6	6	
PCMP0_AL_1	PCM PWM ch.0 output pin (1)	-	-	7	7	
PCMP0_AH_1	PCM PWM ch.0 output pin (1)	-	-	8	8	
PCMP1_BL_1	PCM PWM ch.1 output pin (1)	-	-	13	21	
PCMP1_BH_1	PCM PWM ch.1 output pin (1)	-	-	14	22	
PCMP1_AL_1	PCM PWM ch.1 output pin (1)	-	-	15	23	
PCMP1_AH_1	PCM PWM ch.1 output pin (1)	-	-	16	24	
SG0_SGA_0	Sound generator ch.0 SGA output pin (0)	-	46	52	66	
SG0_SGO_0	Sound generator ch.0 SGO output pin (0)	-	47	53	67	
SG1_SGA_0	Sound generator ch.1 SGA output pin (0)	44	53	60	74	
SG1_SGO_0	Sound generator ch.1 SGO output pin (0)	45	54	61	75	
SG2_SGA_0	Sound generator ch.2 SGA output pin (0)	-	-	62	76	
SG2_SGO_0	Sound generator ch.2 SGO output pin (0)	-	-	63	77	
SG3_SGA_0	Sound generator ch.3 SGA output pin (0)	-	-	64	78	
SG3_SGO_0	Sound generator ch.3 SGO output pin (0)	-	-	65	79	
SG4_SGA_0	Sound generator ch.4 SGA output pin (0)	-	-	66	80	
SG4_SGO_0	Sound generator ch.4 SGO output pin (0)	-	-	67	81	
SG0_SGA_1	Sound generator ch.0 SGA output pin (1)	-	-	-	58	
SG0_SGO_1	Sound generator ch.0 SGO output pin (1)	-	-	-	59	
SG1_SGA_1	Sound generator ch.1 SGA output pin (1)	-	-	-	60	
SG1_SGO_1	Sound generator ch.1 SGO output pin (1)	-	-	-	61	
SG2_SGA_1	Sound generator ch.2 SGA output pin (1)	-	-	-	62	
SG2_SGO_1	Sound generator ch.2 SGO output pin (1)	-	-	-	63	
SG3_SGA_1	Sound generator ch.3 SGA output pin (1)	-	-	-	93	
SG3_SGO_1	Sound generator ch.3 SGO output pin (1)	-	-	-	94	
SG4_SGA_1	Sound generator ch.4 SGA output pin (1)	-	-	-	97	
SG4_SGO_1	Sound generator ch.4 SGO output pin (1)	-	-	-	98	
SG0_SGA_2	Sound generator ch.0 SGA output pin (2)	89	104	122	152	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
SG0_SGO_2	Sound generator ch.0 SGO output pin (2)	90	105	123	153	
SG1_SGA_2	Sound generator ch.1 SGA output pin (2)	91	106	124	154	
SG1_SGO_2	Sound generator ch.1 SGO output pin (2)	92	107	125	155	
DDRHSSPI0_SCLK	DDR HS-SPI clock output pin	-	20	26	34	
DDRHSSPI0_SDATA0	DDR HS-SPI0 data 0 I/O pin	-	22	28	36	
DDRHSSPI0_SDATA1	DDR HS-SPI0 data 1 I/O pin	-	24	30	38	
DDRHSSPI0_SDATA2	DDR HS-SPI0 data 2 I/O pin	-	23	29	37	
DDRHSSPI0_SDATA3	DDR HS-SPI0 data 3 I/O pin	-	26	32	40	
DDRHSSPI1_SDATA0	DDR HS-SPI1 data 0 I/O pin	-	13	19	27	
DDRHSSPI1_SDATA1	DDR HS-SPI1 data 1 I/O pin	-	15	21	29	
DDRHSSPI1_SDATA2	DDR HS-SPI1 data 2 I/O pin	-	14	20	28	
DDRHSSPI1_SDATA3	DDR HS-SPI1 data 3 I/O pin	-	17	23	31	
DDRHSSPI0_SSEL	DDR HS-SPI select 0 output pin	-	25	31	39	
DDRHSSPI1_SSEL	DDR HS-SPI select 1 output pin	-	16	22	30	
SMC0_PWM1P	SMC ch.0 output pin	57	67	80	104	
SMC0_PWM1M	SMC ch.0 output pin	58	68	81	105	
SMC0_PWM2P	SMC ch.0 output pin	59	69	82	106	
SMC0_PWM2M	SMC ch.0 output pin	60	70	83	107	
SMC1_PWM1P	SMC ch.1 output pin	61	71	84	108	
SMC1_PWM1M	SMC ch.1 output pin	62	72	85	109	
SMC1_PWM2P	SMC ch.1 output pin	63	73	86	110	
SMC1_PWM2M	SMC ch.1 output pin	64	74	87	111	
SMC2_PWM1P	SMC ch.2 output pin	67	77	90	114	
SMC2_PWM1M	SMC ch.2 output pin	68	78	91	115	
SMC2_PWM2P	SMC ch.2 output pin	69	79	92	116	
SMC2_PWM2M	SMC ch.2 output pin	70	80	93	117	
SMC3_PWM1P	SMC ch.3 output pin	-	81	94	118	
SMC3_PWM1M	SMC ch.3 output pin	-	82	95	119	
SMC3_PWM2P	SMC ch.3 output pin	-	83	96	120	
SMC3_PWM2M	SMC ch.3 output pin	-	84	97	121	
SMC4_PWM1P	SMC ch.4 output pin	-	-	100	124	
SMC4_PWM1M	SMC ch.4 output pin	-	-	101	125	
SMC4_PWM2P	SMC ch.4 output pin	-	-	102	126	
SMC4_PWM2M	SMC ch.4 output pin	-	-	103	127	
SMC5_PWM1P	SMC ch.5 output pin	-	-	104	128	
SMC5_PWM1M	SMC ch.5 output pin	-	-	105	129	
SMC5_PWM2P	SMC ch.5 output pin	-	-	106	130	
SMC5_PWM2M	SMC ch.5 output pin	-	-	107	131	
MAD0	External bus address output pin	-	-	136	166	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MAD1	External bus address output pin	-	-	137	167	
MAD2	External bus address output pin	-	-	138	168	
MAD3	External bus address output pin	-	-	139	169	
MAD4	External bus address output pin	-	-	140	170	
MAD5	External bus address output pin	-	-	141	173	
MAD6	External bus address output pin	-	-	142	174	
MAD7	External bus address output pin	-	-	143	175	
MAD8	External bus address output pin	-	-	2	2	
MAD9	External bus address output pin	-	-	3	3	
MAD10	External bus address output pin	-	-	4	4	
MAD11	External bus address output pin	-	-	5	5	
MAD12	External bus address output pin	-	-	6	6	
MAD13	External bus address output pin	-	-	7	7	
MAD14	External bus address output pin	-	-	8	8	
MAD15	External bus address output pin	-	-	-	9	
MAD16	External bus address output pin	-	-	-	10	
MAD17	External bus address output pin	-	-	-	11	
MAD18	External bus address output pin	-	-	-	14	
MAD19	External bus address output pin	-	-	-	15	
MAD20	External bus address output pin	-	-	-	16	
MAD21	External bus address output pin	-	-	-	17	
MCSX0	External bus chip select output pin	-	-	118	148	
MCSX1	External bus chip select output pin	-	-	125	155	
MCSX2	External bus chip select output pin	-	-	15	23	
MCSX3	External bus chip select output pin	-	-	16	24	
MCLK	External bus clock output pin	-	-	13	21	
MDATA0	External bus data I/O pin	-	-	126	156	
MDATA1	External bus data I/O pin	-	-	127	157	
MDATA2	External bus data I/O pin	-	-	128	158	
MDATA3	External bus data I/O pin	-	-	129	159	
MDATA4	External bus data I/O pin	-	-	130	160	
MDATA5	External bus data I/O pin	-	-	131	161	
MDATA6	External bus data I/O pin	-	-	132	162	
MDATA7	External bus data I/O pin	-	-	133	163	
MDATA8	External bus data I/O pin	-	-	114	144	
MDATA9	External bus data I/O pin	-	-	115	145	
MDATA10	External bus data I/O pin	-	-	116	146	
MDATA11	External bus data I/O pin	-	-	117	147	
MDATA12	External bus data I/O pin	-	-	121	151	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
MDATA13	External bus data I/O pin	-	-	122	152	
MDATA14	External bus data I/O pin	-	-	123	153	
MDATA15	External bus data I/O pin	-	-	124	154	
MRDY	External bus ready input pin	-	-	113	143	
MDQM0	External bus data mask output pin 0	-	-	14	22	
MDQM1	External bus data mask output pin 1	-	-	-	18	
MWEX	External bus write enable output pin	-	-	12	20	
MOEX	External bus output enable output pin	-	-	11	19	
CS#	LCD Bus IF CS# output pin	-	2	8	8	
WR#	LCD Bus IF WR# output pin	-	5	11	19	
RD#	LCD Bus IF RD# output pin	-	6	12	20	
TE	LCD Bus IF TE input pin	-	10	16	24	
RES#	LCD Bus IF RES# output pin	-	9	15	23	
RS	LCD Bus IF RS output pin	-	8	14	22	
LCDD0	LCD Bus IF LCDD0 I/O pin	-	109	127	157	
LCDD1	LCD Bus IF LCDD1 I/O pin	-	110	128	158	
LCDD2	LCD Bus IF LCDD2 I/O pin	-	111	129	159	
LCDD3	LCD Bus IF LCDD3 I/O pin	-	112	130	160	
LCDD4	LCD Bus IF LCDD4 I/O pin	-	116	136	166	
LCDD5	LCD Bus IF LCDD5 I/O pin	-	117	137	167	
LCDD6	LCD Bus IF LCDD6 I/O pin	-	118	138	168	
LCDD7	LCD Bus IF LCDD7 I/O pin	-	119	139	169	
LCDD8	LCD Bus IF LCDD8 I/O pin	-	-	140	170	
LCDD9	LCD Bus IF LCDD9 I/O pin	-	-	141	173	
LCDD10	LCD Bus IF LCDD10 I/O pin	-	-	142	174	
LCDD11	LCD Bus IF LCDD11 I/O pin	-	-	143	175	
LCDD12	LCD Bus IF LCDD12 I/O pin	-	-	2	2	
LCDD13	LCD Bus IF LCDD13 I/O pin	-	-	3	3	
LCDD14	LCD Bus IF LCDD14 I/O pin	-	-	4	4	
LCDD15	LCD Bus IF LCDD15 I/O pin	-	-	5	5	
LCDD16	LCD Bus IF LCDD16 I/O pin	-	-	6	6	
LCDD17	LCD Bus IF LCDD17 I/O pin	-	-	7	7	
LCD0_COM0	LCDC Common analog output pin	88	103	121	151	
LCD0_COM1	LCDC Common analog output pin	85	100	118	148	
LCD0_COM2	LCDC Common analog output pin	84	99	117	147	
LCD0_COM3	LCDC Common analog output pin	83	98	116	146	
LCD0_SEG0	LCDC Segment(Duty) analog output pin	9	9	15	23	
LCD0_SEG1	LCDC Segment(Duty) analog output pin	8	8	14	22	
LCD0_SEG2	LCDC Segment(Duty) analog output pin	7	7	13	21	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
LCD0_SEG3	LCDC Segment(Duty) analog output pin	6	6	12	20	
LCD0_SEG4	LCDC Segment(Duty) analog output pin	5	5	11	19	
LCD0_SEG5	LCDC Segment(Duty) analog output pin	2	2	8	8	
LCD0_SEG6	LCDC Segment(Duty) analog output pin	-	-	7	7	
LCD0_SEG7	LCDC Segment(Duty) analog output pin	-	-	6	6	
LCD0_SEG8	LCDC Segment(Duty) analog output pin	-	-	5	5	
LCD0_SEG9	LCDC Segment(Duty) analog output pin	-	-	4	4	
LCD0_SEG10	LCDC Segment(Duty) analog output pin	-	-	3	3	
LCD0_SEG11	LCDC Segment(Duty) analog output pin	-	-	2	2	
LCD0_SEG12	LCDC Segment(Duty) analog output pin	-	-	143	175	
LCD0_SEG13	LCDC Segment(Duty) analog output pin	-	-	142	174	
LCD0_SEG14	LCDC Segment(Duty) analog output pin	-	-	141	173	
LCD0_SEG15	LCDC Segment(Duty) analog output pin	-	-	140	170	
LCD0_SEG16	LCDC Segment(Duty) analog output pin	-	119	139	169	
LCD0_SEG17	LCDC Segment(Duty) analog output pin	-	118	138	168	
LCD0_SEG18	LCDC Segment(Duty) analog output pin	-	117	137	167	
LCD0_SEG19	LCDC Segment(Duty) analog output pin	-	116	136	166	
LCD0_SEG20	LCDC Segment(Duty) analog output pin	-	113	133	163	
LCD0_SEG21	LCDC Segment(Duty) analog output pin	-	-	132	162	
LCD0_SEG22	LCDC Segment(Duty) analog output pin	-	-	131	161	
LCD0_SEG23	LCDC Segment(Duty) analog output pin	97	112	130	160	
LCD0_SEG24	LCDC Segment(Duty) analog output pin	96	111	129	159	
LCD0_SEG25	LCDC Segment(Duty) analog output pin	95	110	128	158	
LCD0_SEG26	LCDC Segment(Duty) analog output pin	94	109	127	157	
LCD0_SEG27	LCDC Segment(Duty) analog output pin	93	108	126	156	
LCD0_SEG28	LCDC Segment(Duty) analog output pin	92	107	125	155	
LCD0_SEG29	LCDC Segment(Duty) analog output pin	91	106	124	154	
LCD0_SEG30	LCDC Segment(Duty) analog output pin	90	105	123	153	
LCD0_SEG31	LCDC Segment(Duty) analog output pin	89	104	122	152	
LCD0_V0	LCDC Reference Voltage V0 analog input pin	82	97	115	145	
LCD0_V1	LCDC Reference Voltage V1 analog input pin	81	96	114	144	
LCD0_V2	LCDC Reference Voltage V2 analog input pin	80	95	113	143	
LCD0_V3	LCDC Reference Voltage V3 analog input pin	79	94	112	142	
IND0_OUT_0	Indicator PWM output pin (0)	44	53	60	74	
IND0_OUT_1	Indicator PWM output pin (1)	-	-	74	96	
I2S0_ECLK_0	I2S external clock ch.0 input pin (0)	-	-	5	5	
I2S0_SD_0	I2S serial data ch.0 I/O pin (0)	-	-	6	6	
I2S0_WS_0	I2S word select ch.0 I/O pin (0)	-	-	7	7	
I2S0_SCK_0	I2S continuous serial clock ch.0 I/O pin (0)	-	-	8	8	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
I2S0_ECLK_1	I2S external clock ch.0 input pin (1)	81	96	114	144	
I2S0_SD_1	I2S serial data ch.0 I/O pin (1)	82	97	115	145	
I2S0_WS_1	I2S word select ch.0 I/O pin (1)	83	98	116	146	
I2S0_SCK_1	I2S continuous serial clock ch.0 I/O pin (1)	84	99	117	147	
I2S1_ECLK_0	I2S external clock ch.1 input pin (0)	-	-	143	175	
I2S1_SD_0	I2S serial data ch.1 I/O pin (0)	-	-	2	2	
I2S1_WS_0	I2S word select ch.1 I/O pin (0)	-	-	3	3	
I2S1_SCK_0	I2S continuous serial clock ch.1 I/O pin (0)	-	-	4	4	
JTAG_NTRST	JTAG test reset input pin	31	36	42	50	
JTAG_TCK	JTAG test clock input pin	34	39	45	53	
JTAG_TDI	JTAG test data input pin	33	38	44	52	
JTAG_TDO	JTAG test data output pin	32	37	43	51	
JTAG_TMS	JTAG test mode state input pin	35	40	46	54	
P0_00	General-purpose I/O port	-	-	2	2	
P0_01	General-purpose I/O port	-	-	3	3	
P0_02	General-purpose I/O port	-	-	4	4	
P0_03	General-purpose I/O port	-	-	5	5	
P0_04	General-purpose I/O port	-	-	6	6	
P0_05	General-purpose I/O port	-	-	7	7	
P0_06	General-purpose I/O port	2	2	8	8	
P0_07	General-purpose I/O port	5	5	11	19	
P0_08	General-purpose I/O port	6	6	12	20	
P0_09	General-purpose I/O port	7	7	13	21	
P0_10	General-purpose I/O port	8	8	14	22	
P0_11	General-purpose I/O port	9	9	15	23	
P0_12	General-purpose I/O port	10	10	16	24	
P0_13	General-purpose I/O port	13	13	19	27	
P0_14	General-purpose I/O port	14	14	20	28	
P0_15	General-purpose I/O port	15	15	21	29	
P0_16	General-purpose I/O port	16	16	22	30	
P0_17	General-purpose I/O port	17	17	23	31	
P0_18	General-purpose I/O port	-	20	26	34	
P0_19	General-purpose I/O port	-	22	28	36	
P0_20	General-purpose I/O port	21	23	29	37	
P0_21	General-purpose I/O port	-	24	30	38	
P0_22	General-purpose I/O port	-	25	31	39	
P0_23	General-purpose I/O port	22	26	32	40	
P0_24	General-purpose I/O port	-	29	35	43	
P0_25	General-purpose I/O port	-	44	50	64	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
P0_26	General-purpose I/O port	-	45	51	65	
P0_27	General-purpose I/O port	-	46	52	66	
P0_28	General-purpose I/O port	-	47	53	67	
P0_29	General-purpose I/O port	-	-	54	68	
P0_30	General-purpose I/O port	40	49	56	70	
P0_31	General-purpose I/O port	41	50	57	71	
P1_00	General-purpose I/O port	44	53	60	74	
P1_01	General-purpose I/O port	45	54	61	75	
P1_02	General-purpose I/O port	-	55	62	76	
P1_03	General-purpose I/O port	-	-	63	77	
P1_04	General-purpose I/O port	-	-	64	78	
P1_05	General-purpose I/O port	-	-	65	79	
P1_06	General-purpose I/O port	-	-	66	80	
P1_07	General-purpose I/O port	-	-	67	81	
P1_08	General-purpose I/O port	-	-	74	96	
P1_09	General-purpose I/O port	54	64	77	101	
P1_10	General-purpose I/O port	57	67	80	104	
P1_11	General-purpose I/O port	58	68	81	105	
P1_12	General-purpose I/O port	59	69	82	106	
P1_13	General-purpose I/O port	60	70	83	107	
P1_14	General-purpose I/O port	61	71	84	108	
P1_15	General-purpose I/O port	62	72	85	109	
P1_16	General-purpose I/O port	63	73	86	110	
P1_17	General-purpose I/O port	64	74	87	111	
P1_18	General-purpose I/O port	67	77	90	114	
P1_19	General-purpose I/O port	68	78	91	115	
P1_20	General-purpose I/O port	69	79	92	116	
P1_21	General-purpose I/O port	70	80	93	117	
P1_22	General-purpose I/O port	71	81	94	118	
P1_23	General-purpose I/O port	-	82	95	119	
P1_24	General-purpose I/O port	-	83	96	120	
P1_25	General-purpose I/O port	-	84	97	121	
P1_26	General-purpose I/O port	-	87	100	124	
P1_27	General-purpose I/O port	-	88	101	125	
P1_28	General-purpose I/O port	-	-	102	126	
P1_29	General-purpose I/O port	-	-	103	127	
P1_30	General-purpose I/O port	-	-	104	128	
P1_31	General-purpose I/O port	74	89	105	129	
P2_00	General-purpose I/O port	-	-	106	130	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
P2_01	General-purpose I/O port	-	-	107	131	
P2_02	General-purpose input port	79	94	112	142	
P2_03	General-purpose I/O port	80	95	113	143	
P2_04	General-purpose I/O port	81	96	114	144	
P2_05	General-purpose I/O port	82	97	115	145	
P2_06	General-purpose I/O port	83	98	116	146	
P2_07	General-purpose I/O port	84	99	117	147	
P2_08	General-purpose I/O port	85	100	118	148	
P2_09	General-purpose I/O port	88	103	121	151	
P2_10	General-purpose I/O port	89	104	122	152	
P2_11	General-purpose I/O port	90	105	123	153	
P2_12	General-purpose I/O port	91	106	124	154	
P2_13	General-purpose I/O port	92	107	125	155	
P2_14	General-purpose I/O port	93	108	126	156	
P2_15	General-purpose I/O port	94	109	127	157	
P2_16	General-purpose I/O port	95	110	128	158	
P2_17	General-purpose I/O port	96	111	129	159	
P2_18	General-purpose I/O port	97	112	130	160	
P2_19	General-purpose I/O port	-	-	131	161	
P2_20	General-purpose I/O port	-	-	132	162	
P2_21	General-purpose I/O port	-	113	133	163	
P2_22	General-purpose I/O port	-	116	136	166	
P2_23	General-purpose I/O port	-	117	137	167	
P2_24	General-purpose I/O port	-	118	138	168	
P2_25	General-purpose I/O port	-	119	139	169	
P2_26	General-purpose I/O port	-	-	140	170	
P2_27	General-purpose I/O port	-	-	141	173	
P2_28	General-purpose I/O port	-	-	142	174	
P2_29	General-purpose I/O port	-	-	143	175	
P3_00	General-purpose I/O port	-	-	-	9	
P3_01	General-purpose I/O port	-	-	-	10	
P3_02	General-purpose I/O port	-	-	-	11	
P3_03	General-purpose I/O port	-	-	-	14	
P3_04	General-purpose I/O port	-	-	-	15	
P3_05	General-purpose I/O port	-	-	-	16	
P3_06	General-purpose I/O port	-	-	-	17	
P3_07	General-purpose I/O port	-	-	-	18	
P3_08	General-purpose I/O port	-	-	-	58	
P3_09	General-purpose I/O port	-	-	-	59	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
P3_10	General-purpose I/O port	-	-	-	60	
P3_11	General-purpose I/O port	-	-	-	61	
P3_12	General-purpose I/O port	-	-	-	62	
P3_13	General-purpose I/O port	-	-	-	63	
P3_14	General-purpose I/O port	-	-	-	82	
P3_15	General-purpose I/O port	-	-	-	83	
P3_16	General-purpose I/O port	-	-	-	90	
P3_17	General-purpose I/O port	-	-	-	91	
P3_18	General-purpose I/O port	-	-	-	92	
P3_19	General-purpose I/O port	-	-	-	93	
P3_20	General-purpose I/O port	-	-	-	94	
P3_21	General-purpose I/O port	-	-	-	95	
P3_22	General-purpose I/O port	-	-	-	97	
P3_23	General-purpose I/O port	-	-	-	98	
P3_24	General-purpose I/O port	-	-	-	134	
P3_25	General-purpose I/O port	-	-	-	135	
P3_26	General-purpose I/O port	-	-	-	138	
P3_27	General-purpose I/O port	-	-	-	139	
P3_28	General-purpose I/O port	-	-	-	140	
P3_29	General-purpose I/O port	-	-	-	141	
P3_30	General-purpose I/O port	-	-	-	171	
P3_31	General-purpose I/O port	-	-	-	172	
AVCC5	Analog power supply pin for A/D converter	46	56	68	84	
AVSS	GND pin for A/D converter	47	57	69	85	
AVRH5	Upper-limit reference voltage pin for A/D converter	49	59	71	87	
AVRL5	Lower-limit reference voltage pin for A/D converter	48	58	70	86	
C	External capacity connection output pin	36	41	47	55	
VCC53_1	Power supply pin	1	1	1	1	
		3	3	9	12	
		11	11	17	25	
		78	93	111	137	
		87	102	120	150	
		99	115	135	165	
VCC53_2	Power supply pin	18	18	24	32	
		23	27	33	41	
		25	30	36	44	

Port Name	Description	Pin No. of Package				Remark
		LQFP 100	LQFP 120	LQFP 144	LQFP 176	
VCC5	Power supply pin	38	43	49	57	
		43	52	59	73	
		50	60	72	88	
		53	63	76	100	
DVCC	Power supply pin	56	66	79	103	
		66	76	89	113	
		73	86	99	123	
		76	91	109	133	
DVSS	GND pin	55	65	78	102	
		65	75	88	112	
		72	85	98	122	
		75	90	108	132	
VSS	GND pin	4	4	10	13	
		12	12	18	26	
		19	19	25	33	
		20	21	27	35	
		24	28	34	42	
		26	31	37	45	
		37	42	48	56	
		42	51	58	72	
		51	61	73	89	
		52	62	75	99	
		77	92	110	136	
		86	101	119	149	
		98	114	134	164	
		100	120	144	176	

2. Remark

Notes:

- *The port description list shows the port function of description which is mounted and supported on the product. The function which is not described in this table is not supported and assured.*
- *See the function list of the product as well.*
- *The port name sometime has function name such as MFS0_SCK0, but it is omitted sometime. For example, RIC table describes the input function just as SCK0.*

CHAPTER 11: Port Configuration

This chapter explains the port configuration.

1. Overview
2. Configuration and Block Diagram
3. Operation
4. Registers
5. Configuration Procedure
6. Note

CODE: PORT_CONFIGURATION-JUPI-E2.1

1. Overview

This chapter explains the particular port configuration of the product PKG pins.

The microcontroller has various functions such as general purpose I/O ports, input or output timers, analog input ports and so on. Some of these functions are multiplexed implemented in a pin, and the assignment to a pin is particular for the product.

A port configuration is to determine which function and which input/output direction is applied to a port.

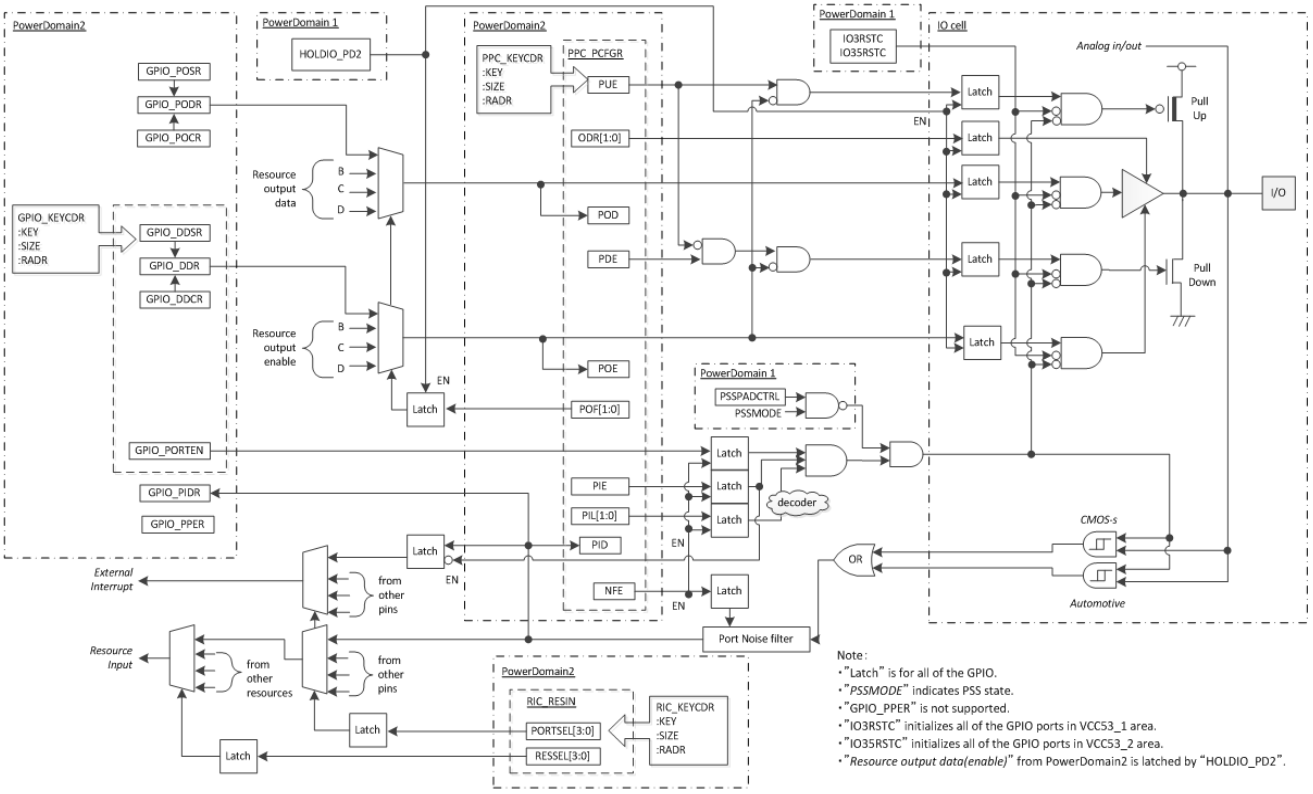
"3. Operation" of this chapter describes following.

- The particular port configurations of the resource input configuration register (RIC)
- The port output function configuration
- The analog I/O setting
- The input level setting
- The output drive capacity setting

See the common information of the registers on I/O Port.

2. Configuration and Block Diagram

Figure 2-1.



There are five latches in I/O cell area. The power supply for them is I/O power VCC5, VCC53_1, or VCC53_2. The state latches are only expected under these power supply turning-on. The power for other latches is internal regulated 1.2 V.

3. Operation

The relation between configuration and operation is described.

3.1. Resource Input Configuration

The resource input configuration (RIC) is a function to select input from an external or output from another internal resource as resource input. A resource which supports either a port input relocation or a resource inputs.

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN000 (0x0000)	SIN0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_14	P2_29	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN001 (0x0002)	SCK0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_13	P2_28	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN002 (0x0004)	SCL0	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN003 (0x0006)	SDA0	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN004 (0x0008)	MFS0_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN005 (0x000A)	SCS0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_15	P2_26	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN007 (0x000E)	SIN1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_25	P0_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN008 (0x0010)	SCK1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_24	P0_04	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN009 (0x0012)	SCL1	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN010 (0x0014)	SDA1	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN011 (0x0016)	MFS1_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN012 (0x0018)	SCS1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_26	P0_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN014 (0x001C)	SIN2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_15	P2_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN015 (0x001E)	SCK2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_14	P2_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN018 (0x0024)	MFS2_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN019 (0x0026)	SCS2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_16	P2_07	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN021 (0x002A)	SIN3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_22	P2_11	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN022 (0x002C)	SCK3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_21	P2_10	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN025 (0x0032)	MFS3_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN026 (0x0034)	SCS3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_23	P2_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN028 (0x0038)	SIN4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_18	P0_09	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN029 (0x003A)	SCK4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_17	P0_08	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN030 (0x003C)	SCL4	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN031 (0x003E)	SDA4	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN032 (0x0040)	MFS4_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN033 (0x0042)	SCS4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_19	P0_10	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN035 (0x0046)	SIN5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_00	P3_22	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN036 (0x0048)	SCK5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_31	P1_08	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN037 (0x004A)	SCL5	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN038 (0x004C)	SDA5	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN039 (0x004E)	MFS5_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN040 (0x0050)	SCS5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_01	P3_23	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN042 (0x0054)	SIN6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_04	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN043 (0x0056)	SCK6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_03	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN044 (0x0058)	SCL6	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN045 (0x005A)	SDA6	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN046 (0x005C)	MFS6_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN047 (0x005E)	SCS6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_30	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN049 (0x0062)	SIN7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_07	P0_02	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN050 (0x0064)	SCK7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_06	P0_01	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN051 (0x0066)	SCL7	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN052 (0x0068)	SDA7	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN053 (0x006A)	MFS7_TRIGGER	RESSEL (0-7)	TOT0	TOT1	TOT2	TOT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN054 (0x006C)	SCS7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_08	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN056 (0x0070)	SIN8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_27	P3_10	P0_21	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN057 (0x0072)	SCK8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_26	P3_09	P0_20	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN058 (0x0074)	SCL8	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN059 (0x0076)	SDA8	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN060 (0x0078)	MFS8_TRIGGER	RESSEL (0-7)	TOT16	TOT17	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN061 (0x007A)	SCS8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_28	-	P0_22	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN063 (0x007E)	SIN9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_20	P3_18	P0_15	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN064 (0x0080)	SCK9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_19	P3_17	P0_14	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN065 (0x0082)	SCL9	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN066 (0x0084)	SDA9	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN067 (0x0086)	MFS9_TRIGGER	RESSEL (0-7)	TOT16	TOT17	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN068 (0x0088)	SCS9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_16	P3_19	P0_16	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN070 (0x008C)	SIN10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_23	P3_13	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN071 (0x008E)	SCK10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P3_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN072 (0x0090)	SCL10	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN073 (0x0092)	SDA10	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN074 (0x0094)	MFS10_TRIGGER	RESSEL (0-7)	TOT16	TOT17	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN075 (0x0096)	SCS10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_24	P0_29	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN079 (0x009E)	SCL11	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN080 (0x00A0)	SDA11	RESSEL (0-7)	80ns noise filter disable	80ns noise filter enable	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN081 (0x00A2)	MFS11_TRIGGER	RESSEL (0-7)	TOT16	TOT17	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN210 (0x01A4)	RX0	RESSEL (0-7)	PORT_PIN	CAN0_RX_AND_TX	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_25	P3_09	P3_30	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN211 (0x01A6)	RX1	RESSEL (0-7)	PORT_PIN	CAN1_RX_AND_TX	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_27	P3_11	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN212 (0x01A8)	RX2	RESSEL (0-7)	PORT_PIN	CAN2_RX_AND_TX	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_00	P3_14	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN213 (0x01AA)	RX3	RESSEL (0-7)	PORT_PIN	CAN3_RX_AND_TX	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_05	P3_17	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN219 (0x01B6)	TIN0	RESSEL (0-7)	PORT_PIN	TOT3	RLT3_UFSET	TOT1	TIOA0	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_25	P3_08	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN220 (0x01B8)	TIN1	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT2	TIOA2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_27	P3_10	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN221 (0x01BA)	TIN2	RESSEL (0-7)	PORT_PIN	TOT1	RLT1_UFSET	TOT3	TIOA4	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P3_16	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN222 (0x01BC)	TIN3	RESSEL (0-7)	PORT_PIN	TOT2	RLT2_UFSET	TOT0	TIOA6	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_08	P3_18	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN235 (0x01D6)	TIN16	RESSEL (0-7)	PORT_PIN	TOT17	RLT17_UFSET	-	TIOA8	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_00	P3_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN236 (0x01D8)	TIN17	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	-	TIOA10	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_30	P0_29	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN251 (0x01F6)	INT0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_18	P2_23	P0_01	-	P3_16	P1_25	P2_19	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN252 (0x01F8)	INT1	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_25	P3_00	P0_02	P0_13	P3_14	-	P2_20	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN253 (0x01FA)	INT2	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_15	P2_02	-	P0_14	P3_18	P1_26	P2_21	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN254 (0x01FC)	INT3	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_22	P2_09	-	P0_16	P3_19	P1_27	P2_22	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN255 (0x01FE)	INT4	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_25	P2_24	-	P0_17	P3_21	P1_28	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN256 (0x0200)	INT5	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_26	P2_26	P0_03	P0_18	P3_22	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN257 (0x0202)	INT6	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_27	P3_09	P0_04	P0_19	P3_23	P1_30	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN258 (0x0204)	INT7	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_28	P3_30	P0_05	P0_20	-	P1_31	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN259 (0x0206)	INT8	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_30	P3_11	P0_06	P0_21	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN260 (0x0208)	INT9	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_31	P3_31	P3_01	P0_23	P1_09	P2_01	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN261 (0x020A)	INT10	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_00	-	P3_02	P0_24	-	P2_03	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN262 (0x020C)	INT11	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_01	P2_27	P3_03	-	P1_11	P2_04	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN263 (0x020E)	INT12	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_02	P0_29	P3_04	-	P1_13	P2_05	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN264 (0x0210)	INT13	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_04	P2_28	-	-	P1_15	P2_06	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN265 (0x0212)	INT14	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_05	P1_06	-	-	P1_16	P2_07	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN266 (0x0214)	INT15	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_07	P2_29	P3_05	-	P1_17	P2_08	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN267 (0x0216)	INT16	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_08	-	P3_06	-	-	P2_10	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN268 (0x0218)	INT17	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_10	P3_17	P3_07	P3_10	P1_18	P2_11	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN269 (0x021A)	INT18	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_12	P3_08	P0_07	P3_13	P1_19	P2_12	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN270 (0x021C)	INT19	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_14	P3_20	P0_08	P1_03	-	P2_13	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN271 (0x021E)	INT20	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_20	P3_15	P0_09	-	P1_21	P2_14	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN272 (0x0220)	INT21	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_23	P3_12	P0_10	-	P1_22	P2_15	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN273 (0x0222)	INT22	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_29	-	P0_11	-	P1_24	P2_16	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN274 (0x0224)	INT23	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P2_00	P0_00	P0_12	-	-	P2_17	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN283 (0x0236)	TEXT0	RESSEL (0-7)	PORT_PIN	TOT0	TOT1	TIOA1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN284 (0x0238)	TEXT1	RESSEL (0-7)	PORT_PIN	TOT0	TOT2	TIOA3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN285 (0x023A)	TEXT2	RESSEL (0-7)	PORT_PIN	TOT0	TOT3	TIOA5	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN286 (0x023C)	TEXT3	RESSEL (0-7)	PORT_PIN	TOT0	TOT1	TIOA7	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN287 (0x023E)	TEXT4	RESSEL (0-7)	PORT_PIN	TOT0	TOT2	TIOA9	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN291 (0x0246)	TEXT8	RESSEL (0-7)	PORT_PIN	RLT0_UFSET	RLT1_UFSET	TIOA17	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN292 (0x0248)	TEXT9	RESSEL (0-7)	PORT_PIN	RLT0_UFSET	RLT2_UFSET	TIOA19	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN293 (0x024A)	TEXT10	RESSEL (0-7)	PORT_PIN	RLT0_UFSET	RLT3_UFSET	TIOA21	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN299 (0x0256)	OCU0	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU1	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN300 (0x0258)	OCU0_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN301 (0x025A)	OCU1_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN302 (0x025C)	OCU2	RESSEL (0-7)	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU3	RESSEL (0-7)	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN303 (0x025E)	OCU2_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN304 (0x0260)	OCU3_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN305 (0x0262)	OCU4	RESSEL (0-7)	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0	FRT1
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU5	RESSEL (0-7)	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0	FRT1
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN306 (0x0264)	OCU4_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN307 (0x0266)	OCU5_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN323 (0x0286)	OCU16	RESSEL (0-7)	FRT8	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU17	RESSEL (0-7)	FRT8	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN324 (0x0288)	OCU16_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN325 (0x028A)	OCU17_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN326 (0x028C)	OCU18	RESSEL (0-7)	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU19	RESSEL (0-7)	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN327 (0x028E)	OCU18_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN328 (0x0290)	OCU19_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN329 (0x0292)	OCU20	RESSEL (0-7)	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	OCU21	RESSEL (0-7)	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN330 (0x0294)	OCU20_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN331 (0x0296)	OCU21_MOD	RESSEL (0-7)	set 1	set 0	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN347 (0x02B6)	IN0	RESSEL (0-7)	PORT_PIN	MFS0_LSYN	-	TOT0	TOT1	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_25	P3_08	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN348 (0x02B8)	IN1	RESSEL (0-7)	PORT_PIN	MFS1_LSYN	-	TOT0	TOT2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_26	P3_09	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN349 (0x02BA)	ICU0	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU1	RESSEL (0-7)	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN350 (0x02BC)	IN2	RESSEL (0-7)	PORT_PIN	MFS2_LSYN	-	TOT0	TOT3	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_27	P3_10	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN351 (0x02BE)	IN3	RESSEL (0-7)	PORT_PIN	MFS3_LSYN	-	TOT0	TOT1	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_28	P3_11	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN352 (0x02C0)	ICU2	RESSEL (0-7)	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU3	RESSEL (0-7)	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN353 (0x02C2)	IN4	RESSEL (0-7)	PORT_PIN	MFS4_LSYN	-	TOT0	TOT2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_00	P3_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN354 (0x02C4)	IN5	RESSEL (0-7)	PORT_PIN	MFS5_LSYN	-	TOT0	TOT3	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_01	P3_13	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN355 (0x02C6)	ICU4	RESSEL (0-7)	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0	FRT1
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU5	RESSEL (0-7)	FRT2	FRT3	FRT4	FRT8	FRT9	FRT10	FRT0	FRT1
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN371 (0x02E6)	IN16	RESSEL (0-7)	PORT_PIN	MFS6_LSYN	-	TOT0	TOT1	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_02	P0_29	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN372 (0x02E8)	IN17	RESSEL (0-7)	PORT_PIN	MFS7_LSYN	-	TOT0	TOT2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_03	P1_04	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN373 (0x02EA)	ICU16	RESSEL (0-7)	FRT8	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU17	RESSEL (0-7)	FRT8	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN374 (0x02EC)	IN18	RESSEL (0-7)	PORT_PIN	MFS8_LSYN	-	TOT0	TOT3	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_30	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN375 (0x02EE)	IN19	RESSEL (0-7)	PORT_PIN	MFS9_LSYN	-	TOT0	TOT1	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_31	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN376 (0x02F0)	ICU18	RESSEL (0-7)	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU19	RESSEL (0-7)	FRT9	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN377 (0x02F2)	IN20	RESSEL (0-7)	PORT_PIN	MFS10_LSYN	-	TOT0	TOT2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_07	P3_15	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN378 (0x02F4)	IN21	RESSEL (0-7)	PORT_PIN	MFS11_LSYN	-	TOT0	TOT3	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_08	P3_17	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN379 (0x02F6)	ICU20	RESSEL (0-7)	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
	ICU21	RESSEL (0-7)	FRT10	FRT0	FRT1	FRT2	FRT3	FRT4	FRT8	FRT9
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN419 (0x0346)	AIN8	RESSEL (0-7)	PORT_PIN	TOT0	TIOA0	TIOA1	TIOA2	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN420 (0x0348)	BIN8	RESSEL (0-7)	PORT_PIN	TOT1	TIOA1	TIOA2	TIOA0	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN421 (0x034A)	ZIN8	RESSEL (0-7)	PORT_PIN	TOT2	TIOA2	TIOA0	TIOA1	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN422 (0x034C)	AIN9	RESSEL (0-7)	PORT_PIN	TOT1	TIOA3	TIOA4	TIOA5	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN423 (0x034E)	BIN9	RESSEL (0-7)	PORT_PIN	TOT2	TIOA4	TIOA5	TIOA3	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN424 (0x0350)	ZIN9	RESSEL (0-7)	PORT_PIN	TOT3	TIOA5	TIOA3	TIOA4	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN443 (0x0376)	TIOB0	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT1	RLT1_UFSET	FRT0_MTSF	OUT0	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN444 (0x0378)	TIOA1	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN445 (0x037A)	TIOB1	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN446 (0x037C)	TIOB2	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT2	RLT2_UFSET	FRT1_MTSF	OUT1	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN447 (0x037E)	TIOA3	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN448 (0x0380)	TIOB3	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN449 (0x0382)	TIOB4	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT3	RLT3_UFSET	FRT2_MTSF	OUT2	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN450 (0x0384)	TIOA5	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN451 (0x0386)	TIOB5	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN452 (0x0388)	TIOB6	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT16	RLT16_UFSET	FRT3_MTSF	OUT3	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN453 (0x038A)	TIOA7	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN454 (0x038C)	TIOB7	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN455 (0x038E)	TIOB8	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT17	RLT17_UFSET	FRT4_MTSF	OUT4	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN456 (0x0390)	TIOA9	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN457 (0x0392)	TIOB9	RESSEL (0-7)	set 0							
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN458 (0x0394)	TIOB10	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT0	RLT0_UFSET	FRT0_MTSF	OUT5	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN459 (0x0396)	TIOA11	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN460 (0x0398)	TIOB11	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN461 (0x039A)	TIOB12	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT1	RLT1_UFSET	FRT8_MTSF	OUT0	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN462 (0x039C)	TIOA13	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN463 (0x039E)	TIOB13	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN464 (0x03A0)	TIOB14	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT2	RLT2_UFSET	FRT9_MTSF	OUT1	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN465 (0x03A2)	TIOA15	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN466 (0x03A4)	TIOB15	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN467 (0x03A6)	TIOB16	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT3	RLT3_UFSET	FRT10_MTSF	OUT16	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN468 (0x03A8)	TIOA17	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN469 (0x03AA)	TIOB17	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN470 (0x03AC)	TIOB18	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT17	RLT17_UFSET	FRT8_MTSF	OUT17	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN471 (0x03AE)	TIOA19	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN472 (0x03B0)	TIOB19	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN473 (0x03B2)	TIOB20	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT1	RLT1_UFSET	FRT9_MTSF	OUT18	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN474 (0x03B4)	TIOA21	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN475 (0x03B6)	TIOB21	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN476 (0x03B8)	TIOB22	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT2	RLT2_UFSET	FRT10_MTSF	OUT19	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_22	P1_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN477 (0x03BA)	TIOA23	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN478 (0x03BC)	TIOB23	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN479 (0x03BE)	TIOB24	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT3	RLT3_UFSET	FRT0_MTSF	OUT20	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_31	P2_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN480 (0x03C0)	TIOA25	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN481 (0x03C2)	TIOB25	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN482 (0x03C4)	TIOB26	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT16	RLT16_UFSET	FRT1_MTSF	OUT21	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_31	P2_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN483 (0x03C6)	TIOA27	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN484 (0x03C8)	TIOB27	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN485 (0x03CA)	TIOB28	RESSEL (0-7)	PORT_PIN	TOT0	RLT0_UFSET	TOT17	RLT17_UFSET	FRT2_MTSF	OUT0	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_31	P2_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN486 (0x03CC)	TIOA29	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN487 (0x03CE)	TIOB29	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN488 (0x03D0)	TIOB30	RESSEL (0-7)	PORT_PIN	TOT16	RLT16_UFSET	TOT0	RLT0_UFSET	FRT3_MTSF	OUT1	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_31	P2_12	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN489 (0x03D2)	TIOA31	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN490 (0x03D4)	TIOB31	RESSEL (0-7)	set 0	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN539 (0x0436)	ADC0_TRG0	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P1_09	P1_08	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN541 (0x043A)	ADC0_HWTRG0	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT0	OUT1	bt_adto_000_a	bt_adto_000_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN542 (0x043C)	bt_adto_000_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN543 (0x043E)	bt_adto_000_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN546 (0x0444)	ADC0_HWTRG1	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT1	OUT2	bt_adto_001_a	bt_adto_001_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN547 (0x0446)	bt_adto_001_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN548 (0x0448)	bt_adto_001_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN551 (0x044E)	ADC0_HWTRG2	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT2	OUT3	bt_adto_002_a	bt_adto_002_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN552 (0x0450)	bt_adto_002_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN553 (0x0452)	bt_adto_002_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN556 (0x0458)	ADC0_HWTRG3	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT3	OUT4	bt_adto_003_a	bt_adto_003_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN557 (0x045A)	bt_adto_003_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN558 (0x045C)	bt_adto_003_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN561 (0x0462)	ADC0_HWTRG4	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT4	OUT5	bt_adto_004_a	bt_adto_004_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN562 (0x0464)	bt_adto_004_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN563 (0x0466)	bt_adto_004_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN566 (0x046C)	ADC0_HWTRG5	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT5	OUT16	bt_adto_005_a	bt_adto_005_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN567 (0x046E)	bt_adto_005_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN568 (0x0470)	bt_adto_005_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN571 (0x0476)	ADC0_HWTRG6	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT16	OUT17	bt_adto_006_a	bt_adto_006_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN572 (0x0478)	bt_adto_006_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN573 (0x047A)	bt_adto_006_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN576 (0x0480)	ADC0_HWTRG7	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT17	OUT18	bt_adto_007_a	bt_adto_007_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN577 (0x0482)	bt_adto_007_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN578 (0x0484)	bt_adto_007_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN581 (0x048A)	ADC0_HWTRG8	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT18	OUT19	bt_adto_008_a	bt_adto_008_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN582 (0x048C)	bt_adto_008_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN583 (0x048E)	bt_adto_008_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN586 (0x0494)	ADC0_HWTRG9	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT19	OUT20	bt_adto_009_a	bt_adto_009_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN587 (0x0496)	bt_adto_009_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN588 (0x0498)	bt_adto_009_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN591 (0x049E)	ADC0_HWTRG10	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT20	OUT21	bt_adto_010_a	bt_adto_010_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN592 (0x04A0)	bt_adto_010_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN593 (0x04A2)	bt_adto_010_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN596 (0x04A8)	ADC0_HWTRG11	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT21	OUT0	bt_adto_011_a	bt_adto_011_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN597 (0x04AA)	bt_adto_011_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN598 (0x04AC)	bt_adto_011_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN601 (0x04B2)	ADC0_HWTRG12	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT0	OUT1	bt_adto_012_a	bt_adto_012_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN602 (0x04B4)	bt_adto_012_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN603 (0x04B6)	bt_adto_012_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN606 (0x04BC)	ADC0_HWTRG13	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT1	OUT2	bt_adto_013_a	bt_adto_013_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN607 (0x04BE)	bt_adto_013_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN608 (0x04C0)	bt_adto_013_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN611 (0x04C6)	ADC0_HWTRG14	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT2	OUT3	bt_adto_014_a	bt_adto_014_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN612 (0x04C8)	bt_adto_014_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN613 (0x04CA)	bt_adto_014_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN616 (0x04D0)	ADC0_HWTRG15	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT3	OUT4	bt_adto_015_a	bt_adto_015_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN617 (0x04D2)	bt_adto_015_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN618 (0x04D4)	bt_adto_015_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN621 (0x04DA)	ADC0_HWTRG16	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT4	OUT5	bt_adto_016_a	bt_adto_016_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN622 (0x04DC)	bt_adto_016_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN623 (0x04DE)	bt_adto_016_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN626 (0x04E4)	ADC0_HWTRG17	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT5	OUT16	bt_adto_017_a	bt_adto_017_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN627 (0x04E6)	bt_adto_017_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN628 (0x04E8)	bt_adto_017_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN631 (0x04EE)	ADC0_HWTRG18	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT16	OUT17	bt_adto_018_a	bt_adto_018_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN632 (0x04F0)	bt_adto_018_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN633 (0x04F2)	bt_adto_018_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN636 (0x04F8)	ADC0_HWTRG19	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT17	OUT18	bt_adto_019_a	bt_adto_019_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN637 (0x04FA)	bt_adto_019_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN638 (0x04FC)	bt_adto_019_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN641 (0x0502)	ADC0_HWTRG20	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT18	OUT19	bt_adto_020_a	bt_adto_020_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN642 (0x0504)	bt_adto_020_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN643 (0x0506)	bt_adto_020_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN646 (0x050C)	ADC0_HWTRG21	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT19	OUT20	bt_adto_021_a	bt_adto_021_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN647 (0x050E)	bt_adto_021_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN648 (0x0510)	bt_adto_021_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN651 (0x0516)	ADC0_HWTRG22	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT20	OUT21	bt_adto_022_a	bt_adto_022_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN652 (0x0518)	bt_adto_022_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN653 (0x051A)	bt_adto_022_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN656 (0x0520)	ADC0_HWTRG23	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT21	OUT0	bt_adto_023_a	bt_adto_023_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN657 (0x0522)	bt_adto_023_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN658 (0x0524)	bt_adto_023_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN661 (0x052A)	ADC0_HWTRG24	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT0	OUT1	bt_adto_024_a	bt_adto_024_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN662 (0x052C)	bt_adto_024_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN663 (0x052E)	bt_adto_024_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN666 (0x0534)	ADC0_HWTRG25	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT1	OUT2	bt_adto_025_a	bt_adto_025_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN667 (0x0536)	bt_adto_025_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN668 (0x0538)	bt_adto_025_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN671 (0x053E)	ADC0_HWTRG26	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT2	OUT3	bt_adto_026_a	bt_adto_026_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN672 (0x0540)	bt_adto_026_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN673 (0x0542)	bt_adto_026_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN676 (0x0548)	ADC0_HWTRG27	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT3	OUT4	bt_adto_027_a	bt_adto_027_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN677 (0x054A)	bt_adto_027_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN678 (0x054C)	bt_adto_027_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN681 (0x0552)	ADC0_HWTRG28	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT4	OUT5	bt_adto_028_a	bt_adto_028_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN682 (0x0554)	bt_adto_028_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN683 (0x0556)	bt_adto_028_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN686 (0x055C)	ADC0_HWTRG29	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT5	OUT16	bt_adto_029_a	bt_adto_029_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN687 (0x055E)	bt_adto_029_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN688 (0x0560)	bt_adto_029_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN691 (0x0566)	ADC0_HWTRG30	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT16	OUT17	bt_adto_030_a	bt_adto_030_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN692 (0x0568)	bt_adto_030_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN693 (0x056A)	bt_adto_030_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN696 (0x0570)	ADC0_HWTRG31	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT17	OUT18	bt_adto_031_a	bt_adto_031_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN697 (0x0572)	bt_adto_031_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN698 (0x0574)	bt_adto_031_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN701 (0x057A)	ADC0_HWTRG32	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT18	OUT19	bt_adto_032_a	bt_adto_032_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN702 (0x057C)	bt_adto_032_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN703 (0x057E)	bt_adto_032_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN706 (0x0584)	ADC0_HWTRG33	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT19	OUT20	bt_adto_033_a	bt_adto_033_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN707 (0x0586)	bt_adto_033_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN708 (0x0588)	bt_adto_033_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN711 (0x058E)	ADC0_HWTRG34	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT20	OUT21	bt_adto_034_a	bt_adto_034_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN712 (0x0590)	bt_adto_034_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN713 (0x0592)	bt_adto_034_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN716 (0x0598)	ADC0_HWTRG35	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT21	OUT0	bt_adto_035_a	bt_adto_035_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN717 (0x059A)	bt_adto_035_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN718 (0x059C)	bt_adto_035_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN721 (0x05A2)	ADC0_HWTRG36	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT0	OUT1	bt_adto_036_a	bt_adto_036_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN722 (0x05A4)	bt_adto_036_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN723 (0x05A6)	bt_adto_036_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN726 (0x05AC)	ADC0_HWTRG37	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT1	OUT2	bt_adto_037_a	bt_adto_037_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN727 (0x05AE)	bt_adto_037_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN728 (0x05B0)	bt_adto_037_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN731 (0x05B6)	ADC0_HWTRG38	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT2	OUT3	bt_adto_038_a	bt_adto_038_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN732 (0x05B8)	bt_adto_038_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN733 (0x05BA)	bt_adto_038_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN736 (0x05C0)	ADC0_HWTRG39	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT3	OUT4	bt_adto_039_a	bt_adto_039_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN737 (0x05C2)	bt_adto_039_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN738 (0x05C4)	bt_adto_039_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN741 (0x05CA)	ADC0_HWTRG40	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT4	OUT5	bt_adto_040_a	bt_adto_040_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN742 (0x05CC)	bt_adto_040_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN743 (0x05CE)	bt_adto_040_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN746 (0x05D4)	ADC0_HWTRG41	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT5	OUT16	bt_adto_041_a	bt_adto_041_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN747 (0x05D6)	bt_adto_041_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN748 (0x05D8)	bt_adto_041_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN751 (0x05DE)	ADC0_HWTRG42	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT0_UFSET	RLT1_UFSET	OUT16	OUT17	bt_adto_042_a	bt_adto_042_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN752 (0x05E0)	bt_adto_042_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN753 (0x05E2)	bt_adto_042_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN756 (0x05E8)	ADC0_HWTRG43	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT1_UFSET	RLT2_UFSET	OUT17	OUT18	bt_adto_043_a	bt_adto_043_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN757 (0x05EA)	bt_adto_043_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN758 (0x05EC)	bt_adto_043_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN761 (0x05F2)	ADC0_HWTRG44	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT2_UFSET	RLT3_UFSET	OUT18	OUT19	bt_adto_044_a	bt_adto_044_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN762 (0x05F4)	bt_adto_044_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN763 (0x05F6)	bt_adto_044_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN766 (0x05FC)	ADC0_HWTRG45	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT3_UFSET	RLT16_UFSET	OUT19	OUT20	bt_adto_045_a	bt_adto_045_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN767 (0x05FE)	bt_adto_045_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN768 (0x0600)	bt_adto_045_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN771 (0x0606)	ADC0_HWTRG46	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT16_UFSET	RLT17_UFSET	OUT20	OUT21	bt_adto_046_a	bt_adto_046_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN772 (0x0608)	bt_adto_046_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN773 (0x060A)	bt_adto_046_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN776 (0x0610)	ADC0_HWTRG47	RESSEL (0-7)	PORT_PIN(ADC0_TRG0)	RLT17_UFSET	RLT0_UFSET	OUT21	OUT0	bt_adto_047_a	bt_adto_047_b	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN777 (0x0612)	bt_adto_047_a	RESSEL (0-7)	BT0_ADTO	BT1_ADTO	BT2_ADTO	BT3_ADTO	BT4_ADTO	BT5_ADTO	BT6_ADTO	BT7_ADTO
		RESSEL (8-15)	BT8_ADTO	BT9_ADTO	BT10_ADTO	BT11_ADTO	BT12_ADTO	BT13_ADTO	BT14_ADTO	BT15_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN778 (0x0614)	bt_adto_047_b	RESSEL (0-7)	BT16_ADTO	BT17_ADTO	BT18_ADTO	BT19_ADTO	BT20_ADTO	BT21_ADTO	BT22_ADTO	BT23_ADTO
		RESSEL (8-15)	BT24_ADTO	BT25_ADTO	BT26_ADTO	BT27_ADTO	BT28_ADTO	BT29_ADTO	BT30_ADTO	BT31_ADTO
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1189 (0x094A)	DDRHSSPI_MSTART	RESSEL (0-7)	-	TOT0	TOT16	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1192 (0x0950)	I2S0_WS	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_05	P2_06	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1193 (0x0952)	I2S0_SD	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_04	P2_05	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1194 (0x0954)	I2S0_SCK	RESSEL (0-7)	-	-	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_06	P2_07	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1195 (0x0956)	I2S0_ECLK	RESSEL (0-7)	PORT_PIN	SYSC1_CLK_CD2	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	P0_03	P2_04	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1199 (0x095E)	I2S1_ECLK	RESSEL (0-7)	PORT_PIN	SYSC1_CLK_CD2	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1200 (0x095E)	SOUND_RST_N	RESSEL (0-7)	set 1 (Negate reset)	set 0 (Assert reset)	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1211 (0x0976)	DMA[10]	RESSEL (0-7)	Ext_IRQ_0	Ext_IRQ_8	Ext_IRQ_16	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1212 (0x0978)	DMA[11]	RESSEL (0-7)	Ext_IRQ_1	Ext_IRQ_9	Ext_IRQ_17	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1213 (0x097A)	DMA[12]	RESSEL (0-7)	Ext_IRQ_2	Ext_IRQ_10	Ext_IRQ_18	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1214 (0x097C)	DMA[13]	RESSEL (0-7)	Ext_IRQ_3	Ext_IRQ_11	Ext_IRQ_19	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1215 (0x097E)	DMA[14]	RESSEL (0-7)	Ext_IRQ_4	Ext_IRQ_12	Ext_IRQ_20	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1216 (0x0980)	DMA[15]	RESSEL (0-7)	Ext_IRQ_5	Ext_IRQ_13	Ext_IRQ_21	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1217 (0x0982)	DMA[16]	RESSEL (0-7)	Ext_IRQ_6	Ext_IRQ_14	Ext_IRQ_22	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1218 (0x0984)	DMA[17]	RESSEL (0-7)	Ext_IRQ_7	Ext_IRQ_15	Ext_IRQ_23	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1219 (0x0986)	DMA[56]	RESSEL (0-7)	BT0_IRQ-0	BT0_IRQ-1	BT24_IRQ-0	BT24_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1220 (0x0988)	DMA[57]	RESSEL (0-7)	BT1_IRQ-0	BT1_IRQ-1	BT25_IRQ-0	BT25_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1221 (0x098A)	DMA[58]	RESSEL (0-7)	BT2_IRQ-0	BT2_IRQ-1	BT26_IRQ-0	BT26_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1222 (0x098C)	DMA[59]	RESSEL (0-7)	BT3_IRQ-0	BT3_IRQ-1	BT27_IRQ-0	BT27_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1223 (0x098E)	DMA[60]	RESSEL (0-7)	BT4_IRQ-0	BT4_IRQ-1	BT28_IRQ-0	BT28_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1224 (0x0990)	DMA[61]	RESSEL (0-7)	BT5_IRQ-0	BT5_IRQ-1	BT29_IRQ-0	BT29_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1225 (0x0992)	DMA[62]	RESSEL (0-7)	BT6_IRQ-0	BT6_IRQ-1	BT30_IRQ-0	BT30_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1226 (0x0994)	DMA[63]	RESSEL (0-7)	BT7_IRQ-0	BT7_IRQ-1	BT31_IRQ-0	BT31_IRQ-1	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1227 (0x0996)	DMA[64]	RESSEL (0-7)	BT8_IRQ-0	BT8_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1228 (0x0998)	DMA[65]	RESSEL (0-7)	BT9_IRQ-0	BT9_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1229 (0x099A)	DMA[66]	RESSEL (0-7)	BT10_IRQ-0	BT10_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1230 (0x099C)	DMA[67]	RESSEL (0-7)	BT11_IRQ-0	BT11_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1231 (0x099E)	DMA[68]	RESSEL (0-7)	BT12_IRQ-0	BT12_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1232 (0x09A0)	DMA[69]	RESSEL (0-7)	BT13_IRQ-0	BT13_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1233 (0x09A2)	DMA[70]	RESSEL (0-7)	BT14_IRQ-0	BT14_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1234 (0x09A4)	DMA[71]	RESSEL (0-7)	BT15_IRQ-0	BT15_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1235 (0x09A6)	DMA[72]	RESSEL (0-7)	BT16_IRQ-0	BT16_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1236 (0x09A8)	DMA[73]	RESSEL (0-7)	BT17_IRQ-0	BT17_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1237 (0x09AA)	DMA[74]	RESSEL (0-7)	BT18_IRQ-0	BT18_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1238 (0x09AC)	DMA[75]	RESSEL (0-7)	BT19_IRQ-0	BT19_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1239 (0x09AE)	DMA[76]	RESSEL (0-7)	BT20_IRQ-0	BT20_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1240 (0x09B0)	DMA[77]	RESSEL (0-7)	BT21_IRQ-0	BT21_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1241 (0x09B2)	DMA[78]	RESSEL (0-7)	BT22_IRQ-0	BT22_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1242 (0x09B4)	DMA[79]	RESSEL (0-7)	BT23_IRQ-0	BT23_IRQ-1	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1243 (0x09B6)	DMA[80]	RESSEL (0-7)	FRT0_Match	FRT0_Zero	FRT2_Match	FRT2_Zero	FRT4_Match	FRT4_Zero	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1244 (0x09B8)	DMA[81]	RESSEL (0-7)	FRT1_Match	FRT1_Zero	FRT3_Match	FRT3_Zero	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1245 (0x09BA)	DMA[82]	RESSEL (0-7)	FRT8_Match	FRT8_Zero	FRT10_Match	FRT10_Zero	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1246 (0x09BC)	DMA[83]	RESSEL (0-7)	FRT9_Match	FRT9_Zero	-	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1247 (0x09BE)	DMA[84]	RESSEL (0-7)	ICU0_IRQ0	ICU2_IRQ0	ICU4_IRQ0	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1248 (0x09C0)	DMA[85]	RESSEL (0-7)	ICU16_IRQ0	ICU18_IRQ0	ICU20_IRQ0	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1249 (0x09C2)	DMA[86]	RESSEL (0-7)	ICU1_IRQ1	ICU3_IRQ1	ICU5_IRQ1	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1250 (0x09C4)	DMA[87]	RESSEL (0-7)	ICU17_IRQ1	ICU19_IRQ1	ICU21_IRQ1	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1251 (0x09C6)	DMA[88]	RESSEL (0-7)	OCU0_IRQ0	OCU2_IRQ0	OCU4_IRQ0	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1252 (0x09C8)	DMA[89]	RESSEL (0-7)	OCU16_IRQ0	OCU18_IRQ0	OCU20_IRQ0	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Register (Offset)	Resource	RESSEL[3:0] / PORTSEL[3:0]	Source for Resource Input							
			0	1	2	3	4	5	6	7
			8	9	10	11	12	13	14	15
RIC_RESIN1253 (0x09CA)	DMA[90]	RESSEL (0-7)	OCU1_IRQ1	OCU3_IRQ1	OCU5_IRQ1	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1254 (0x09CC)	DMA[91]	RESSEL (0-7)	OCU17_IRQ1	OCU19_IRQ1	OCU21_IRQ1	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1255 (0x09CE)	DMA[92]	RESSEL (0-7)	RLT0_IRQ	RLT2_IRQ	RLT16_IRQ	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1256 (0x09D0)	DMA[93]	RESSEL (0-7)	RLT1_IRQ	RLT3_IRQ	RLT17_IRQ	-	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-
RIC_RESIN1257 (0x09D2)	DMA[94]	RESSEL (0-7)	DMAC_RLT0	DMAC_RLT1	DMAC_RLT2	DMAC_RLT3	-	-	-	-
		RESSEL (8-15)	-	-	-	-	-	-	-	-
		PORTSEL (0-7)	-	-	-	-	-	-	-	-
		PORTSEL (8-15)	-	-	-	-	-	-	-	-

Notes:

- *If BTx_ADTO is used by ADCx_HWTRGx, set by the two RIC_RESIN (Resource Input Setting Register).*
  Select the bt_adto_xxx_x by ADCx_HWTRGx and then, select the BTx_ADTO by bt_adto_xxx_x.
- *When both GPIO_PORTEN.GPORTEN and PPC_PCFGR.PIE are configured as 0, the input signal is disconnected and external interrupt cannot be detected. During disconnecting, I/O internally outputs "low" to internal logic, and if ELVR is configured as low-level-detection, falling-edge-detection, or both-edge-detection it will be detected as external interrupt with EIRR=1.*
- *OCUx_MODn is described as MODn pin in Traveo™ Platform hardware manual.*

3.2. Port Output Function Configuration

The port output function configuration (POF) is a function to select a function to output to a port.

A resource which supports a port output relocation has its PPC_PCFGR.POF to configure resource output.

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR000 (0x0000)	P0_00	GPIO_PODR0:POD00	-	LCDD12	MFS7_SOT_1	I2S1_SD_0	-	BT0_TIOA1_1	MAD8
PPC_PCFGR001 (0x0002)	P0_01	GPIO_PODR0:POD01	-	LCDD13	MFS7_SCK_1	I2S1_WS_0	-	BT1_TIOA2_1	MAD9
PPC_PCFGR002 (0x0004)	P0_02	GPIO_PODR0:POD02	-	LCDD14	-	I2S1_SCK_0	-	BT1_TIOA3_1	MAD10
PPC_PCFGR003 (0x0006)	P0_03	GPIO_PODR0:POD03	MFS1_SOT_1	LCDD15	-	-	PCMP0_BL_1	BT2_TIOA4_1	MAD11
PPC_PCFGR004 (0x0008)	P0_04	GPIO_PODR0:POD04	MFS1_SCK_1	LCDD16	-	I2S0_SD_0	PCMP0_BH_1	BT2_TIOA5_1	MAD12
PPC_PCFGR005 (0x000A)	P0_05	GPIO_PODR0:POD05	-	LCDD17	-	I2S0_WS_0	PCMP0_AL_1	BT3_TIOA6_1	MAD13
PPC_PCFGR006 (0x000C)	P0_06	GPIO_PODR0:POD06	MFS1_CS0_1	CS#	-	I2S0_SCK_0	PCMP0_AH_1	BT3_TIOA7_1	MAD14
PPC_PCFGR007 (0x000E)	P0_07	GPIO_PODR0:POD07	-	WR#	-	-	-	MFS4_SOT_1	MOEX
PPC_PCFGR008 (0x0010)	P0_08	GPIO_PODR0:POD08	-	RD#	-	-	-	MFS4_SCK_1	MWEX
PPC_PCFGR009 (0x0012)	P0_09	GPIO_PODR0:POD09	-	-	-	-	PCMP1_BL_1	-	MCLK
PPC_PCFGR010 (0x0014)	P0_10	GPIO_PODR0:POD10	-	RS	-	-	PCMP1_BH_1	MFS4_CS0_1	MDQM0
PPC_PCFGR011 (0x0016)	P0_11	GPIO_PODR0:POD11	-	RES#	-	-	PCMP1_AL_1	MFS4_CS1_1	MCSX2
PPC_PCFGR012 (0x0018)	P0_12	GPIO_PODR0:POD12	-	-	-	-	PCMP1_AH_1	MFS4_CS2_1	MCSX3

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR013 (0x001A)	P0_13	GPIO_PODR0:POD13	MFS2_SOT_0	DDRHSSPI1_SDATA0	-	-	MFS9_SOT_2	-	-
PPC_PCFGR014 (0x001C)	P0_14	GPIO_PODR0:POD14	MFS2_SCK_0	DDRHSSPI1_SDATA2	-	-	MFS9_SCK_2	-	-
PPC_PCFGR015 (0x001E)	P0_15	GPIO_PODR0:POD15	-	DDRHSSPI1_SDATA1	-	-	-	-	-
PPC_PCFGR016 (0x0020)	P0_16	GPIO_PODR0:POD16	MFS2_CS0_0	DDRHSSPI1_SSEL	-	-	MFS9_CS0_2	-	-
PPC_PCFGR017 (0x0022)	P0_17	GPIO_PODR0:POD17	MFS2_CS1_0	DDRHSSPI1_SDATA3	-	-	MFS9_CS1_2	-	-
PPC_PCFGR018 (0x0024)	P0_18	GPIO_PODR0:POD18	MFS2_CS2_0	DDRHSSPI0_SCLK	-	-	-	-	-
PPC_PCFGR019 (0x0026)	P0_19	GPIO_PODR0:POD19	MFS2_CS3_0	DDRHSSPI0_SDATA0	-	-	MFS8_SOT_2	-	-
PPC_PCFGR020 (0x0028)	P0_20	GPIO_PODR0:POD20	MFS3_SOT_0	DDRHSSPI0_SDATA2	-	-	MFS8_SCK_2	-	-
PPC_PCFGR021 (0x002A)	P0_21	GPIO_PODR0:POD21	MFS3_SCK_0	DDRHSSPI0_SDATA1	-	-	-	-	-
PPC_PCFGR022 (0x002C)	P0_22	GPIO_PODR0:POD22	-	DDRHSSPI0_SSEL	-	-	MFS8_CS0_2	-	-
PPC_PCFGR023 (0x002E)	P0_23	GPIO_PODR0:POD23	MFS3_CS0_0	DDRHSSPI0_SDATA3	-	-	-	-	-
PPC_PCFGR024 (0x0030)	P0_24	GPIO_PODR0:POD24	MFS3_CS1_0	-	-	-	-	-	-
PPC_PCFGR025 (0x0032)	P0_25	GPIO_PODR0:POD25	MFS8_SOT_0	-	MFS0_SDA	OCU0_OUT0_0	-	BT0_TIOA0_0	PCMP0_BL_0
PPC_PCFGR026 (0x0034)	P0_26	GPIO_PODR0:POD26	MFS8_SCK_0	CAN0_TX_0	MFS0_SCL	OCU0_OUT1_0	RLT0_TOT_0	BT0_TIOA1_0	PCMP0_BH_0
PPC_PCFGR027 (0x0036)	P0_27	GPIO_PODR0:POD27	-	SG0_SGA_0	-	-	-	BT1_TIOA2_0	PCMP0_AL_0

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFG R028 (0x0038)	P0_28	GPIO_POD R0:POD28	MFS8_CS0_0	SG0_SGO_0	CAN1_TX_0	PWUTRG_0	RLT1_TOT_0	BT1_TIOA3_0	PCMP0_AH_0
PPC_PCFG R029 (0x003A)	P0_29	GPIO_POD R0:POD29	MFS10_CS0_1	-	BT9_TIOA18_1	OCU8_OUT0_1	-	-	-
PPC_PCFG R030 (0x003C)	P0_30	GPIO_POD R0:POD30	MFS6_CS0_0	-	-	OCU8_OUT0_0	-	BT4_TIOA8_0	-
PPC_PCFG R031 (0x003E)	P0_31	GPIO_POD R0:POD31	MFS6_CS1_0	-	-	OCU8_OUT1_0	RLT17_TOT_0	BT4_TIOA9_0	-
PPC_PCFG R100 (0x0040)	P1_00	GPIO_POD R1:POD00	-	SG1_SGA_0	IND0_OUT_0	OCU1_OUT0_0	TRACE0_0	BT2_TIOA4_0	PCMP1_BL_0
PPC_PCFG R101 (0x0042)	P1_01	GPIO_POD R1:POD01	TRACE1_0	SG1_SGO_0	CAN2_TX_0	OCU1_OUT1_0	RLT16_TOT_0	BT2_TIOA5_0	PCMP1_BH_0
PPC_PCFG R102 (0x0044)	P1_02	GPIO_POD R1:POD02	MFS6_SOT_0	SG2_SGA_0	MFS6_SDA	OCU2_OUT0_0	TRACE2_0	BT3_TIOA6_0	PCMP1_AL_0
PPC_PCFG R103 (0x0046)	P1_03	GPIO_POD R1:POD03	MFS6_SCK_0	SG2_SGO_0	MFS6_SCL	OCU2_OUT1_0	TRACE3_0	BT3_TIOA7_0	PCMP1_AH_0
PPC_PCFG R104 (0x0048)	P1_04	GPIO_POD R1:POD04	-	SG3_SGA_0	BT9_TIOA19_1	OCU8_OUT1_1	RLT17_TOT_1	TRACE_CLK_0	-
PPC_PCFG R105 (0x004A)	P1_05	GPIO_POD R1:POD05	MFS7_SOT_0	SG3_SGO_0	-	OCU9_OUT0_1	TRACE_CTL_0	MFS7_SDA	-
PPC_PCFG R106 (0x004C)	P1_06	GPIO_POD R1:POD06	MFS7_SCK_0	SG4_SGA_0	CAN3_TX_0	OCU9_OUT1_1	RLT2_TOT_1	MFS7_SCL	-
PPC_PCFG R107 (0x004E)	P1_07	GPIO_POD R1:POD07	SYSC0_CLK_1	SG4_SGO_0	-	OCU9_OUT0_0	RLT2_TOT_0	BT5_TIOA10_0	-
PPC_PCFG R108 (0x0050)	P1_08	GPIO_POD R1:POD08	MFS7_CS0_0	-	-	OCU10_OUT0_0	MFS5_SCK_1	BT5_TIOA11_0	IND0_OUT_1
PPC_PCFG R109 (0x0052)	P1_09	GPIO_POD R1:POD09	MFS7_CS1_0	-	-	OCU10_OUT1_0	RLT3_TOT_0	SYSC0_CLK_0	RTC0_WOT
PPC_PCFG R110 (0x0054)	P1_10	GPIO_POD R1:POD10	-	-	-	-	SMC0_PWM1P	BT6_TIOA12_0	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR111 (0x0056)	P1_11	GPIO_PODR1:POD11	-	-	-	-	SMC0_PWM1M	BT6_TIOA13_0	-
PPC_PCFGR112 (0x0058)	P1_12	GPIO_PODR1:POD12	MFS0_SOT_0	-	MFS8_SDA	-	SMC0_PWM2P	BT7_TIOA14_0	-
PPC_PCFGR113 (0x005A)	P1_13	GPIO_PODR1:POD13	MFS0_SCK_0	-	MFS8_SCL	-	SMC0_PWM2M	BT7_TIOA15_0	-
PPC_PCFGR114 (0x005C)	P1_14	GPIO_PODR1:POD14	-	-	-	-	SMC1_PWM1P	BT8_TIOA16_0	-
PPC_PCFGR115 (0x005E)	P1_15	GPIO_PODR1:POD15	MFS0_CS0_0	-	-	-	SMC1_PWM1M	BT8_TIOA17_0	-
PPC_PCFGR116 (0x0060)	P1_16	GPIO_PODR1:POD16	MFS9_CS0_0	-	-	-	SMC1_PWM2P	BT9_TIOA18_0	-
PPC_PCFGR117 (0x0062)	P1_17	GPIO_PODR1:POD17	MFS9_CS1_0	-	-	-	SMC1_PWM2M	BT9_TIOA19_0	-
PPC_PCFGR118 (0x0064)	P1_18	GPIO_PODR1:POD18	MFS9_SOT_0	-	MFS9_SDA	-	SMC2_PWM1P	BT10_TIOA20_0	-
PPC_PCFGR119 (0x0066)	P1_19	GPIO_PODR1:POD19	MFS9_SCK_0	-	MFS9_SCL	-	SMC2_PWM1M	BT10_TIOA21_0	-
PPC_PCFGR120 (0x0068)	P1_20	GPIO_PODR1:POD20	-	-	-	-	SMC2_PWM2P	BT11_TIOA22_0	-
PPC_PCFGR121 (0x006A)	P1_21	GPIO_PODR1:POD21	MFS10_SOT_0	-	MFS10_SDA	-	SMC2_PWM2M	BT11_TIOA23_0	-
PPC_PCFGR122 (0x006C)	P1_22	GPIO_PODR1:POD22	MFS10_SCK_0	-	MFS10_SCL	-	SMC3_PWM1P	-	-
PPC_PCFGR123 (0x006E)	P1_23	GPIO_PODR1:POD23	-	-	-	-	SMC3_PWM1M	BT12_TIOA24_0	-
PPC_PCFGR124 (0x0070)	P1_24	GPIO_PODR1:POD24	MFS10_CS0_0	-	-	-	SMC3_PWM2P	BT12_TIOA25_0	-
PPC_PCFGR125 (0x0072)	P1_25	GPIO_PODR1:POD25	MFS11_CS0_0	-	-	-	SMC3_PWM2M	BT13_TIOA26_0	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR126 (0x0074)	P1_26	GPIO_POD R1:POD26	MFS11_CS1_0	-	-	-	SMC4_PWM1P	BT13_TIOA27_0	-
PPC_PCFGR127 (0x0076)	P1_27	GPIO_POD R1:POD27	MFS11_SOT_0	-	MFS11_SDA	-	SMC4_PWM1M	BT14_TIOA28_0	-
PPC_PCFGR128 (0x0078)	P1_28	GPIO_POD R1:POD28	MFS11_SCK_0	-	MFS11_SCL	-	SMC4_PWM2P	BT14_TIOA29_0	-
PPC_PCFGR129 (0x007A)	P1_29	GPIO_POD R1:POD29	-	-	-	-	SMC4_PWM2M	BT15_TIOA30_0	-
PPC_PCFGR130 (0x007C)	P1_30	GPIO_POD R1:POD30	MFS5_SOT_0	-	MFS5_SDA	-	SMC5_PWM1P	BT15_TIOA31_0	-
PPC_PCFGR131 (0x007E)	P1_31	GPIO_POD R1:POD31	MFS5_SCK_0	-	MFS5_SCL	-	SMC5_PWM1M	-	-
PPC_PCFGR200 (0x0080)	P2_00	GPIO_POD R2:POD00	-	-	-	-	SMC5_PWM2P	-	-
PPC_PCFGR201 (0x0082)	P2_01	GPIO_POD R2:POD01	MFS5_CS0_0	-	-	-	SMC5_PWM2M	-	-
PPC_PCFGR202 (0x0084)	P2_02	GPIO_POD R2:POD02	-	-	-	-	-	-	-
PPC_PCFGR203 (0x0086)	P2_03	GPIO_POD R2:POD03	-	-	-	-	-	-	-
PPC_PCFGR204 (0x0088)	P2_04	GPIO_POD R2:POD04	MFS2_SOT_1	-	-	-	-	BT12_TIOA24_1	MDATA8
PPC_PCFGR205 (0x008A)	P2_05	GPIO_POD R2:POD05	MFS2_SCK_1	-	-	-	I2S0_SD_1	BT12_TIOA25_1	MDATA9
PPC_PCFGR206 (0x008C)	P2_06	GPIO_POD R2:POD06	-	-	-	-	I2S0_WS_1	BT13_TIOA26_1	MDATA10
PPC_PCFGR207 (0x008E)	P2_07	GPIO_POD R2:POD07	MFS2_CS0_1	-	-	-	I2S0_SCK_1	BT13_TIOA27_1	MDATA11
PPC_PCFGR208 (0x0090)	P2_08	GPIO_POD R2:POD08	MFS2_CS1_1	-	-	-	-	BT14_TIOA28_1	MCSX0

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR209 (0x0092)	P2_09	GPIO_PODR2:POD09	MFS3_SOT_1	-	-	-	-	BT14_TIOA29_1	MDATA12
PPC_PCFGR210 (0x0094)	P2_10	GPIO_PODR2:POD10	MFS3_SCK_1	-	-	-	SG0_SGA_2	BT15_TIOA30_1	MDATA13
PPC_PCFGR211 (0x0096)	P2_11	GPIO_PODR2:POD11	-	-	-	-	SG0_SGO_2	BT15_TIOA31_1	MDATA14
PPC_PCFGR212 (0x0098)	P2_12	GPIO_PODR2:POD12	MFS3_CS0_1	-	-	-	SG1_SGA_2	-	MDATA15
PPC_PCFGR213 (0x009A)	P2_13	GPIO_PODR2:POD13	MFS3_CS1_1	-	-	-	SG1_SGO_2	-	MCSX1
PPC_PCFGR214 (0x009C)	P2_14	GPIO_PODR2:POD14	MFS3_CS2_1	-	-	-	-	-	MDATA0
PPC_PCFGR215 (0x009E)	P2_15	GPIO_PODR2:POD15	MFS3_CS3_1	LCDD0	-	-	-	-	MDATA1
PPC_PCFGR216 (0x00A0)	P2_16	GPIO_PODR2:POD16	MFS4_SOT_0	LCDD1	MFS4_SDA	-	-	-	MDATA2
PPC_PCFGR217 (0x00A2)	P2_17	GPIO_PODR2:POD17	MFS4_SCK_0	LCDD2	MFS4_SCL	-	-	-	MDATA3
PPC_PCFGR218 (0x00A4)	P2_18	GPIO_PODR2:POD18	-	LCDD3	-	-	-	-	MDATA4
PPC_PCFGR219 (0x00A6)	P2_19	GPIO_PODR2:POD19	MFS4_CS0_0	-	-	-	-	-	MDATA5
PPC_PCFGR220 (0x00A8)	P2_20	GPIO_PODR2:POD20	MFS4_CS1_0	-	-	-	-	-	MDATA6
PPC_PCFGR221 (0x00AA)	P2_21	GPIO_PODR2:POD21	MFS4_CS2_0	-	-	-	-	-	MDATA7
PPC_PCFGR222 (0x00AC)	P2_22	GPIO_PODR2:POD22	MFS4_CS3_0	LCDD4	-	-	-	-	MAD0
PPC_PCFGR223 (0x00AE)	P2_23	GPIO_PODR2:POD23	MFS1_SOT_0	LCDD5	-	-	-	MFS1_SDA	MAD1

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR224 (0x00B0)	P2_24	GPIO_POD R2:POD24	MFS1_SCK_0	LCDD6	-	-	-	MFS1_SCL	MAD2
PPC_PCFGR225 (0x00B2)	P2_25	GPIO_POD R2:POD25	-	LCDD7	-	-	-	-	MAD3
PPC_PCFGR226 (0x00B4)	P2_26	GPIO_POD R2:POD26	MFS1_CS0_0	LCDD8	MFS0_CS0_1	-	-	-	MAD4
PPC_PCFGR227 (0x00B6)	P2_27	GPIO_POD R2:POD27	MFS1_CS1_0	LCDD9	-	-	-	MFS0_SOT_1	MAD5
PPC_PCFGR228 (0x00B8)	P2_28	GPIO_POD R2:POD28	MFS1_CS2_0	LCDD10	MFS0_SCK_1	-	-	-	MAD6
PPC_PCFGR229 (0x00BA)	P2_29	GPIO_POD R2:POD29	MFS1_CS3_0	LCDD11	-	-	-	BT0_TIOA0_1	MAD7
PPC_PCFGR300 (0x00C0)	P3_00	GPIO_POD R3:POD00	MFS1_CS1_1	-	-	-	-	BT4_TIOA8_1	MAD15
PPC_PCFGR301 (0x00C2)	P3_01	GPIO_POD R3:POD01	MFS1_CS2_1	-	-	-	-	BT4_TIOA9_1	MAD16
PPC_PCFGR302 (0x00C4)	P3_02	GPIO_POD R3:POD02	MFS1_CS3_1	-	-	-	-	BT5_TIOA10_1	MAD17
PPC_PCFGR303 (0x00C6)	P3_03	GPIO_POD R3:POD03	-	-	-	-	-	BT5_TIOA11_1	MAD18
PPC_PCFGR304 (0x00C8)	P3_04	GPIO_POD R3:POD04	-	-	BT10_TIOA20_1	-	-	-	MAD19
PPC_PCFGR305 (0x00CA)	P3_05	GPIO_POD R3:POD05	-	-	-	-	-	-	MAD20
PPC_PCFGR306 (0x00CC)	P3_06	GPIO_POD R3:POD06	-	-	-	-	-	-	MAD21
PPC_PCFGR307 (0x00CE)	P3_07	GPIO_POD R3:POD07	-	-	-	-	-	MFS4_CS3_1	MDQM1
PPC_PCFGR308 (0x00D0)	P3_08	GPIO_POD R3:POD08	MFS8_SOT_1	SG0_SGA_1	BT6_TIOA12_1	OCU0_OUT0_1	-	TRACE0_1	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR309 (0x00D2)	P3_09	GPIO_PODR3:POD09	MFS8_SCK_1	SG0_SGO_1	BT6_TIOA13_1	OCU0_OUT1_1	RLT0_TOT_1	TRACE1_1	-
PPC_PCFGR310 (0x00D4)	P3_10	GPIO_PODR3:POD10	-	SG1_SGA_1	BT7_TIOA14_1	OCU1_OUT0_1	-	TRACE2_1	CAN0_TX_1
PPC_PCFGR311 (0x00D6)	P3_11	GPIO_PODR3:POD11	MFS10_SOT_1	SG1_SGO_1	BT7_TIOA15_1	OCU1_OUT1_1	RLT1_TOT_1	TRACE3_1	-
PPC_PCFGR312 (0x00D8)	P3_12	GPIO_PODR3:POD12	MFS10_SCK_1	SG2_SGA_1	BT8_TIOA16_1	OCU2_OUT0_1	-	TRACE_CLK_1	CAN1_TX_1
PPC_PCFGR313 (0x00DA)	P3_13	GPIO_PODR3:POD13	-	SG2_SGO_1	BT8_TIOA17_1	OCU2_OUT1_1	RLT16_TOT_1	TRACE_CTL_1	-
PPC_PCFGR314 (0x00DC)	P3_14	GPIO_PODR3:POD14	-	-	-	-	-	-	-
PPC_PCFGR315 (0x00DE)	P3_15	GPIO_PODR3:POD15	-	-	BT11_TIOA22_1	OCU10_OUT0_1	-	-	CAN2_TX_1
PPC_PCFGR316 (0x00E0)	P3_16	GPIO_PODR3:POD16	MFS9_SOT_1	-	-	OCU9_OUT1_0	-	-	-
PPC_PCFGR317 (0x00E2)	P3_17	GPIO_PODR3:POD17	MFS9_SCK_1	-	BT11_TIOA23_1	OCU10_OUT1_1	-	-	-
PPC_PCFGR318 (0x00E4)	P3_18	GPIO_PODR3:POD18	-	-	BT10_TIOA21_1	-	-	-	CAN3_TX_1
PPC_PCFGR319 (0x00E6)	P3_19	GPIO_PODR3:POD19	MFS9_CS0_1	SG3_SGA_1	-	-	RLT3_TOT_1	-	-
PPC_PCFGR320 (0x00E8)	P3_20	GPIO_PODR3:POD20	MFS9_CS1_1	SG3_SGO_1	-	-	-	-	-
PPC_PCFGR321 (0x00EA)	P3_21	GPIO_PODR3:POD21	-	-	-	PWUTRG_1	MFS5_SOT_1	-	-
PPC_PCFGR322 (0x00EC)	P3_22	GPIO_PODR3:POD22	-	SG4_SGA_1	-	-	-	-	-
PPC_PCFGR323 (0x00EE)	P3_23	GPIO_PODR3:POD23	-	SG4_SGO_1	-	-	MFS5_CS0_1	-	-

Register (Offset)	Port	Resource Functional Outputs							
		POF=0	POF=1	POF=2	POF=3	POF=4	POF=5	POF=6	POF=7
PPC_PCFGR324 (0x00F0)	P3_24	GPIO_PODR3:POD24	-	-	-	-	-	-	-
PPC_PCFGR325 (0x00F2)	P3_25	GPIO_PODR3:POD25	-	-	-	-	-	-	-
PPC_PCFGR326 (0x00F4)	P3_26	GPIO_PODR3:POD26	-	-	-	-	-	-	-
PPC_PCFGR327 (0x00F6)	P3_27	GPIO_PODR3:POD27	-	-	-	-	-	-	-
PPC_PCFGR328 (0x00F8)	P3_28	GPIO_PODR3:POD28	MFS2_CS2_1	-	-	-	-	-	-
PPC_PCFGR329 (0x00FA)	P3_29	GPIO_PODR3:POD29	MFS2_CS3_1	-	-	-	-	-	-
PPC_PCFGR330 (0x00FC)	P3_30	GPIO_PODR3:POD30	-	-	-	-	-	-	-
PPC_PCFGR331 (0x00FE)	P3_31	GPIO_PODR3:POD31	-	-	-	-	CAN0_TX_2	-	-

Notes:

- *The hyphen indicates that setting is prohibited.*
- *If the in/out function on ports is used, set the register of PPC_PCFGR(Port Setting Register) and RIC_RESIN(Resource Input Setting Register).*

3.3. The Analog I/O Setting

If analog terminals assignment of the ADC and the LCDC are used, must set the following before using ADC and LCDC.

- The PPC_PCFGR:PIE register of target port makes disable digital input to set to 0.
- The corresponding POF should be set to 0.
- The GPIO_DDR register of target port makes input control to set to 0.
- The PPC_PCFGR:PDE/PUE register of target port makes disable Pull-Up/Pull-Down to set to 0.

Note:

- *An analog I/O port for A/D converter performs as pull-up when the software configures pull-up and down for a port simultaneously.*

3.4. Input Level Setting

This section shows the I/O port input level settings.

Pin No. of Package				Port Name	Value of PPC_PCFGR:PIL[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
-	-	2	2	P0_00	CMOS-hys	Automotive	-	-	
-	-	3	3	P0_01	CMOS-hys	Automotive	-	-	
-	-	4	4	P0_02	CMOS-hys	Automotive	-	-	
-	-	5	5	P0_03	CMOS-hys	Automotive	-	-	
-	-	6	6	P0_04	CMOS-hys	Automotive	-	-	
-	-	7	7	P0_05	CMOS-hys	Automotive	-	-	
2	2	8	8	P0_06	CMOS-hys	Automotive	-	-	
5	5	11	19	P0_07	CMOS-hys	Automotive	-	-	
6	6	12	20	P0_08	CMOS-hys	Automotive	-	-	
7	7	13	21	P0_09	CMOS-hys	Automotive	-	-	
8	8	14	22	P0_10	CMOS-hys	Automotive	-	-	
9	9	15	23	P0_11	CMOS-hys	Automotive	-	-	
10	10	16	24	P0_12	CMOS-hys	Automotive	-	-	
13	13	19	27	P0_13	CMOS-hys	Automotive	-	-	
14	14	20	28	P0_14	CMOS-hys	Automotive	-	-	
15	15	21	29	P0_15	CMOS-hys	Automotive	-	-	
16	16	22	30	P0_16	CMOS-hys	Automotive	-	-	
17	17	23	31	P0_17	CMOS-hys	Automotive	-	-	
-	20	26	34	P0_18	CMOS-hys	Automotive	-	-	
-	22	28	36	P0_19	CMOS-hys	Automotive	-	-	
21	23	29	37	P0_20	CMOS-hys	Automotive	-	-	
-	24	30	38	P0_21	CMOS-hys	Automotive	-	-	
-	25	31	39	P0_22	CMOS-hys	Automotive	-	-	
22	26	32	40	P0_23	CMOS-hys	Automotive	-	-	
-	29	35	43	P0_24	CMOS-hys	Automotive	-	-	
-	44	50	64	P0_25	CMOS-hys	Automotive	-	-	
-	45	51	65	P0_26	CMOS-hys	Automotive	-	-	
-	46	52	66	P0_27	CMOS-hys	Automotive	-	-	
-	47	53	67	P0_28	CMOS-hys	Automotive	-	-	
-	-	54	68	P0_29	CMOS-hys	Automotive	-	-	
40	49	56	70	P0_30	CMOS-hys	Automotive	-	-	
41	50	57	71	P0_31	CMOS-hys	Automotive	-	-	
44	53	60	74	P1_00	CMOS-hys	Automotive	-	-	
45	54	61	75	P1_01	CMOS-hys	Automotive	-	-	
-	55	62	76	P1_02	CMOS-hys	Automotive	-	-	
-	-	63	77	P1_03	CMOS-hys	Automotive	-	-	
-	-	64	78	P1_04	CMOS-hys	Automotive	-	-	

Pin No. of Package				Port Name	Value of PPC_PCFGR:PIL[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
-	-	65	79	P1_05	CMOS-hys	Automotive	-	-	
-	-	66	80	P1_06	CMOS-hys	Automotive	-	-	
-	-	67	81	P1_07	CMOS-hys	Automotive	-	-	
-	-	74	96	P1_08	CMOS-hys	Automotive	-	-	
54	64	77	101	P1_09	CMOS-hys	Automotive	-	-	
57	67	80	104	P1_10	CMOS-hys	Automotive	-	-	
58	68	81	105	P1_11	CMOS-hys	Automotive	-	-	
59	69	82	106	P1_12	CMOS-hys	Automotive	-	-	
60	70	83	107	P1_13	CMOS-hys	Automotive	-	-	
61	71	84	108	P1_14	CMOS-hys	Automotive	-	-	
62	72	85	109	P1_15	CMOS-hys	Automotive	-	-	
63	73	86	110	P1_16	CMOS-hys	Automotive	-	-	
64	74	87	111	P1_17	CMOS-hys	Automotive	-	-	
67	77	90	114	P1_18	CMOS-hys	Automotive	-	-	
68	78	91	115	P1_19	CMOS-hys	Automotive	-	-	
69	79	92	116	P1_20	CMOS-hys	Automotive	-	-	
70	80	93	117	P1_21	CMOS-hys	Automotive	-	-	
71	81	94	118	P1_22	CMOS-hys	Automotive	-	-	
-	82	95	119	P1_23	CMOS-hys	Automotive	-	-	
-	83	96	120	P1_24	CMOS-hys	Automotive	-	-	
-	84	97	121	P1_25	CMOS-hys	Automotive	-	-	
-	87	100	124	P1_26	CMOS-hys	Automotive	-	-	
-	88	101	125	P1_27	CMOS-hys	Automotive	-	-	
-	-	102	126	P1_28	CMOS-hys	Automotive	-	-	
-	-	103	127	P1_29	CMOS-hys	Automotive	-	-	
-	-	104	128	P1_30	CMOS-hys	Automotive	-	-	
74	89	105	129	P1_31	CMOS-hys	Automotive	-	-	
-	-	106	130	P2_00	CMOS-hys	Automotive	-	-	
-	-	107	131	P2_01	CMOS-hys	Automotive	-	-	
79	94	112	142	P2_02	CMOS-hys	Automotive	-	-	
80	95	113	143	P2_03	CMOS-hys	Automotive	-	-	
81	96	114	144	P2_04	CMOS-hys	Automotive	-	-	
82	97	115	145	P2_05	CMOS-hys	Automotive	-	-	
83	98	116	146	P2_06	CMOS-hys	Automotive	-	-	
84	99	117	147	P2_07	CMOS-hys	Automotive	-	-	
85	100	118	148	P2_08	CMOS-hys	Automotive	-	-	
88	103	121	151	P2_09	CMOS-hys	Automotive	-	-	
89	104	122	152	P2_10	CMOS-hys	Automotive	-	-	
90	105	123	153	P2_11	CMOS-hys	Automotive	-	-	
91	106	124	154	P2_12	CMOS-hys	Automotive	-	-	
92	107	125	155	P2_13	CMOS-hys	Automotive	-	-	

Pin No. of Package				Port Name	Value of PPC_PCFGR:PIL[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
93	108	126	156	P2_14	CMOS-hys	Automotive	-	-	
94	109	127	157	P2_15	CMOS-hys	Automotive	-	-	
95	110	128	158	P2_16	CMOS-hys	Automotive	-	-	
96	111	129	159	P2_17	CMOS-hys	Automotive	-	-	
97	112	130	160	P2_18	CMOS-hys	Automotive	-	-	
-	-	131	161	P2_19	CMOS-hys	Automotive	-	-	
-	-	132	162	P2_20	CMOS-hys	Automotive	-	-	
-	113	133	163	P2_21	CMOS-hys	Automotive	-	-	
-	116	136	166	P2_22	CMOS-hys	Automotive	-	-	
-	117	137	167	P2_23	CMOS-hys	Automotive	-	-	
-	118	138	168	P2_24	CMOS-hys	Automotive	-	-	
-	119	139	169	P2_25	CMOS-hys	Automotive	-	-	
-	-	140	170	P2_26	CMOS-hys	Automotive	-	-	
-	-	141	173	P2_27	CMOS-hys	Automotive	-	-	
-	-	142	174	P2_28	CMOS-hys	Automotive	-	-	
-	-	143	175	P2_29	CMOS-hys	Automotive	-	-	
-	-	-	9	P3_00	CMOS-hys	Automotive	-	-	
-	-	-	10	P3_01	CMOS-hys	Automotive	-	-	
-	-	-	11	P3_02	CMOS-hys	Automotive	-	-	
-	-	-	14	P3_03	CMOS-hys	Automotive	-	-	
-	-	-	15	P3_04	CMOS-hys	Automotive	-	-	
-	-	-	16	P3_05	CMOS-hys	Automotive	-	-	
-	-	-	17	P3_06	CMOS-hys	Automotive	-	-	
-	-	-	18	P3_07	CMOS-hys	Automotive	-	-	
-	-	-	58	P3_08	CMOS-hys	Automotive	-	-	
-	-	-	59	P3_09	CMOS-hys	Automotive	-	-	
-	-	-	60	P3_10	CMOS-hys	Automotive	-	-	
-	-	-	61	P3_11	CMOS-hys	Automotive	-	-	
-	-	-	62	P3_12	CMOS-hys	Automotive	-	-	
-	-	-	63	P3_13	CMOS-hys	Automotive	-	-	
-	-	-	82	P3_14	CMOS-hys	Automotive	-	-	
-	-	-	83	P3_15	CMOS-hys	Automotive	-	-	
-	-	-	90	P3_16	CMOS-hys	Automotive	-	-	
-	-	-	91	P3_17	CMOS-hys	Automotive	-	-	
-	-	-	92	P3_18	CMOS-hys	Automotive	-	-	
-	-	-	93	P3_19	CMOS-hys	Automotive	-	-	
-	-	-	94	P3_20	CMOS-hys	Automotive	-	-	
-	-	-	95	P3_21	CMOS-hys	Automotive	-	-	
-	-	-	97	P3_22	CMOS-hys	Automotive	-	-	
-	-	-	98	P3_23	CMOS-hys	Automotive	-	-	
-	-	-	134	P3_24	CMOS-hys	Automotive	-	-	

Pin No. of Package				Port Name	Value of PPC_PCFGR:PIL[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
-	-	-	135	P3_25	CMOS-hys	Automotive	-	-	
-	-	-	138	P3_26	CMOS-hys	Automotive	-	-	
-	-	-	139	P3_27	CMOS-hys	Automotive	-	-	
-	-	-	140	P3_28	CMOS-hys	Automotive	-	-	
-	-	-	141	P3_29	CMOS-hys	Automotive	-	-	
-	-	-	171	P3_30	CMOS-hys	Automotive	-	-	
-	-	-	172	P3_31	CMOS-hys	Automotive	-	-	

Notes:

- *The hyphen of "Value of PPC_PCFGR:PIL[1:0]" indicates that setting is prohibited except for the initial values(PIL[1:0]=00).*
- *"CMOS-hys" is CMOS hysteresis input level.*
- *"Automotive" is Automotive hysteresis input level.*
- *To get detailed information about the input level, see the DC characteristics of the Datasheet.*

3.5. Output Drive Capacity Setting

This section shows the I/O port output drive capacity settings.

Note that the drive capacity depends on power supply voltage. The table only describes the representative value when 5 V is used as standard value of power supply usage. Please see the actual characteristics in datasheet.

Pin No. of Package				Port Name	Value of PPC_PCFGR:ODR[1:0]				Remark
TEQFP 100	TEQFP 120	TEQFP 144	TEQFP 176		2'b00	2'b01	2'b10	2'b11	
-	-	2	2	P0_00	1mA	2mA	-	5mA	
-	-	3	3	P0_01	1mA	2mA	-	5mA	
-	-	4	4	P0_02	1mA	2mA	-	5mA	
-	-	5	5	P0_03	1mA	2mA	-	5mA	
-	-	6	6	P0_04	1mA	2mA	-	5mA	
-	-	7	7	P0_05	1mA	2mA	-	5mA	
2	2	8	8	P0_06	1mA	2mA	-	5mA	
5	5	11	19	P0_07	1mA	2mA	-	5mA	
6	6	12	20	P0_08	1mA	2mA	-	5mA	
7	7	13	21	P0_09	1mA	2mA	-	5mA	
8	8	14	22	P0_10	1mA	2mA	-	5mA	
9	9	15	23	P0_11	1mA	2mA	-	5mA	
10	10	16	24	P0_12	1mA	2mA	-	5mA	
13	13	19	27	P0_13	1mA	2mA	5mA	15mA	
14	14	20	28	P0_14	1mA	2mA	5mA	15mA	
15	15	21	29	P0_15	1mA	2mA	5mA	15mA	
16	16	22	30	P0_16	1mA	2mA	5mA	15mA	
17	17	23	31	P0_17	1mA	2mA	5mA	15mA	
-	20	26	34	P0_18	1mA	2mA	5mA	15mA	
-	22	28	36	P0_19	1mA	2mA	5mA	15mA	
21	23	29	37	P0_20	1mA	2mA	5mA	15mA	
-	24	30	38	P0_21	1mA	2mA	5mA	15mA	
-	25	31	39	P0_22	1mA	2mA	5mA	15mA	
22	26	32	40	P0_23	1mA	2mA	5mA	15mA	
-	29	35	43	P0_24	1mA	2mA	5mA	15mA	
-	44	50	64	P0_25	1mA	2mA	-	5mA	*2
-	45	51	65	P0_26	1mA	2mA	-	5mA	*2
-	46	52	66	P0_27	1mA	2mA	-	5mA	
-	47	53	67	P0_28	1mA	2mA	-	5mA	
-	-	54	68	P0_29	1mA	2mA	-	5mA	
40	49	56	70	P0_30	1mA	2mA	-	5mA	
41	50	57	71	P0_31	1mA	2mA	-	5mA	
44	53	60	74	P1_00	1mA	2mA	-	5mA	
45	54	61	75	P1_01	1mA	2mA	-	5mA	
-	55	62	76	P1_02	1mA	2mA	-	5mA	*1

Pin No. of Package				Port Name	Value of PPC_PCFGR:ODR[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
-	-	63	77	P1_03	1mA	2mA	-	5mA	*1
-	-	64	78	P1_04	1mA	2mA	-	5mA	
-	-	65	79	P1_05	1mA	2mA	-	5mA	*1
-	-	66	80	P1_06	1mA	2mA	-	5mA	*1
-	-	67	81	P1_07	1mA	2mA	-	5mA	
-	-	74	96	P1_08	1mA	2mA	-	5mA	
54	64	77	101	P1_09	1mA	2mA	-	5mA	
57	67	80	104	P1_10	1mA	2mA	5mA	30mA	
58	68	81	105	P1_11	1mA	2mA	5mA	30mA	
59	69	82	106	P1_12	1mA	2mA	5mA	30mA	*2
60	70	83	107	P1_13	1mA	2mA	5mA	30mA	*2
61	71	84	108	P1_14	1mA	2mA	5mA	30mA	
62	72	85	109	P1_15	1mA	2mA	5mA	30mA	
63	73	86	110	P1_16	1mA	2mA	5mA	30mA	
64	74	87	111	P1_17	1mA	2mA	5mA	30mA	
67	77	90	114	P1_18	1mA	2mA	5mA	30mA	*2
68	78	91	115	P1_19	1mA	2mA	5mA	30mA	*2
69	79	92	116	P1_20	1mA	2mA	5mA	30mA	
70	80	93	117	P1_21	1mA	2mA	5mA	30mA	*2
71	81	94	118	P1_22	1mA	2mA	5mA	30mA	*2
-	82	95	119	P1_23	1mA	2mA	5mA	30mA	
-	83	96	120	P1_24	1mA	2mA	5mA	30mA	
-	84	97	121	P1_25	1mA	2mA	5mA	30mA	
-	87	100	124	P1_26	1mA	2mA	5mA	30mA	
-	88	101	125	P1_27	1mA	2mA	5mA	30mA	*2
-	-	102	126	P1_28	1mA	2mA	5mA	30mA	*2
-	-	103	127	P1_29	1mA	2mA	5mA	30mA	
-	-	104	128	P1_30	1mA	2mA	5mA	30mA	*2
74	89	105	129	P1_31	1mA	2mA	5mA	30mA	*2
-	-	106	130	P2_00	1mA	2mA	5mA	30mA	
-	-	107	131	P2_01	1mA	2mA	5mA	30mA	
79	94	112	142	P2_02	-	-	-	-	Input only
80	95	113	143	P2_03	1mA	2mA	-	5mA	
81	96	114	144	P2_04	1mA	2mA	-	5mA	
82	97	115	145	P2_05	1mA	2mA	-	5mA	
83	98	116	146	P2_06	1mA	2mA	-	5mA	
84	99	117	147	P2_07	1mA	2mA	-	5mA	
85	100	118	148	P2_08	1mA	2mA	-	5mA	
88	103	121	151	P2_09	1mA	2mA	-	5mA	
89	104	122	152	P2_10	1mA	2mA	-	5mA	
90	105	123	153	P2_11	1mA	2mA	-	5mA	

Pin No. of Package				Port Name	Value of PPC_PCFGR:ODR[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
91	106	124	154	P2_12	1mA	2mA	-	5mA	
92	107	125	155	P2_13	1mA	2mA	-	5mA	
93	108	126	156	P2_14	1mA	2mA	-	5mA	
94	109	127	157	P2_15	1mA	2mA	-	5mA	
95	110	128	158	P2_16	1mA	2mA	-	5mA	*2
96	111	129	159	P2_17	1mA	2mA	-	5mA	*2
97	112	130	160	P2_18	1mA	2mA	-	5mA	
-	-	131	161	P2_19	1mA	2mA	-	5mA	
-	-	132	162	P2_20	1mA	2mA	-	5mA	
-	113	133	163	P2_21	1mA	2mA	-	5mA	
-	116	136	166	P2_22	1mA	2mA	-	5mA	
-	117	137	167	P2_23	1mA	2mA	-	5mA	*2
-	118	138	168	P2_24	1mA	2mA	-	5mA	*2
-	119	139	169	P2_25	1mA	2mA	-	5mA	
-	-	140	170	P2_26	1mA	2mA	-	5mA	
-	-	141	173	P2_27	1mA	2mA	-	5mA	
-	-	142	174	P2_28	1mA	2mA	-	5mA	
-	-	143	175	P2_29	1mA	2mA	-	5mA	
-	-	-	9	P3_00	1mA	2mA	-	5mA	
-	-	-	10	P3_01	1mA	2mA	-	5mA	
-	-	-	11	P3_02	1mA	2mA	-	5mA	
-	-	-	14	P3_03	1mA	2mA	-	5mA	
-	-	-	15	P3_04	1mA	2mA	-	5mA	
-	-	-	16	P3_05	1mA	2mA	-	5mA	
-	-	-	17	P3_06	1mA	2mA	-	5mA	
-	-	-	18	P3_07	1mA	2mA	-	5mA	
-	-	-	58	P3_08	1mA	2mA	-	5mA	
-	-	-	59	P3_09	1mA	2mA	-	5mA	
-	-	-	60	P3_10	1mA	2mA	-	5mA	
-	-	-	61	P3_11	1mA	2mA	-	5mA	
-	-	-	62	P3_12	1mA	2mA	-	5mA	
-	-	-	63	P3_13	1mA	2mA	-	5mA	
-	-	-	82	P3_14	1mA	2mA	-	5mA	
-	-	-	83	P3_15	1mA	2mA	-	5mA	
-	-	-	90	P3_16	1mA	2mA	-	5mA	
-	-	-	91	P3_17	1mA	2mA	-	5mA	
-	-	-	92	P3_18	1mA	2mA	-	5mA	
-	-	-	93	P3_19	1mA	2mA	-	5mA	
-	-	-	94	P3_20	1mA	2mA	-	5mA	
-	-	-	95	P3_21	1mA	2mA	-	5mA	
-	-	-	97	P3_22	1mA	2mA	-	5mA	

Pin No. of Package				Port Name	Value of PPC_PCFGR:ODR[1:0]				Remark
TEQFP	TEQFP	TEQFP	TEQFP		2'b00	2'b01	2'b10	2'b11	
100	120	144	176						
-	-	-	98	P3_23	1mA	2mA	-	5mA	
-	-	-	134	P3_24	1mA	2mA	-	5mA	
-	-	-	135	P3_25	1mA	2mA	-	5mA	
-	-	-	138	P3_26	1mA	2mA	-	5mA	
-	-	-	139	P3_27	1mA	2mA	-	5mA	
-	-	-	140	P3_28	1mA	2mA	-	5mA	
-	-	-	141	P3_29	1mA	2mA	-	5mA	
-	-	-	171	P3_30	1mA	2mA	-	5mA	
-	-	-	172	P3_31	1mA	2mA	-	5mA	

Notes:

- *The hyphen of "Value of PPC_PCFGR:ODR[1:0]" indicates that setting is prohibited except for the initial values(ODR[1:0]=00).*
- **1 When the PPC_PCFGR:POF[2:0] value setting is "3","6"(SDA or SCL function setting), the output drive capacity is "I^2C" regardless of the ODR setting. Then, the port status is to be the Open Drain, and IOL is to be 3mA.*
- **2 When the PPC_PCFGR:POF[2:0] value setting is "3","6"(SDA or SCL function setting), the port status is to be the Pseudo Open Drain, and IOL is to be the configured value for ODR.*
- *To get detailed information about the drive capability, see the DC characteristics of the Datasheet.*

3.6. Port Status

3.6.1. Hi-z Control

Traveo™ Platform hardware manual has description of System Special Setting Register (SYSC0_SPECFGR).

The [bit23] PSSPADCTRL: PSS-time port configuring bit should be configured as below.

− 0: Do not perform Hi-z control.
− 1: Perform Hi-z control.

Notes:

- *At RUN mode, if configured as Hi-z control, it doesn't affect the port status immediately, but after executing WFI instruction to update the profile registers, then it turns out Hi-z status during PSS mode.*
- *As opposite control from PSS to RUN, the port status of Hi-z will automatically be released without reconfiguration of SYSC0_SPECFGR.PSSPADCTRL = 0.*

3.6.2. Port Status Hold during PSS Mode

All of the GPIO area can be kept the port status during PSS mode by System Special Setting Register (SYSC0_SPECFGR).

The [bit31] to [bit24] HOLDIO_PDx: HOLD data latch bit should be configured as below.

− 0: Do not retain control.
− 1: Retain control.

Notes:

- *During MCU RUN mode, I/O port status will be latched immediately after SYSC0_SPECFGR.HOLDIO_PDx= 1 (Retain control) configured.*
- *After SYSC0_SPECFGR.HOLDIO_PDx= 1, the status of followings will be latched. Please note that PID is not included, that is, input data cannot be latched*.
  - *PPC_PCFGRijj POD, POE, PIL, PUE, PDE, ODR, NFE, and POF. (excluding PID: Input data cannot be latched)*
  - *RIC_RESINx.PORTSEL and RESSEL*
- *At turning from MCU PSS mode to RUN, the latched status will not be released automatically. Configuration SYSC0_SPECFGR.HOLDIO_PDx = 0 should be necessary for releasing the status.*

3.7. Function Port Group

A port group specifies an I/O port combination for a peripheral function. A peripheral function has some port groups and should be configured and used within a port group of them which is defined in the following table in order to satisfy AC specification.

Do not take a port combination which is not described in the following table as s port group.

Function	Port Group
Multi-Function Serial Ch.0	Group1 - MFS0_SCK_0 - MFS0_SIN_0 - MFS0_SOT_0 - MFS0_CS0_0 Group2 - MFS0_SCK_1 - MFS0_SIN_1 - MFS0_SOT_1 - MFS0_CS0_1
Multi-Function Serial Ch.1	Group1 - MFS1_SCK_0 - MFS1_SIN_0 - MFS1_SOT_0 - MFS1_CS0_0 - MFS1_CS1_0 - MFS1_CS2_0 - MFS1_CS3_0 Group2 - MFS1_SCK_1 - MFS1_SIN_1 - MFS1_SOT_1 - MFS1_CS0_1 - MFS1_CS1_1 - MFS1_CS2_1 - MFS1_CS3_1

Function	Port Group
Multi-Function Serial Ch.2	Group1 - MFS2_SCK_0 - MFS2_SIN_0 - MFS2_SOT_0 - MFS2_CS0_0 - MFS2_CS1_0 - MFS2_CS2_0 - MFS2_CS3_0 Group2 - MFS2_SCK_1 - MFS2_SIN_1 - MFS2_SOT_1 - MFS2_CS0_1 - MFS2_CS1_1 - MFS2_CS2_1 - MFS2_CS3_1
Multi-Function Serial Ch.3	Group1 - MFS3_SCK_0 - MFS3_SIN_0 - MFS3_SOT_0 - MFS3_CS0_0 - MFS3_CS1_0 Group2 - MFS3_SCK_1 - MFS3_SIN_1 - MFS3_SOT_1 - MFS3_CS0_1 - MFS3_CS1_1 - MFS3_CS2_1 - MFS3_CS3_1

Function	Port Group
Multi-Function Serial Ch.4	Group1 - MFS4_SCK_0 - MFS4_SIN_0 - MFS4_SOT_0 - MFS4_CS0_0 - MFS4_CS1_0 - MFS4_CS2_0 - MFS4_CS3_0 Group2 - MFS4_SCK_1 - MFS4_SIN_1 - MFS4_SOT_1 - MFS4_CS0_1 - MFS4_CS1_1 - MFS4_CS2_1 - MFS4_CS3_1
Multi-Function Serial Ch.5	Group1 - MFS5_SCK_0 - MFS5_SIN_0 - MFS5_SOT_0 - MFS5_CS0_0 Group2 - MFS5_SCK_1 - MFS5_SIN_1 - MFS5_SOT_1 - MFS5_CS0_1

Function	Port Group
Multi-Function Serial Ch.6	Group1 - MFS6_SCK_0 - MFS6_SIN_0 - MFS6_SOT_0 - MFS6_CS0_0 - MFS6_CS1_0
Multi-Function Serial Ch.7	Group1 - MFS7_SCK_0 - MFS7_SIN_0 - MFS7_SOT_0 - MFS7_CS0_0 - MFS7_CS1_0 Group2 - MFS7_SCK_1 - MFS7_SIN_1 - MFS7_SOT_1
Multi-Function Serial Ch.8	Group1 - MFS8_SCK_0 - MFS8_SIN_0 - MFS8_SOT_0 - MFS8_CS0_0 Group2 - MFS8_SCK_1 - MFS8_SIN_1 - MFS8_SOT_1 Group3 - MFS8_SCK_2 - MFS8_SIN_2 - MFS8_SOT_2 - MFS8_CS0_2

Function	Port Group
Multi-Function Serial Ch.9	Group1 - MFS9_SCK_0 - MFS9_SIN_0 - MFS9_SOT_0 - MFS9_CS0_0 - MFS9_CS1_0 Group2 - MFS9_SCK_1 - MFS9_SIN_1 - MFS9_SOT_1 - MFS9_CS0_1 - MFS9_CS1_1 Group3 - MFS9_SCK_2 - MFS9_SIN_2 - MFS9_SOT_2 - MFS9_CS0_2 - MFS9_CS1_2
Multi-Function Serial Ch.10	Group1 - MFS10_SCK_0 - MFS10_SIN_0 - MFS10_SOT_0 - MFS10_CS0_0 Group2 - MFS10_SCK_1 - MFS10_SIN_1 - MFS10_SOT_1 - MFS10_CS0_1
Multi-Function Serial Ch.11	Group1 - MFS11_SCK_0 - MFS11_SIN_0 - MFS11_SOT_0 - MFS11_CS0_0 - MFS11_CS1_0

3.8. Key Code Register

The write access to I/O Port register is protected by Key Code Register.

Table 3-1 Relationship between I/O Port Register and Key Code Register

I/O Port Register	Key Code Register	Remark
Data Direction Register (GPIO_DDRi)	GPIO Key Code Register (GPIO_KEYCDR)	-
Data Direction Set Register (GPIO_DDSRi)	GPIO Key Code Register (GPIO_KEYCDR)	-
Data Direction Clear Register (GPIO_DDCRi)	GPIO Key Code Register (GPIO_KEYCDR)	-
Port Output Data Register (GPIO_PODRi)	-	-
Port Output Set Register (GPIO_POSRi)	-	-
Port Output Clear Register (GPIO_POCRi)	-	-
Port Input Enable Register (GPIO_PORTEN)	GPIO Key Code Register (GPIO_KEYCDR)	-
Port Input Data Register (GPIO_PIDRi)	-	-
GPIO Key Code Register (GPIO_KEYCDR)	-	-
Port Setting Register (PPC_PCFGRijj)	PPC Key Code Register (PPC_KEYCDR)	-
PPC Key Code Register (PPC_KEYCDR)	-	-
Resource Input Setting Register (RIC_RESINn)	RIC Key Code Register (RIC_KEYCDR)	-
RIC Key Code Register (RIC_KEYCDR)	-	-

4. Registers

See the common information of the registers on I/O Port.

5. Configuration Procedure

The configuration procedure of resource I/O port is described.

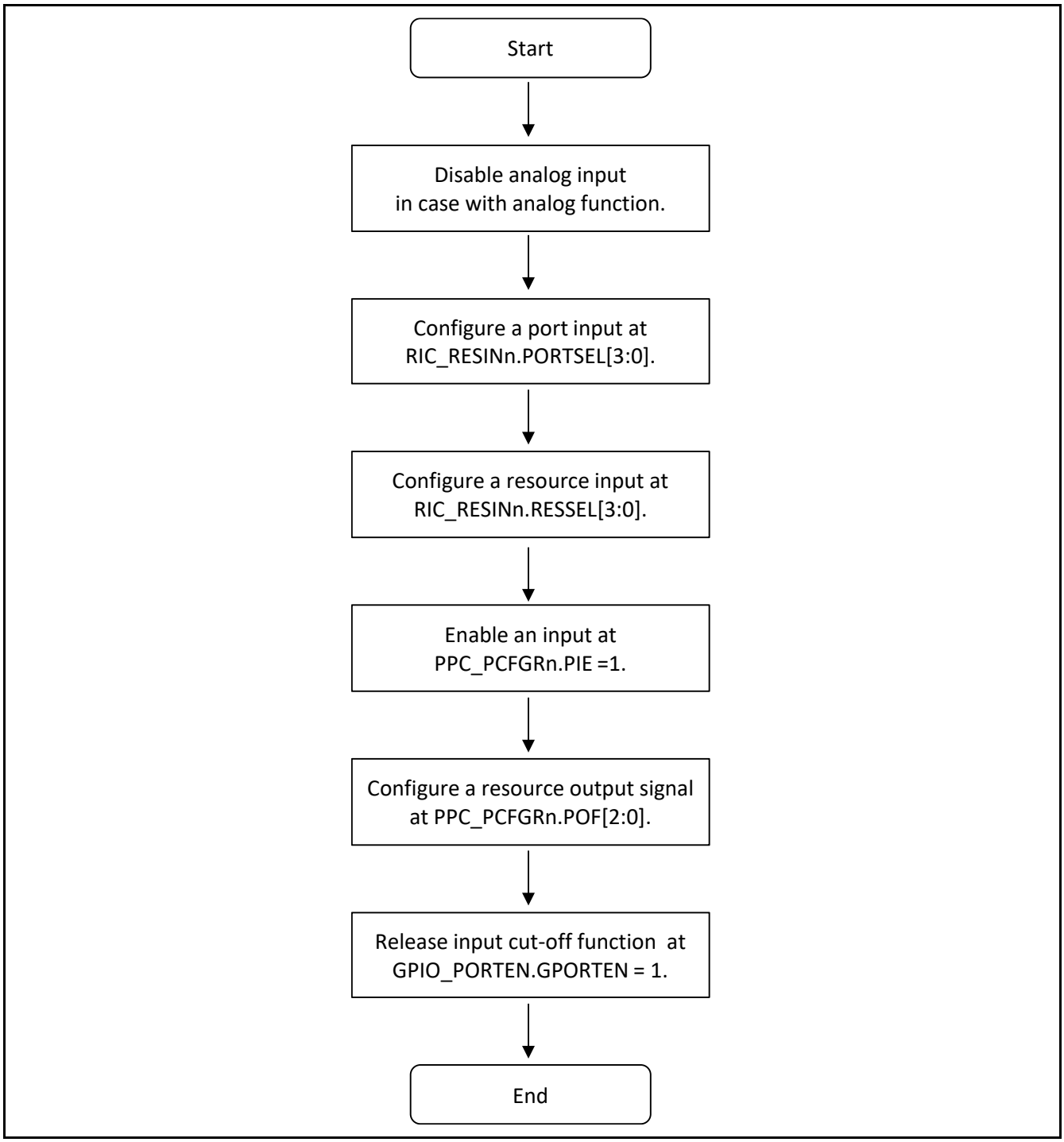
Category	I/O Direction	Referable Procedure	Remark
Resource I/O Port	Both Direction	See 5.1	
	Input	See 5.2	
	Output	See 5.3	
Port Function	Input	See 5.4	
	Output	See 5.5	
Analog Function	Input or Output	See 5.6	

Notes:

- *Glitch at output port may sometime be observed when the following case.*
  1. *from input to output*
  2. *from output to input*
  3. *from input to input*
  4. *from output to output*

5.1. Resource I/O Port (Both Direction)

Figure 5-1: Procedure of Resource I/O Port (Both Direction)

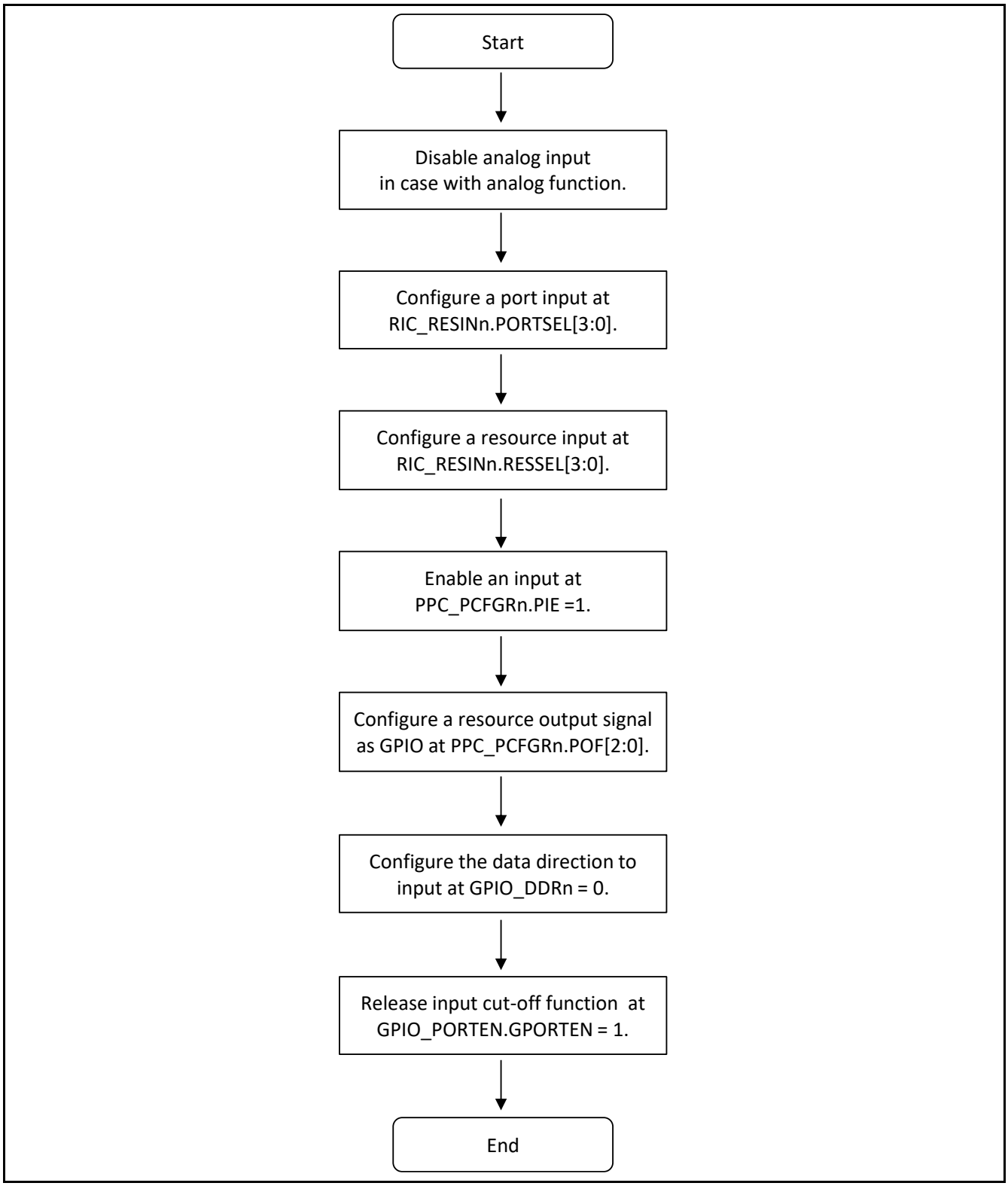


Note:

- *The dedicated input/output direction configuration should be necessary for SCK (MSF).*

5.2. Resource Input

Figure 5-2: Procedure of Resource Input

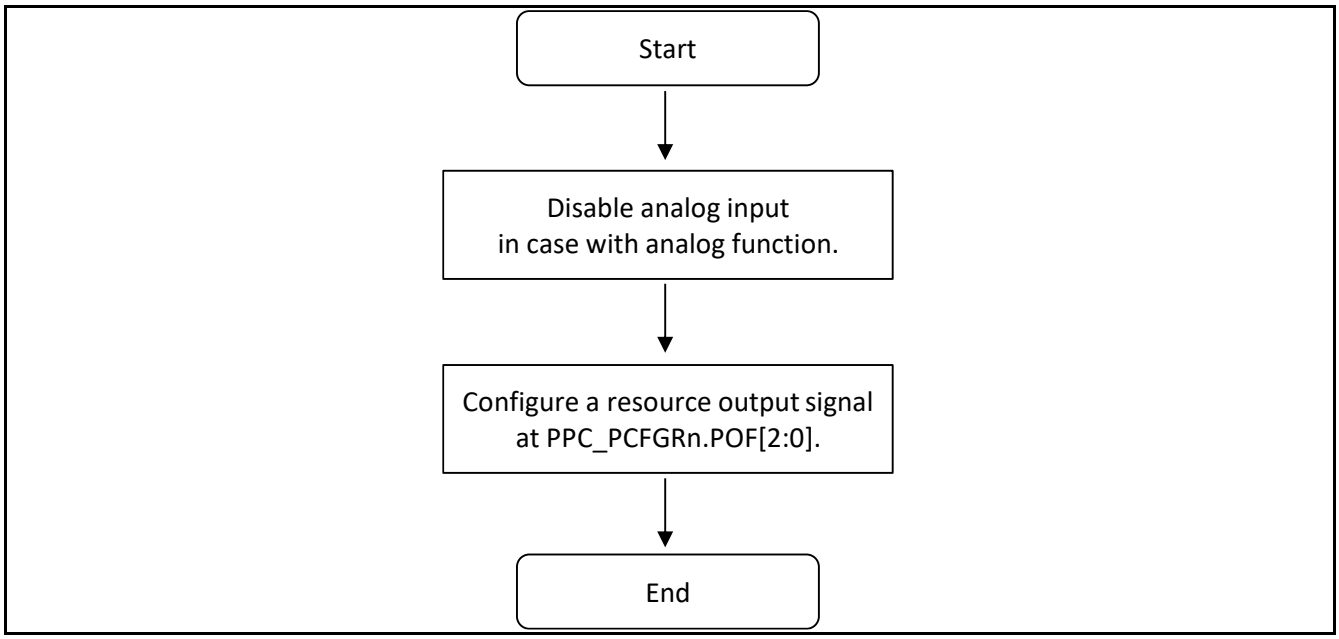


Notes:

- *RIC_RESINn register is only used for a resource which supports relocation function or resource input function from the other resource. The RIC_RESINn should be configured as above.*
- *A resource which supports neither relocation nor the other resource input doesn't have its RIC_RESINn register. Configurations can be skipped in above procedure.*

5.3. Resource Output

Figure 5-3: Procedure of Resource Output

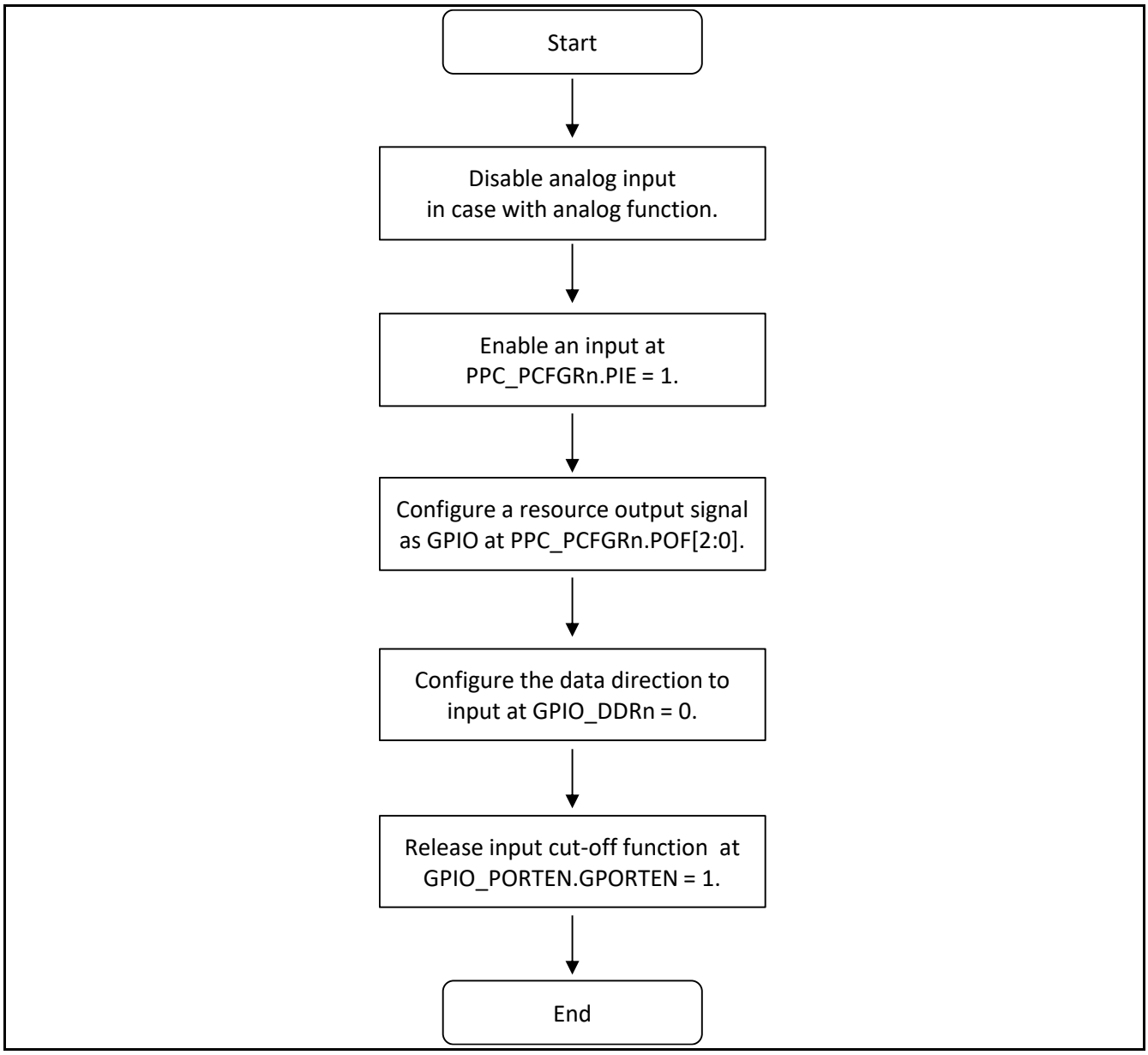


Note:

- *The dedicated output control configuration as well as POF should be necessary for bellows.*
  - *SOUT(MFS),*
  - *SGA,SGO(SG)*
  - *WOT(RTC)*
  - *MFS_CS (MFS)*

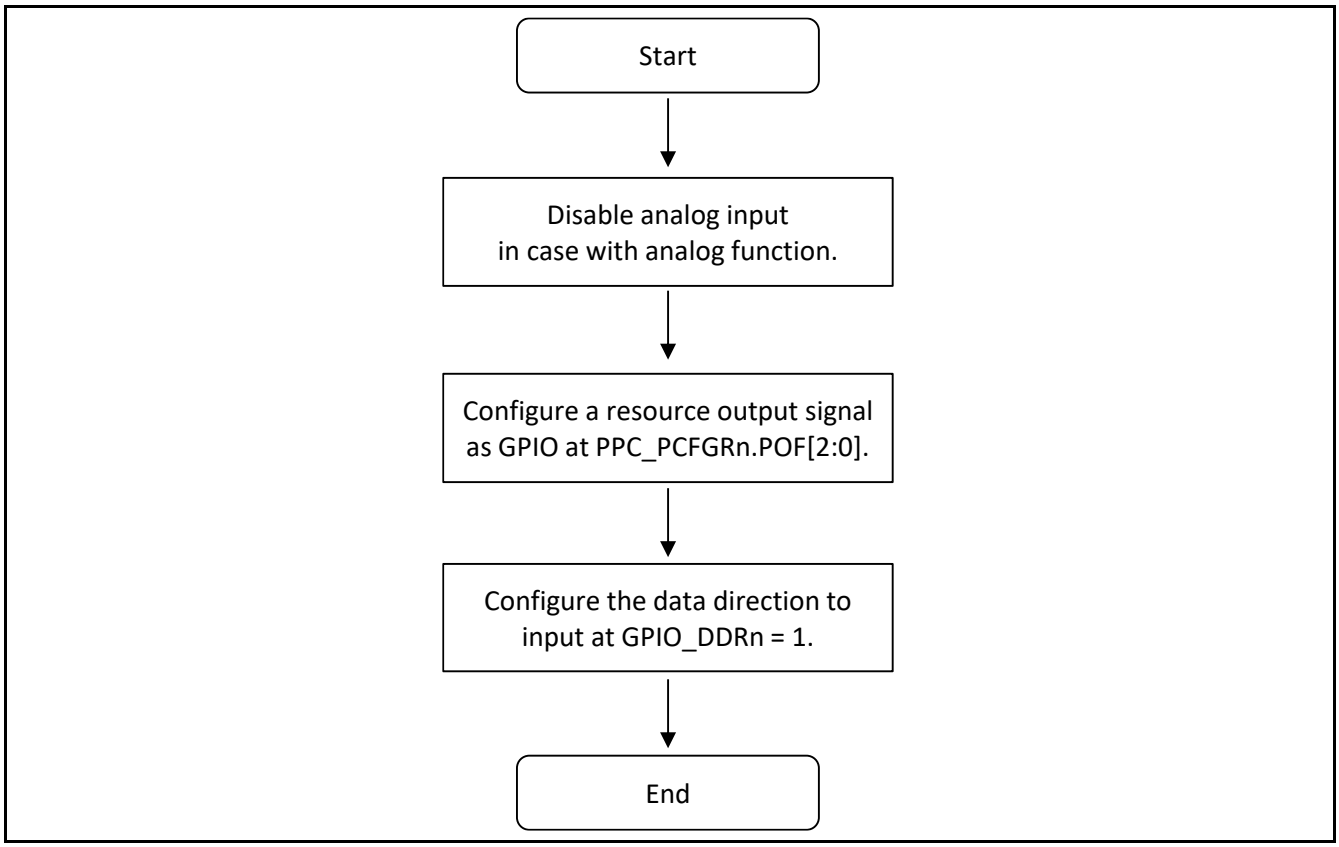
5.4. Port Function Input

Figure 5-4: Procedure of Port Input



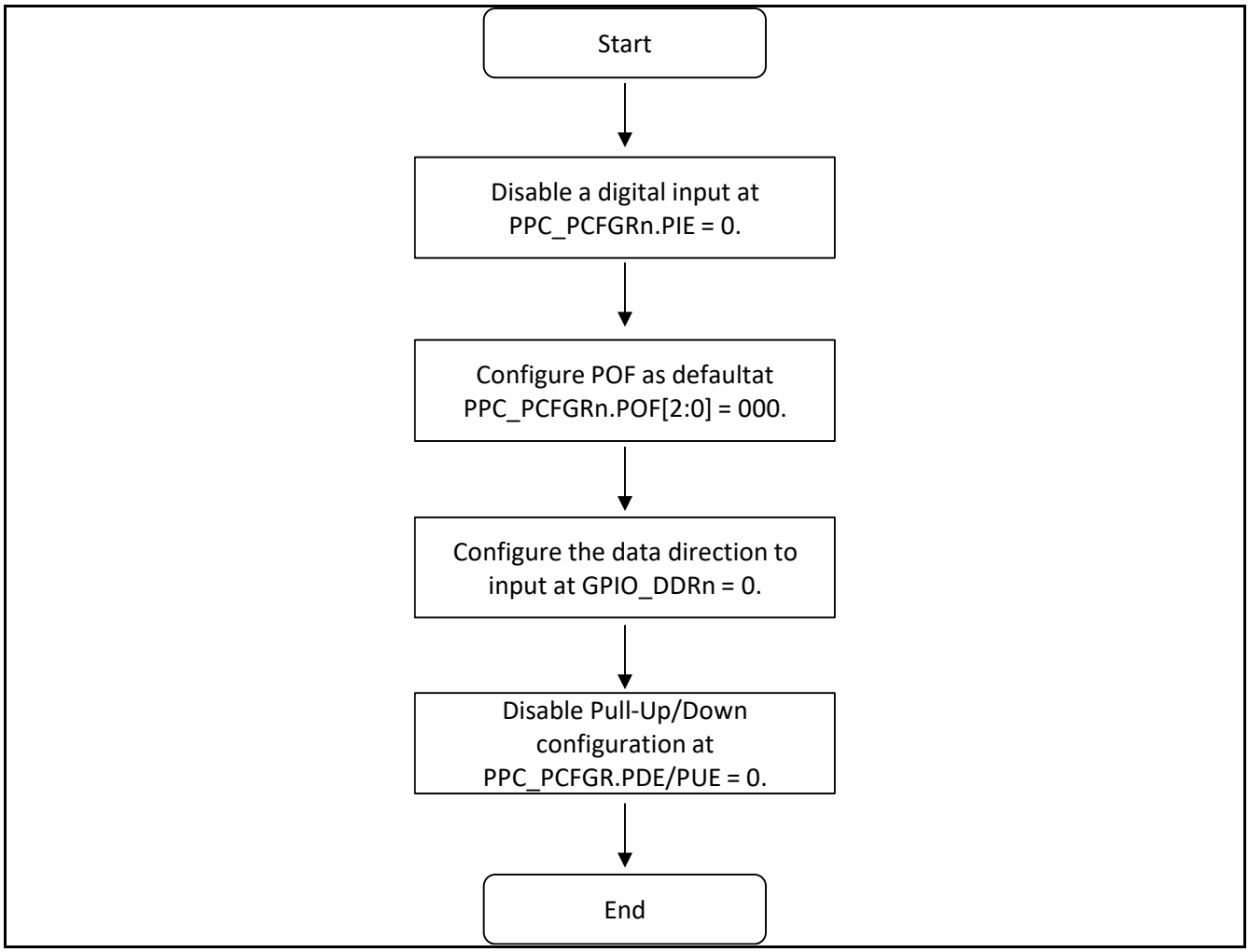
5.5. Port Function Output

Figure 5-5: Procedure of Port Output



5.6. Analog Function Input or Output

Figure 5-6: Procedure of Analog Function Input or Output



Notes:

- *The procedure is for analog input or output function of A/D converter and LCDC of which the ports have multiplexed usage.*
- *Regarding the analog switch setting, see 'CHAPTER of 12/10/8-BIT Analog to Digital Converter ' and 'CHAPTER of LCDC Controller'.*

6. Note

6.1. Base Timer Port

The following two operation modes function of base timer are limited due to the product does not have BTx_TIOB2x+1 ports and BTx_TIOA2x+1 input function (x = 0, 1, 2,,, 15 are BT unit number).

- Mode0: 16-bit timer standard mode

  BTx_TIOB2x+1 are internally solid and do not support 16bit timer mode.

- Mode1: 32-bit timer full-function mode

  BTx_TIOB2x+1 and BTx_TIOA2x+1 input functions are internally solid and do not support PWC function using TIN and the function using TGIN.

6.2. I2C Port Configuration

Port configuration of I2C should be done with the following order.

(1) Set the MD (Operation Mode Setting Bits) of the SMR (Serial Mode Register) to I2C mode.

(2) Set the POF (Port Output Function Selection Bit) of the PPC_PCFGRijj (Port Setting Register) to I2C.

Otherwise, I2C fast-mode output pins "P1_02(MFS6_SDA), P1_03(MFS6_SCL), P1_05(MFS7_SDA), P1_06(MFS7_SCL)" would output unintentional "high level" from the ports during the error procedure (2) -> (1) due to I/O circuit spec.

CHAPTER 12: State Transition

This chapter explains the state transition.

1. Overview
2. Diagram of State transition
3. Fetching the Operation Mode
4. Changes to PSS and RUN

CODE: STATE_TRANSITION-JUPI-E1

1. Overview

This section gives a brief overview of State transition

Refer to the chapter of "Low-power Consumption" in Traveo™ Platform hardware manual for the detailed information for performing a change state.

2. Diagram of State Transition

This section shows diagram of state transitions.

The device state transitions for this series are shown below.

Figure 2-1 Diagram of Device State Transitions

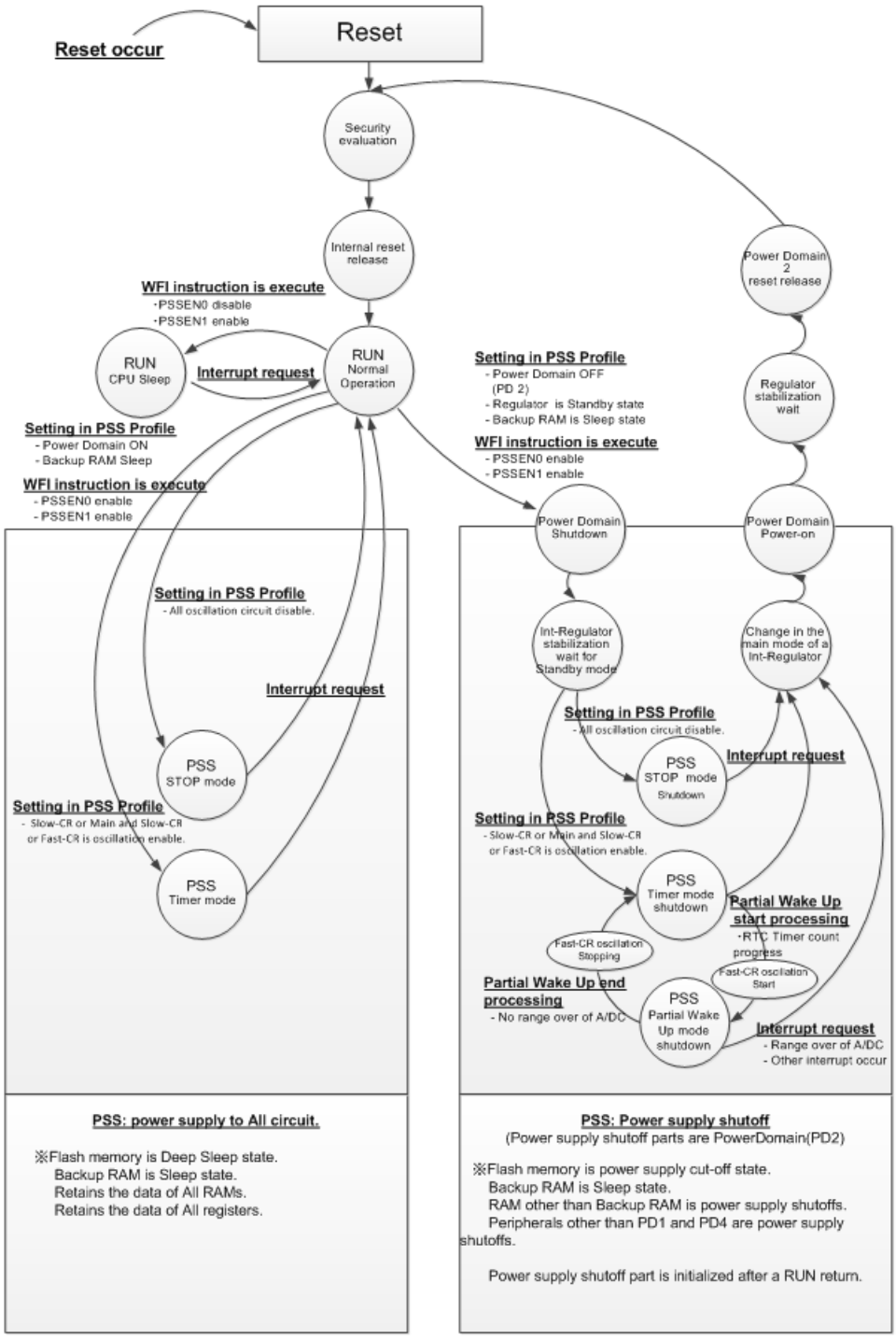


Figure 2-2 RUN/PSS State Definitions

S6J3400 supports only the below defined operating state.

Internal state and a frequency definition

		RUN Normal Operation	RUN CPU Sleep	PSS Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Timer mode shutdown	PSS STOP mode Shutdown
Clocks	CLK_CPU	Customer any Frequency	←	Prohibition of an oscillation	Prohibition of an oscillation	Prohibition of an oscillation	Prohibition of an oscillation	Prohibition of an oscillation
	CLK_FCLK							
	CLK_ATB							
	CLK_DBG							
	CLK_HPM							
	CLK_DMA							
	CLK_MEMC							
	CLK_SYSC1							
	CLK_TRC							
	CLK_SYSC0H			Customer any Frequency		Customer any Frequency		
	CLK_COMH							
	CLK_LLPBM			Prohibition of an oscillation		Prohibition of an oscillation		
	CLK_LCP							
	CLK_LCP0							
	CLK_LCP0A							
	CLK_LCP1							
	CLK_LCP1A							
oscillation state of a source clock	Slow-CR	100KHz	←	100KHz / disable	Prohibition of an oscillation	100KHz	100KHz / disable	Prohibition of an oscillation
	Fast-CR	4MHz	←	4MHz / disable	Prohibition of an oscillation	4MHz / disable (Fast-CR is unrelated to the PSS profile to oscillate.)	4MHz / disable	Prohibition of an oscillation
	Main oscillator	4MHz~16MHz	←	4~16MHz / disable	Prohibition of an oscillation	Prohibition of an oscillation	4~16MHz / disable	Prohibition of an oscillation
	Sub oscillator	32KHz	←	32KHz / disable	Prohibition of an oscillation	Prohibition of an oscillation	32KHz / disable	Prohibition of an oscillation
	PLL	Customer any Frequency	←	Prohibition of an oscillation				
	SSCG_PLL	Customer any Frequency	←					

The macro operating state in each mode, and the state of the power domain.

		RUN Normal Operation	RUN CPU Sleep	PSS Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Timer mode shutdown	PSS STOP mode Shutdown
CPU(PD2)	-	enable	disable	disable	disable	Power down	Power down	Power down
FLASH(PD2)	-	Operation	NOP	Deep Sleep	Deep Sleep	Power down	Power down	Power down
Backup RAM(PD4)	-	Operation	NOP	NOP	NOP	NOP	NOP	NOP
Int-Regulator	-	Main mode	Main mode	Main mode	Main mode	Standby mode	Standby mode	Standby mode
Power Domain	PD2	ON	ON	ON	ON	OFF	OFF	OFF
	PD4	ON	ON	ON	ON	ON	ON	ON

The conditions for changing in each state.

		RUN Normal Operation	RUN CPU Sleep	PSS Timer mode	PSS STOP mode	PSS PWU mode Shutdown	PSS Timer mode shutdown	PSS STOP mode Shutdown
Setup for changes	PSSEN0	-	disable	enable	enable	enable	enable	enable
	PSSEN1	-	enable	enable	enable	enable	enable	enable
Changes start command		-	Execution of a WFI command.					

3. Fetching the Operation Mode

This section describes the Fetching the Operation Mode.

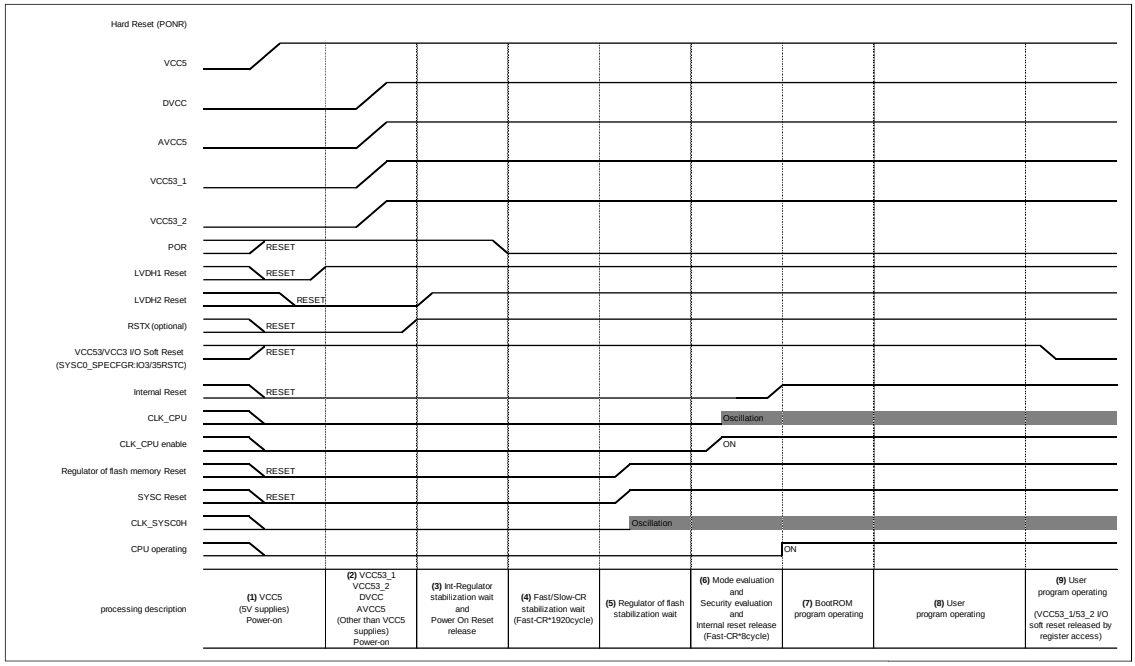
The operation mode is fetched by sampling the RST (Reset).

The following shows an operation sequence from an occurrence of reset cause to the determination of an operation mode.

The conditions for starting of a power supply and the correlation of a power supply are shown in Figure3-1.

Operation Mode Fetch Timing Chart of RESET other than PONR is shown in Figure3-2.

Figure3-1 Operation Mode Fetch Timing Chart of PONR



1) Each power supply is stable before reset release for that domain

Below table describes reset asserted until stabilization of each supply:

#	Power supply	Reset	Who must monitor the supply?	Corr. phase in POR release diagram (sheet "POR diagram")	Explanation of Reset
1	**VCC5**	LVDH1 reset	CY MCU	(1)	VCC5 is monitored by LVDH1 which issues & releases reset above minimum operation voltage
2	**VDD (internal 1.2V)**	POR	CY MCU	(3)	VDD rises above minimum operation voltage during Int-Regulator stabilization wait shown in (3) in sheet POR Diagram
3	**VCC53_1**	LVDH1 reset	CY MCU	(1)	LVDH1 can monitor VCC53_1 only if it's shorted with VCC5 on the board.
		LVDH2 reset	CY MCU	(1) or (2)	VCC53_1 is monitored by LVDH2 which issues & releases reset above minimum operation voltage
4	**VCC53_2**	LVDH1 reset or RSTX	CY MCU or User (*see note)	(1) or (2)	LVDH1 can monitor VCC53_2 only if it's shorted with VCC5 on the board. Otherwise, this supply has to be monitored on the board and RSTX has to be issued to inactivate the domain until the supply is stable.
		LVDH2 reset	CY MCU	(1) or (2)	LVDH2 can monitor VCC53_2 only if it's shorted with VCC53_1 on the board.
5	**DVCC**	LVDH1 reset or RSTX	CY MCU or User (*see note)	(1)	LVDH1 can monitor DVCC only if it's shorted with VCC5 on the board. Otherwise, this supply has to be monitored on the board and RSTX has to be issued to inactivate the domain until the supply is stable.
		LVDH2 reset	CY MCU	(1) or (2)	LVDH2 can monitor DVCC only if it's shorted with VCC53_1 on the board.
6	**AVCC5**	LVDH1 reset or RSTX	CY MCU or User (*see note)	(1)	LVDH1 can monitor AVCC5 only if it's shorted with VCC5 on the board. Otherwise, this supply has to be monitored on the board and RSTX has to be issued to inactivate the domain until the supply is stable.
		LVDH2 reset	CY MCU	(1) or (2)	LVDH2 can monitor AVCC5 only if it's shorted with VCC53_1 on the board.

2) Isolation is valid before each pair of adjacent supplies is stable during power-up sequence

Below table describes isolation available for each adjacent supply pair:

#	From	To	Isolation method	Isolator assert timing (sheet "POR diagram")	Isolator release timing (sheet "POR diagram")
1	VCC5	VDD	POR & LVDH1 initialize both domains until VCC5 stablization	(1) POR assertion	(2) LVDH1 release
2	VCC53_1	VDD	3.3/5V I/O reset (SYSC0_SPECFGR:IO35RSTC) works as isolation until VCC53_1 powers up and user inverts the bit in software	(1) POR assertion	(9) Reset release by software (SYSC0_SPECFGR:IO35RSTC=0)
3	DVCC	VDD	LVDH1 reset (if DVCC shorted w/ VCC5 on board) or LVDH2 reset (if DVCC shorted w/ VCC53_1 on board) or RSTX, etc. (otherwise) works as isolation until DVCC powers up and user inverts the bit in software	(1) POR assertion or RSTX etc. assertion	(1) LVDH1 or LVDH2 or RSTX etc. release
5	AVCC5	VDD	LVDH1 reset (if AVCC5 shorted w/ VCC5 on board) or LVDH2 reset (if AVCC5 shorted w/ VCC53_1 on board) or RSTX, etc. (otherwise) works as isolation until AVCC5 powers up and user inverts the bit in software	(1) POR assertion or RSTX etc. assertion	(1) LVDH1 or LVDH2 or RSTX etc. release
6	VCC53_2	VDD	3.3V/5V I/O reset (SYSC0_SPECFGR:IO3RSTC) works as isolation until VCC53_2 powers up and user inverts the bit in software	(1) POR assertion	(9) Reset release by software (SYSC0_SPECFGR:IO3RSTC=0)
7	VDD	VCC5	POR initialized both domains until VDD stabilization	(1) POR assertion	(3) POR release
8	VDD	VCC53_1	POR initialized both domains until VDD stabilization	(1) POR assertion	(3) POR release
9	VDD	DVCC	POR initialized both domains until VDD stabilization	(1) POR assertion	(3) POR release
10	VDD	AVCC5	POR initialized both domains until VDD stabilization	(1) POR assertion	(3) POR release
11	VDD	VCC53_2	POR initialized both domains until VDD stabilization	(1) POR assertion	(3) POR release
12	VCC5	VCC53_1	POR & LVDH1 initialize both domains until VCC5 stablization	(1) POR assertion	(2) LVDH1 release
13	VCC5	VCC53_2	POR & LVDH1 initialize both domains until VCC5 stablization	(1) POR assertion	(2) LVDH1 release

Existing power supply pairs (# in parenthesis = corresp. entry in above table):

		To						
		VCC5	VDD	VCC53_1	DVCC	AVCC5	VCC53_2	
From	VCC5		x (1)	x (12)			x (13)	VCC5 interacts only with regulated domain
	VDD	x (7)		x (8)	x (9)	x (10)	x (11)	VDD interacts with Always-ON domain I/Os
	VCC53_1		x (2)					VCC53_1 powers only I/Os in VDD domain
	DVCC		x (3)					DVCC powers only I/Os & analog SWs in VDD domain
	AVCC5		x (5)					AVCC5 powers analog SWs & A/D in VDD domain
	VCC53_2		x (6)					VCC53_2 powers only I/Os in VDD domain

Figure3-2 Operation Mode Fetch Timing Chart of Other Reset

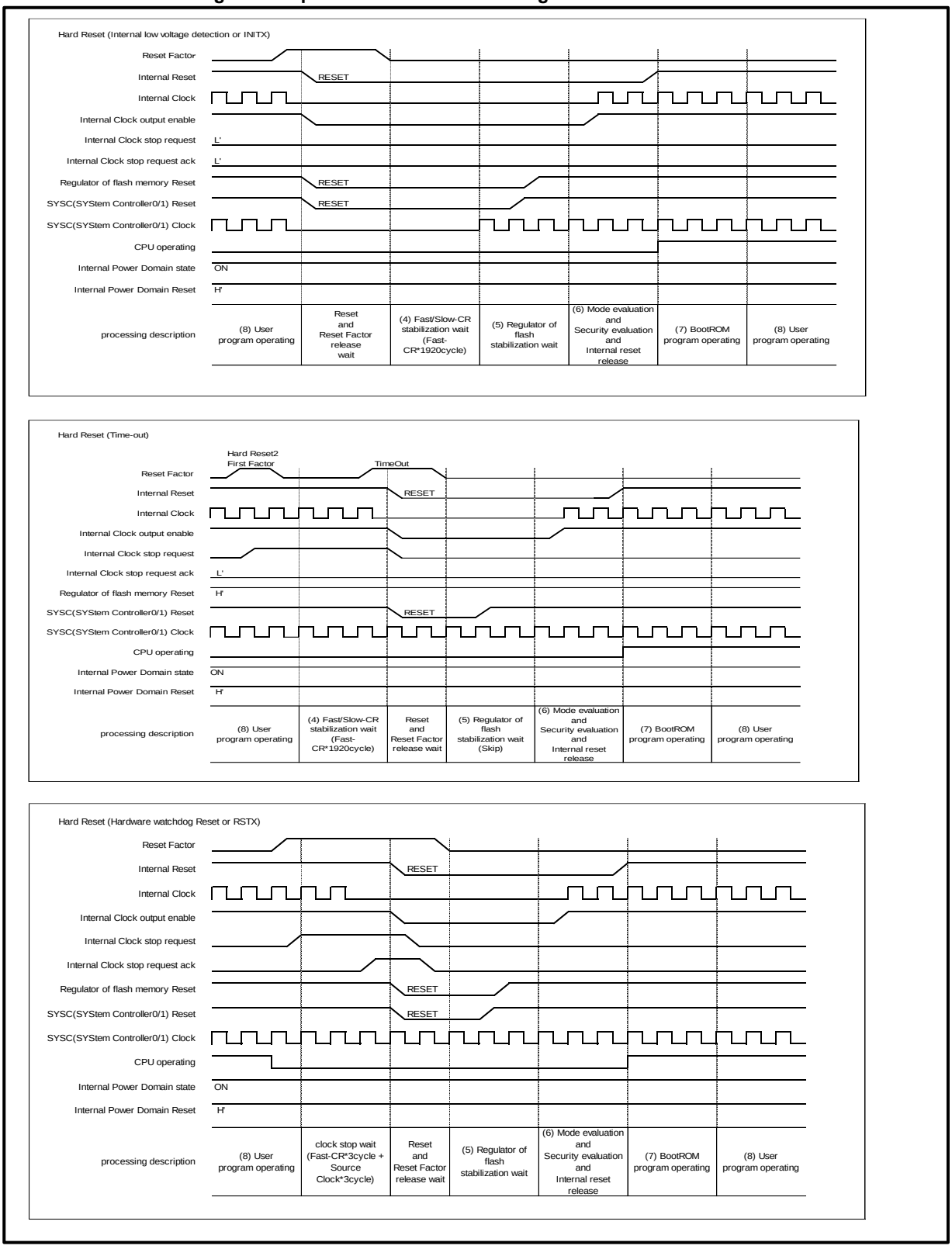
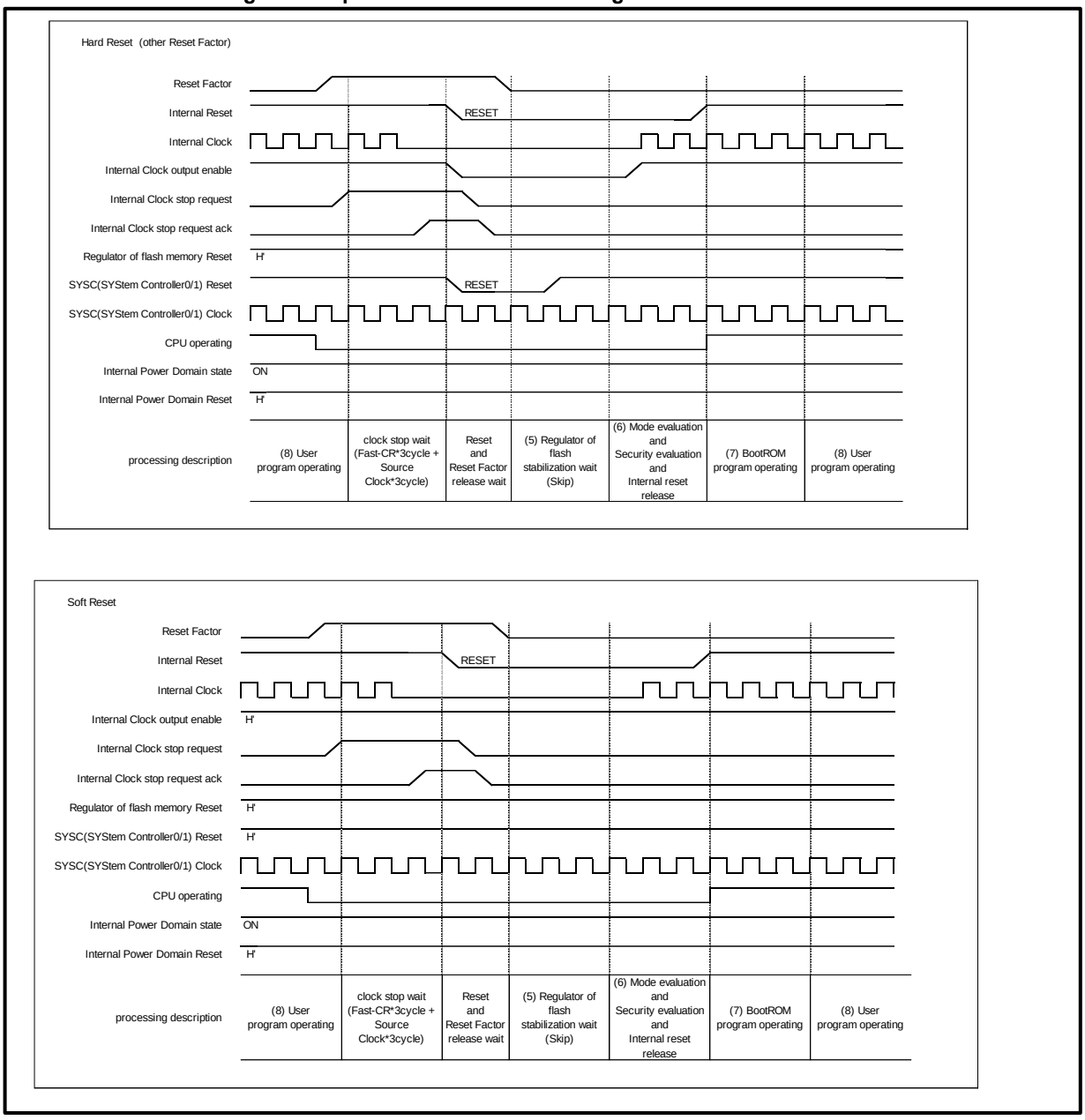


Figure3-3 Operation Mode Fetch Timing Chart of Other Reset



4. Changes to PSS and RUN

This section shows the sequence which changes to PSS and RUN.

Figure4-1 Changes to PSS Timing Chart

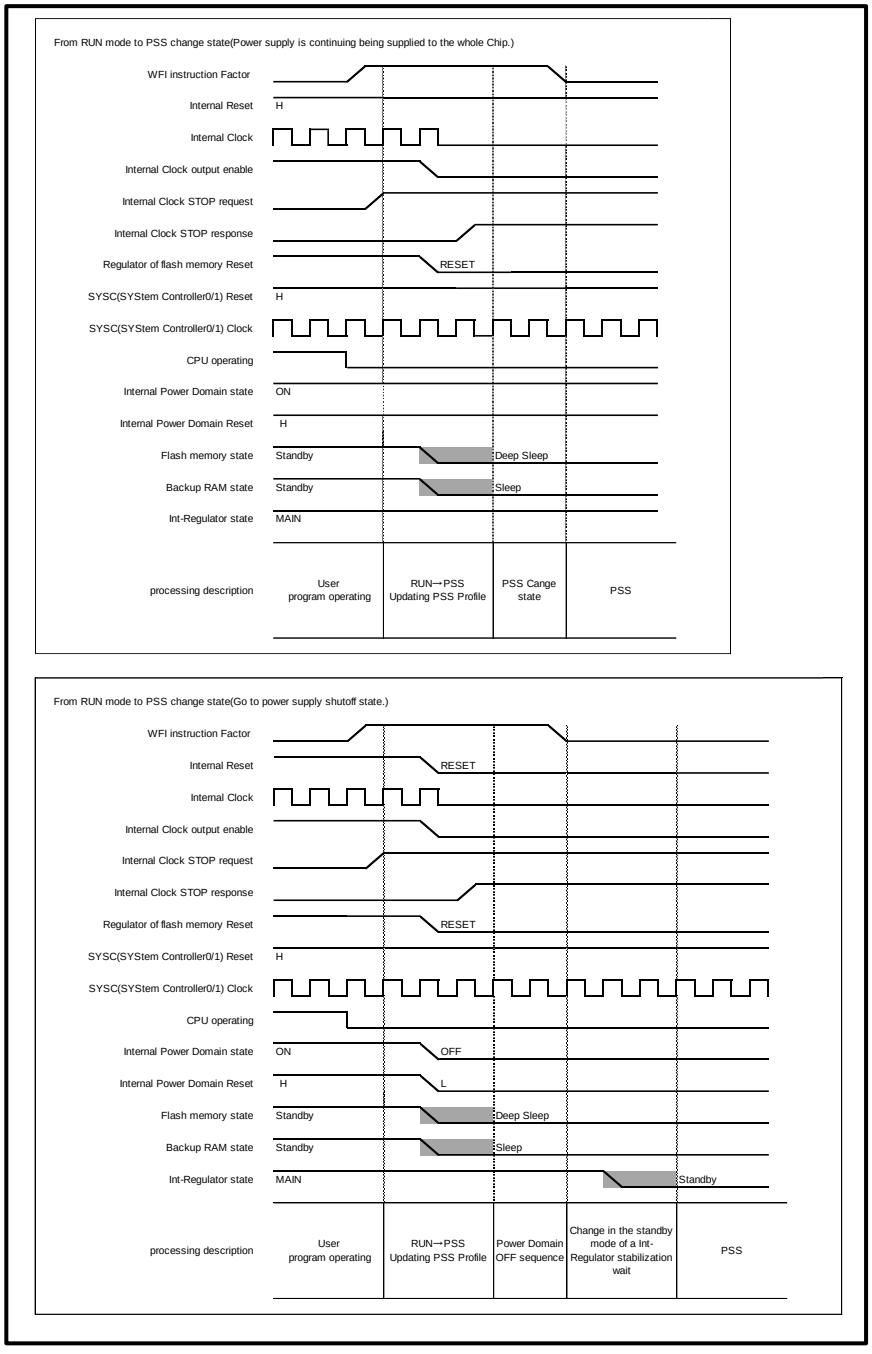


Figure4-2 Changes to PSS Timing Chart

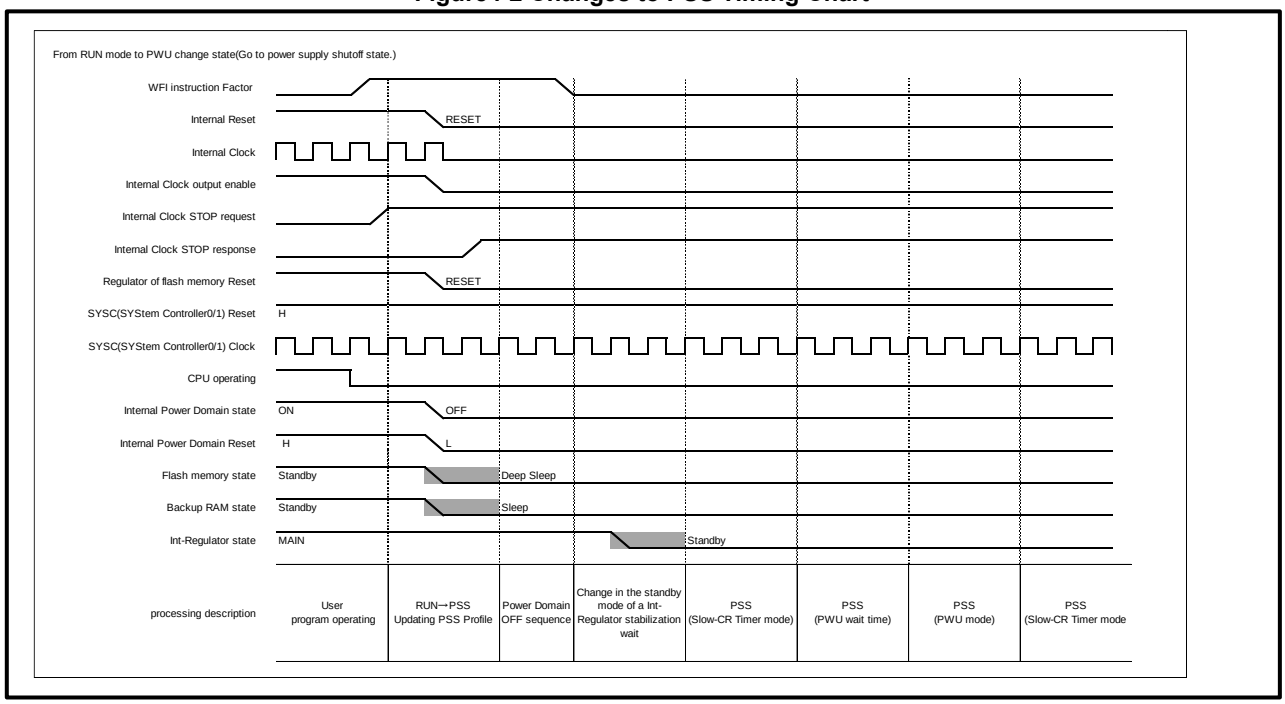
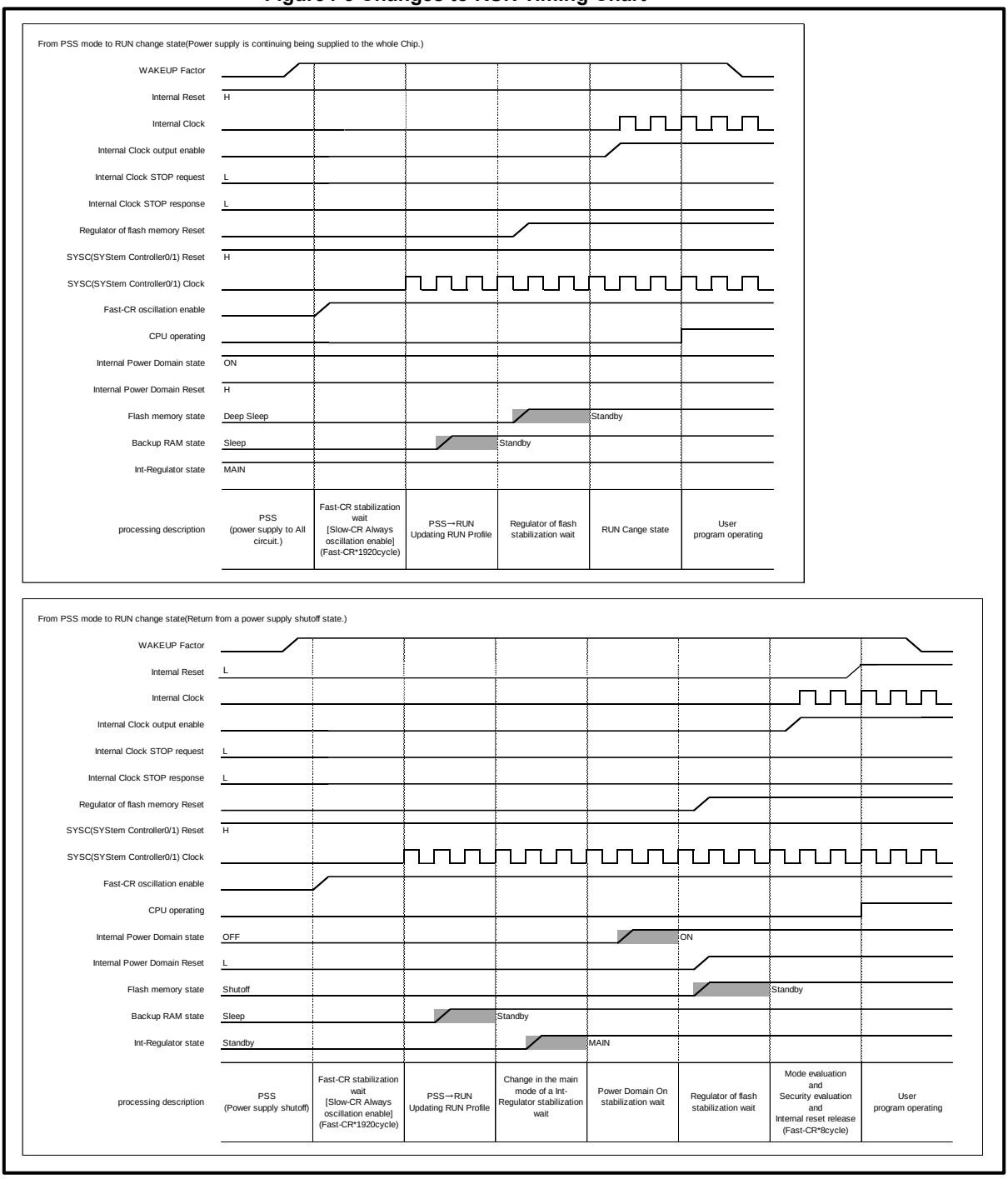


Figure4-3 Changes to RUN Timing Chart



CHAPTER 13: Pin Status in Each CPU State

This chapter explains pin status in each cpu state.

1. PIN STATUS IN EACH CPU STATE
2. Note

CODE: PIN_STATUS-JUPI-E2

1. Pin Status in Each CPU State

This section shows the pin status in each CPU state.

Power supply pin	Pin No.(176pin)	Pin No.(144pin)	Pin No.(100pin)	Pin No.(100pin)	Pin Name	GPORTEN Control	External Reset Factor 1			External Reset Factor 2				External Reset Factor 3		I/O Reset		CPU Sleep	PSS mode (*1, *9)	
							While Generating External Factor	After Releasing External Factor		While Generating External Factor	After Releasing External Factor					RSTX_IO_3V	RSTX_IO_35V			
							While Issuing Internal Reset	While Issuing Internal Reset	After Issuing Internal Reset (Before Setting GPORTEN)	Before Issuing Internal Reset	While Issuing Internal Reset	While Issuing Internal Reset	After Issuing Internal Reset (Before Setting GPORTEN)	While Issuing Internal Reset	After Issuing Internal Reset (Before Setting GPORTEN)	While Issuing Internal Reset	While Issuing Internal Reset		SYSC0_SPECFGR.PSSPADCTRL=0 (*2)	SYSC0_SPECFGR.PSSPADCTRL=1 (*2)
VCC33_1	2	2			P0_00	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	3	3			P0_01	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	4	4			P0_02	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	5	5			P0_03	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	6	6			P0_04	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	7	7			P0_05	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	8	8	2	2	P0_06	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	9				P3_00	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	10				P3_01	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	11				P3_02	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	14				P3_03	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	15				P3_04	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	16				P3_05	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	17				P3_06	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	18				P3_07	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	19	11	5	5	P0_07	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	20	12	6	6	P0_08	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	21	13	7	7	P0_09	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	22	14	8	8	P0_10	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	23	15	9	9	P0_11	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	24	16	10	10	P0_12	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	27	19	13	13	P0_13	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	28	20	14	14	P0_14	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	29	21	15	15	P0_15	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	30	22	16	16	P0_16	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	31	23	17	17	P0_17	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	34	26	20		P0_18	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	36	28	22		P0_19	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	37	29	23	21	P0_20	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	38	30	24		P0_21	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	39	31	25		P0_22	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	40	32	26	22	P0_23	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	43	35	29		P0_24	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	Hi-Z/Previous value retained	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	46	38	32	27	X1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	47	39	33	28	X0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	48	40	34	29	MODE	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	49	41	35	30	RSTX	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	50	42	36	31	JTAG_NTRST	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	51	43	37	32	JTAG_TDO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	52	44	38	33	JTAG_TDI	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	53	45	39	34	JTAG_TCK	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	54	46	40	35	JTAG_TMS	-	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
	56				P1_08	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	59				P1_09	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	60				P1_10	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	61				P1_11	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	62				P1_12	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	63				P1_13	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	64				P1_14	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	65				P1_15	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	66				P1_16	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	67				P1_17	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	68				P1_18	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	69	55	48	39	NMIX		Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled	Input enabled
VCC5	70	56	49	40	XOA / P0_30	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	71	57	50	41	XIA / P0_31	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	74	60	53	44	P1_00	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained (*7)	Hi-Z/Input Blocked (*3)
	75	61	54	45	P1_01	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	76	62	55		P1_02	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	77	63			P1_03	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	78	64			P1_04	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	79	65			P1_05	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	80	66			P1_06	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	81	67			P1_07	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained (*6)	Hi-Z/Input Blocked (*3)
	82				P3_14	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	83				P3_15	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	90				P3_16	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	91				P3_17	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	92				P3_18	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	93				P3_19	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	94				P3_20	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	95				P3_21	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained (*4)	Hi-Z/Input Blocked (*3)
	96	74			P1_08	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained (*7)	Hi-Z/Input Blocked (*3)
	97				P3_22	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	98				P3_23	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	101	77	64	54	P1_09	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained (*5,*6)	Hi-Z/Input Blocked (*3)
	104	80	67	57	P1_10	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	105	81	68	58	P1_11	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	106	82	69	59	P1_12	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	107	83	70	60	P1_13	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	108	84	71	61	P1_14	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	109	85	72	62	P1_15	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	110	86	73	63	P1_16	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Previous value retained (*8)	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Blocked	No influence	No influence	Last state retained	Last state retained	Hi-Z/Input Blocked (*3)
	111	87	74	64	P1_17	Controlled	Hi-Z/Input Blocked	Hi-Z/Input Blocked	Hi-Z/Input Bloc

The reset factor in this table is shown below.

- External Reset Factor 1
  - Power-on reset (PONR)
  - RSTX pin + MD pin simultaneous assert reset (INITX)
  - RAM retention low-voltage detection reset (RVD)
  - Internal power supply low-voltage detection reset (LVDL1R)
- External Reset Factor 2
  - RSTX pin input reset (RSTX)
  - External power supply low-voltage detection reset (LVDH1R)
  - Extended external power supply low-voltage detection reset (LVDH2R)
- External Reset Factor 3
  - Illegal mode detection reset (IMR)
  - Hardware watchdog reset (HWDR)
  - Software watchdog reset (SWDR)
  - Clock supervisor reset (CSVMOR, CSVSOR, CSVPRn, CSVSRn, CSVFCRR, CSVSCRR)
  - Profile error reset (PRFERR)
  - Software trigger hard reset (SHRST)
  - Software reset (SRST)
- I/O Reset
  - The pins located in VCC53_1 power domain are controlled by SYSC0_SPECFGR.IO3RSTC register. When SYSC0_SPECFGR.IO3RSTC=1, the status of these pins are Hi-Z/Previous value retained. (RSTX_IO_3V）
  - The pins located in VCC53_2 power domain are controlled by SYSC0_SPECFGR.IO35RSTC register. When SYSC0_SPECFGR.IO35RSTC=1, the status of these pins are Hi-Z/Previous value retained. (RSTX_IO_35V）

The followings show additional information for asterisked item in this table.

*1 Refer to the chapter of "State Transition" on this manual for the state definition of the PSS mode. To transition to the PSS mode with power domains 2 shut off, be sure to set HOLDIO_PD2 (SYSC_SSPECFGR.HOLDIO_PD2=1) before transition to the PSS mode.

*2 The pin state of the time when HOLDIO_PD2 was set (SYSC_SPECFGR.HOLDIO_PD2=1) is retained for the pin written as "Last state retained". If the power domain2 shutoff has not been issued and HOLDIO_PD2 has not been set (SYSC_SPECFGR.HOLDIO_PD2=0), the last state is retained.

*3 When the external interrupts are enabled, input is not blocked.

*4 When the PWU trigger output is enabled in the PSS-PWU mode, the pin state is PWU trigger output.

*5 When the RTC output is enabled in the PSS mode, the pin state is RTC output.

*6 When the Clock monitor output is enabled in the PSS mode, the pin state is Clock monitor output.

*7 When the Indicator PWM output is enabled in the PSS mode, the pin state is Indicator PWM output.

*8 When this external reset factor is LVDH2R, these pins are not influenced.

*9 The pin state is as the following table when the pins with SEG/COM are configured for LCD function.

PSS mode	Power supply shutoff	Power supply		
Operation mode	-	PSS STOP mode	PSS Timer mode	
Timer mode operation enable	-	-	Disabled (LCR0:LCEN=0)	Enabled (LCR0:LCEN=1)
Pin state	Low output / input blocked			LCD display continued

LCR0:LCEN change LCRS:LCSEN in static drive.

2. Note

No description.

CHAPTER 14: Low Voltage Detection

This chapter explains the function of low voltage detection.

1. Overview
2. Configuration of Supervised Power Domain
3. Operation
4. Registers

CODE: LVD-JUPI-E2

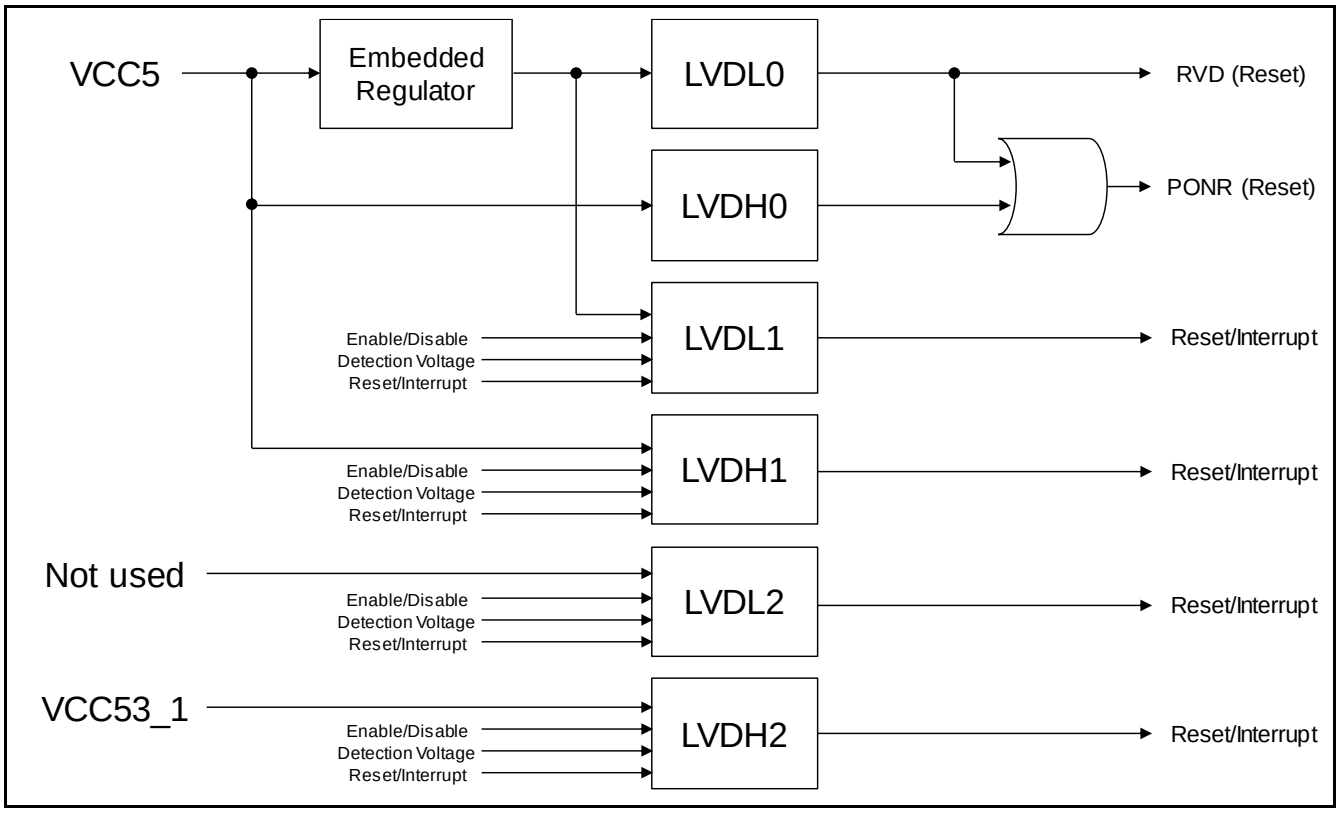
1. Overview

Low-Voltage Detection (LVD) gives the following functions.

- Power on reset (PONR) generation.
- RAM retention low-voltage detection reset (RVD) generation.
- Internal regulator output supervision.
- External power supply supervision.

2. Configuration of Supervised Power Domain

Figure 2-1 Supervised Power Domain



3. Operation

3.1. LVD Operation

Table 3-1 LVD Description

Channel	Monitor	Detected Voltage	Operation after Detection	Enable/Disable	Description (Purpose)
LVDL0	Embedded regulator output	Specific	Reset only	Always enable	RAM retention voltage supervision. Power on condition supervision.
LVDH0	VCC5	Specific	Reset only	Always enable	Power on condition supervision.
LVDL1	Embedded regulator output	Configurable	Reset or interrupt	Configurable	Internal power supply supervision.
LVDH1	VCC5	Configurable	Reset or interrupt	Configurable	Internal regulator input and 5V I/O power supply supervision.
LVDL2	N/A	-	-	-	-
LVDH2	VCC53_1	Configurable	Reset or interrupt	Configurable	5/3V I/O power supply supervision.

Notes:

- *PONR will be*
  - *generated by means of either LVDL0 or LVDH0 detected.*
  - *released by means of both LVDL0 and LVDH0 released.*
- *The voltage for both to detect and to release is specified in the datasheet.*

3.2. Configurations

Function enable/disable, detected voltage threshold, and operation option after low-voltage detection are selectable and configurable for LVDL1, LVDH1, and LVDH2 by software.

As for LVDL0 and LVDH0 anything cannot be configured by software. The functions are always enabled, detect specific voltage, and only generate reset.

The voltage range of detect and release for each is specified in datasheet.

Bit Name	Description
LVDL1S, LVDH1S, LVDH2S	0: Reset, 1: Interrupt
LVDL1V, LVDH1V, LVDH2V	Threshold voltage can be configured to the bit range.
LVDL1E, LVDH1E, LVDH2E	0: Disable, 1: Enable

4. Registers

See the RUN/PSS/APP/STS Low-voltage Detection setting Register on Traveo™ Platform hardware manual.

Here, configurable value and its operation are described.

[Bit30] LVDL1S Internal low-voltage detection voltage operation selection bit

Bit 30	Description
0	Reset (Initial value)
1	Interrupt

[Bit27:25] LVDL1V Internal low-voltage detection voltage setting bits

Bit 27:25	Voltage [V]
000 *	0.875(Initial value)
001 *	0.950
010	Not supported
011	Not supported
100	Not supported
101	Not supported
110	Not supported
111	Not supported

*: These detection voltage level settings are below the minimum operation voltage. Between these detection voltage and the minimum operation voltage, MCU functions are not guaranteed except for the low voltage detector.

Note that although the detection level is below the minimum operation guarantee voltage, the LVD reset factor flag is set as the voltage drops below the detection level.

[Bit24] LVDL1E Internal low-voltage detection operation enable bit

Bit 24	Description
0	STOP operation
1	Enable operation (Initial value)

Note:

- *LVDL1 corresponds to 1.2V power supply low-voltage detection ch.1*

[Bit14] LVDH1S Internal low-voltage detection voltage operation selection bit

Bit 14	Description
0	Reset (Initial value)
1	Not supported

[Bit11:9] LVDH1V External low-voltage detection voltage setting bits

Bit 11:9	Voltage [V]
000 *	2.70
001	2.80
010	3.60
011	3.80
100	4.00
101	4.20
110 *	2.50
111 *	2.60 (Initial value)

*: These detection voltage level settings are below the minimum operation voltage. Between these detection voltage and the minimum operation voltage, MCU functions are not guaranteed except for the low voltage detector.

Note that although the detection level is below the minimum operation guarantee voltage, the LVD reset factor flag is set as the voltage drops below the detection level.

The table shows the representative detected values. See the electric characteristics of LVD on datasheet in detail.

[Bit8] LVDH1E Internal low-voltage detection operation enable bit

Bit 8	Description
0	STOP operation
1	Enable operation (Initial value)

Note:

- *LVDH1 corresponds to 3.3/5.0V power supply low-voltage detection ch.1*

Note:

- *LVDH1 reset does not clear the LVDH1S, LVDH1V, LVDH1E configured value. The following factors clear the value to initial value.*
  - *Power-on reset*
  - *LVDL0(RAM maintenance low voltage detection reset)*
  - *LVDL1(Internal low voltage detection reset)*
  - *Illegal mode detection reset*
  - *INITX (RSTX pin + MD pin simultaneous assert reset)*

[bit6] LVDH2S: LVDH2 operation selection bit

Bit	Description
0	Reset
1	Interrupt (Initial value)

[bit4:1] LVDH2V: LVDH2 voltage setting bits

Bits	Voltage [V]
0000	Not supported
0001	2.80 (Initial value)
0010	3.60
0011	3.80
0100	4.00
0101	4.20
0110	Not supported
0111	Not supported

[bit0] LVDH2E: LVDH2 operation enable bit

Bit	Description
0	STOP operation (Initial value)
1	Enable operation

Note:

- *LVDH2 reset does not clear the LVDH2S, LVDH2V, LVDH2E configured value. The following factors clear the value to initial value.*
  - *Power-on reset*
  - *LVDL0(RAM maintenance low voltage detection reset)*
  - *LVDL1(Internal low voltage detection reset)*
  - *Illegal mode detection reset*
  - *INITX (RSTX pin + MD pin simultaneous assert reset)*

CHAPTER 15: Serial Programming

This chapter explains serial programming.

1. Overview
2. Memory Map
3. Flash Sector Configuration
4. Port Configuration
5. Operation
6. Note

CODE: SERIAL_PRG-JUPI-E1

1. Overview

This chapter describes the FLASH serial programming. Please refer Traveo™ Platform hardware manual if necessary.

2. Memory Map

See the chapter of “Memory and Base Address Map” on this hardware manual.

3. Flash Sector Configuration

See the chapter of “TCFLASH” and “WorkFLASH” of Traveo™ Platform hardware manual.

4. Port Configuration

Table 4-1: Port Configuration and Usage

Port Name	Configuration	Remark
MODE	Pull-down	-
RSTX	Reset input	-
X0	Oscillation input	-
X1	Oscillation output	-
P1_14/SIN0_0	Pull-up: Clock asynchronous mode Pull-down: Clock synchronous mode Used for a serial input function after reset operation.	-
P1_12/SOT0_0	Pull-up: necessary until reset release. Used for a serial output function after reset operation.	-
P1_13/SCK0_0	Used for synchronous mode.	-

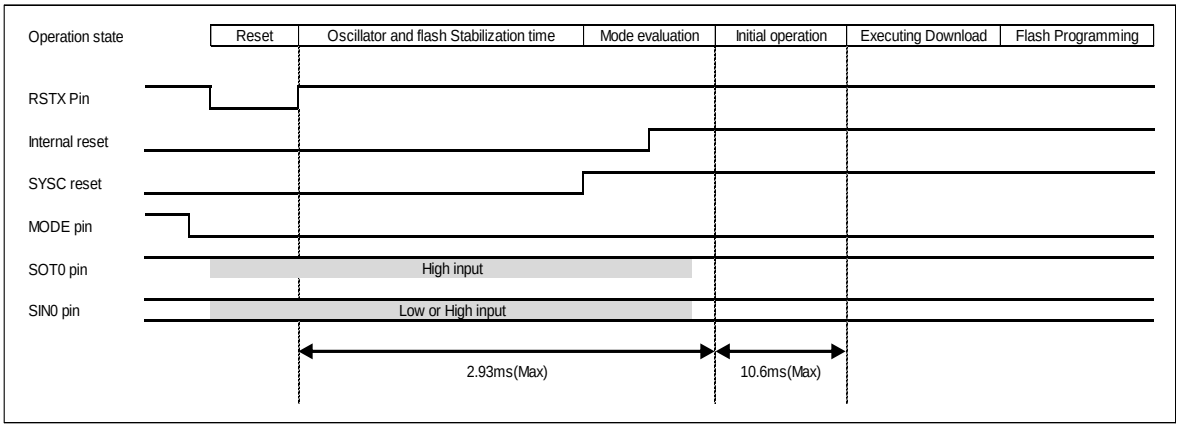
Notes:

- *See the chapter of "Port Description" on this manual" for Port name.*
- *See the "Pin Assignment" in Datasheet.*

5. Operation

5.1. Timing Chart

Figure 5-1



5.2. RAM Executing Communication Protocol

5.2.1. Command Format

5.2.1.1 Download Command

Byte Position	Byte Value	Explanation
1	00H	Download command
2	XXH	Download start address (A7-A0)
3	XXH	Download start address (A15-A8)
4	XXH	Download start address (A23-A16)
5	XXH	Download start address (A31-A24)
6	XXH	Download count number (BC7-BC0)
7	XXH	Download count number (BC15-BC8)
8	XXH	Download count number (BC23-BC16)
9	XXH	Download count number (BC31-BC24)
10	XXH	Sum value of download command(*1)

(*1) The SUM value is calculated in 8-bits, and the overflow of simple addition is ignored.

5.2.1.2 Executing Command

Byte Position	Byte Value	Explanation
1	C0H	Executing command
2	00H	Dummy data (For adjusting of command length)
3	00H	Dummy data (For adjusting of command length)
4	00H	Dummy data (For adjusting of command length)
5	00H	Dummy data (For adjusting of command length)
6	00H	Dummy data (For adjusting of command length)
7	00H	Dummy data (For adjusting of command length)
8	00H	Dummy data (For adjusting of command length)
9	00H	Dummy data (For adjusting of command length)
10	C0H	SUM value of executing command

5.2.1.3 Reset Command

Byte Position	Byte Value	Explanation
1	18H	Reset command

5.2.1.4 Full Chip Erase Command

Word Position	Word Value	Explanation
1	C0356EA5H	Key0
2	2E830596H	Key1
3	01A00000H	Key2

Word Position	Word Value	Explanation
4	F3A033EDH	Key3
5	370E6A51H	Key4
6	B0412000H	Key5
7	00000002H	Key6
8	AA805510H	Key7

5.2.2. Command Sequence

5.2.2.1 Download Command Sequence

Byte Count	Host (Tool) ⇒ Micom	Micom ⇒ Host (Tool)	
1 Byte	Receipt of down load command (00H)		
4 Byte	Receipt of download start address		
4 Byte	Receipt of download byte number		
1 Byte	SUM value of command		
1 Byte		Command response Normal end SUM malfunction Command malfunction	 : 01H : X2H : X4H
N Byte	Receipt of download data	Data input procedure (Little Endian) D7～D0 D15～D8 D23～D16 D31～D24	
1 Byte	SUM value of download data (*1)		
1 Byte		Download processing response	
		Busy Normal end SUM malfunction	: 00H : 01H : 02H

(*1) About downloaded N byte data, SUM value is calculated in 8-bits, and the overflow of simple addition is ignored.

5.2.2.2 Executing Command Sequence

Byte Count	Host (Tool) ⇒ Micom	Micom ⇒ Host (Tool)	
1 Byte	Receipt of executing command (C0H)		
8 Byte	Receipt of dummy data (00H)		
1 Byte	SUM value of the command (C0H)		
(1 Byte)		Command response (when SUM value is malfunction) (*1)	
		SUM malfunction	: C2H

(*1) Only when the SUM malfunction occurs, command response is returned. Command response is not returned except the SUM malfunction occurs. After receiving executing command, it jumps to the download start address specified by the download command if the SUM is not malfunctioning.

5.2.2.3 Reset Command Sequence

Byte Count	Host (Tool) ⇒ Micom	Micom ⇒ Host (Tool)	
1 Byte	Download Command (18H)		
1 Byte		Command response (*1) Normal end	: 11H

(*1) Response of reset command is always normal end. (11H)

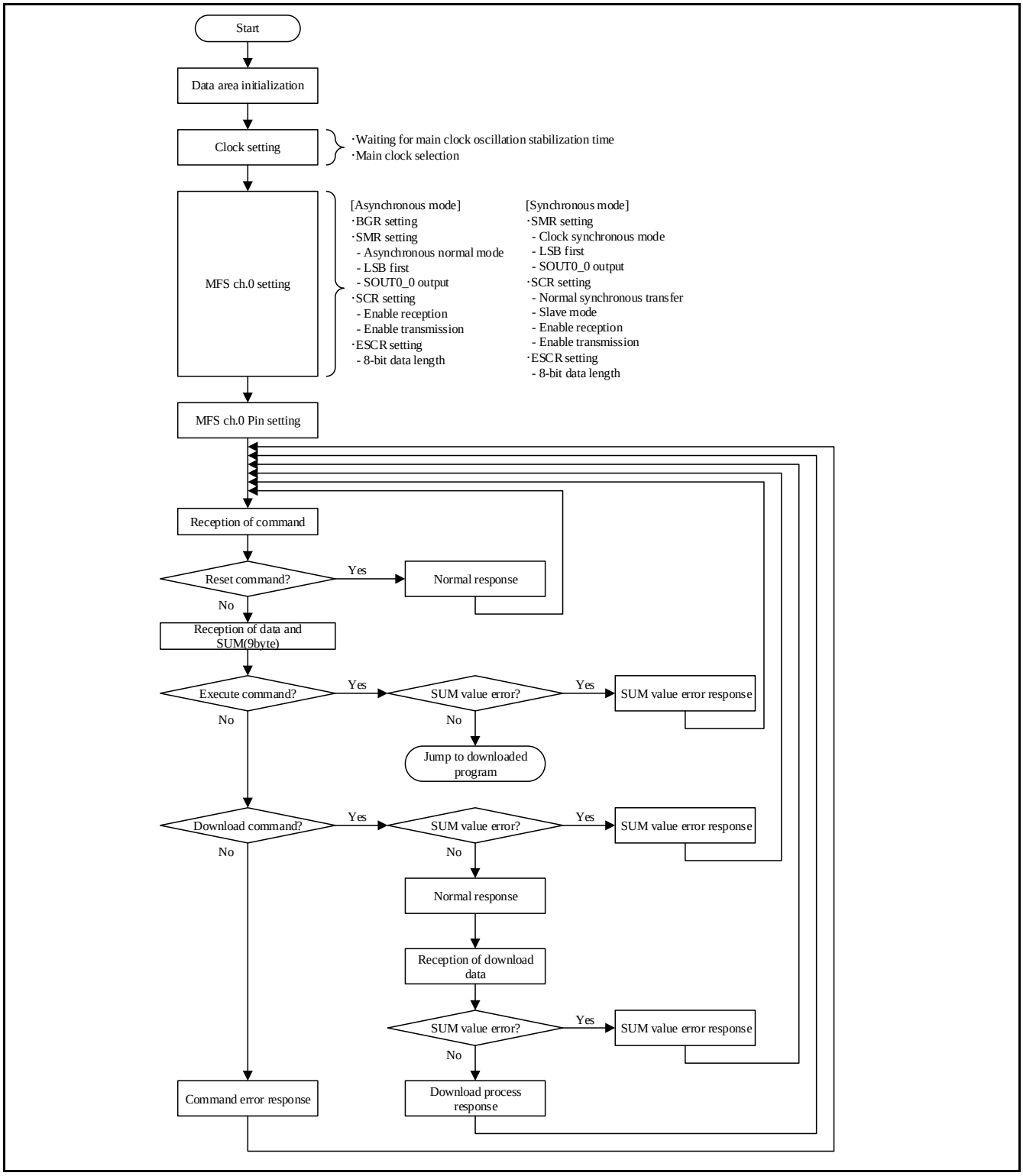
5.2.2.4 Full Chip Erase Command Sequence

Byte Count	Host (Tool) ⇒ Micom	Micom ⇒ Host (Tool)	
1-32 Byte	Key0-7Receive	-	
4 Byte	-	Command response (*1) Normal end	: 12345678H
4 Byte	-	Command response (*1) Normal end	: BABEFACEH

(*1) Command response is returned only when it is normal end. In other cases, hard reset is operated and command response is not returned.

5.3. Operation Flow

Figure 5-2: Operation Flow Chart



6. Note

See Traveo™ Platform hardware manual.

CHAPTER 16: Base Timer

This chapter explains the functions and operations of the Base Timer.

1. Overview of the Base Timer
2. Block Diagrams of the Base Timer
3. Operations of the Base Timer
4. 32-Bit Mode Operation
5. Debug Mode
6. Interrupts from the Base Timer
7. Start of DMA Controller (DMAC)
8. Registers of the Base Timer
9. Notes on Using the Base Timer
10. Base Timer Description by Function Mode

CODE:BT-JUPI-E3

1. Overview of the Base Timer

Only one of the following timer functions can be selected for the Base Timer in the FMD2 to FMD0 bit settings in the Timer Control Register (TMCR): Reset mode, 16-bit PWM Timer, 16-bit PPG Timer, 16-/32-bit Reload Timer, and 16-/32-bit PWC Timer. This section provides an overview of the various selectable timers.

Relationship Between Mode Settings and Various Timer Functions

FMD2 to FMD0 Bit Settings in the Timer Control Register (TMCR)	Function
000	Reset mode
001	16-bit PWM Timer
010	16-bit PPG Timer
011	16/32-bit Reload Timer
100	16/32-bit PWC Timer

Reset Mode

Reset mode is the state in which the Base Timer macros have been reset (to the initial values in each register). To use another timer function or switch the T32 bit setting, first enter this mode and then set the other timer function or the T32 bit. However, after a macro reset, it is possible to set a timer function and the T32 bit without entering this mode.

16-Bit PWM Timer

The 16-bit PWM Timer operates in 16-bit periods.

This timer consists of a 17-bit down counter that takes into account the start delay period, a 16-bit data register with a cycle setting buffer, a 16-bit compare register with a duty setting buffer, a Start Delay Value Setting Register (PSDR), an ADC Trigger Value Setting Register (ADTR), and a pin controller.

The down counter will be described in 16 bit formats when the 16-bit PWM Timer is irrelevant to the start delay control.

The registers with buffers store the cycle and duty data, enabling rewriting while the timer is operating.

The counter clock of the 17-bit down counter can be selected from 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

The start of the 16-bit PWM Timer can be selected from a software trigger and 3 types of external events (rising-edge, falling-edge, and both-edges detection).

The start delay function can delay the PWM control start time after trigger input.

An ADC trigger signal is output when the ADC Trigger Value Setting Register (ADTR) matches the count value of the 16-bit down counter.

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

16-Bit PPG Timer

This timer consists of a 16-bit down counter, a 16-bit data register for the "H" width setting, a 16-bit data register for the "L" width setting, and a pin controller.

The count clock of the 16-bit down counter can be selected from 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

The start of the 16-bit PPG Timer can be selected from a software trigger and 3 types of external events (rising-edge, falling-edge, and both-edges detection).

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

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 12 types of internal clocks (internal clock divided by 1/2/4/8/16/32/64/128/256/512/1024/2048) and 3 types of external clocks (rising-edge, falling-edge, and both-edges detection).

One-shot mode and continuous mode can be selected. In one-shot mode, counting stops when an underflow occurs. In continuous mode, counting is repeated following a reload.

An external factor pin can be selected as a trigger input function or a gate function when the 16/32-bit Reload Timer is operating.

If the trigger input function is selected, the start of the 16/32-bit Reload Timer can be selected from a software trigger and 3 types of external events (rising-edge, falling-edge, and both-edges detection).

If the gate function is selected, the 16/32-bit Reload Timer counts only while a valid level is being input to the external factor pin.

The external timer match starting can start a timer, when a match of the value set as the External Timer Compare Data Register (ETCDR) and an external timer value is detected.

16-/32-Bit PWC Timer

This timer consists of a 16-bit up counter, measurement input pins, and a control register.

With the input of external pulses, the timer measures the time between events.

The reference count clock can be selected from 12 types of internal clocks (divided by 1/2/4/8/16/32/64/128/256/512/1024/2048).

The supported measurement modes:

- "H" pulse width (rising to falling) / "L" pulse width (falling to rising)
- Rising cycle (rising to rising) / falling cycle (falling to falling)

- Edge-to-edge measurement (rising or falling to falling or rising)

An interrupt request can be generated at the measurement end time.

The measurement is 1-time only or continuous. Either can be selected.

2. Block Diagrams of the Base Timer

Figure 2-1 to Figure 2-4 are block diagrams of the Base Timer in each mode.

Figure 2-1 Block Diagram of the 16-Bit PWM Timer

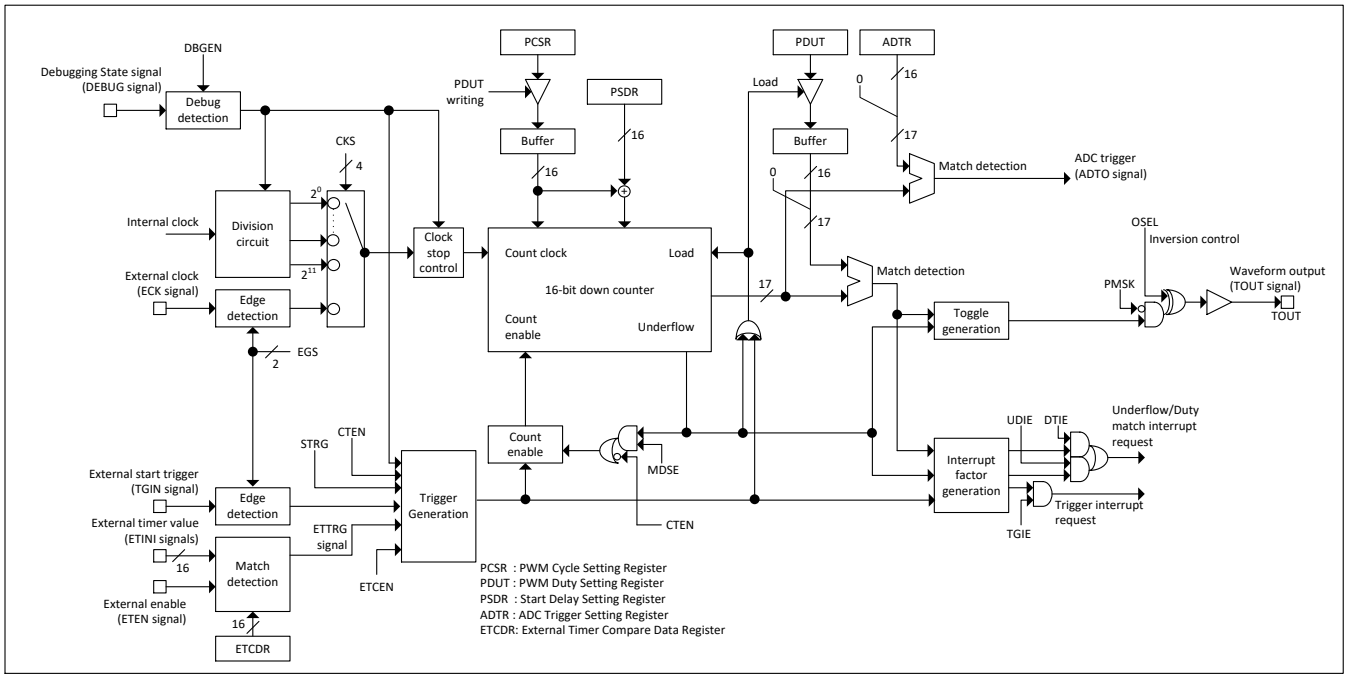


Figure 2-2 Block Diagram of the 16-Bit PPG Timer

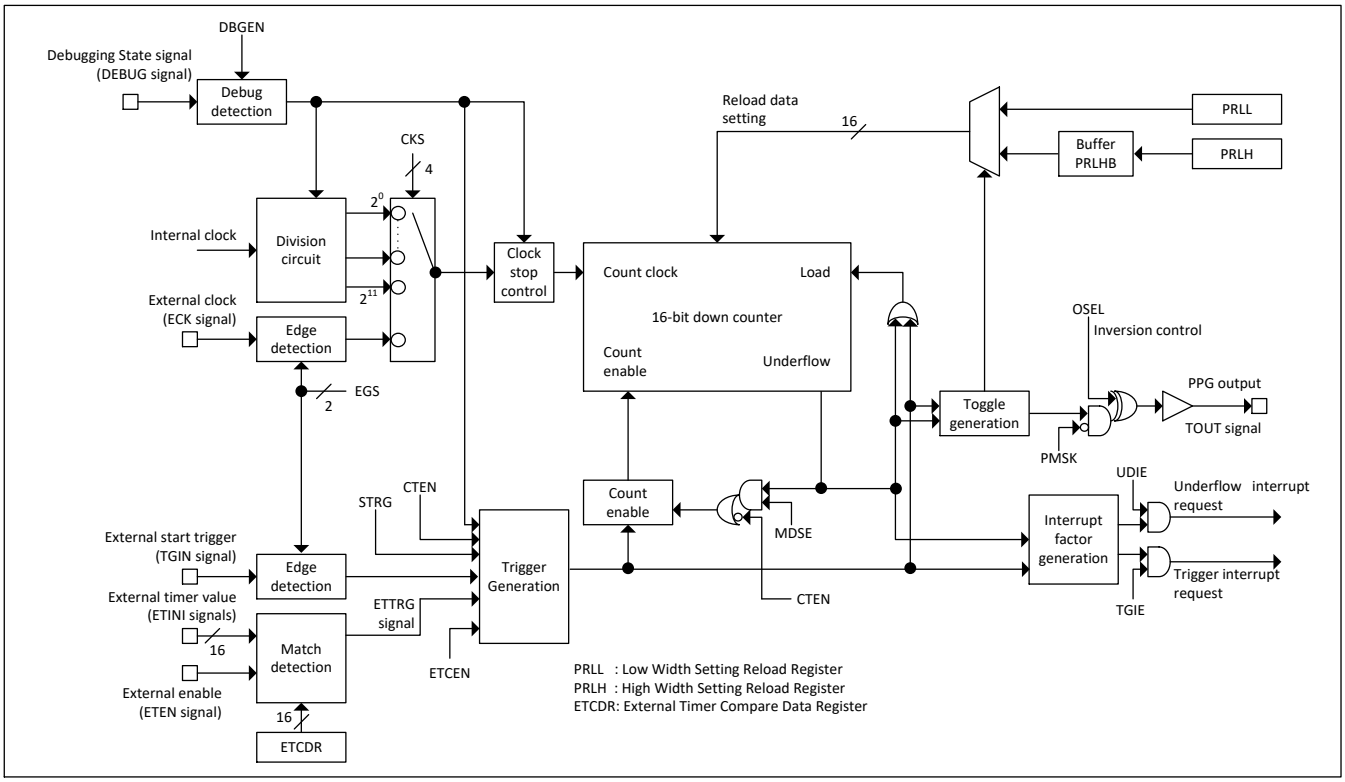


Figure 2-3 Block Diagram of the 16-/32-Bit Reload Timer (ch1, ch0)

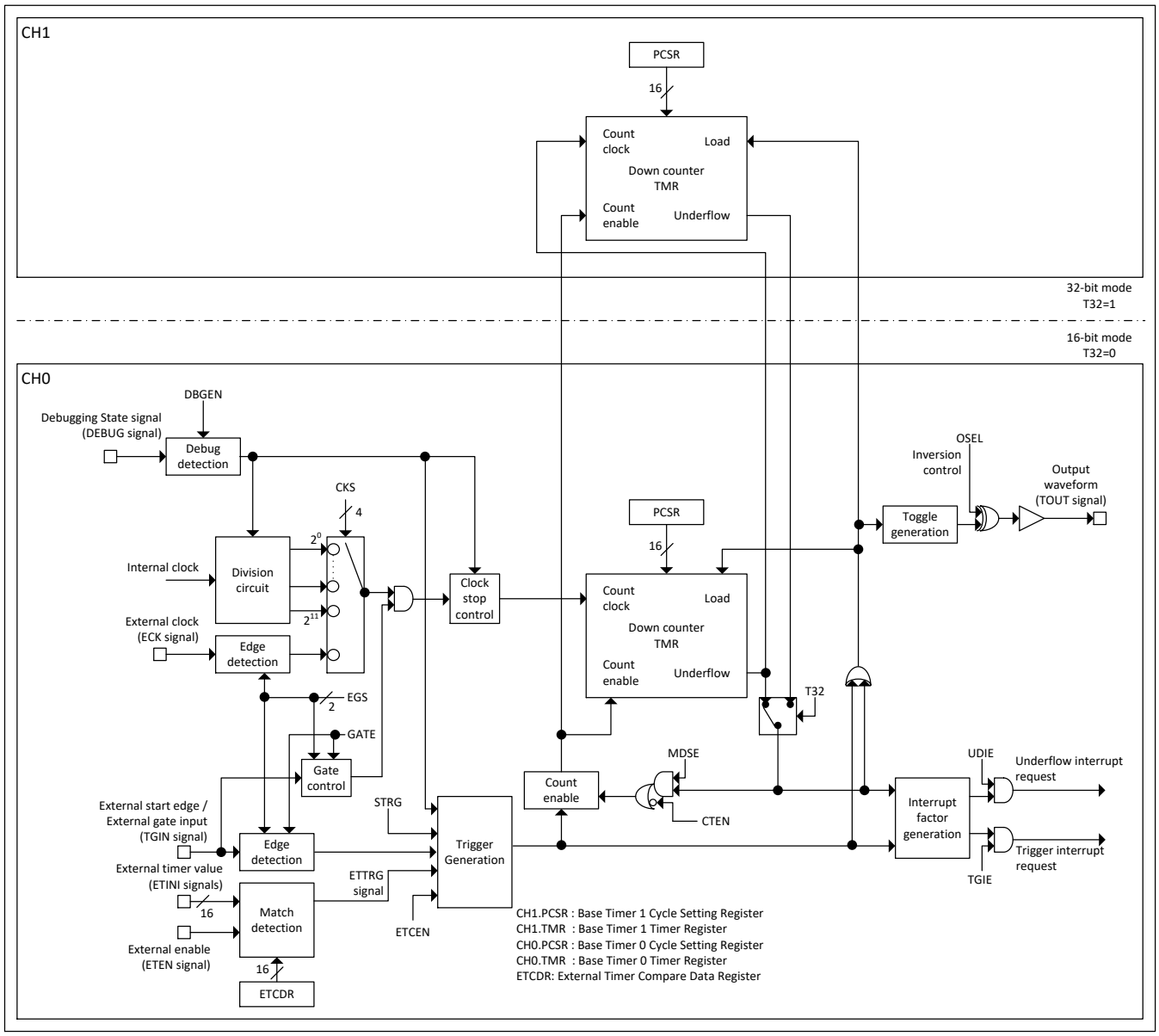
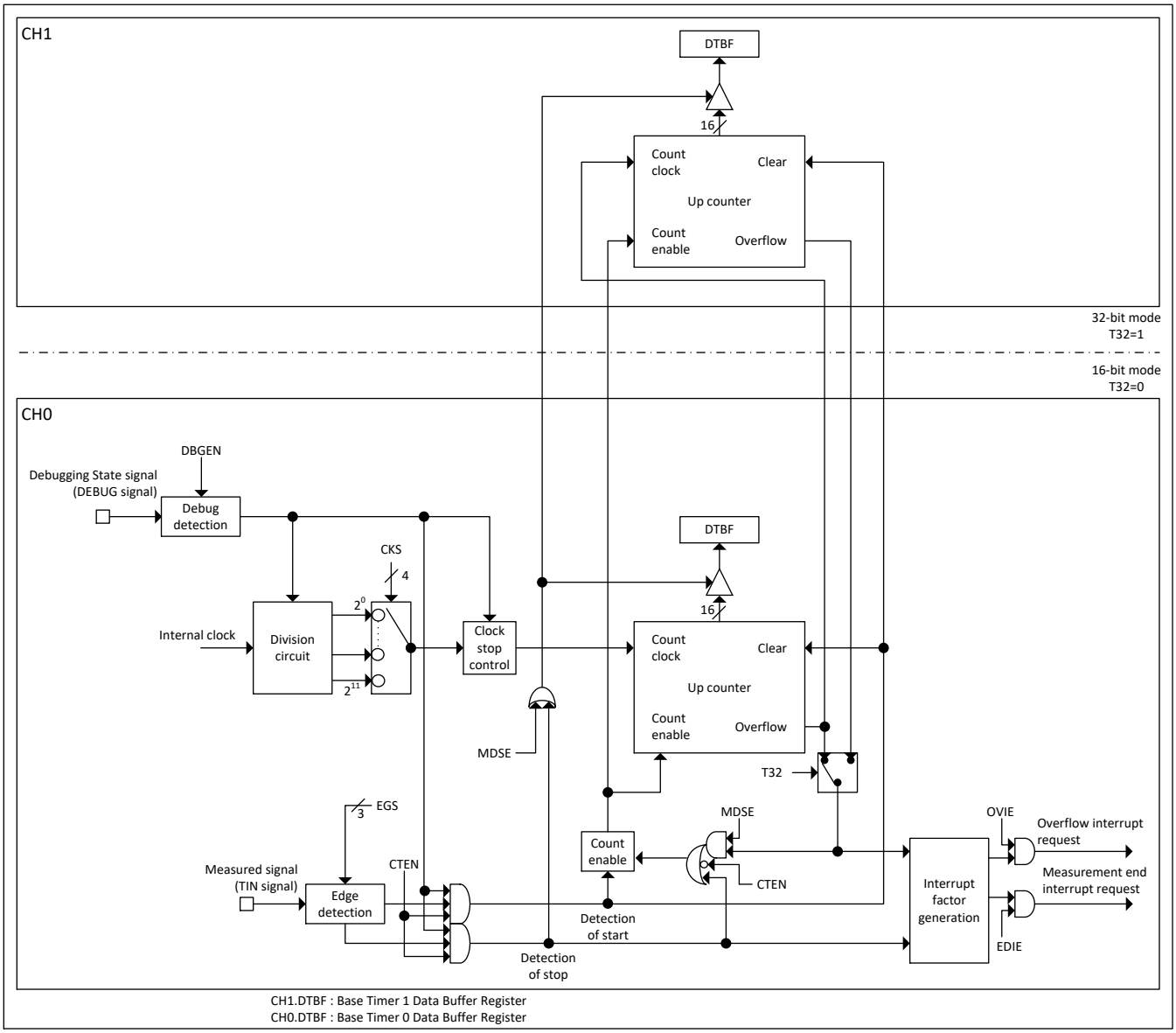


Figure 2-4 Block Diagram of the 16-/32-Bit PWC Timer (ch1, ch0)



3. Operations of the Base Timer

This section explains the operations of the Base Timer.

Operations of Base Timer

Reset Mode

Reset mode is the state in which the Base Timer macros have been reset (to the initial values in each register). To use another timer function or switch the T32 bit setting, first enter this mode and then set the other timer function or the T32 bit. However, after a macro reset, it is possible to set a timer function and the T32 bit without entering this mode. If this mode is set for the even-numbered channel when 32-bit mode is set, the odd-numbered channel is reset at the same time, so Reset mode need not be set for the odd-numbered channel.

16-Bit PWM Timer

The 16-bit PWM Timer starts counting down from the set cycle value upon trigger activation. In the beginning the output level is "L". If the 16-bit down counter matches the set value in the duty setting register, the output is inverted to the "H" level. Then, the output is inverted to the "L" level again when the counter underflows. Thus, this timer can generate a waveform with an arbitrary cycle and duty.

16-Bit PPG Timer

The 16-bit PPG Timer starts counting down from the set value in the Low Width Setting Reload Register (PRLL) upon trigger activation. In the beginning the output level is "L". The output is inverted to the "H" level when the counter underflows. Subsequently, the counter starts counting down from the set value in the High Width Setting Reload Register (PRLH). The output is inverted to the "L" level when the counter underflows. Thus, this timer can generate a waveform with an arbitrary "L" width and "H" width.

16-Bit Reload Timer

The 16-bit Reload Timer starts counting down from the set cycle value upon trigger activation. An interrupt flag is set to "1" when the 16-bit down counter underflows. The output level is defined by the MDSE bit setting - the polarity of the output is toggled each time an underflow is detected, or, the output is set to "H" when counting starts and switches to "L" when an underflow is detected.

32-Bit Reload Timer

With the same basic operation as the 16-bit Reload Timer, this timer uses two channels, an even-numbered channel and an odd-numbered channel, to operate as a 32-bit Reload Timer. The even-numbered channel performs the lower 16-bit timer operations, and the odd-numbered channel performs the upper 16-bit timer operations. The interrupt controller and output waveform control conform to the settings of only the even-numbered channel. To set a cycle, write it first to the upper register (odd-numbered channel) and then to the lower register (even-numbered channel).

To read the timer value, read it first from the lower register (even-numbered channel) and then from the upper register (odd-numbered channel).

16-Bit PWC Timer

The PWC Timer starts the 16-bit up counter upon detection of the measurement start edge, and stops the counter upon the detection of a measurement end edge. The resulting count value will be stored as the measured pulse width in the Data Buffer Register (DTBF).

32-Bit PWC Timer

With the same basic operation as the 16-bit PWC Timer, this timer uses two channels, an even-numbered channel and an odd-numbered channel, to operate as a 32-bit PWC Timer. The even-numbered channel performs the lower 16-bit count operations, and the odd-numbered channel performs the upper 16-bit count operations. The interrupt controller conforms to the settings of only the even-numbered channel. To read a measurement value or count value, read it first from the lower register (even-numbered channel) and then from the upper register (odd-numbered channel).

4. 32-Bit Mode Operation

The Reload Timer and PWC are capable of 32-bit mode operation using two channels. This section shows the basic functions/operations of the 32-bit mode function.

32-Bit Mode Function

This function realizes the operation of the 32-bit data Reload Timer or 32-bit data PWC Timer by using two Base Timer channels. The value of the timer or counter in operation can also be read when the lower 16-bit timer or counter value of the even-numbered channel is read. This is because the upper 16-bit timer or counter value of the odd-numbered channel is also fetched at the time.

32-Bit Mode Settings

First, reset the state by setting the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to "000" for Reset mode. Then, in the same way as in 16-bit mode, make Reload Timer or PWC Timer selection and operation settings. To set 32-bit operation mode at this time, write "1" to the T32 bit in the TMCR register. Keep the T32 bit of the odd-numbered channel at "0". There is no need to set the odd-numbered channel to Reset mode. For the Reload Timer, set the upper 16-bit reload value of the 32 bits in the Cycle Setting Register (PCSR) of the odd-numbered channel. Then, set the lower 16-bit reload value in the Cycle Setting Register (PCSR) of the even-numbered channel.

After the writing of the T32 bit, the change to 32-bit operation mode is reflected immediately, so change the settings for both channels while the counter is in a stopped state.

To change from 32-bit mode to 16-bit mode, set the FMD2 to FMD0 bits in the TMCR register of the even-numbered channel to "000" for Reset mode. This resets the states of both the even-numbered and odd-numbered channels. As a result, the settings for 16-bit mode can be made for each of the channels.

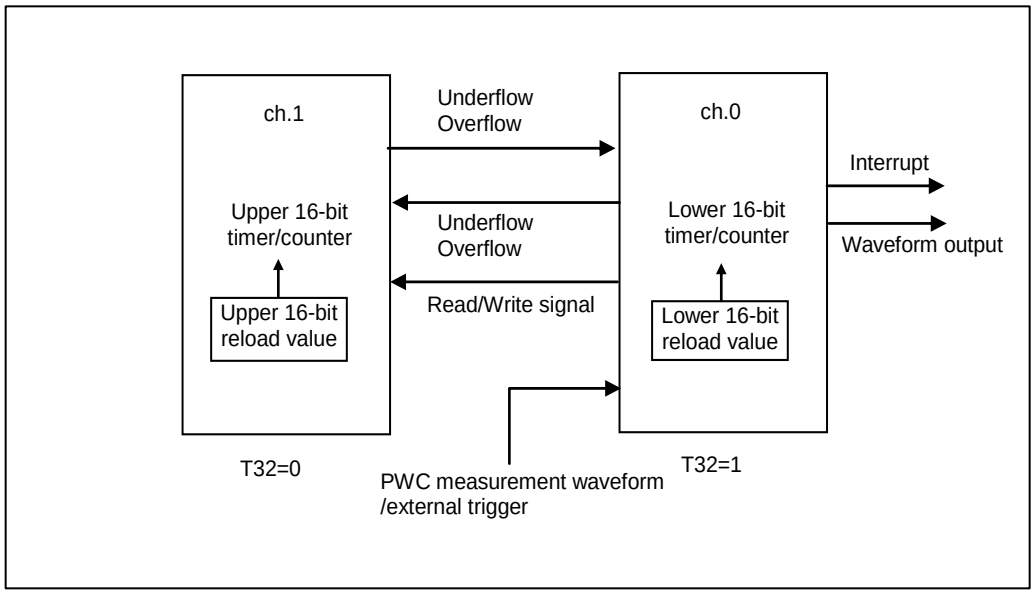
32-Bit Mode Operation

After 32-bit mode is set, if the Reload Timer or PWC Timer is started with even-numbered channel control, the even-numbered channel timer/counter performs the lower 16-bit operations. The odd-numbered channel timer/counter performs the upper 16-bit operations.

Operation in 32-bit mode conforms to the settings made for the even-numbered channel. Thus, the settings made for the odd-numbered channel (excluding those in the Cycle Setting Register (PCSR) at the Reload Timer time) are ignored. The timer start, waveform output, and interrupt signals of the even-numbered channel are valid. (The corresponding signals for the odd-numbered channel will be masked to "L".)

Figure 4-1 shows the configuration of ch.0 and ch.1.

Figure 4-1 Configuration of 32-Bit Mode Operation (for ch.0 and ch.1)



5. Debug Mode

If it goes into a Debug mode, operation of the Base Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the Debug mode.

- The debug enable bit (DBGEN) of the Base Timer Debug Register (BT_DEBUG) is "1".
- The debug signal (DEBUG) is active.

It becomes the following operation when it goes into the Debug mode.

- The prescaler stop.
- The timer stop.
- The output waveform stop.
  - It holds the output value at the time of the stop.
- The trigger is ignored in PWM Timer, PPG Timer or Reload timer.
- The measurement edge is ignored in PWC Timer.

It becomes the following operation when it restores from the Debug mode.

- The prescaler is resumed from the stopped value.
- The timer is resumed from the stopped value.
- The output waveform resumes the output from the stopped state.

Notes:

- *Register access is possible in the Debug mode.*
- *When the processor is in the debug state, the debug signal (DEBUG) becomes active. For the definition of debug state, refer to Section 12.8 of the Arm Cortex-R5 Technical Reference Manual.*

6. Interrupts from the Base Timer

This section shows a list summarizing the interrupt request flags, interrupt enable bits, and interrupt factors in each function of the Base Timer.

Interrupt Controller Bits and Interrupt Factors of Each Function

Table 6-1 shows the interrupt controller bits and interrupt factors of each function.

Table 6-1 Interrupt Controller Bits and Interrupt Factors in Each Mode

	Status Control Register (STC)			
	Interrupt Request Flag Bit	**Interrupt Request Enable Bit**	**Interrupt Factor**	**Interrupt Factor Output Signal**
PWM Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	DTIR : bit1	DTIE : bit5	Detection of duty match	
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
PPG Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
Reload Timer Function	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ0
	TGIR : bit2	TGIE : bit6	Detection of timer activation trigger	IRQ1
PWC Timer Function	OVIR : bit0	OVIE : bit4	Detection of overflow	IRQ0
	EDIR : bit2	EDIE : bit6	Detection of measurement end	IRQ1

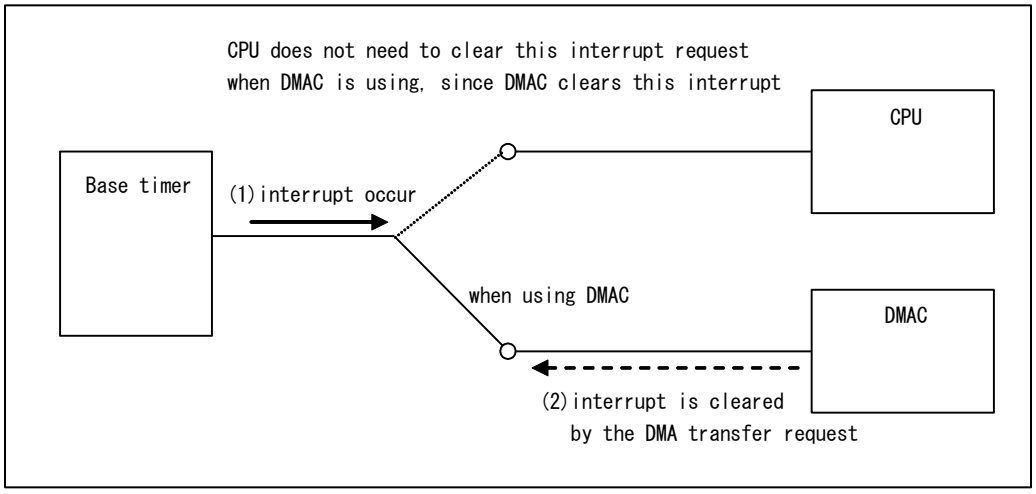
7. Start of DMA Controller (DMAC)

The generation of a Base Timer interrupt request can be used to start the DMAC.

DMA Transfer Operation Using Base Timer Interrupt Factor

The generation of a Base Timer interrupt factor can be used to start the DMAC. Figure 7-1 shows an overview of DMAC start with the Base Timer.

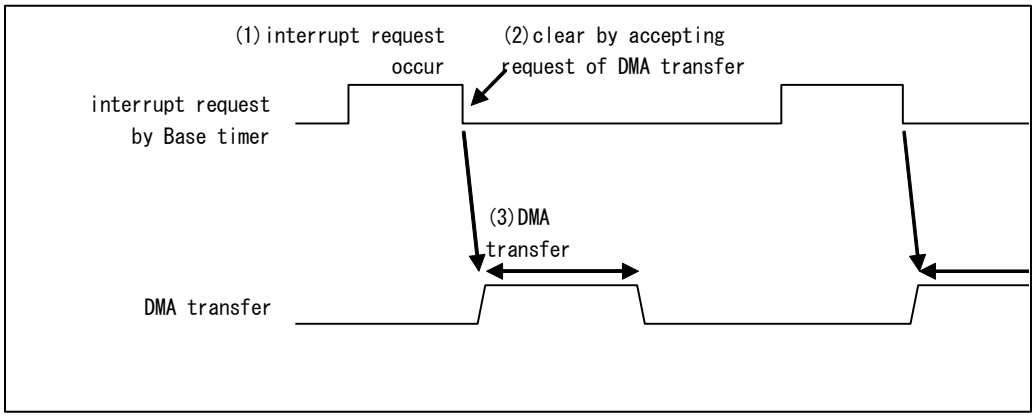
Figure 7-1 Overview of DMAC Start with the Base Timer



Configure the DMAC before starting it with the Base Timer. For details on DMAC settings, see the chapter of "DMA Controller" in Traveo™ Platform hardware manual and "6. Interrupts from the Base Timer" on this manual.

Figure 7-2 shows an example of a DMA transfer operation with a Base Timer interrupt request.

Figure 7-2 DMA Transfer Operation Example



8. Registers of the Base Timer

This section lists the registers of each mode of the Base Timer.

List of Registers when 16-Bit PWM Timer is Selected

Table 8-1 List of Registers When the 16-Bit PWM Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.1.9
TMCR2	Timer Control Register 2	10.1.9
STC	Status Control Register	10.1.9
STCC	Status Control Clear Register	10.1.9
STCS	Status Control Set Register	10.1.9
PCSR	PWM Cycle Setting Register	10.1.10
PDUT	PWM Duty Setting Register	10.1.11
TMR	Timer Register	10.1.12
PSDR	Start Delay Value Setting Register	10.1.13
ADTR	ADC Trigger Value Setting Register	10.1.14
BT_DEBUG	Base Timer Debug Register	10.1.15
ETCDR	External Timer Compare Data Register	10.1.16

List of Registers when 16-Bit PPG Timer is Selected

Table 8-2 List of Registers When the 16-Bit PPG Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.2.7
TMCR2	Timer Control Register 2	10.2.7
STC	Status Control Register	10.2.7
STCC	Status Control Clear Register	10.2.7
STCS	Status Control Set Register	10.2.7
PRLL	Low Width Setting Reload Register	10.2.8
PRLH	High Width Setting Reload Register	10.2.9
TMR	Timer Register	10.2.10
BT_DEBUG	Base Timer Debug Register	10.2.11
ETCDR	External Timer Compare Data Register	10.2.12

List of Registers when Reload Timer is Selected

Table 8-3 List of Registers When the Reload Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.3.4
TMCR2	Timer Control Register 2	10.3.4
STC	Status Control Register	10.3.4
STCC	Status Control Clear Register	10.3.4
STCS	Status Control Set Register	10.3.4

Abbreviated Register Name	Register Name	Reference
PCSR	Cycle Setting Register	10.3.5
TMR	Timer Register	10.3.6
BT_DEBUG	Base Timer Debug Register	10.2.11
ETCDR	External Timer Compare Data Register	10.2.12

List of Registers when PWC Timer is Selected

Table 8-4 List of Registers When the PWC Timer Is Selected

Abbreviated Register Name	Register Name	Reference
TMCR	Timer Control Register	10.4.2
TMCR2	Timer Control Register 2	10.4.2
STC	Status Control Register	10.4.2
STCC	Status Control Clear Register	10.4.2
STCS	Status Control Set Register	10.4.2
DTBF	Data Buffer Register	10.4.3
BT_DEBUG	Base Timer Debug Register	10.2.11

Table 8-5 Offset Addresses

CH No.	Offset_Address (Common PERI #1)	CH No.	Offset_Address (Common PERI #1)	CH No.	Offset_Address (Common PERI #1)
24	0x0084_6000	12	0x0088_8000	0	0x0080_8000
25	0x0084_6400	13	0x0088_8400	1	0x0080_8400
26	0x0084_6800	14	0x0088_8800	2	0x0080_8800
27	0x0084_6C00	15	0x0088_8C00	3	0x0080_8C00
28	0x0084_7000	16	0x0088_9000	4	0x0080_9000
29	0x0084_7400	17	0x0088_9400	5	0x0080_9400
30	0x0084_7800	18	0x0088_9800	6	0x0080_9800
31	0x0084_7C00	19	0x0088_9C00	7	0x0080_9C00
32	0x0084_8000	20	0x0088_A000	8	0x0080_A000
33	0x0084_8400	21	0x0088_A400	9	0x0080_A400
34	0x0084_8800	22	0x0088_A800	10	0x0080_A800
35	0x0084_8C00	23	0x0088_AC00	11	0x0080_AC00

CH No.	Offset_Address (Common PERI #1)	CH No.	Offset_Address (Common PERI #1)	CH No.	Offset_Address (Common PERI #1)
60	0x0084_F000	48	0x0084_C000	36	0x0084_9000
61	0x0084_F400	49	0x0084_C400	37	0x0084_9400
62	0x0084_F800	50	0x0084_C800	38	0x0084_9800
63	0x0084_FC00	51	0x0084_CC00	39	0x0084_9C00
-	-	52	0x0084_D000	40	0x0084_A000
-	-	53	0x0084_D400	41	0x0084_A400
-	-	54	0x0084_D800	42	0x0084_A800
-	-	55	0x0084_DC00	43	0x0084_AC00
-	-	56	0x0084_E000	44	0x0084_B000
-	-	57	0x0084_E400	45	0x0084_B400
-	-	58	0x0084_E800	46	0x0084_B800
-	-	59	0x0084_EC00	47	0x0084_BC00

Table 8-6 Register Map (Channel No.: 0) (12) (24) (36) (48) (60)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/ Reserved 00000000_00000000 XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSELmn*2 11110000
0x0000_0034	Reserved 11111111_11111111		BT_BTSSSRn*3 11111111_11111111	
0x0000_0038	Reserved 11111111_11111111		BT_BTTRRn*3 11110000_00000000	

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

*2 mn = 01, 1213, 2425, 3637, 4849 and 6061

*3 n = 0, 12, 24, 36, 48 and 60

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 8-7 Register Map (Channel No.: 1, 3, 5, 7, 9, and 11) (13, 15, 17, 19, 21, and 23) (25, 27, 29, 31, 33, and 35)(37, 39, 41, 43, 45, and 47) (49, 51, 53, 55, 57, and 59) (61 and 63)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR / Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 8-8 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34) (38, 40, 42, 44, and 46) (50, 52, 54, 56, and 58) (62)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR / Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSELmn*2 11110000

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

*2 mn = 23, 45, 67, 89, 1011, 1415, 1617, 1819, 2021, 2223, 2627, 2829, 3031, 3233, 3435, 3839, 4041, 4243, 4445, 4647, 5051, 5253, 5455, 5657, 5859 and 6263

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

9. Notes on Using the Base Timer

This section explains precautions on use of the Base Timer.

Notes to observe when accessing register

Status Control Register (STC) Access

- To clear a specific bit in this register, write "1" to the corresponding bit in the Status Control Clear Register (STCC).
- To set a specific bit in this register, write "1" to the corresponding bit in the Status Control Set Register (STCS).
- Direct writing to this register is possible only during writing to all bits.

Notes to observe on configuration using program, common to use of timers

- Rewriting of the following bits in the TMCR2 register and TMCR register during operation is prohibited. Be sure that such rewriting is done either before the timer starts or after it stops.

Bit	Bit Name	Description
[TMCR2 bit0], [TMCR bit14,13,12]	CKS3 to CKS0	Clock selection bits
[bit10,9,8]	EGS2, EGS1, EGS0	Measurement edge selection bits
[bit7]	T32	32-bit timer selection bit (when the Reload Timer and PWC function are selected)
[bit6,5,4]	FMD2 to FMD0	Timer function selection bits
[bit2]	MDSE	Measurement mode (single/continuous) selection bit

- All the registers of the Base Timer are initialized when the FMD2 to FMD0 bits in the TMCR register are set to "000" for Reset mode. For this reason, all the registers must be reconfigured.
- The settings for bits other than the FMD2 to FMD0 bits in the TMCR register are ignored and initialized when the FMD2 to FMD0 bits in the TMCR register are set to "000" for Reset mode.

Notes on using the 16-bit PWM/PPG/Reload Timer

- If the interrupt request flag set timing and clear timing overlap, the flag set has priority, and the clear operation is disabled.
- If the load timing and count timing overlap, the load operation has priority for the down counter.
- After configuring the timer function with the FMD2 to FMD0 bits in the TMCR register, set the cycle, duty, "H" width, and "L" width.
- If the Base Timer is configured in one-shot mode and is restarted when count end is detected, the counter will reload the count value and restart.

Notes on using the PWC Timer

- If "1" is written to the count start enable bit (CTEN), the counter is cleared. Thus, any data in the counter before the start is enabled is invalid.
- If the PWC mode setting (FMD2 to FMD0 = "100") and the measurement start setting (CTEN="1") are set simultaneously after a system reset assertion or upon leaving Reset mode, the PWC Timer operation may vary due to the state of the immediately preceding measurement signal.
- When operating in continuous measurement mode, if a measurement start edge is detected at the same time that the Base Timer has been restarted, the timer starts counting immediately from 0x0001.
- If a restart is performed after a count operation begins, operations such as the following may occur, depending on the timing the restart has been performed.
  - For a restart at the same time as a measurement end edge in pulse-width single measurement mode

The restart is performed and the measurement start edge wait state begins, but the measurement end flag (EDIR) is set to "1".

  - For a restart at the same time as a measurement end edge in pulse-width continuous measurement mode

The restart is performed and the measurement start edge wait state begins, but the measurement end flag (EDIR) is set to "1" and the measurement results at that point are transferred to DTBF.

Note the operation of the flag when using the interrupt controller, etc. at the restart time during operation as described above.

10. Base Timer Description by Function Mode

This section explains each function of the Base Timer.

Functions of the Base Timer

10.1 PWM Timer Function

10.2 PPG Timer Function

10.3 Reload Timer Function

10.4 PWC Timer Function

10.1. PWM Timer Function

Only one of the following timer functions can be selected for the Base Timer in the FMD2 to FMD0 bit settings in the Timer Control Register (TMCR): 16-bit PWM Timer, 16-bit PPG Timer, 16-/32-bit Reload Timer, and 16-/32-bit PWC Timer. This section explains the timer function with the PWM setting.

10.1.1 16-Bit PWM Timer Operation
10.1.2 One-Shot Operation
10.1.3 Start Delay Function
10.1.4 ADC Trigger Output Timing
10.1.5 External Timer Match Starting
10.1.6 Interrupt Factors and Timing Chart
10.1.7 Output Waveform
10.1.9 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.1.10 PWM Cycle Setting Register (PCSR)
10.1.11 PWM Duty Setting Register (PDUT)
10.1.12 Timer Register (TMR)
10.1.13 Start Delay Value Setting Register (PSDR)
10.1.14 ADC Trigger Value Setting Register (ADTR)
10.1.15 Base Timer Debug Register (BT_DEBUG)
10.1.16 External Timer Compare Data Register (ETCDR)

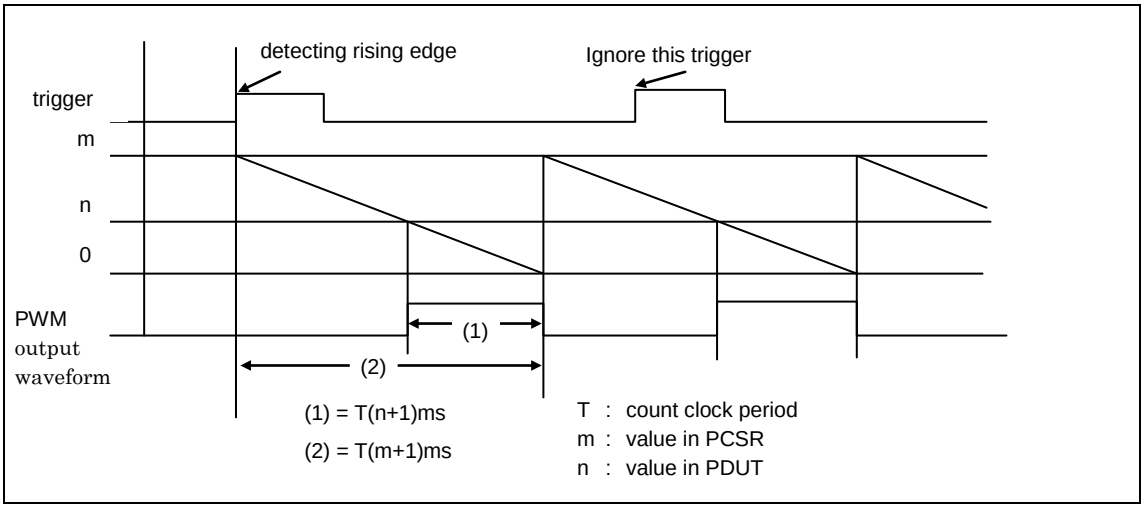
10.1.1. 16-Bit PWM Timer Operation

In PWM operation, waveforms of the set cycle can be output either singly or continuously upon the detection of a trigger. The cycle of the output pulse can be controlled through PCSR value changes. The duty ratio can be controlled through PDUT value changes. After writing data to PCSR, be sure to write to PDUT.

Continuous Operation

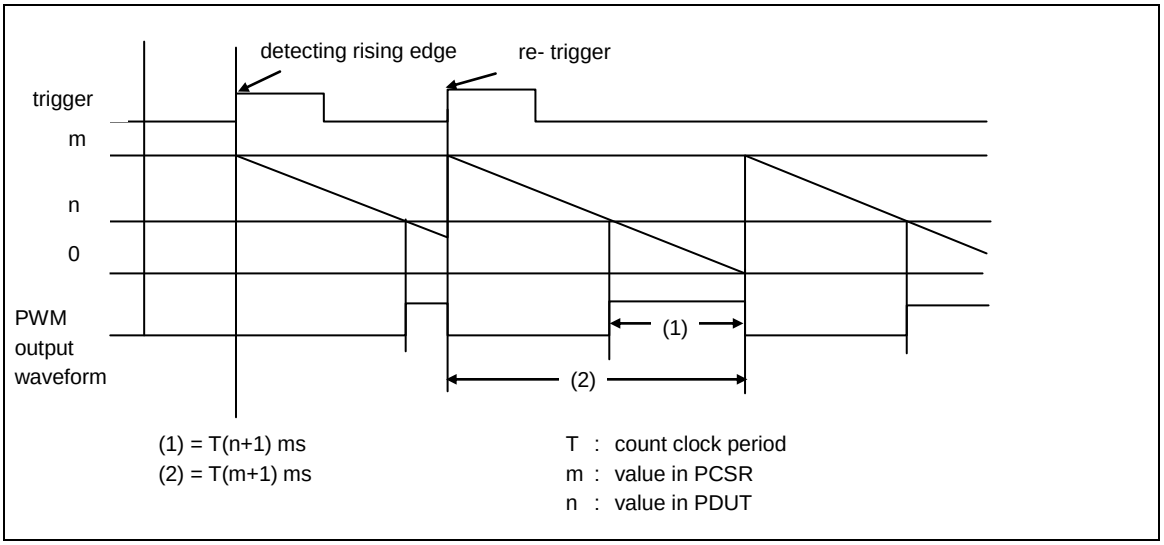
When Restart is Disabled (RTGEN="0")

Figure 10-1 PWM Operation Timing Chart (When Restart Is Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-2 PWM Operation Timing Chart (When Restart Is Enabled)



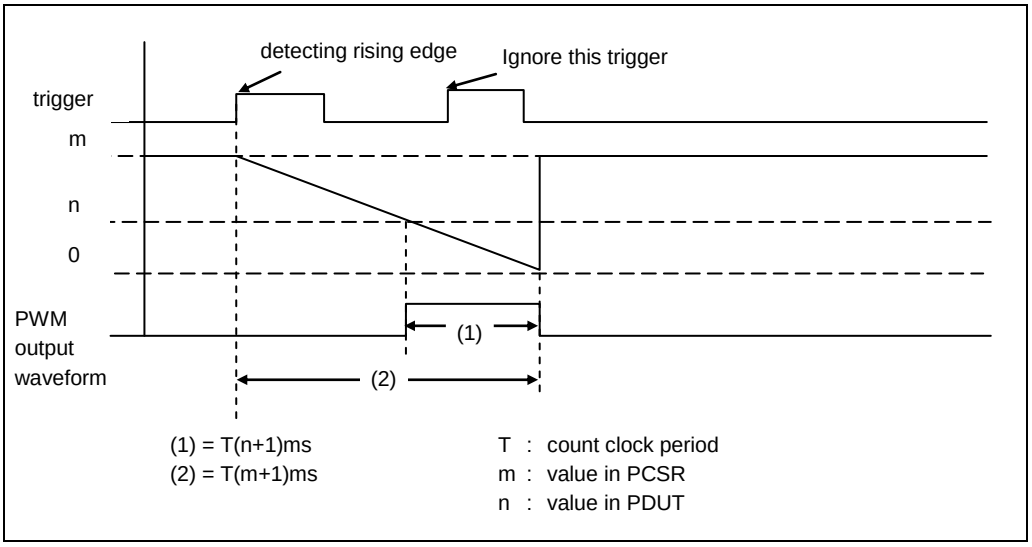
10.1.2. One-Shot Operation

In one-shot operation, a trigger can cause the output of a single pulse of any width. If restart is enabled, the counter is reloaded when an edge is detected during operation.

One-Shot Operation

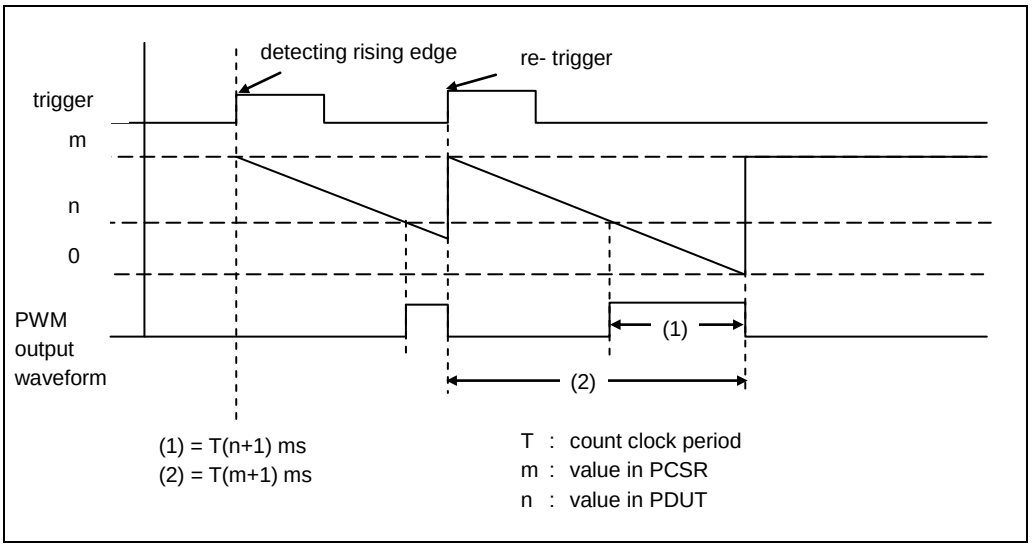
When Restart is Disabled (RTGEN="0")

Figure 10-3 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-4 One-Shot Operation Timing Chart (Trigger Restart Enabled)



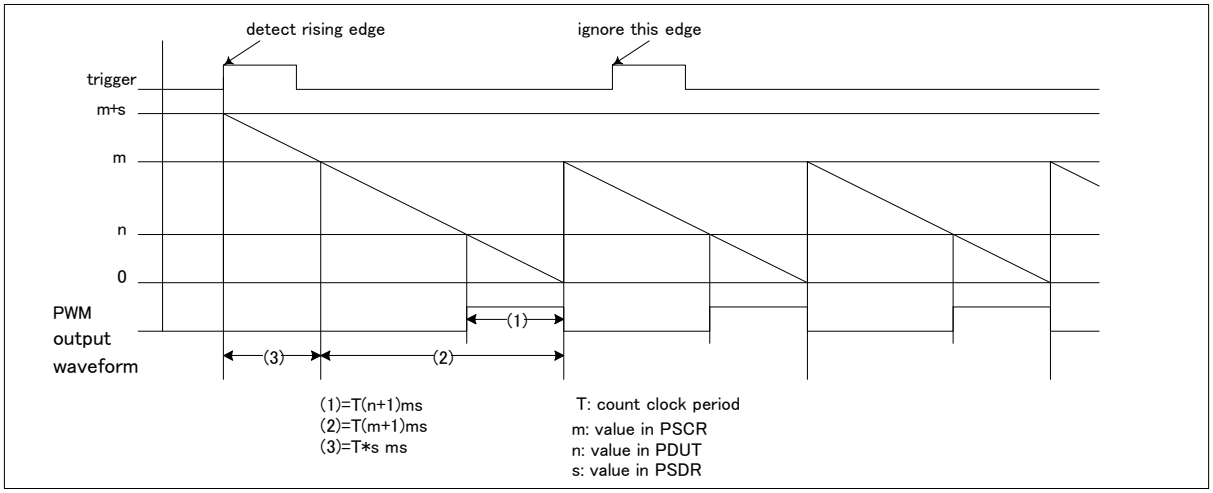
10.1.3. Start Delay Function

The start delay function can delay the time when PWM control starts after a trigger input.

Start Delay Operation in Continuous Operation

When Restart is Disabled (RTGEN="0")

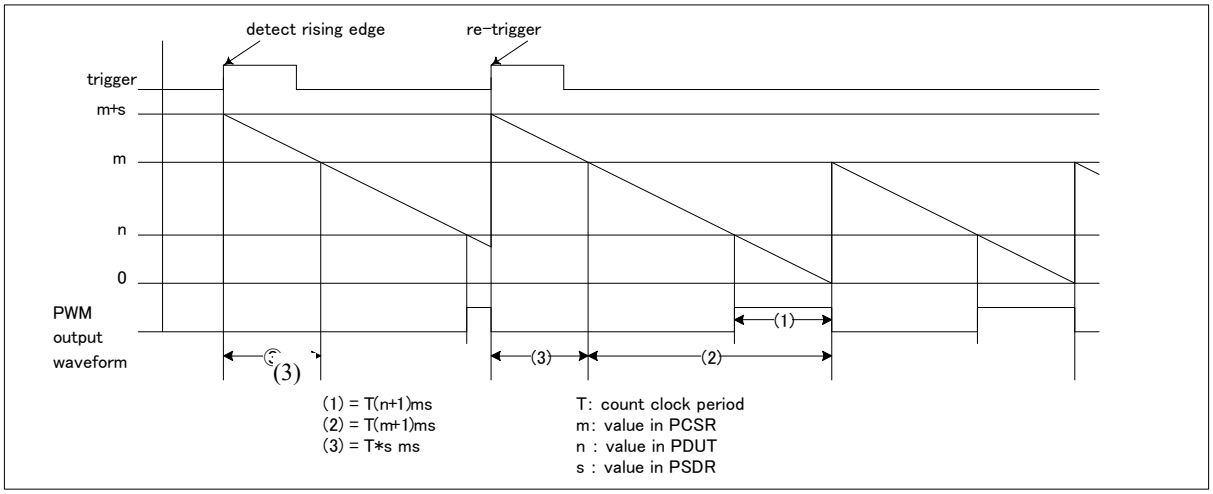
Figure 10-5 Start Delay Operation in Continuous Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-6 Start Delay Operation in Continuous Operation (Trigger Restart Enabled)

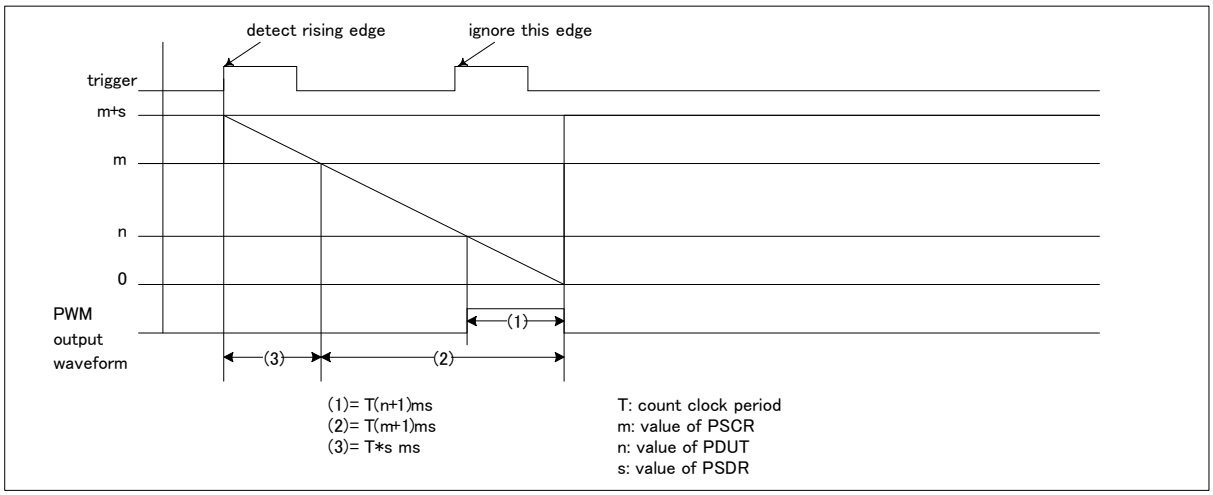


No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation in One-Shot Operation

When Restart is Disabled (RTGEN="0")

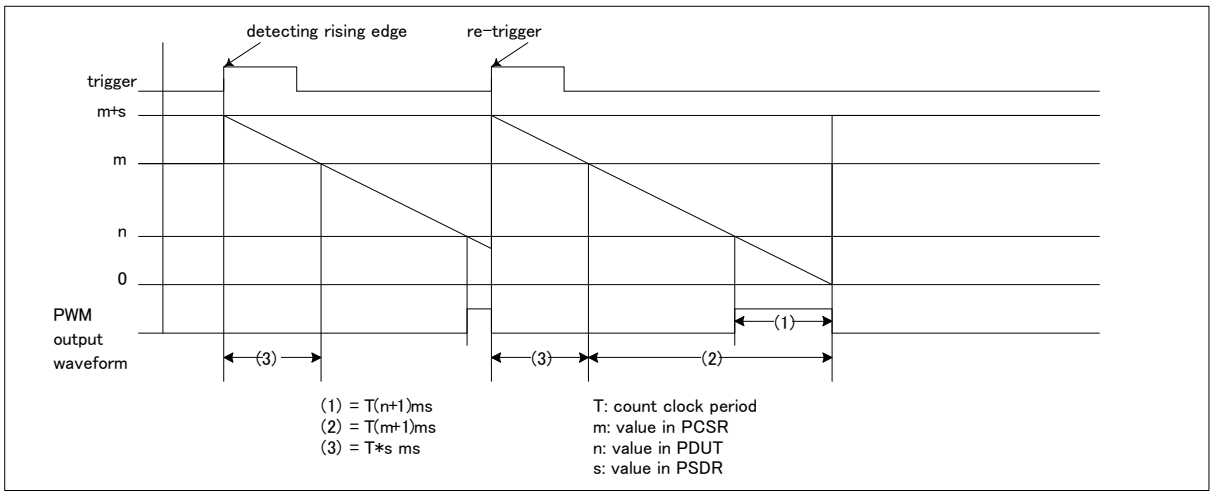
Figure 10-7 Start Delay Operation in One-Shot Operation (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-8 Start Delay Operation in One-Shot Operation (Trigger Restart Enabled)



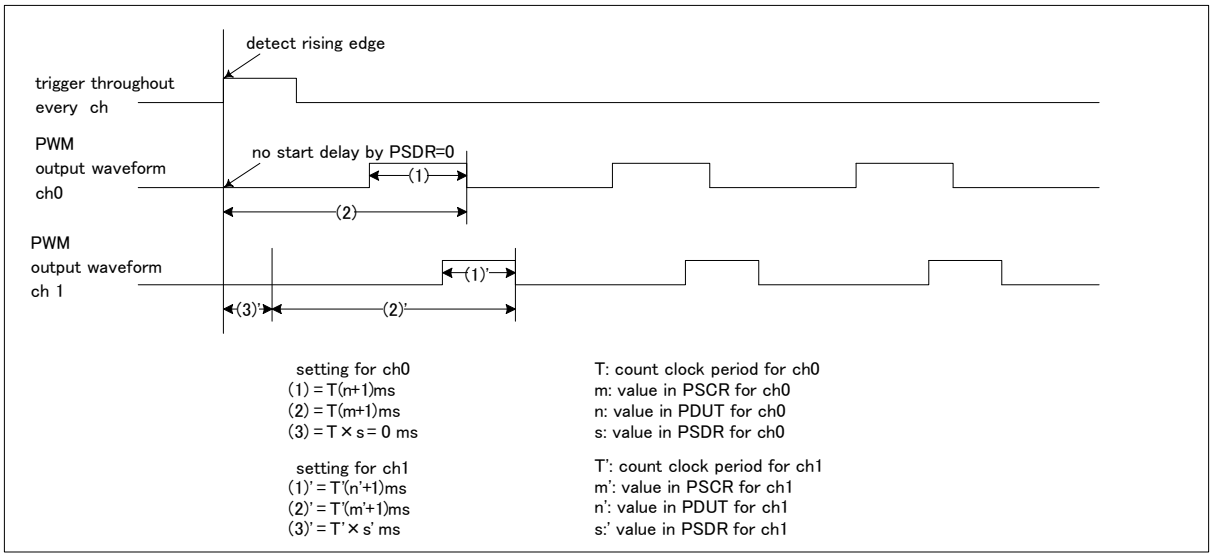
No PWM waveform is output during the start delay period ((3) above).

Start Delay Operation with Multiple Channels interlocked

The trigger can be controlled to interlock several channels in a specific phase relationship.

When Restart is Disabled (RTGEN="0")

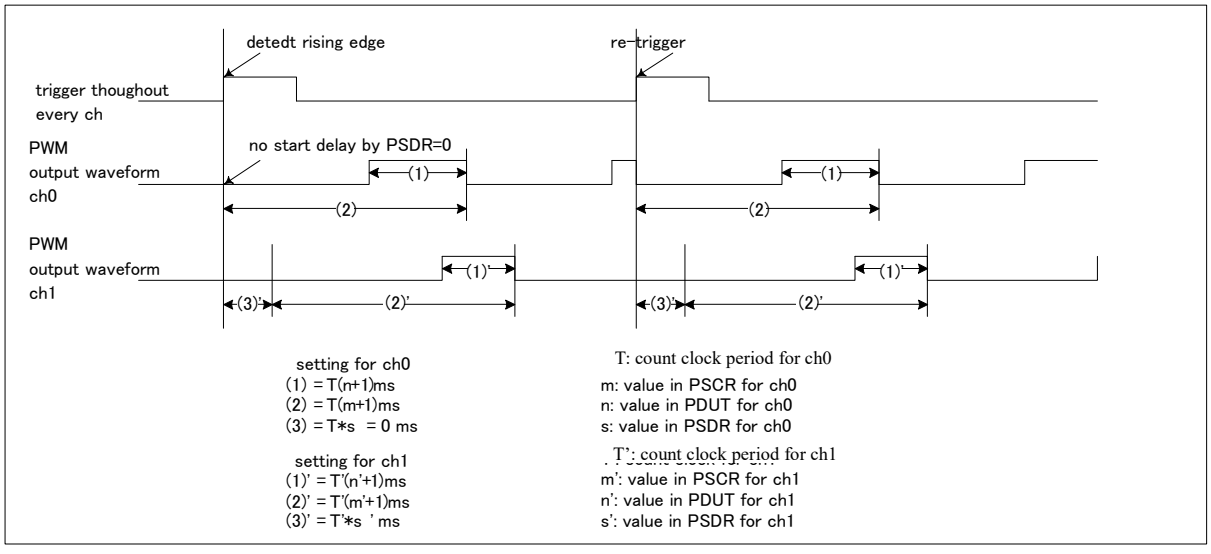
Figure 10-9 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Disabled)



No PWM waveform is output during the start delay period ((3) above).

When Restart is Enabled (RTGEN="1")

Figure 10-10 Start Delay Operation in Continuous Operation with Multiple Channels Interlocked (Trigger Restart Enabled)



No PWM waveform is output during the start delay period ((3) above).

10.1.4. ADC Trigger Output Timing

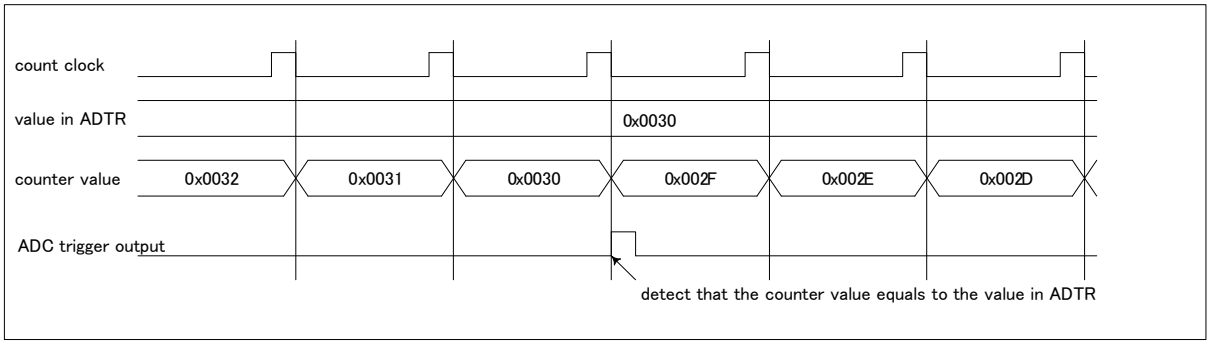
This section explains the ADC trigger output timing.

ADC Trigger Output Timing

ADC trigger is output when the 16-bit down counter is compared with the ADTR register and they match.

Figure 10-11 shows a timing chart of ADC trigger output when ADTR is 0x0030.

Figure 10-11 Timing Chart of ADC Trigger Output



10.1.5. External Timer Match Starting

If the external timer match starting function is used, the 16-/17-bit down counter can start in synchronization with an external timer.

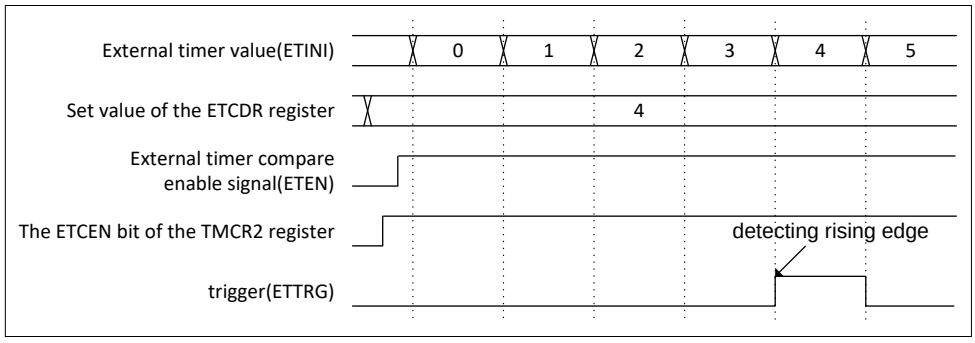
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1".
- The ETCEN bit of the TMCR2 register is "1".
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

Figure 10-12 Example of External Timer Match Starting Operation (ETCDR=4)



10.1.6. Interrupt Factors and Timing Chart

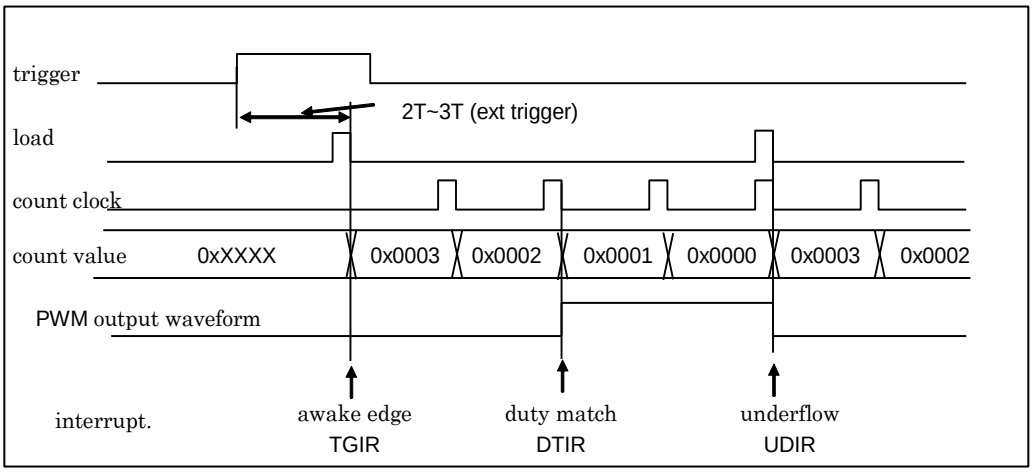
This section shows interrupt factors and a timing chart.

Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

The time required for a counter value to be loaded following the input of a trigger is, for software triggers T, and for external triggers 2T to 3T (T: internal clock cycle).

Figure 10-13 shows the interrupt factors and a timing chart where the cycle setting value is 3 and the duty value is 1.

Figure 10-13 PWM Timer Interrupt Factors and Timing Chart



10.1.7. Output Waveform

This section shows PWM output.

Output methods for All "L" or All "H" in PWM Output

Figure 10-14 shows an output method where the PWM output is all "L". Figure 10-15 shows an output method where it is all "H".

Figure 10-14 Example Where PWM Output Is All "L" Level (TMCR.OSEL="0")

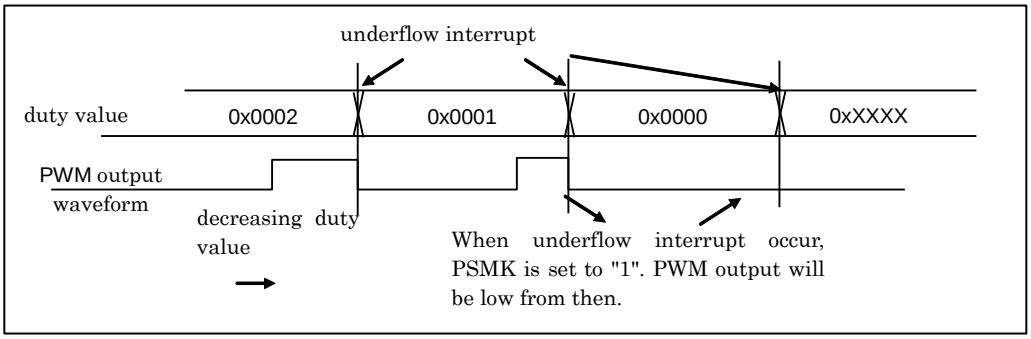
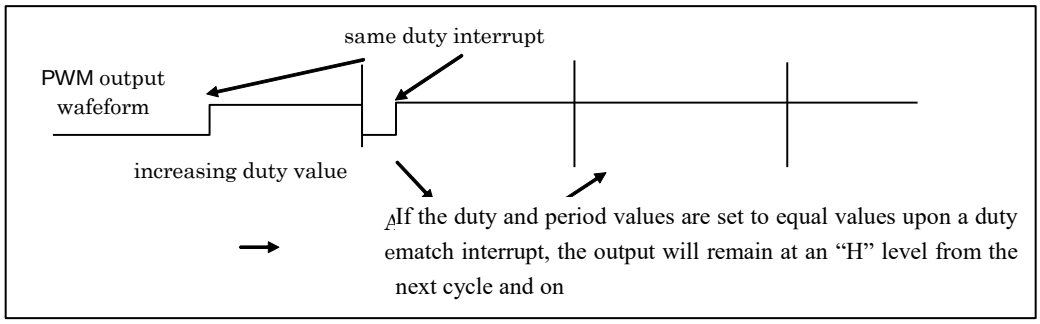


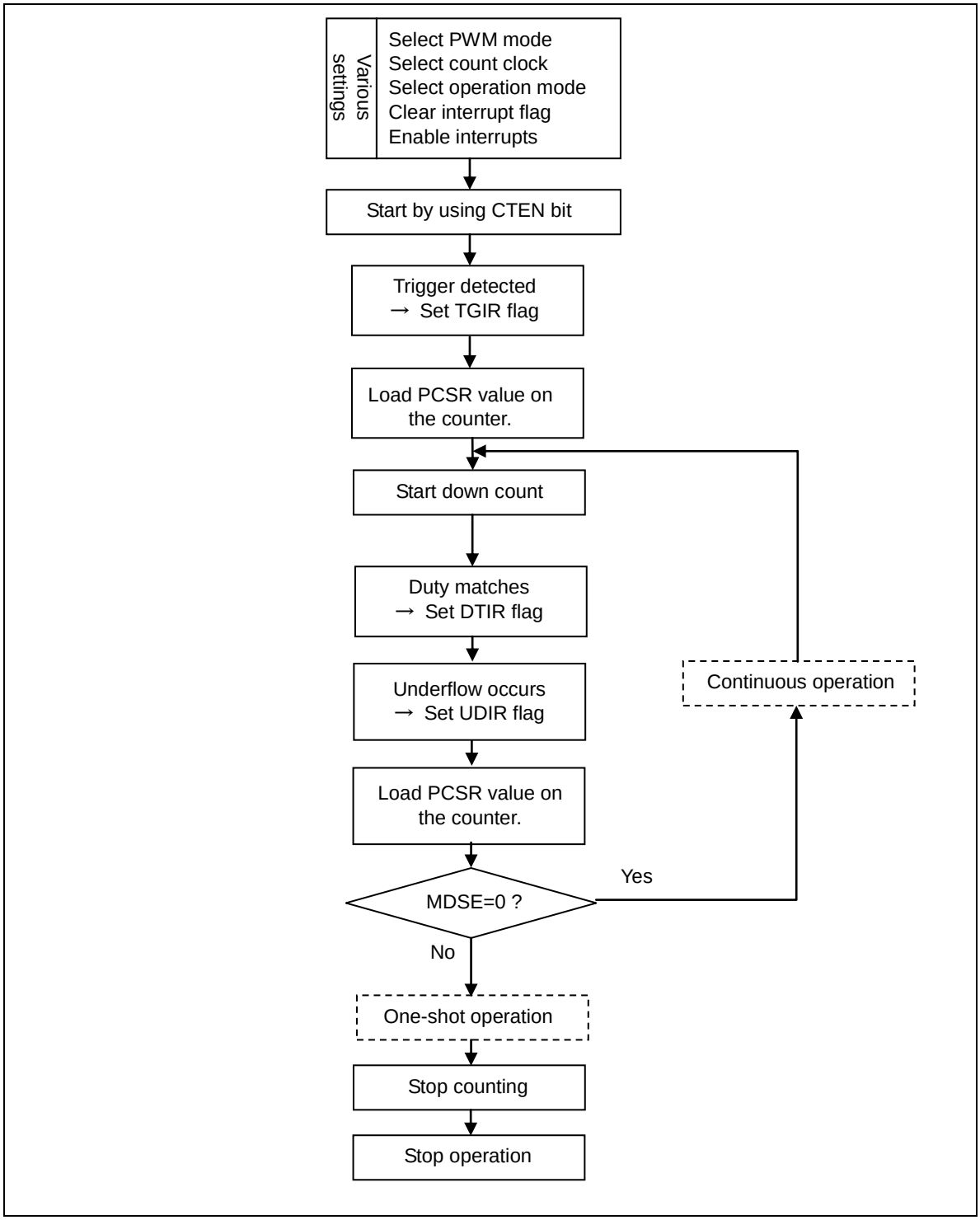
Figure 10-15 Example Where PWM Output Is All "H" Level (TMCR.OSEL="0")



10.1.8. PWM Timer Operation Flow

This section shows the PWM Timer operation flow.

PWM Timer Operation Flow



10.1.9. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

The Timer Control Register (TMCR) controls the PWM Timer. Note that there are bits that cannot be rewritten during PWM Timer operation.

For details on writing to the Status Control Register (STC), see "9.Notes on Using the Base Timer".

10.1.9.1 Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

- These bits select a count clock for the 17-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the count operation enable bit (CTEN).

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 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-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] RTGEN: Restart Enable Bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Disable restart.
1	Enable restart.

[bit10] PMSK: Pulse Output Mask Bit

- This bit controls the output waveform level of the PWM output waveform.
- When the bit is "0", the PWM waveform is output as is.
- When the bit is "1", PWM output is masked to "L" output regardless of the cycle or duty setting value.

Note:

- *If the output polarity specification bit (OSEL) in the Timer Control Register (lower byte of TMCR) is set for inverted output, setting the PMSK bit to "1" results in masking to "H" output.*

Bit	Description
0	Normal output
1	Fixed at "L" output

[bit9:8] EGS1 to EGS0: Trigger Input Edge Selection Bits

- These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
- For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

EGS1	EGS0	Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

10.1.9.2 Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	Reserved	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

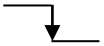
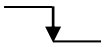
[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- These bits select the timer function.
- If "001" is set in the FMD2 to FMD0 bits, the PWM function is selected.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

FMD2	FMD1	FMD0	Description
0	0	0	Reset mode
0	0	1	16-bit PWM Timer
0	1	0	16-bit PPG Timer
0	1	1	16/32-bit Reload Timer
1	0	0	16/32-bit PWC Timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] OSEL: Output Polarity Specification Bit

- This bit sets the PWM output polarity.

Polarity	After Reset	Duty Match	Underflow
Normal	"L" output		
Inverse	"H" output		

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- This bit selects either operation for continuous pulse output or one-shot operation for single-pulse output.
- Modify the bit while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit1] CTEN: Count Operation Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description
0	Stop operation.
1	Enable operation.

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

10.1.9.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved						ETCEN	CKS3
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit7:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.1.9Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)".

10.1.9.4 Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	DTIE	UDIE	Reserved	TGIR	DTIR	UDIR
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.TGIEC bit clears this bit.
- Writing "1" to the STCS.TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] DTIE: Duty Match Interrupt Request Enable Bit

- This bit controls interrupt requests of the duty match interrupt request bit (bit1 DTIR).
- If the DTIE bit is enabled and the DTIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.DTIEC bit clears this bit.
- Writing "1" to the STCS.DTIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.UDIEC bit clears this bit.
- Writing "1" to the STCS.UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC.TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] DTIR: Duty Match Interrupt Request Bit

- The DTIR bit is set to "1" when the count value matches the duty setting value.
- Writing "1" to the STCC.DTIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit0] UDIR: Underflow Interrupt Request Bit

- The UDIR bit is set to "1" when the count value underflows from 0x0000 to 0xFFFF.
- Writing "1" to the STCC.UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

10.1.9.5 Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	DTIEC	UDIEC	Reserved	TGIRC	DTIRC	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] DTIEC: Duty Match Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.DTIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the DTIE bit.

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] DTIRC: Duty Match Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.DTIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the DTIR bit.

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

10.1.9.6 Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	DTIES	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] DTIES: Duty Match Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.DTIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the DTIE bit.

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3-0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

10.1.10. PWM Cycle Setting Register (PCSR)

The PWM Cycle Setting Register (PCSR) is a register with a buffer for setting a cycle. There is a transfer to the Timer Register (TMR) at the start time and at the underflow time.

bit	15	8
Field	PCSR[15:8]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXX XXXX	

bit	7	0
Field	PCSR[7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXX XXXX	

These bits compose the register with a buffer for setting a cycle. There is a transfer to the Timer Register (TMR) at the start time and at the underflow time.

When initializing and rewriting the Cycle Setting Register (PCSR), be sure to write to the duty setting register after writing to the Cycle Setting Register (PCSR).

- Use 16- or 32-bit data access for the PCSR register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PCSR register.

10.1.11. PWM Duty Setting Register (PDUT)

The PWM Duty Setting Register (PDUT) is a register with a buffer for setting a duty. An underflow causes a buffer transfer.

bit	15							8
Field	PDUT[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

bit	7							0
Field	PDUT[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXX XXXX							

These bits compose the register with a buffer for setting a duty. An underflow causes a transfer from the buffer.

If the set values of the Cycle Setting Register (PCSR) and duty setting register are the same value, the output for normal polarity is all "H", and the output for inverse polarity is all "L".

If PCSR < PDUT in the set values, the output for normal polarity is all "L", and the output for inverse polarity is all "H".

- Use 16- or 32-bit data access for the PDUT register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a duty in the PDUT register.

10.1.12. Timer Register (TMR)

The Timer Register (TMR) can read the value of the17-bit down counter.

Bit	31	24
Field	Reserved	
R/W Attribute	R0,W0	
Protection Attribute	-	
Initial Value	0000 0000	

bit	23	17	16
Field	Reserved		TMR[16]
R/W Attribute	R0,W0		R,WX
Protection Attribute	-		
Initial Value	0000000		0

Bit	15	8
Field	TMR[15:8]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0000 0000	

Bit	7	0
Field	TMR[7:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	0000 0000	

[bit31:17] Reserved: Reserved Bits

The read value is "0".

[bit16:0] TMR16 to TMR0: Timer Value Data Bits

The Timer Register (TMR) can read the value of the 17-bit down counter that takes into account the start delay period.

- During trigger input, a value of "start delay period (PSDR) + PWM cycle (PCSR)" is reloaded into the TMR register.
- A value of "PWM cycle (PCSR)" is reloaded into the TMR register when an underflow occurs.
- Use 32-bit data access for the TMR register.

10.1.13. Start Delay Value Setting Register (PSDR)

The Start Delay Value Setting Register (PSDR) sets a start delay value for the start delay function.

bit	15							8
Field	PSDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

bit	7							0
Field	PSDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

The PSDR[15:0] bits set a start delay value to delay the start of the PWM Timer after trigger input.

- To not use the start delay function, set the PSDR[15:0] bits to 0x0000.
- No PWM waveform is output during start delay control period after trigger input.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PSDR register.
- Use 16- or 32-bit data access for the PSDR register.
- Any modification to the PSDR[15:0] bits is reflected immediately after the setting is changed. Set the PSDR[15:0] bits while counting is stopped (TMCR:CTEN="0").

10.1.14. ADC Trigger Value Setting Register (ADTR)

The ADC Trigger Value Setting Register (ADTR) sets the time for ADC trigger output.

bit	15							8
Field	ADTR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	00000 0000							

bit	7							0
Field	ADTR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	0000 0000							

The ADC trigger is output when the 16-bit down counter is compared with the ADTR[15:0] bits and they match.

- For the ADTR register, set a value equal to or less than the PCSR[15:0] bits in the PCSR register.
- Use 16- or 32-bit data access for the ADTR register.
- After configuring the PWM function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the ADTR register.

10.1.15. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15	8
Field	Reserved	
R/W Attribute	R0,W0	
Protection Attribute	-	
Initial Value	0000 0000	

bit	7	1	0
Field	Reserved		DBGEN
R/W Attribute	R0,W0		R/W
Protection Attribute	-		
Initial Value	0000000		0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.1.16. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15							8
Field	ETCDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	01111111							

bit	7							0
Field	ETCDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	11111111							

It is a register compared with an external timer signal (ETINI). Refer to "10.1.5 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.2. PPG Timer Function

Only one of the following timer functions can be selected for the Base Timer in the FMD2 to FMD0 bit settings in the Timer Control Register (TMCR): 16-bit PWM Timer, 16-bit PPG Timer, 16-/32-bit Reload Timer, and 16-/32-bit PWC Timer. This section explains the timer function with the PPG setting.

10.2.1 16-Bit PPG Timer Operation
10.2.2 Continuous Operation
10.2.3 One-Shot Operation
10.2.4 External Timer Match Starting
10.2.5 Interrupt Factors and Timing Chart
10.2.6 PPG Timer Operation Flow
10.2.7 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.2.8 Low Width Setting Reload Register (PRLL)
10.2.9 High Width Setting Reload Register (PRLH)
10.2.10 Timer Register (TMR)
10.2.11 Base Timer Debug Register (BT_DEBUG)
10.2.12 External Timer Compare Data Register (ETCDR)

10.2.1. 16-Bit PPG Timer Operation

PPG Timer operation can control output pulses by using the "L" width and "H" width settings for the output pulses in the respective reload registers.

Overview of Operation

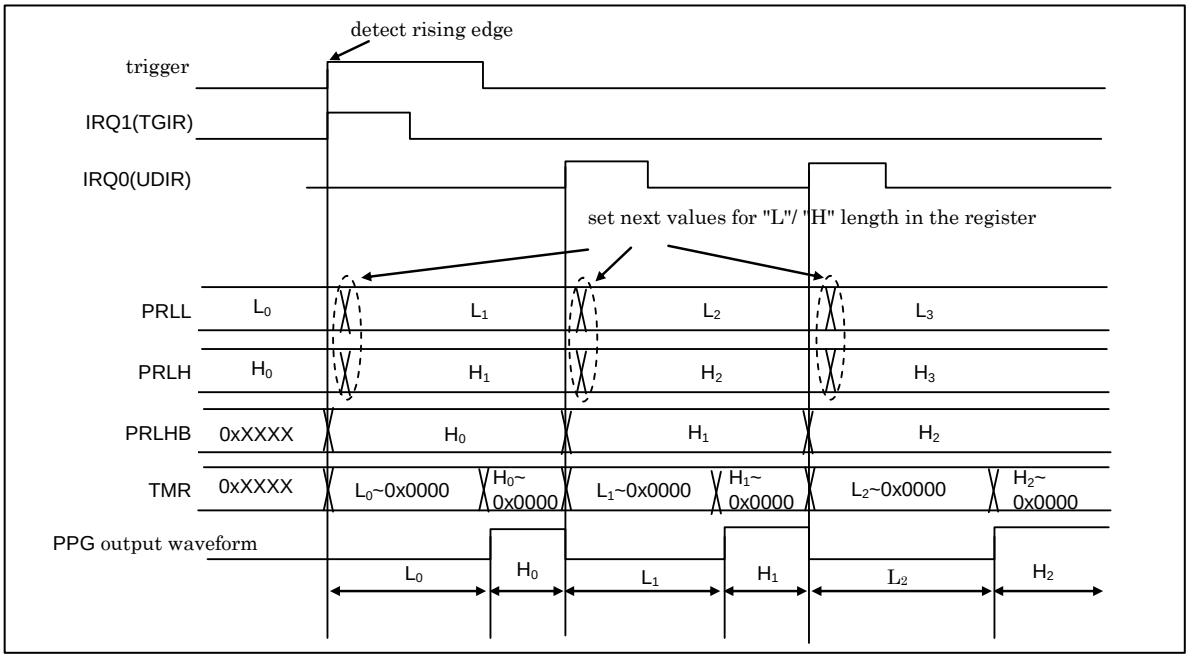
There are two 16-bit reload registers for the "L" width and "H" width settings and one buffer for the "H" width setting (PRLL, PRLH, and PRLHB).

An activation trigger first causes loading of the set value of PRLL into the 16-bit down counter. At the same time, the set value of PRLH is transferred to PRLHB. The PPG output level changes to "L", and the counter counts down each count clock. The detection of an underflow causes reloading of the PRLHB value into the counter, inverts the PPG output waveform and continues counting down each count clock. The detection of another underflow causes inversion of the PPG output waveform, reloading of the set value of PRLL into the counter, and transfer of the set value of PRLH to PRLHB.

The pulse output by the output waveform from this operation has an "L" width and "H" width corresponding to each reload register value.

Timing of Writing to Reload Registers

Data is written to the reload registers PRLL and PRLH when an activation trigger is detected, and also before the next cycle change occurs following an underflow interrupt factor (UDIR) being asserted to "1". The data that is set at this time is the settings for the next cycle. The set data in PRLL and PRLH is automatically transferred to TMR and PRLHB, respectively, when an activation trigger is detected or when an underflow occurs at the "H" width count end time. The data transferred to PRLHB is automatically reloaded into TMR when an underflow occurs at the "L" width count end time.



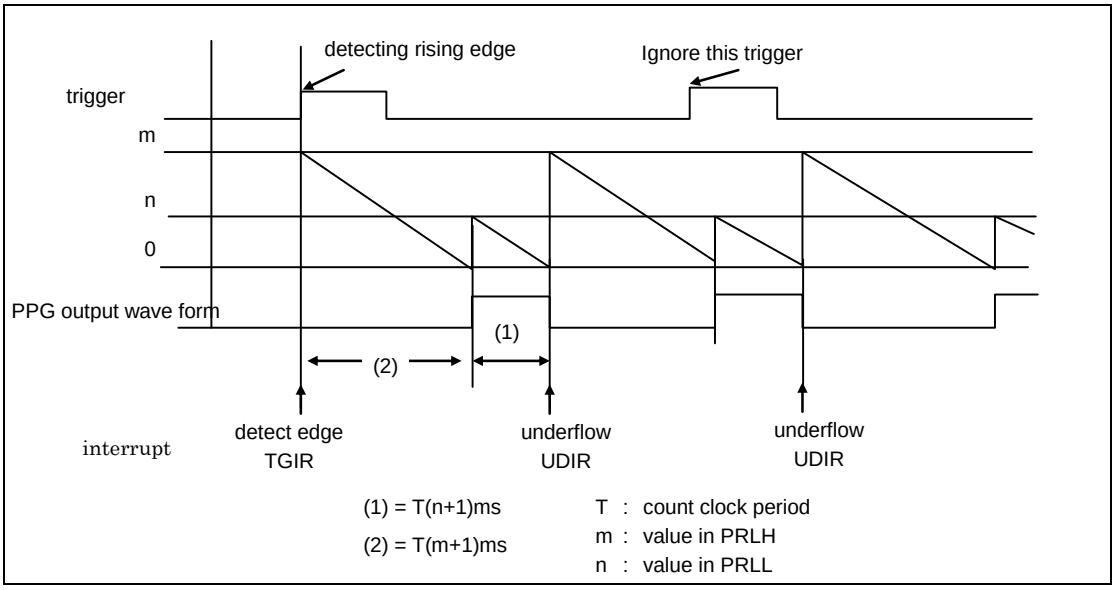
10.2.2. Continuous Operation

In continuous operation, pulses can be output continuously through updating of the "L" width and "H" width at the set timing for each interrupt factor. If restart is enabled, the counter is reloaded when an edge is detected during operation.

Continuous Operation

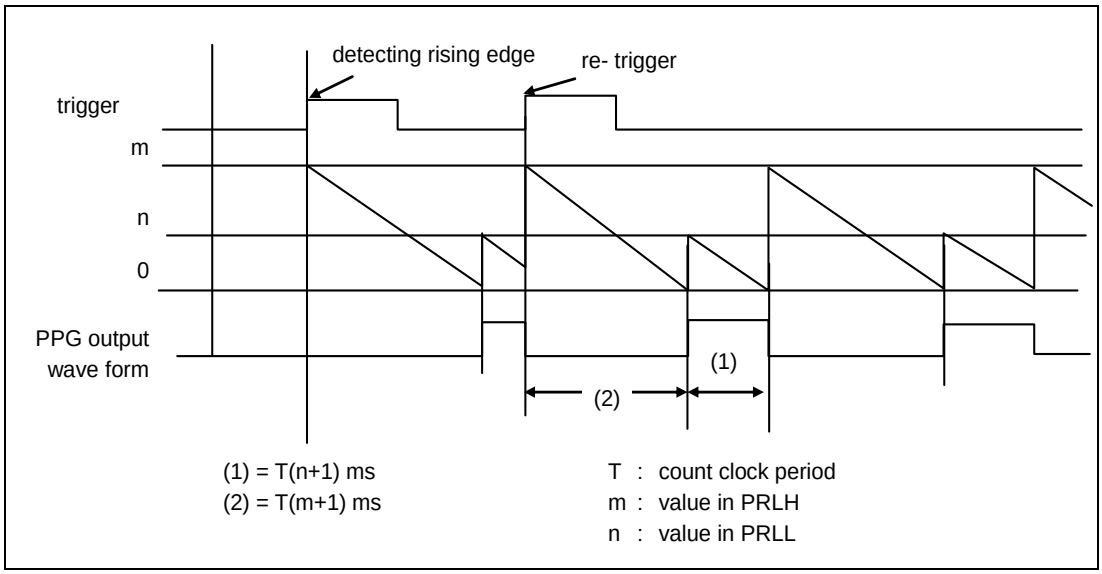
When Restart is Disabled (RTGEN="0")

Figure 10-16 PPG Operation Timing Chart (When Restart Is Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-17 PPG Operation Timing Chart (When Restart Is Enabled)



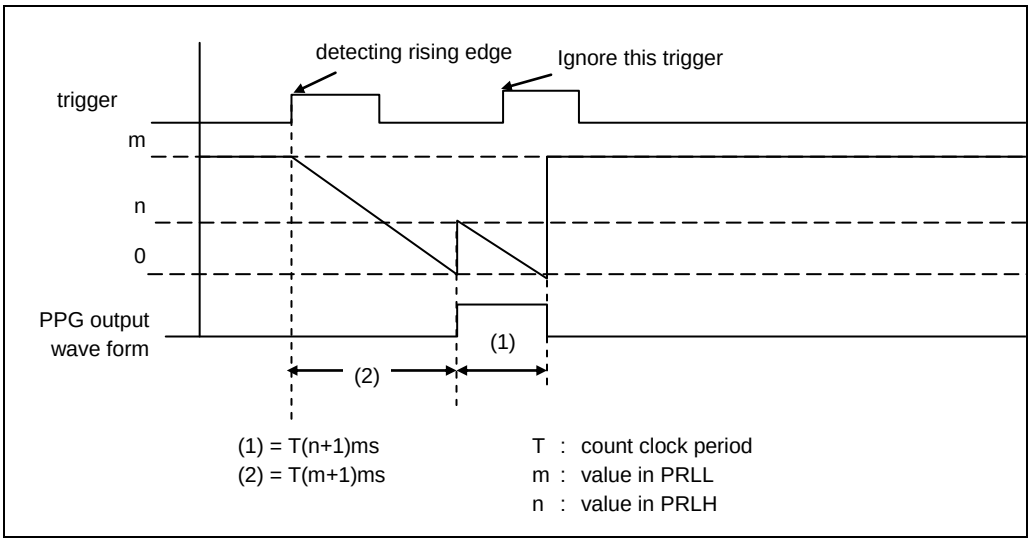
10.2.3. One-Shot Operation

In one-shot operation, a trigger can cause the output of a single pulse of any width. If restart is enabled, the counter is reloaded when an edge is detected during operation.

One-Shot Operation

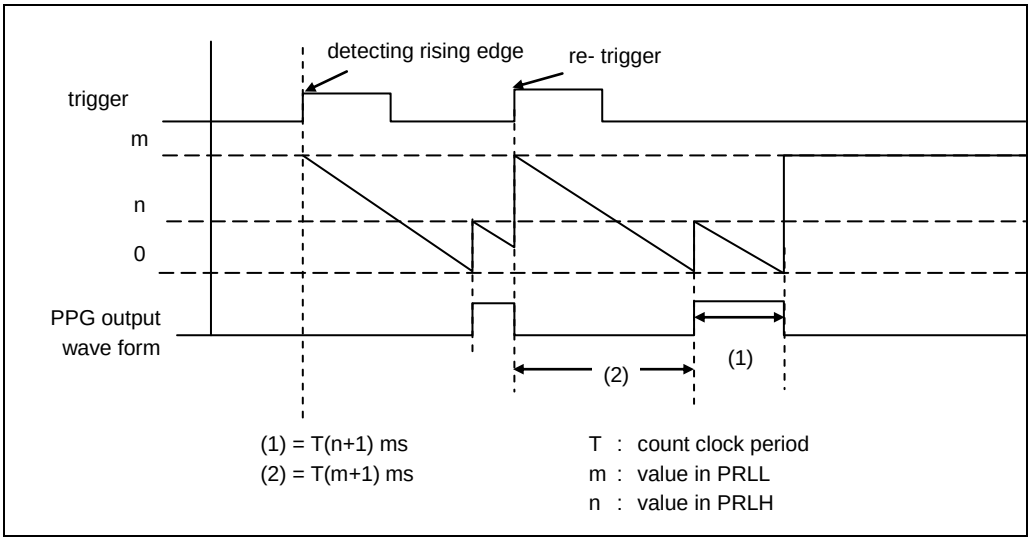
When Restart is Disabled (RTGEN="0")

Figure 10-18 One-Shot Operation Timing Chart (Trigger Restart Disabled)



When Restart is Enabled (RTGEN="1")

Figure 10-19 One-Shot Operation Timing Chart (Trigger Restart Enabled)



Relationship Between Reload Value and Pulse Width

The output pulse width is the following value: 1 is added to the value written to the 16-bit reload register, and the sum is multiplied by the cycles of the count clock. Therefore, when the value of the reload register is 0x0000, the pulse width is equal to 1 count clock cycle. Similarly, when the value of the reload register is 0xFFFF, the pulse width is equal to 65536 count clock cycles. The pulse width calculation formula is as follows.

PL = T x (L + 1)	PL: "L" pulse width
PH = T x (H + 1)	PH: "H" pulse width
	T: Count clock cycle
	L: PRLL value
	H: PRLH value

10.2.4. External Timer Match Starting

If the external timer match starting function is used, the 16-/17-bit down counter can start in synchronization with an external timer.

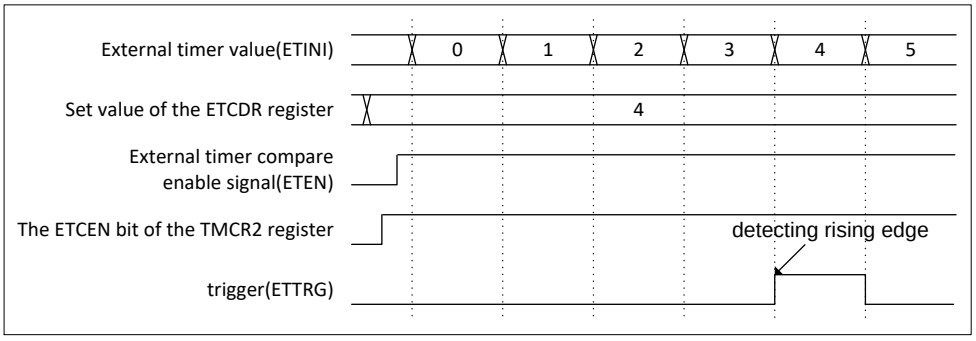
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1".
- The ETCEN bit of the TMCR2 register is "1".
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

Figure 10-20 The Example of External Timer Match Starting Operation (ETCDR=4)



10.2.5. Interrupt Factors and Timing Chart

This section shows interrupt factors and a timing chart.

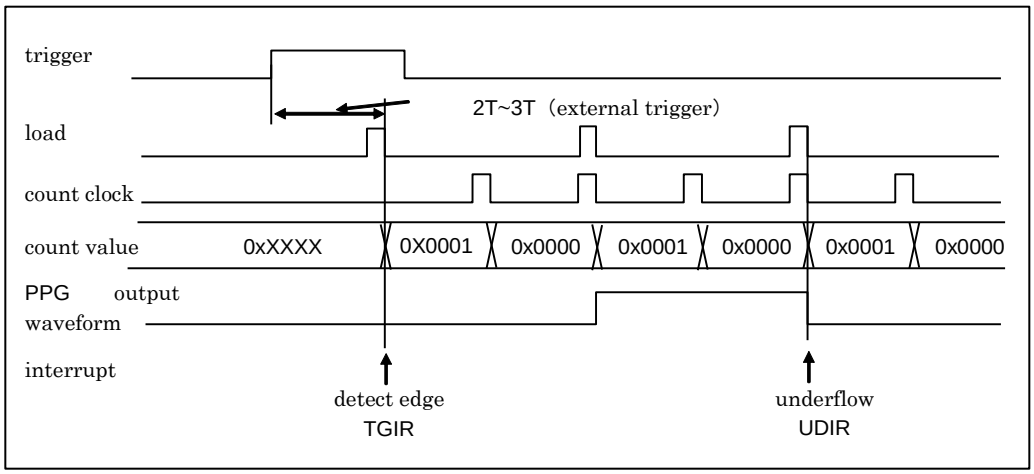
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

For the period from trigger application until the loading of a counter value, the required software trigger time is T, and the required external trigger time is 2T to 3T (T: Internal clock cycle).

An interrupt factor is generated when a PPG activation trigger is detected or when an underflow is detected at the "H" level output time.

Figure 10-21 shows the interrupt factors and a timing chart where the "L" width setting value is 1 and the "H" width setting value is 1.

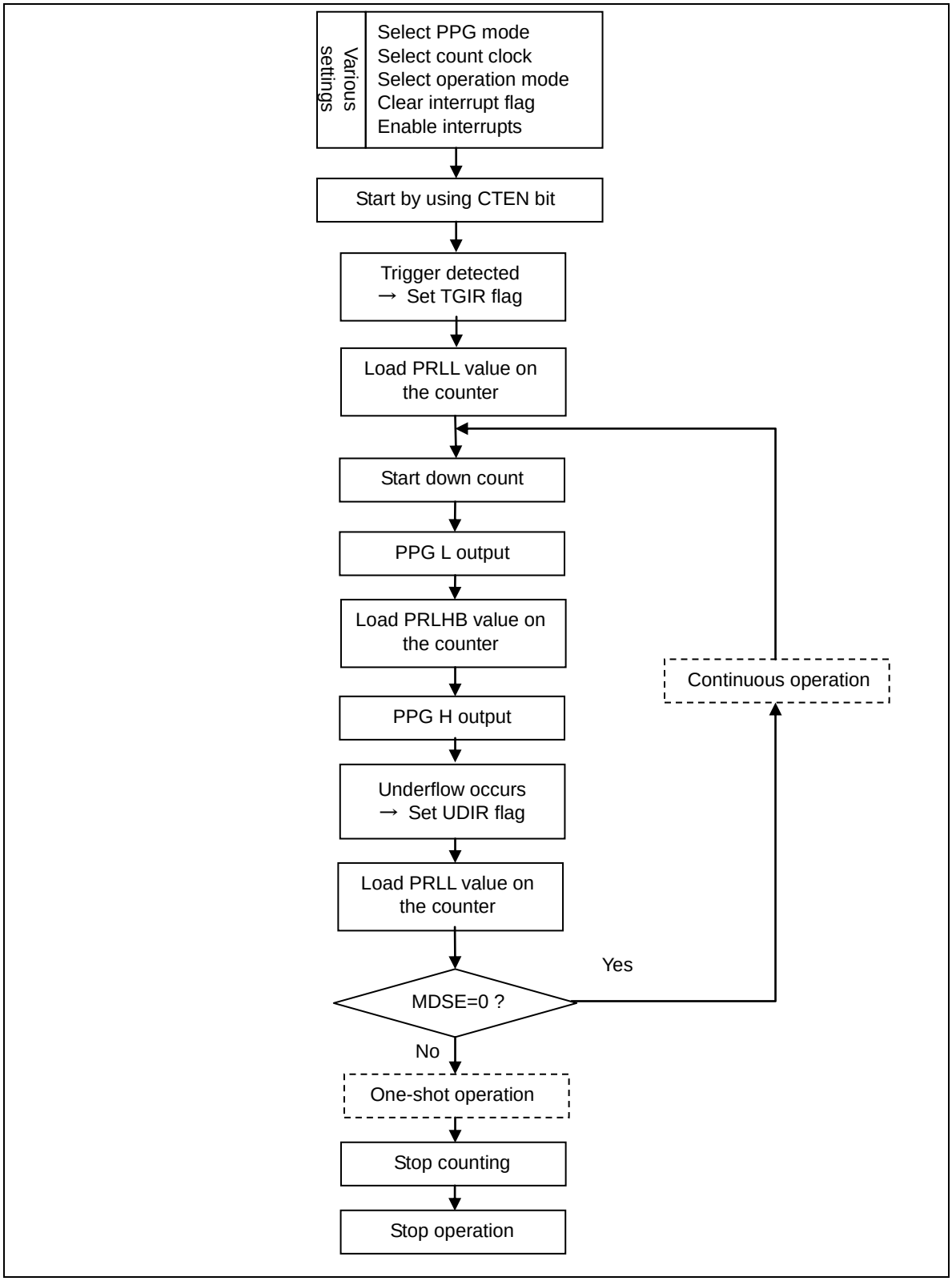
Figure 10-21 PPG Timer Interrupt Factors and Timing Chart



10.2.6. PPG Timer Operation Flow

This section shows the PPG Timer operation flow.

PPG Timer Operation Flow


Various settings
Select PPG mode
Select count clock
Select operation mode
Clear interrupt flag
Enable interrupts
Start by using CTEN bit
Trigger detected
→ Set TGIR flag
Load PRLL value on the counter
Start down count
PPG L output
Load PRLHB value on the counter
PPG H output
Underflow occurs
→ Set UDIR flag
Load PRLL value on the counter
MDSE=0 ?
Yes
No
Continuous operation
One-shot operation
Stop counting
Stop operation

10.2.7. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.2.7.1 Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 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-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] RTGEN: Restart Enable Bit

This bit enables restart by a software trigger or trigger input.

Bit	Description
0	Disable restart.
1	Enable restart.

[bit10] PMSK: Pulse Output Mask Bit

- This bit controls the output waveform level of the PPG output waveform.
- When the bit is "0", the PPG waveform is output as is.
- When the bit is "1", PPG output is masked to "L" output regardless of the cycle or duty setting value.

Note:

- *If the output polarity specification bit (OSEL) in the Timer Control Register (lower byte of TMCR) is set for inverted output, setting the PMSK bit to "1" results in masking to "H" output.*

Bit	Description
0	Normal output
1	Fixed at "L" output

[bit9:8] EGS1 to EGS0: Trigger Input Edge Selection Bits

- These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
- For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

EGS1	EGS0	Description
0	0	Trigger input invalid
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

10.2.7.2 Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	Reserved	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

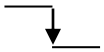
[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- These bits select the timer function.
- If "010" is set in the FMD2 to FMD0 bits, the PPG function is selected.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

FMD2	FMD1	FMD0	Description
0	0	0	Reset mode
0	0	1	16-bit PWM Timer
0	1	0	16-bit PPG Timer
0	1	1	16/32-bit Reload Timer
1	0	0	16/32-bit PWC Timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] OSEL: Output Polarity Specification Bit

- This bit sets the PPG output polarity.

Polarity	After Reset	"L" Width Count End	"H" Width Count End
Normal	"L" output		
Inverse	"H" output		

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- This bit selects either operation for continuous pulse output or one-shot operation for single-pulse output.
- Modify the bit while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous operation
1	One-shot operation

[bit1] CTEN: Count Operation Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description
0	Stop operation.
1	Enable operation.

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

10.2.7.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved						ETCEN	CKS3
R/W Attribute	R0,W0						R/W	R/W
Protection Attribute	-							
Initial Value	000000						0	0

[bit7:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.2.7 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)".

10.2.7.4 Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
R/W Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.TGIEC bit clears this bit.
- Writing "1" to the STCS.TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.UDIEC bit clears this bit.
- Writing "1" to the STCS.UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC.TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] UDIR: Underflow Interrupt Request Bit

- During counting from the set "H" width value, the UDIR bit is set to "1" when the count value underflows and changes from 0x0000 to 0xFFFF.
- Writing "1" to the STCC.UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

10.2.7.5 Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	Reserved	UDIEC	Reserved	TGIRC	Reserved	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

10.2.7.6 Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	Reserved	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

10.2.8. Low Width Setting Reload Register (PRLL)

The Low Width Setting Reload Register (PRLL) is the register used to set the "L" width of the PPG output waveform. An underflow at the activation trigger detection time or after the end of "H" width counting causes a transfer to the Timer Register (TMR).

bit	15						8
Field	PRLL[15:8]						
R/W Attribute	R/W						
Protection Attribute	-						
Initial Value	XXXXXXXX						

bit	7						0
Field	PRLL[7:0]						
R/W Attribute	R/W						
Protection Attribute	-						
Initial Value	XXXXXXXX						

These bits compose the register used to set the "L" width of the PPG output waveform. An underflow at the activation trigger detection time or "H" width count end time causes a transfer to the Timer Register (TMR).

- Use 16- or 32-bit data access for the PRLL register.
- After configuring the PPG function with the FMD2 to FMD0 bits in the TMCR register, set the "L" width in the PRLL register.

10.2.9. High Width Setting Reload Register (PRLH)

The High Width Setting Reload Register (PRLH) is the register with a buffer used to set the "H" width of the PPG output waveform. An underflow when an activation trigger is detected or after "H" width counting ends causes a transfer from PRLH to the buffer register. An underflow at the "L" width count end time causes a transfer from the buffer register to the Timer Register (TMR).

bit	15	8
Field	PRLH[15:8]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXX	

bit	7	0
Field	PRLH[7:0]	
R/W Attribute	R/W	
Protection Attribute	-	
Initial Value	XXXXXXXX	

These bits compose the register used to set the "H" width of the PPG output waveform. An underflow at the activation trigger detection time or "H" width count end time causes a transfer from PRLH to the buffer register. An underflow at the "L" width count end time causes a transfer from the buffer register to the Timer Register (TMR).

- Use 16- or 32-bit data access for the PRLH register.
- After configuring the PPG function with the FMD2 to FMD0 bits in the TMCR register, set the "H" width in the PRLH register.

10.2.10. Timer Register (TMR)

The Timer Register (TMR) can read the value of the 16-bit down counter.

bit	15	14	13	12	11	10	9	8
Field	TMR[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	TMR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	00000000							

The Timer Register (TMR) can read the value of the 16-bit down counter.

- Use 16- or 32-bit data access for the TMR register.

10.2.11. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15							8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7							0
Field	Reserved							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.2.12. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15							8
Field	ETCDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	01111111							

bit	7							0
Field	ETCDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	11111111							

It is a register compared with an external timer signal (ETINI). Refer to "10.2.4 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.3. Reload Timer Function

Only one of the following timer functions can be selected for the Base Timer in the FMD2 to FMD0 bit settings in the Timer Control Register (TMCR): 16-bit PWM Timer, 16-bit PPG Timer, 16-/32-bit Reload Timer, and 16-/32-bit PWC Timer. This section explains the timer function with the Reload Timer setting.

10.3.1 Operations of 16-Bit Reload Timer
10.3.2 External Timer Match Starting
10.3.3 Reload Timer Operation Flow
10.3.4 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.3.5 Cycle Setting Register (PCSR)
10.3.6 Timer Register (TMR)
10.3.7 Base Timer Debug Register (BT_DEBUG)
10.3.8 External Timer Compare Data Register (ETCDR)

10.3.1. Operations of 16-Bit Reload Timer

The Reload Timer operates by performing a countdown from the set value in the Cycle Setting Register (PCSR), in synchronization with the count clock. When the count value becomes "0", the counting ends, or the timer automatically loads the cycle setting to continue operating until the countdown is stopped.

Count Operation when Internal Clock is Selected

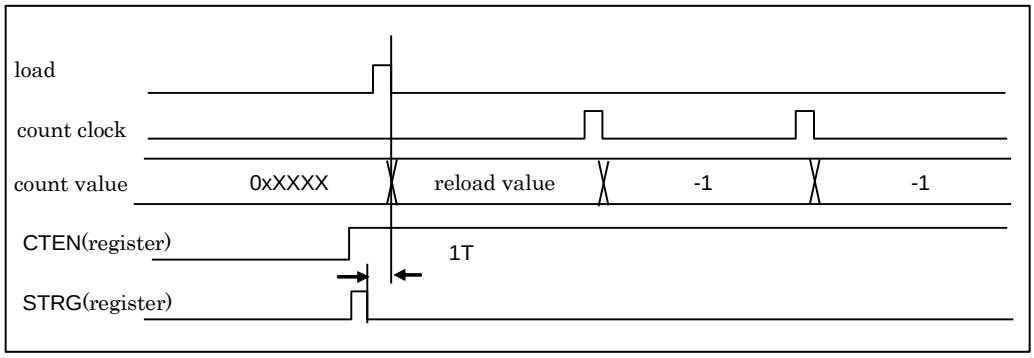
In order to start a count operation and enable counting at the same time, write "1" to both the CTEN bit and STRG bit in the Timer Control Register (TMCR). In a state where the timer has started (CTEN="1"), trigger input with the STRG bit is always enabled regardless of the operation mode.

With count operation enabled, the start of the timer by a software trigger or external trigger results in the Cycle Setting Register (PCSR) value being loaded into the counter and a countdown being started.

The time required from the generation of a counter start trigger until the loading of Cycle Setting Register (PCSR) data into the counter is 1T (T: Internal clock cycle).

Figure 10-22 shows the start and operation of the counter using a software trigger.

Figure 10-22 Count Operation When the Internal Clock Is Selected



Underflow Operation

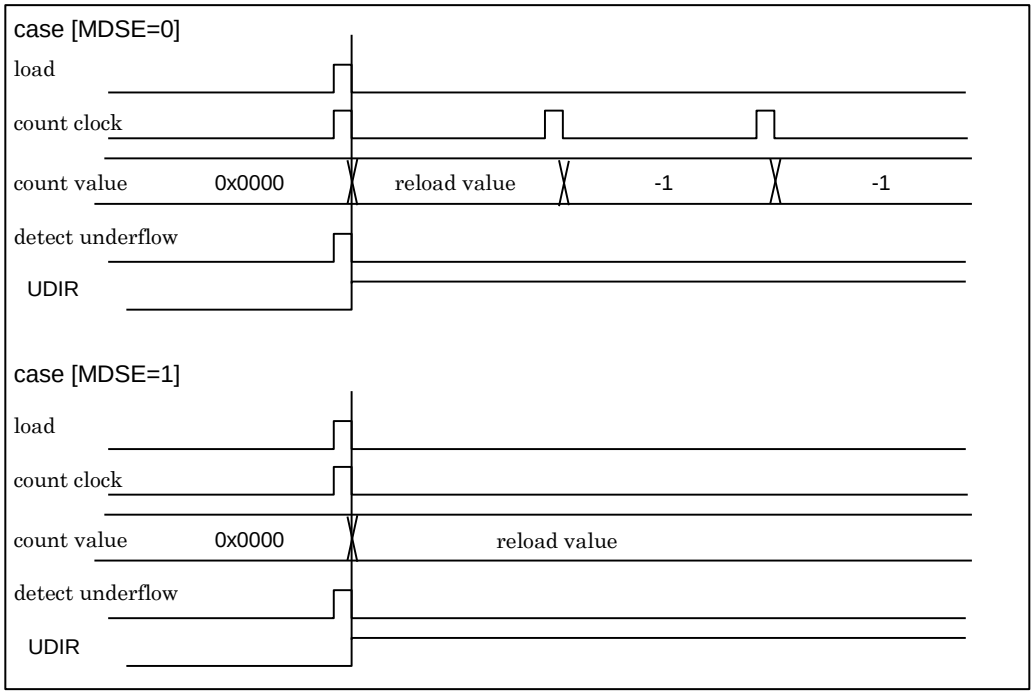
If the counter value changes from 0x0000 to 0xFFFF, an underflow has occurred. Therefore, an underflow occurs at the count of [set value of Cycle Setting Register (PCSR) + 1].

The Cycle Setting Register (PCSR) contents are loaded into the counter when an underflow occurs. If the MDSE bit in the Timer Control Register (TMCR) is "0", the count operation continues. If the MDSE bit is "1", the count operation stops with the loaded counter value unchanged.

The UDIR bit in the Status Control Register (STC) is set to "1" by an underflow. If the UDIE bit is "1" at this time, an interrupt request is generated.

Figure 10-23 shows an underflow operation timing chart.

Figure 10-23 Underflow Operation Timing Chart

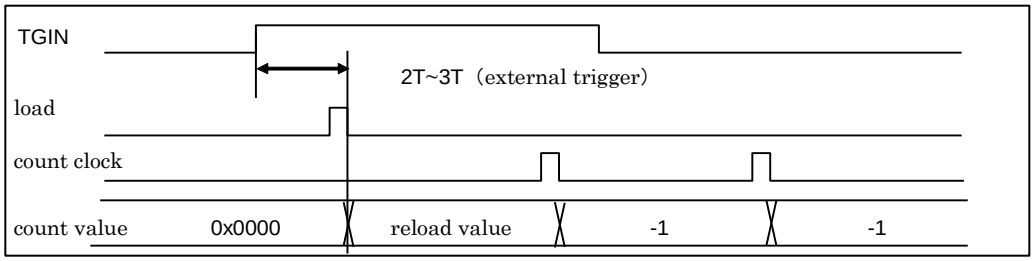


Operation of Input Pin Function

If the trigger input function (TMCR2.GATE="0") is selected, the TGIN pin can be used as trigger input. When a valid edge is input to the TGIN pin, the Cycle Setting Register (PCSR) contents are loaded into the counter, and a count operation begins. For the period from trigger application until the loading of a counter value, the required time is 2T to 3T (T: internal clock cycle).

Figure 10-24 shows trigger input operation where the valid edge specification is a rising edge.

Figure 10-24 Trigger Input Operation (TMCR2.GATE="0")



If the gate function (TMCR2.GATE="1") is selected, the TGIN pin can be used as the gate function.

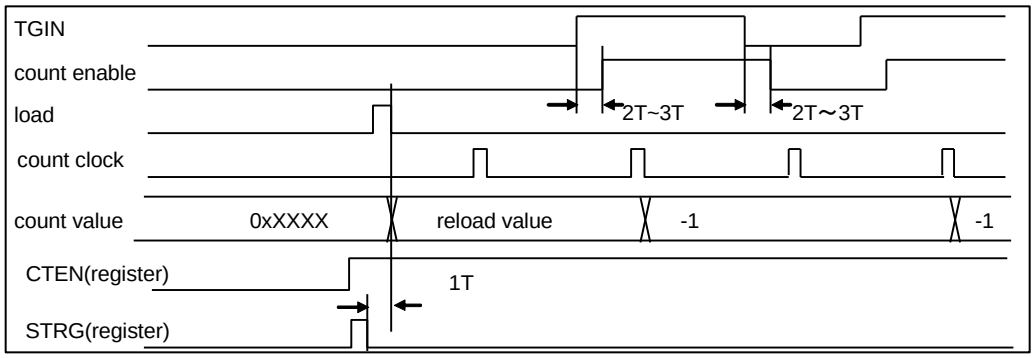
With count operations enabled (TMCR.CTEN="1"), the start of the timer by a software trigger (TMCR.STRG="1") results in the Cycle Setting Register (PCSR) value being loaded into the counter. The count clock counts down only while a valid level is being input to the TGIN pin.

In order to synchronize the signal that is input from the TGIN pin, the time required from the input of a valid level until count enable becomes effective is 2T to 3T at the start and end of the count operation(T: internal clock cycle).

The gate function and trigger input function are controlled exclusively. Therefore, activation by an external event cannot be used when the gate function is in use.

Figure 10-25 shows gate function operation where the valid level is "H".

Figure 10-25 Gate Function Operation (TMCR2.GATE="1")

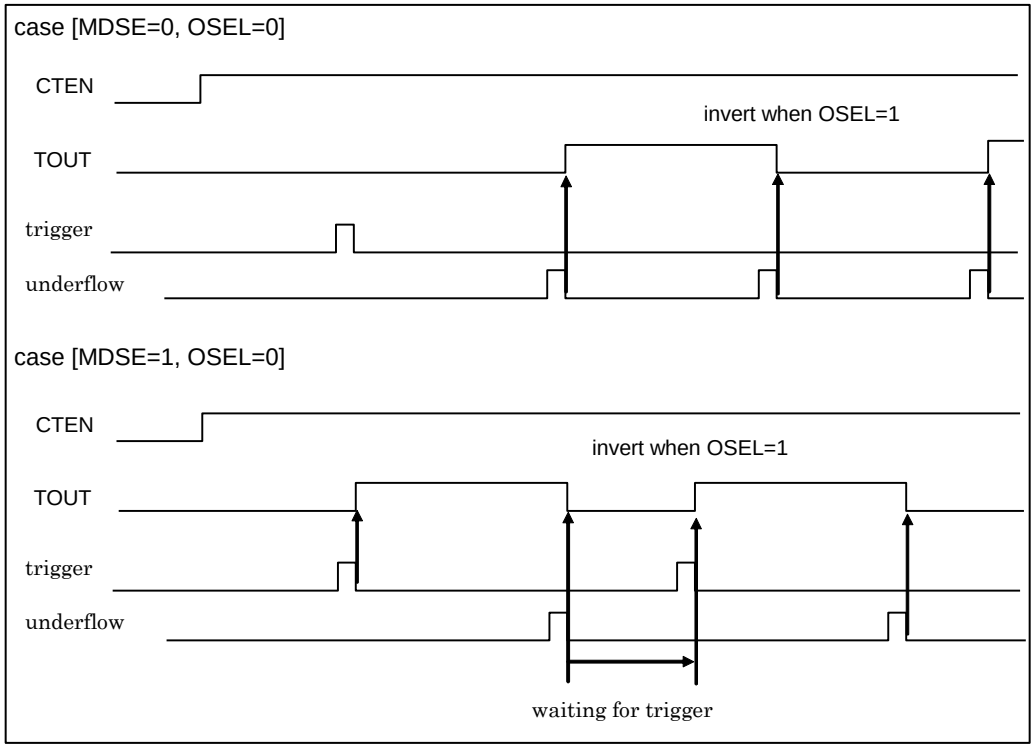


Operation of Output Pin Function

The TOUT output pin functions as toggle output in reload mode and as pulse output in one-shot mode. The toggle output is inverted by an underflow. The pulse output indicates that counting is in progress. The OSEL bit in the Timer Control Register (TMCR) can set the output polarity. If OSEL is "0", the initial value is "0" for toggle output, and "1" is output during counting for one-shot pulse output. If OSEL is "1", the output waveform is inverted.

Figure 10-26 shows an output pin function operation timing chart.

Figure 10-26 Output Pin Function Operation Timing Chart



10.3.2. External Timer Match Starting

If the external timer match starting function is used, the 16-/17-bit down counter can start in synchronization with an external timer.

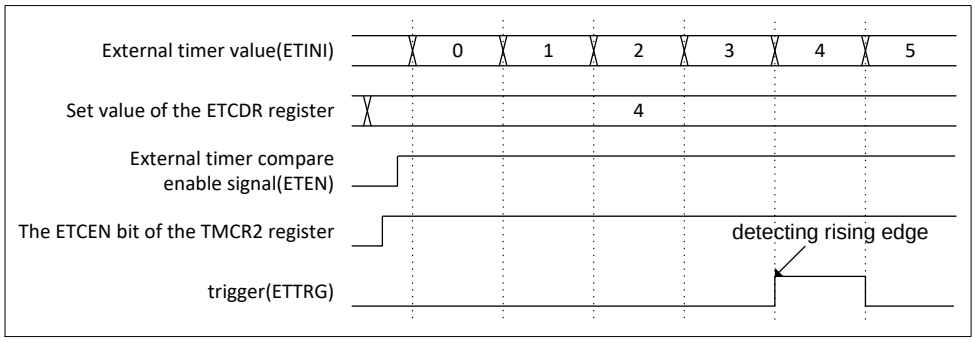
If all the following conditions are satisfied, starting by an external timer is performed.

- The CTEN bit of the TMCR register is "1."
- The ETCEN bit of the TMCR2 register is "1."
- The external timer compare enable signal (ETEN) is active.

When timer is running, the restart by the external timer match starting does not occur. If performing external timer match starting again, perform the following processings.

1. The CTEN bit of the TMCR register is set to "0".
2. The CTEN bit of the TMCR register is set to "1" again.

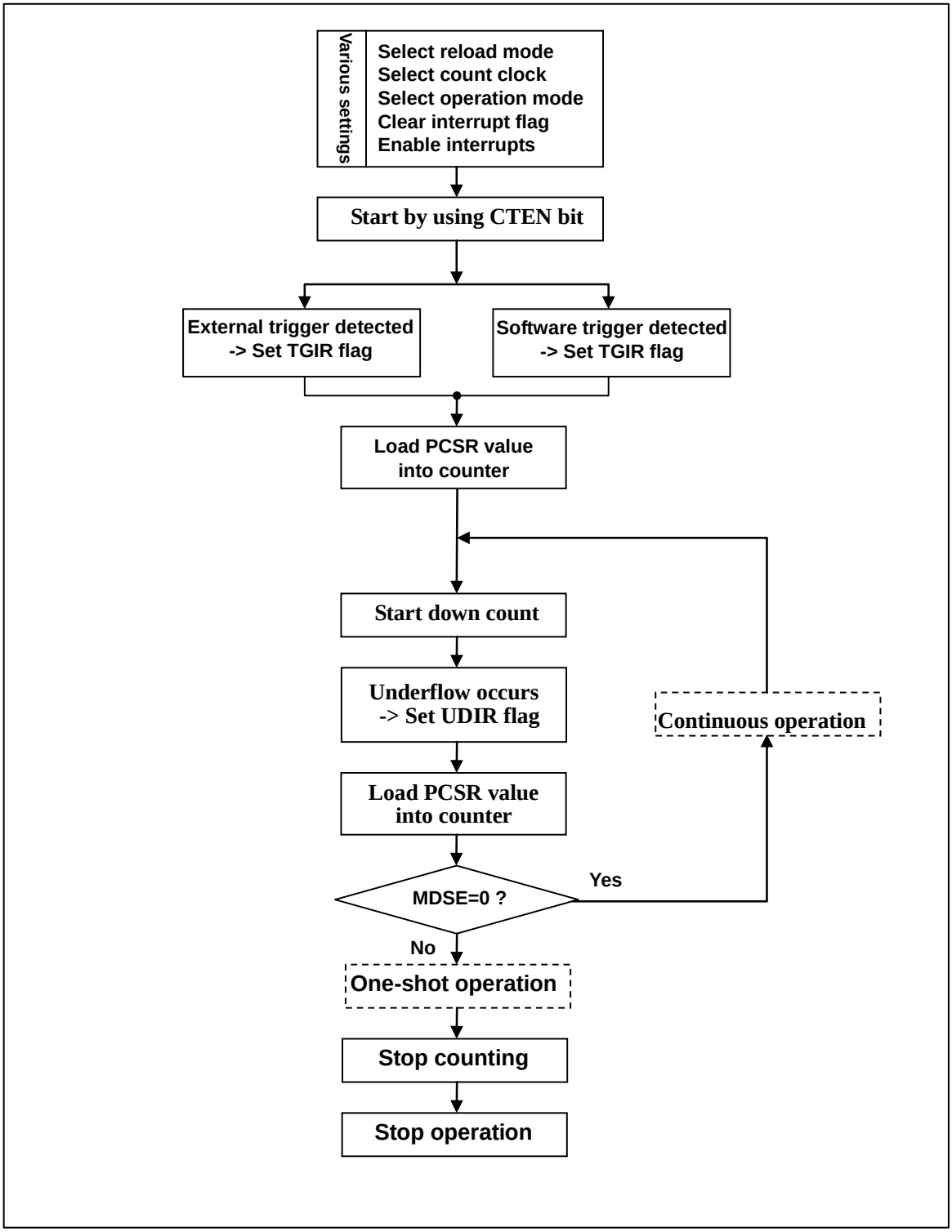
Figure 10-27 The Example of External Timer Match Starting Operation (ETCDR=4)



10.3.3. Reload Timer Operation Flow

This section shows the Reload Timer operation flow.

Reload Timer Operation Flow



10.3.4. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

The Timer Control Register (TMCR) controls timer operation.

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.3.4.1 Timer Control Register (Upper Byte of TMCR)

bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	Reserved		EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0		R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0		0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 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-edges event)
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11:10] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit9:8] EGS1, EGS0: Trigger Input Edge and Gate Function Level Selection Bits

- Trigger input selected (TMCR2.GATE="0")
  - These bits select the valid edge for an input waveform as an external start factor, and they set trigger conditions.
  - For the initial value or the "00" setting, no valid edge is selected for an input waveform, so no external waveform causes a start.
- Gate function selected (TMCR2.GATE="1")
  - These bits select a valid level corresponding to an input waveform as an external count factor, and the down counter counts down only while the selected level is valid.

Note:

- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

- Modify EGS1 and EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

EGS1	EGS0	Description	
		Trigger Input Selected (TMCR2.GATE="0")	Gate Function Selected (TMCR2.GATE="1")
0	0	Trigger input invalid	"L" level
0	1	External trigger(rising edge)	"H" level
1	0	External trigger(falling edge)	"L" level
1	1	External trigger(both edges)	"H" level

10.3.4.2 Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] T32: 32-Bit Timer Selection Bit

- This bit selects the 32-bit timer function.
- If the Reload Timer function is selected with "011" set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit timer mode.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bit can be modified at the same time as "1" is written to the CTEN bit. (See "4. 32-Bit Mode Operation")

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- These bits select the timer function.
- If "011" is set in the FMD2 to FMD0 bits, the Reload Timer function is selected.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

FMD2	FMD1	FMD0	Description
0	0	0	Reset mode
0	0	1	16-bit PWM Timer
0	1	0	16-bit PPG Timer
0	1	1	16/32-bit Reload Timer
1	0	0	16/32-bit PWC Timer
Other than above			Setting prohibited

[bit3] OSEL: Output Polarity Specification Bit

- This bit selects either normal or inverse output as the timer output level.
- When combined with the mode selection bit (bit2 MDSE), this bit generates an output waveform as follows.

MDSE	OSEL	Output Waveform
0	0	Toggle output with "L" at count start time
0	1	Toggle output with "H" at count start time
1	0	Rectangular output of "H" during counting
1	1	Rectangular output of "L" during counting

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2] MDSE: Mode Selection Bit

- If the MDSE bit is set to "0", reload mode is selected. The Cycle Setting Register (PCSR) value is loaded into the counter at the same time that the count value underflows from 0x0000 to 0xFFFF, and the count operation continues.
- If the MDSE bit is set to "1", one-shot mode is selected. When the count value underflows from 0x0000 to 0xFFFF, operation stops.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Reload mode
1	One-shot mode

[bit1] CTEN: Timer Enable Bit

- This bit enables operation of the down counter.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.

Bit	Description
0	Stop operation.
1	Enable operation.

[bit0] STRG: Software Trigger Bit

- If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.
- The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *If "1" is written to the STRG bit, the software trigger becomes valid regardless of the EGS1 and EGS0 settings.*

Bit	Description
0	Invalid
1	Startup by software

10.3.4.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	GATE	Reserved					ETCEN	CKS3
R/W Attribute	R/W	R0,W0					R/W	R/W
Protection Attribute	-							
Initial Value	0	00000					0	0

[bit7] GATE: Gate Input Enable Bit

This bit selects whether to use the external factor pin for the trigger input function or gate function during Reload Timer operation.

- Trigger input function: A countdown begins when a valid edge is input to the external factor pin.
- Gate function: A countdown is performed only while a valid level is being input to the external factor pin.

After configuring the Reload Timer function with the FMD2 to FMD0 bits in the TMCR register, set the GATE bit.

Bit	Description
0	Trigger input function
1	Gate function

[bit6:2] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit1] ETCEN: External Timer Compare Enable Bit

It is the bit which enables match detection of an external timer value and the ETCDR register value.

Bit	Description
0	Match detection disable
1	Match detection enable

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.3.4 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)"

10.3.4.4 Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
R/W Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIE: Trigger Interrupt Request Enable Bit

- This bit controls interrupt requests of the trigger interrupt request bit (bit2 TGIR).
- If the TGIE bit is enabled and the TGIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.TGIEC bit clears this bit.
- Writing "1" to the STCS.TGIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIE: Underflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).
- If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.UDIEC bit clears this bit.
- Writing "1" to the STCS.UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIR: Trigger Interrupt Request Bit

- The TGIR bit is set to "1" when a software trigger or trigger input is detected.
- Writing "1" to the STCC.TGIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] UDIR: Underflow Interrupt Request Bit

- The UDIR bit is set to "1" when the count value underflows and changes from 0x0000 to 0xFFFF.
- Writing "1" to the STCC.UDIRC bit clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

10.3.4.5 Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIEC	Reserved	UDIEC	Reserved	TGIRC	Reserved	UDIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W	R0,W0	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIEC: Trigger Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.TGIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIEC: Underflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.UDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIE bit.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] TGIRC: Trigger Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.TGIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the TGIR bit.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] UDIRC: Underflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC.UDIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the UDIR bit.

10.3.4.6 Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	TGIES	Reserved	UDIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] TGIES: Trigger Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC:TGIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the TGIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] UDIES: Underflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.UDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the UDIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

10.3.5. Cycle Setting Register (PCSR)

The Cycle Setting Register (PCSR) is a register that retains the initial count value. In 32-bit mode, if this is an even-numbered channel, this will display the initial value of the lower 16-bit count. If this is an odd-numbered channel, this will display the initial value of the upper 16-bit count. The initial value at the reset time is undefined. Be sure to access this register with 16- or 32-bit data transfer instructions.

bit	15	14	13	12	11	10	9	8
Field	PCSR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7	6	5	4	3	2	1	0
Field	PCSR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	XXXXXXXX							

These bits compose the register used to set a cycle. An underflow causes a transfer to the Timer Register.

- Use 16- or 32-bit data access for the PCSR register.
- After configuring the Reload Timer function with the FMD2 to FMD0 bits in the TMCR register, set a cycle in the PCSR register.
- To write data to the PCSR register in 32-bit mode, access first the upper 16-bit data (data for the odd-numbered channel) and then the lower 16-bit data (data for the even-numbered channel).

10.3.6. Timer Register (TMR)

The Timer Register (TMR) is a register that can read the count value of a timer. In 32-bit mode, if this is an even-numbered channel, this will display the value of the lower 16-bit count. If this is an odd-numbered channel, this will display the value of the upper 16-bit count. The initial value is undefined.

Be sure to read this register with 16- or 32-bit data transfer instructions.

bit	15	14	13	12	11	10	9	8
Field	TMR[15:8]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	7	6	5	4	3	2	1	0
Field	TMR[7:0]							
R/W Attribute	R,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

The Timer Register (TMR) can read the value of the 16-bit down counter.

- Use 16- or 32-bit data access for the TMR register.
- To read the TMR register in 32-bit mode, access first the lower 16-bit data (data for the even-numbered channel) and then the upper 16-bit data (data for the odd-numbered channel).

10.3.7. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

10.3.8. External Timer Compare Data Register (ETCDR)

The External Timer Compare Data Register (ETCDR) is the register which sets the value which is started by an external timer.

bit	15	14	13	12	11	10	9	8
Field	ETCDR[15:8]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	01111111							

bit	7	6	5	4	3	2	1	0
Field	ETCDR[7:0]							
R/W Attribute	R/W							
Protection Attribute	-							
Initial Value	11111111							

It is a register compared with an external timer signal (ETINI). Refer to "10.3.2 External Timer Match Starting" for details.

- Use 16- or 32-bit data access for the ETCDR register.

10.4. PWC Timer Function

Only one of the following timer functions can be selected for the Base Timer in the FMD2 to FMD0 bit settings in the Timer Control Register (TMCR): 16-bit PWM Timer, 16-bit PPG Timer, 16-/32-bit Reload Timer, and 16-/32-bit PWC Timer. This section explains the timer function with the PWC setting.

10.4.1 Operations of PWC Timer
10.4.2 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)
10.4.3 Data Buffer Register (DTBF)
10.4.4 Base Timer Debug Register (BT_DEBUG)

10.4.1. Operations of PWC Timer

The PWC Timer has a pulse width measurement function. The timer can select 12 types of count clocks, and it can measure the time and cycle between any input pulse events by using a counter. This section shows the basic functions/operations of the pulse width measurement function.

10.4.1.1 Pulse Width Measurement Function

After the start, the function does not perform a count operation until the counter is cleared to 0x0000 and the set measurement start edge is input. The function starts counting up from 0x0001 when the measurement start edge is detected. It stops counting when the measurement end edge is detected. The count value at this time is stored as a pulse width in the register.

An interrupt request can be generated at the measurement end time and at the overflow occurrence time.

After measurement ends, the function operates according to the measurement mode as follows.

- In single measurement mode: It stops operating.
- In continuous measurement mode: It first transfers the counter value to the buffer register and then stops counting until the measurement start edge is input again.

Figure 10-28 Pulse Width Measurement Operation (Single Measurement Mode/"H" Width Measurement)

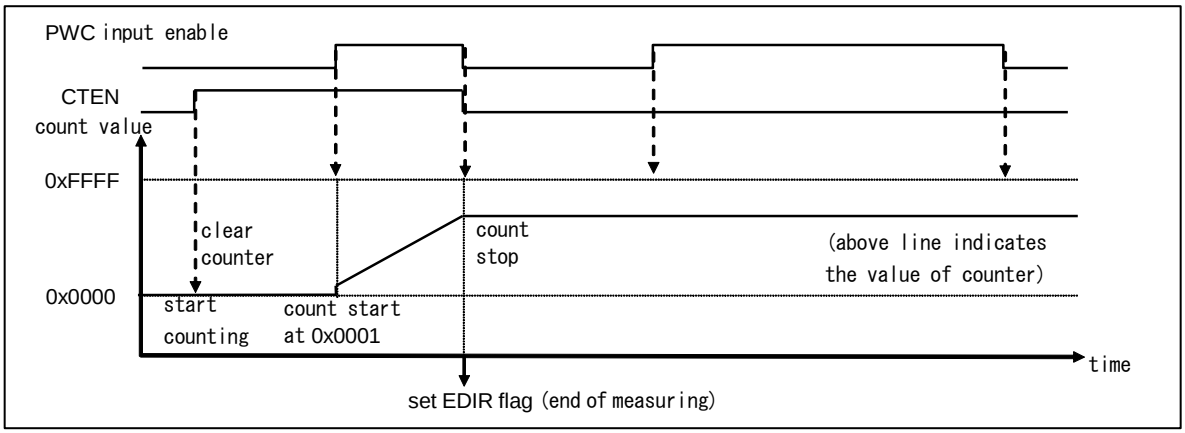
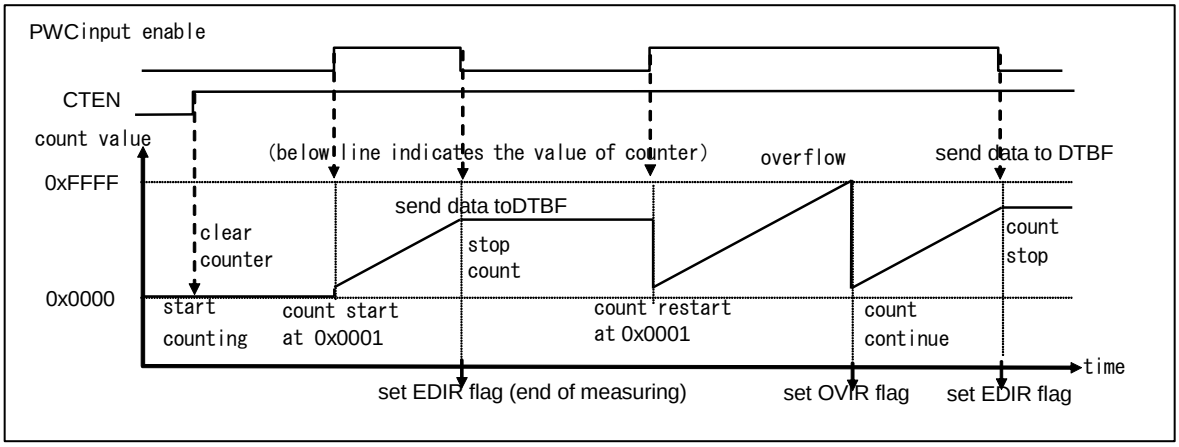


Figure 10-29 Pulse Width Measurement Operation (Continuous Measurement Mode/"H" Width Measurement)



10.4.1.2 Selection of Count Clock

12 types of count clocks can be selected for the counter through the setting of TMCR2 register bit0 (CKS3) and TMCR register bit14 to 12 (CKS2, CKS1, and CKS0).

The following count clocks can be selected.

TMCR2 and TMCR Registers CKS3, CKS2, CKS1, and CKS0 Bits	Internal Count Clock Selected
0000	Internal clock [initial value]
0001	Internal clock divided by 4
0010	Internal clock divided by 16
0011	Internal clock divided by 128
0100	Internal clock divided by 256
0101	Setting prohibited
0110	
0111	
1000	Internal clock divided by 512
1001	Internal clock divided by 1024
1010	Internal clock divided by 2048
1011	Internal clock divided by 2
1100	Internal clock divided by 8
1101	Internal clock divided by 32
1110	Internal clock divided by 64
1111	Setting prohibited

The initial value selected after a reset is the internal clock.

Be sure to select a count clock before starting the counter.

10.4.1.3 Selection of Operation Mode

To select each operation mode/measurement mode, set TMCR.

Operation mode setting ... TMCR bit10 to 8: EGS2, EGS1, EGS0 (Select a measurement edge.)

Measurement mode setting ... TMCR bit2: MDSE (Select single measurement/continuous measurement.)

The following table lists a selection of operation modes.

Operation Mode		MDSE	EGS2	EGS1	EGS0
Rising to falling H pulse width measurement	Continuous measurement mode: Buffer enabled	0	0	0	0
	Single measurement mode: Buffer disabled	1	0	0	0
Rising to rising Cycle measurement between rising edges	Continuous measurement mode: Buffer enabled	0	0	0	1
	Single measurement mode: Buffer disabled	1	0	0	1
Falling to falling Cycle measurement between falling edges	Continuous measurement mode: Buffer enabled	0	0	1	0
	Single measurement mode: Buffer disabled	1	0	1	0
Measurement between rising to falling or falling to rising	Continuous measurement mode: Buffer enabled	0	0	1	1
	Single measurement mode: Buffer disabled	1	0	1	1
Falling to rising L pulse width measurement	Continuous measurement mode: Buffer enabled	0	1	0	0
	Single measurement mode: Buffer disabled	1	1	0	0
Setting prohibited		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

The initial values selected after a reset are "H" pulse width measurement and continuous measurement mode.

Be sure to select an operation mode before starting the counter.

10.4.1.4 Starting and Stopping Pulse Width Measurement

To start/restart/forcibly stop each operation, set bit1 (CTEN bit) in TMCR.

Pulse width measurement is started/restarted by the writing of "1" to the CTEN bit and forcibly stopped by the writing of "0" to the CTEN bit.

CTEN	Function
1	Start/Restart pulse width measurement.
0	Stop pulse width measurement.

10.4.1.5 Post-Start Operation

The operations after the start of pulse width measurement mode do not include counting until the measurement start edge is input. After the detection of the measurement start edge, the 16-bit up counter starts counting from 0x0001.

10.4.1.6 Restart

After the start, a repeated start performed during operation (writing "1" again in a state where the CTEN bit is "1") is called a "restart." The operation in any such restart is described below.

■ In the measurement start edge wait state:

There is no effect on operation.

■ During measurement:

The count is cleared to 0x0000. Then, the measurement start edge wait state begins again. If the measurement end edge is detected at the same time as a restart, the measurement end flag (EDIR) is set to "1". Then, if the mode is continuous measurement mode, the measurement results are transferred to DTBF.

10.4.1.7 About Stopping

In single measurement mode, the count operation is automatically stopped by a counter overflow or the end of measurement, so stopping it does not need to be a concern. In continuous measurement mode, to stop counting before counting is automatically stopped, you need to forcibly stop it.

10.4.1.8 Counter clearing and Initial Value

The 16-bit up counter is cleared to 0x0000 in the following cases.

■ Upon a reset

- When "1" is written to bit1 (CTEN bit) in TMCR (even including restart times)

The 16-bit up counter is initialized to 0x0001 in the following case.

- When the measurement start edge is detected

10.4.1.9 Pulse Width Measurement Operation Details

Single Measurement and Continuous Measurement

The modes for pulse width measurement are a mode where measurement is 1-time only and a mode where measurement is continuous. Each mode is selected with the MDSE bit in TMCR. (See "10.4.1.3 Selection of Operation Mode") The modes differ as described below.

Single measurement mode:

The first input of the measurement end edge stops the counting by the counter and sets the measurement end flag (EDIR) in STC to "1". No subsequent measurements are made.

However, if a restart is performed at the same time, the mode enters the measurement start wait state.

Continuous measurement mode:

The input of the measurement end edge stops the counting by the counter and sets the measurement end flag (EDIR) in STC to "1". The counting remains stopped until the measurement start edge is input again. The counter is initialized to 0x0001 and measurement begins when the measurement start edge is input again. The measurement results of the counter are transferred to DTBF at the measurement end time.

Be sure to select/change the measurement mode while the counter is stopped.

Measurement Result Data

The handling of measurement results and counter values and the function of DTBF differ between single measurement mode and continuous measurement mode. The measurement results from each mode differ as described below.

Single measurement mode:

If DTBF is read during operation, the count value being measured is obtained.

If DTBF is read after measurement ends, the measurement result data is obtained.

Continuous measurement mode:

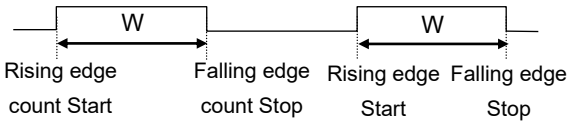
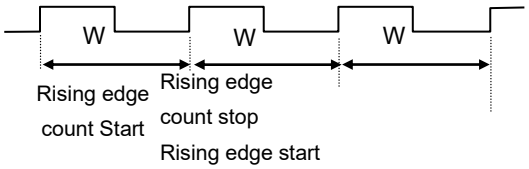
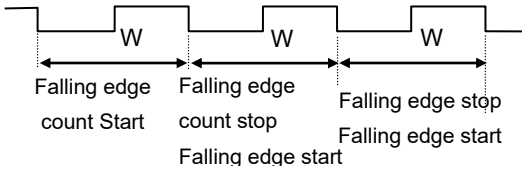
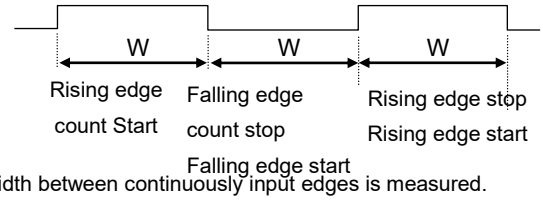
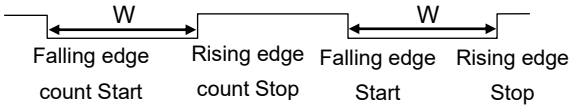
The measurement results of the counter are transferred to DTBF at the measurement end time.

The immediately preceding measurement results are obtained when DTBF is read, and the last measurement results are retained even during a measurement operation. The count value being measured cannot be read.

In continuous measurement mode, if the next measurement ends before the measurement results are read, the last measurement results are overwritten by the new measurement results. In such cases, the error flag (ERR) is set to "1" in STC. The error flag (ERR) is automatically cleared when DTBF is read.

Measurement Modes and Count Operations

The measurement mode can be selected from the following 5 types, depending on which part of the input pulse is measured. The following table explains them.

Measurement Mode	EGS2, 1, 0	Measurement Description (W: Pulse Width Measured)
"H" pulse width measurement	000	 The width of the "H" period is measured. Count (measurement)start: Upon detection of a rising edge Count (measurement)end: Upon detection of a falling edge
Cycle measurement between rising edges	001	 The cycle between rising edges is measured. Count (measurement)start: Upon detection of a rising edge Count (measurement) end: Upon detection of a rising edge
Cycle measurement between falling edges	010	 The cycle between falling edges is measured. Count (measurement) start: Upon detection of a falling edge Count (measurement)end: Upon detection of a falling edge
Pulse width measurement between all edges	011	 The width between continuously input edges is measured. Count (measurement) start: Upon detection of an edge Count (measurement) end: Upon detection of an edge
"L" pulse width measurement	100	 The width of the "L" period is measured. Count (measurement) start: Upon detection of a falling edge Count (measurement) end: Upon detection of a rising edge

Regardless of the measurement mode, after the counter is cleared to 0x0000 by the start of measurement, the counter does not do any counting until the measurement start edge is input. From the input of the measurement start edge, the counter continues counting up for each count clock until the measurement end edge is input.

For measurements such as pulse width measurement and cycle measurement between all edges in continuous measurement mode, the end edge is the next measurement start edge.

Pulse Width/Cycle Calculation Method

The method of calculating the pulse width/cycle from the obtained measurement result data in DTBF after measurement ends is shown below.

Tw = n x t	Tw: Measured pulse width/cycle n: Measurement result data in DTBF t: Count clock cycle

Interrupt Request Generation

The following two interrupt requests can be generated.

- Interrupt request due to a counter overflow

If counting up during measurement causes an overflow, the overflow flag (OVIR) is set to "1". Furthermore, if overflow interrupt requests are enabled, an interrupt request is generated.

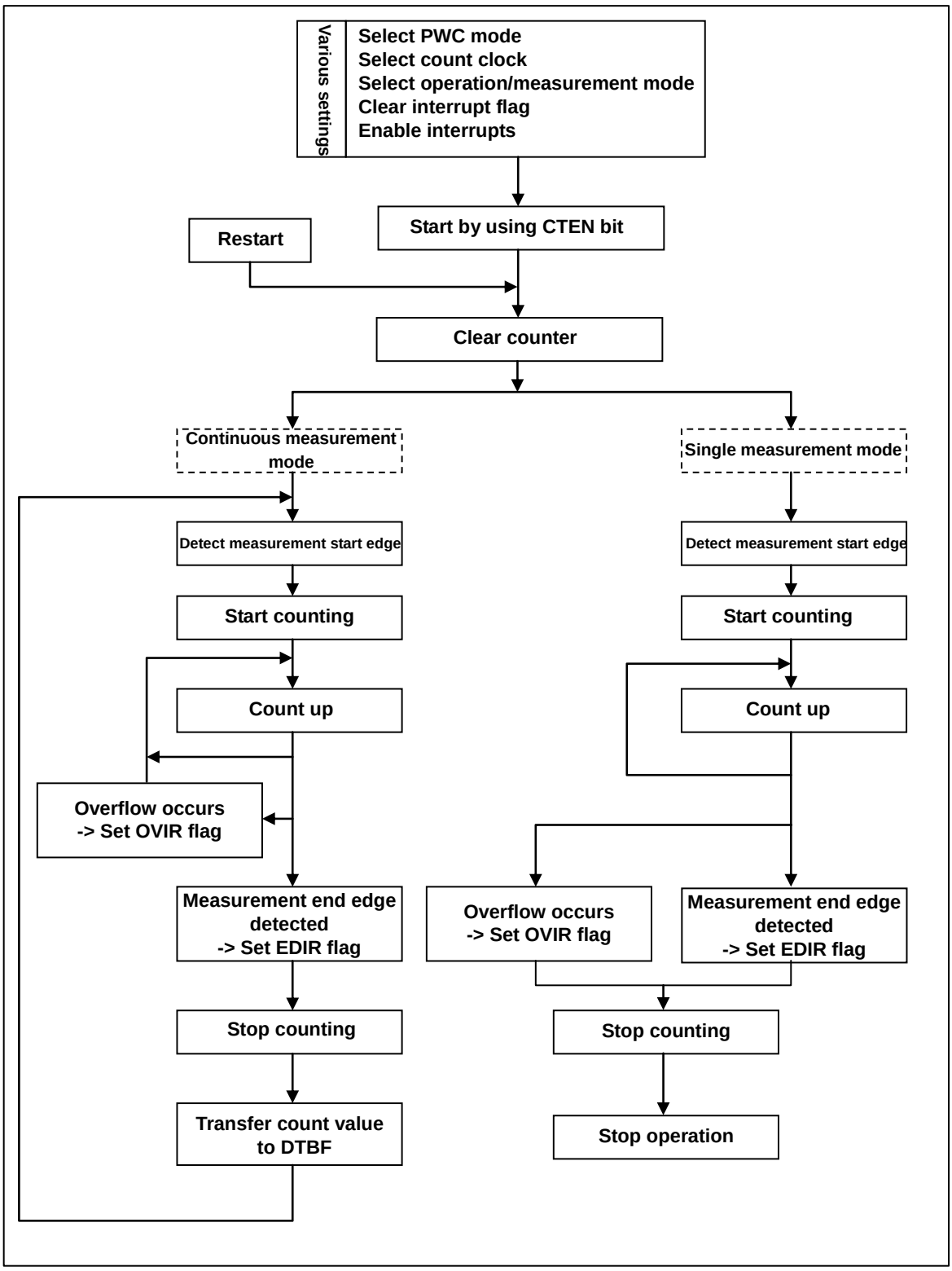
- Interrupt request due to measurement end

If the measurement end edge is detected, the measurement end flag (EDIR) in STC is set to "1". Furthermore, if measurement end interrupt requests are enabled, an interrupt request is generated.

The measurement end flag (EDIR) is automatically cleared by the reading of the measurement results, DTBF.

Pulse Width Measurement Operation Flow

Figure 10-30 Pulse Width Measurement Operation Flow



10.4.2. Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)

The Timer Control Register (TMCR) controls timer operation.

For details on writing to the Status Control Register (STC), see "9. Notes on Using the Base Timer".

10.4.2.1 Timer Control Register (Upper Byte of TMCR)

Bit	15	14	13	12	11	10	9	8
Field	Reserved	CKS2	CKS1	CKS0	Reserved	EGS2	EGS1	EGS0
R/W Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[TMCR2:bit0,bit14:12] CKS3 to CKS0: Count Clock Selection Bits

- These bits select a count clock for the 16-bit down counter.
- Any modification to the count clock is reflected immediately after the setting is changed. Therefore, modify CKS3 to CKS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

CKS3	CKS2	CKS1	CKS0	Description
0	0	0	0	Internal clock
0	0	0	1	Internal clock divided by 4
0	0	1	0	Internal clock divided by 16
0	0	1	1	Internal clock divided by 128
0	1	0	0	Internal clock divided by 256
0	1	0	1	Setting prohibited
0	1	1	0	
0	1	1	1	
1	0	0	0	Internal clock divided by 512
1	0	0	1	Internal clock divided by 1024
1	0	1	0	Internal clock divided by 2048
1	0	1	1	Internal clock divided by 2
1	1	0	0	Internal clock divided by 8
1	1	0	1	Internal clock divided by 32
1	1	1	0	Internal clock divided by 64
1	1	1	1	Setting prohibited

[bit11] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit10:8] EGS2 to EGS0: Measurement Edge Selection Bits

- These bits set a measurement edge condition.
- Modify EGS2 to EGS0 while counting is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

EGS2	EGS1	EGS0	Description
0	0	0	"H" pulse width measurement (rising to falling)
0	0	1	Cycle measurement between rising edges (rising to rising)
0	1	0	Cycle measurement between falling edges (falling to falling)
0	1	1	Pulse width measurement between all edges (rising or falling to falling or rising)
1	0	0	"L" pulse width measurement (falling to rising)
1	0	1	Setting prohibited
1	1	0	
1	1	1	

10.4.2.2 Timer Control Register (Lower Byte of TMCR)

bit	7	6	5	4	3	2	1	0
Field	T32	FMD2	FMD1	FMD0	Reserved	MDSE	CTEN	Reserved
R/W Attribute	R/W	R/W	R/W	R/W	R0,W0	R/W	R,W	R0,W0
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] T32: 32-Bit Timer Selection Bit

- This bit selects the 32-bit timer function.
- If the PWC function is selected with "100" set in the FMD2 to FMD0 bits, setting the T32 bit to "1" results in operation in 32-bit PWC mode.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bit can be modified at the same time as "1" is written to the CTEN bit. (See 32-Bit Mode Operation.)

Bit	Description
0	16-bit timer mode
1	32-bit timer mode

[bit6:4] FMD2 to FMD0: Timer Function Selection Bits

- These bits select the timer function.
- If "100" is set in the FMD2 to FMD0 bits, the PWC Timer function is selected.
- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

FMD2	FMD1	FMD0	Description
0	0	0	Reset mode
0	0	1	16-bit PWM Timer
0	1	0	16-bit PPG Timer
0	1	1	16/32-bit Reload Timer
1	0	0	16/32-bit PWC Timer
1	0	1	Setting prohibited
1	1	0	
1	1	1	

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] MDSE: Mode Selection Bit

- Modify these bits while the timer is stopped (CTEN="0"). However, note that the bits can be modified at the same time as "1" is written to the CTEN bit.

Bit	Description
0	Continuous measurement mode (buffer register enabled)
1	Single measurement mode (stop after 1 measurement)

[bit1] CTEN: Timer Enable Bit

- This bit enables the starting or restarting of the up counter.
- If "1" is written with the counter in the operation enabled state (CTEN bit is "1"), a restart is assumed, the counter is cleared, and operation enters the measurement start edge wait state.
- If "0" is written to this bit when counter operation is enabled (CTEN bit is "1"), the counter stops.
- After measurement ends in single measurement mode, CTEN is cleared.

Bit	Description
0	Stop operation.
1	Enable operation.

Note:

- *During timer operation of I/O mode 6 or I/O mode 4, if the falling edge is output from the even-numbered channel, this bit of the odd-numbered channel is cleared to "0".*

[bit0] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

10.4.2.3 Timer Control Register 2 (TMCR2)

bit	7	6	5	4	3	2	1	0
Field	Reserved							CKS3
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit7:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] CKS3: Count Clock Selection Bit

See "Count clock selection bits" in "10.4.2 Timer Control Registers (TMCR, TMCR2), Status Control Register (STC), Status Control Clear Register (STCC), and Status Control Set Register (STCS)"

10.4.2.4 Status Control Register (STC)

bit	7	6	5	4	3	2	1	0
Field	ERR	EDIE	Reserved	OVIE	Reserved	EDIR	Reserved	OVIR
R/W Attribute	R,WX	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] ERR: Error Flag Bit

- This bit is a flag indicating that, in continuous measurement mode, the next measurement has ended before the measurement results in the DTBF register have been read. In this case, the DTBF register values are updated with the new measurement results, so the immediately preceding measurement results are lost.
- Measurement continues regardless of the ERR bit value.
- The ERR bit is read-only. Writing to the bit has no effect on the bit value.
- Reading the measurement results (DTBF) clears the ERR bit.

Bit	Description
0	Normal state
1	Overwrite any unread measurement results with the next measurement results.

[bit6] EDIE: Measurement End Interrupt Request Enable Bit

- This bit controls interrupt requests of the measurement end interrupt request bit (bit2 EDIR).
- If the EDIE bit is enabled and the EDIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.EDIEC bit clears this bit.
- Writing "1" to the STCS.EDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIE: Overflow Interrupt Request Enable Bit

- This bit controls interrupt requests of the overflow interrupt request bit (bit0 OVIR).
- If the OVIE bit is enabled and the OVIR bit is set to "1", an interrupt request is issued to the CPU.
- Writing "1" to the STCC.OVIEC bit clears this bit.
- Writing "1" to the STCS.OVIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit2] EDIR: Measurement End Interrupt Request Bit

- This bit indicates that measurement has ended by setting the flag to "1" at the end time.
- Reading the measurement results (DTBF) clears the EDIR bit. The EDIR bit is read-only. Writing to the bit has no effect on the bit value.

Bit	Description
0	Read measurement results (DTBF).
1	Detect the interrupt factor.

[bit1] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit0] OVIR: Overflow Interrupt Request Bit

- The flag is set to "1" when the count value overflows from 0xFFFF to 0x0000.
- Writing "1" to the STCC.OVIRC clears this bit.
- This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

10.4.2.5 Status Control Clear Register (STCC)

bit	7	6	5	4	3	2	1	0
Field	Reserved	EDIEC	Reserved	OVIEC	Reserved			OVIRC
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			R0,W
Protection Attribute	-							
Initial Value	0	0	0	0	000			0

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] EDIEC: Measurement End Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.EDIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the EDIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIEC: Overflow Interrupt Request Enable Clear Bit

- If "1" is written to this bit, the STC.OVIE bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the OVIE bit.

[bit3:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] OVIRC: Overflow Interrupt Request Clear Bit

- If "1" is written to this bit, the STC:OVIR bit is cleared to "0".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Clear the OVIR bit.

10.4.2.6 Status Control Set Register (STCS)

bit	7	6	5	4	3	2	1	0
Field	Reserved	EDIES	Reserved	OVIES	Reserved			
R/W Attribute	R0,W0	R0,W	R0,W0	R0,W	R0,W0			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit7] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit6] EDIES: Measurement End Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.EDIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the EDIE bit.

[bit5] Reserved: Reserved Bit

The read value is "0".

If writing to this bit, write "0".

[bit4] OVIES: Overflow Interrupt Request Enable Set Bit

- If "1" is written to this bit, the STC.OVIE bit is set to "1".
- The read value of this bit is always "0".

Bit	Description
0	Invalid
1	Set the OVIE bit.

[bit3:0] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

10.4.3. Data Buffer Register (DTBF)

The Data Buffer Register (DTBF) is a register that can read the measurement value or count value of the PWC Timer. In 32-bit mode, if this is an even-numbered channel the lower 16 bits will be read similarly, for odd number channels, the upper 16 bits will be read.

Be sure to read this register with 16- or 32-bit data transfer instructions.

bit	15	8
Field	DTBF[15:8]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000	

bit	7	0
Field	DTBF[7:0]	
R/W Attribute	R,WX	
Protection Attribute	-	
Initial Value	00000000	

- These bits compose the DTBF register as a read-only register in either continuous measurement mode or single measurement mode. Writing does not change the register value.
- In continuous measurement mode (TMCR bit2 MDSE = "0"), the register is used as a buffer register to store the last measurement results.
- In single measurement mode (TMCR bit2 MDSE = "1"), the up counter is directly accessed with the DTBF register. Reading is allowed even during counting so that the count value can be read. After measurement ends, the measurement results are stored as is.
- Use 16- or 32-bit data access for the DTBF register.

10.4.4. Base Timer Debug Register (BT_DEBUG)

The Base Timer Debug Register (BT_DEBUG) performs enable/disable setting of debug.

bit	15	8
Field	Reserved	
R/W Attribute	R0,W0	
Protection Attribute	-	
Initial Value	00000000	

bit	7	0
Field	Reserved	DBGEN
R/W Attribute	R0,W0	R/W
Protection Attribute	-	
Initial Value	0000000	0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

■ It is the bit which permits stopping operation of a Base Timer with the debug signal (DEBUG).

Bit	Description
0	Disable
1	Enable

CHAPTER 17: Base Timer I/O Selection Function

This chapter explains the base timer I/O selection function.

1. **Overview**
2. **Configuration**
3. **Explanation of Operation**
4. **Registers**

CODE:BTSEL-JUPI-E2

1. Overview

This section provides an overview of the base timer I/O selection function.

The base timer I/O selection function is a function for selecting a signal I/O method for the base timer by setting an I/O mode.

By switching the timer function, the base timer mounted in a channel can be used as any of the timers described below for each channel. The I/O method for the respective functions can be selected.

Timer outputs can be simultaneously read using TOUT Read Register (BT_BTTRR).

The following 7 patterns can be selected for I/O pin connections.

- 16-bit timer standard mode
- 32-bit timer full-function mode
- PPG trigger 2-channel sharing mode
- Timer start/stop mode
- Simultaneous soft start mode
- Timer start/stop and simultaneous soft start mode
- Timer start mode

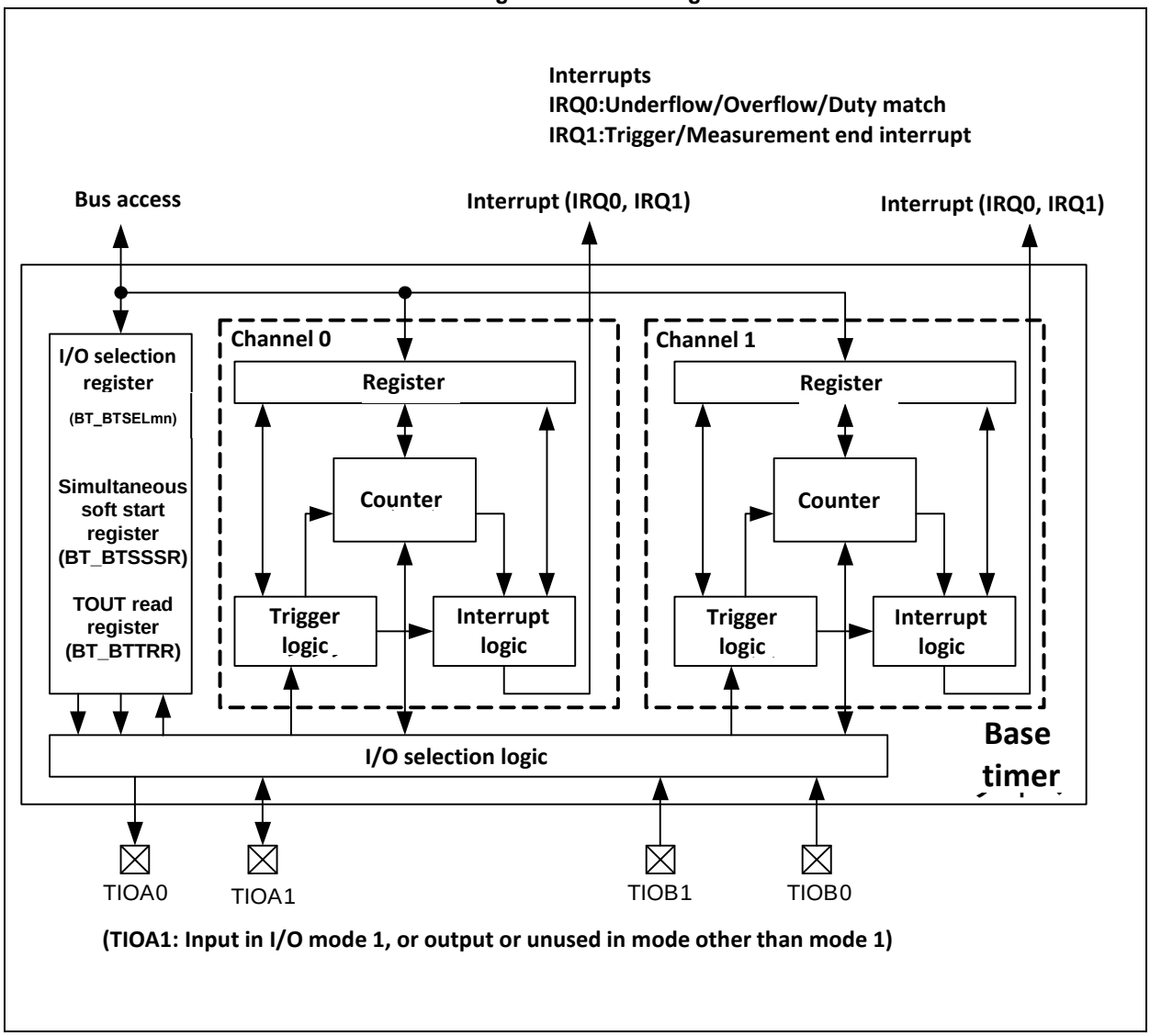
The 32-bit reload timer and 32-bit PWC timer can be implemented by using 2 channels of the mounted base timer. These 2 channels are channel m (m is an even number) and channel n (n = m + 1).

2. Configuration

The 32-bit reload timer and 32-bit PWC timer can be implemented by using 2 channels of the mounted base timer. These 2 channels are channel m (m is an even number) and channel n (n = m + 1).

This section explains the configuration of the base timer and I/O selection function.

Figure 2-1 Block Diagram



Configuration for channel 0 and channel 1

3. Explanation of Operation

This section explains base timer I/O assignment.

Before using a timer, make an I/O setting for the base timer by using the I/O mode selection bit (BTSEL01). You can select one of the following 7 modes.

■ I/O mode 0: 16-bit timer standard mode

In this mode, the base timer in 1 channel operates individually and separately from the others.

■ I/O mode 1: 32-bit timer full-function mode

The signals of the even-numbered channels of the base timer are individually assigned to external pins in operation in this mode.

■ I/O mode 2: PPG trigger 2-channel sharing mode

This mode enables simultaneous input of external start triggers to the base timers of 2 channels. The base timers of the 2 channels can be started simultaneously.

■ I/O mode 4: Timer start/stop mode

In this mode, an even-numbered channel controls the start/stop of an odd-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel, and stopped by the falling edge*.

■ I/O mode 5: Simultaneous soft start mode

In this mode, software starts multiple channels simultaneously.

■ I/O mode 6: Soft start timer start/stop mode

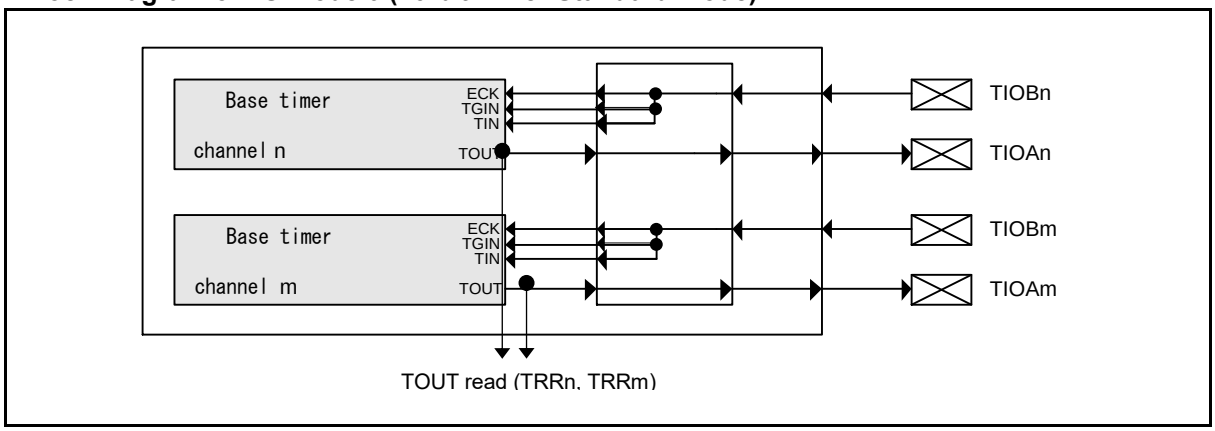
In this mode, an even-numbered channel controls the start/stop of an odd-numbered channel. Software starts the even-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel, and stopped by the falling edge*.

■ I/O mode 7: Timer start mode

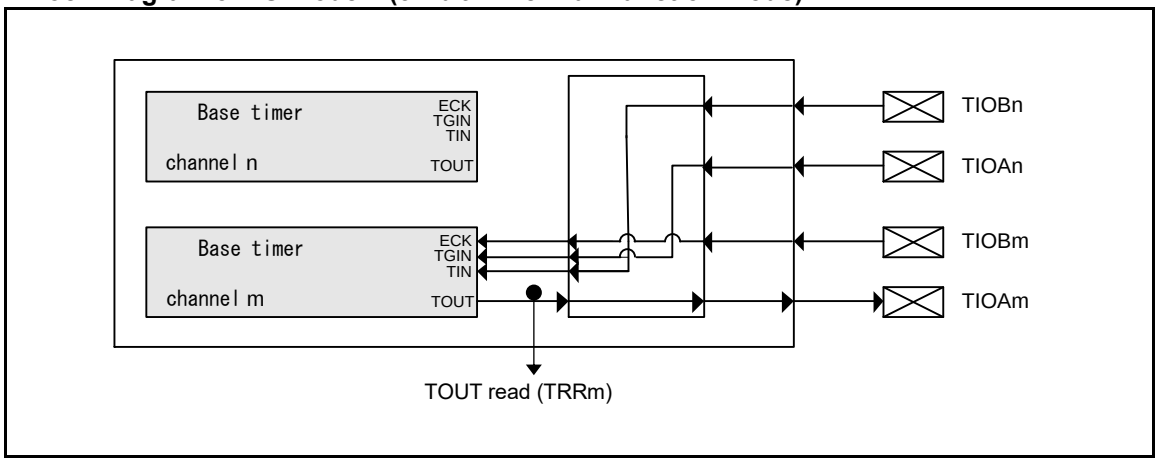
In this mode, an even-numbered channel controls the start of an odd-numbered channel. The odd-numbered channel is started by the rising edge* of an output signal from the even-numbered channel.

*: Use the trigger input selection bit (BTxx_TMCR:EGS) for the setting.

Block Diagram of I/O Mode 0 (16-bit Timer Standard Mode)



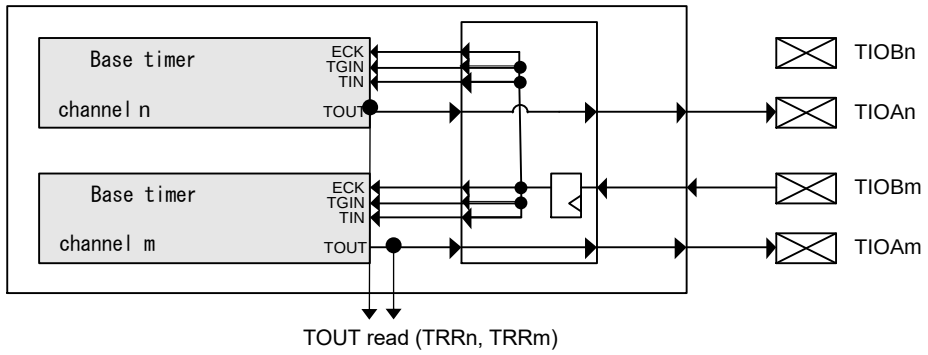
Block Diagram of I/O Mode 1 (32-bit Timer Full-function Mode)



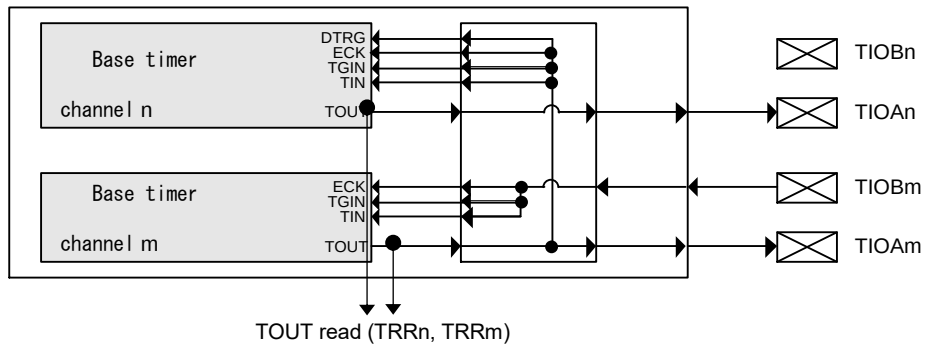
Note:

- *If I/O mode 1 is set, set read value of TRRn is undefined.*

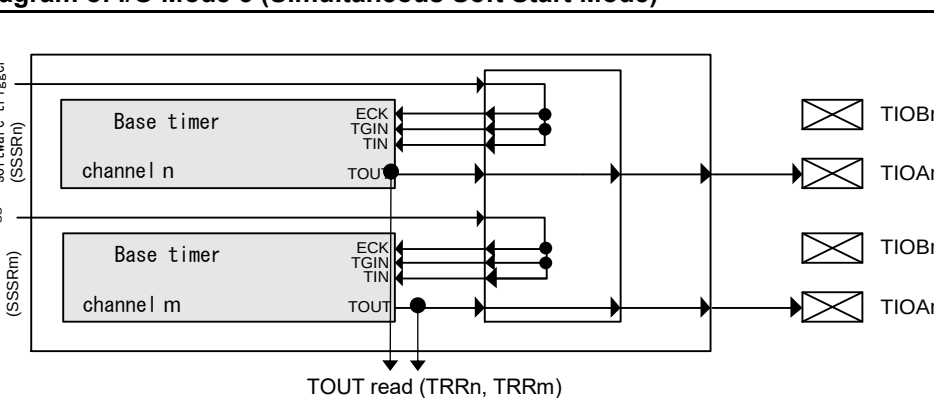
Block Diagram of I/O Mode 2 (PPG Trigger 2-channel Sharing Mode)



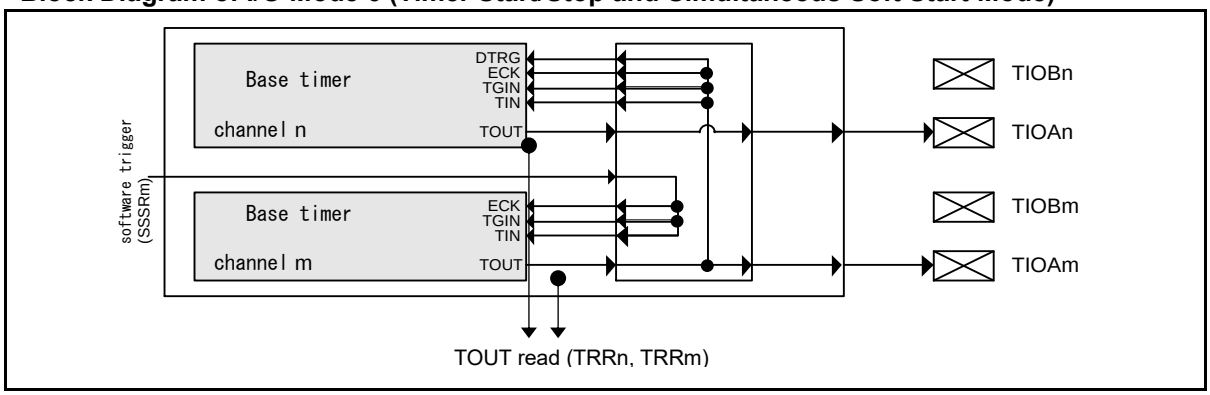
Block Diagram of I/O Mode 4 (Timer Start/Stop Mode)



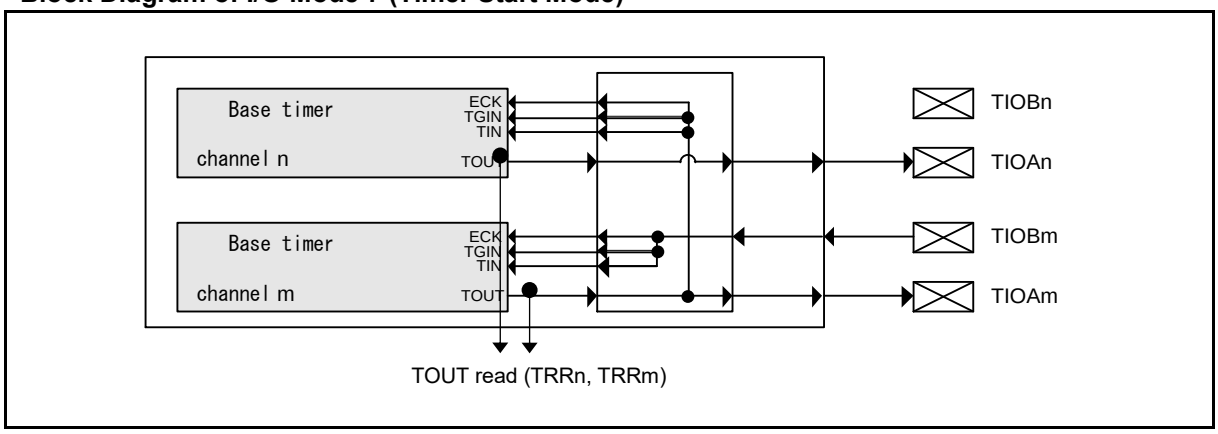
Block Diagram of I/O Mode 5 (Simultaneous Soft Start Mode)



Block Diagram of I/O Mode 6 (Timer Start/Stop and Simultaneous Soft Start Mode)



Block Diagram of I/O Mode 7 (Timer Start Mode)



Note:

- *If I/O mode 1 is set, set TIOAn of the corresponding odd-numbered channel to port input mode by using the GPIO setting.*

4. Registers

This section explains the base timer I/O selection function registers.

Table 4-1 List of Base Timer I/O Selection Registers

Abbreviated Register Name	Register Name	Reference
BT_BTSEL01	I/O selection register (channel 0, 1)	4.1
BT_BTSEL23	I/O selection register (channel 2, 3)	4.1
BT_BTSEL45	I/O selection register (channel 4, 5)	4.1
BT_BTSEL67	I/O selection register (channel 6, 7)	4.1
BT_BTSEL89	I/O selection register (channel 8, 9)	4.1
BT_BTSEL1011	I/O selection register (channel 10, 11)	4.1
BT_BTSEL1213	I/O selection register (channel 12,1 3)	4.1
BT_BTSEL1415	I/O selection register (channel 14,1 5)	4.1
BT_BTSEL1617	I/O selection register (channel 16,1 7)	4.1
BT_BTSEL1819	I/O selection register (channel 18,1 9)	4.1
BT_BTSEL2021	I/O selection register (channel 20, 21)	4.1
BT_BTSEL2223	I/O selection register (channel 22, 23)	4.1
BT_BTSEL2425	I/O selection register (channel 24, 25)	4.1
BT_BTSEL2627	I/O selection register (channel 26, 27)	4.1
BT_BTSEL2829	I/O selection register (channel 28, 29)	4.1
BT_BTSEL3031	I/O selection register (channel 30, 31)	4.1
BT_BTSEL3233	I/O selection register (channel 32, 33)	4.1
BT_BTSEL3435	I/O selection register (channel 34, 35)	4.1
BT_BTSEL3637	I/O selection register (channel 36, 37)	4.1
BT_BTSEL3839	I/O selection register (channel 38, 39)	4.1
BT_BTSEL4041	I/O selection register (channel 40, 41)	4.1
BT_BTSEL4243	I/O selection register (channel 42, 43)	4.1
BT_BTSEL4445	I/O selection register (channel 44, 45)	4.1
BT_BTSEL4647	I/O selection register (channel 46, 47)	4.1
BT_BTSEL4849	I/O selection register (channel 48, 49)	4.1
BT_BTSEL5051	I/O selection register (channel 50, 51)	4.1
BT_BTSEL5253	I/O selection register (channel 52, 53)	4.1
BT_BTSEL5455	I/O selection register (channel 54, 55)	4.1
BT_BTSEL5657	I/O selection register (channel 56, 57)	4.1
BT_BTSEL5859	I/O selection register (channel 58, 59)	4.1
BT_BTSEL6061	I/O selection register (channel 60, 61)	4.1
BT_BTSEL6263	I/O selection register (channel 62, 63)	4.1
BT_BTSSSR0	Simultaneous soft start register (channel 0)	4.2
BT_BTSSSR12	Simultaneous soft start register (channel 12)	4.2
BT_BTSSSR24	Simultaneous soft start register (channel 24)	4.2
BT_BTSSSR36	Simultaneous soft start register (channel 36)	4.2
BT_BTSSSR48	Simultaneous soft start register (channel 48)	4.2
BT_BTSSSR60	Simultaneous soft start register (channel 60)	4.2
BT_BTTRR0	TOUT read register (channel 0)	4.3
BT_BTTRR12	TOUT read register (channel 12)	4.3

Abbreviated Register Name	Register Name	Reference
BT_BTTRR24	TOUT read register (channel 24)	4.3
BT_BTTRR36	TOUT read register (channel 36)	4.3
BT_BTTRR48	TOUT read register (channel 48)	4.3
BT_BTTRR60	TOUT read register (channel 60)	4.3

Tables 4-24-2, 4-3, and 4-4 show the register map of each mode.

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 4-2 Register Map (Channel No.: 0) (12) (24) (36) (48) (60)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR/ Reserved 00000000_00000000 XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSELmn*2 11110000
0x0000_0034	Reserved 11111111_11111111		BT_BTSSSRn*3 11111111_11111111	
0x0000_0038	Reserved 11111111_11111111		BT_BTTRRn*3 11110000_00000000	

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

*2 mn = 01, 1213, 2425, 3637, 4849 and 6061

*3 n = 0, 12, 24, 36, 48 and 60

The "***/***/***/***" expression in each table corresponds to the reload/PWM/ PPG/PWC timer, respectively.

Table 4-3 Register Map (Channel No.: 1, 3, 5, 7, 9, and 11) (13, 15, 17, 19, 21, and 23) (25, 27, 29, 31, 33, and 35) (37, 39, 41, 43, 45, and 47) (49, 51, 53, 55, 57, and 59) (61 and 63)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR / Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

Table 4-4 Register Map (Channel No.: 2, 4, 6, 8, and 10) (14, 16, 18, 20, and 22) (26, 28, 30, 32, and 34) (38, 40, 42, 44, and 46) (50, 52, 54, 56, and 58) (62)

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_0000	Reserved 00000000_00000000		BTxx_PCSR/ BTxx_PCSR/ BTxx_PRLL / Reserved XXXXXXXX_XXXXXXXX	
0x0000_0004	Reserved 00000000_00000000		Reserved/ BTxx_PDUT/ BTxx_PRLH/ BTxx_DTBF XXXXXXXX_XXXXXXXX / 00000000_00000000 (BTxx_DTBF)	
0x0000_0008	Reserved/ BTxx_TMR/ Reserved/ Reserved 00000000_00000000		BTxx_TMR/ BTxx_TMR/ BTxx_TMR / Reserved 00000000_00000000 / XXXXXXXX_XXXXXXXX*1	
0x0000_000C	Reserved 00000000_00000000		BTxx_TMCR 00000000_00000000	
0x0000_0010	Reserved 00000000_00000000		BTxx_TMCR2 00000000	BTxx_STC 00000000
0x0000_0014	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCC 00000000
0x0000_0018	Reserved 00000000_00000000		Reserved 00000000	BTxx_STCS 00000000

Offset	Register Name/Initial Value			
	+3	+2	+1	+0
0x0000_001C	Reserved 00000000_00000000		Reserved/ BTxx_PSDR/ Reserved/ Reserved 00000000_00000000	
0x0000_0020	Reserved 00000000_00000000		Reserved/ BTxx_ADTR/ Reserved/ Reserved 00000000_00000000	
0x0000_0024	Reserved 00000000_00000000		BT_DEBUG 00000000	
0x0000_0028	Reserved 00000000_00000000_00000000_00000000			
0x0000_002C	Reserved 00000000_00000000		ETCDR/ ETCDR / ETCDR/ Reserved 01111111_11111111 / 00000000_00000000 (Reserved)	
0x0000_0030	Reserved 11111111_11111111		Reserved 11111111	BT_BTSELmn*2 11110000

*1 The initial value is XXXXXXXX_XXXXXXXX only during reload timer operation.

*2 mn = 23, 45, 67, 89, 1011, 1415, 1617, 1819, 2021, 2223, 2627, 2829, 3031, 3233, 3435, 3839, 4041, 4243, 4445, 4647, 5051, 5253, 5455, 5657, 5859 and 6263

The "***/***/***/***" expression in each table corresponds to the reload/PWM/PPG/PWC timer, respectively.

4.1. I/O Selection Registers (BT_BTSELmn)

This section shows the bit configuration of the I/O selection registers.

These bits set the I/O modes of the 2 channels of base timer channel m (m is 0 or an even number) and channel n (n = m +1: odd number) for the following connection.

Bit	31	8
Field	Reserved	
R/W Attribute	R1,WX	
Protection Attribute	-	
Initial Value	11111111_11111111_11111111	

Bit	7	6	5	4	3	2	1	0
Field	Reserved				BTSEL01			
R/W Attribute	R1,WX				R/W			
Protection Attribute	-							
Initial Value	1111				0000			

[bit31:4] Reserved: Reserved bits

[bit3:0] BTSEL01: I/O mode selection bits

These bits set the I/O modes of the 2 channels of base timer channel m and channel n for the following connection.

Bits				Description
0	0	0	0	I/O mode 0: 16-bit timer standard mode
0	0	0	1	I/O mode 1: 32-bit timer full-function mode
0	0	1	0	I/O mode 2: PPG trigger 2-channel sharing mode
0	0	1	1	Setting prohibited
0	1	0	0	I/O mode 4: Timer start/stop mode
0	1	0	1	I/O mode 5: Simultaneous soft start mode
0	1	1	0	I/O mode 6: Timer start/stop and simultaneous soft start mode
0	1	1	1	I/O mode 7: Timer start mode
1	x	x	x	Setting prohibited

Note:

- *You cannot initialize this register by setting reset mode (TMCR:FMD2 to 0="0b000"). After setting reset mode, rewrite this register.*

4.2. Simultaneous Soft Start Register (BT_BTSSSR)

This section shows the bit configuration of the simultaneous soft start register.

These bits represent input signals in I/O modes 5 and 6. This register can be used to generate triggers for all channels simultaneously.

Bit	31	16
Field	Reserved	
R/W Attribute	R1,WX	
Protection Attribute	-	
Initial Value	11111111_11111111	

Bit	15	14	13	12	11	10	9	8
Field	Reserved				SSSR11	SSSR10	SSSR9	SSSR8
R/W Attribute	R1, WX				R1,W	R1,W	R1,W	R1,W
Protection Attribute	-							
Initial Value	1111				1	1	1	1

Bit	7	6	5	4	3	2	1	0
Field	SSSR7	SSSR6	SSSR5	SSSR4	SSSR3	SSSR2	SSSR1	SSSR0
R/W Attribute	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W
Protection Attribute	-							
Initial Value	1	1	1	1	1	1	1	1

[bit31:12] Reserved: Reserved bits

[bit11:0] SSSR[11:0]: Simultaneous soft start bits

These bits represent input signals in I/O modes 5 and 6. For details on connections, see the block diagram of each I/O mode in "3. Explanation of Operation."

Writing "1" starts the corresponding channel, and writing "0" has no effect. Up to 12 channels with channel numbers 0 to 11 can be started simultaneously.

Correspondence of the channel number is below.

BT_BTSSSR0 : SSSR0-11 = ch0-11

BT_BTSSSR12 : SSSR0-11 = ch12-23

BT_BTSSSR24 : SSSR0-11 = ch24-35

BT_BTSSSR36 : SSSR0-11 = ch36-47

BT_BTSSSR48 : SSSR0-11 = ch48-59

BT_BTSSSR60 : SSSR0-3 = ch60-63

SSSR [X]	Description
0	No operation
1	Assigns the "1" pulse to input and starts the corresponding channel.

[X] represents the channel number of the base timer. It is a value from 0 to 11.

4.3. TOUT Read Register (BT_BTTRR)

This section shows the bit configuration of the TOUT read register.

By using these bits, software can read TOUT status from the Base Timer channels.

Bit	31							16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111_11111111							

Bit	15	14	13	12	11	10	9	8
Field	Reserved				TRR11	TRR10	TRR9	TRR8
R/W Attribute	R1, WX				R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	1111				0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit31:12] Reserved: Reserved bits

[bit11:0] TRR[11:0]: Simultaneous TOUT read bits

These bits represent TOUT status of the Base Timer channels.. For details on connections, see the block diagram of each I/O mode in 3. Explanation of Operation.

TRR[X]	Description
0	TOUT status is '0'
1	TOUT status is '1'

[X] represents the channel number of the base timer. It is a value from 0 to 11.

CHAPTER 18: Base Timer Simultaneous Operation

This chapter explains the base timers simultaneous Soft Start .

1. Overview
2. Configuration
3. Explanation of Simultaneous Soft Start Operation
4. Overview of the 16-bit Global Timer
5. Explanation of the 16-bit Global Timer Operation
6. Registers of the 16-bit Global Timer
7. Precautions for Using This Device

CODE: BTSIM-JUPI-E3

1. Overview

This section provides an overview of the Base Timer simultaneous soft start.

There are the undermentioned two operational modes in the soft start of the Base timer simultaneous.

- The operation by the write to simultaneous soft start registers.
  (Refer to the chapter of Base Timer I/O Selection Function on this manual.)

- The operation which uses an external timer
  (Refer to the chapter of Base Timer 10.2.4 External Timer Match Starting on this manual.)

This Chapter describes the operation which uses an external timer.

As an external Timer, it uses Global Timer described in this Chapter.

2. Configuration

This section explains the configuration of the Base Timer simultaneous start which uses a Global Timer

From the Global Timer, the undermentioned signals are outputted and it is inputted into each Base Timer.

■ Global timer value

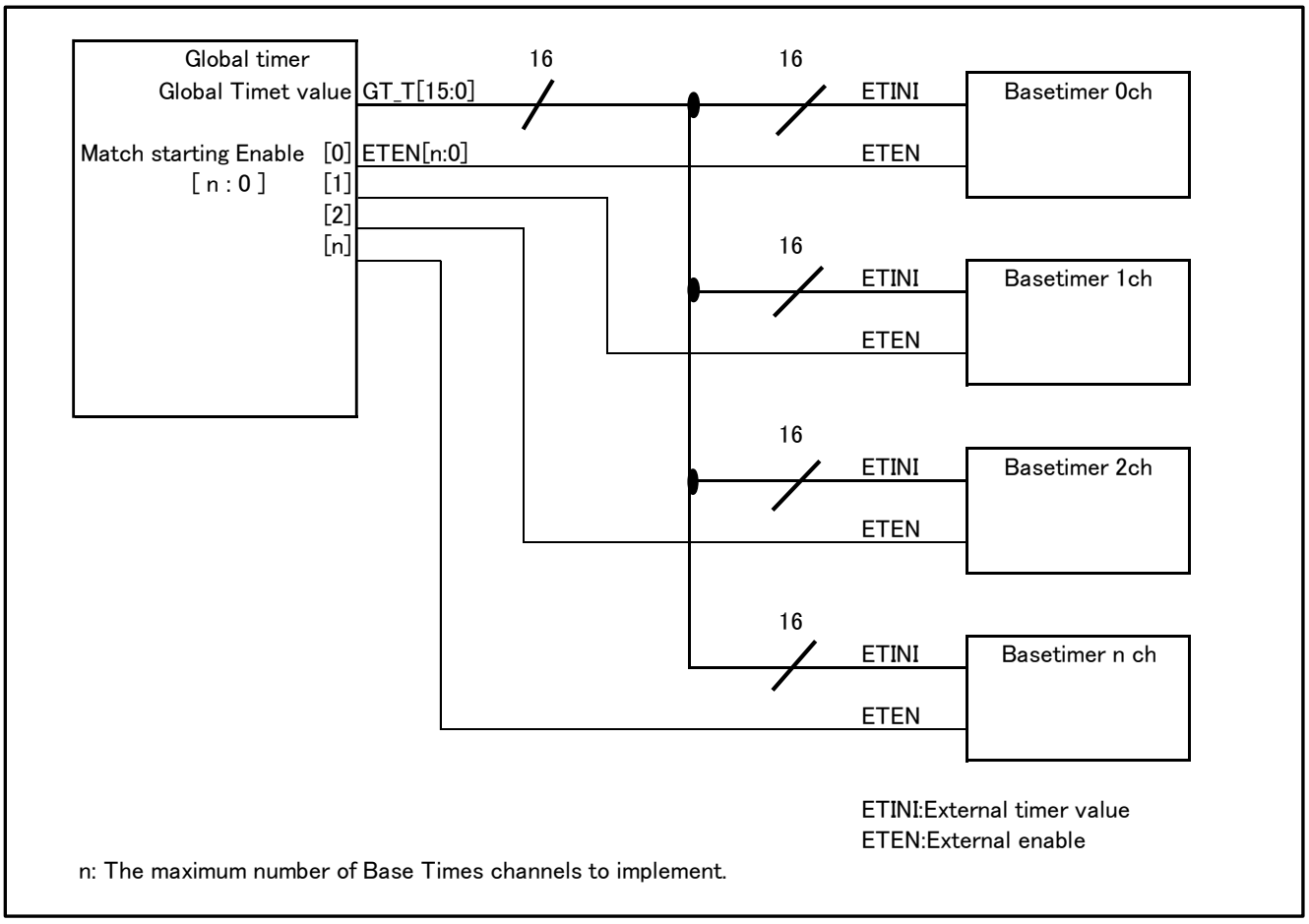
It is a 16-bit timer counter value in the Global Timer.

■ Match starting Enable

It outputs, the Global Timer match enabling signal to each base timer channel.

.

Figure 2-1 Block Diagram



3. Explanation of Simultaneous Soft Start Operation

This section explains the operation of Base Timer Simultaneous Soft Start.

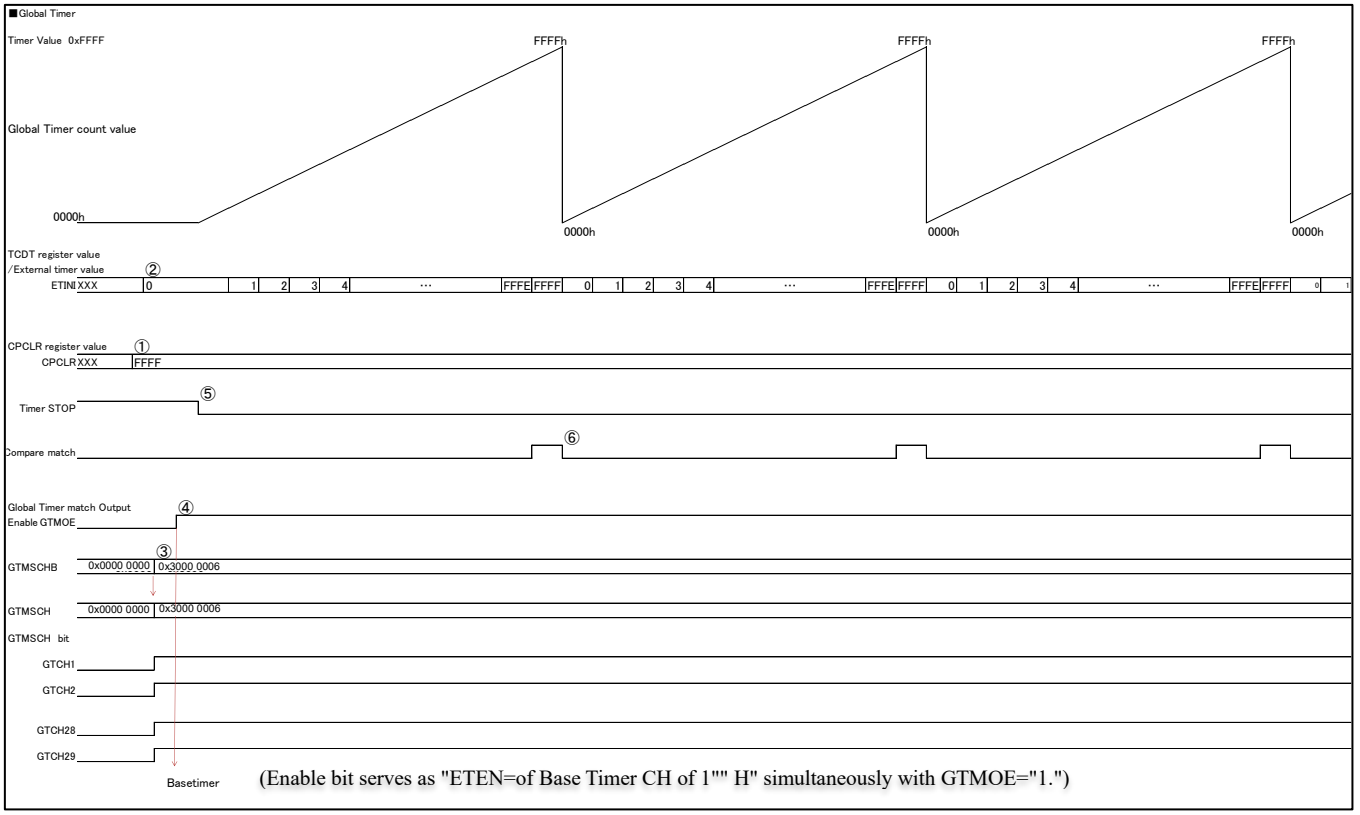
The Procedure in which the simultaneous soft start of the Base Timer was simplified below is shown.

1. It performs an output set of a waveform set of the Base Timer, etc. of operation
2. It performs a set of the Global Timer of operation.
3. Initialization of Global Timer.
4. Set of the Base Timer channel to use an external timer match startup.
5. Global Timer Start.
6. A match of the Global Timer value and the value of external timer compare data in the Base Timer will start the Base Timer.

3.1. Global Timer and Base Timers Operation Waveform

The waveform of operation which controls STOP bit in Timer State Control Register (TCCS) starts, and starts Global Timer, and each Base Timers starts is shown below.

Figure 3-1 Global Timer Waveform (It sets a STOP bit as "0" from "1", and starts Global Timer.)



① Global Timer various sets.

It writes an any value in the Global Clear Register (CPCLR), and sets up the Global Timer period.（ex:0xFFFF）

It writes any value in GT_CLK [3:0], and sets up Global Timer counter clock frequency.

MODE set by GT_MSCTR.MODE bit.

(In the above-mentioned waveform, MODE Bit is "0" and is buffer disable.)

② It writes "0x0000" in the Global Timer Data Register (GT_TCDT), and clears the Global Timer value.

③ It writes an arbitrary value in Global Timer Match Channel Buffer Register (GT_MSCHB0~1).

(It writes "1" in Global Timer Match Starting Channel Buffer Bits (GTCH) of the Base Timer channel to use an external timer match start.)

Since a timer is stopping, the Global Timer Match Channel Buffer Register (GT_MSCHB0~1) set point is immediately transmitted to Global Timer Match Starting Channel register (GT_MSCH0~1).

For output disable (GTMOE="0"), All External Timer Match Enabling Signals (ETEN) are "L."

④ After a Global Timer Match Channel Buffer Register (GT_MSCHB0~1) set, it sets it as GTMOE bit ="1" (Enable).

According to the Global Timer Match Starting Channel register (GT_MSCH0~1) set point, the External Timer Match Enable Signal (ETEN) of each Base Timer Channel changes to "H".

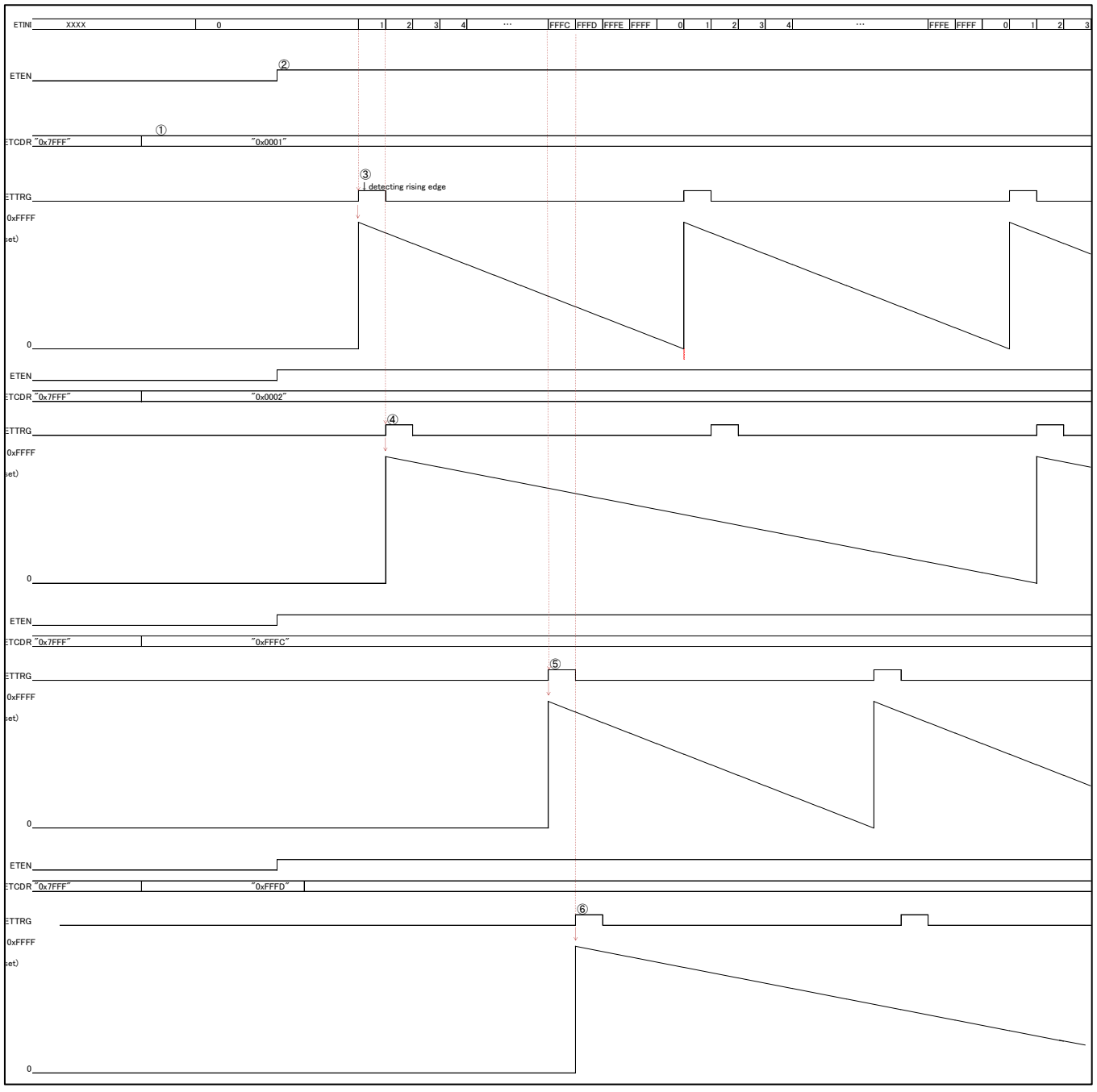
⑤ It writes "0" in STOP bit of Global Timer State Control Register(GT_TCCS).

Global Timer start. (It does not stop Global Timer in start once.)

⑥ When a match the Global Timer Compare Clear Register (GT_CPCLR) value and Global Timer value, it is a comparison clear flag generate.

In accordance with the setup value, the Base Timer operates after the Global Timer start.

Figure 3-2 Base Timer Waveform



① Base timer various sets. (For details, refer to the chapter of Base Timer on this manual)

Note:

- *Please set ETCDR of the channel which uses an external timer match start as a larger value than "0x0000".*

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH1)

④ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH2)

⑤ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH28)

⑥ When a match the External Time Comparison Data Register(ETCDR) value and a Global Timer value , ETTRG signal occurs and Base Timer start.(CH29)

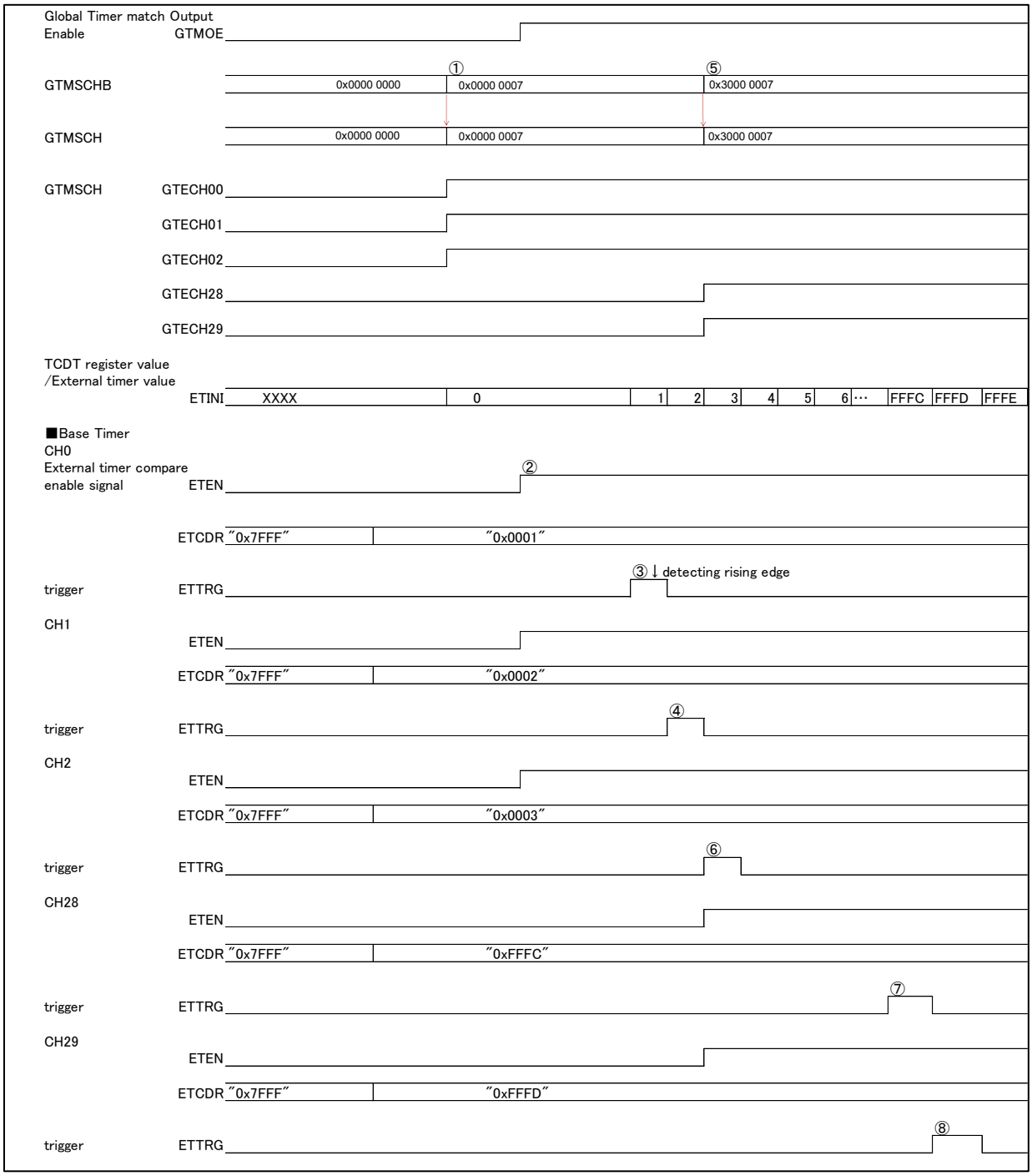
By adding the channel which carries out a Simultaneous Soft Start to GT_MSCHB0~1 Register, and carrying out an enable set, it can carry out the additional operation of Base Timers also after the Global Timer operation.

In that case, Base Timers which want to run an additional simultaneous soft start before needs to be set up.

3.2. Global Timer Match Starting Channel Buffer Operation by MODE Set

The MODE value in Global Timer Match Starting Control register, the update time of the Global Timer Match Starting Channel register value has a difference.

Figure 3-3 The waveform in the Case of Adding Base Timers CH of the Global Timer Match Start Enable. (MODE=0)



① It is a start set from Base Timers CH0 to CH2.

A data is immediately updated by GTMSCH for MODE=0, without carrying out a buffer.

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH0)

④ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH1)

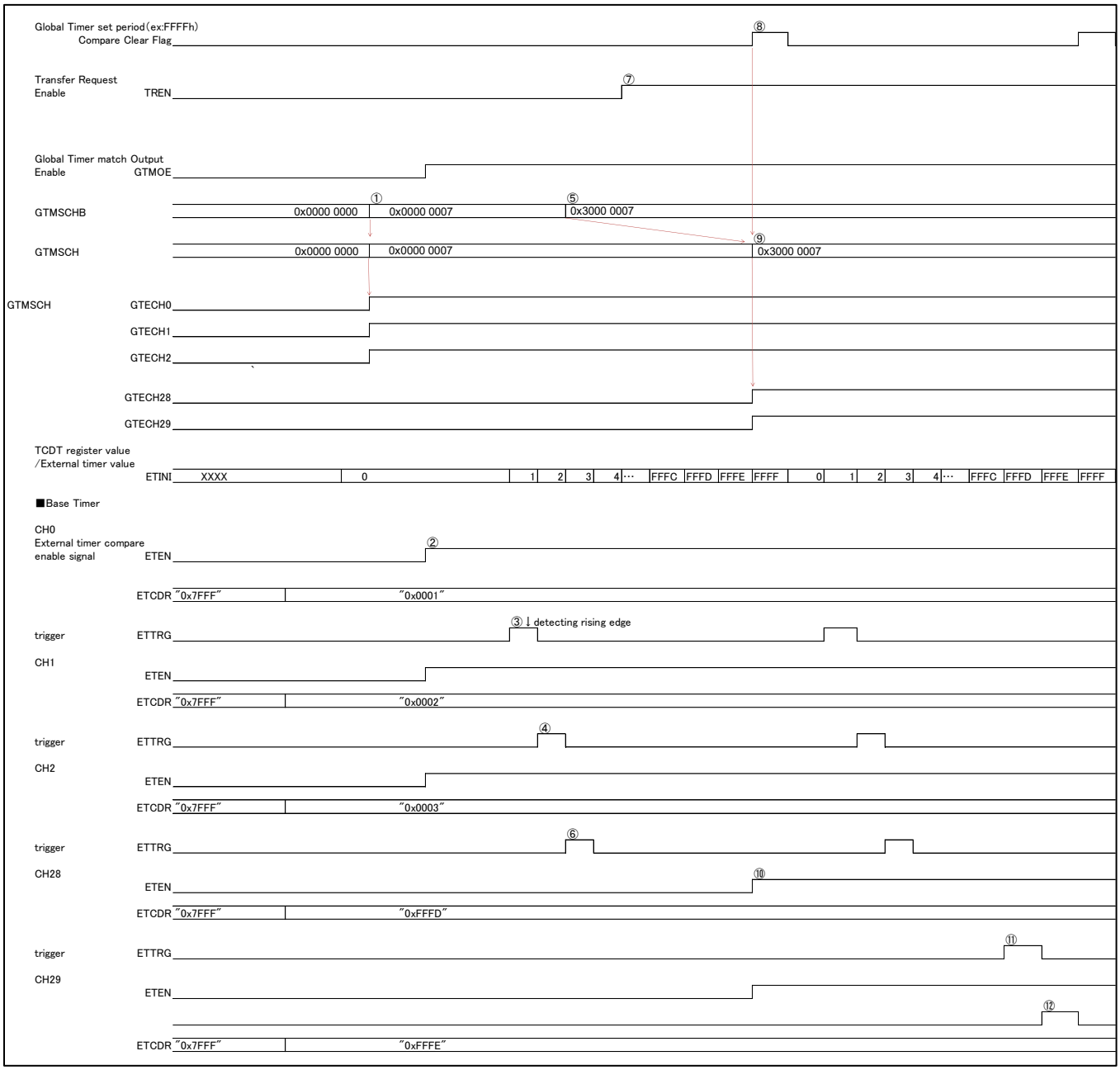
⑤ It is a start set from Base Timers CH28 to CH29.

⑥ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH2)

⑦ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH28)

⑧ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value , ETTRG signal occurs and Base Timer starts.(CH29)

Figure 3-4 The Waveform in the Case of Adding Base Timers CH of the Global Timer Match Start Enable. (MODE=1)



① It is a start set from Base Timers CH0 to CH2.

A data is immediately updated by GTMSCH for MODE=0, without carrying out a buffer.

② Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

③ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value, ETTRG signal occurs and Base Timer starts.(CH0)

④ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value, ETTRG signal occurs and Base Timer starts.(CH1)

⑤ It is a start set from Base Timers CH28 to CH29.

It is not updated until the period comparison match flag of Global Timer is outputted for buffer enable mode.

⑥ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value, ETTRG signal occurs and Base Timer starts.(CH2)

⑦ Enable the transfer request of Register data.

⑧ By the signal outputted for every period of setup Global Timer.

⑨ Global Timer Match Starting Channel register (GT_MSCH0~1) is updated.

⑩ Simultaneously with GTMOE="1", the ETEN signal of Base Timer CH by which the start set was carried out. "H" It becomes.

⑪ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value, ETTRG signal occurs and Base Timer starts.(CH28)

⑫ When a match the External Time Comparison Data Register(ETCDR) value and the Global Timer value, ETTRG signal occurs and Base Timer starts.(CH29)

4. Overview of the 16-bit Global Timer

The 16-bit Global Timer supports the 16-bit up count mode.

Functions of the 16-bit Global Timer

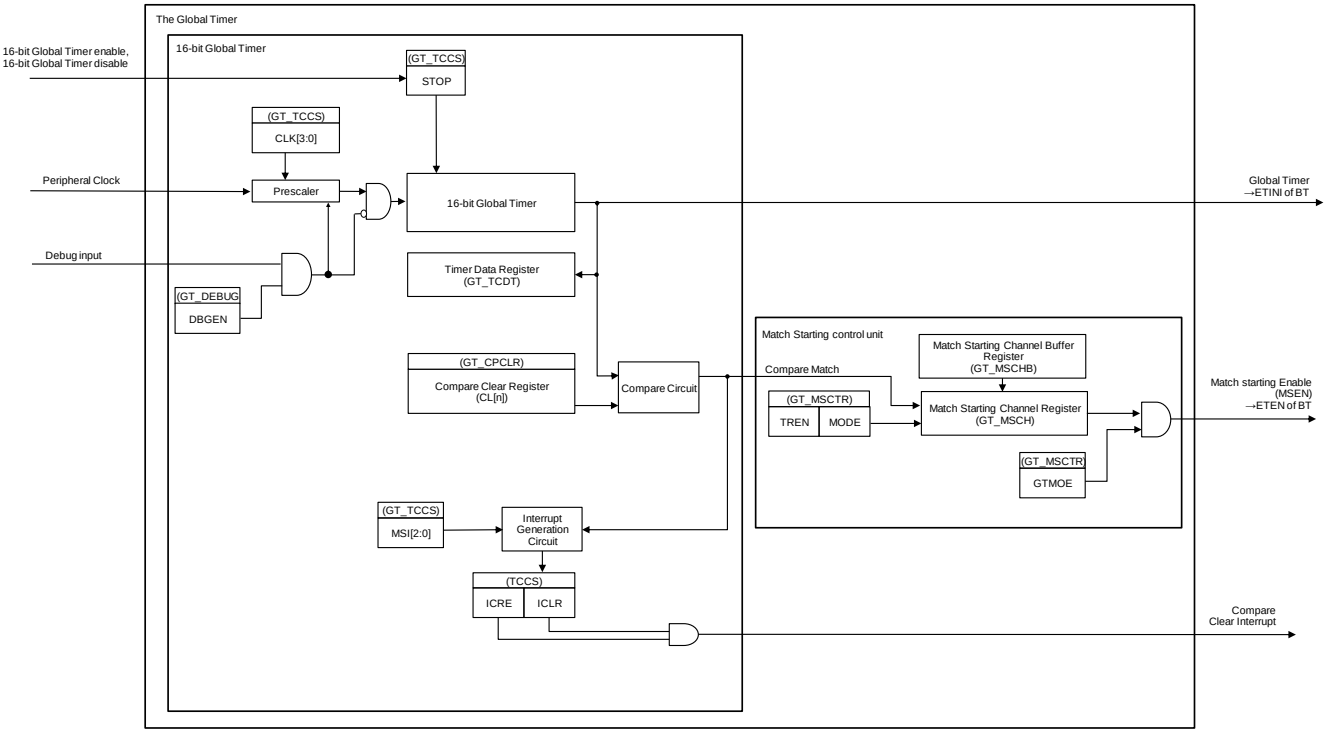
- The 16-bit Global Timer is composed of the 16-bit up counter, control register, 16-bit compare clear register, and prescaler.
- The 12 types of counter operation clocks (φ, φ/2, φ/4, φ/8, φ/16, φ/32, φ/64, φ/128, φ/256, φ/512, φ/1024, and φ/2048) (φ: peripheral clock: CLK_LCP1A) can be selected.
- A compare clear interrupt is generated when the values of the compare clear register and the 16-bit counter are compared and their values match.
- The compare clear register has a buffer register (the data written to this buffer register is transferred to the compare clear register). When the 16-bit counter is stopped, the data is transferred as soon as data is written to the buffer. When the 16-bit counter is operating, data is transferred from the buffer upon detecting the count value "0000H".
- The count value is reset to "0000H" when a hardware reset occurs; software clears the timer, or the value of the compare clear register and the count value match.

4.1. Configuration Diagram of the 16-bit Global Timer

Figure 4-1Figure 4-1 Configuration Diagram of 16-bit Global Timer

is a configuration diagram of the 16-bit Global Timer.

Figure 4-1 Configuration Diagram of 16-bit Global Timer



5. Explanation of the 16-bit Global Timer Operation

This section provides a summary of the operation of the 16-bit Global Timer.

Operation of the 16-bit Global Timer

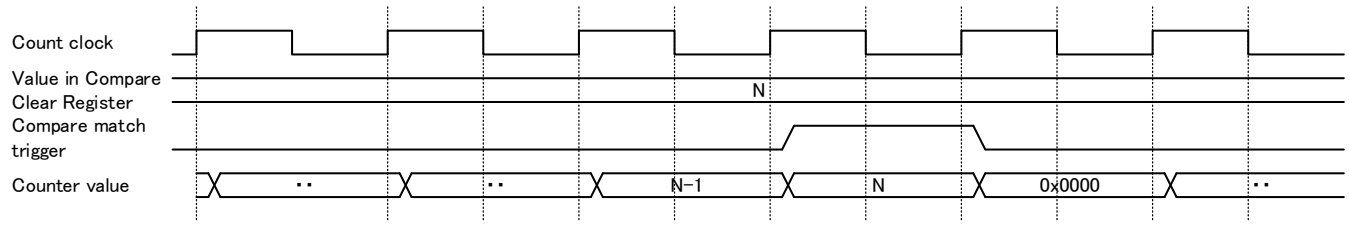
The 16-bit Global Timer starts counting from the value set in the Global Timer Data Register (GT_TCDT) after the timer is set to enabled (GT_TCCS:STOP).

Counter Clear

The count value of the Global Timer is cleared to 0 when any of the following conditions is met.

- When the count value matches the value of the Global Timer Compare Clear Register (GT_CPCLR) in the up count.
- When "0000H" is written to the Global Timer Data Register (GT_TCDT) while the Global Timer is stopped (STOP: bit6=1 of the Global Timer State Control Register (GT_TCCS))
- When the hardware is reset. Upon reset, the counter is cleared immediately.

Figure 5-1 Clear Timing of 16-bit Global Timer



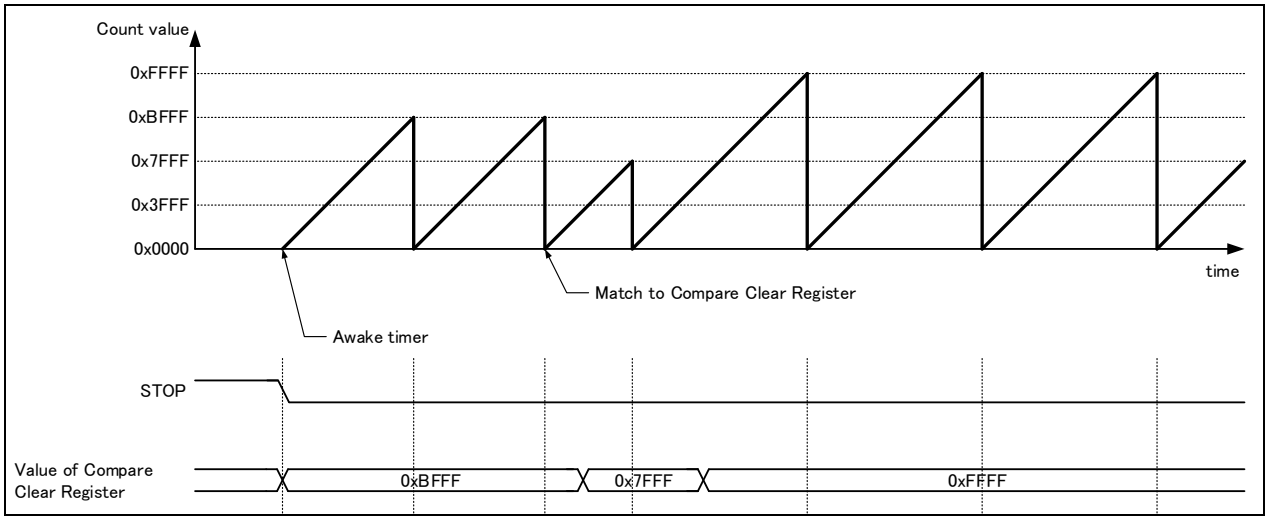
Note:

- *Count clock is the count operation clocks.*

Compare Clear

The data can be directly written to the Global Timer Compare Clear Register (GT_CPCLR).

Figure 5-2 Up Count Operation

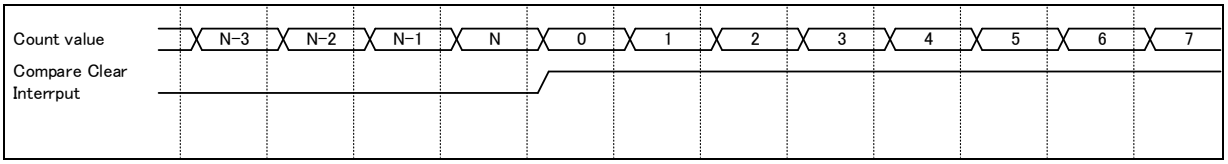


Timer Interrupt

The 16-bit Global Timer can generate the following interrupts.

- Compare clear interrupt

Figure 5-3 Interrupt Generated in Up Count



Interrupt Mask Function

The compare clear interrupt can be masked.

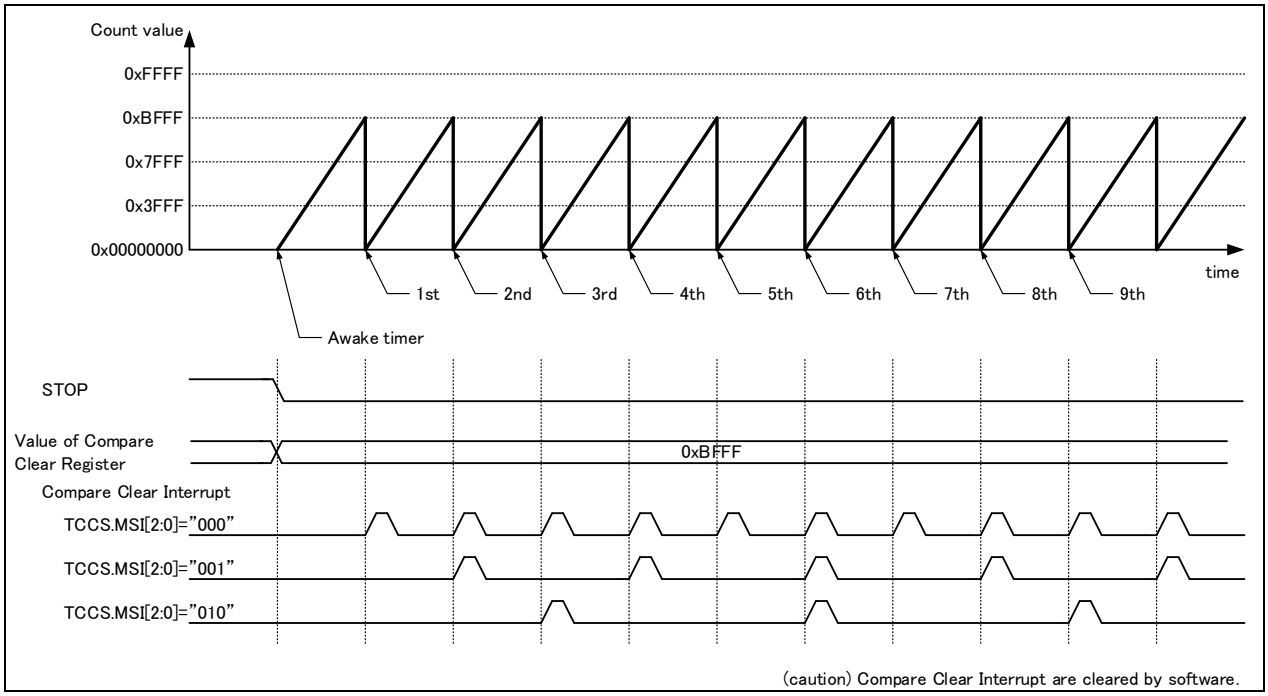
The following describes the case when interrupt is masked.

- The interrupt request flag can be masked by setting the interrupt mask selection bits (MSI2 to GT_MSI0: bit12 to bit10) of the Global Timer State Control Register (GT_TCCS). The interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10) are the 3-bit reload down register that reloads the value when the mask count value reaches "000B".
  The mask count value can also be loaded by directly writing data to the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10). The number of mask counts is the value set in the interrupt mask selection bits (MSI2 to MSI0: bit12 to bit10).
  The interrupt request flag is not masked when the mask count value (MSI2 to MSI0: bit12 to bit10) becomes "000B".

  Note:

  - *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 16-bit Global Timer stopped.*

Figure 5-4 Compare Clear Interrupts to Be Masked in Up Count



5.1. Interrupts of the 16-bit Global Timer

The 16-bit Global Timer has one type of interrupt. : compare interrupt

Global Timer interrupt

Table 5-1 shows the interrupt control bits and the interrupt factor of the Global Timer.

Table 5-1 Interrupt Control Bits and Interrupt Factor of Global Timer

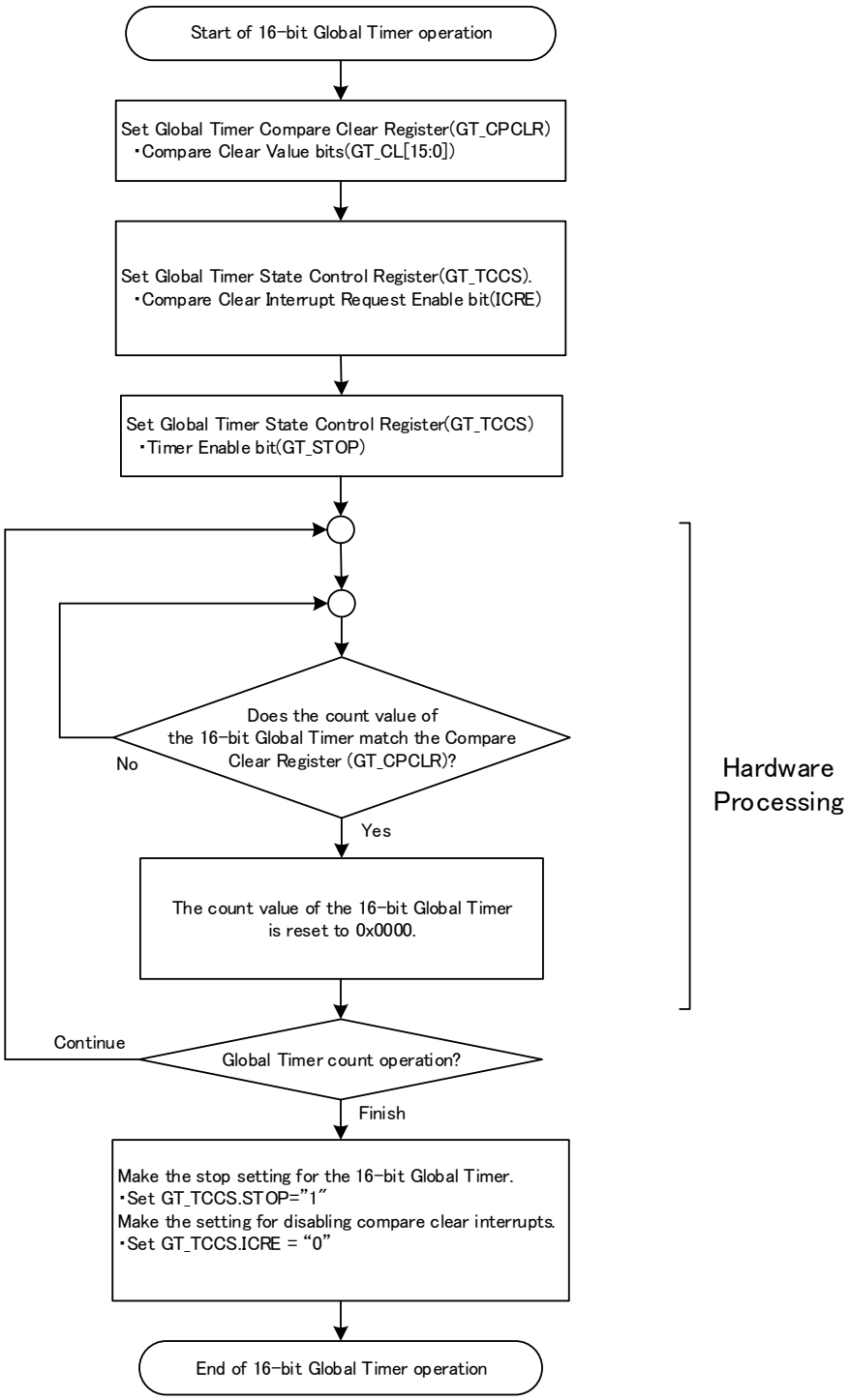
	Global Timer
	Compare Clear
Interrupt request flag bit	Compare clear interrupt flag bit (ICLR:bit9) of the Global Timer State Control Register (GT_TCCS)
Interrupt request enable bit	Compare clear interrupt request enable bit (ICRE:bit8) of the Global Timer State Control Register (GT_TCCS)
Interrupt Factor	The value of the Global Timer matches the value of the Global Timer Compare Clear Register (GT_CPCLR).

"1" is set to the compare clear interrupt flag (ICLR: bit9) of the Global Timer State Control Register (GT_TCCS) when the value of the Global Timer matches the value of the Global Timer Compare Clear Register (GT_CPCLR). When the interrupt request is set to enabled (ICRE: bit8=1 of the Global Timer State Control Register (GT_TCCS)) in this state, then the interrupt request signal (MIRQ) becomes "H".

5.2. Setting Procedure Example of the 16-bit Global Timer

This section provides a setting procedure example of the Global Timer.

Figure 5-5 Setting Procedure Example of Global Timer Operation



5.3. Debug Mode

If it goes into a debug mode, operation of the 16-bit Global Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the debug mode.

- The Debug Enable bit (DBGEN) in the Global Timer Debug Register (GT_DEBUG) is "1."
- The debug input (Debug input) is active.

It becomes the following operation when it goes into the debug mode.

- The prescaler stop. The state of prescaler is kept.
- The 16-bit Global Timer stop. Timer value and timer output value are kept.

It becomes the following operation when it restores from the debug mode.

- The prescaler is resumed from the stopped value.
- The 16-bit Global Timer is resumed from the stopped value.

Note:

- *Register access is possible in the debug mode.*

6. Registers of the 16-bit Global Timer

This section provides the register list of the 16-bit Global Timer.

Offset	Address offset value / Register name			
	+3	+2	+1	+0
0000_0000	GT_CPCLR 11111111 11111111 11111111 11111111			
0000_0004	GT_TCDT 11111111 11111111 00000000 00000000			
0000_0008	GT_TCCS 11111111 11111111 00000000 01000000			
0000_000C	-			
0000_0010	GT_TCCSC 11111111 11111111 00000000 00000000			
0000_0014	GT_TCCSS 11111111 11111111 00000000 00000000			
0000_0018	GT_DEBUG 11111111 11111111 00000000 00000000			
0000_001C	-			
0000_0020	GT_MSCTR 11111111 11111111 11111111 00000000			
0000_0024	GT_MSCHB0 00000000 00000000 00000000 00000000			
0000_0028	GT_MSCHB1 00000000 00000000 00000000 00000000			
0000_002C	GT_MSCH0 00000000 00000000 00000000 00000000			
0000_0030	GT_MSCH1 00000000 00000000 00000000 00000000			
0000_0034 | 0000_03FC	-			

Registers of the 16-bit Global Timer

Table 6-1 Register List of 16-bit Global Timer

Abbreviated Register Name	Register Name	Reference
GT_CPCLR	Global Timer Compare Clear Register	6.1
GT_TCDT	Global Timer Data Register	6.2
GT_TCCS	Global Timer State Control Register	6.3
GT_TCCSC	Global Timer State Control Clear Register	6.4
GT_TCCSS	Global Timer State Control Set Register	6.5
GT_DEBUG	Global Timer Debug Register	6.6
GT_MSCTR	Global Timer Match Starting Control Register	6.7
GT_MSCHB0~1	Global Timer Match Starting Channel Buffer Register	6.8
GT_MSCH0~1	Global Timer Match Starting Channel Register	6.9

Register Bit Locations of the 16-bit Global Timer

Table 6-2 Register Bit Locations of the 16-bit Global Timer

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
GT_CPCLR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
GT_TCDT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	T15	T14	T13	T12	T11	T10	T09	T08
	T07	T06	T05	T04	T03	T02	T01	T00
GT_TCCS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	MSI2	MSI1	MSI0	ICLR	ICRE
	Reserved	STOP	Reserved	Reserved	CLK3	CLK2	CLK1	CLK0
GT_TCCSC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICLRC	ICREC
	Reserved	STOPC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
GT_TCCSS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICRES
	Reserved	STOPS	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
GT_DEBUG	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DBGEN

GT_MSCTR	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	TREN	MODE	GTMOE
GT_MSCHB0	GTCH31	GTCH30	GTCH29	GTCH28	GTCH27	GTCH26	GTCH25	GTCH24
	GTCH23	GTCH22	GTCH21	GTCH20	GTCH19	GTCH18	GTCH17	GTCH16
	GTCH15	GTCH14	GTCH13	GTCH12	GTCH11	GTCH10	GTCH9	GTCH8
	GTCH7	GTCH6	GTCH5	GTCH4	GTCH3	GTCH2	GTCH1	GTCH0
GT_MSCHB1	GTCH63	GTCH62	GTCH61	GTCH60	GTCH59	GTCH58	GTCH57	GTCH56
	GTCH55	GTCH54	GTCH53	GTCH52	GTCH51	GTCH50	GTCH49	GTCH48
	GTCH47	GTCH46	GTCH45	GTCH44	GTCH43	GTCH42	GTCH41	GTCH40
	GTCH39	GTCH38	GTCH37	GTCH36	GTCH35	GTCH34	GTCH33	GTCH32
GT_MSCH0	GTCHR31	GTCHR30	GTCHR29	GTCHR28	GTCHR27	GTCHR26	GTCHR25	GTCHR24
	GTCHR23	GTCHR22	GTCHR21	GTCHR20	GTCHR19	GTCHR18	GTCHR17	GTCHR16
	GTCHR15	GTCHR14	GTCHR13	GTCHR12	GTCHR11	GTCHR10	GTCHR9	GTCHR8
	GTCHR7	GTCHR6	GTCHR5	GTCHR4	GTCHR3	GTCHR2	GTCHR1	GTCHR0
GT_MSCH1	GTCHR63	GTCHR62	GTCHR61	GTCHR60	GTCHR59	GTCHR58	GTCHR57	GTCHR56
	GTCHR55	GTCHR54	GTCHR53	GTCHR52	GTCHR51	GTCHR50	GTCHR49	GTCHR48
	GTCHR47	GTCHR46	GTCHR45	GTCHR44	GTCHR43	GTCHR42	GTCHR41	GTCHR40
	GTCHR39	GTCHR38	GTCHR37	GTCHR36	GTCHR35	GTCHR34	GTCHR33	GTCHR32

6.1. Global Timer Compare Clear Register (GT_CPCLR)

Compare Clear Register (GT_CPCLR)

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	1	1	1	1	1	1	1	1

[bit31:16] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit15:0] CL15 to CL00: Compare clear value bits

The compare clear register (GT_CPCLR) is used to compare the count value of the Global Timer.

The count value of the Global Timer is reset to "0000H" when the value of this register matches the count value of the Global Timer.

Notes:

- *Do not set “0x0000” in the Global Timer Compare Clear Register (GT_CPCLR).*
- *For access to this register, use half word access instructions.*

6.2. Global Timer Data Register (GT_TCDT)

The timer data register (GT_TCDT) reads the count value of the Global Timer. This register can also be used to set the count value of the Global Timer.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	T15	T14	T13	T12	T11	T10	T09	T08
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	T07	T06	T05	T04	T03	T02	T01	T00
R/W Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:16] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit15:0] T15 to T00: Global Timer data value bits

The timer data register (GT_TCDT) is used to read the count value of the Global Timer.

The count value can be set by writing a value to this register. However, a value needs to be written while the Global Timer is stopped (STOP: bit 6=1 of the timer state control register (GT_TCCS)).

The counter is cleared to the count value "0x0000" as soon as any of the following factors occurs.

- Hardware reset
- The value of the compare clear register (GT_CPCLR) matches.

Note:

- *For access to this register, use half word access instructions.*

6.3. Global Timer State Control Register (GT_TCCS)

The timer state control register (GT_TCCS) is a register that is used to control the operation of the Global Timer.

For details on writing to this register, see "7.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved			MSI2	MSI1	MSI0	ICLR	ICRE
R/W Attribute	R0/WX			R,W	R,W	R,W	R	R/W
Protection Attribute	-			-	-	-	-	-
Initial Value	000			0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOP	Reserved		CLK3	CLK2	CLK1	CLK0
R/W Attribute	R0/WX	R/W	R0/WX		R/W	R/W	R/W	R/W
Protection Attribute	-	-	-		-	-	-	-
Initial Value	0	1	00		0	0	0	0

[bit31:16] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit15:13] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit12:10] MSI2 to MSI0: Interrupt mask selection bits

- These bits are used to set the number of times to mask the compare clear interrupt flag.
- When these bits are set to "000B", the interrupt flags are not masked.

Bit			Description
MSI2	MSI1	MSI0	
0	0	0	Generate an interrupt flag at the first match occurrence.
0	0	1	Generate an interrupt flag at the second match occurrence.
0	1	0	Generate an interrupt flag at the third match occurrence.
0	1	1	Generate an interrupt flag at the fourth match occurrence.
1	0	0	Generate an interrupt flag at the fifth match occurrence.
1	0	1	Generate an interrupt flag at the sixth match occurrence.
1	1	0	Generate an interrupt flag at the seventh match occurrence.
1	1	1	Generate an interrupt flag at the eighth match occurrence.

Note:

The value read is a mask counter value. The mask counter is a decrement counter.

- *The written data is written to the mask register.*
- *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 16-bit Global Timer stopped.*

[bit9] ICLR: Compare clear interrupt flag bit

This bit is set to "1" when the value of the Global Timer Compare Clear Registers (GT_CPCLR) and the value of the Global Timer match.

This bit is a read-only bit. Writing data to these bits has no effect on the operation.

This bit is cleared by writing "1" to the ICLRC bit of the Global Timer State Control Clear Register (GT_TCCSC).

bit	Description
0	No compare clear match.
1	Compare clear match.

Note:

- *This bit is set to "1" when the interrupt flag set by the interrupt mask selection bits (MSI2 to MSI0) occurs. This bit is not set to "1" when no interrupt occurs.*

[bit8] ICRE: Compare clear interrupt request enable bit

When this bit is set to "1" and the compare clear interrupt flag bit (ICLR: bit9) is set to "1", an interrupt request to the CPU is generated.

This bit is cleared to "0" by writing "1" to the ICREC bit of the Global Timer State Control Clear Register (GT_TCCSC).
This bit is set to "1" by writing "1" to the ICRES bit of the Global Timer State Control Set Register (GT_TCCSS).

bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit7] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOP: Timer enable bit

This bit is used to stop/start counting of the Global Timer.
When this bit is set to "0".
The Global Timer starts counting.
When this bit is set to "1".
The Global Timer stops counting.

This bit is cleared to "0" by writing "1" to the STOPC bit of the Global Timer State Control Clear Register (GT_TCCSC).
This bit is set to "1" by writing "1" to the STOPS bit of the Global Timer State Control Set Register (GT_TCCSS).

bit	Description
0	Enable counting (start counting).
1	Disable counting (stop counting).

Note:

- *After counting of the Global Timer is stopped , the following initialization setting is necessary.*
  1. Initializes of the Global Timer count value.

     It writes "0000_0000h" to the Global Timer Data Register(GT_TCDT)[15:0].
  2. Disable of an ETEN(external timer compare enabling signal) signal.

     It sets Global Timer Match Output Enable bit(GTMODE) in Global Timer Match Starting Control Register(GT_MSCTR) to “0”

[bit5:4] Reserved bits

"0" is always read from this bit..
Writing data to these bits has no effect on the operation.

[bit3:0] CLK3 to CLK0: Clock frequency selection bits

These bits are used to select the count clock frequency of the Global Timer.
The clock frequency is changed immediately upon setting these bits.

bit				Description					
CLK3	CLK2	CLK1	CLK0	Count Clock	ϕ = 40 MHz	ϕ = 20 MHz	ϕ = 10 MHz	ϕ = 5 MHz	ϕ = 2.5 MHz
0	0	0	0	Φ	25 ns	50 ns	100 ns	200 ns	400 ns
0	0	0	1	Φ/2	50 ns	100 ns	200 ns	400 ns	800 ns
0	0	1	0	Φ/4	100 ns	200 ns	400 ns	800 ns	1.6 us
0	0	1	1	Φ/8	200 ns	400 ns	800 ns	1.6 us	3.2 us
0	1	0	0	Φ/16	400 ns	800 ns	1.6 us	3.2 us	6.4 us
0	1	0	1	Φ/32	800 ns	1.6 us	3.2 us	6.4 us	12.8 us
0	1	1	0	Φ/64	1.6 us	3.2 us	6.4 us	12.8 us	25.6 us
0	1	1	1	Φ/128	3.2 us	6.4 us	12.8 us	25.6 us	51.2 us
1	0	0	0	Φ/256	6.4 us	12.8 us	25.6 us	51.2 us	102.4 us
1	0	0	1	Φ/512	12.8 us	25.6 us	51.2 us	102.4 us	204.8 us
1	0	1	0	Φ/1024	25.6 us	51.2 us	102.4 us	204.8 us	409.6 us
1	0	1	1	Φ/2048	51.2 us	102.4 us	204.8 us	409.6 us	819.2 us
Other settings are prohibited.				-	-	-	-	-	-

ϕ : Peripheral clock: CLK_LCP1A

6.4. Global Timer State Control Clear register (GT_TCCSC)

The Global Timer State Control Clear Register (GT_TCCSC) is a register that is used to clear bits of the Global Timer State Control Register (GT_TCCS).

For details on writing to this register, see "7.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved						ICLRC	ICREC
R/W Attribute	R0,WX						R0,W	R0,W
Protection Attribute	-						-	-
Initial Value	000000						0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOPC	Reserved					
R/W Attribute	R0,WX	R0,W	R0,WX					
Protection Attribute	-	-	-					
Initial Value	0	0	00000					

[bit31:16] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit15:10] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit9] ICLRC: ICLR clear bit

When "1" is written to this bit, the compare clear interrupt flag bit (ICLR) of the Global Timer State Control Register (GT_TCCS) is cleared. "0" is always read from this bit.

bit	Description
0	Affect neither this bit nor the compare clear interrupt flag bit (ICLR) of the Global Timer State Control Register (GT_TCCS).
1	Clear the compare clear interrupt flag bit (ICLR) of the Global Timer State Control Register (GT_TCCS).

[bit8] ICREC: ICRE clear bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS) is cleared. "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).
1	Clear the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).

[bit7] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOPC: STOP clear bit

When "1" is written to this bit, the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS) is cleared. "0" is always read from this bit.

bit	Description
0	Affect neither this bit nor the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).
1	Clear the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).

[bit5:0] Reserved bits

"0" is always read from this bit. Always write "0" to this bit.

Writing data to these bits has no effect on the operation.

6.5. Global Timer State Control Set register (GT_TCCSS)

The Global Timer State Control Set Register (GT_TCCSS) is a register that is used to set bits of the Global Timer State Control Register (GT_TCCS).

For details on writing to this register, see "7.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved							ICRES
R/W Attribute	R0,WX							R0,W
Protection Attribute	-							-
Initial Value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved	STOPS	Reserved					
R/W Attribute	R0,WX	R0,W	R0,WX					
Protection Attribute	-	-	-					
Initial Value	0	0	000000					

[bit31:16] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit15:9] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit8] ICRES: ICRE set bit

When "1" is written to this bit, the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).
1	Set the compare clear interrupt request enable bit (ICRE) of the Global Timer State Control Register (GT_TCCS).

[bit7] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit6] STOPS: STOP set bit

When "1" is written to this bit, the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS) is set to "1". "0" is always read from this bit.

Bit	Description
0	Affect neither this bit nor the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).
1	Set the timer enable bit (STOP) of the Global Timer State Control Register (GT_TCCS).

[bit5:0] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

6.6. Global Timer Debug Register (GT_DEBUG)

The Global Timer Debug Register (GT_DEBUG) performs enable/disable setting of debug.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R1,WX							
PROT_TYPE	-							
INITIAL_VALUE	11111111							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R1,WX							
PROT_TYPE	-							
INITIAL_VALUE	11111111							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE	-							
INITIAL_VALUE	00000000							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved							DBGEN
ACCESS_TYPE	R0,W0							R/W
PROT_TYPE	-							
INITIAL_VALUE	0000000							0

[bit31:16] Reserved bits

The read value is "1".

If writing to this bit, write "1".

[bit15:1] Reserved bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable bit

It is the bit which permits stopping operation of a 16-bit Global Timer with the debug input (Debug input).

bit	Description
0	Disable
1	Enable

6.7. Global Timer Match Starting Control Register (GT_MSCTR)

This Global Timer Match Start Control Register controls the Global Timer Match Start Enable Signal which it inputs into the Base Timer.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	Reserved					TREN	MODE	GTMOE
R/W Attribute	R0,WX					R/W	R/W	R/W
Protection Attribute	-							
Initial Value	00000					0	0	0

[bit31:8] Reserved bits

"1" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit7:3] Reserved bits

"0" is always read from this bit.
Writing data to these bits has no effect on the operation.

[bit2] TREN: Transfer Request Enable bit

It uses this bit in order to enable a transmission request of the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) value.

When this bit is "0", even if the count value of the Global Timer is match with a comparison clear register (GT_CPCLR), the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) value is not transmitted.

When this bit is "1", even if the count value of the Global Timer is match with a comparison clear register (GT_CPCLR), the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) value is transmitted to the Global Timer Match Starting Channel Register (GT_MSCH0~1).

Bit	Description
0	Disable the transfer request of Register data
1	Enable the transfer request of Register data

Note:

- *When changing the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) set, it sets this bit to "0."*

[bit1] MODE: Mode Select bit

It uses this bit in order to enable the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1).
The Global Timer match starting enabling signal of each Base Timer channel is controlled by this bit setup as follows.

- When the is MODE ="0"
  - The Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) is invalid.
  - After writing "1" in GTMOE bit, it becomes external timer comparison enabling signal to the Base Timer channel set as 1 (ETEN)" H" by a GT_MSCH0~1 register at once.
- When the is MODE ="1"
  - The Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) is valid.
  - The data which was written in the Global Timer Match Start Channel Buffer Register (GT_MSCHB0~1) and was held, After writing "1" in TREN bit, if the Global Timer counting value and the Global Time Compare Clear Register (GT_CPCLR) matches, it is transmitted to the Global Timer Match Starting Channel Register (GT_MSCH0~1), the External Timer Compare Enable Signal (ETEN) to the Base Timer Channel set as “1” in the Global Timer Match Start Channel Register (GT_MSCH0~1) becomes "H".

Bit	Description
0	Buffer Disable
1	Buffer Enable

[bit0] GTMOE: Global Timer Match Output Enable bit

It uses this as an output Enable of the External Timer Compare Enable Signal (ETEN) of Base Timers.

When this bit is "0", the External Timer Compare Enable Signal (ETEN) to the channel of all Base Timers is set to "L.".

- It outputs "H" as an external timer compare enabling signal (ETEN) at the time of GTCH="1."
- It outputs "L" as an external timer compare enabling signal (ETEN) at the time of GTCH="0.".

Bit	Description
0	Disable the output of all External Timer Compare Enable Signals.
1	Enable the output of all External Timer Compare Enable Signals.

6.8. Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1)

The Global Timer Match Startup Channel Buffer Register (GT_MSCHB0~1) is a buffer register of the Global Timer Match Startup Channel Register (GT_MSCH0~1)

There is the undermentioned two timing by which this register value is transmitted to the Global Timer Match Startup Channel Register (GT_MSCH0~1).

- The Global Timer is stopping (STOP ="1" of a Global Timer Status Control Register (GT_TCCS)), or it is at the disabled (MODE ="0" of Global Timer Match Starting Control Register (GT_MSCTR)) time about the STOP = "1" buffer function of the Global Timer Status Control Register (GT_TCCS), the value of a Global Timer Match Start Channel Buffer Register (GT_MSCHB0~1) is transmitted to a Global Timer Match Start Channel Register (GT_MSCH0~1) at once.
- The buffer function is at the enable (MODE ="1" of Global Timer Match Starting Control Register (GT_MSCTR)) time, it sets a transmission request of a register value as enable (TREN bit ="1" of Global Timer Match Starting Control Register (GT_MSCTR)), and when the count value of the Global Timer matches the value of the Global Timer Compare Clear Register(GT_CPCLR), the value of a Global Timer Match Start Channel Buffer Register (GT_MSCHB0~1) is transmitted to the Global Timer Match Start Channel Register (GT_MSCH0~1).

Global Timer Match Starting Channel Buffer registers (GT_MSCHB0)

Bit	31	30	29	28	27	26	25	24
Field	GTCH31	GTCH30	GTCH29	GTCH28	GTCH27	GTCH26	GTCH25	GTCH24
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	GTCH23	GTCH22	GTCH21	GTCH20	GTCH19	GTCH18	GTCH17	GTCH16
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	GTCH15	GTCH14	GTCH13	GTCH12	GTCH11	GTCH10	GTCH9	GTCH8
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GTCH7	GTCH6	GTCH5	GTCH4	GTCH3	GTCH2	GTCH1	GTCH0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] GTCH31~0 : Global Timer Match Starting Channel Buffer bits

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Global Timer Match Starting Channel Buffer registers (GT_MSCHB1)

bit	31	30	29	28	27	26	25	24
Field	GTCH63	GTCH62	GTCH61	GTCH60	GTCH59	GTCH58	GTCH57	GTCH56
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	GTCH55	GTCH54	GTCH53	GTCH52	GTCH51	GTCH50	GTCH49	GTCH48
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	GTCH47	GTCH46	GTCH45	GTCH44	GTCH43	GTCH42	GTCH41	GTCH40
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GTCH39	GTCH38	GTCH37	GTCH36	GTCH35	GTCH34	GTCH33	GTCH32
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] GTCH63~32 : Global Timer Match Starting Channel Buffer bits

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Notes:

- *At the time of the Global Timer stop, it is after the Global Timer Match Start Channel Buffer Register (GT_MSCHB0~1) set, please permit the External Timer Compare Enabling-Signal Output (GTMOE ="1" of Global Timer Match Starting Control Register (GT_MSCTR)).*
- *At the case of TREN = "1" (transmission request Clearance) of Global Timer Match Starting Control register (GT_MSCTR), please do not change the Global Timer Match Start Channel Buffer-Register (GT_MSCHB0~1) set.*
- *For access to this register, use word access instructions.*

6.9. Global Timer Match Starting Channel Register (GT_MSCH0~1)

This register displays the value transmitted from the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1) register.

Global Timer Match Starting Channel registers (GT_MSCH0)

bit	31	30	29	28	27	26	25	24
Field	GTCHR31	GTCHR30	GTCHR29	GTCHR28	GTCHR27	GTCHR26	GTCHR25	GTCHR24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	GTCHR23	GTCHR22	GTCHR21	GTCHR20	GTCHR19	GTCHR18	GTCHR17	GTCHR16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	GTCHR15	GTCHR14	GTCHR13	GTCHR12	GTCHR11	GTCHR10	GTCHR9	GTCHR8
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GTCHR7	GTCHR6	GTCHR5	GTCHR4	GTCHR3	GTCHR2	GTCHR1	GTCHR0
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] GTCHR31~0 : Global Timer Match Starting Channel bits

This bit becomes external timer compare enabling-signal to the Base Timer channel which the Global Timer match start is permitted and corresponds when it is 1" (ETEN)" H".

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Global Timer Match Starting Channel registers (GT_MSCH1)

bit	31	30	29	28	27	26	25	24
Field	GTCHR63	GTCHR62	GTCHR61	GTCHR60	GTCHR59	GTCHR58	GTCHR57	GTCHR56
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	GTCHR55	GTCHR54	GTCHR53	GTCHR52	GTCHR51	GTCHR50	GTCHR49	GTCHR48
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	GTCHR47	GTCHR46	GTCHR45	GTCHR44	GTCHR43	GTCHR42	GTCHR41	GTCHR40
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GTCHR39	GTCHR38	GTCHR37	GTCHR36	GTCHR35	GTCHR34	GTCHR33	GTCHR32
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:0] GTCHR63~32 : Global Timer Match Starting Channel bits

This bit becomes external timer compare enabling-signal to the Base Timer channel which the Global Timer match start is permitted and corresponds when it is 1" (ETEN)" H".

Bit	Description
0	Global Timer Match Starting Disabled
1	Global Timer Match Starting Enabled

Note:

- *For access to this register, use word access instructions.*

7. Precautions for Using This Device

The following shows the notes when using the 16-bit Global Timer.

Notes to observe when accessing a register

When Accessing the Compare Clear Register (GT_CPCLR)

Use half word access instructions to the Global Timer Compare Clear Register (GT_CPCLR)

When Accessing the Timer State Control Registers (GT_TCCS)

- This register supports writing from the bit-band alias area. For the bit-band alias area, see the chapter of "Bit-Band Unit" in Traveo™ Platform hardware manual.
- To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit of the Global Timer State Control Clear Register (GT_TCCSC).
- To set a specified bit of this register, set the bit by writing "1" to the applicable bit of the Global Timer State Control Set Register (GT_TCCSS).
- Data can be written directly to this register only when writing to all bits.
- In the normal reading mode, the interrupt mask counter value is read from MSI2 to MSI0.
- If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 16-bit Global Timer stopped.

When Accessing the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1)

Use word access instructions to the Global Timer Match Starting Channel Buffer Register (GT_MSCHB0~1)

When Accessing the Global Timer Match Starting Channel Register (GT_MSCH0~1)

Use word access instructions to the Global Timer Match Starting Channel Register (GT_MSCH0~1)

Notes when operating the Global Timer

When Setting Using a Program

- The compare clear flag is not set if the timer starts counting when the value of the Global Timer Compare Clear Register (GT_CPCLR) matches the count value.
- Set any values other than 0 to the Global Timer Compare Clear Register (GT_CPCLR). Note that the following operations occur if 0 is set.
  - When the count value is updated to 0 and fixed to 0. Then, the compare clear flag are set at every count clock.

CHAPTER 19: 32-Bit Free-Run Timer

This chapter describes the functions of the 32-bit Free-Run Timer.

1. Overview of the 32-bit Free-run Timer
2. Block Diagram of the 32-bit Free-run Timer
3. Operation of the 32-bit Free-run Timer
4. Registers of the 32-bit Free-run Timer
5. Precautions for Using This Device

FRT-JUPI-E3

1. Overview of the 32-Bit Free-Run Timer

The 32-bit Free-run Timer supports the 32-bit up count mode or up/down count mode. This timer can be used with the 32-bit Input Capture and the 32-bit Output Compare.

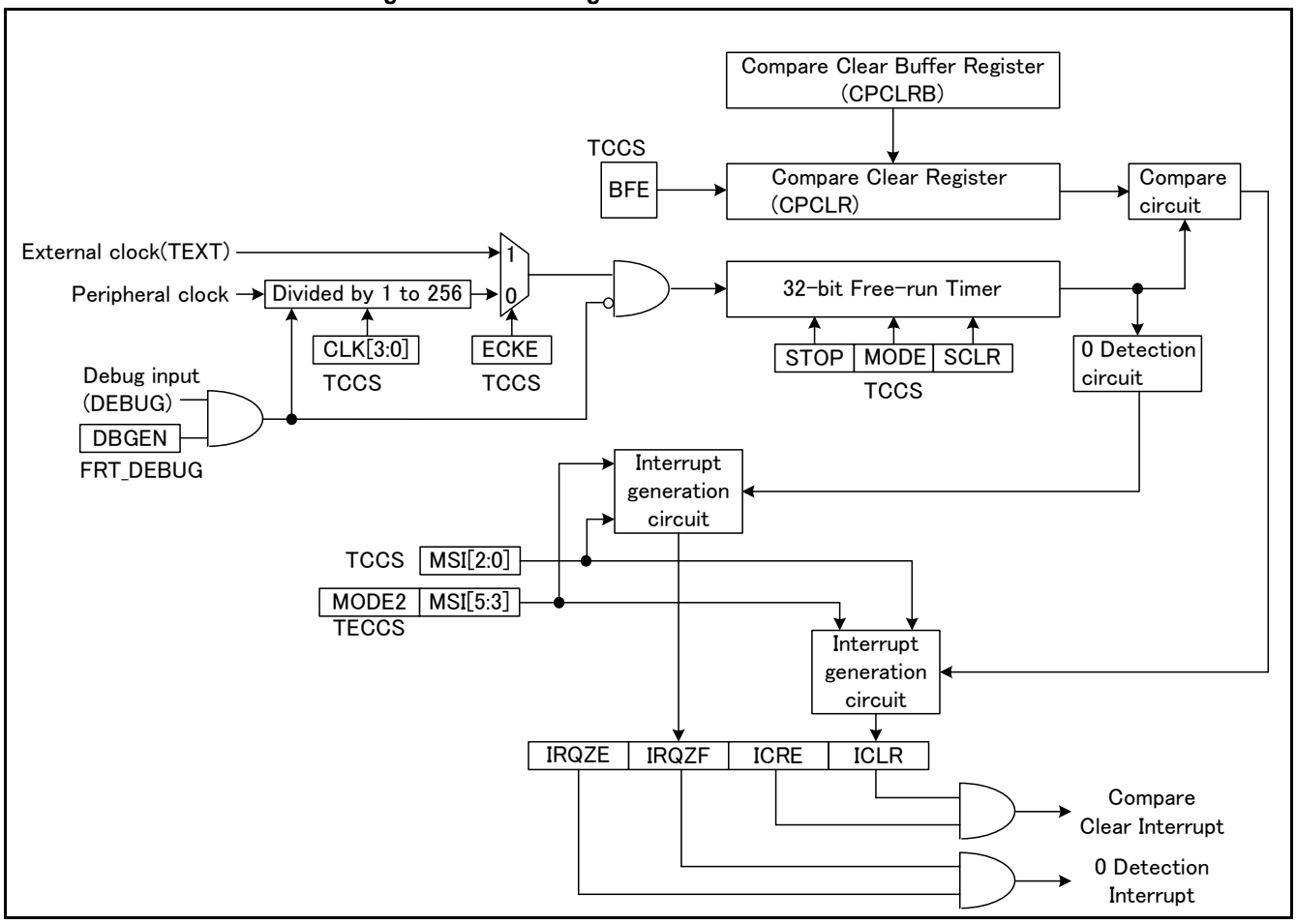
Functions of the 32-Bit Free-Run Timer

- The 32-bit Free-run Timer is composed of the 32-bit up/down counter, Timer State Control Register, 32-bit Compare Clear Register, 32-bit Compare Clear Buffer Register, and Prescaler.
- The 9 types of count clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, and ϕ/256) (ϕ: peripheral clock) can be selected.
- A Compare Clear Interrupt is generated when the values of the Compare Clear Register and the 32-bit counter are compared and their values match. A 0 detection interrupt is generated when the 32-bit counter detects the count value 0x00000000.
- The Compare Clear Register has a buffer register (the data written to this buffer register is transferred to the Compare Clear Register). When the 32-bit counter is stopped, the data is transferred as soon as data is written to the buffer. When the 32-bit counter is operating, data is transferred from the buffer upon detecting the count value 0x00000000.
- The count value is reset to 0x00000000 when a hardware reset occurs, clears the timer, or the value of the Compare Clear Register and the count value match in up count mode.
- The output value of the 32-bit Free-run Timer can be used as a clock count of the 32-bit Output Compare and the 32-bit Input Capture.

2. Block Diagram of the 32-Bit Free-Run Timer

This section shows a block diagram of 32-bit Free-run Timer

Figure 2-1 Block Diagram of 32-Bit Free-Run Timer



3. Operation of the 32-Bit Free-Run Timer

This section describes the operation of the 32-bit Free-run Timer.

Operation of the 32-Bit Free-Run Timer

The 32-bit Free-run Timer starts counting from the value set in the Timer Data Register (TCDT) after the timer is set to enabled (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS)). When the 32-bit Output Compare and the 32-bit Input Capture are connected to the 32-bit Free-run Timer, then the timer count value is used as base time of the 32-bit Output Compare and the 32-bit Input Capture.

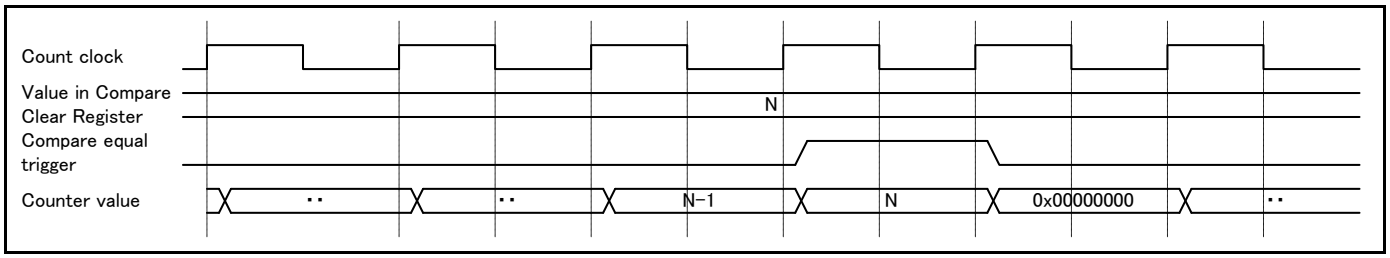
Counter Clear

The count value of the 32-bit Free-run Timer is cleared to 0x00000000 when any of the following conditions is met.

- When the count value matches the value in the Compare Clear Register (CPCLR) in the up count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0")
- When "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) while the 32-bit Free-run Timer is operating (Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="0")
- When 0x00000000 is written to the Timer Data Register (TCDT) while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1")
- When the hardware is reset. Upon reset, the counter is cleared immediately.

When "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) or when the count value matches the value in the Compare Clear Register (CPCLR), the counter is cleared synchronously with the count timing.

Figure 3-1 Clear Timing of 32-Bit Free-Run Timer



Note:

- *The count value of the 32-bit Free-run Timer is not cleared if "1" is written to the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) while the timer is stopped.*

Timer Mode

Either of the following modes can be selected for the 32-bit Free-run Timer.

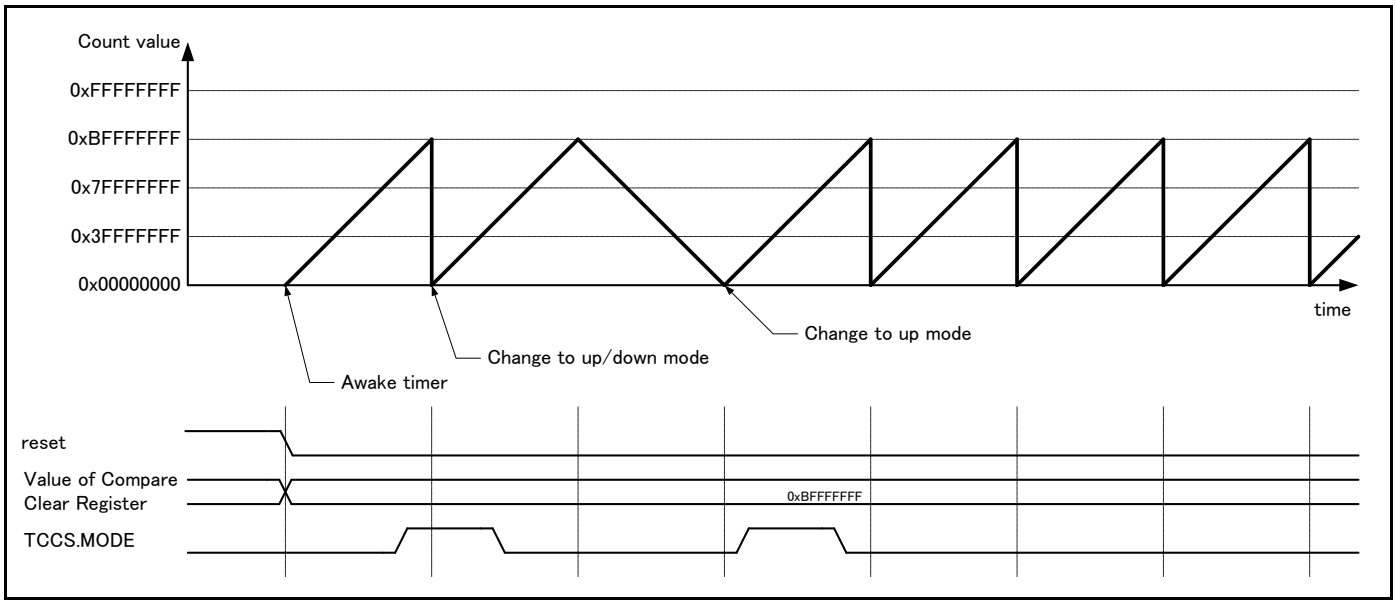
- Up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0")
- Up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1")

In the up count mode, the counter starts counting from the Timer Data Register (TCDT) that is set in advance. The counter continues to count up until the count value matches the value in the Compare Clear Register (CPCLR). Then, the counter is cleared to 0x00000000 and starts counting up again.

In the up/down count mode, the counter starts counting from the Timer Data Register (TCDT) that is set in advance. The counter continues to count up until the count value matches the value in the Compare Clear Register (CPCLR). Then, the counter changes the counting direction from up count to down count. It continues to count down until the counter value reaches 0x00000000 and starts counting up again.

A value can be written to the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) at any time, even when the timer is operating or stopped. The value written to this bit during the timer operation is stored in a buffer. The count mode changes when the count value becomes 0x00000000.

Figure 3-2 Change of Timer Mode (While Timer Is Operating)



Compare Clear Buffer

The Compare Clear Register (CPCLR) has a buffer function that can be enabled or disabled. When the buffer function is enabled (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="1"), the data written to the Compare Clear Buffer Register (CPCLRB) is transferred to the Compare Clear Register (CPCLR) upon detecting the count value 0x00000000. When the buffer function is disabled (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="0"), the data can be directly written to the Compare Clear Register (CPCLR).

Figure 3-3 Up Count Mode Operation When Compare Clear Buffer Is Disabled (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="0")

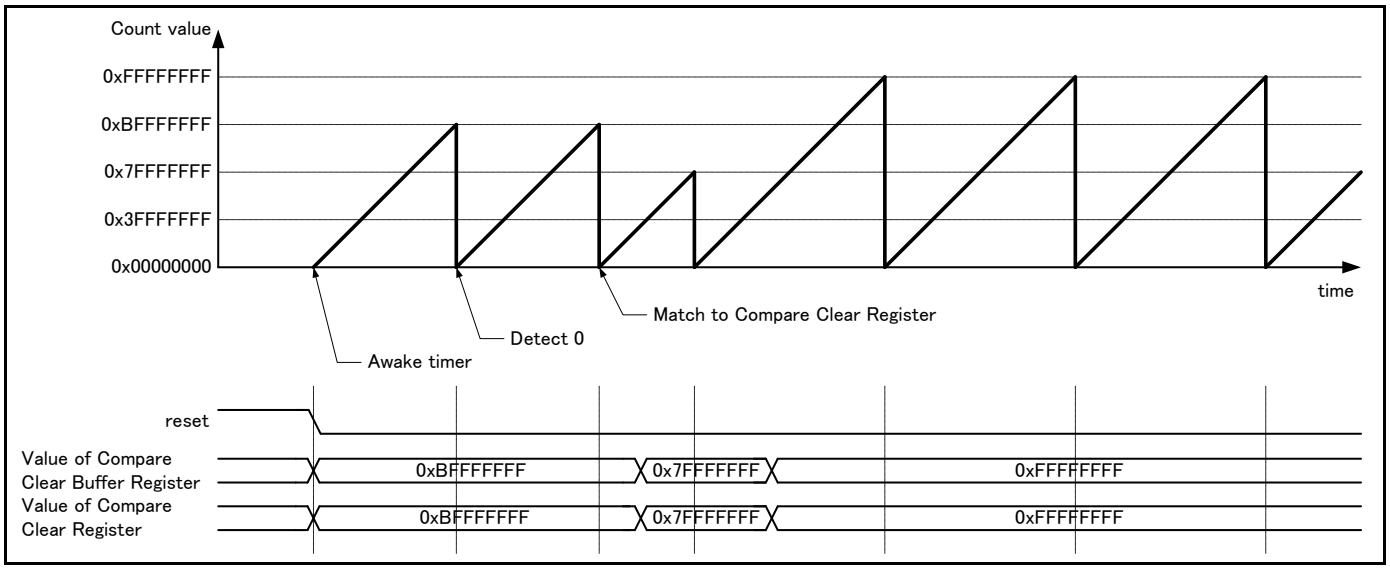


Figure 3-4 Up Count Mode Operation When Compare Clear Buffer Is Enabled (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="1")

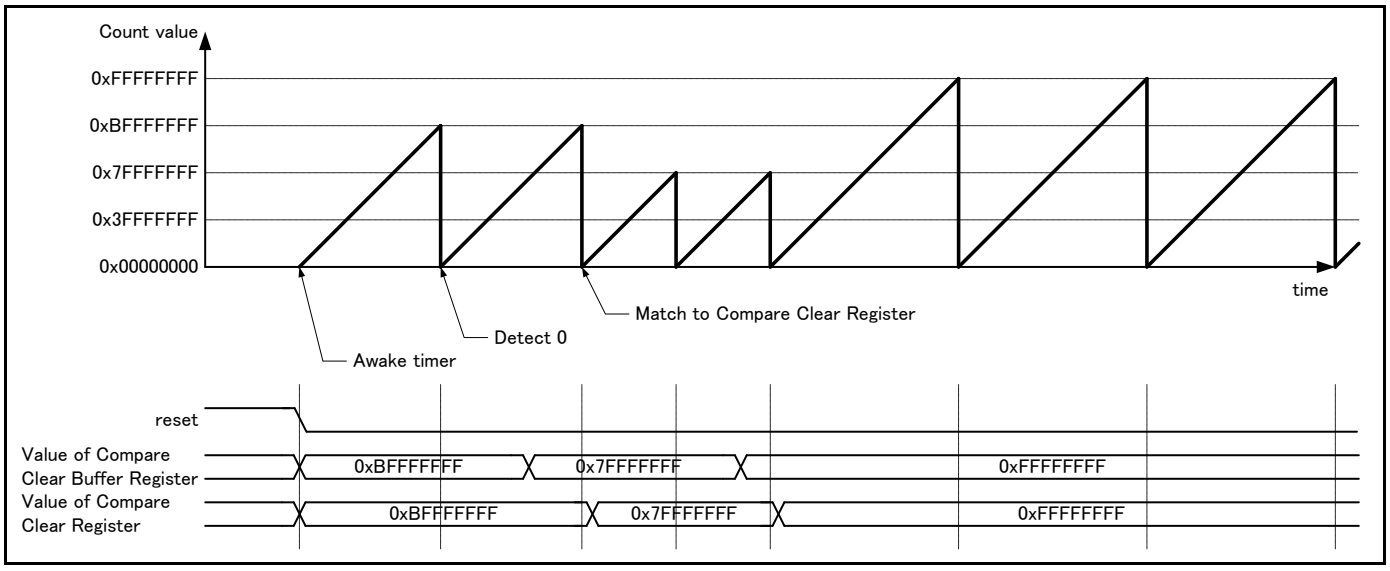
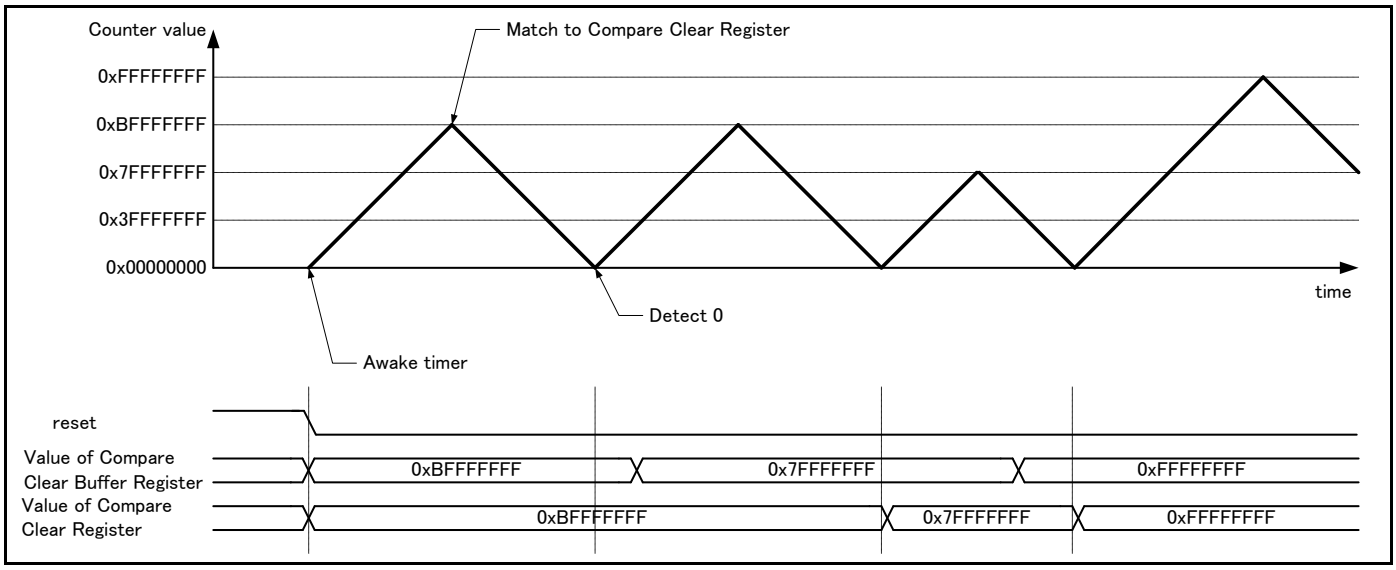


Figure 3-5 Up/Down Count Mode Operation When Compare Clear Buffer Is Enabled (the Compare Clear Buffer Enable Bit(BFE) of Timer State Control Register (TCCS) ="1")



Timer Interrupt

The 32-bit Free-run Timer can generate the following 2 interrupts.

- Compare Clear Interrupt
- 0 Detection Interrupt

The Compare Clear Interrupt is generated when the count value matches the value in the Compare Clear Register (CPCLR). The 0 detection interrupt is generated when the count value reaches 0x00000000.

For Zero Detection and Compare Clear Detection, detection begins when counting is enabled on the 32-bit Free-run Timer and the count value is updated from the current value to the next value.

Note:

- *The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1").*

Figure 3-6 Interrupt Generated in Up Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="0")

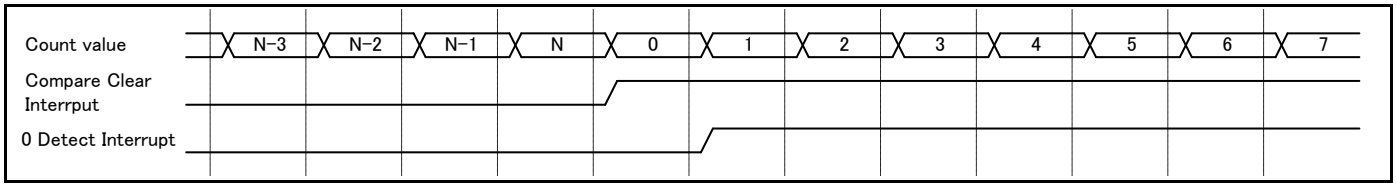
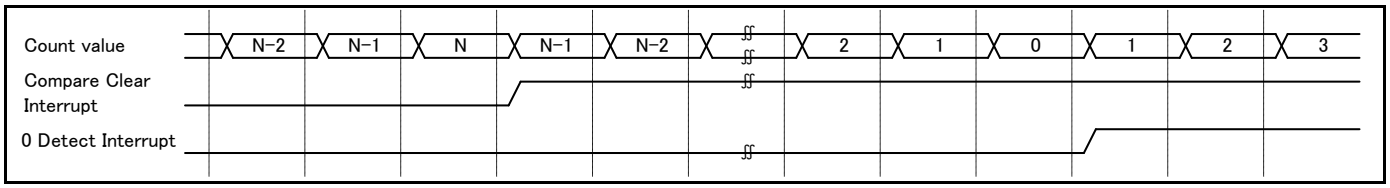


Figure 3-7 Interrupt Generated in Up/Down Count Mode (the Timer Count Mode Bit(MODE) in the Timer State Control Register (TCCS) ="1")



Interrupt Mask Function

Either the 0 detection interrupt or the Compare Clear Interrupt, or both interrupts can be masked.

The following describes the case when either interrupt is masked.

- An interrupt request flag can be masked by setting the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS). The Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS) are the 3-bit reload down register that reloads the value when the mask count value reaches "000".
  The mask count value can also be loaded by directly writing data to the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS). The number of mask counts is the value set in the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS). An interrupt request flag is not masked when the mask count value (setting in the Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS)) becomes "000".
- The mask control of an interrupt request varies depending on the count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)). In the up count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), only the Compare Clear Interrupt Request Flag can be masked and the 0 detection interrupt is generated every time a timer counter value of 0 is detected. In the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), only the 0 Detection Interrupt Flag(IRQZF) in the Timer State Control Register (TCCS) can be masked.

The following describes the case when both interrupt requests are masked.

- Both interrupts can be masked only when the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1") and when the Timer Extended State Control Register (TECCS) is set to the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="1" and the Timer State Control Register (TCCS) is set to the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1".
- Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS) are used to mask the 0 detection interrupt. Compare Clear Interrupt Mask Selection bits(MSI[5:3]) in the Timer Extended State Control Register (TECCS) are used to mask the Compare Clear Interrupt.

  Notes:

  - *The 0 detection interrupt is not generated when the timer is cleared (the Timer Clear bit(SCLR) in the Timer State Control Register (TCCS) ="1").*
  - *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.*

Figure 3-8 Compare Clear Interrupts to Be Masked in Up Count Mode

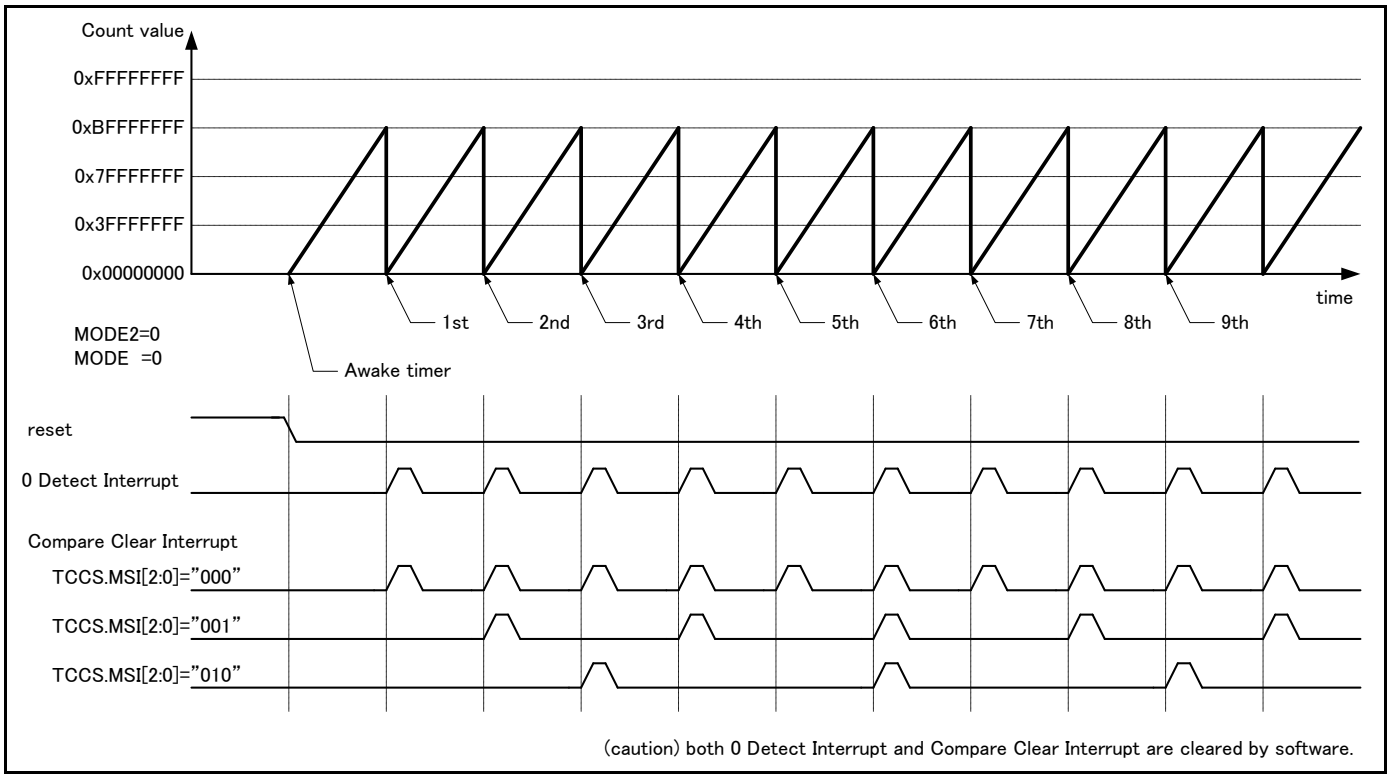


Figure 3-9 0 Detection Interrupts to Be Masked in Up/Down Count Mode

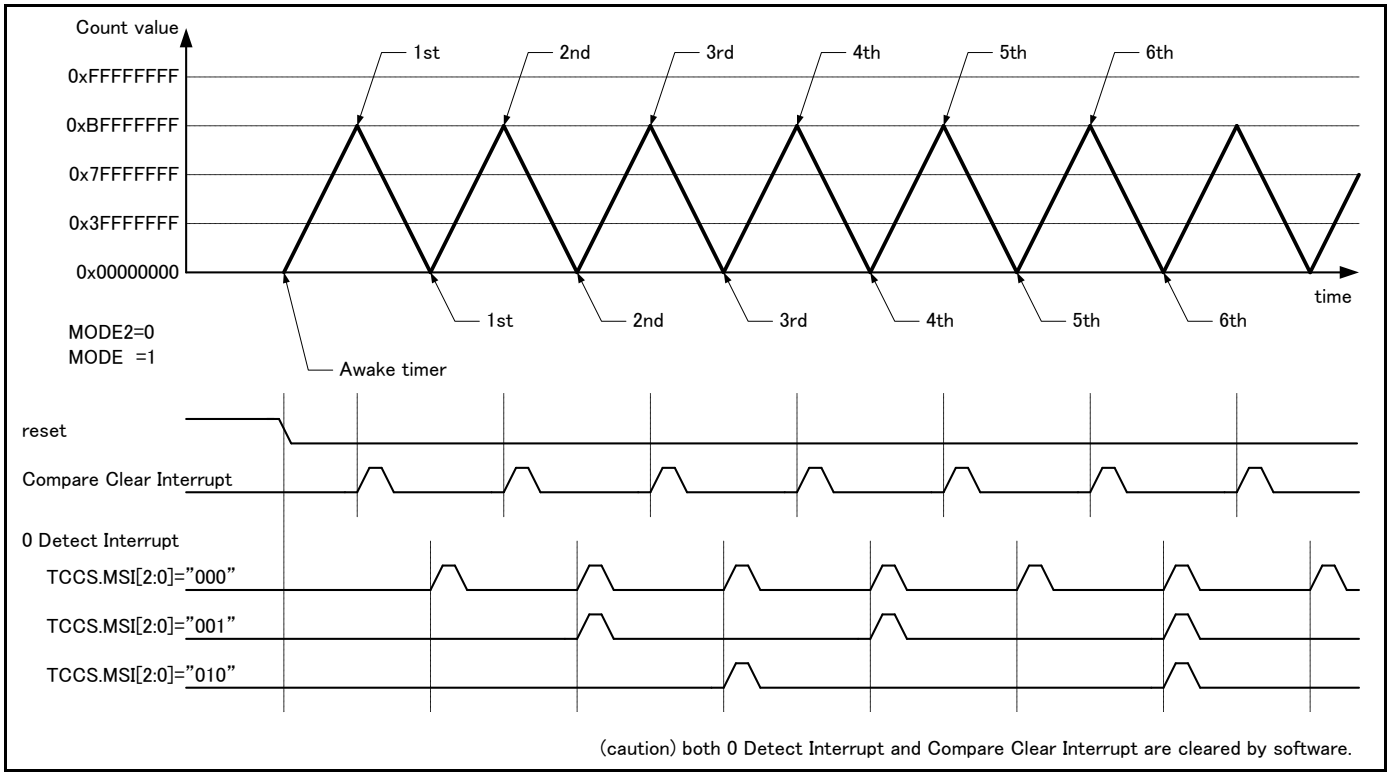
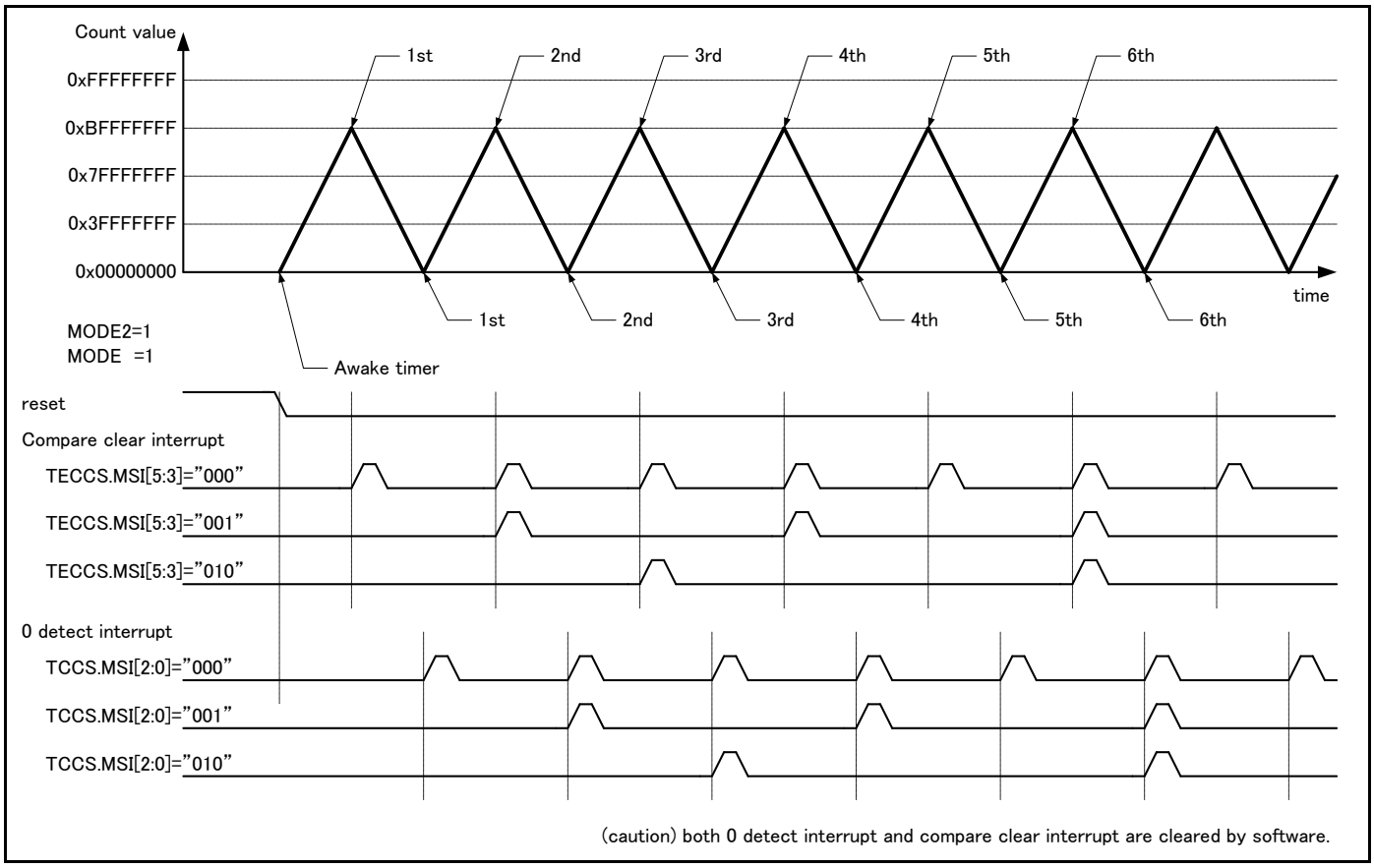


Figure 3-10 0 Detection Interrupts and Compare Clear Interrupts to Be Masked in Up/Down Count Mode

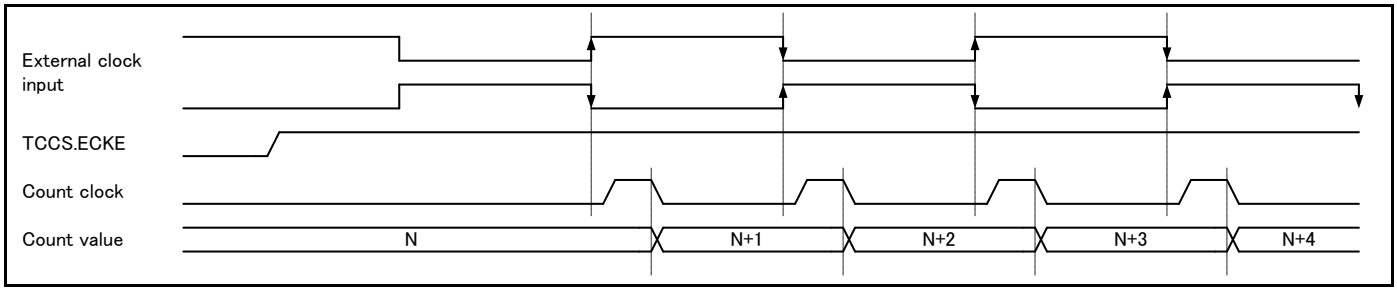


Selected External Count Clock

The 32-bit Free-run Timer counts based on an input clock (peripheral clock or external clock).

If the external clock mode (the Clock Selection bit(ECKE) in the Timer State Control Register (TCCS) ="1") is selected, the 32-bit Free-run Timer starts counting at a rising edge of the external input when the initial value of the external clock input (TEXT) is "H". After that, the 32-bit Free-run Timer counts at both edges. When the initial value of the external input is "L", the 32-bit Free-run Timer starts counting at a falling edge of the external input. After that, the 32-bit Free-run Timer counts up at both edges.

Figure 3-11 Count Timing of 32-Bit Free-Run Timer (in Up Count Mode)



Note:

- *When the external clock input is used, the timer counts at both edges of the external clock.*

3.1. Interrupts of the 32-Bit Free-Run Timer

There are the 2 types of interrupts as the interrupts of the 32-bit Free-run Timer: Compare Clear Interrupt and 0 Detection Interrupt.

32-Bit Free-Run Timer Interrupt

Table 3-1 shows the interrupt control bits and the interrupt factor of the 32-bit Free-run Timer.

Table 3-1 Interrupt Control Bits and Interrupt Factor of 32-Bit Free-Run Timer

	32-Bit Free-Run Timer	
	Compare Clear	**0 Detection**
Interrupt request flag	Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS)	0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS)
Interrupt request enable bit	Compare Clear Interrupt request enable bit (ICRE) in the Timer State Control Register (TCCS)	0 detection interrupt request enable bit (IRQZE) in the Timer State Control Register (TCCS)
Interrupt Factor	The value of the 32-bit Free-run Timer matches the value of the Compare Clear Register (CPCLR).	The 32-bit Free-run Timer value becomes 0x00000000.

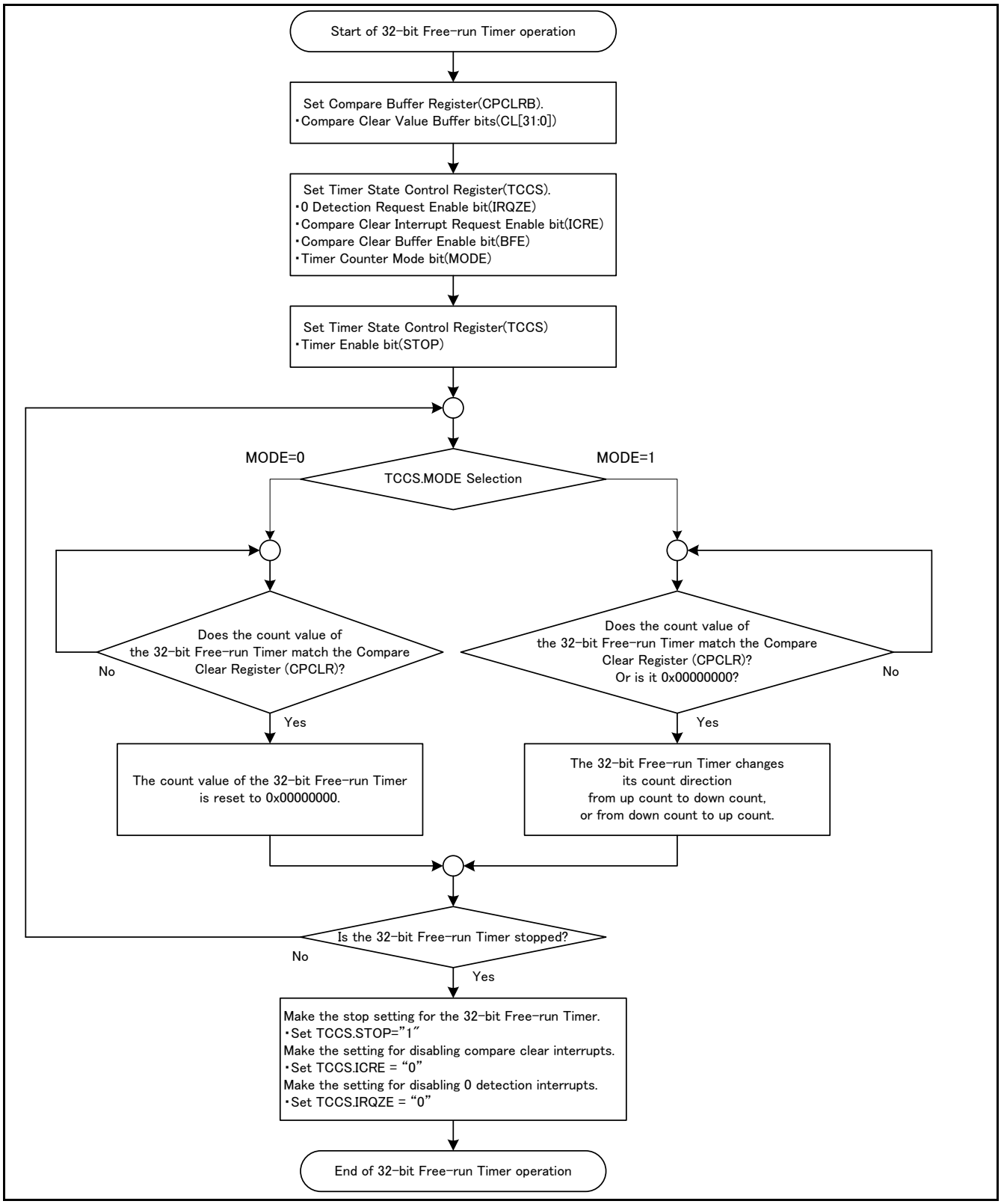
"1" is set to the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) when the value of the 32-bit Free-run Timer matches the value of the Compare Clear Register (CPCLR). When the interrupt request is set to enabled (the Compare Clear Interrupt Request Enable bit(ICRE) in the Timer State Control Register (TCCS) ="1") in this state, then the interrupt request is generated.

"1" is set to the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) when the value of the 32-bit Free-run Timer becomes 0x00000000. When the interrupt request is set to enabled (the 0 Detection Request Enable bit(IRQZE) in the Timer State Control Register (TCCS) ="1") in this state, then the interrupt request is generated.

3.2. Setting Procedure Example of the 32-Bit Free-Run Timer

This section provides a setting procedure example of the 32-bit Free-run Timer.

Figure 3-12 Setting Procedure Example of 32-Bit Free-Run Timer Operation



3.3. Debug Mode

If it goes into a debug mode, operation of the 32-bit Free-run Timer can be stopped and debugged.

If all the following conditions are satisfied, it will go into the debug mode.

- The Debug Enable bit (DBGEN) in the Free-run Timer Debug Register (FRT_DEBUG) is "1."
- The debug input (DEBUG) is active.

It becomes the following operation when it goes into the debug mode.

- The prescaler stop. The state of prescaler is kept.
- The 32-bit Free-run Timer stop. Timer value and timer output value are kept.

It becomes the following operation when it restores from the debug mode.

- The prescaler is resumed from the stopped value.
- The 32-bit Free-run Timer is resumed from the stopped value.

Note:

- *Register access is possible in the debug mode.*

4. Registers of the 32-Bit Free-Run Timer

This section provides the register list of the 32-bit Free-run Timer.

Table 4-1 Register Map

Common Peripheral #0 Group (Channel No.:0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 12 and 13)

Offset	Address offset value / Register name				Block name
	+3	+2	+1	+0	
0000_0000	FRTxx_CPCLRB/FRTxx_CPCLR 11111111_11111111_11111111_11111111				Common Peripheral #0
0000_0004	FRTxx_TCDT 00000000_00000000_00000000_00000000				
0000_0008	FRTxx_TCCS 11111111_11111111_00000000_01000000				
0000_000C	FRTxx_TECCS 11111111_11111111_00000000_11111111				
0000_0010	FRTxx_TCCSC 00000000_00000000_00000000_00000000				
0000_0014	FRTxx_TCCSS 00000000_00000000_00000000_00000000				
0000_0018	Reserved 00000000_00000000		FRTxx_DEBUG 00000000_00000000		
0000_001C | 0000_03FC	-				

Note:

- *xx is the channel number (0 to 5, 8 to 13).*

Registers of the 32-Bit Free-Run Timer

Table 4-2 Register List of 32-Bit Free-Run Timer

Abbreviated Register Name	Register Name	Reference
CPCLRB/CPCLR	Compare Clear Buffer Register, Compare Clear Register	4.1
TCDT	Timer Data Register	4.2
TCCS	Timer State Control Register	4.3
TECCS	Timer Extended State Control Register	4.4
TCCSC	Timer State Control Clear Register	4.5
TCCSS	Timer State Control Set Register	4.6
FRT_DEBUG	Free-run Timer Debug Register	4.7

4.1. Compare Clear Buffer Register (CPCLRB)/Compare Clear Register (CPCLR)

The Compare Clear Buffer Register (CPCLRB) is the buffer register of the Compare Clear Register (CPCLR). Both registers are located in the same address.

Compare Clear Buffer Register (CPCLRB)

bit	31	30	29	28	27	26	25	24
Field	CL[31:24]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	CL[23:16]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	CL[15:8]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	CL[7:0]							
R/W Attribute	W							
Protection Attribute	-							
Initial Value	11111111							

[bit31:0] CL[31:0]: Compare Clear Value Buffer Bits

The Compare Clear Buffer Register (CPCLRB) is a buffer register that is located in the same address as the Compare Clear Register (CPCLR).

The value of the Compare Clear Buffer Register (CPCLRB) is immediately transferred to the Compare Clear Register (CPCLR) when the buffer function is set to disabled (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="0") or the 32-bit Free-run Timer is stopped.

The value of the Compare Clear Buffer Register (CPCLRB) is transferred to the Compare Clear Register (CPCLR) when the buffer function is set to enabled (the Compare Clear Buffer Enable bit(BFE) in the Timer State Control Register (TCCS) ="1") and 0 is detected as the count value of the 32-bit Free-run Timer.

Notes:

- *Do not set 0x00000000 in the Compare Clear Buffer Register (CPCLRB).*
- *For access to this register, use 32-bit access instructions.*

Compare Clear Register (CPCLR)

bit	31	30	29	28	27	26	25	24
Field	CL[31:24]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	CL[23:16]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	CL[15:8]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

bit	7	6	5	4	3	2	1	0
Field	CL[7:0]							
R/W Attribute	R							
Protection Attribute	-							
Initial Value	11111111							

[bit31:0] CL[31:0]: Compare Clear Value Bits

The Compare Clear Register (CPCLR) is used to compare the count value of the 32-bit Free-run Timer.

In the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), the count value of the 32-bit Free-run Timer is reset to 0x00000000 when the value of this register matches the count value of the 32-bit Free-run Timer.

In the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the 32-bit Free-run Timer changes the counting direction from up count to down count when the value of the Compare Clear Register (CPCLR) matches the count value of the 32-bit Free-run Timer. The counting direction is changed from down count to up count when 0 is detected.

Note:

- *For access to this register, use 32-bit access instructions.*

4.2. Timer Data Register (TCDT)

The Timer Data Register (TCDT) reads the count value of the 32-bit Free-run Timer. This register can also be used to set the count value of the 32-bit Free-run Timer.

bit	31	30	29	28	27	26	25	24
Field	T[31:24]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	T[23:16]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	T[15:8]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	T[7:0]							
R/W Attribute	R,W							
Protection Attribute	-							
Initial Value	00000000							

[bit31:0] T[31:0]: Timer Data Value Bits

The Timer Data Register (TCDT) is used to read the count value of the 32-bit Free-run Timer.

The count value can be set by writing a value to this register. However, a value needs to be written while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").

The counter is cleared to the count value 0x00000000 as soon as any of the following factors occurs.

- Hardware reset. A counter is cleared immediately after being hardware reset.
- The Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" while the 32-bit Free-run Timer is operating (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="0").
- The value of the Compare Clear Register (CPCLR) matches the timer count value in the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0").
- The Timer Data Register (TCDT) is set to 0x00000000 while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").

If the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" or the value of the Compare Clear Register (CPCLR) matches the timer count value, the counter is cleared in synchronization with the count timing.

Notes:

- *The 32-bit Free-run Timer is not cleared to 0x00000000 even if the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) is set to "1" while the 32-bit Free-run Timer is stopped (the Timer Enable bit(STOP) in the Timer State Control Register (TCCS) ="1").*
- *For access to this register, use 32-bit access instructions.*

4.3. Timer State Control Register (TCCS)

The Timer State Control Register (TCCS) is a register that is used to control the operation of the 32-bit Free-run Timer.

For details on writing to this register, see "5.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	15	14	13	12	11	10	9	8
Field	ECKE	IRQZF	IRQZE	MSI[2:0]			ICLR	ICRE
R/W Attribute	R/W	R,WX	R/W	R,W			R,WX	R/W
Protection Attribute	-							
Initial Value	0	0	0	000			0	0

bit	7	6	5	4	3	2	1	0
Field	BFE	STOP	MODE	SCLR	CLK[3:0]			
R/W Attribute	R/W	R/W	R/W	R0,W	R/W			
Protection Attribute	-							
Initial Value	0	1	0	0	0000			

[bit31:16] Reserved Bits

Reading these bits returns an undefined value.

Writing data to these bits has no effect on the operation.

[bit15] ECKE: Clock Selection Bit

This bit is used to select a bus clock or an external clock as the count clock of the 32-bit Free-run Timer.

When this bit is set to "0", a bus clock is selected.

To select the count clock frequency, the Clock Frequency Selection bits (CLK[3:0]) must also be selected.

When this bit is set to "1", an external clock is selected and the clock is input from the external clock input (TEXT).

This bit is cleared to "0" by writing "1" to the Clock Selection Clear bit(ECKEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Clock Selection Set bit(ECKES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Bus clock
1	External clock

Note:

- *The count clock is changed immediately when this bit is changed. Therefore, change this bit when the 32-bit Output Compare and the 32-bit Input Capture are stopped if the 32-bit Free-run Timer is connected to them.*

[bit14] IRQZF: 0 Detection Interrupt Flag

This flag is set to "1" when the count value of the 32-bit Free-run Timer is 0x00000000.

This flag is a read-only bit. Writing data to this flag has no effect on the operation.

This flag is cleared by writing "1" to the 0 Detection Interrupt Flag Clear(IRQZFC) in the Timer State Control Clear Register (TCCSC).

Bit	Description
0	0 is not detected.
1	0 is detected.

Notes:

- *This bit is not set to "1" when the timer is cleared ("1" is written to the Timer Clear bit(SCLR)) while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP)="0").*
- *In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), this flag is set to "1" when an interrupt that is set by the Interrupt Mask Selection bits (MSI[2:0]) occurs. This flag is not set to "1" when no interrupt occurs.*
- *In the up count mode (the Timer Count Mode bit(MODE) ="0"), this flag is set to "1" every time 0 is detected regardless of the value of the Interrupt Mask Selection bits (MSI[2:0]).*

[bit13] IRQZE: 0 Detection Request Enable Bit

When this bit is set to "1" and the 0 Detection Interrupt Flag (IRQZF) is set to "1", an interrupt request is generated.

This bit is cleared to "0" by writing "1" to the 0 Detection Request Enable Clear bit(IRQZEC) in the Timer State Control Clear Register (TCCSC).

This bit is set to "1" by writing "1" to the 0 Detection Request Enable Set bit (IRQZES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit12:10] MSI[2:0]: Interrupt Mask Selection Bits

For the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="0"

- In the up count mode (the Timer Count Mode bit(MODE)="0"), these bits are used to set the number of times to mask the Compare Clear Interrupt flag. In the up/down count mode bit (the Timer Count Mode bit(MODE) ="1"), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- When these bits are set to "000", the interrupt flags are not masked.

For the Interrupt Mask Mode bit 2(MODE2) in the Timer Extended State Control Register (TECCS) ="1"

- In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), these bits are used to set the number of times to mask the 0 detection interrupt flag.
- In the up count mode (the Timer Count Mode bit(MODE) ="0"), it is prohibited to use these bits to make this setting.

Bit MSI[2:0]	Description
000	Generate an interrupt flag at the first match occurrence.
001	Generate an interrupt flag at the second match occurrence.
010	Generate an interrupt flag at the third match occurrence.
011	Generate an interrupt flag at the fourth match occurrence.
100	Generate an interrupt flag at the fifth match occurrence.
101	Generate an interrupt flag at the sixth match occurrence.
110	Generate an interrupt flag at the seventh match occurrence.
111	Generate an interrupt flag at the eighth match occurrence.

Notes:

- *The value read is a mask counter value. The mask counter is a decrement counter.*
- *The written data is written to the mask register.*
- *If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.*

[bit9] ICLR: Compare Clear Interrupt Flag

This flag is set to "1" when the value of the Compare Clear Register (CPCLR) and the value of the 32-bit Free-run Timer match.

This flag is a read-only bit. Writing data to these bits has no effect on the operation.

This flag is cleared by writing "1" to the Compare Clear Interrupt Flag Clear(ICLRC) in the Timer State Control Clear Register (TCCSC).

Bit	Description
0	No compare clear match.
1	Compare clear match.

Notes:

- *In the up count mode (the Timer Count Mode bit(MODE) ="0"), this flag is set to "1" when the interrupt flag set by the Interrupt Mask Selection bits (MSI[2:0]) occurs. This flag is not set to "1" when no interrupt occurs.*
- *In the up/down count mode (the Timer Count Mode bit(MODE) ="1"), regardless of the value of the Interrupt Mask Selection bits (MSI[2:0]), this flag is set to "1" every time the compare clear occurs.*

[bit8] ICRE: Compare Clear Interrupt Request Enable Bit

When this bit is set to "1" and the Compare Clear Interrupt Flag (ICLR) is set to "1", an interrupt request is generated.

This bit is cleared to "0" by writing "1" to the Compare Clear Interrupt Request Enable Clear bit(ICREC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Compare Clear Interrupt Request Enable Set bit(ICRES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit7] BFE: Compare Clear Buffer Enable Bit

This bit is used to enable the Compare Clear Buffer Register (CPCLRB).

When this bit is set to "0":

The Compare Clear Buffer Register (CPCLRB) becomes invalid. Thus, data can be written directly to the Compare Clear Register (CPCLR).

When this bit is set to "1":

The Compare Clear Buffer Register (CPCLRB) becomes valid. The data written and stored in the Compare Clear Buffer Register (CPCLRB) is transferred to the Compare Clear Register (CPCLR) when "0" is detected as the count value of the 32-bit Free-run Timer.

This bit is cleared to "0" by writing "1" to the Compare Clear Buffer Enable Clear bit(BFEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Compare Clear Buffer Enable Set bit(BFES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Disable the Compare Clear Buffer.
1	Enable the Compare Clear Buffer.

[bit6] STOP: Timer Enable Bit

This bit is used to stop/start counting of the 32-bit Free-run Timer.

When this bit is set to "0":

The 32-bit Free-run Timer starts counting.

When this bit is set to "1":

The 32-bit Free-run Timer stops counting.

This bit is cleared to "0" by writing "1" to the Timer Enable Clear bit(STOPC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Timer Enable Set bit(STOPS) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Enable counting (start counting).
1	Disable counting (stop counting).

[bit5] MODE: Timer Count Mode Bit

This bit is used to select a count mode of the 32-bit Free-run Timer.

When this bit is set to "0":

The up count mode is selected. The timer continues to count up until the count value matches the value of the Compare Clear Register (CPCLR) and is reset to 0x00000000. After that, it starts counting up again.

When this bit is set to "1":

The up/down count mode is selected. The timer continues to count up until the count value matches the value of the Compare Clear Register (CPCLR). After that, the count direction changes to down count. When the count value reaches 0x00000000, the count direction changes to up count again.

A value can be written to this bit regardless of whether the timer is operating (the Timer Enable bit (STOP) ="0") or stopped (the Timer Enable bit(STOP) ="1"). When the timer is operating, the value written to this bit is stored in a buffer. After that, when the timer value becomes 0x00000000, the count mode is set based on the value stored in the buffer.

This bit is cleared to "0" by writing "1" to the Timer Count Mode Clear bit(MODEC) in the Timer State Control Clear Register (TCCSC).
This bit is set to "1" by writing "1" to the Timer Count Mode Set bit(MODES) in the Timer State Control Set Register (TCCSS).

Bit	Description
0	Up count mode
1	Up/down count mode

[bit4] SCLR: Timer Clear Bit

This bit is used to initialize the 32-bit Free-run Timer.

Initializing the value of the 32-bit Free-run Timer:

The 32-bit Free-run Timer is initialized to 0x00000000 at the next count clock when "1" is written to this bit while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) ="0").

The 32-bit Free-run Timer is not initialized if "1" is written to this bit when the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) ="1").

Initializing the count direction of the 32-bit Free-run Timer:

If this bit is set to "1" while the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer is initialized and then starts counting up.

If this bit is set to "1" while the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) = "1") and then the 32-bit Free-run Timer is restarted (the Timer Enable bit (STOP) = "0"), the 32-bit Free-run Timer always counts up when starting operation.

Suppose the 32-bit Free-run Timer stops operating (the Timer Enable bit (STOP) = "1") while counting down and this bit is set to "1". After that, even if the 32-bit Free-run Timer resumes operation (the Timer Enable bit (STOP) = "1"), the 32-bit Free-run Timer counts up when starting operation.

This bit is set to "1" by writing "1" to the Timer Clear Set bit(SCLRS) in the Timer State Control Set Register (TCCSS).

The read value is always "0".

Bit	Description	
	During Read Operation	During Write Operation
0	"0" always read	Do not initialize the counter.
1		Initialize the counter to 0x00000000.

Notes:

- *The 0 detection interrupt is not generated even if this bit is set to "1".*
- *The timer is not cleared when "0" is written to this bit before the next count clock after "1" is written to this bit.*
- *In the up/down count mode, when "0" is written to the Timer Clear bit(SCLR) before updating the timer count after "1" is written to the Timer Clear bit(SCLR) while the timer is counting down, the count value is not updated and the count direction is changed to up count.*

[bit3:0] CLK[3:0]: Clock Frequency Selection Bits

These bits are used to select the count clock frequency of the 32-bit Free-run Timer.

The clock frequency is changed immediately upon setting these bits.

CLK[3:0]	Description					
	Count Clock	ϕ = 40 MHz	ϕ = 20 MHz	ϕ = 10 MHz	ϕ = 5 MHz	ϕ = 2.5 MHz
0000	ϕ	25 ns	50 ns	100 ns	200 ns	400 ns
0001	ϕ/2	50 ns	100 ns	200 ns	400 ns	800 ns
0010	ϕ/4	100 ns	200 ns	400 ns	800 ns	1.6 us
0011	ϕ/8	200 ns	400 ns	800 ns	1.6 us	3.2 us
0100	ϕ/16	400 ns	800 ns	1.6 us	3.2 us	6.4 us
0101	ϕ/32	800 ns	1.6 us	3.2 us	6.4 us	12.8 us
0110	ϕ/64	1.6 us	3.2 us	6.4 us	12.8 us	25.6 us
0111	ϕ/128	3.2 us	6.4 us	12.8 us	25.6 us	51.2 us
1000	ϕ/256	6.4 us	12.8 us	25.6 us	51.2 us	102.4 us
Other settings are prohibited.	-	-	-	-	-	-

ϕ: Bus clock.

4.4. Timer Extended State Control Register (TECCS)

The Timer Extended State Control Register (TECCS) is an extended control register that controls the operation of the 32-bit Free-run Timer.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

bit	15	14	13	12	11	10	9	8
Field	Reserved				MODE2	MSI[5:3]		
R/W Attribute	RX,WX				R/W	R,W		
Protection Attribute	-							
Initial Value	XXXX				0	000		

bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	RX,WX							
Protection Attribute	-							
Initial Value	XXXXXXXX							

[bit31:12] Reserved: Reserved Bits

Reading these bits returns an undefined value.

Writing data to these bits has no effect on the operation.

[bit11] MODE2: Interrupt Mask Mode Bit 2

This bit is used to independently mask the 0 detection interrupt and the Compare Clear Interrupt when the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1").

If "1" is written to this bit when the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the value set to the Compare Clear Interrupt Mask Selection bits (MSI[5:3]) of this register is enabled and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) is masked the specified number of times. The value set to the Interrupt Mask Selection bits (MSI[2:0]) in the Timer State Control Register (TCCS) becomes valid as the number of times to mask the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS).

Bit		Description
MODE2	MODE*	
0	0	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are invalid.
0	1	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are invalid.
1	0	The setting is prohibited (the operation is not guaranteed).
1	1	The setting values of Compare Clear Interrupt Mask Selection bits(MSI[5:3]) are valid.

* Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS)

Note:

- *The operation when "1" is written to this bit is not guaranteed when the 32-bit Free-run Timer is in the up count mode.*

[bit10:8] MSI[5:3]: Compare Clear Interrupt Mask Selection Bits

These bits are valid only when the Interrupt Mask Mode bit 2 (MODE2) is "1" and the 32-bit Free-run Timer is in the up/down count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"). These bits are used to set the number of times to mask the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS).

The number of times to mask the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) is set by Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS).

The Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) is not masked when these bits are set to "000".

Bit	Description
MSI[5:3]	
000	Generate an interrupt flag at the first match occurrence.
001	Generate an interrupt flag at the second match occurrence.
010	Generate an interrupt flag at the third match occurrence.
011	Generate an interrupt flag at the fourth match occurrence.
100	Generate an interrupt flag at the fifth match occurrence.
101	Generate an interrupt flag at the sixth match occurrence.
110	Generate an interrupt flag at the seventh match occurrence.
111	Generate an interrupt flag at the eighth match occurrence.

Notes:

- *The value read is a mask counter value. The mask counter is a decrement counter.*
- *The written data is written to the mask register.*
- *While the 32-bit Free-run Timer is operating (the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) ="0"), the value written to the mask register is reloaded to the counter only when the mask counter becomes "000".*
- *When the 32-bit Free-run Timer is stopped (the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) ="1"), the value written to the mask register is immediately reloaded to the mask counter.*

[bit7:0] Reserved: Reserved Bits

Reading these bits returns an undefined value.

Writing data to these bits has no effect on the operation.

4.5. Timer State Control Clear Register (TCCSC)

The Timer State Control Clear Register (TCCSC) is a register that is used to clear bits of the Timer State Control Register (TCCS).

For details on writing to this register, see "5.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field		Reserved						
R/W Attribute		R0,W0						
Protection Attribute		-						
Initial Value		00000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	ECKEC	IRQZFC	IRQZEC	Reserved			ICLRC	ICREC
R/W Attribute	R0,W	R0,W	R0,W	R0,W0			R0,W	R0,W
Protection Attribute	-							
Initial Value	0	0	0	0			0	0

bit	7	6	5	4	3	2	1	0
Field	BFEC	STOPC	MODEC	Reserved				
R/W Attribute	R0,W	R0,W	R0,W	R0,W0				
Protection Attribute	-							
Initial Value	0	0	0	00000				

[bit31:16] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit15] ECKEC: Clock Selection Clear Bit

It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "0".	

[bit14] IRQZFC: 0 Detection Interrupt Flag Clear

It is the bit which sets the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) to "0".	

[bit13] IRQZEC: 0 Detection Request Enable Clear Bit

It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "0".	

[bit12:10] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit9] ICLRC: Compare Clear Interrupt Flag Clear

It is the bit which sets the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) to "0".	

[bit8] ICREC: Compare Clear Interrupt Request Enable Clear Bit

It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "0".	

[bit7] BFEC: Compare Clear Buffer Enable Clear Bit

It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "0".	

[bit6] STOPC: Timer Enable Clear Bit

It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "0".	

[bit5] MODEC: Timer Count Mode Clear Bit

It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "0".	

[bit4:0] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

4.6. Timer State Control Set Register (TCCSS)

The Timer State Control Set Register (TCCSS) is a register that is used to set bits of the Timer State Control Register (TCCS).

For details on writing to this register, see "5.Precautions for Using This Device."

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	ECKES	Reserved	IRQZES	Reserved				ICRES
R/W Attribute	R0,W	R0,W0	R0,W	R0,W0				R0,W
Protection Attribute	-							
Initial Value	0	0	0	0000				0

bit	7	6	5	4	3	2	1	0
Field	BFES	STOPS	MODES	SCLRS	Reserved			
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-							
Initial Value	0	0	0	0	0000			

[bit31:16] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit15] ECKES: Clock Selection Set Bit

It is the bit which sets the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Clock Selection bit (ECKE) in the Timer State Control Register (TCCS) to "1".	

[bit14] Reserved: Reserved Bit

Reading this bit returns "0".

Always write "0" to this bit.

[bit13] IRQZES: 0 Detection Request Enable Set Bit

It is the bit which sets the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the 0 Detection Request Enable bit (IRQZE) in the Timer State Control Register (TCCS) to "1".	

[bit12:9] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

[bit8] ICRES: Compare Clear Interrupt Request Enable Set Bit

It is the bit which sets the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Interrupt Request Enable bit (ICRE) in the Timer State Control Register (TCCS) to "1".	

[bit7] BFES: Compare Clear Buffer Enable Set Bit

It is the bit which sets the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Compare Clear Buffer Enable bit (BFE) in the Timer State Control Register (TCCS) to "1".	

[bit6] STOPS: Timer Enable Set Bit

It is the bit which sets the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Enable bit (STOP) in the Timer State Control Register (TCCS) to "1".	

[bit5] MODES: Timer Count Mode Set Bit

It is the bit which sets the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) to "1".	

[bit4] SCLRS: Timer Clear Set Bit

It is the bit which sets the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".

Bit	Description	
	Writing	Reading
0	No effect.	Always read "0".
1	Set the Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) to "1".	

[bit3:0] Reserved: Reserved Bits

Reading these bits returns "0".

Always write "0" to these bits.

4.7. Free-Run Timer Debug Register(FRT_DEBUG)

The Free-run Timer Debug Register (FRT_DEBUG) performs enable/disable setting of debug.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBGEN
R/W Attribute	R0,W0							R/W
Protection Attribute	-							
Initial Value	0000000							0

[bit15:1] Reserved: Reserved Bits

The read value is "0".

If writing to this bit, write "0".

[bit0] DBGEN: Debug Enable Bit

It is the bit which permits stopping operation of a 32-bit Free-run Timer with the debug input (DEBUG).

Bit	Description
0	Disable
1	Enable

5. Precautions for Using This Device

The following shows the notes when using the 32-bit Free-run Timer.

Notes to Observe when Accessing a Register

When Accessing the Compare Clear Register (CPCLR) or the Compare Clear Buffer Register (CPCLRB)

Use 32-bit access instructions to the Compare Clear Register (CPCLR) and the Compare Clear Buffer Register (CPCLRB).

When Accessing the Timer State Control Register (TCCS)

- To clear a specified bit of this register, clear the bit by writing "1" to the applicable bit of the Timer State Control Clear Register (TCCSC).
- To set a specified bit of this register, set the bit by writing "1" to the applicable bit of the Timer State Control Set Register (TCCSS).
- Data can be written directly to this register only when writing to all bits.
- In the normal reading mode, the interrupt mask counter value is read from Interrupt Mask Selection bits(MSI[2:0]) in the Timer State Control Register (TCCS).
- If Interrupt mask function "by setting TCCS.MSI[2:0] Register" is used, please write TCCS[15:8], TCCSC[15:8] and TCCSS[15:8] after 32bit free-run timer stopped.

When Accessing the Timer Extended State Control Register (TECCS)

- In the normal reading mode, the interrupt mask counter value is read from Compare Clear Interrupt Mask Selection bits(MSI[5:3]) in the Timer Extended State Control Register (TECCS).

Notes when Operating the 32-Bit Free-Run Timer

When Setting Using a Program

- When the hardware is reset, the count value becomes 0x00000000, but the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) is not set.
- Because the timer mode bit (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)) has a buffer, a specified timer mode is enabled after 0 is detected.
- The Timer Clear bit (SCLR) in the Timer State Control Register (TCCS) ="1" initializes the timer. However, this bit does not generate the 0 detection interrupt.
- The Compare Clear Flag is not set if the timer starts counting when the value of the Compare Clear Register (CPCLR) matches the count value.
- Set any values other than 0x00000000 to the Compare Clear Register (CPCLR). Note that the following operations occur if 0x00000000 is set.
  - When the timer mode bit (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS)) is in the up count mode (the Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="0"), the count value is updated to 0x00000000 and fixed to 0x00000000. Then, the 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set at every count clock.
  - When the Timer Count Mode bit (MODE) in the Timer State Control Register (TCCS) is in the up/down count mode (Timer Count Mode bit(MODE) in the Timer State Control Register (TCCS) ="1"), the count value is counted up from 0x00000000 to 0xFFFFFFFF, and this up count operation is repeated. The 0 Detection Interrupt Flag (IRQZF) in the Timer State Control Register (TCCS) and the Compare Clear Interrupt Flag (ICLR) in the Timer State Control Register (TCCS) are set to "1" when the count value becomes 0x00000000.

CHAPTER 20: 32-Bit Input Capture

This chapter describes the functions of the 32-bit input capture.

1. Overview of the 32-Bit Input Capture
2. Explanation of 32-Bit Input Capture Operation
3. Registers of the 32-Bit Input Capture
4. Precautions for Using This Device

CODE: ICU-JUPI-E1

1. Overview of the 32-Bit Input Capture

The 32-bit input capture can measure the input pulse width and external clock cycle based on the value of the 32-bit free-run timer.

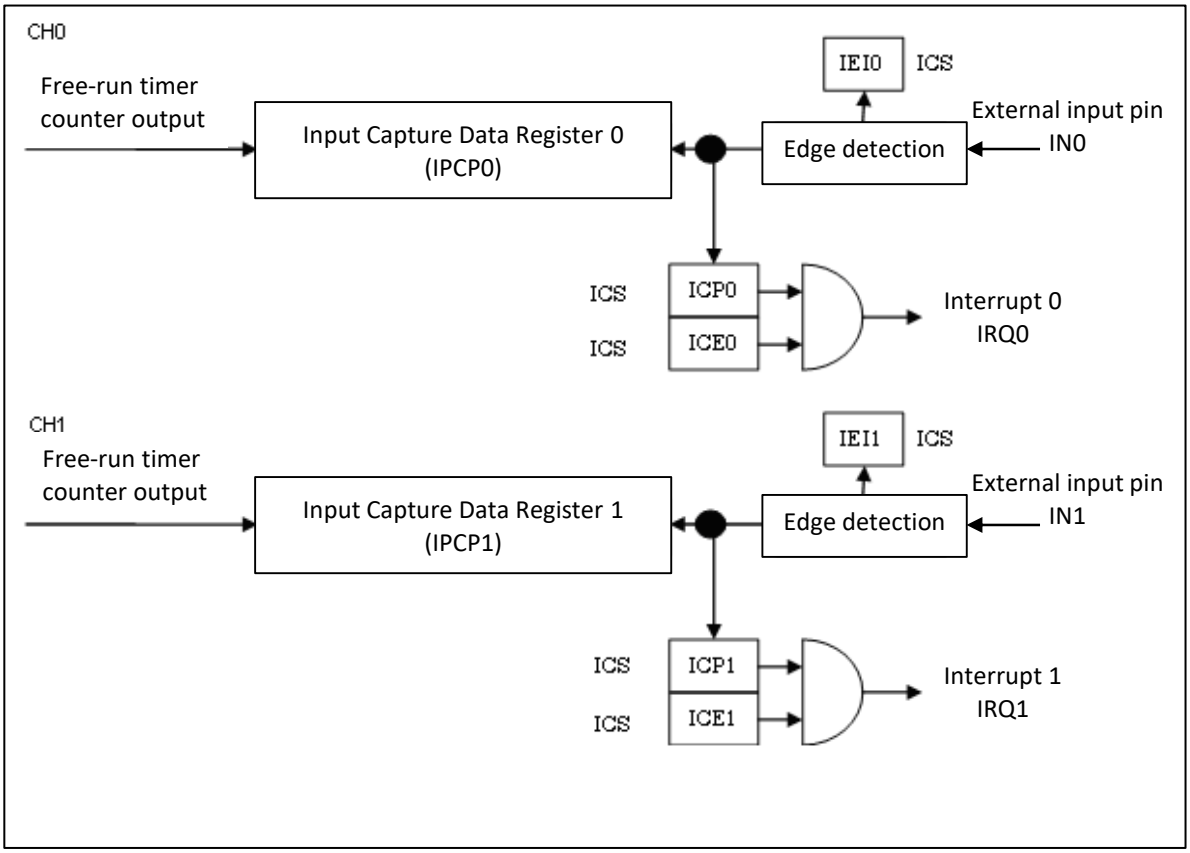
Functions of the 32-Bit Input Capture

- The 32-bit input capture is composed of 2 input captures. Each channel can be operated individually.
- Each input capture is composed of 2 independent external input pins (IN0 and IN1), capture registers that correspond to the pins, and the capture control register. When an edge signal is detected on an external input pin, the value of the free-run timer can be stored in the capture register. At the same time, an interrupt is generated.
- 3 types of trigger edges (rising edge, falling edge, and both edges) of an external input signal can be selected. There is a register that indicates whether a trigger edge is a rising or falling edge.
- An interrupt is generated when a valid edge is detected from an external input signal.

Configuration Diagram of the 32-Bit Input Capture

Figure 1-1 is a configuration diagram of the 32-bit input capture.

Figure 1-1 Configuration Diagram of 32-Bit Input Capture



2. Explanation of 32-Bit Input Capture Operation

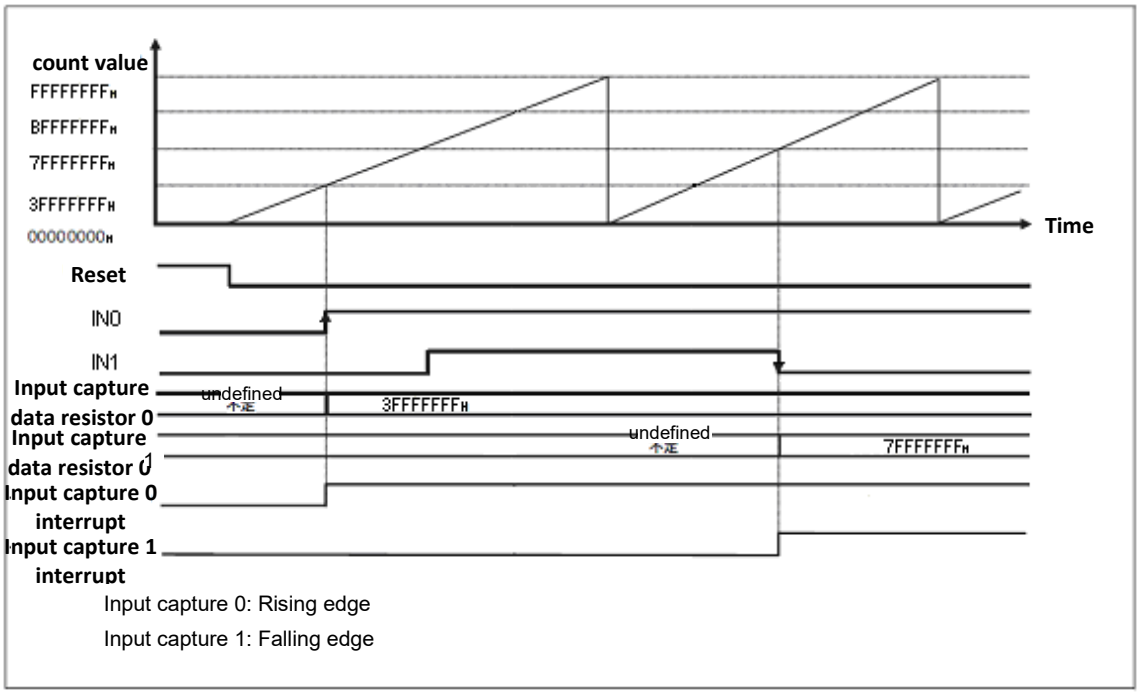
This section provides an overview of 32-bit input capture operation.

Operation of the 32-Bit Input Capture

The 32-bit input capture is used to detect a specified valid edge. When a valid edge is detected, an interrupt flag is set. Then the value of the 32-bit free-run timer is loaded to the input capture register.

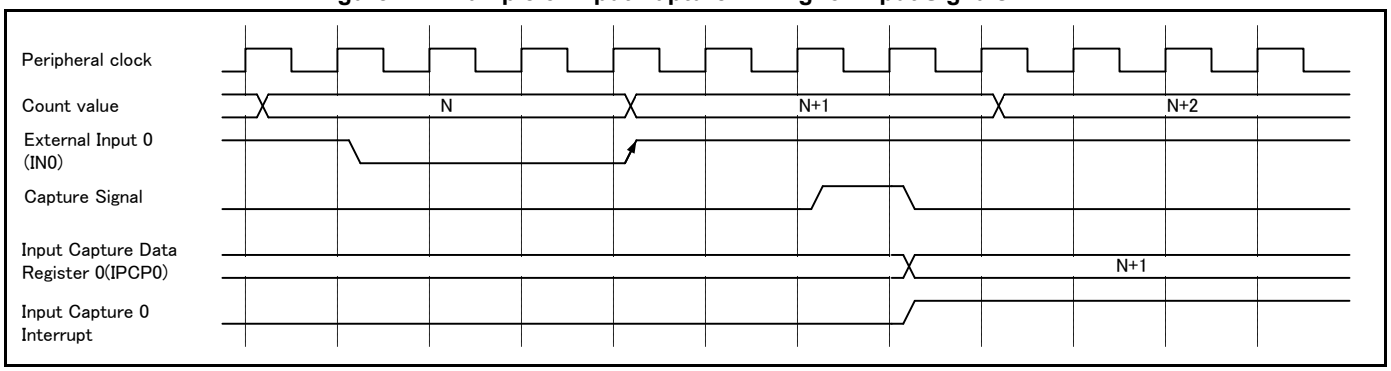
Input Capture Operation

Figure 2-1 Example of Input Capture Timing



Input Capture Input Timing

Figure 2-2 Example of Input Capture Timing for Input Signals



2.1. Interrupt of the 32-Bit Input Capture

As the interrupt of the 32-bit input capture, there is an input capture interrupt triggered by an external input signal.

Input Capture Interrupt

Table 2-1 shows the interrupt control bits and interrupt factor of the input capture.

Table 2-1 Interrupt Control Bits and Interrupt Factor of Input Capture 1, 0

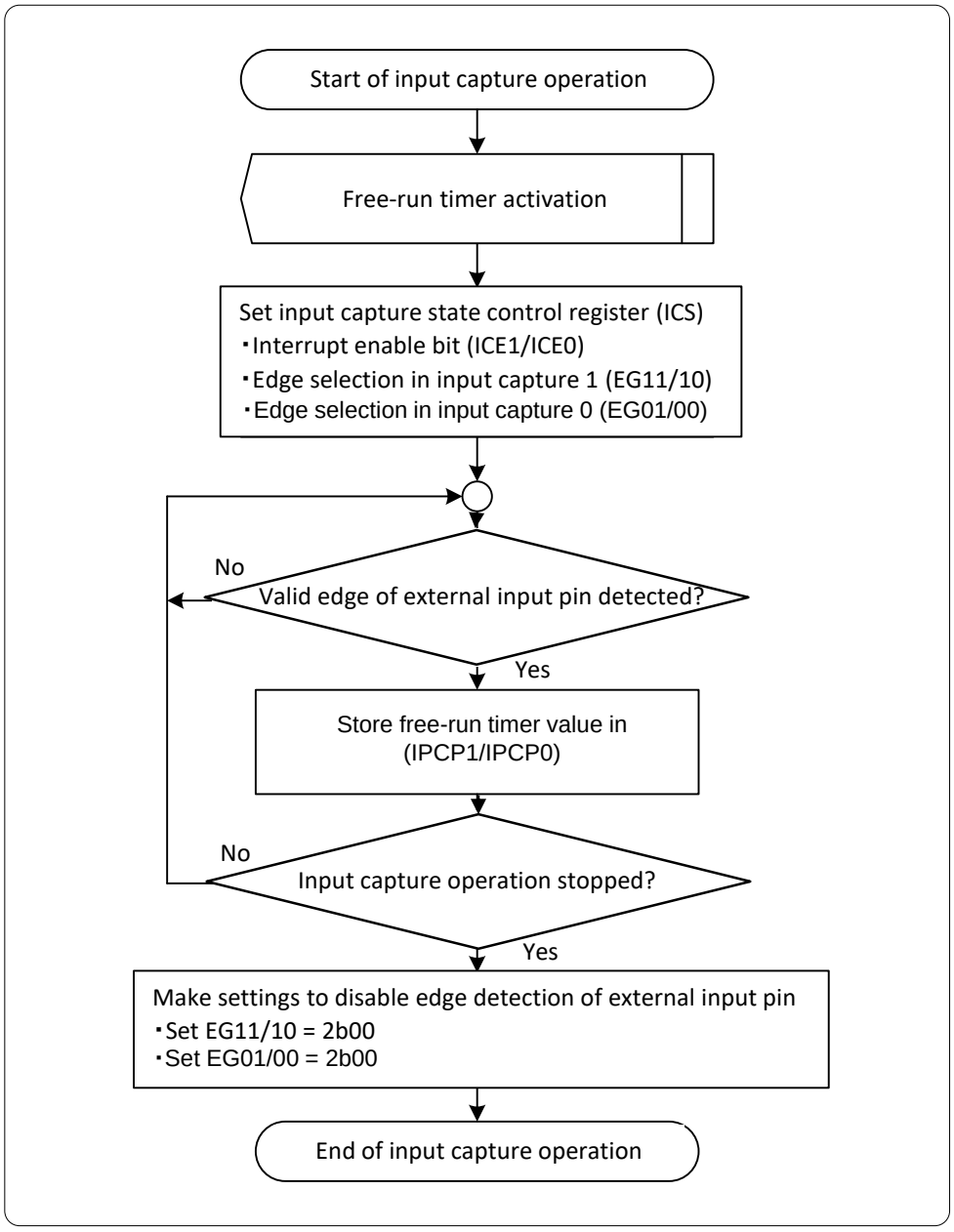
	Input Capture 1, 0
Interrupt Request Flag Bit	The interrupt request flag bits ICP1, 0 (bit7, bit6) in the input capture state control register (ICS)
Interrupt Request Enable Bit	The interrupt request enable bits ICE1, 0 (bit5, bit4) in the input capture state control register (ICS)
Interrupt Factor	A valid edge is detected on the external input pin (IN1, IN0).

For the interrupt capture, when a valid edge is detected on the external input pin (IN1, IN0), the interrupt request flag (ICP1, ICP0: bit7, bit6) in the input capture state control register (ICS) is set to "1". When an interrupt request is set to enabled (ICE1, ICE0: bit5, bit4 = 0b11 in the input capture state control register (ICS)) in this state, then the interrupt request is output to the interrupt controller.

2.2. Setting Procedure Example

This section provides a setting procedure example of the input capture.

Figure 2-3 Procedure Example of Setting Input Capture Operation



3. Registers of the 32-Bit Input Capture

This section provides the register list of the 32-bit input capture.

Table 3-1 Register Map

Common Peripheral Group #0 (Channel No.:0, 1, 2, 8, 9, and 10)

Offset	Address Offset Value / Register Name				Block Name
	+3	+2	+1	+0	
0000_8000	ICUxx_IPCP0 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				Common Peripheral #0
0000_8004	ICUxx_IPCP1 XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX				
0000_8008	ICUxx_ICS 11111111_11111111_11111100_00000000				
0000_800C	ICUxx_ICSC 00000000_00000000_00000000_00000000				
0000_8010	ICUxx_ICSS 00000000_00000000_00000000_00000000				
0000_8014 0000_83FC	-				

Notes:

- "X": Initial value undefined
- "xx": Pair channel number (0 to 2, 8 to 10)

Register List of the 32-Bit Input Capture

Table 3-1-1 Register List of the 32-Bit Input Capture

Abbreviated Register Name	Register Name	Reference
IPCP0/IPCP1	Input capture data registers 0, 1	3.1
ICS	Input capture state control register	3.2
ICSC	Input capture state control clear register	3.3
ICSS	Input capture state control set register	3.4

Register Bit Locations of 32-Bit Input Capture

Table 3-2 Register Bit Locations of 32-Bit Input Capture

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IPCP0	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
IPCP1	CP31	CP30	CP29	CP28	CP27	CP26	CP 25	CP24
	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
ICS	Reserved							
	Reserved							
	Reserved						IEI1	IEI0
	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
ICSC	Reserved							
	Reserved							
	Reserved							
	ICP1C	ICP0C	ICE1C	ICE0C	Reserved			
ICSS	Reserved							
	Reserved							
	Reserved							
	Reserved		ICE1S	ICE0S	Reserved			

3.1. Input Capture Data Registers 0, 1 (IPCP0, IPCP1)

Input capture data registers 0, 1 (IPCP0, IPCP1) are used to store the count value of the free-run timer when a valid edge of an external input signal is detected.

Input Capture Data Register (IPCP0)

bit	31	30	29	28	27	26	25	24
Field	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] CP[31:00]: Input Capture Data Value Bit

The input capture data register 0 (IPCP0) is used to store the value of the free-run timer when a valid edge of the external input pin IN0 signal is detected.

The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.

Note:

- *For access to this register, use word access instructions.*

Input Capture Data Register (IPCP1)

bit	31	30	29	28	27	26	25	24
Field	CP31	CP30	CP29	CP28	CP27	CP26	CP25	CP24
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	CP23	CP22	CP21	CP20	CP19	CP18	CP17	CP16
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8
Field	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
R/W Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	X	X	X	X	X	X	X	X

[bit31:0] CP[31:00]: Input Capture Data Value Bit

Input capture data register 1 (IPCP1) is used to store the value of the free-run timer when a valid edge of the external input pin IN1 signal is detected.

The free-run timer that is mentioned here indicates the operating state of the free-run timer that is connected to the input capture.

Note:

- *For access to this register, use word access instructions.*

3.2. Input Capture State Control Register (ICS)

The input capture state control register (ICS) is used to select an edge, enable an interrupt request, and control an interrupt request flag. This register is also used to indicate a valid edge detected in the input captures 0, 1.

For details on writing to this register, see "4. Precautions for Using This Device."

Flag bits of the input capture state control registers are only read if the APB-IF is valid.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R1,WX							
Protection Attribute	-							
Initial Value	11111111							

bit	15	14	13	12	11	10	9	8
Field	Reserved						IEI1	IEI0
R/W Attribute	R1,WX						R,WX	R,WX
Protection Attribute	-						-	-
Initial Value	111111						0	0

bit	7	6	5	4	3	2	1	0
Field	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
R/W Attribute	R,WX	R,WX	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit31:10] Reserved: Reserved Bits

The read value is "1".
Writing data to these bits has no effect on the operation.

[bit9] IEI1: Valid Edge Indication Bit (Input Capture 1)

- This bit is a valid edge indication bit in input capture data register 1. This bit indicates that a rising or falling edge is detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is read-only.

Bit	Description
0	A falling edge is detected.
1	A rising edge is detected.

Notes:

- *The value read is meaningless if the edge selection bit (EG11, EG10: bit3, bit2) is set to "00_B".*
- *If the edge selection bit (EG11, EG10: bit3, bit2) is set to any value other than "00_B", then the value is updated when the interrupt request flag (ICP1) is set.*

[bit8] IEI0: Valid Edge Indication Bit (Input Capture 0)

- This bit is a valid edge indication bit in input capture data register 0. This bit indicates that a rising or falling edge is detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is a read-only bit.

Bit	Description
0	A falling edge is detected.
1	A rising edge is detected.

Notes:

- *The value read is meaningless if the edge selection bit (EG01, EG00: bit1, bit0) is set to "00_B".*
- *If the edge selection bit (EG01, EG00: bit1, bit0) is set to any value other than "00_B", then the value is updated when the interrupt request flag (ICP0) is set.*

[bit7] ICP1: Interrupt Request Flag Bit (Input Capture 1)

- This bit is used as an interrupt request flag of input capture 1.
- This bit is set to "1" immediately upon detecting a valid edge on the external input pin (IN1).
- An interrupt is generated as soon as this bit is set to "1" when the interrupt request enable bit (ICE1: bit5) is "1".
- This bit is a read-only bit. Writing data to these bits has no effect on the operation.

- This bit is cleared to "0" by writing "1" to the ICP1C bit in the ICSC register.

Bit	Description
0	No valid edge is detected.
1	A valid edge is detected.

[bit6] ICP0: Interrupt Request Flag Bit (Input Capture 0)

- This bit is used as an interrupt request flag of input capture 0.
- This bit is set to "1" immediately upon detecting a valid edge on the external input pin (IN0).
- An interrupt is generated as soon as this bit is set to "1" when the interrupt request enable bit (ICE0: bit4) is "1".
- This bit is a read-only bit. Writing data to these bits has no effect on the operation.
- This bit is cleared to "0" by writing "1" to the ICP0C bit in the ICSC register.

Bit	Description
0	No valid edge is detected.
1	A valid edge is detected.

[bit5] ICE1: Interrupt Request Enable Bit (Input Capture 1)

- This bit is used to enable an interrupt request of input capture1.
- An interrupt of input capture 1 is generated when the interrupt request flag bit (ICP1: bit7) is set while this bit is "1".
- This bit is cleared to "0" by writing "1" to the ICE1C bit in the ICSC register.
- This bit is set to "1" by writing "1" to the ICE1S bit in the ICSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4] ICE0: Interrupt Request Enable Bit (Input Capture 0)

- This bit is used to enable an interrupt request of input capture 0.
- An interrupt of input capture 0 is generated when the interrupt request flag bit (ICP0: bit6) is set while this bit is "1".
- This bit is cleared to "0" by writing "1" to the ICE0C bit in the ICSC register.
- This bit is set to "1" by writing "1" to the ICE0S bit in the ICSS register.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3:2] EG11, EG10: Edge Selection Bits (Input Capture 1)

- These bits are used to enable the operation of input capture 1. These bits specify a valid edge of the external input (IN1).

Bits		Description
EG11	EG10	
0	0	No edge is detected (stop).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

[bit1:0] EG01, EG00: Edge Selection Bits (Input Capture 0)

■ These bits are used to enable the operation of input capture 0. These bits specify a valid edge of the external input (IN0).

Bits		Description
EG01	EG00	
0	0	No edge is detected (stop).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

3.3. Input Capture State Control Clear Register (ICSC)

The input capture state control clear register (ICSC) is a register to clear a bit in the input capture state control register (ICS).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	ICP1C	ICP0C	ICE1C	ICE0C	Reserved			
R/W Attribute	R0,W	R0,W	R0,W	R0,W	R0,W0			
Protection Attribute	-	-	-	-	-			
Initial Value	0	0	0	0	0000			

[bit31:8] Reserved: Reserved Bits

The read value is "0".
Always write "0" to this bit.

[bit7] ICP1C: ICP1 Clear Bit

- Writing "1" to this bit clears the ICP1 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request flag bit (input capture1) (ICP1) in the input capture state control register (ICS).
1	Clear the interrupt request flag bit (input capture1) (ICP1) in the input capture state control register (ICS).

[bit6] ICP0C: ICP0 Clear Bit

- Writing "1" to this bit clears the ICP0 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request flag bit (input capture0) (ICP0) in the input capture state control register (ICS).
1	Clear the interrupt request flag bit (input capture0) (ICP0) in the input capture state control register (ICS).

[bit5] ICE1C: ICE1 Clear Bit

- Writing "1" to this bit clears the ICE1 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).
1	Clear the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).

[bit4] ICE0C: ICE0 Clear Bit

- Writing "1" to this bit clears the ICE0 bit in the ICS register.
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).
1	Clear the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).

[bit3:0] Reserved: Reserved Bits

The read value is "0".
Always write "0" to this bit.

3.4. Input Capture State Control Set Register (ICSS)

The input capture state control set register (ICSS) is a register that sets the bit in the input capture state control register (ICS).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,W0							
Protection Attribute	-							
Initial Value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved		ICE1S	ICE0S	Reserved			
R/W Attribute	R0,W0		R0,W	R0,W	R0,W0			
Protection Attribute	-		-	-	-			
Initial Value	00		0	0	0000			

[bit31:6] Reserved: Reserved Bits

The read value is "0".
Always write "0" to this bit.

[bit5] ICE1S: ICE1 Set Bit

- Writing "1" to this bit sets the ICE1 bit in the ICS register to "1".
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).
1	Set the interrupt request enable bit (input capture 1) (ICE1) in the input capture state control register (ICS).

[bit4] ICE0S: ICE0 Set Bit

- Writing "1" to this bit sets the ICE0 bit in the ICS register to "1".
- "0" is always read from this bit.

Bit	Description
0	Do not affect this bit and the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).
1	Set the interrupt request enable bit (input capture 0) (ICE0) in the input capture state control register (ICS).

[bit3:0] Reserved: Reserved Bit

The read value is "0".
Always write "0" to this bit.

4. Precautions for Using This Device

The following shows the notes when using the 32-bit input capture.

Notes to Observe when Accessing a Register

When Accessing the Input Capture Data Registers 0, 1 (IPCP0, 1)

Use word access instructions for the input capture data registers 0, 1 (IPCP0, 1).

When Accessing the Input Capture State Control Register (ICS)

- This register supports writing from the bit-band alias area. For the bit-band alias area, see the chapter of "Bit-Band Unit" in Traveo™ Platform hardware manual.
- To clear a specified bit in this register, clear the bit by writing "1" to the applicable bit in the input capture state control clear register (ICSC).
- To set a specified bit in this register, set the bit by writing "1" to the applicable bit in the input capture state control set register (ICSS).
- Data can be written directly to this register only when writing to all bits.

Notes on Interrupt Processing

- The valid edge indication bit (IEI1, IEI0: bit9, bit8) in the input capture state control register (ICS) indicates a detected latest edge when the level of the external input pin (IN0, IN1) switches while an interrupt routine is processing after the interrupt request flag (ICP1, ICP0) in the input capture state control register (ICS) is set to "1".

Notes on Input Capture Operation

About Capture Timing

- Capture resolution is 1 peripheral clock because the input capture operates according to the peripheral clock timing. The resolution is 1 timer count because the capture data is the timer counter value of the free-run timer.

CHAPTER 21: 32-bit Reload Timer



This chapter explains 32-bit Reload Timer

1. Overview
2. Configuration and Block Diagram
3. Operation of the 32-bit Reload Timer
4. Registers

CODE: RLT-JUPI-E1

1. Overview

This section gives a brief overview of the 32-bit Reload Timer.

Features of 32-bit Reload Timer

The 32-bit Reload Timer consists of a 32-bit down counter, a 32-bit Reload register, one input(TIN), one output (TOT), and control registers. The 32-bit Reload Timer has the following features:

- External and internal clock/event source
- Trigger signal programmable as rising/falling edge or both
- Gated count function
- One-shot or reload counter mode
- Counter state can be made visible at external pin
- Prescaler with six different settings for the internal clock and two different settings for the external clock
- Several Reload Timers can be cascaded to form a longer Reload Timer
- Support for MCU debug mode

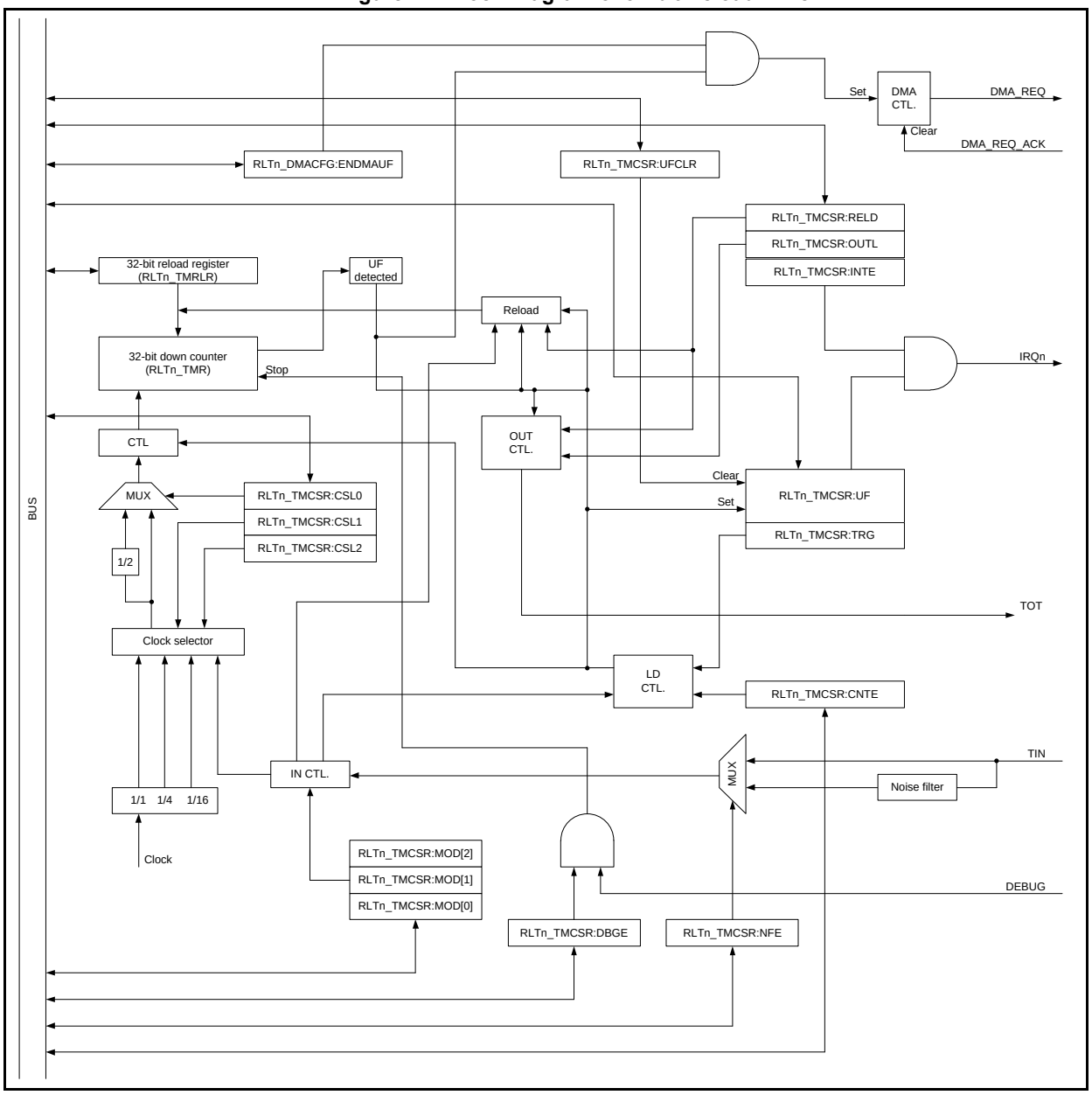
DMA and Interrupts

The 32-bit Reload Timer can generate DMA request which can be used to start DMA transfers.
The 32-bit Reload Timer can generate interrupt in case of underflow.

2. Configuration and Block Diagram

This section shows a block diagram of the 32-bit Reload Timer.

Figure 2-1 Block Diagram of 32-bit Reload Timer



3. Operation of the 32-bit Reload Timer

This section describes the operation of the 32-bit Reload Timer.

3.1 Internal Clock and External Event Counter Operations of 32-bit Reload Timer
3.2 Underflow Operation of 32-bit Reload Timer
3.3 Output Functions of 32-bit Reload Timer
3.4 Counter Operation State
3.5 DMA Operation

3.1. Internal Clock and External Event Counter Operations of 32-bit Reload Timer

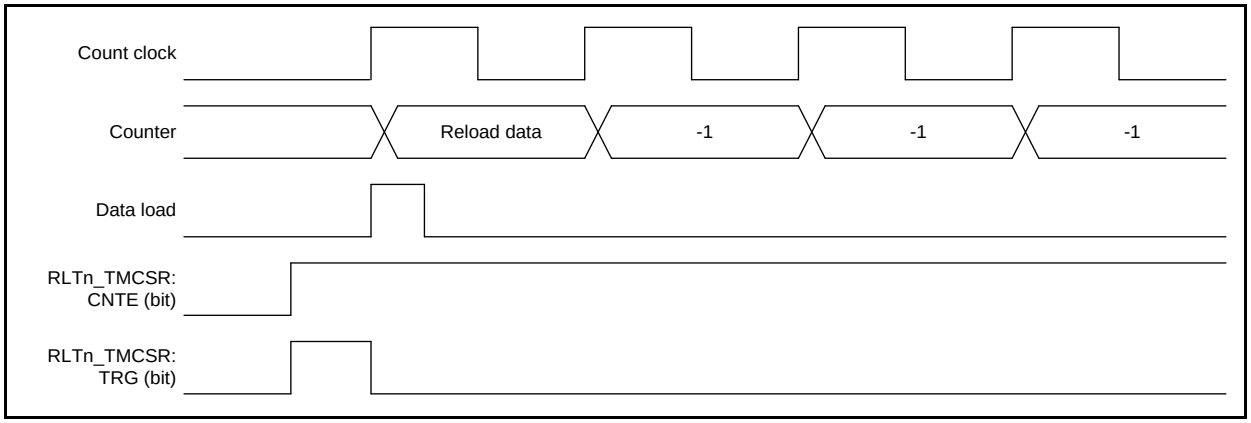
In internal clock mode, the peripheral clock with different divider settings can be selected as the clock source for operating the Reload Timer. The external input TIN can be selected as either a trigger input, or as a gate input by a register setting. In event counter mode, the TIN is used as an external event input. Each active edge on this input (rising, falling, or both) decrements the counter.
When RLTn_TMCSR:CSL0 = 1, then each second event is counted.

Internal Clock Operation of 32-bit Reload Timer

Writing "1" to both RLTn_TMCSR:CNTE and RLTn_TMCSR:TRG bits enables and starts counting at the same time. Using the RLTn_TMCSR:TRG bit as a trigger input is always available when the timer is enabled (RLTn_TMCSR:CNTE = 1), regardless of the operation mode.

Figure 3-1 shows counter activation and counter operation.

Figure 3-1 Activation and Operation of 32-bit Reload Timer Counter



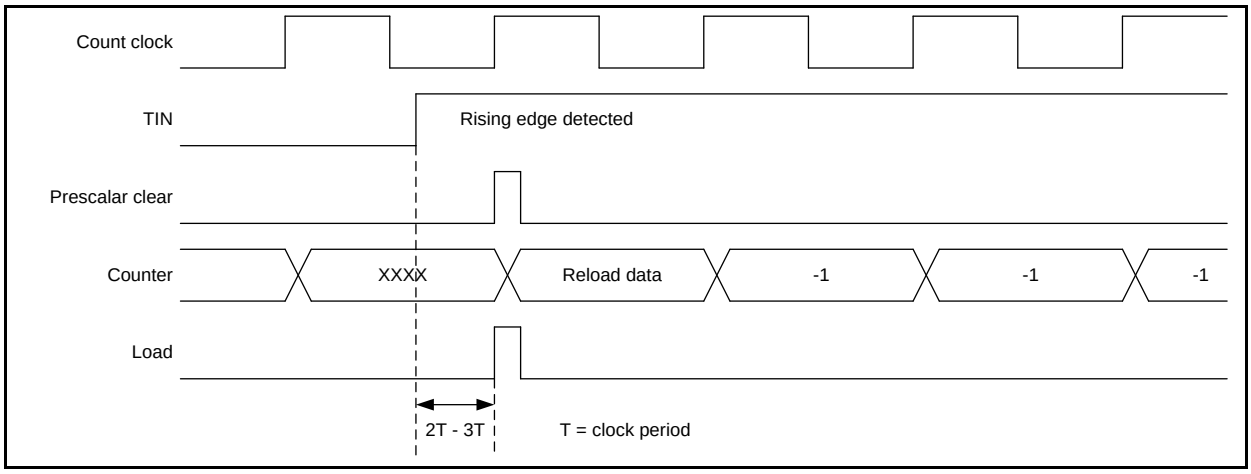
Input Functions of 32-bit Reload Timer (In Internal Clock Mode)

The TIN input can be used as either a trigger input, or as a gate input, when an internal clock is selected as the clock source.

Trigger Input

When used as a trigger input, an active edge causes the timer to load the reload register contents and resets the internal prescaler. Then, count operation starts. For the minimum pulse width length of TIN refer to device specific datasheet.

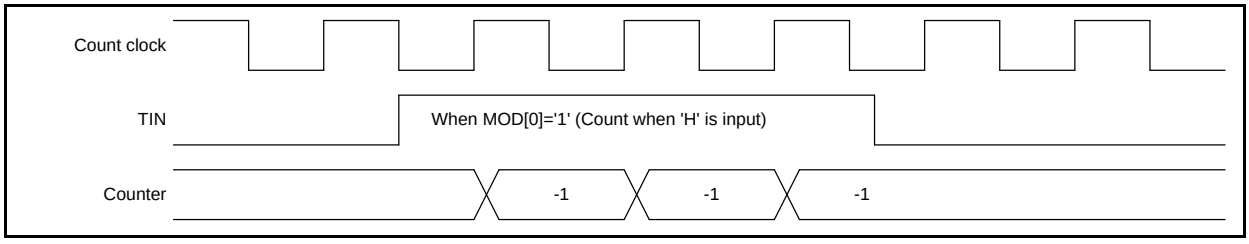
Figure 3-2 Trigger Input Operation of 32-bit Reload Timer



Gate Input

When used as a gate input, the counter only counts while the active level specified by the RLTn_TMCSR:MOD[0] bit is input to the TIN. In this case, the count clock continues to operate unless stopped. The software trigger can be used in gate mode, regardless of the gate level. For the minimum pulse width length of TIN refer to device specific datasheet.

Figure 3-3 Gate Input Operation of 32-bit Reload Timer



External Event Counter

When external event count mode is selected, the TIN is used as an external event input. The counter counts on the active edge specified in the RLTn_TMCSR. For the minimum pulse width length of TIN refer to device specific datasheet.

3.2. Underflow Operation of 32-bit Reload Timer

An underflow is defined for this timer as the time when the counter value changes from 0x00000000 to 0xFFFFFFFF or reload occurs (RLTn_TMCSR:RELD= 1). Therefore, an underflow occurs after (reload register setting + 1) counts.

Underflow Operation of 32-bit Reload Timer

If the RLTn_TMCSR:RELD bit is "1" and an underflow occurs, the contents of the reload register is loaded into the counter and counting continues.

If the RLTn_TMCSR:RELD bit is "0", counting stops when counter reaches 0xFFFFFFFF.

The RLTn_TMCSR:UF bit is set when the underflow occurs. If the RLTn_TMCSR:INTE bit is "1" at this time, an interrupt request is generated.

Figure 3-4 shows the operation when an underflow occurs with various values of RLTn_TMCSR:RELD. Figure 3-5 shows the clearing operation of underflow flag.

Figure 3-4 Underflow Operation of 32-bit Reload Timer

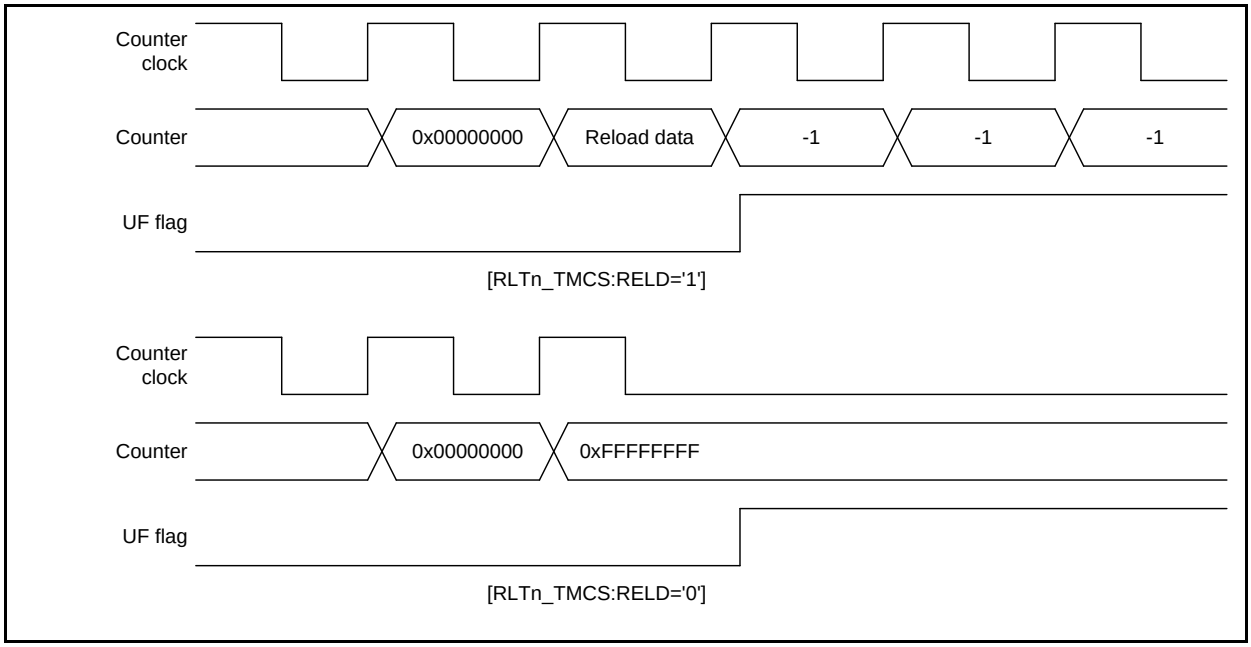
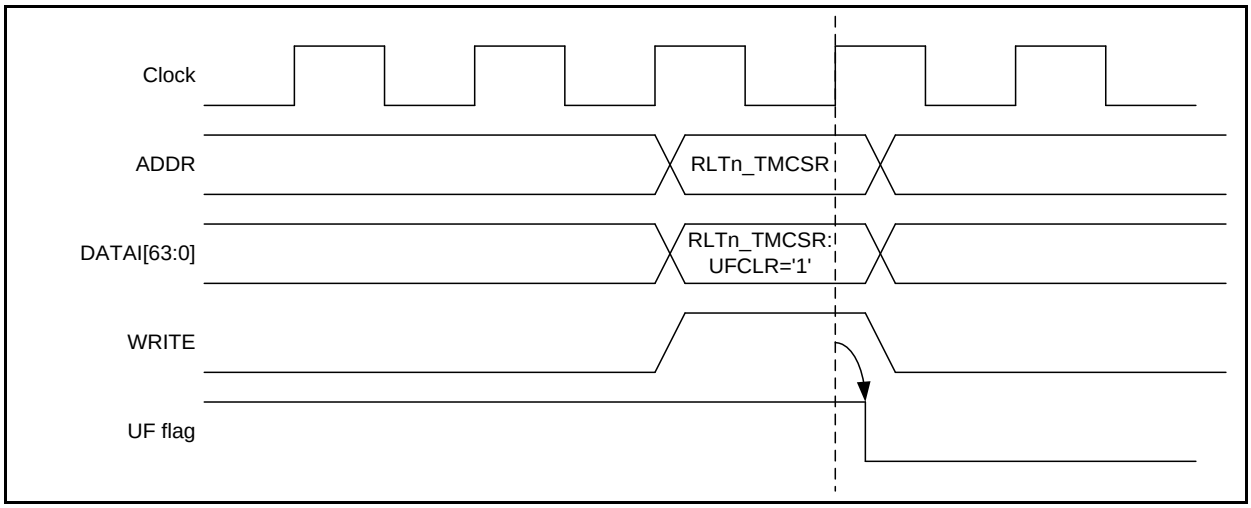


Figure 3-5 Clearing of Underflow Bit



3.3. Output Functions of 32-bit Reload Timer

In reload mode, the TOT output performs toggle output (inverts at each underflow). In one-shot mode, the TOT output is used as a pulse output that shows the configured level while the counting is in progress.

Output Signal Functions of 32-bit Reload Timer

The RLTn_TMCSR:OUTL bit sets the output polarity.
When RLTn_TMCSR:OUTL = 0, the initial value for toggle output is "L" and the one-shot pulse output is "H" while the count is in progress.
When RLTn_TMCSR:OUTL = 1, the output waveforms are opposite.
Figure 3-6 and Figure 3-7 show the output signal functions.

Figure 3-6 Output Signal Function of 32-bit Reload Timer in Reload Mode

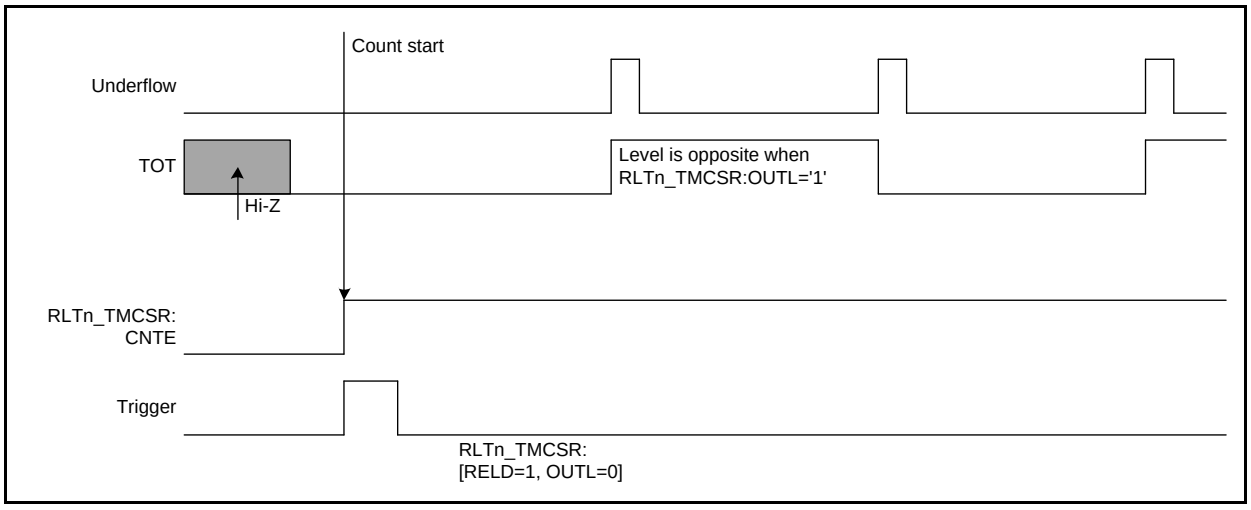
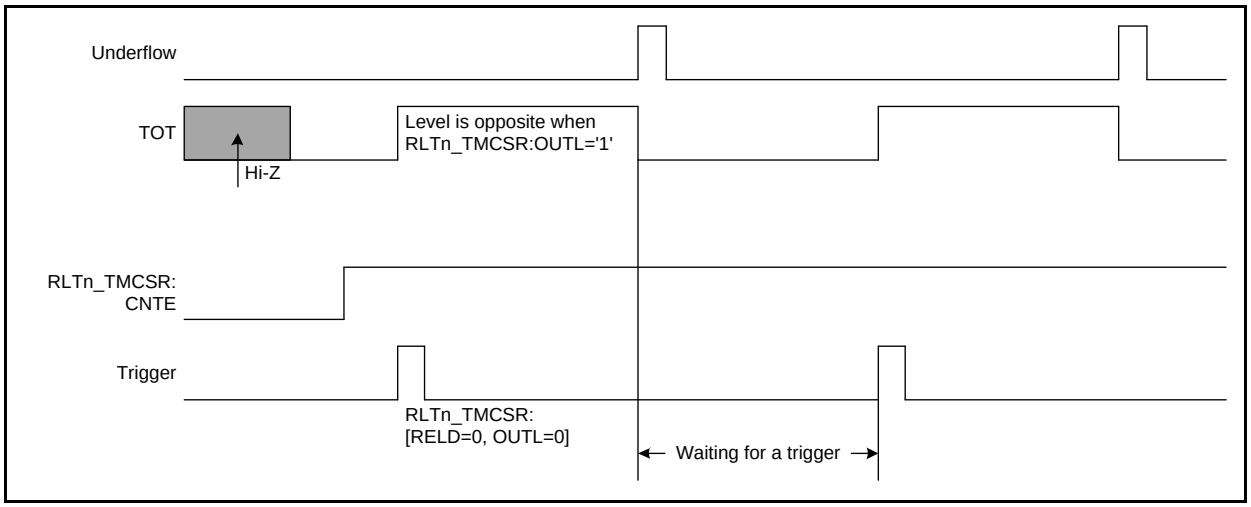


Figure 3-7 Output Signal Function of 32-bit Reload Timer in One-shot Mode



3.4. Counter Operation State

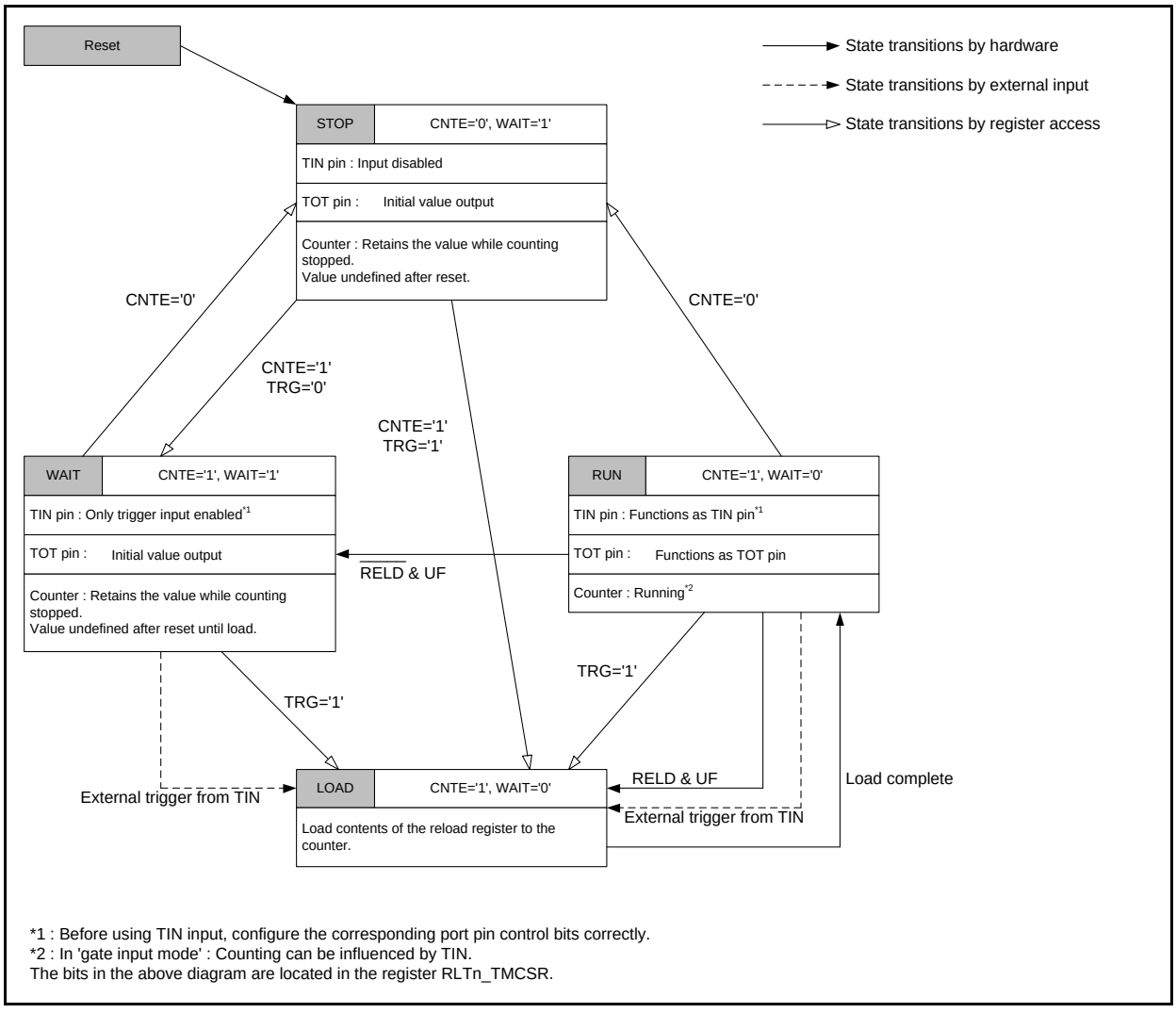
The counter state is determined by RLTn_TMCSR:CNTE bit in the Timer Control Status Register and the internal WAIT signal.

Available States for Reload Timer Are:

Stop state: RLTn_TMCSR:CNTE = 0 and WAIT = 1
Wait state: RLTn_TMCSR:CNTE = 1 and WAIT = 1
Run state: RLTn_TMCSR:CNTE = 1 and WAIT = 0

Counter Operation States

Figure 3-8 Counter Operation States



*1 : Before using TIN input, configure the corresponding port pin control bits correctly.
*2 : In 'gate input mode' : Counting can be influenced by TIN.
The bits in the above diagram are located in the register RLTn_TMCSR.

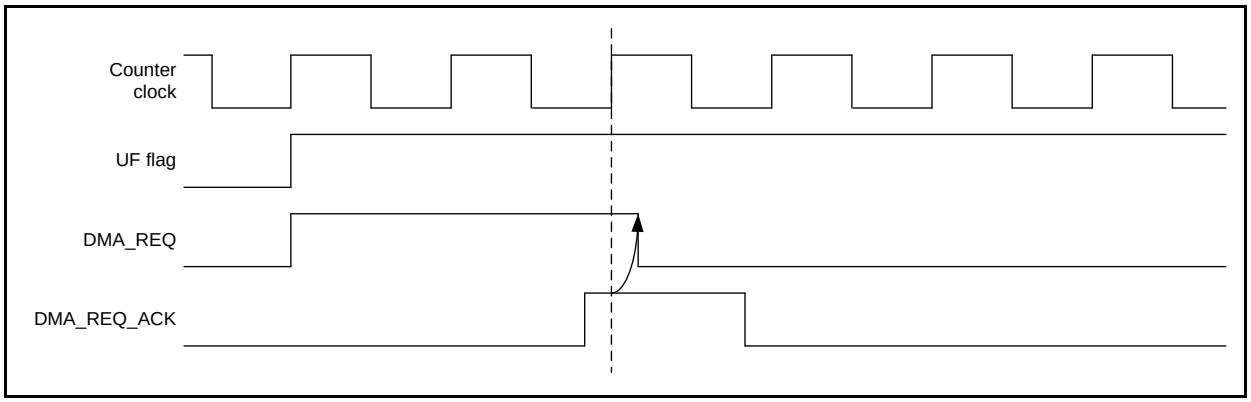
3.5. DMA Operation

The DMA support is determined by the RLTn_DMACFG:ENDMAUF bit. Setting this bit enables DMA request generation for DMA. Assertion of DMA_REQ_ACK signal acknowledges request and hence DMA_REQ signal gets de-asserted.

Enabling DMA Support

Writing "1" to RLTn_DMACFG:ENDMAUF enables DMA request generation for DMA when RLTn_TMCSR:UF bit sets. However, writing "0" to RLTn_DMACFG:ENDMAUF disables DMA request generation even if RLTn_TMCSR:UF bit sets. When DMA_REQ_ACK is asserted the DMA_REQ signal gets de-asserted by acknowledging the DMA request. Figure 3-9 shows behavior of DMA_REQ_ACK when asserted.

Figure 3-9 De-Asserting DMA_REQ Signal



4. Registers

This section describes the registers of 32-bit Reload Timer.

Table 4-1 Memory Layout of 32-bit Reload Timer Registers

Offset	+3	+2	+1	+0
0x00000000	RLTn_DMACFG 00000000_00000000_00000000_00000000			
0x00000004	Reserved			
0x00000008	RLTn_TMCSR 00000000_00000000_00000000_00000000			
0x0000000C	Reserved			
0x00000010	RLTn_TMRLR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			
0x00000014	RLTn_TMR XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX			

Note:

- *The initial register value after reset indicates as follows:*
  - "1": Initial value "1"
  - "0": Initial value "0"
  - “X”: Initial value undefined
  - “-“: Reserved bit/Undefined bit
  - “*”: Initial value "0" or "1" according to the setting

4.1. DMA Configuration Register (RLTn_DMACFG)

The DMA Configuration Register controls the DMA request generation for underflow condition.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ENDMAUF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:1] Reserved

[bit0] ENDMAUF : DMA enable for underflow (ENDMAUF)

Bit	Description
0	No DMA request is generated
1	A DMA request is generated when the counter, RLTn_TMR, underflows

4.2. Timer Control Status Register (RLTn_TMCSR)

The Timer Control Status Register controls the operation mode and interrupt of the 32-bit Reload Timer.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CNTE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	TRG	UFCLR	UF
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MOD[2]	MOD[1]	MOD[0]	CSL2	CSL1	CSL0	Reserved	NFE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DBGE	Reserved	OUTL	RELD	INTE	Reserved	Reserved	Reserved
ACCESS_TYPE	R/W	R0,W0	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:25] Reserved

[bit24] CNTE: Count enable

The Count Enable (CNTE) bit is a timer count enable bit.

Bit	Description
0	Stops count operation
1	Sets the timer to wait for a trigger

[bit23:19] Reserved

[bit18] TRG: Trigger

Software trigger bit.

Bit	Description
0	No effect
1	Applies a software trigger, causing the timer to load the reload register content to the counter and starts counting

Notes:

- *Applying a trigger using this register is only valid when RLTn_TMCSR:CNTE = 1. If RLTn_TMCSR:CNTE = 0, writing "1" to TRG has no effect.*
- *Set this bit in 'gate input mode' and in 'event count mode' to load the reload register content before counting starts.*

[bit17] UFCLR: Underflow interrupt clear

Bit	Description
0	No effect
1	Clears RLTn_TMCSR:UF bit

[bit16] UF: Underflow

The Underflow (UF) is timer interrupt request flag.

Bit	Description
0	No underflow occurred
1	When an underflow occurred

Note:

- *UF bit is cleared by writing "1" to RLTn_TMCSR:UFCLR bit.*

[bit15:13] MOD: Operation mode

The Operation Mode (MOD[2:0]) bits set the operation mode and input (TIN) functions.
Table 4-2 and Table 4-3 list the MOD[2:0] bit settings.

[bit12] CSL2: Clock select 2

The Clock Select 2 (CSL2) bit specifies the clock/event source and the clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit11] CSL1: Clock select 1

The Clock Select 1 (CSL1) bit specifies the clock/event source and the clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit10] CSL0: Clock select 0

The Clock Select 0 (CSL0) bit specifies the count clock division ratio.
Table 4-4 lists the selected clock sources for different CSL0/1/2 settings.

[bit9] Reserved

[bit8] NFE: Noise filter enable

This bit is used to enable/disable the noise filter for the TIN input.

Bit	Description
0	Noise filter for TIN is disabled
1	Noise filter for TIN is enabled

[bit7] DBGE: Debug mode enable

This bit is used to enable/disable debug mode for RLT.

Bit	Description
0	Debug mode disabled
1	Debug mode enabled

Note:

When DBGE is set to "1" and the processor is in debug state, the timer counter operation is paused, and writing to the RLTn_TMRLR register directly updates the timer counter (RLTn_TMR register). When the processor leaves debug state or DBGE is set to "0", the timer counter operation is resumed.

[bit6] Reserved

[bit5] OUTL: Output level

This Output Level (OUTL) bit sets the output level for the TOT.
See Table 4-5.

[bit4] RELD: Reload

This Reload (RELD) bit enables reload operations.
See Table 4-5.

Bit	Description
0	The timer operates in one-shot mode. In this mode, the count operation stops when an underflow occurs due to the counter value changing from 0x00000000 to 0xFFFFFFFF
1	The timer operates in reload mode. In this mode, the timer loads the reload register contents into the counter and continues counting whenever an underflow occurs

[bit3] INTE: Interrupt enable

Timer interrupt request enable bit.

Bit	Description
0	No interrupt request is generated even when the RLTn_TMCSR:UF bit changes to "1"
1	An interrupt request is generated when the RLTn_TMCSR:UF bit changes to "1"

[bit2:0] Reserved

Table 4-2 RLTn_TMCSR:MOD[2:0] Bit Settings for Internal Clock Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b00, 0b01, or 0b10)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
0	0	0	Trigger disabled	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edge
1	x	0	Gate input	"L" level
1	x	1		"H" level

Table 4-3 RLTn_TMCSR:MOD[2:0] Bit Settings for Event Counter Mode (RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 = 0b11)

MOD[2]	MOD[1]	MOD[0]	Input Function	Active Edge or Level
x	0	0	-	-
	0	1	Event input	Rising edge
	1	0		Falling edge
	1	1		Both edge

Table 4-4 Clock Sources for RLTn_TMCSR:CSL0 / RLTn_TMCSR:CSL1 / RLTn_TMCSR:CSL2 Bit Settings

CSL2	CSL1	CSL0	Count Clock (Time for Peripheral Clock)
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
1	0	1	1/32
1	1	0	External event count mode, each event on TIN
1	1	1	External event count mode, each second event on TIN

Table 4-5 RLTn_TMCSR: RLTn_TMCSR:OUTL, and RLTn_TMCSR:RELD Settings

OUTL	RELD	Output Waveform
0	0	Output an "H" level pulse during counting.
1	0	Output an "L" level pulse during counting.
0	1	Toggle output. Starts with "L" level output. Changes level on timer reload.
1	1	Toggle output. Starts with "H" level output. Changes level on timer reload.

Note:

- *Bits marked 'x' in the table can be set to any value.*

4.3. 32-bit Reload Register (RLTn_TMRLR)

The Timer Reload Register holds the reload value of the 32-bit Reload Timer.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMRLR[31:24]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMRLR[23:16]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMRLR[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMRLR[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMRLR: Timer reload register

The Timer Reload Register (RLTn_TMRLR) is a 32-bit reload register holds the reload value. Initial value is undefined.
This register can be accessed only in 32-bit/64-bit mode.
When RLTn_TMCSR:DBGE is set to "1" and the processor is in debug state, writing to this register updates the timer counter immediately.

Note:

- *This register is not initialized by Hard Reset (i.e. “X”).*

4.4. 32-bit Timer Register (RLTn_TMR)

Reading this register returns the count value of the 32-bit Reload Timer. The initial value is undefined.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TMR[31:24]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TMR[23:16]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TMR[15:8]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TMR[7:0]							
ACCESS_TYPE	R,WX							
PROT_TYPE	-							
INITIAL_VALUE	XXXXXXXX							

[bit31:0] TMR: Timer register

Reading this Timer Register (RLTn_TMR) returns the count value of the 32-bit Reload Timer. Initial value is undefined. This register can be accessed only in 32-bit/64-bit mode.

Note:

- *This register is not initialized by Hard Reset (i.e. "X").*

CHAPTER 22: 12/10/8-Bit Analog to Digital Converter

This chapter explains the functions and operations of the 12/10/8-bit A/D Converter.

1. Overview
2. Configuration and Block Diagram
3. Operation of A/D converter
4. Setup procedure examples
5. Registers

CODE: FIP022-E03.0

1. Overview

The A/D Converter converts analog input voltages into digital values. The A/D Converter features eight range comparators, 64 pulse detection units, 64 separate conversion data registers and four multiple conversion channels.

Features of the A/D Converter

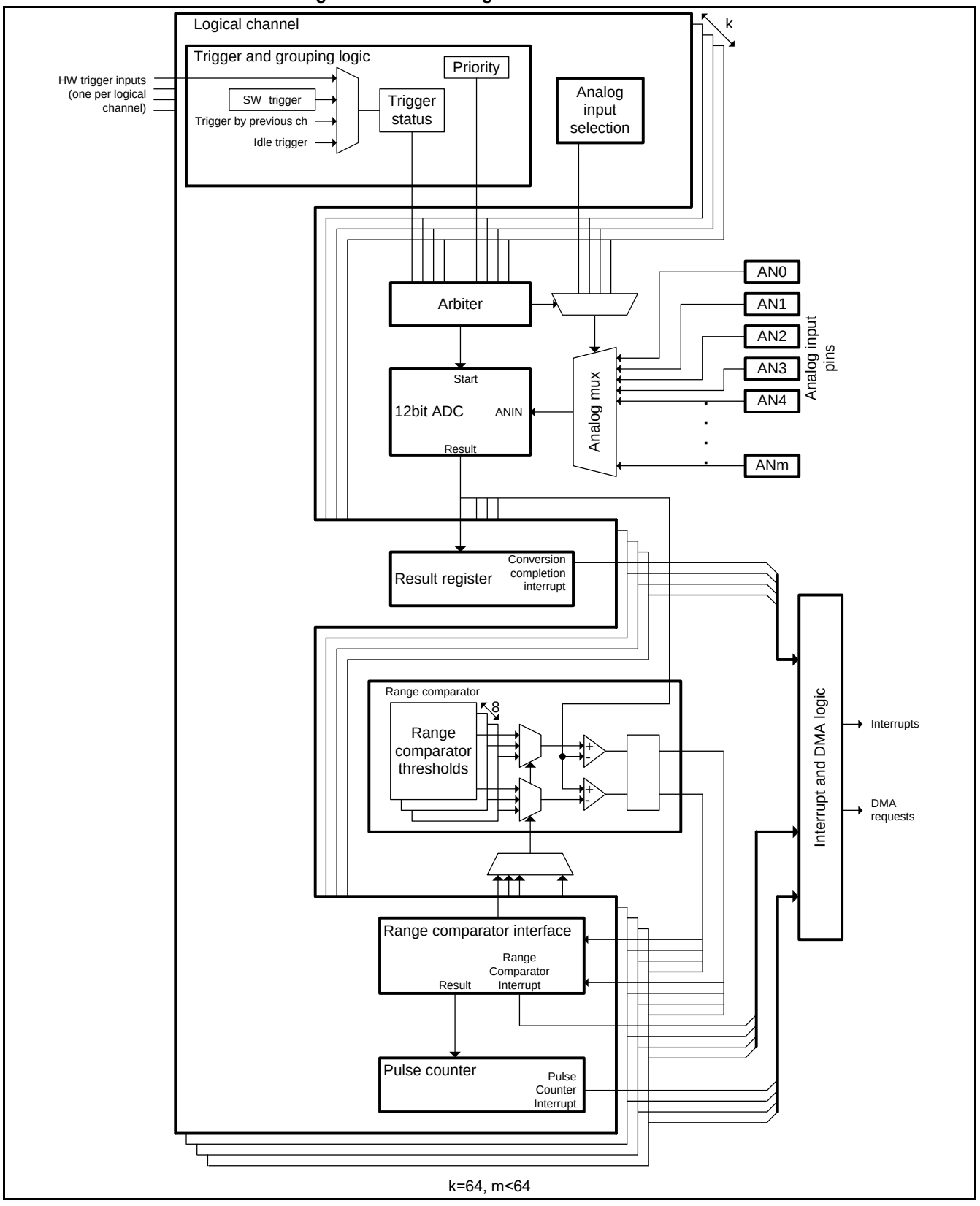
- Conversion time: Refer to the datasheet
- 12/10/8-bit resolution
- RC type successive approximation conversion with sample & hold circuit
- Programmable configuration of 64 logical channels by mapping them to analog inputs
  - Each logical channel is mapped to exactly one analog input
  - Several logical channels can be mapped to the same analog input
- 64 dedicated logical channel conversion data registers
- Programmable selection of one of the four global sampling times for each logical channel
- Conversion request for each logical channel i is possible by software trigger only, hardware or software trigger, trigger by conversion completion of the preceding logical channel i-1 and idle trigger
- All active conversion requests are compared according to the corresponding logical channel priorities (selectable range is 0-15, 0 is the highest and 15 is the lowest priority) and A/D conversion is started for the logical channel with the highest priority
- When the higher priority conversion requests during an active A/D conversion, it is possible to select the following behavior
  - An active A/D conversion can be interrupted, and the higher priority conversion starts after interrupt operation
  - An active A/D conversion cannot be interrupted, hence the wait time between the higher priority conversion request and its conversion start can be up to the maximum A/D conversion period. In case of multiple conversion channel processing that are configured to be non-interruptible, the wait time can increase to the time needed to the maximum multiple channel conversion period
- The consecutive logical channels can be configured as a group, where the start channel conversion request is set to software trigger only, hardware or software trigger or idle trigger while all other channel conversion requests are set to the preceding logical channel conversion completion
- If a group conversion is interrupted by a higher priority conversion request, after the higher priority request is processed the following group configurations are possible:
  - Resume with the next logical channel in the group
  - Restart with the group start channel or the last converted channel configured as "resume" channel within a group
  - Stop the group processing until next start channel conversion request is issued
- First four logical channels can be configured as:
  - Multiple conversion channels offering the possibility to perform 1 to 16 conversions of the same logical channel and accumulate the result
  - A/D Converter calibration channels providing the configuration for conversion of A/D Converter reference voltages in order to calculate the offset and gain corrections
- Eight range comparators are selectable for every logical channel, comparing the full range (12-bit)/upper 8-bit of the conversion result
- Programmable upper and lower thresholds for each range comparator
- The comparison results will set flags per logical channel, depending on the configuration. Possible configurations are:
  - “Outside range”: The flags are set if the A/D result is below the lower OR above the upper threshold
  - “Inside range”: The flags are set if the A/D result is above the lower AND below the upper threshold
- The results of the range comparator can be filtered to ignore short spikes
- During an active A/D conversion, it can be forced stop by software
- It is possible to select the operation after A/D conversion finished (and next conversion is not requested)

  - A/D converter goes to idle (power-down) state after A/D conversion finished
  - A/D converter does not go idle (power-down) state, it can start A/D sampling without the resumption time
- Interrupt request generation to CPU is provided for:
  - Conversion done interrupt (end of a logical channel conversion)
  - Group interrupted interrupt (a group processing is interrupted just before the logical channel conversion is to be started or in case of multiple conversion channel before the last conversion is started)
  - Range comparison interrupt
  - Pulse detection interrupt
- Conversion done interrupts of up to four logical channels can trigger DMA requests to transfer the A/D conversion results to memory. The conversion done interrupts of the group last logical channels are good candidates for DMA burst setup, since the group result registers can be read linearly
- Debug mode provides the possibility to freeze further A/D conversion processing at the end of the current conversion

2. Configuration and Block Diagram

This section shows a block diagram of A/D converter.

Figure 2-1 Block Diagram of A/D Converter



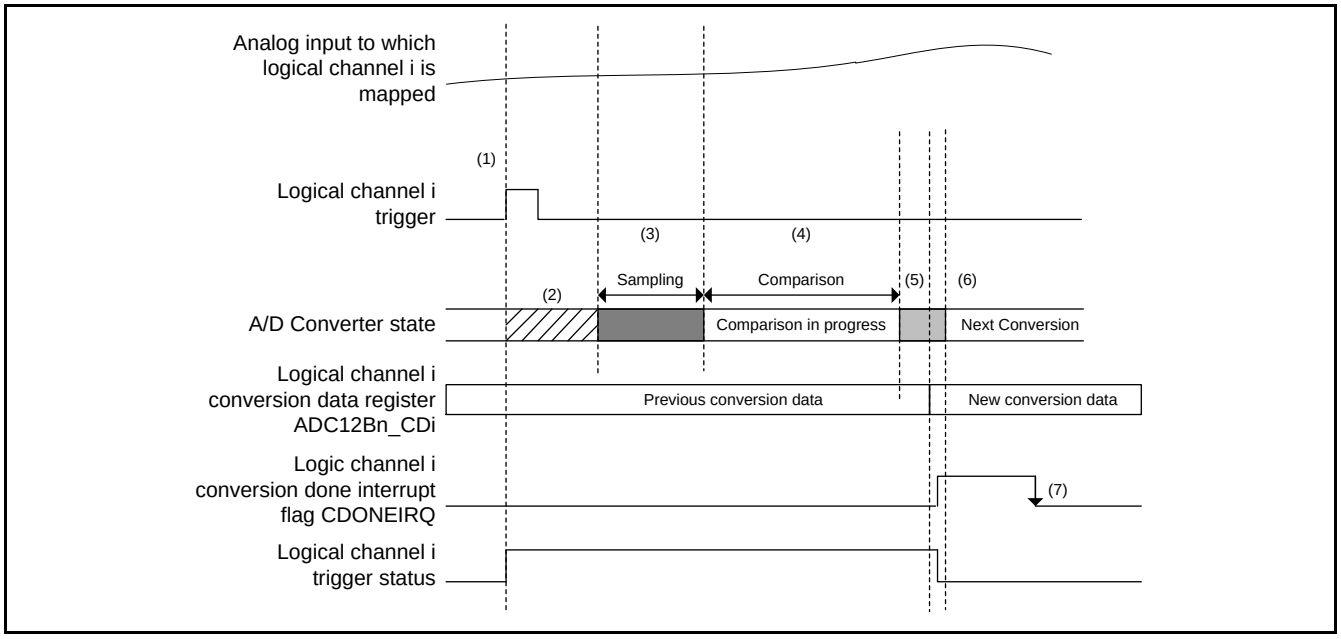
3. Operation of A/D Converter

A/D Converter operates using the successive approximation method with 12-bit or 10-bit or 8-bit resolution. There is one A/D Conversion Data Register per logical channel which is updated each time the assigned logical channel is converted. First four logical channels can be configured as multiple conversion channels or used for A/D Converter calibration.
The range comparator compares the converted values with the configured values in the threshold registers and accordingly generates an interrupt for "inside range" or "outside range", depending on the configuration. Any of eight range comparators can be configured for any of 64 logical channels.
A/D Pulse Detection function detects events of desired length and also filters parasitic inverted events. Each logical channel has a dedicated pulse detection function.

3.1. A/D Conversion Flow

The basic A/D conversion flow is shown on Figure 3-1.

Figure 3-1 A/D Conversion Flow



The important steps marked on the figure are:

(1) A/D conversion request (trigger) is issued for particular logical channel i and the corresponding trigger status is set.

(2) Period between the trigger and actual conversion start depends on current A/D Converter state condition

- If A/D Converter is in idle (power-down) state (power-down mode is enabled (ADC12Bn_CTRL.PDDMD = "0")) i.e. there are no trigger status set at the moment the trigger is issued, it is first waited until the resumption time (configured as A/D converter resumption time register (ADC12Bn_RT)) elapses. At the end of resumption time, the priority arbiter compares the priorities of all logical channels with set trigger status and inactive data protection feature. The conversion of the logical channel i start when this channel wins the arbitration, i.e. after all logical channels with higher priority are converted.

- If A/D converter is active at the moment the trigger is issued, the operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).
  - Forced stop is enabled (ADC12Bn_CTRL.FSMD = "1"): Even if the sampling phase is started, an active A/D conversion can be interrupted. So, the logical channel i starts after interrupt operations instead of currently performed conversion when this channel wins the priority arbitration.
  - Forced stop is disabled (ADC12Bn_CTRL.FSMD = "0"): The conversion of the logical channel i starts when this channel wins the priority arbitration and currently preformed conversion is finished, since an A/D conversion cannot be interrupted once its sampling phase is started. Accordingly, even if the logical channel i has the highest priority, in worst case it can happen that its conversion is delayed by the maximum configured time needed to convert one logical channel (including the case of multiple conversion channel that cannot be interrupted).

(3) Sampling period. A/D converter internal input signal level goes to the level of the analog input to which the logical channel i is mapped.

(4) A/D Comparison.

(5) A/D conversion finalization period. The conversion result is stored in the dedicated conversion data register ADC12Bn_CDi, the conversion done interrupt flag is set and the trigger status is cleared.

(6) Started next conversion of the logical channel with the highest priority at the moment. If there is no trigger for any logical channel, the operation is dependent on the setting of power-down disable mode (ADC12Bn_CTRL.PDDMD).

- Power down disable mode (ADC12Bn_CTRL.PDDMD = "1"): Even if the A/D conversion is finished (and next conversion is not requested), A/D converter does not go idle (power-down) state. It can start A/D sampling without A/D converter resumption time.
- Power down enable mode (ADC12Bn_CTRL.PDDMD = "0"): A/D converter goes to idle (power-down) state after A/D conversion finished (and next conversion is not requested).

If the trigger status of any logical channel is already set, next conversion is started without A/D converter resumption time.

(7) After the conversion data register (ADC12Bn_CDi) is read or by writing "1" to the corresponding conversion done interrupt clear bit (ADC12Bn_CDONEIRQC0~1.CDONEIRQCi), the conversion done interrupt flag is cleared.

In case A/D Converter reconfiguration takes place during operation (A/D Converter is not in power-down state), the following rules are to be obeyed:

- Do not reconfigure logical channels belonging to the group that is currently converted.
- Set the trigger types (ADC12Bn_CHCTRL0~63.TRGTYP[1:0]) of the logical channels affected by the reconfiguration to software trigger only ("00" setting).
- Clear all the trigger status flags (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST) of the affected logical channels, by writing "1" to the corresponding bits of ADC12Bn_TRGCL0~1 (or ADC12Bn_CHCTRL0~63.TRGCL) registers.
- Reconfigure the logical channels.

3.2. Logical Channel Mapping to Analog Input Signals

64 logical channels are mapped to analog input signals that need to be converted into the digital values. The mapping is done over ADC12Bn_CHCTRL0~63.ANIN configuration fields in the way that ANIN value controls the number of the analog input signal AN, that is propagated to the A/D Converter input in case the logical channel is converted.

Figure 3-2 illustrates an example of possible logical channel mapping to analog inputs. Each logical channel is mapped to exactly one analog input and it is allowed to map several logical channels to the same analog input (logical channels 4, 5, 6 and 62 are mapped to the analog input AN8).

Figure 3-2 Example of Logical Channel Mapping to Analog Inputs

Example of logical channel mapping to analog inputs

Logical channel number	0	1	2	3	4	5	6	7	…	58	59	60	61	62	63
ANIN bit field setting	52	14	63	0	8	8	8	32	…	1	2	3	28	8	4
Analog input signal	AN52	AN14	AN63	AN0	AN8	AN8	AN8	AN32	…	AN1	AN2	AN3	AN28	AN8	AN4

Example of logical channel mapping to analog inputs

Logical channel number	0	1	2	3	4	5	6	7	…	26	27	28	29	30	31
ANIN bit field setting	30	14	31	0	8	8	8	16	…	1	2	3	28	8	4
Analog input signal	AN30	AN14	AN31	AN0	AN8	AN8	AN8	AN16	…	AN1	AN2	AN3	AN28	AN8	AN4

The described way of logical channel organization provides following benefits:

- Configurable grouping and consecutive conversion of channels independent on physical pin order.
- Mapping of several logical channels to the same analog input allows repetitive conversion of the same analog input with only one interrupt at the end.

3.3. Logical Channel Data Protection Function

Every logical channel has its own conversion data register ADC12Bn_CD. They are written by hardware at the end of conversion of the dedicated channel. The CPU can read the data registers any time. If a conversion is finished and the data of the previous conversion of the same channel has not been read out, previous data could be overwritten and previous conversion result lost. To avoid this, the data protection function can be enabled so that the next conversion of this logical channel is not started until the previous data has been read out. The data protection function is controlled by the corresponding ADC12Bn_CHCTRL0~63.DP bits:

- If for some logical channel the dedicated ADC12Bn_CHCTRLi.DP bit is equal to "0", then conversion is continued and former conversion data are overwritten.
- In case of a regular logical channel, if the dedicated ADC12Bn_CHCTRLi.DP bit is equal to "1" and conversion done interrupt flag is set to "1", this logical channel cannot be selected for the conversion even though its trigger status may be set. The channel can be selected for conversion again after its conversion done interrupt flag has been cleared, e.g. by reading the conversion data register.
- In case of a multiple conversion logical channel, the channel cannot be selected for conversion until its conversion done interrupt flag or group interrupted interrupt flag have been cleared.
  - if the dedicated DP bit is equal to "1" and conversion done interrupt flag is set to "1"
  - if the dedicated DP bit is equal to "1", group interrupted interrupt flag is set to "1", group interrupted interrupt is enabled and multiple conversions are already started

Note:

1. *It should set DP to "1" only for the logical channels configured as a group (for details about group configuration please see section "Group processing" in chapter "3. Operation of A/D Converter") having all its logical channels set to "Resume" (ADC12Bn_CHCTRL0~63.RSMRST = "01"). In case of other group configurations:*

- *"Restart" setting (ADC12Bn_CHCTRL0~63.RSMRST = "10") - the group may be restarted before the conversion done interrupt of the last channel in the group (meaning the interrupt that indicates that the conversion of the entire group has been completed) is asserted, so the CPU does not know that there is already a result available in this channel.*
- *"Stop" setting (ADC12Bn_CHCTRL0~63.RSMRST = "00") - with the group interrupted interrupt enabled, it could be possible for the CPU to clear the conversion done interrupt flags of the affected channels, but the use case of both "Stop" and "Restart" is that the results of an interrupted group are not interesting, so they do not need to be protected.*

2. *For groups consisting of only one multiple conversion channel, it should use DP as follows:*
   - *"Resume" setting (ADC12Bn_CHCTRL0~63.RSMRST = "01") - when setting DP to "1", disable the group interrupted interrupt (set the corresponding ADC12Bn_GRPIRQE0.GRPIRQE0~3 to "0"): The conversion result will be protected only after all conversions of the multiple conversion channel have been performed.*
     If the corresponding ADC12Bn_GRPIRQE0.GRPIRQE0~3 is set to "1", then the conversion result will be protected at the time the channel is interrupted, and the conversions will not be resumed before the group interrupted interrupt flag has been cleared. When the conversion is resumed, the conversion counter is reset and first conversion is started. It means the multiple conversion cannot resume from the interrupted state.
   - *"Restart" setting (ADC12Bn_CHCTRL0~63.RSMRST = "10") - when setting DP to "1", disable the group interrupted interrupt (set the corresponding ADC12Bn_GRPIRQE0.GRPIRQE0~3 to "0"): The conversion result will be protected only after all conversions of the multiple conversion channel have been performed. Even if the corresponding ADC12Bn_GRPIRQE0.GRPIRQE0~3 is set to "1", the conversion result will be protected only after all conversions of the multiple conversion channel have been performed.*
   - *"Stop" setting (ADC12Bn_CHCTRL0~63.RSMRST = "00") - when setting DP to "1", enable the group interrupted interrupt (set the corresponding ADC12Bn_GRPIRQE0.GRPIRQE0~3 to "1"): The conversion result will be protected when the channel is interrupted or when all conversions have been completed.*

3.4. Logical Channel Triggering and Priority

Each logical channel has the configuration bit field ADC12Bn_CHCTRL0~63.TRGTYP[1:0], controlling the way a conversion request (trigger) can be issued. The following settings are possible:

- Software trigger only (TRGTYP = "00") - in order to set the dedicated trigger status flags (ADC12Bn_CHSTAT0~63.TRGST and ADC12Bn_TRGST0~1.TRGST) the corresponding software trigger bit (ADC12Bn_CHCTRL0~63.SWTRG) must be set to "1".
- Hardware or software trigger (TRGTYP = "01") - the corresponding trigger status flags can be set not only through software trigger bit, but also in case of the rising edge event on dedicated hardware trigger input.
- Trigger by completion of the preceding logical channel (TRGTYP = "10") - the trigger status flags are set only at the end of preceding logical channel conversion, i.e. updating of conversion data register ADC12Bn_CD of the preceding channel triggers this channel if the trigger status flags of the preceding channel are still set. This allows combining several consecutive logical channels into a group that will be sequentially converted after the start channel is triggered. To configure a group, the start channel trigger type is set to software trigger only, hardware or software trigger or idle trigger, while all other channel trigger types are set to the preceding logical channel conversion completion. It does not make sense to configure the first logical channel to trigger type 2, since the channels would never be triggered.
- Idle trigger (TRGTYP = "11") - the trigger status flags are set whenever there is no active conversion request, i.e. there is no logical channel having trigger status flag set and inactive data protection function. Accordingly, trigger status flags of all

logical channels with idle trigger type are set at the same time. The further processing depends on their priority. The idle trigger type allows to form a regular group (the start channel is set to idle trigger and consecutive channels have preceding channel completion trigger type).

Once the logical channel trigger status is set to "1", the channel priority (bit fields ADC12Bn_CHCTRL0~63.CHPRI) configures the conversion order of logical channels with the inactive data protection function. In case of the same priority, active trigger status and inactive data protection function, the logical channel having lower number wins the priority arbitration. The priority can be set in the range 0 - 15. The setting 0 (CHPRI = "0000") is the highest priority, whereas the lowest possible priority setting is 15 (CHPRI = "1111").

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"):

When the higher priority conversion was requested during an active A/D conversion, the operation of interrupt is dependent on the setting of the channel priority (ADC12Bn_CHCTRL0~63.CHPRI) only.

- If the priority value (CHPRI) of the requested channel is lower than active channel, it can be interrupted and the conversion of the channel with the higher priority is started.
- If the priority value (CHPRI) is higher or same, it cannot be interrupted.

Figure 3-3 Example of Logical Channel Priority Configuration

Logical channel number	0	1	2	3	4	5	6	7	…	58	59	60	61	62	63
CHPRI bit field setting	0	12	13	5	1	6	3	10	…	1	2	3	14	15	15
Data protection function	1	0	0	0	0	0	0	0	…	0	1	0	0	0	0
Trigger status flag	1	1	1	0	1	0	0	1	…	1	0	1	1	1	1

The example on Figure 3-3 would result in following priority arbitration and corresponding logical channel conversion sequence:

(1) Logical channel 4.

Logical channel 0 has higher priority, but its data protection function is active.

(2) Logical channel 58.

This logical channel has the same priority as logical channel 4, but it is converted as second since its number is higher.

(3) Logical channel 60.

Although the channel 59 has higher priority, its trigger status is not set and even the data protection is active. Moreover, logical channel 6 has the same priority and even lower number, but its trigger status is not set.

(4) Logical channel 7.

Even though the logical channels 3 and 5 have higher priority, their trigger status is not set.

(5) Logical channel 1.

(6) Logical channel 2.

(7) Logical channel 61.

(8) Logical channel 62.

It has lower number than the channel 63.

(9) Logical channel 63.

3.5. Group Processing

A group is defined by the trigger type configuration of consecutive logical channels. The first channel of the group has the trigger type set to software trigger only, hardware or software trigger or idle trigger. If the following channel trigger type is not set to preceding logical channel conversion completion, the group consists of only one channel. Otherwise, the group continues until the last channel in a row having trigger type set to preceding logical channel conversion completion. After the first channel of the group is triggered and converted, automatically the second channel is triggered and so on until the whole group is converted.

Figure 3-4 Group Configuration Examples

a)

Group	1			2		
Channel	0	1	2	3	4	5
Priority	7	8	9	1	2	3
Trigger type	1	2	2	1	2	2

b)

Group	1			2		
Channel	0	1	2	3	4	5
Priority	4	3	3	6	5	5
Trigger type	1	2	2	0	2	2

c)

Group	1			2	3		
Channel	0	1	2	3	4	5	6
Priority	0	2	2	0	15	14	14
Trigger type	1	2	2	0	3	2	2

d)

Group	x					
Channel	0	1	2	3	4	5
Priority	0	2	2	0	14	15
Trigger type	2	2	2	0	2	2

Figure 3-4 shows some possible group configurations:

a) In the example two groups are configured, the group 1 consists of logical channels 0, 1 and 2 and the group 2 includes logical channels 3, 4 and 5. If both groups are triggered (trigger status is set for the starting channels 0 and 3), the group 2 would be converted first due to higher priority setting. Since the first channel in the groups has the highest priority, the groups are "re-triggerable" (if the next starting channel trigger appears before the group conversion is finished, the conversion of the group would be restarted).

b) Here the first channel of the groups has a lower priority then the remaining channels within the groups, hence it is not possible to re-trigger the group processing. The processing of the group 2 can be started only by the software trigger of the logical channel 3. The group 1 allows the triggering by the software or hardware trigger of the channel 0.

c) This example contains three groups, where the group 2 consists of only one logical channel 3. The group 3 shows the configuration of the group starting with the idle trigger type, i.e. the trigger status of the logical channel 4 is set to "1" if there is no logical channel having trigger status flag set and inactive data protection function.

d) In this case first three logical channels are set to trigger type 2 and since there will be no event of preceding channel conversion completion, the logical channels 0, 1 and 2 will never be converted.

Between the different groups (the group 1, the group 2): the priority of all the logical channels should be configured as [the group 1 > the group 2] or [the group 1 < the group 2].

(Example: When the priorities of all the channels included in the group 1 are set as 7-9, all the channels of the group 2 should be set lower (or higher) than 7-9.)

After a group processing is started (at least the conversion of the first channel of the group is started), it can be interrupted.

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"): If the triggered channel is higher priority than the current channel in the group, current conversion of group processing is interrupted after interrupt operation and the conversion of the channel with the higher priority is started.

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"): If an A/D conversion is already started, it cannot be stopped. Hence if the triggered channel is higher priority than the next channel in the group, the group processing is interrupted before the conversion start of the next logical channel in the group.

The configuration bit fields ADC12Bn_CHCTRL0~63.RSMRST define how the group processing is to be continued after the higher priority conversion requests are executed. Note that RSMRST setting of the first channel in the group does not matter, since the group cannot be interrupted before the conversion of its first channel started.

If stop group setting (RSMRST[1:0] = "00") is configured for a logical channel and the group is interrupted just before conversion start of that channel, the current processing of the group is stopped. Consequently, the trigger status flag of the logical channel (set by the conversion completion of the preceding channel in the group) is cleared.

Figure 3-5 shows an example of stopping a group processing.

(1) Logic channel 0 trigger is issued and processing of the group 0 is initiated.

(2) Processing of the group 0, conversion of logical channels 0 -> 1 -> 2 -> start of channel 3 conversion.

(3) Logic channel 6 trigger is issued during the conversion of channel 3.

(4) The operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"):

Since the priority of the channel 6 is higher than priority of the channel 3, the processing of the group 0 is interrupted during the conversion of channel 3. After interrupted operation, processing continues with the conversion of the channel 6.

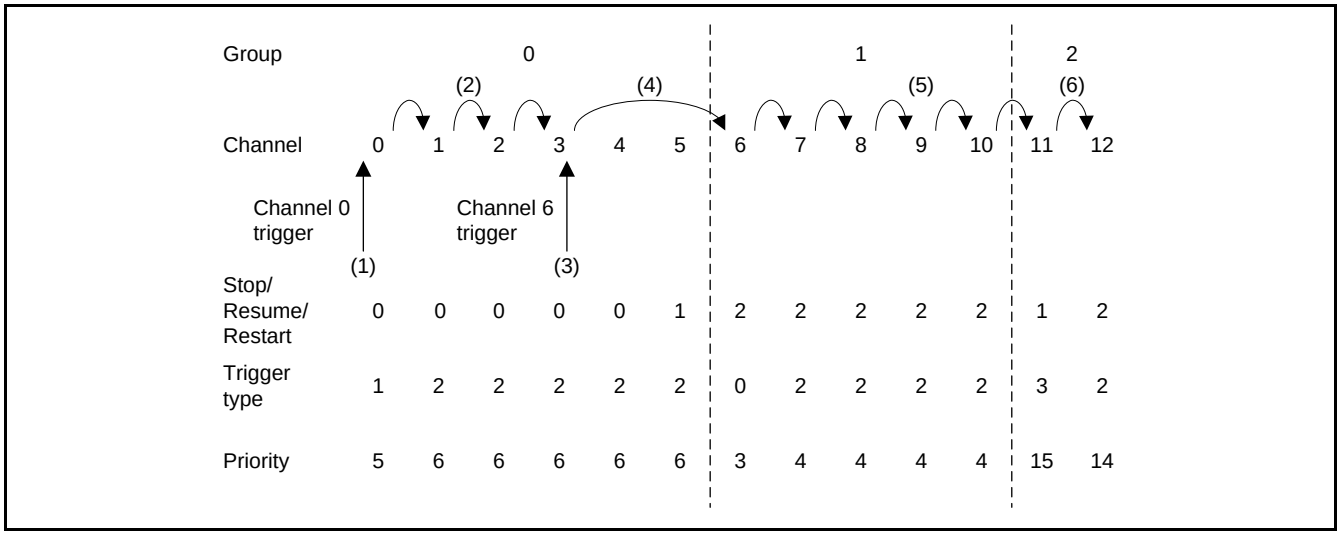
Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"):

Since the priority of the channel 6 is higher than priority of the channel 4, the processing of the group 0 is interrupted after the conversion of channel 3 is finished. Moreover, the trigger status flag of the channel 4 is cleared and processing continues with the conversion of the channel 6.

(5) Group 1 is converted.

(6) Idle trigger group 2 is converted. Group 0 will not be processed until next channel 0 trigger appears.

Figure 3-5 Stopping of the Group Processing



Resume group setting (RSMRST[1:0] = "01") for a logical channel means that if the group is interrupted, the current processing of the group will be resumed with this channel after higher priority conversions are done.

When the channel which is configured as resume (ADC12Bn_CHCTRL0~63.RSMRST = "01") is in interrupted operation, group interrupted interrupt flag (corresponding bits of ADC12Bn_CHSTAT0~63.GRPIRQ and ADC12Bn_GRPIRQ0~1.GRPIRQ) is set to "1" whenever converting of other channel occurred.

Figure 3-6 shows an example of resuming a group processing (Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1")).

Figure 3-7 shows an example of resuming a group processing (Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0")).

(1) Logic channel 0 trigger is issued and processing of the group 0 is initiated.

(2) Processing of the group 0, conversion of logical channels 0 -> 1 -> 2 -> start of channel 3 conversion.

(3) Logic channel 6 trigger is issued during the conversion of channel 3.

(4) The operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1", Figure 3-6):

Since the priority of the channel 6 is higher than priority of the channel 3, the processing of the group 0 is interrupted during the conversion of channel 3.

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0", Figure 3-7):

Since the priority of the channel 6 is higher than priority of the channel 4, the processing of the group 0 is interrupted after the conversion of channel 3 is finished.

(5) Group 1 is converted.

(6) The operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1", Figure 3-6):

After group 1 is finished, processing of group 0 is continued with conversion of channel 3 since its trigger status flag is still set.

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0", Figure 3-7):

After group 1 is finished, processing of group 0 is continued with conversion of channel 4 since its trigger status flag is still set.

(7) At the end of group 0 processing, idle trigger group 2 is converted.

Figure 3-6 Resuming of the Group Processing (Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"))

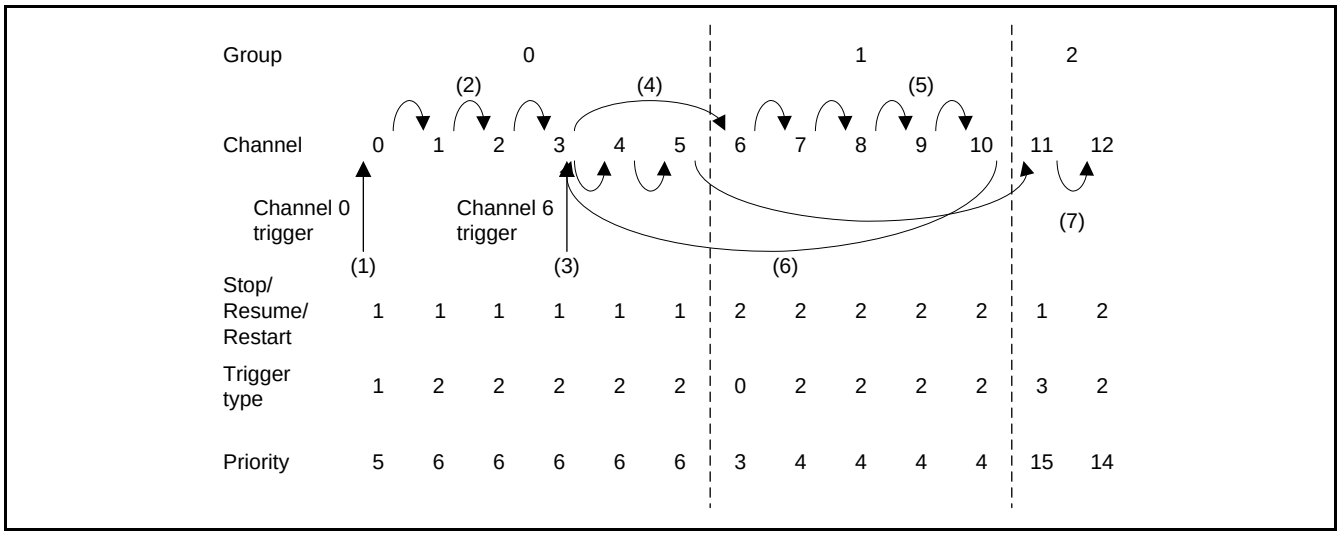
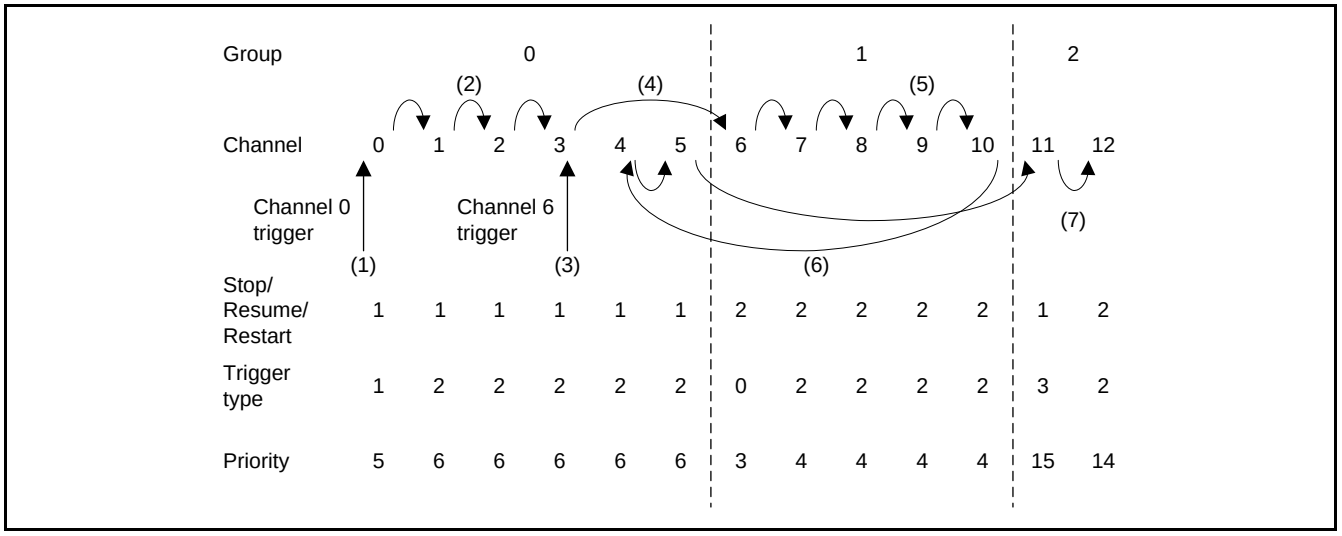


Figure 3-7 Resuming of the Group Processing (Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"))



Restart group setting (RSMRST[1:0] = "10") for a logical channel configures that if the group is interrupted, its trigger status is cleared and the processing of the group will be restarted after higher priority conversions are done:

with the closest previous channel of the group set to resume (RSMRST = "1"), accordingly the trigger status flag of that channel is set

with the first channel of the group (consequently, the trigger status flag of the first channel is set) if there are no previous channels set to resume.

Figure 3-8 shows an example of restarting a group processing with its first channel.

(1) Logic channel 0 trigger is issued and processing of the group 0 is initiated.

(2) Processing of the group 0, conversion of logical channels 0 -> 1 -> 2 -> start of channel 3 conversion.

(3) Logic channel 6 trigger is issued during the conversion of channel 3.

(4) The operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"):

Since the priority of the channel 6 is higher than priority of the channel 3, the processing of the group 0 is interrupted during the conversion of channel 3 and trigger status of channel 0 is set.

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"):

Since the priority of the channel 6 is higher than priority of the channel 4, the processing of the group 0 is interrupted after the conversion of channel 3 is finished and trigger status of channel 0 is set.

(5) Group 1 is converted.

(6) After group 1 is finished, processing of group 0 is restarted with conversion of channel 0 since its trigger status flag is set to "1".

(7) Group 0 is converted.

(8) At the end of group 0 processing, idle trigger group 2 is converted.

Figure 3-8 Restarting of the Whole Group Processing

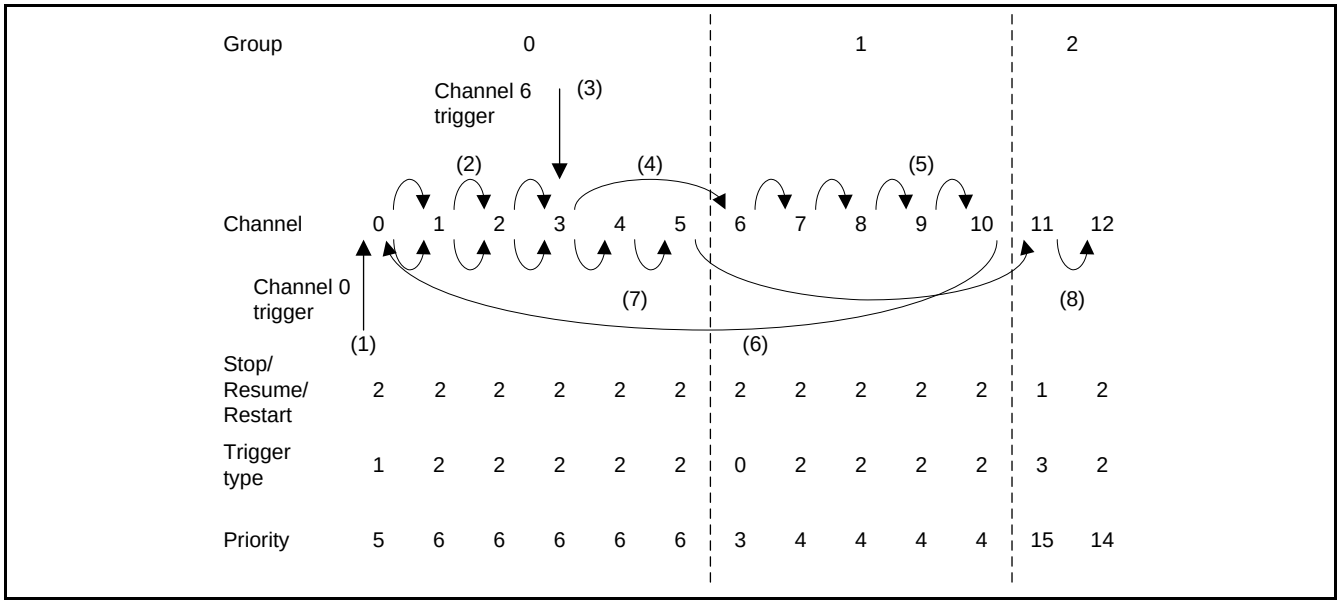


Figure 3-9 shows an example of restarting a group processing with its first channel during the conversion of the last channel of the group processing (Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1")).

Figure 3-10 shows an example of restarting a group processing with its first channel during the conversion of the last channel of the group processing (Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0")).

(1) Logic channel 0 trigger is issued and processing of the group 0 is initiated.

(2) Processing of the group 0, conversion of logical channels 0 -> 1 -> 2 -> 3 -> 4 -> start of channel 5 conversions (last channel of the group 0).

(3) Logic channel 6 trigger is issued during the conversion of channel 5.

The operations after (3) are dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1", Figure 3-9):

(4) Since the priority of the channel 6 is higher than priority of the channel 5, the processing of the group 0 is interrupted during the conversion of channel 5 and trigger status of channel 0 is set.

(5) Group 1 is converted.

(6) After group 1 is finished, processing of group 0 is restarted with conversion of channel 0 since its trigger status flag is set to "1".

(7) Group 0 is converted.

(8) At the end of group 0 processing, idle trigger group 2 is converted.

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0", Figure 3-10):

(4) After the conversion of channel 5 is finished (group 0 is converted), processing continues with the conversion of the channel 6.

(5) Group 1 is converted.

(6) At the end of group 1 processing, idle trigger group 2 is converted (group 0 is not restarted).

Figure 3-9 Restarting a Group Processing with Its First Channel during the Conversion of the Last Channel of the Group Processing (Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"))

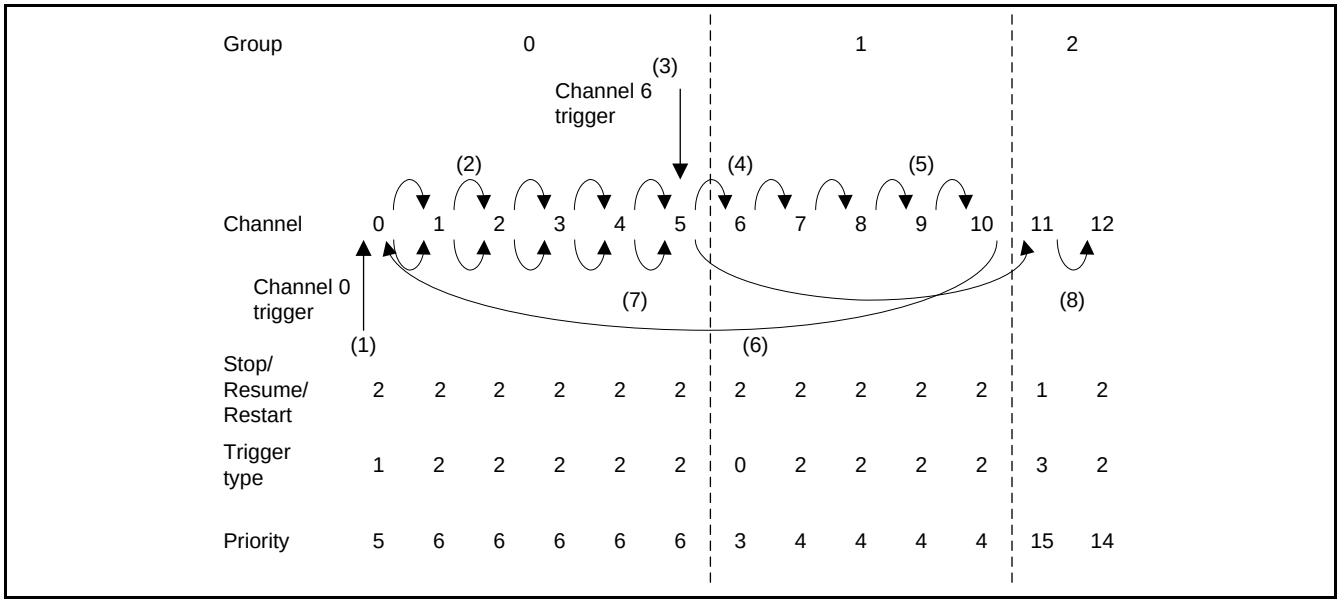


Figure 3-10 Restarting a Group Processing with Its First Channel during the Conversion of the Last Channel of the Group Processing (Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"))

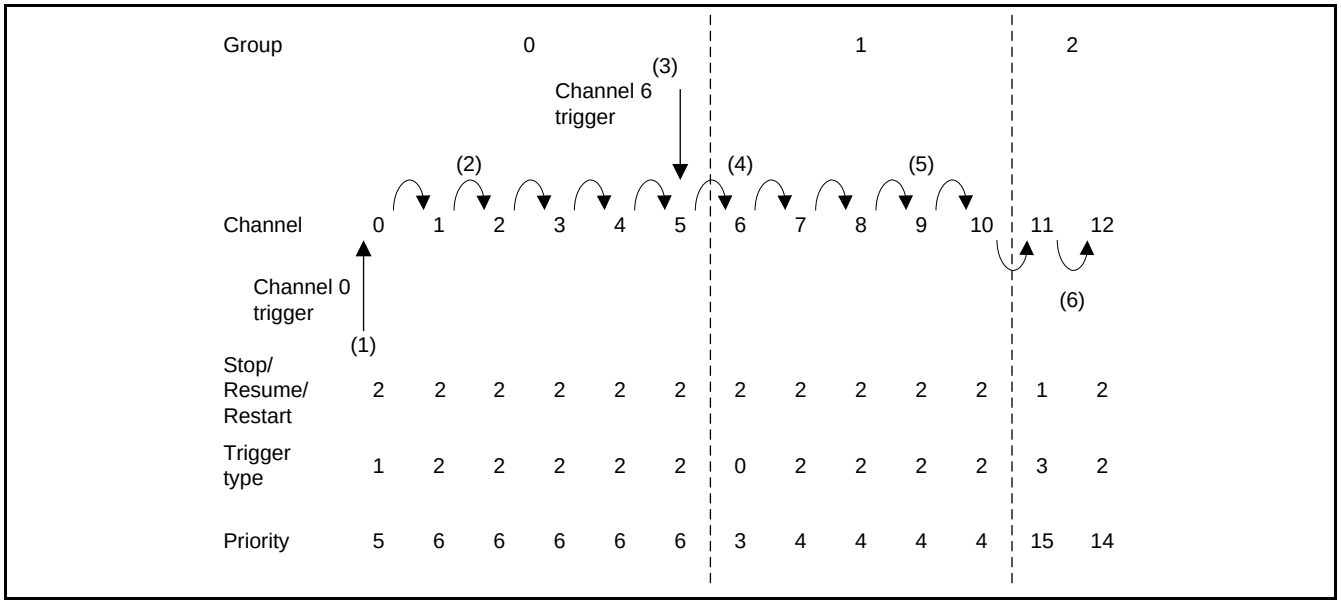


Figure 3-11 shows an example of restarting a group processing with its last converted channel set to resume. These way subgroups can be formed within a group.

(1) Logic channel 0 trigger is issued and processing of the group 0 is initiated.

(2) Processing of the group 0, conversion of logical channels 0 -> 1 -> 2 -> start of channel 3 conversion.

(3) Logic channel 6 trigger is issued during the conversion of channel 3.

(4) The operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"):

Since the priority of the channel 6 is higher than priority of the channel 3, the processing of the group 0 is interrupted during the conversion of channel 3 and trigger status of channel 1 is set (channel 1 is the last converted channel set to resume).

Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"):

Since the priority of the channel 6 is higher than priority of the channel 4, the processing of the group 0 is interrupted after the conversion of channel 3 is finished and trigger status of channel 1 is set (channel 1 is the last converted channel set to resume).

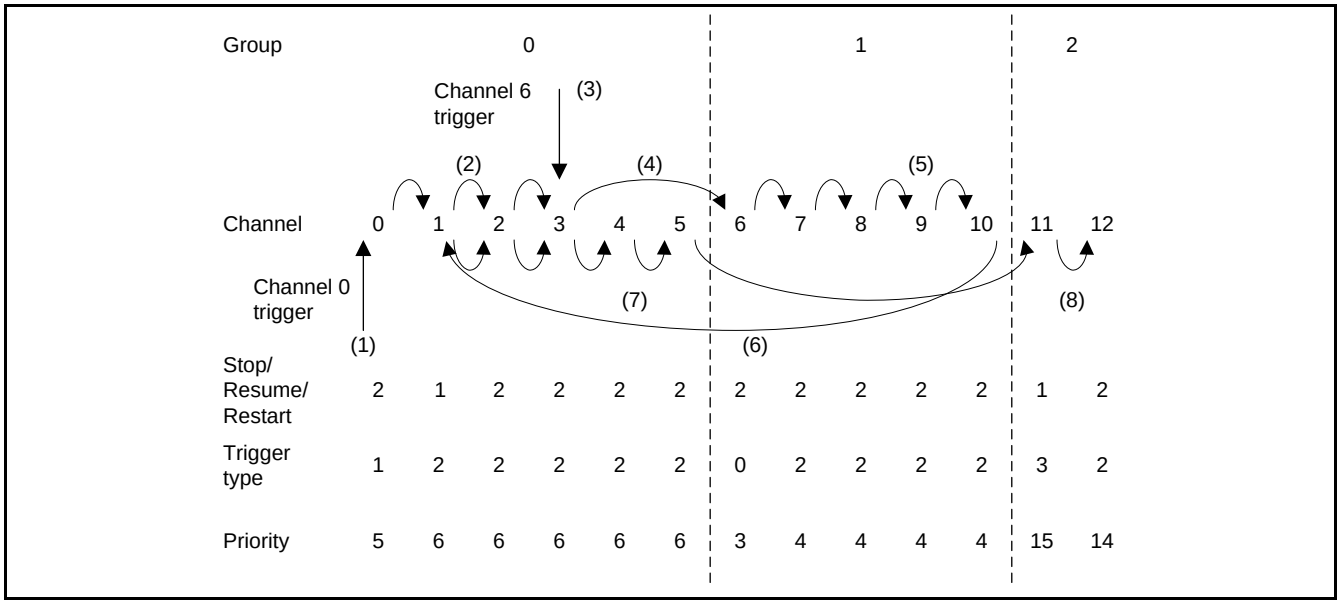
(5) Group 1 is converted.

(6) After group 1 is finished, processing of group 0 is restarted with conversion of channel 1 since its trigger status flag is set to "1".

(7) Subgroup (channels 1 till 5) of the group 0 is converted.

(8) At the end of group 0 processing, idle trigger group 2 is converted.

Figure 3-11 Restarting of the Group Processing with a Subgroup



3.6. Multiple Conversion Logical Channels

First four logical channels can be configured so that after the channel is triggered, several conversions (up to 16) are performed and conversion result is accumulated. These channels are referred to as multiple conversion channels. The following is valid for multiple conversion channels:

- All features and rules provided for regular logical channels (mapping to analog inputs, data protection, triggering rules, priority, channel grouping and behavior within a group) apply also to multiple conversion channels.
- The number of conversions to be performed when the channel is triggered is defined by ADC12Bn_MCCTRL0~3.CNVNUM bit field. If CNVNUM is set to "0", a multiple conversion channel behaves exactly like a regular channel.
- Dedicated A/D conversion data registers ADC12Bn_CD0~3 hold the sum of the single conversion results. Accordingly, ADC12Bn_CD0~3 registers are extended to 16 bits.
- Forced stop mode is enabled (ADC12Bn_CTRL.FSMD = "1"): In case the conversion request with higher priority is issued during multiple conversion, the multiple conversion channel is interrupted after interrupt operation, and the conversion of the channel with the higher priority is started.
  Forced stop mode is disabled (ADC12Bn_CTRL.FSMD = "0"): Once the first conversion of multiple conversion channel is started, the channel processing cannot be interrupted by higher priority requests until CNVNUM+1 conversions are finished, if the dedicated bit ADC12Bn_MCCTRL0~3.ICIRQY (intra-channel interrupt ability) is set to "1". The setting ADC12Bn_MCCTRL0~3.ICIRQY = "0" allows to interrupt multiple conversion channel processing between single conversions, if a higher priority request appears.
- In the case that a multiple conversion channel is interrupted between/during single conversions, the dedicated group interrupted interrupt flags (ADC12Bn_CHSTAT0~3.GRPIRQ and ADC12Bn_GRPIRQ0.GRPIRQ) are set. The corresponding Resume/Restart/Stop bit field (ADC12Bn_CHCTRL0~3.RSMRST) and intra-channel interrupt ability setting (ADC12Bn_MCCTRL0~3.ICIRQY) determine the further channel processing (after higher priority requests are converted).
  Table 3-1 shows the description about multiple conversion after interrupted (ADC12Bn_CTRL.FSMD = "0").
  Table 3-2 shows the description about multiple conversion after interrupted (ADC12Bn_CTRL.FSMD = "1").

Table 3-1 Description about Multiple Conversion after Interrupted (ADC12Bn_CTRL.FSMD = "0")

Settings		Description of Trigger Status and Conversion Counter after Interrupted
ADC12Bn_MCCTRL0~3. ICIRQY	ADC12Bn_CHCTRL0~63. RSMRST[1:0]	
"0" (can be interrupted between single conversion)	"00" (Stop)	Imposes the trigger status flag clearing. Keeps current value of the conversion counter. It is cleared by next conversion request.
	"01" (Resume)	Keeps the trigger status flag set and current value of the conversion counter.
	"10" (Restart)	The trigger status flag stays set only if the channel is first channel in the group, otherwise the trigger status of the channel is cleared and instead the trigger status flag of the first channel in the subgroup is set. The conversion counter is cleared immediately. It means that the accumulated conversion result before the interruption is not really usable because it is not known how many conversions are done.
"1" (cannot be interrupted between single conversion)	Don't care	Don’t care. Multiple conversion is not interrupted.

Table 3-2 Description about Multiple Conversion after Interrupted (ADC12Bn_CTRL.FSMD = "1")

Settings		Description of Trigger Status and Conversion Counter after Interrupted
ADC12Bn_MCCTRL0~3. ICIRQY	ADC12Bn_CHCTRL0~63. RSMRST[1:0]	
"0" (can be interrupted between single conversion)	"00" (Stop)	Imposes the trigger status flag clearing. Keeps current value of the conversion counter. It is cleared by next conversion request.
	"01" (Resume)	Keeps the trigger status flag set. The conversion counter is cleared immediately. It means that the accumulated conversion result before the interruption is not really usable because it is not known how many conversions are done.
	"10" (Restart)	The trigger status flag stays set only if the channel is first channel in the group, otherwise the trigger status of the channel is cleared and instead the trigger status flag of the first channel in the subgroup is set. The conversion counter is cleared immediately. It means that the accumulated conversion result before the interruption is not really usable because it is not known how many conversions are done.
"1" (cannot be interrupted between single conversion)	"00" (Stop)	Imposes the trigger status flag clearing. Keeps current value of the conversion counter. It is cleared by next conversion request.
	"01" (Resume)	Keeps the trigger status flag set and current value of the conversion counter.
	"10" (Restart)	The trigger status flag stays set only if the channel is first channel in the group, otherwise the trigger status of the channel is cleared and instead the trigger status flag of the first channel in the subgroup is set. The conversion counter is cleared immediately. It means that the accumulated conversion result before the interruption is not really usable because it is not known how many conversions are done.

- In the case that range comparison is enabled for a multiple conversion channel, every conversion result is passed to the range comparator.
- For notes of data protection about multiple conversion, refer to section "Logical channel data protection function" in chapter "3. Operation of A/D Converter".

3.7. Forced Stop

- Forced stop mode (ADC12Bn_CTRL.FSMD = "1"):

During an active A/D conversion, it can be forced stop by the following ways.

  - Writing ADC12Bn_CTRL.FSTP = "1":
    All channel's trigger status flags are reset.
    The current A/D conversion is stopped after interrupt operation.
  - Writing "1" to the corresponding bits of ADC12Bn_TRGCL0~1.TRGCL or ADC12Bn_CHCTRLi.TRGCL:
    The corresponding trigger status flag is reset immediately.
    The current A/D conversion is stopped after interrupt operation,

- Not forced stop mode (ADC12Bn_CTRL.FSMD = "0"):

The A/D conversion cannot stop.

  - Writing ADC12Bn_CTRL.FSTP = "1": invalid.
  - Writing "1" to the corresponding bits of ADC12Bn_TRGCL0~1.TRGCL or ADC12Bn_CHCTRLi.TRGCL:
    The corresponding trigger status flag is reset immediately.
    The current A/D conversion is not stopped, but A/D conversion result (the following) is not update.
    - A/D conversion done interrupt flag (the corresponding bits of ADC12Bn_CDONEIRQ0~1.CDONEIRQ or ADC12Bn_CHSTATi.CDONEIRQ)
    - A/D conversion data (the corresponding register of ADC12Bn_CD0~63)
    - Converted logical channel number (ADC12Bn_STAT.ACH[5:0] (ADC12Bn_CTRL.ACHMD is set as "1"))

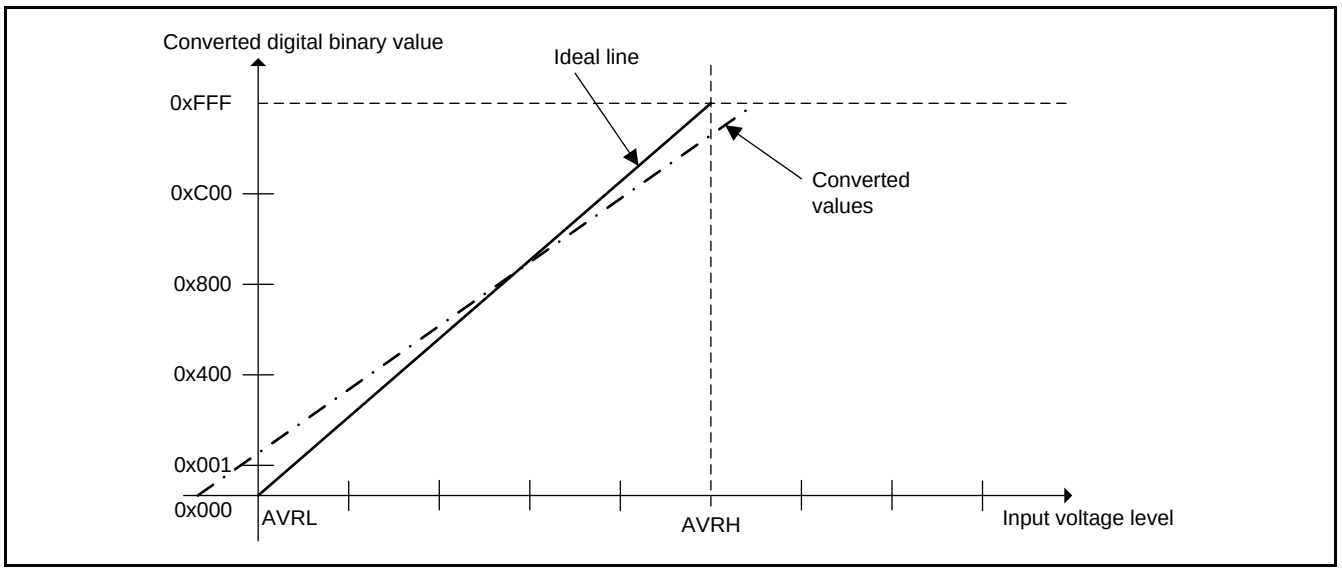
Even if the trigger status flag of the idle trigger channel i (ADC12Bn_CHCTRLi.TRGTYP[1:0] = "11") is cleared (by writing "1" to the corresponding bits of ADC12Bn_TRGCL0~1 or ADC12Bn_CHCTRLi.TRGCL), it is set "1" again immediately when the trigger status flag of all the channels is "0".

3.8. A/D Converter Calibration

Additional feature of multiple conversion logical channels is selection of A/D Converter reference voltages AVRH or AVRL for conversion instead of dedicated analog input signals. If AVRL conversion is wanted it is enough to set one of ADC12Bn_MCCTRL0~3.AVRLSEL bits to "1" and trigger corresponding channel. For AVRH conversion it is needed to set one of ADC12Bn_MCCTRL0~3.AVRHSEL bits to "1", set corresponding ADC12Bn_MCCTRL0~3.AVRLSEL bit to "0" and trigger the dedicated channel.

If AVRHSEL or ARVRLSEL bits is set to "1", during the conversion of the corresponding channel mapping of the logical channel to an analog inputs is disabled.

Figure 3-12 Example of Converted Digital Value Dependence on Input Voltage Level



This feature can be used to perform A/D Converter calibration. As it is shown on Figure 3-12 ideal characteristics of A/D Converter would have:

- transition between digital values 0x000 and 0x001 at AVRL + 0.5LSB input voltage level and
- transition between digital values 0xFFEh and 0xFFF at AVRH - 1.5LSB input voltage level.

If this is not the case, A/D Converter can be calibrated by performing following steps:

- Set gain correction setting (ADC12Bn_GCV.GCV) to "0".
- Set a multiple conversion channel to highest priority and its AVRLSEL bit to "1".
- For different ADC12Bn_OCV.OCV settings trigger the multiple conversion channel, i.e. perform AVRL voltage conversion. It is better to configure multiple conversions of AVRL and calculate average result (accumulated result divided by the number of executed conversions).
- Find ADC12Bn_OCV.OCV setting x for which the transition between digital values 0x000 and 0x001 occurs.
- Set AVRLSEL bit back to "0".
- Set AVRHSEL bit to "1".
- For different ADC12Bn_OCV.OCV settings, trigger the multiple conversion channel, i.e. perform AVRH voltage conversion. It is better to configure multiple conversions of AVRH and calculate average result.
- Find ADC12Bn_OCV.OCV setting y for which the transition between digital values 0xFFE and 0xFFF occurs.
- Set A/D Converter offset compensation setting register ADC12Bn_OCV to the value calculated as "(x+y)/2 + 2".
- Set AVRHSEL bit back to "0".
- Set AVRLSEL bit to "1".
- For different ADC12Bn_GCV.GCV settings, trigger the multiple conversion channel, i.e. perform AVRL voltage conversion. It is better to configure multiple conversions of AVRL and calculate average result.
- Find ADC12Bn_GCV.GCV setting z for which the transition between digital values 0x000 and 0x001 occurs.
- Set A/D Converter gain compensation setting register ADC12Bn_GCV to the value calculated as "z + 1".
- Set AVRLSEL and AVRHSEL bits to "0".
- Trigger and perform further logical channel conversions (analog inputs AN to which are logical channels mapped are converted by calibrated A/D Converter).

3.9. DMA Transfer Function

There are four conversion done DMA triggers available. They can be configured through the corresponding ADC12Bn_CDDS0~3 registers:

- The bit field CDCHNUM specifies the logical channel number whose conversion done interrupt flag issues a DMA request,
- CDCHEN bit controls enabling/disabling of the DMA request.

The feature provides an efficient way to transfer A/D conversion results to memory. Selecting of the group last logical channels through the four CDCHNUM bit fields can configure four different groups for DMA transfers. Since the group is always made of consecutive logical channels and the corresponding conversion data registers can be read linearly, the DMA transfer for the whole group is possible after the conversion of the last channel of the group is finished.

3.10. Range Comparator Function

The range comparator offers eight comparison groups with an upper and a lower threshold register each. The 64 logical channels can be enabled for range comparison and assigned to one of the eight comparators individually.

If range comparator is enabled for a logical channel, it provides two result flags:

- Interrupt flags ADC12Bn_RCIRQ0~1.RCIRQ and ADC12Bn_CHSTAT0~63.RCIRQ, signaling that the A/D conversion result is outside the range or inside the range ("inverted" operation, i.e. ADC12Bn_CHCTRL0~63.RCINVSEL = "1").
- Over threshold flags ADC12Bn_RCOTF0~1.RCOTF and ADC12Bn_CHSTAT0~63.RCOTF, showing that the A/D conversion value exceeded the upper threshold in the case of outside range detection.

Furthermore, each logical channel can be enabled to send an interrupt request to the CPU, if dedicated flags ADC12Bn_RCIRQ0~1.RCIRQ63~0 and ADC12Bn_CHSTAT0~63.RCIRQ are set.

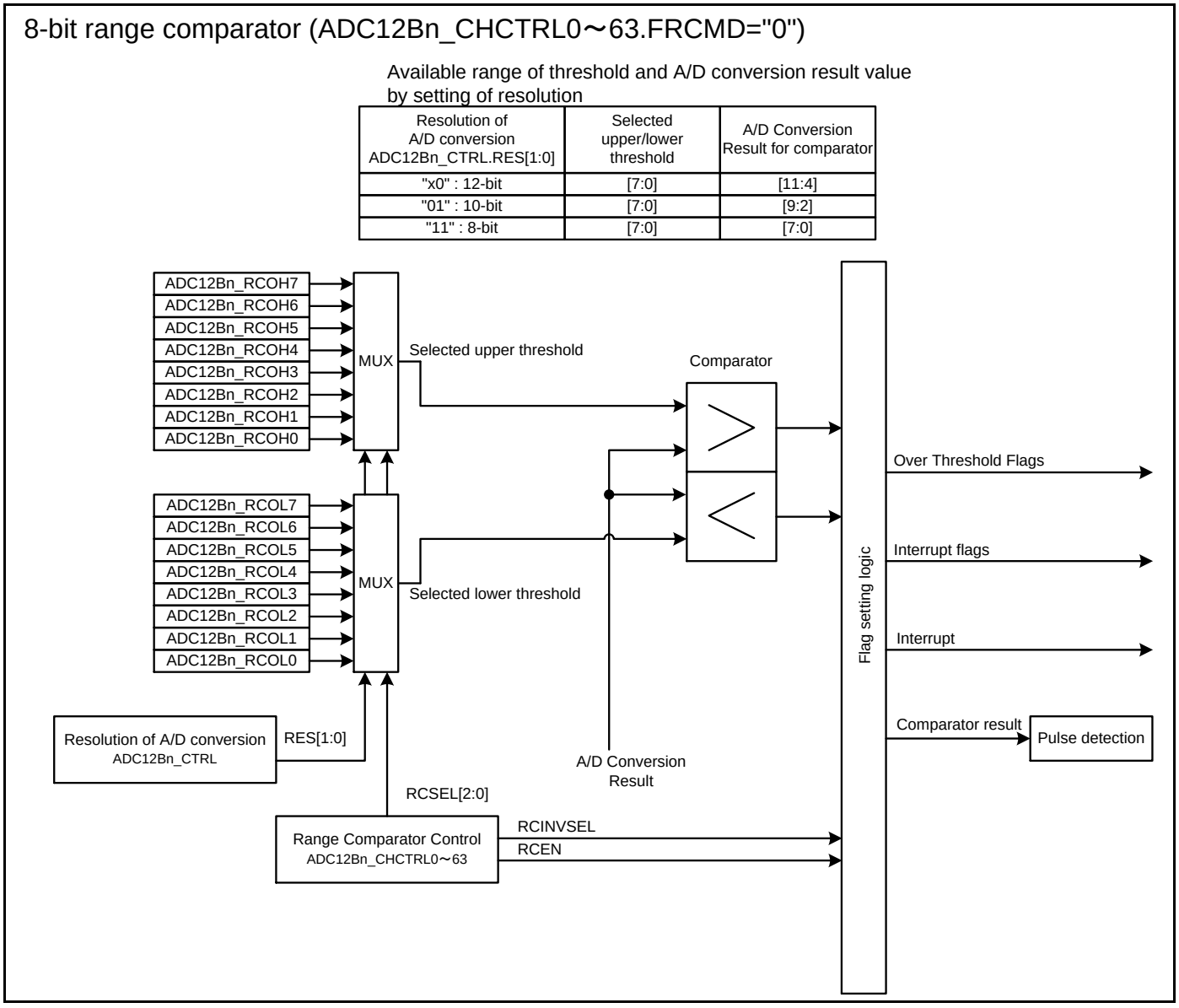
The range comparator can choose 12/8-bit mode by the full range comparator mode (ADC12Bn_CTRL.FRCMD).

ADC12Bn_CTRL.FRCMD	Description
"1" : 12-bit range comparator	ADC12Bn_FRCOL0~7/FRCOH0~7 are used for range comparator upper/lower threshold value. Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields. The upper/lower comparator compares the 12 bits, 10 bits or 8 bits of the A/D conversion result. ADC12Bn_CTRL.RES[1:0] define this resolution. − If 12-bit resolution (ADC12Bn_CTRL.RES[1:0] = "x0"): Compares the 12 bits of A/D conversion result. − If 10-bit resolution (ADC12Bn_CTRL.RES[1:0] = "01"): Compares the 10 bits of A/D conversion result. − If 8-bit resolution (ADC12Bn_CTRL.RES[1:0] = "11"): Compares the 8 bits of A/D conversion result.
"0" : 8-bit range comparator	ADC12Bn_RCOL0~7/RCOH0~7 are used for range comparator upper/lower threshold value. Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields. The upper/lower comparator compares the upper 8 bits of the A/D conversion result.

Figure 3-13 shows the 8-bit range comparator structure.

Figure 3-14 shows the 12-bit range comparator structure.

Figure 3-13 8-bit Range Comparator Structure



8-bit range comparator (ADC12Bn_CHCTRL0～31.FRCMD="0")

Available range of threshold and A/D conversion result value by setting of resolution

Resolution of A/D conversion ADC12Bn_CTRL.RES[1:0]	Selected upper/lower threshold	A/D Conversion Result for comparator
"x0" : 12-bit	[7:0]	[11:4]
"01" : 10-bit	[7:0]	[9:2]
"11" : 8-bit	[7:0]	[7:0]

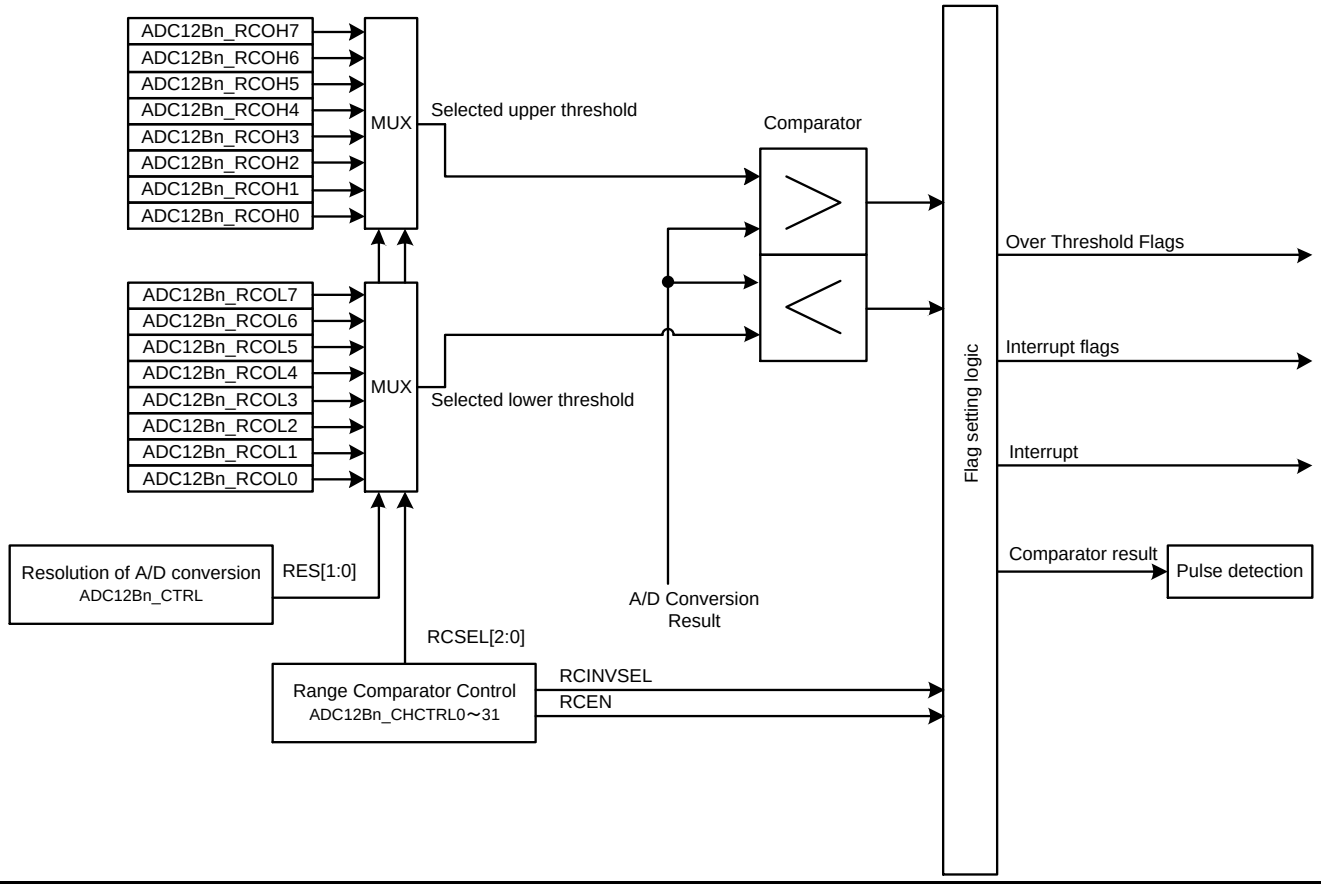
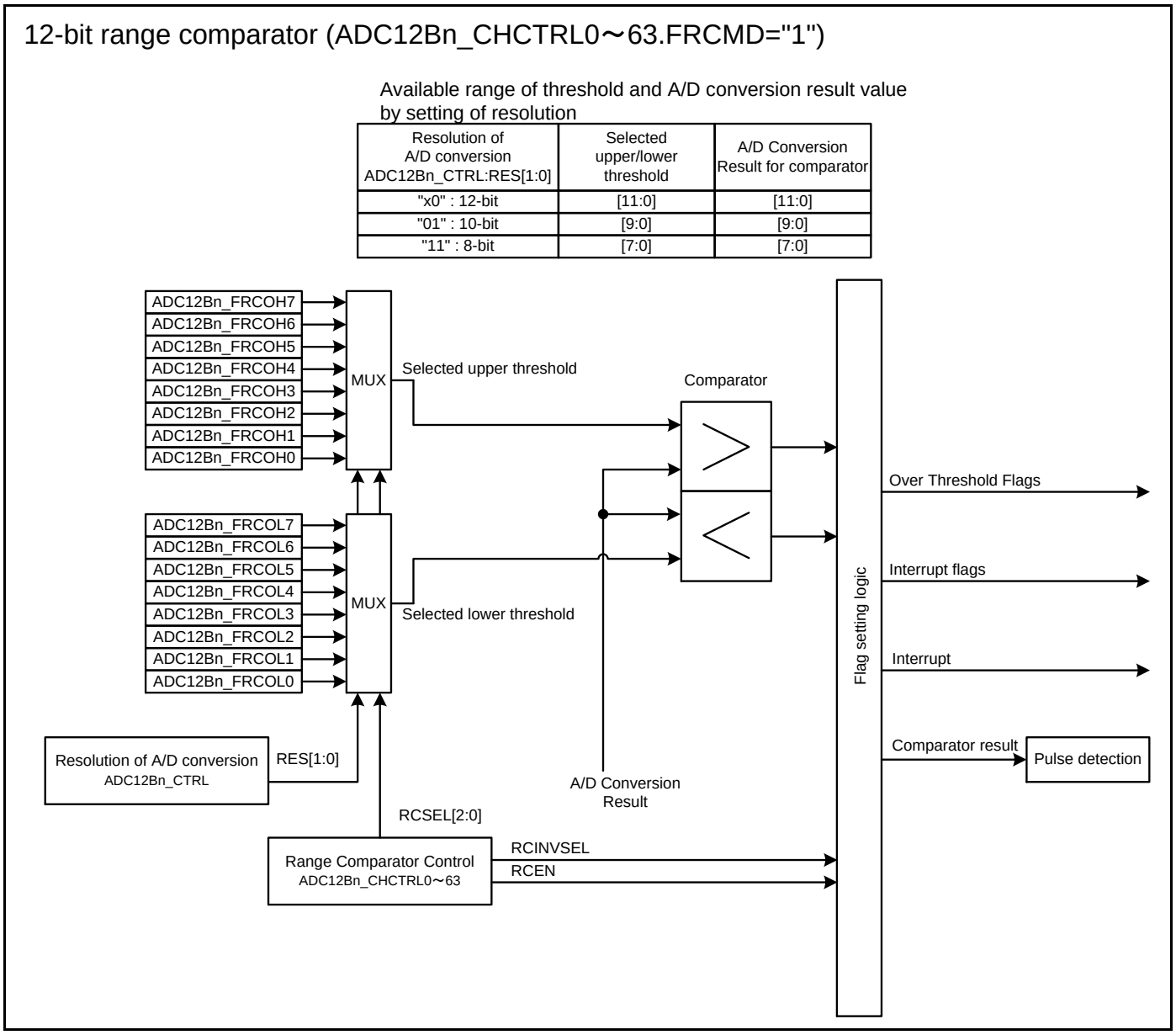


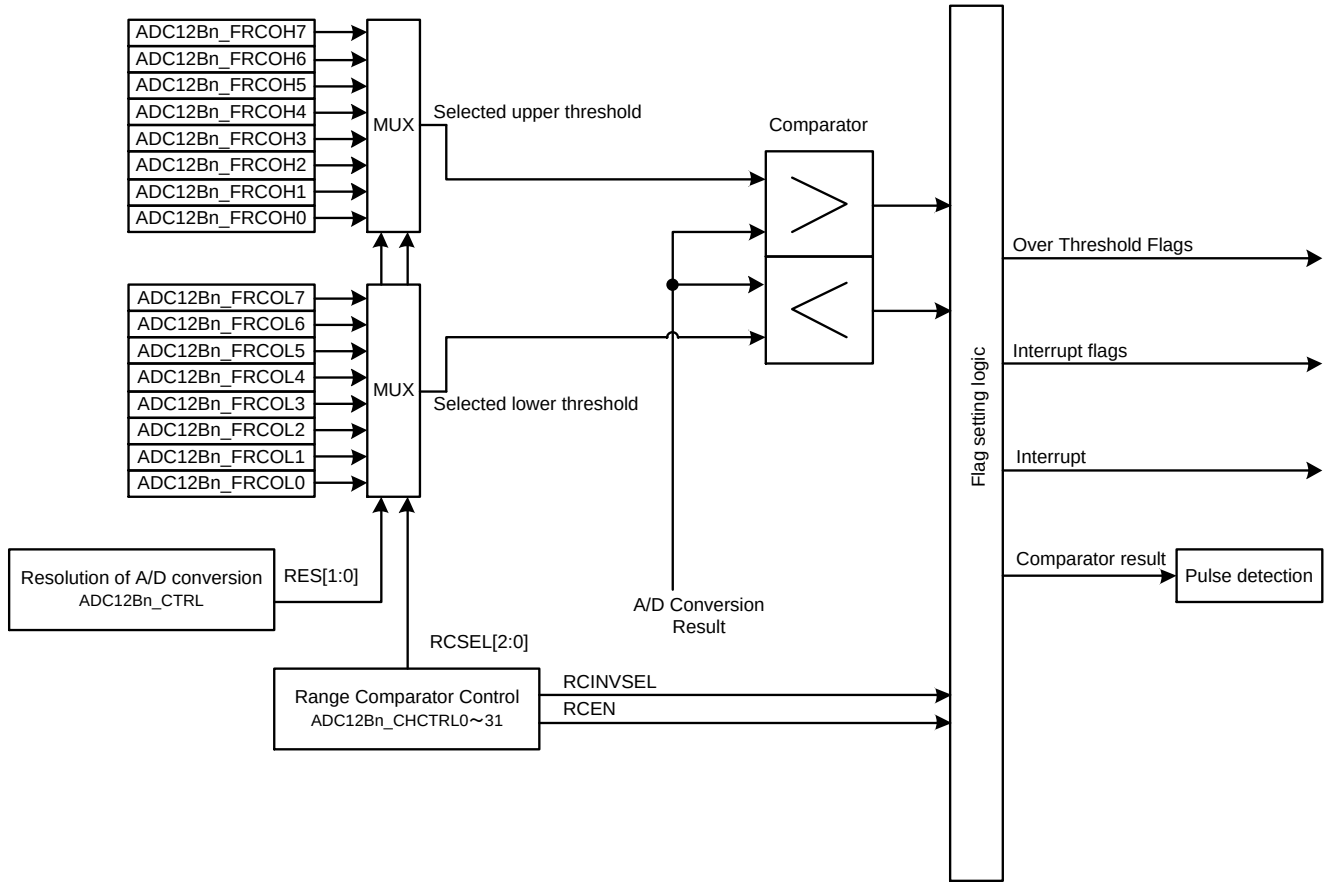
Figure 3-14 12-bit Range Comparator Structure



12-bit range comparator (ADC12Bn_CHCTRL0～31.FRCMD="1")

Available range of threshold and A/D conversion result value by setting of resolution

Resolution of A/D conversion ADC12Bn_CTRL:RES[1:0]	Selected upper/lower threshold	A/D Conversion Result for comparator
"x0" : 12-bit	[11:0]	[11:0]
"01" : 10-bit	[9:0]	[9:0]
"11" : 8-bit	[7:0]	[7:0]



3.11. Pulse Detection Function

The result of the comparison from the range comparator can be filtered by the pulse detection function. For every logical channel, the pulse detection function has a pair of reload registers to store the initial value for the positive and negative down counters (ADC12Bn_PCCTRL0~63.PCTPRL[7:0] and ADC12Bn_PCCTRL0~63.PCTNRL[4:0]). The positive and the negative counters decrement on positive and negative events obtained from the result of the comparison done by the range comparator.

The features of the pulse detection function are:

- Detect events with desired length
- Filter parasitic inverted events

- Each logical channel has a pulse detection function module associated with it
- Interrupt can be generated on detection of a pulse.

The output of the range comparator for particular logical channel signifies either a positive event or a negative event depending on the configuration of ADC12Bn_CHCTRL0~63.RCINVSEL register in the dedicated channel control register and the converted digital value of the A/D Converter as explained in Table 3-3 "Generation of positive/negative events". Whenever a positive event occurs the corresponding positive counter ADC12Bn_PCCTRL0~63.PCTPCT is decremented. Similarly, the dedicated negative counter ADC12Bn_PCCTRL0~63.PCTNCT decrements with each negative event. The purpose of the positive counter is to detect consecutive range comparator events of desired length. The negative counter can be used to force a restart of the positive counter if a negative events of a certain length are detected due to spikes, noise etc.

Table 3-3 Generation of Positive/Negative Events

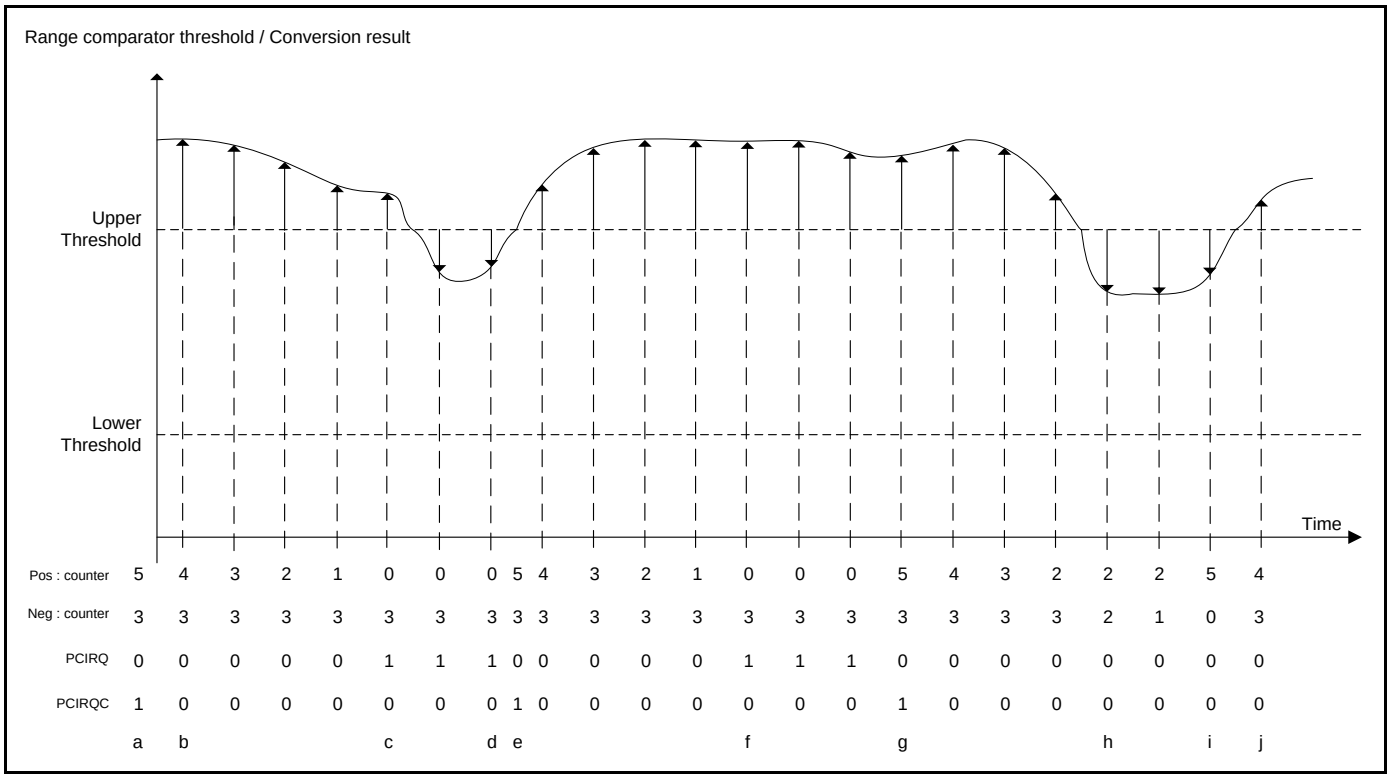
Inverted Range Selection RCINVSEL Bit	Range Comparator Output	Events
0 (configured for “outside range”)	inside range	Negative event
	outside range	Positive event
1 (configured for “inside range”)	outside range	Negative event
	inside range	Positive event

The following steps describe the working principle.

- The positive counter is decremented with each positive event of the corresponding logical channel.
- The corresponding pulse counter interrupt flags (ADC12Bn_CHSTAT0~63.PCIRQ and ADC12Bn_PCIRQ0~1.PCIRQ63~0) are set as the positive counter reaches zero. This flag remains set until it is cleared through writing dedicated ADC12Bn_PCIRQC0~1.PCIRQC63~0 bit to "1". The positive counter and the negative counter are stopped as long as PCIRQ flag of the channel is "1".
- If PCIRQ is set and the corresponding enable bit ADC12Bn_PCIRQE0~1.PCIRQE63~0 is equal to "1", an interrupt is generated.
- The negative counter is decremented with each negative event of the corresponding logical channel except while the corresponding PCIRQ flag is set.
- Positive counter is reloaded with the value set in the reload register ADC12Bn_PCCTRL0~63.PCTPRL when:
  - Negative counter reaches zero,
  - "1" is written to dedicated ADC12Bn_PCIRQC0~1.PCIRQC63~0 bit.
- Negative counter is reloaded with the value set in the reload register ADC12Bn_PCCTRL0~63.PCTNRL when:
  - Any positive event occurs
  - "1" is written to dedicated ADC12Bn_PCIRQC0~1.PCIRQC63~0 bit
  - The positive counter reaches zero and PCIRQ flag is set. The negative counter will hold the reload value as long as PCIRQ flag is not cleared.

The Figure 3-15 shows the operation of the pulse detection function for channel 0 with ADC12Bn_CHCTRL0.RCINVSEL = "0" configured for outside range, reload register ADC12Bn_PCCTRL0.PCTPRL = "101" and reload register ADC12Bn_PCCTRL0.PCTNRL = "011".

Figure 3-15 Example of Pulse Detection Operation



a) Reload counters with appropriate reload value by writing "1" to ADC12Bn_PCIRQC0.PCIRQC0.

b) Positive counter decrements with positive events.

c) Positive counter expires (becomes equal to "0"), the pulse counter interrupt flag PCIRQ is set.

d) A series of negative events does not decrement the negative counter as the PCIRQ flag is set.

e) The pulse counter interrupt flag PCIRQ is cleared and the positive as well as the negative counter are reloaded.

f) Positive counter expires and PCIRQ flag is to "1".

g) Software clears PCIRQ flag and reloads positive and negative counter.

h) Negative event decrements negative counter.

i) Negative counter expires and reloads positive counter.

j) Positive counter decrements and negative counter reloads with positive event.

3.12. Debug Mode

When the ADC12Bn_CTRL.DBGE bit is set to "1" and the processor is in debug state, the A/D Converter completes the current conversion, but further conversion is stopped. When the processor leaves debug state or ADC12Bn_CTRL.DBGE is set to "0", conversion continues with the next channel from where it had stopped.

The ADC12Bn_STAT.BUSY flag is not affected even while ADC12Bn_CTRL.DBGE bit is set to "1" and the processor is in debug state: If all trigger status bits are cleared, ADC12Bn_STAT.BUSY flag is set to "0", and the A/D Converter goes to idle (power-down) state; if any trigger status bit is set again, the A/D Converter leaves idle state, and the ADC12Bn_STAT.BUSY flag is set to "1" after the resumption time elapses.

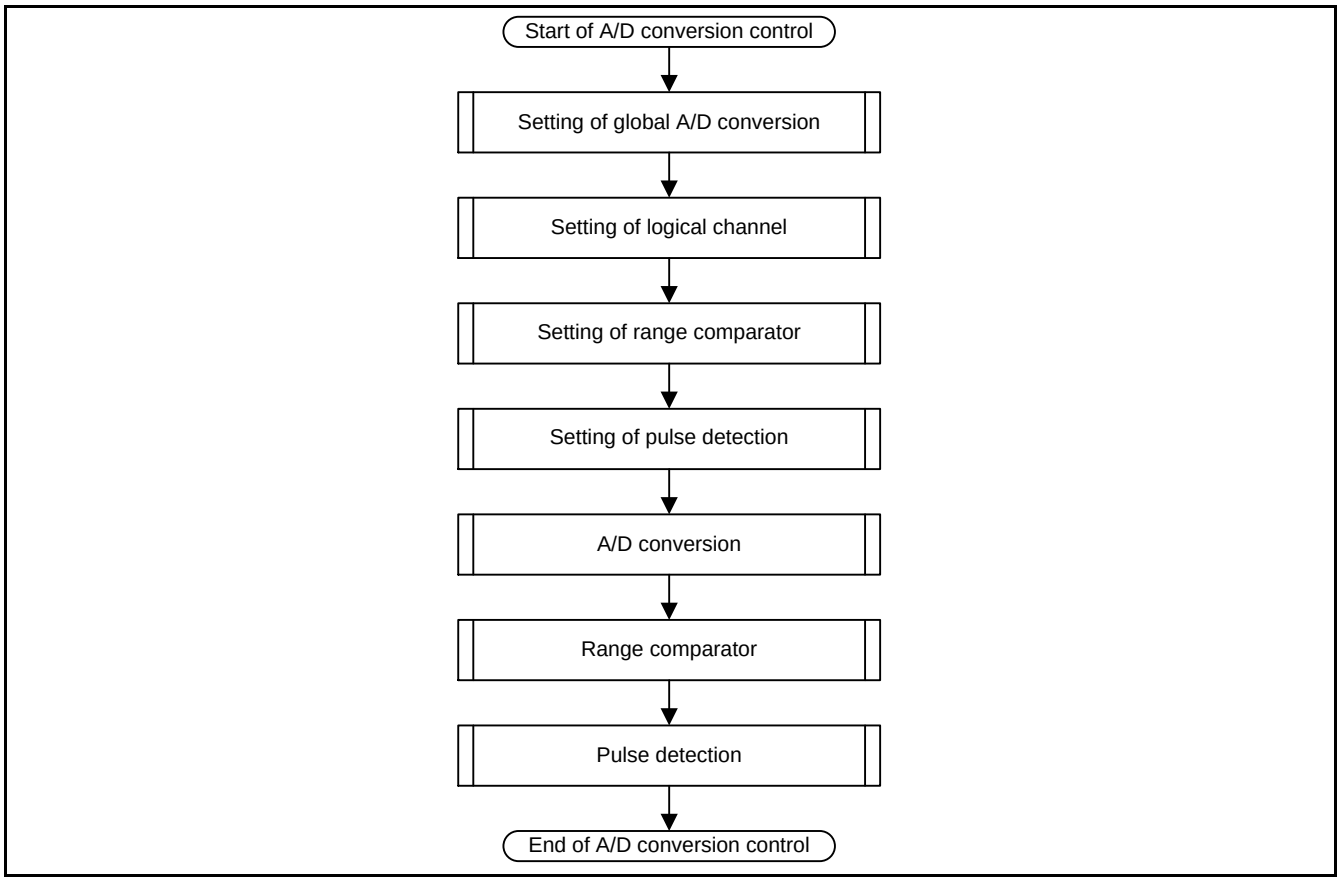
4. Setup Procedure Examples

This section shows examples of setup procedure of A/D converter.

4.1. Control of A/D Conversion

Figure 4-1 shows main flow of controlling A/D converter.

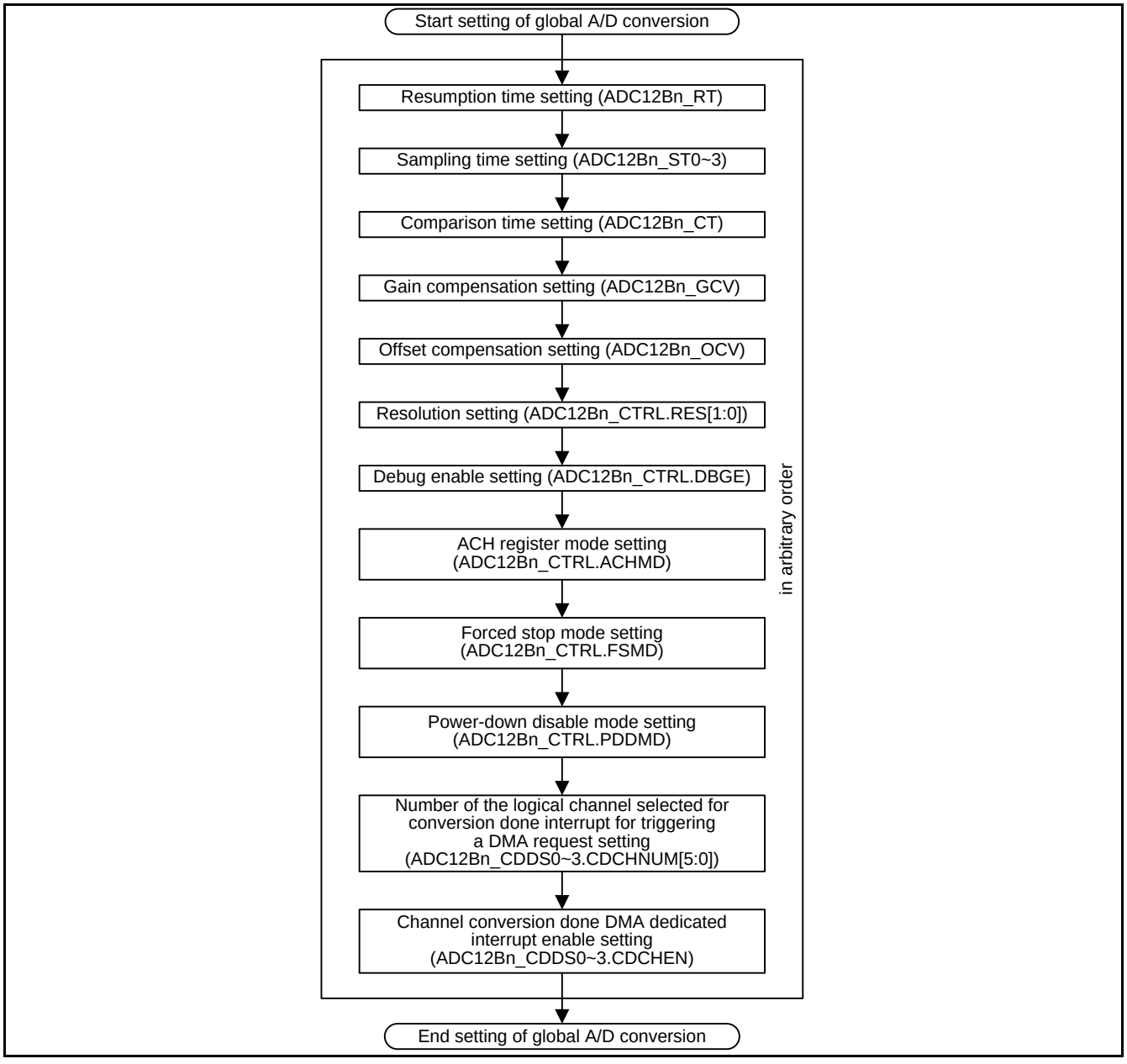
Figure 4-1 Main Flow of Controlling A/D Converter



4.2. Setting of Global A/D Conversion

Figure 4-2 shows the setting procedure of global A/D conversion.

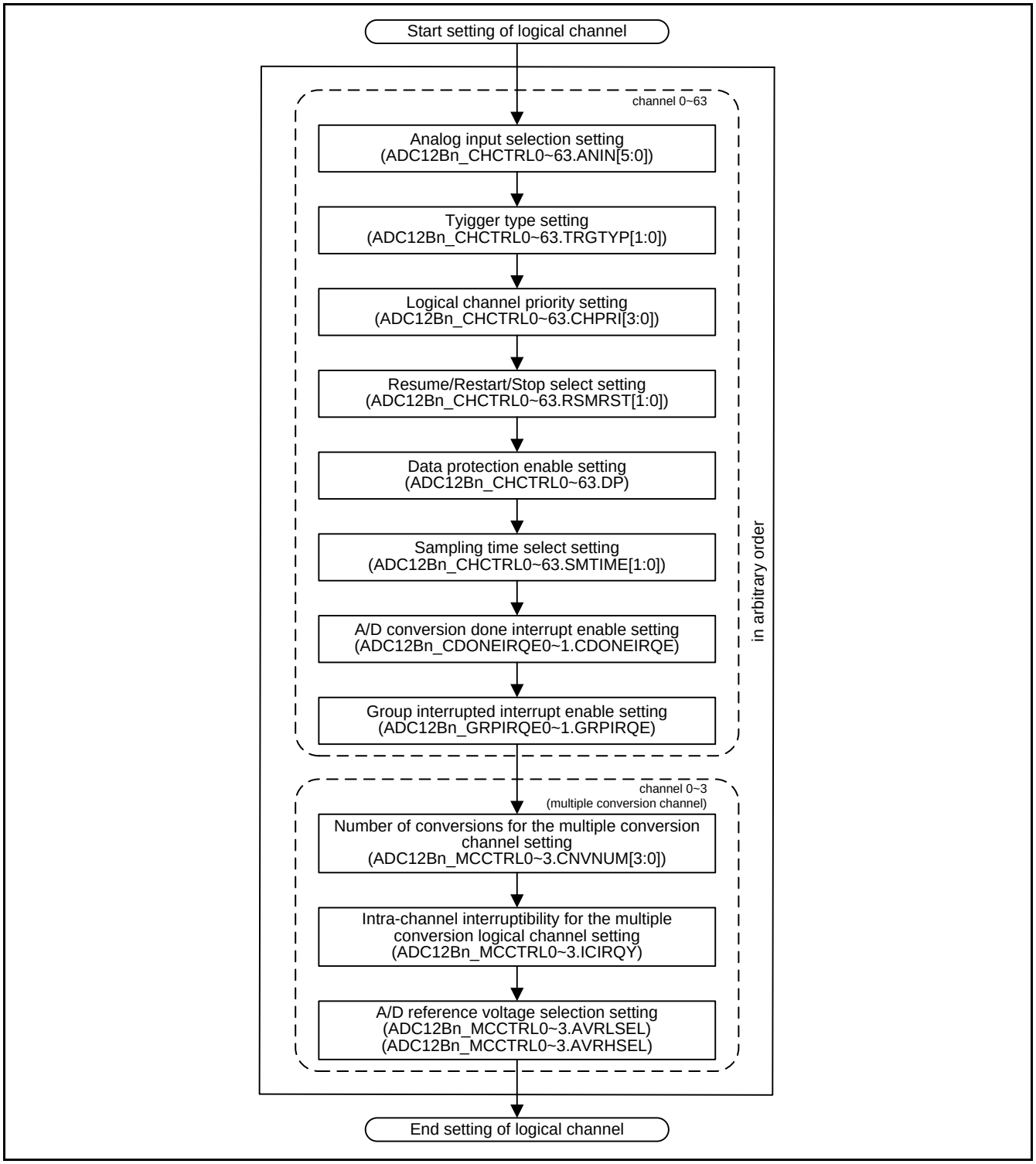
Figure 4-2 Setting of Global A/D Conversion



4.3. Setting of Logical Channel

Figure 4-3 shows the setting procedure of logical channel.

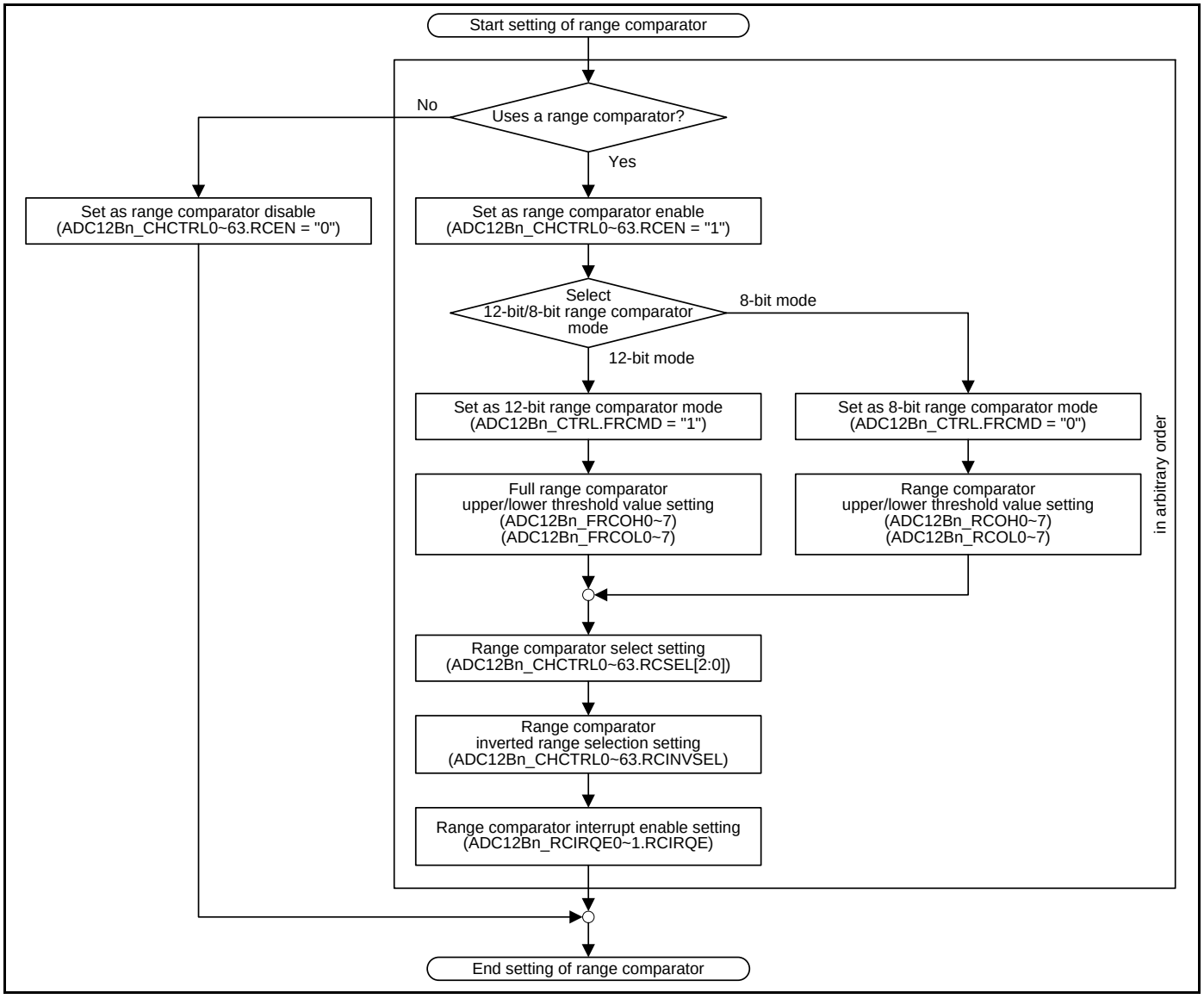
Figure 4-3 Setting of Logical Channel



4.4. Setting of Range Comparator

Figure 4-4 shows the setting procedure of range comparator.

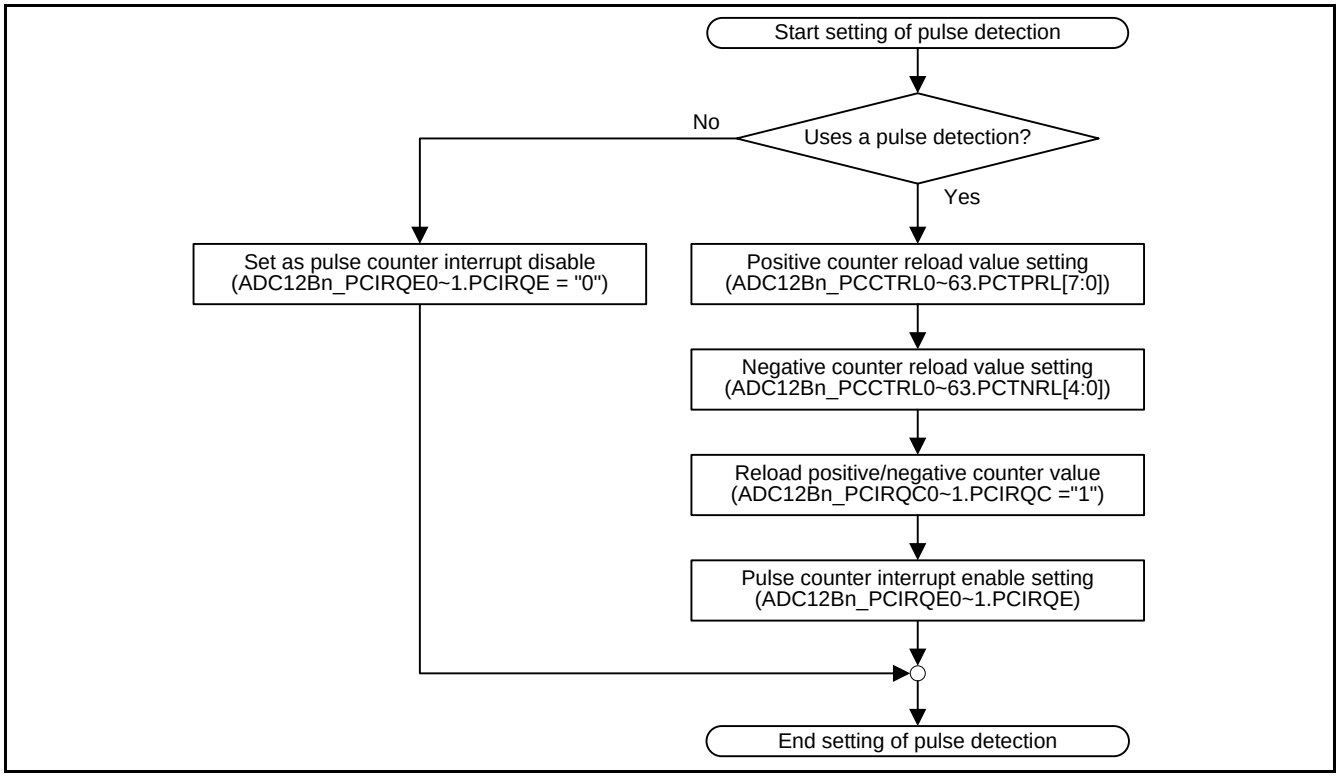
Figure 4-4 Setting of Range Comparator



4.5. Setting of Pulse Detection

Figure 4-5 shows setting procedure of pulse detection.

Figure 4-5 Setting of Pulse Detection

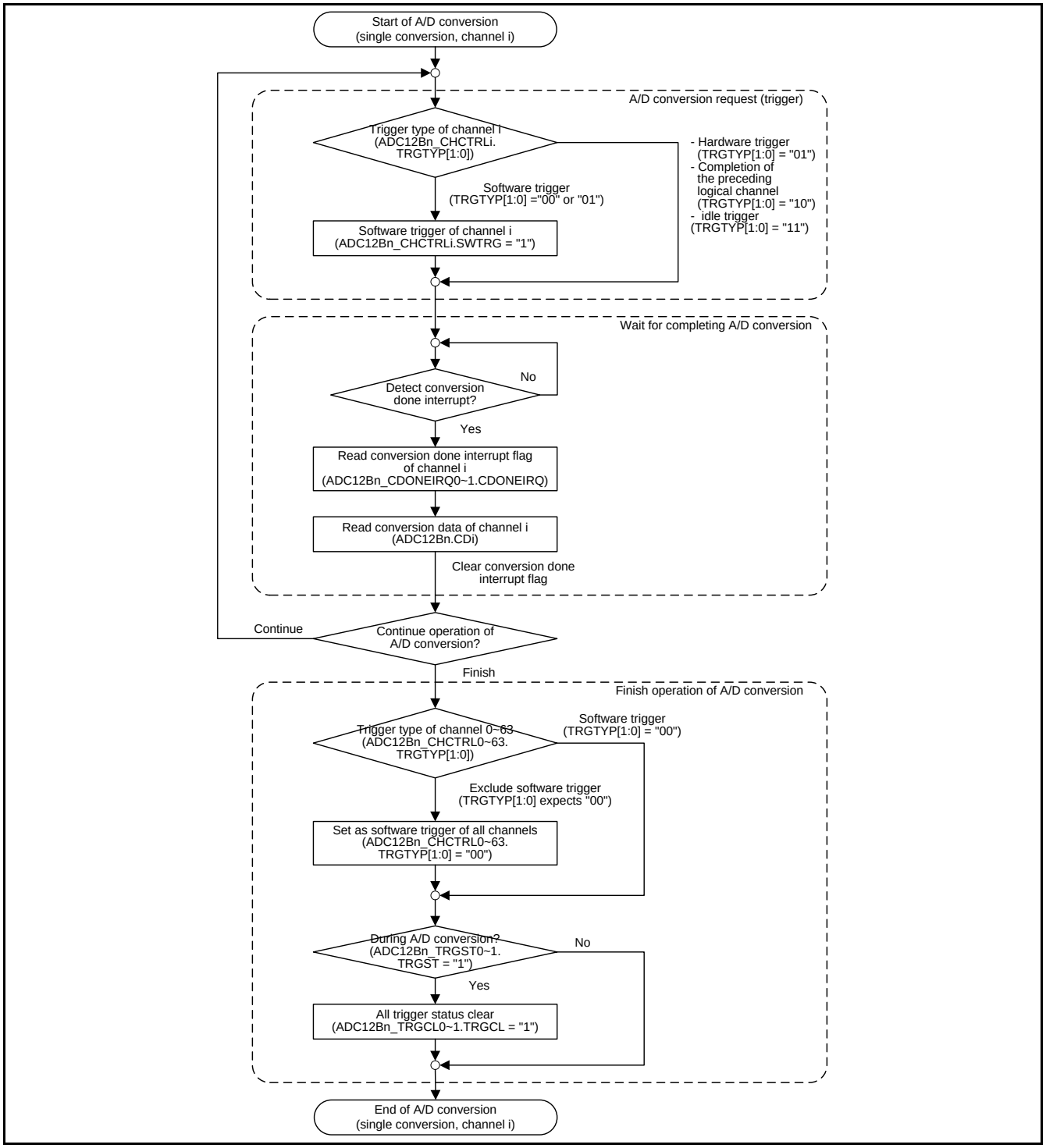


4.6. A/D Conversion

■ Single conversion

Figure 4-6 shows the example of A/D conversion (single conversion) operation.

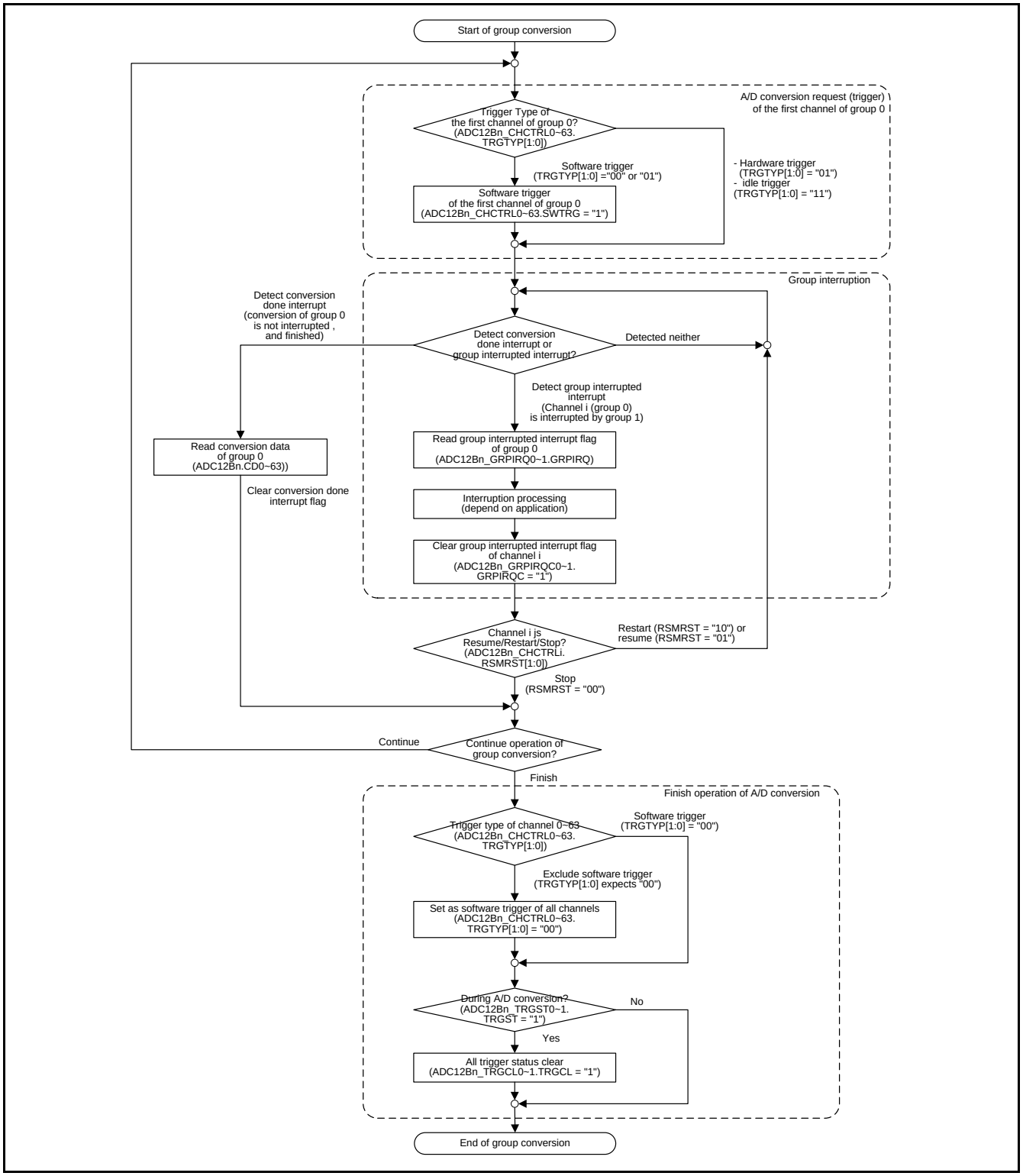
Figure 4-6 A/D Conversion (single conversion, channel i)



■ Group conversion

Figure 4-7 shows the example of group conversion operation.

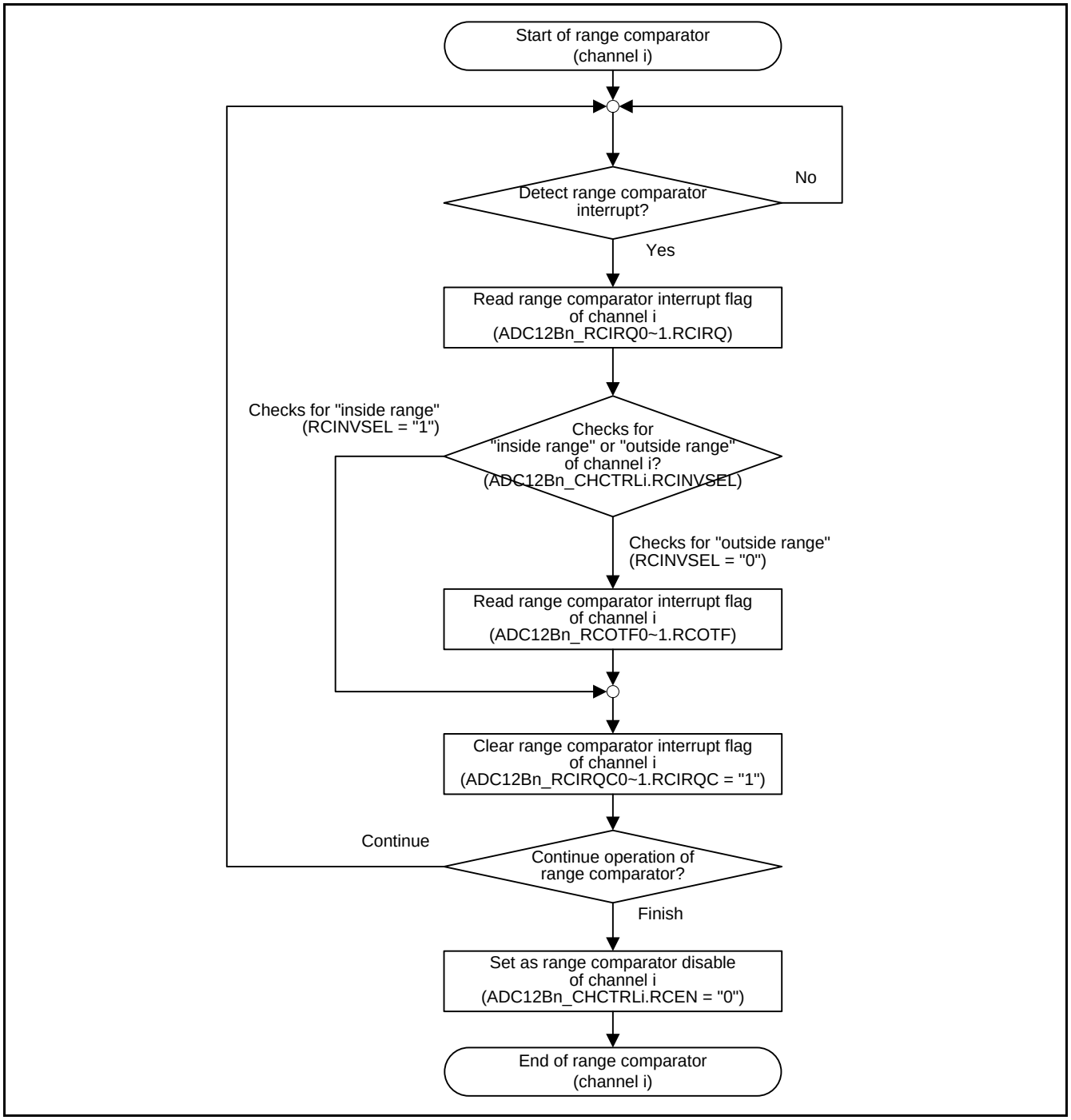
Figure 4-7 Group Conversion



4.7. Range Comparator

Figure 4-8 shows the example of range comparator operation.

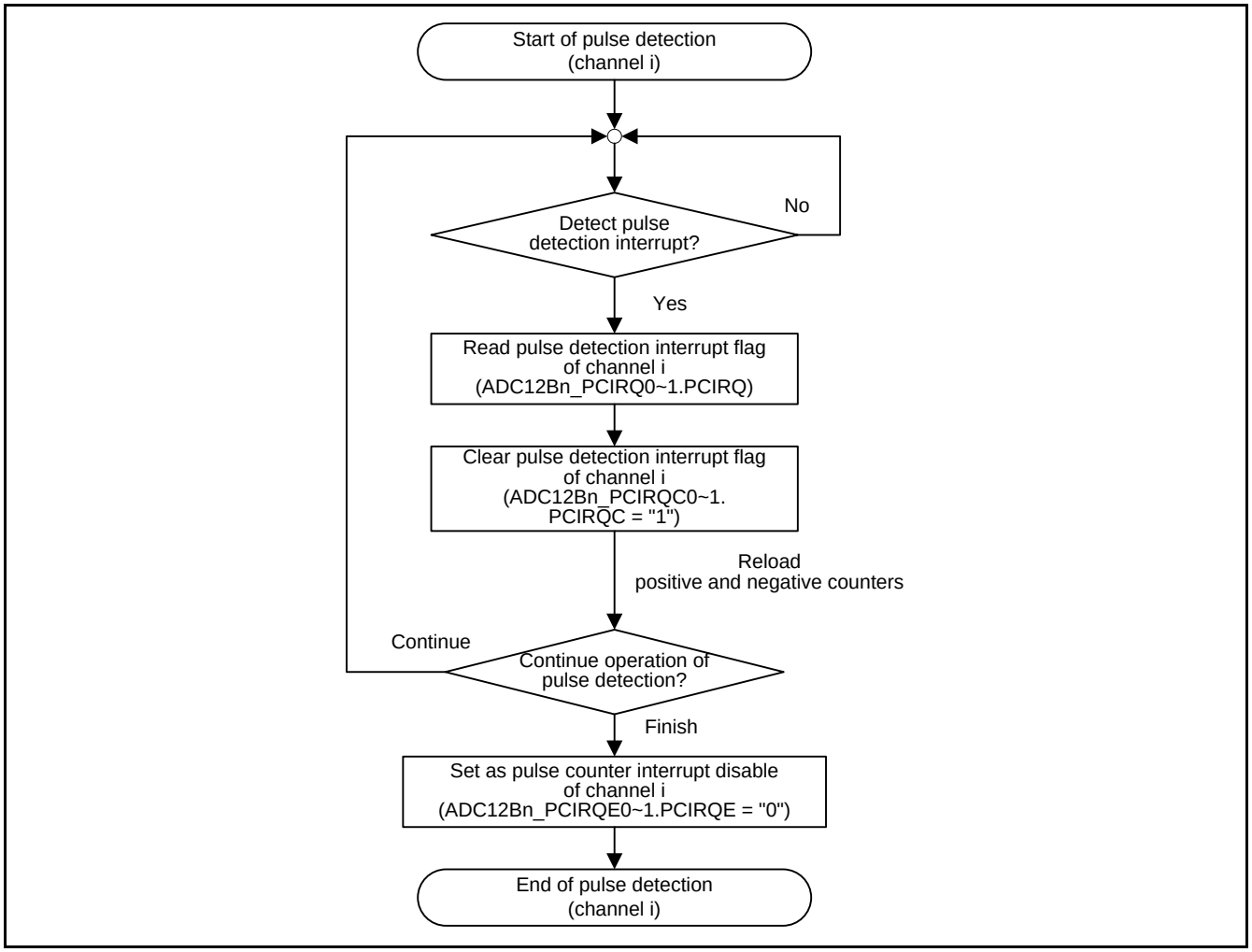
Figure 4-8 Range Comparator (channel i)



4.8. Pulse Detection

Figure 4-9 shows the example of pulse detection operation.

Figure 4-9 Pulse Detection (channel i)



5. Registers

The A/D converter contains registers to configure the operation of A/D conversion and to store the converted values. It also contains registers to configure the range comparators and registers to control and store the status of the pulse detection function. This section describes the registers of the A/D Converter in details.
The suffix "n" in the register name indicates that the register is an instance "n" of the module.

Registers of A/D Converter

Logical channel related registers are:

- A/D Channel Control Registers (ADC12Bn_CHCTRL0~63)
- A/D Channel Status Registers (ADC12Bn_CHSTAT0~63)
- A/D Conversion Data Registers (ADC12Bn_CD0~63)
- Pulse Counter Control Registers (ADC12Bn_PCCTRL0~63)
- A/D Conversion Done Interrupt Flags (ADC12Bn_CDONEIRQ0~1)
- A/D Conversion Done Interrupt Enable Registers (ADC12Bn_CDONEIRQE0~1)
- A/D Conversion Done Interrupt Clear Registers (ADC12Bn_CDONEIRQC0~1)
- Group Interrupted Interrupt Flags (ADC12Bn_GRPIRQ0~1)
- Group Interrupted Interrupt Enable Registers (ADC12Bn_GRPIRQE0~1)
- Group Interrupted Interrupt Clear Registers (ADC12Bn_GRPIRQC0~1)
- Range Comparator Interrupt Flags (ADC12Bn_RCIRQ0~1)
- Range Comparator Interrupt Enable Registers (ADC12Bn_RCIRQE0~1)
- Range Comparator Interrupt Clear Registers (ADC12Bn_RCIRQC0~1)
- Pulse Counter Interrupt Flags (ADC12Bn_PCIRQ0~1)
- Pulse Counter Interrupt Enable Registers (ADC12Bn_PCIRQE0~1)
- Pulse Counter Interrupt Clear Registers (ADC12Bn_PCIRQC0~1)
- A/D Channel Trigger Status Flags (ADC12Bn_TRGST0~1)
- A/D Channel Trigger Status Clear Registers (ADC12Bn_TRGCL0~1)
- A/D Channel Trigger Overrun Flags (ADC12Bn_TRGOR0~1)
- A/D Channel Trigger Overrun Clear Registers (ADC12Bn_TRGORC0~1)
- Range Comparator Over Threshold Flags (ADC12Bn_RCOTF0~1)

Global A/D Converter registers are:

- Conversion Done DMA Select Registers (ADC12Bn_CDDS0~3)
- A/D Converter Resumption Time Setting Register (ADC12Bn_RT)
- A/D Converter Comparison Time Setting Register (ADC12Bn_CT)
- A/D Converter Sampling Time Setting Registers (ADC12Bn_ST0~3)
- A/D Converter Offset Compensation Setting Register (ADC12Bn_OCV)
- A/D Converter Gain Compensation Setting Register (ADC12Bn_GCV)
- A/D Converter Global Control Register (ADC12Bn_CTRL)
- A/D Converter Global Status Register (ADC12Bn_STAT)
- Range Comparator Upper Threshold Registers (ADC12Bn_RCOH0~7)
- Range Comparator Lower Threshold Registers (ADC12Bn_RCOL0~7)
- Full Range Comparator Upper Threshold Registers (ADC12Bn_FRCOH0~7)
- Full Range Comparator Lower Threshold Registers (ADC12Bn_FRCOL0~7)

Multiple conversion logical channel related registers are:

- A/D Multiple Conversion Channel Control Registers (ADC12Bn_MCCTRL0~3)
- A/D Multiple Conversion Channel Status Registers (ADC12Bn_MCSTAT0~3)

5.1. A/D Channel Control Registers (ADC12Bn_CHCTRL0~63)

The A/D Channel Control Registers configure the logical channel specific settings. ADC12Bn_CHCTRL0 (described here) is dedicated to channel 0, ADC12Bn_CHCTRL63 is dedicated to channel 63. Other registers (ADC12Bn_CHCTRL1,.....ADC12Bn_CHCTRL62) have similar bit fields.

REGISTER_NAME	ADC12Bn_CHCTRLi (i = 0~63)
OFFSET	0x0000 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0: 63
NUMERIC_TYPE	-
OTHER	-

A/D Channel Control Register (ADC12Bn_CHCTRL0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved						TRGCL	SWTRG
ACCESS_TYPE	R0,W0						R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0x00						0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCEN	RCINVSEL	Reserved	RCSEL[2]	RCSEL[1]	RCSEL[0]	SMTIME[1]	SMTIME[0]
ACCESS_TYPE	R/W	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	DP	RSMRST[1]	RSMRST[0]	CHPRI[3]	CHPRI[2]	CHPRI[1]	CHPRI[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGTYP[1]	TRGTYP[0]	ANIN[5]	ANIN[4]	ANIN[3]	ANIN[2]	ANIN[1]	ANIN[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:26] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit25] TRGCL : Trigger Status Clear bit

bit	Description
0	No effect.
1	Clears corresponding trigger status bits ADC12Bn_CHSTAT0.TRGST and ADC12Bn_TRGST0.TRGST0.

Reading this bit always returns "0".
This bit is identical to the corresponding bit ADC12Bn_TRGCL0.TRGCL0.

[bit24] SWTRG : Software trigger bit

bit	Description
0	No effect.
1	Sets corresponding trigger status bits ADC12Bn_CHSTAT0.TRGST and ADC12Bn_TRGST0.TRGST0. When trigger status bits are cleared and set at the same time, clearing has priority. When TRGTYP bits are set to "10" or "11", writing SWTRG bit to "1" has no effect.

Reading this bit always returns "0".

Note:

- *Do not write "1" to this bit and reconfigure TRGTYP with the same access.*

[bit23] RCEN : Range Comparator Enable bit

bit	Description
0	Range comparator disabled.
1	Range comparator enabled.

[bit22] RCINVSEL : Range Comparator Inverted Range Selection bit

bit	Description
0	The comparison checks for "outside range", i.e. the over threshold and interrupt flags (ADC12Bn_RCOTF0.RCOTF0, ADC12Bn_CHSTAT0.RCOTF, ADC12Bn_RCIRQ0.RCIRQ0 and ADC12Bn_CHSTAT0.RCIRQ) are set when the ADC result is above the upper threshold OR below the lower threshold. That is called "outside range".
1	The comparison checks for "inside range" i.e., the interrupt flags (ADC12Bn_CHSTAT0.RCIRQ and ADC12Bn_RCIRQ0.RCIRQ0) are set when the ADC result is below or equal the upper threshold AND above or equal the lower threshold. That is called "inside range" mode.

[bit21] Reserved : Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit20:18] RCSEL[2:0] : Range Comparator Select bit

RCSEL[2:0]	Description
000	Select range comparator 0, defined by ADC12Bn_RCOH0 and ADC12Bn_RCOL0 registers.
...	...
111	Select range comparator 7, defined by ADC12Bn_RCOH7 and ADC12Bn_RCOL7 registers.

[bit17:16] SMTIME[1:0] : Sampling Time Select bits

SMTIME[1:0]	Description
00	ADC12Bn_ST0 register value is selected as channel sampling time.
01	ADC12Bn_ST1 register value is selected as channel sampling time.
10	ADC12Bn_ST2 register value is selected as channel sampling time.
11	ADC12Bn_ST3 register value is selected as channel sampling time.

[bit15] Reserved : Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit14] DP : Data protection Enable bit

bit	Description
0	Data protection function disabled.
1	Data protection function enabled.

[bit13:12] RSMRST[1:0] : Resume/Restart/Stop Select bits

RSMRST[1:0]	Description
00	Stop the group processing until next group start channel conversion request is issued.
01	Resume. If the group is interrupted, resume the group processing with this channel.
10	Restart. After the group is interrupted, restart with the start channel or the last converted channel configured as "resume" channel.
11	Reserved.

RSMRST[1:0] are not allowed to update during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

[bit11:8] CHPRI[3:0] : Logical channel priority

CHPRI[3:0]	Description
0000	Highest priority.
...	...
1111	Lowest priority.

CHPRI[3:0] are not allowed to update during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

[bit7:6] TRGTYP[1:0] : Trigger Type bits

TRGTYP[1:0]	Description
00	Software trigger only.
01	Software or hardware trigger.
10	Trigger by conversion completion and updating of conversion data register ADC12Bn_CD of the preceding channel.
11	Idle trigger, the channel trigger status is set if there is no channel having trigger status flag set and inactive data protection function.

TRGTYP[1:0] are not allowed to update during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

[bit5:0] ANIN[5:0] : Analog Input Selection bits

ANIN[5:0]	Description
000000	Analog input AN0 is selected.
...	...
111111	Analog input AN63 is selected.

The effective bits of ANIN[5:0] change according to the number of analog channels.

For the supported number of analog channels, please refer to the Device Data Sheet.

5.2. A/D Channel Status Registers (ADC12Bn_CHSTAT0~63)

These registers store the status information of the corresponding logical channels related to interrupt flags and trigger status. ADC12Bn_CHSTAT0 (described here) is dedicated to channel 0, ADC12Bn_CHSTAT63 is dedicated to channel 63. Other registers (ADC12Bn_CHSTAT1,.....ADC12Bn_CHSTAT62) have similar bit fields.

REGISTER_NAME	ADC12Bn_CHSTATi (i = 0~63)
OFFSET	0x0100 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0: 63
NUMERIC_TYPE	-
OTHER	-

A/D Channel Status Register (ADC12Bn_CHSTAT0)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	RX,WX							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		RCOTF	PCIRQ	RCIRQ	GRPIRQ	CDONEIRQ	TRGST
ACCESS_TYPE	RX,WX		R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	00		0	0	0	0	0	0

[bit15:6] Reserved : Reserved bits

Reading this bit returns an undefined.

Writing data to these bits has no effect on the operation.

[bit5] RCOTF : Range Comparator Over Threshold flag

bit	Description
0	The conversion result is less than or equal to the upper threshold.
1	The conversion result is above the upper threshold.

This bit is identical to the corresponding bit in the ADC12Bn_RCOTF0 register. For more details, see ADC12Bn_RCOTF0~1 register description.

[bit4] PCIRQ : Pulse Counter Interrupt flag

bit	Description
0	Not detected.
1	Pulse counter interrupt detected.

This bit is identical to the corresponding bit in the ADC12Bn_PCIRQ0 register. For more details, see ADC12Bn_PCIRQ0~1 register description.

[bit3] RCIRQ : Range Comparator Interrupt flag

bit	Description
0	Not detected.
1	Range comparator interrupt detected.

This bit is identical to the corresponding bit in the ADC12Bn_RCIRQ0 register. For more details, see ADC12Bn_RCIRQ0~1 register description

[bit2] GRPIRQ : Group Interrupted Interrupt flag

bit	Description
0	Not detected.
1	Group interrupted interrupt detected.

This bit is identical to the corresponding bit in the ADC12Bn_GRPIRQ0 register. For more details, see ADC12Bn_GRPIRQ0~1 register description.

[bit1] CDONEIRQ : Conversion done Interrupt flag

bit	Description
0	Not detected
1	Conversion done Interrupt detected.

This bit is identical to the corresponding bit in the ADC12Bn_CDONEIRQ0 register. For more details, see ADC12Bn_CDONEIRQ0~1 register description.

[bit0] TRGST : Trigger Status flag

bit	Description
0	Conversion request not detected.
1	Conversion request detected.

This bit is identical to the corresponding bit in the ADC12Bn_TRGST0 register. For more details, see ADC12Bn_TRGST0~1 register description.

5.3. A/D Conversion Data Registers (ADC12Bn_CD0~63)

There are 64 A/D conversion data registers, one per logical channel. The registers are written by hardware at the end of conversion if the trigger status is still set. Registers ADC12Bn_CD0~3 are dedicated to multiple conversion channels so their width is 16 bits. Width of registers ADC12Bn_CD4~63 is 12 bits.

REGISTER_NAME	ADC12Bn_CDi (i = 0~63)
OFFSET	0x0180 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0: 63
NUMERIC_TYPE	-
OTHER	-

A/D Conversion Data Result Registers (ADC12Bn_CD0~3)

Here is the register ADC12Bn_CD0 described, the registers ADC12Bn_CD1~3 have similar bit fields.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	D[15]	D[14]	D[13]	D[12]	D[11]	D[10]	D[9]	D[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	X	X	X	X	X	X	X	X

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	X	X	X	X	X	X	X	X

[bit15:0] D[15:0] : A/D Conversion Data bits

These bits store the conversion data.

The register is updated at the end of the A/D conversion only in the case the corresponding trigger status (ADC12Bn_TRGST0.TRGST0 and ADC12Bn_CHSTAT0.TRGST bits) is still "1".

The logical channel can be configured as multiple conversion channel, if ADC12Bn_MCCTRL0.CNVNUM is greater than 0. In that case the result of conversions is accumulated until the number of conversions reaches ADC12Bn_MCCTRL0.CNVNUM (the result of the first conversion is always directly stored and the following conversion results are added on current register value). Accordingly, if 12-bit conversion resolution is selected the conversion data can become 16-bit wide.

If ADC12Bn_MCCTRL0.CNVNUM is equal 0 (only one consecutive conversion is requested), the conversion data are provided in range of:

- bit[7:0] for 8-bit conversion resolution (bits 15~8 are "0"),
- bit[9:0] for 10-bit conversion resolution (bits 15~10 are "0"),
- bit[11:0] for 12-bit conversion resolution (bits 15~12 are "0").

A/D Conversion Data Registers (ADC12Bn_CD4~63)

Here is the register ADC12Bn_CD4 described, the registers ADC12Bn_CD5~63 have similar bit fields.

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved				D[11]	D[10]	D[9]	D[8]
ACCESS_TYPE	RX,WX				R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0000				X	X	X	X

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	X	X	X	X	X	X	X	X

[bit15:12] Reserved : Reserved bits

Reading this bit returns an undefined.

Writing data to these bits has no effect on the operation.

[bit11:0] D[11:0] : A/D Conversion Data bits

These bits store the conversion data, provided in range of:

- bit[7:0] for 8-bit conversion resolution (bits 11~8 are "0"),
- bit[9:0] for 10-bit conversion resolution (bits 11~10 are "0") and
- bit[11:0] for 12-bit conversion resolution.

The register is updated at the end of the A/D conversion only in the case the corresponding trigger status (ADC12Bn_TRGST0.TRGST4 and ADC12Bn_CHSTAT4.TRGST bits) is still "1".

5.4. Pulse Counter Control Registers (ADC12Bn_PCCTRL0~63)

These registers hold the reload and current values of positive and negative counters of the pulse detection function for the corresponding logical channel. The positive counters count down the positive events of the range comparator and negative counters count down the negative events of the range comparator. ADC12Bn_PCCTRL0 (described here) is dedicated to channel 0, ADC12Bn_PCCTRL63 is dedicated to channel 63. Other registers (ADC12Bn_PCCTRL1,.....ADC12Bn_PCCTRL62) have similar bit fields.

REGISTER_NAME	ADC12Bn_PCCTRLi (i = 0~63)
OFFSET	0x0200 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0: 63
NUMERIC_TYPE	-
OTHER	-

Pulse Counter Control Register (ADC12Bn_PCCTRL0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved			PCTNCT[4]	PCTNCT[3]	PCTNCT[2]	PCTNCT[1]	PCTNCT[0]
ACCESS_TYPE	RX,WX			R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	000			0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved			PCTNRL[4]	PCTNRL[3]	PCTNRL[2]	PCTNRL[1]	PCTNRL[0]
ACCESS_TYPE	R0,W0			R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	000			0	0	0	1	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCTPCT[7]	PCTPCT[6]	PCTPCT[5]	PCTPCT[4]	PCTPCT[3]	PCTPCT[2]	PCTPCT[1]	PCTPCT[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCTPRL[7]	PCTPRL[6]	PCTPRL[5]	PCTPRL[4]	PCTPRL[3]	PCTPRL[2]	PCTPRL[1]	PCTPRL[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	1	0

[bit31:29] Reserved : Reserved bits

Reading this bit returns an undefined.

Writing data to these bits has no effect on the operation.

[bit28:24] PCTNCT[4:0] : Pulse Negative Counter Register

This register reflects the current counter value of the pulse detection negative counter. Reload value is determined by PCTNRL.

The negative counter is reloaded under one of the following conditions:

- Writing "1" to the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit.
- Any positive event from the appropriate range comparator.

Therefore, in order to reload negative counter immediately after set to PCTNRL[4:0], the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit should be written "1".

[bit23:21] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit20:16] PCTNRL[4:0] : Pulse Negative Counter Reload Register

These register bits hold the reload value of the negative counter PCTNCT used for counting negative events of the range comparator. Do not set this bit field to "00000".

The negative counter is reloaded with the value from this register when "1" is written to the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit or on any positive event from the appropriate range comparator.

For further explanation of negative events and operation of pulse detection function refer to section "Pulse Detection Function" in chapter "3 Operation of A/D Converter"

[bit15:8] PCTPCT[7:0] : Pulse Positive Counter Register

This register reflects the current counter value of the pulse detection positive counter. Reload value is determined by PCTPRL.

The positive counter is reloaded under one of the following conditions:

- Writing "1" to the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit.
- Expiration of the corresponding negative counter (PCTNCT).

Therefore, in order to reload positive counter immediately after set to PCTPRL[7:0], the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit should be written "1".

[bit7:0] PCTPRL[7:0] : Pulse Positive Counter Reload Register

These register bits hold the reload value of the positive counter PCTPCT used for counting positive events of the range comparator. Do not set this bit field to "00000000".

The positive counter is reloaded with the value from this register when "1" is written to the corresponding ADC12Bn_PCIRQC0~1.PCIRQC bit or on expiration of the corresponding negative counter (PCTNCT).

For further explanation of positive events and operation of pulse detection function refer to section "Pulse Detection Function" in chapter "3 Operation of A/D Converter"

5.5. A/D Conversion Done Interrupt Flag Registers (ADC12Bn_CDONEIRQ0~1)

A/D Conversion Done Interrupt Flag Registers ADC12Bn_CDONEIRQ0~1 contain the status of conversion done interrupt flags for all 64 logical channels.

REGISTER_NAME	ADC12Bn_CDONEIRQi (i = 0~1)
OFFSET	0x0300 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Conversion Done Interrupt Flag Register (ADC12Bn_CDONEIRQ0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQ31	CDONEIRQ30	CDONEIRQ29	CDONEIRQ28	CDONEIRQ27	CDONEIRQ26	CDONEIRQ25	CDONEIRQ24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQ23	CDONEIRQ22	CDONEIRQ21	CDONEIRQ20	CDONEIRQ19	CDONEIRQ18	CDONEIRQ17	CDONEIRQ16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQ15	CDONEIRQ14	CDONEIRQ13	CDONEIRQ12	CDONEIRQ11	CDONEIRQ10	CDONEIRQ9	CDONEIRQ8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQ7	CDONEIRQ6	CDONEIRQ5	CDONEIRQ4	CDONEIRQ3	CDONEIRQ2	CDONEIRQ1	CDONEIRQ0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQ31~0 : Conversion Done Interrupt flags

bit	Description
0	Conversion done interrupt request not detected.
1	Conversion done interrupt request detected.

This bit is set when conversion data is stored in corresponding conversion data register ADC12Bn_CD0~31 and corresponding trigger status (ADC12Bn_TRGST0.TRGST and ADC12Bn_CHSTAT0~31.TRGST bits) is still "1".

In case of multiple conversion channels this bit is set when the last conversion is done, final conversion result is accumulated and corresponding trigger status (ADC12Bn_TRGST0.TRGST and ADC12Bn_CHSTAT0~31.TRGST bits) is still "1".

This bit is cleared by writing "1" to the corresponding ADC12Bn_CDONEIRQC0 bits or by reading the corresponding conversion data register ADC12Bn_CD0~31 (except by the debug master, DAP). All ADC12Bn_CD0~31 read access types (8/16/32-bit) clear the flag.

If this bit is set and cleared at the same time, clearing has higher priority.

This bit is identical to the CDONEIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 registers.

A/D Conversion Done Interrupt Flag Register (ADC12Bn_CDONEIRQ1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQ63	CDONEIRQ62	CDONEIRQ61	CDONEIRQ60	CDONEIRQ59	CDONEIRQ58	CDONEIRQ57	CDONEIRQ56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQ55	CDONEIRQ54	CDONEIRQ53	CDONEIRQ52	CDONEIRQ51	CDONEIRQ50	CDONEIRQ49	CDONEIRQ48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQ47	CDONEIRQ46	CDONEIRQ45	CDONEIRQ44	CDONEIRQ43	CDONEIRQ42	CDONEIRQ41	CDONEIRQ40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQ39	CDONEIRQ38	CDONEIRQ37	CDONEIRQ36	CDONEIRQ35	CDONEIRQ34	CDONEIRQ33	CDONEIRQ32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQ63~32 : Conversion Done Interrupt flags

bit	Description
0	Conversion done interrupt request not detected.
1	Conversion done interrupt request detected.

This bit is set when conversion data is stored in corresponding conversion data register ADC12Bn_CD32~63 and corresponding trigger status (ADC12Bn_TRGST1.TRGST and ADC12Bn_CHSTAT32~63.TRGST bits) is still "1".

This bit is cleared by writing "1" to the corresponding ADC12Bn_CDONEIRQC1 bits or by reading the corresponding conversion data register ADC12Bn_CD32~63 (except by the debug master, DAP). All ADC12Bn_CD32~63 read access types (8/16/32-bit) clear the flag.

If this bit is set and cleared at the same time, clearing has higher priority.

This bit is identical to the CDONEIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 registers.

5.6. A/D Conversion Done Interrupt Enable Registers (ADC12Bn_CDONEIRQE0~1)

These registers contain enable bits for all 64 logical channels, dedicated to the generation of conversion done interrupt.

REGISTER_NAME	ADC12Bn_CDONEIRQEi (i = 0~1)
OFFSET	0x0308 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Conversion Done Interrupt Enable Register (ADC12Bn_CDONEIRQE0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQE31	CDONEIRQE30	CDONEIRQE29	CDONEIRQE28	CDONEIRQE27	CDONEIRQE26	CDONEIRQE25	CDONEIRQE24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQE23	CDONEIRQE22	CDONEIRQE21	CDONEIRQE20	CDONEIRQE19	CDONEIRQE18	CDONEIRQE17	CDONEIRQE16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQE15	CDONEIRQE14	CDONEIRQE13	CDONEIRQE12	CDONEIRQE11	CDONEIRQE10	CDONEIRQE9	CDONEIRQE8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQE7	CDONEIRQE6	CDONEIRQE5	CDONEIRQE4	CDONEIRQE3	CDONEIRQE2	CDONEIRQE1	CDONEIRQE0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQE31~0 : Conversion Done Interrupt Enable bits

bit	Description
0	Conversion done interrupt disabled
1	Conversion done interrupt enabled.

Conversion done interrupt is issued when the bit is "1" and the corresponding interrupt flags ADC12Bn_CDONEIRQ0.CDONEIRQ31~0 and ADC12Bn_CHSTAT0~31.CDONEIRQ are set to "1".

A/D Conversion Done Interrupt Enable Register (ADC12Bn_CDONEIRQE1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQE63	CDONEIRQE62	CDONEIRQE61	CDONEIRQE60	CDONEIRQE59	CDONEIRQE58	CDONEIRQE57	CDONEIRQE56
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQE55	CDONEIRQE54	CDONEIRQE53	CDONEIRQE52	CDONEIRQE51	CDONEIRQE50	CDONEIRQE49	CDONEIRQE48
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQE47	CDONEIRQE46	CDONEIRQE45	CDONEIRQE44	CDONEIRQE43	CDONEIRQE42	CDONEIRQE41	CDONEIRQE40
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQE39	CDONEIRQE38	CDONEIRQE37	CDONEIRQE36	CDONEIRQE35	CDONEIRQE34	CDONEIRQE33	CDONEIRQE32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQE63~32 : Conversion Done Interrupt Enable bits

bit	Description
0	Conversion done interrupt disabled
1	Conversion done interrupt enabled.

Conversion done interrupt is issued when the bit is "1" and the corresponding interrupt flags ADC12Bn_CDONEIRQ1.CDONEIRQ63~32 and ADC12Bn_CHSTAT32~63.CDONEIRQ are set to "1".

5.7. A/D Conversion Done Interrupt Clear Registers (ADC12Bn_CDONEIRQC0~1)

These registers contain bits for clearing corresponding conversion done interrupt flags in the ADC12Bn_CDONEIRQ0~1 registers. The 64 bits are assigned to 64 logical channels.

REGISTER_NAME	ADC12Bn_CDONEIRQCi (i = 0~1)
OFFSET	0x0310 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Conversion Done Interrupt Clear Register (ADC12Bn_CDONEIRQC0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQC31	CDONEIRQC30	CDONEIRQC29	CDONEIRQC28	CDONEIRQC27	CDONEIRQC26	CDONEIRQC25	CDONEIRQC24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQC23	CDONEIRQC22	CDONEIRQC21	CDONEIRQC20	CDONEIRQC19	CDONEIRQC18	CDONEIRQC17	CDONEIRQC16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQC15	CDONEIRQC14	CDONEIRQC13	CDONEIRQC12	CDONEIRQC11	CDONEIRQC10	CDONEIRQC9	CDONEIRQC8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQC7	CDONEIRQC6	CDONEIRQC5	CDONEIRQC4	CDONEIRQC3	CDONEIRQC2	CDONEIRQC1	CDONEIRQC0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQC31~0 : Conversion Done Interrupt Clear bits

bit	Description
0	No effect.
1	Conversion Done Interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_CDONEIRQ0 register and CDONEIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register are cleared.

A/D Conversion Done Interrupt Clear Register (ADC12Bn_CDONEIRQC1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CDONEIRQC63	CDONEIRQC62	CDONEIRQC61	CDONEIRQC60	CDONEIRQC59	CDONEIRQC58	CDONEIRQC57	CDONEIRQC56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CDONEIRQC55	CDONEIRQC54	CDONEIRQC53	CDONEIRQC52	CDONEIRQC51	CDONEIRQC50	CDONEIRQC49	CDONEIRQC48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CDONEIRQC47	CDONEIRQC46	CDONEIRQC45	CDONEIRQC44	CDONEIRQC43	CDONEIRQC42	CDONEIRQC41	CDONEIRQC40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CDONEIRQC39	CDONEIRQC38	CDONEIRQC37	CDONEIRQC36	CDONEIRQC35	CDONEIRQC34	CDONEIRQC33	CDONEIRQC32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CDONEIRQC63~32 : Conversion Done Interrupt Clear bits

bit	Description
0	No effect.
1	Conversion Done Interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_CDONEIRQ1 register and CDONEIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register are cleared.

5.8. Group Interrupted Interrupt Flag Registers (ADC12Bn_GRPIRQ0~1)

These registers contain the status of group interrupted interrupt flags for all 64 logical channels.

REGISTER_NAME	ADC12Bn_GRPIRQi (i = 0~1)
OFFSET	0x0318 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Group Interrupted Interrupt Flag Register (ADC12Bn_GRPIRQ0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQ31	GRPIRQ30	GRPIRQ29	GRPIRQ28	GRPIRQ27	GRPIRQ26	GRPIRQ25	GRPIRQ24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQ23	GRPIRQ22	GRPIRQ21	GRPIRQ20	GRPIRQ19	GRPIRQ18	GRPIRQ17	GRPIRQ16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQ15	GRPIRQ14	GRPIRQ13	GRPIRQ12	GRPIRQ11	GRPIRQ10	GRPIRQ9	GRPIRQ8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQ7	GRPIRQ6	GRPIRQ5	GRPIRQ4	GRPIRQ3	GRPIRQ2	GRPIRQ1	GRPIRQ0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQ31~0 : Group Interrupted Interrupt flags

Group Interrupted Interrupt flags for the case the group gets interrupted and stopped before finishing.

bit	Description
0	Group interrupted interrupt not detected.
1	Group interrupted interrupt detected.

The bit is set to "1" if following conditions are fulfilled:

- The corresponding trigger status flags of the channel (ADC12Bn_CHSTAT0~31.TRGST and ADC12Bn_TRGST0.TRGST31~0) are set to "1"
- The channel is not first in the group i.e. the trigger type bits TRGTYP are set to "10" in the corresponding ADC12Bn_CHCTRL0~31 register
- The channel did not win arbitration for the next conversion, i.e. there was a channel with active trigger status and higher priority.

For multiple conversion logical channels, this bit is set to "1" also in the case the multiple conversions are started and they are interrupted before the last conversion is performed.

This bit is cleared by writing "1" to the corresponding ADC12Bn_GRPIRQC0 bits.

This bit is identical to the GRPIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 registers

Group interrupted Interrupt Flag Register (ADC12Bn_GRPIRQ1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQ63	GRPIRQ62	GRPIRQ61	GRPIRQ60	GRPIRQ59	GRPIRQ58	GRPIRQ57	GRPIRQ56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQ55	GRPIRQ54	GRPIRQ53	GRPIRQ52	GRPIRQ51	GRPIRQ50	GRPIRQ49	GRPIRQ48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQ47	GRPIRQ46	GRPIRQ45	GRPIRQ44	GRPIRQ43	GRPIRQ42	GRPIRQ41	GRPIRQ40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQ39	GRPIRQ38	GRPIRQ37	GRPIRQ36	GRPIRQ35	GRPIRQ34	GRPIRQ33	GRPIRQ32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQ63~32 : Group Interrupted Interrupt flags

Group Interrupted Interrupt flags for the case the group gets interrupted and stopped before finishing.

bit	Description
0	Group interrupted interrupt not detected.
1	Group interrupted interrupt detected.

The bit is set to "1" if following conditions are fulfilled:

- The corresponding trigger status flags of the channel (ADC12Bn_CHSTAT32~63.TRGST and ADC12Bn_TRGST1.TRGST63~32) are set to "1"
- The channel is not first in the group i.e. the trigger type bits TRGTYP are set to "10" in the corresponding ADC12Bn_CHCTRL32~63 register
- The channel did not win arbitration for the next conversion, i.e. there was a channel with active trigger status and higher priority.

For multiple conversion logical channels, this bit is set to "1" also in the case the multiple conversions are started and they are interrupted before the last conversion is performed.

This bit is cleared by writing "1" to the corresponding ADC12Bn_GRPIRQC1 bits.

This bit is identical to the GRPIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 registers

5.9. Group Interrupted Interrupt Enable Registers (ADC12Bn_GRPIRQE0~1)

These registers contain enable bits for all 64 logical channels, dedicated to the generation of group interrupted interrupt.

REGISTER_NAME	ADC12Bn_GRPIRQEi (i = 0~1)
OFFSET	0x0320 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Group interrupted Interrupt Enable Register (ADC12Bn_GRPIRQE0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQE31	GRPIRQE30	GRPIRQE29	GRPIRQE28	GRPIRQE27	GRPIRQE26	GRPIRQE25	GRPIRQE24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQE23	GRPIRQE22	GRPIRQE21	GRPIRQE20	GRPIRQE19	GRPIRQE18	GRPIRQE17	GRPIRQE16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQE15	GRPIRQE14	GRPIRQE13	GRPIRQE12	GRPIRQE11	GRPIRQE10	GRPIRQE9	GRPIRQE8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQE7	GRPIRQE6	GRPIRQE5	GRPIRQE4	GRPIRQE3	GRPIRQE2	GRPIRQE1	GRPIRQE0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQE31~0 : Group Interrupted Interrupt Enable bits

bit	Description
0	Group interrupted interrupt disabled.
1	Group interrupted interrupt enabled.

Group interrupted interrupt is issued when this bit is "1" and the corresponding interrupt flags (ADC12Bn_GRPIRQ0.GRPIRQ31~0 and ADC12Bn_CHSTAT0~31.GRPIRQ) are set to "1".

Group interrupted Interrupt Enable Register (ADC12Bn_GRPIRQE1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQE63	GRPIRQE62	GRPIRQE61	GRPIRQE60	GRPIRQE59	GRPIRQE58	GRPIRQE57	GRPIRQE56
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQE55	GRPIRQE54	GRPIRQE53	GRPIRQE52	GRPIRQE51	GRPIRQE50	GRPIRQE49	GRPIRQE48
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQE47	GRPIRQE46	GRPIRQE45	GRPIRQE44	GRPIRQE43	GRPIRQE42	GRPIRQE41	GRPIRQE40
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQE39	GRPIRQE38	GRPIRQE37	GRPIRQE36	GRPIRQE35	GRPIRQE34	GRPIRQE33	GRPIRQE32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQE63~32 : Group Interrupted Interrupt Enable bits

bit	Description
0	Group interrupted interrupt disabled.
1	Group interrupted interrupt enabled.

Group interrupted interrupt is issued when this bit is "1" and the corresponding interrupt flags (ADC12Bn_GRPIRQ1.GRPIRQ63~32 and ADC12Bn_CHSTAT32~63.GRPIRQ) are set to "1".

5.10. Group Interrupted Interrupt Clear Registers (ADC12Bn_GRPIRQC0~1)

These registers contain bits for clearing corresponding group interrupted interrupt flags in the ADC12Bn_GRPIRQ0~1 registers. The 64 bits are assigned to 64 logical channels.

REGISTER_NAME	ADC12Bn_GRPIRQCi (i = 0~1)
OFFSET	0x0328 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Group interrupted Interrupt Clear Register (ADC12Bn_GRPIRQC0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQC31	GRPIRQC30	GRPIRQC29	GRPIRQC28	GRPIRQC27	GRPIRQC26	GRPIRQC25	GRPIRQC24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQC23	GRPIRQC22	GRPIRQC21	GRPIRQC20	GRPIRQC19	GRPIRQC18	GRPIRQC17	GRPIRQC16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQC15	GRPIRQC14	GRPIRQC13	GRPIRQC12	GRPIRQC11	GRPIRQC10	GRPIRQC9	GRPIRQC8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQC7	GRPIRQC6	GRPIRQC5	GRPIRQC4	GRPIRQC3	GRPIRQC2	GRPIRQC1	GRPIRQC0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQC31~0 : Group Interrupted Clear bits

bit	Description
0	No effect.
1	Group interrupted interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_GRPIRQ0 register and GRPIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register are cleared.

Group interrupted Interrupt Clear Register (ADC12Bn_GRPIRQC1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	GRPIRQC63	GRPIRQC62	GRPIRQC61	GRPIRQC60	GRPIRQC59	GRPIRQC58	GRPIRQC57	GRPIRQC56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GRPIRQC55	GRPIRQC54	GRPIRQC53	GRPIRQC52	GRPIRQC51	GRPIRQC50	GRPIRQC49	GRPIRQC48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GRPIRQC47	GRPIRQC46	GRPIRQC45	GRPIRQC44	GRPIRQC43	GRPIRQC42	GRPIRQC41	GRPIRQC40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	GRPIRQC39	GRPIRQC38	GRPIRQC37	GRPIRQC36	GRPIRQC35	GRPIRQC34	GRPIRQC33	GRPIRQC32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] GRPIRQC63~32 : Group Interrupted Clear bits

bit	Description
0	No effect.
1	Group interrupted interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_GRPIRQ1 register and GRPIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register are cleared.

5.11. Range Comparator Interrupt Flag Registers (ADC12Bn_RCIRQ0~1)

Range Comparator Interrupt Flag Registers ADC12Bn_RCIRQ0~1 contain the status of range comparator interrupt flags for all 64 logical channels.

REGISTER_NAME	ADC12Bn_RCIRQi (i = 0~1)
OFFSET	0x0330 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Range Comparator Interrupt Flag Register (ADC12Bn_RCIRQ0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQ31	RCIRQ30	RCIRQ29	RCIRQ28	RCIRQ27	RCIRQ26	RCIRQ25	RCIRQ24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQ23	RCIRQ22	RCIRQ21	RCIRQ20	RCIRQ19	RCIRQ18	RCIRQ17	RCIRQ16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQ15	RCIRQ14	RCIRQ13	RCIRQ12	RCIRQ11	RCIRQ10	RCIRQ9	RCIRQ8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQ7	RCIRQ6	RCIRQ5	RCIRQ4	RCIRQ3	RCIRQ2	RCIRQ1	RCIRQ0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQ31~0 : Range Comparator Interrupt flags

bit	Description
0	Range comparator interrupt not detected.
1	Range comparator interrupt detected.

This flag shows that an outside range or inside range condition has been found on the corresponding logical channel.

This bit is set under the following conditions:

- The range comparison for this channel is enabled ADC12Bn_CHCTRL0~31.RCEN is set
- The conversion of the logical channel is just finished
- An interrupt condition is met (see Table 5-1)

This bit is cleared by writing "1" to the corresponding ADC12Bn_RCIRQC0.RCIRQC bit.

This bit is identical to the RCIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register.

Range Comparator Interrupt Flag Register (ADC12Bn_RCIRQ1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQ63	RCIRQ62	RCIRQ61	RCIRQ60	RCIRQ59	RCIRQ58	RCIRQ57	RCIRQ56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQ55	RCIRQ54	RCIRQ53	RCIRQ52	RCIRQ51	RCIRQ50	RCIRQ49	RCIRQ48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQ47	RCIRQ46	RCIRQ45	RCIRQ44	RCIRQ43	RCIRQ42	RCIRQ41	RCIRQ40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQ39	RCIRQ38	RCIRQ37	RCIRQ36	RCIRQ35	RCIRQ34	RCIRQ33	RCIRQ32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQ63~32 : Range Comparator Interrupt flags

bit	Description
0	Range comparator interrupt not detected.
1	Range comparator interrupt detected.

This flag shows that an outside range or inside range condition has been found on the corresponding logical channel.

This bit is set under the following conditions:

- The range comparison for this channel is enabled ADC12Bn_CHCTRL32~63.RCEN is set
- The conversion of the logical channel is just finished
- An interrupt condition is met (see Table 5-1)

This bit is cleared by writing "1" to the corresponding ADC12Bn_RCIRQC1.RCIRQC bit.

This bit is identical to the RCIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register.

Table 5-1 Range Comparator Interrupt Condition

Mode	Inverted Range Selection ADC12Bn_CHCTRL0~63.RCINVSEL	Conversion Result above Upper Threshold	Conversion Result below Lower Threshold	Interrupt Condition
outside range	0	1	x	INT condition: above range, ADC12Bn_RCOTF0~1.RCOTF and ADC12Bn_CHSTAT0~63.RCOTF are set.
		0	0	-
		0	1	INT condition: below range, ADC12Bn_RCOTF0~1.RCOTF and ADC12Bn_CHSTAT0~63.RCOTF are cleared.
inside range	1	1	x	-
		0	0	INT condition: inside range
		0	1	-

5.12. Range Comparator Interrupt Enable Registers (ADC12Bn_RCIRQE0~1)

These registers contain enable bits for all 64 logical channels, dedicated to the generation of range comparator interrupt.

REGISTER_NAME	ADC12Bn_RCIRQEi (i = 0~1)
OFFSET	0x0338 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Range Comparator Interrupt Enable Register (ADC12Bn_RCIRQE0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQE31	RCIRQE30	RCIRQE29	RCIRQE28	RCIRQE27	RCIRQE26	RCIRQE25	RCIRQE24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQE23	RCIRQE22	RCIRQE21	RCIRQE20	RCIRQE19	RCIRQE18	RCIRQE17	RCIRQE16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQE15	RCIRQE14	RCIRQE13	RCIRQE12	RCIRQE11	RCIRQE10	RCIRQE9	RCIRQE8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQE7	RCIRQE6	RCIRQE5	RCIRQE4	RCIRQE3	RCIRQE2	RCIRQE1	RCIRQE0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQE31~0 : Range Comparator Interrupt Enable bits

bit	Description
0	Range comparator interrupt disabled.
1	Range comparator interrupt enabled

Range comparator interrupt is issued when this bit is "1" and the corresponding interrupt flags ADC12Bn_RCIRQ0.RCIRQ31~0 and ADC12Bn_CHSTAT0~31.RCIRQ are set to "1".

Range Comparator Interrupt Enable Register (ADC12Bn_RCIRQE1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQE63	RCIRQE62	RCIRQE61	RCIRQE60	RCIRQE59	RCIRQE58	RCIRQE57	RCIRQE56
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQE55	RCIRQE54	RCIRQE53	RCIRQE52	RCIRQE51	RCIRQE50	RCIRQE49	RCIRQE48
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQE47	RCIRQE46	RCIRQE45	RCIRQE44	RCIRQE43	RCIRQE42	RCIRQE41	RCIRQE40
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQE39	RCIRQE38	RCIRQE37	RCIRQE36	RCIRQE35	RCIRQE34	RCIRQE33	RCIRQE32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQE63~32 : Range Comparator Interrupt Enable bits

bit	Description
0	Range comparator interrupt disabled.
1	Range comparator interrupt enabled

Range comparator interrupt is issued when this bit is "1" and the corresponding interrupt flags ADC12Bn_RCIRQ1.RCIRQ63~32 and ADC12Bn_CHSTAT32~63.RCIRQ are set to "1".

5.13. Range Comparator Interrupt Clear Registers (ADC12Bn_RCIRQC0~1)

These registers contain bits for clearing corresponding range comparator interrupt flags in the ADC12Bn_RCIRQ0~1 registers. The 64 bits are assigned to 64 logical channels.

REGISTER_NAME	ADC12Bn_RCIRQCi (i = 0~1)
OFFSET	0x0340 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Range Comparator Interrupt Clear Register (ADC12Bn_RCIRQC0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQC31	RCIRQC30	RCIRQC29	RCIRQC28	RCIRQC27	RCIRQC26	RCIRQC25	RCIRQC24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQC23	RCIRQC22	RCIRQC21	RCIRQC20	RCIRQC19	RCIRQC18	RCIRQC17	RCIRQC16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQC15	RCIRQC14	RCIRQC13	RCIRQC12	RCIRQC11	RCIRQC10	RCIRQC9	RCIRQC8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQC7	RCIRQC6	RCIRQC5	RCIRQC4	RCIRQC3	RCIRQC2	RCIRQC1	RCIRQC0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQC31~0 : Range Comparator Interrupt Clear bits

bit	Description
0	No effect.
1	Range comparator interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_RCIRQ0 register and RCIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register are cleared.

Range Comparator Interrupt Clear Register (ADC12Bn_RCIRQC1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCIRQC63	RCIRQC62	RCIRQC61	RCIRQC60	RCIRQC59	RCIRQC58	RCIRQC57	RCIRQC56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCIRQC55	RCIRQC54	RCIRQC53	RCIRQC52	RCIRQC51	RCIRQC50	RCIRQC49	RCIRQC48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCIRQC47	RCIRQC46	RCIRQC45	RCIRQC44	RCIRQC43	RCIRQC42	RCIRQC41	RCIRQC40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCIRQC39	RCIRQC38	RCIRQC37	RCIRQC36	RCIRQC35	RCIRQC34	RCIRQC33	RCIRQC32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCIRQC63~32 : Range Comparator Interrupt Clear bits

bit	Description
0	No effect.
1	Range comparator interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_RCIRQ1 register and RCIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register are cleared.

5.14. Pulse Counter Interrupt Flag Registers (ADC12Bn_PCIRQ0~1)

Pulse Comparator Interrupt Flag Registers ADC12Bn_PCIRQ0~1 contain the status of pulse counter interrupt flags for all 64 logical channels.

REGISTER_NAME	ADC12Bn_PCIRQi (i = 0~1)
OFFSET	0x0348 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Pulse Counter Interrupt Flag Register (ADC12Bn_PCIRQ0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQ31	PCIRQ30	PCIRQ29	PCIRQ28	PCIRQ27	PCIRQ26	PCIRQ25	PCIRQ24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQ23	PCIRQ22	PCIRQ21	PCIRQ20	PCIRQ19	PCIRQ18	PCIRQ17	PCIRQ16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQ15	PCIRQ14	PCIRQ13	PCIRQ12	PCIRQ11	PCIRQ10	PCIRQ9	PCIRQ8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQ7	PCIRQ6	PCIRQ5	PCIRQ4	PCIRQ3	PCIRQ2	PCIRQ1	PCIRQ0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQ31~0 : Pulse Counter Interrupt flags

bit	Description
0	Pulse counter interrupt not detected.
1	Pulse counter interrupt detected.

This register returns the status of the pulse counter interrupt flag which is set when positive counter ADC12Bn_PCCTRL0~31.PCTPCT of the corresponding logical channel decrements to zero. The positive counter and negative counter are stopped as long as the pulse counter interrupt flag of the appropriate channel is set.

This bit is cleared by writing "1" to the corresponding bit in the ADC12Bn_PCIRQC0 register.

This bit is identical to the PCIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register.

Pulse Counter Interrupt Flag Register (ADC12Bn_PCIRQ1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQ63	PCIRQ62	PCIRQ61	PCIRQ60	PCIRQ59	PCIRQ58	PCIRQ57	PCIRQ56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQ55	PCIRQ54	PCIRQ53	PCIRQ52	PCIRQ51	PCIRQ50	PCIRQ49	PCIRQ48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQ47	PCIRQ46	PCIRQ45	PCIRQ44	PCIRQ43	PCIRQ42	PCIRQ41	PCIRQ40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQ39	PCIRQ38	PCIRQ37	PCIRQ36	PCIRQ35	PCIRQ34	PCIRQ33	PCIRQ32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQ63~32 : Pulse Counter Interrupt flags

bit	Description
0	Pulse counter interrupt not detected.
1	Pulse counter interrupt detected.

This register returns the status of the pulse counter interrupt flag which is set when positive counter ADC12Bn_PCCTRL32~63.PCTPCT of the corresponding logical channel decrements to zero. The positive counter and negative counter are stopped as long as the pulse counter interrupt flag of the appropriate channel is set.

This bit is cleared by writing "1" to the corresponding bit in the ADC12Bn_PCIRQC1 register.

This bit is identical to the PCIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register.

5.15. Pulse Counter Interrupt Enable Registers (ADC12Bn_PCIRQE0~1)

These registers contain enable bits for all 64 logical channels, dedicated to the generation of pulse counter interrupt.

REGISTER_NAME	ADC12Bn_PCIRQEi (i = 0~1)
OFFSET	0x0350 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Pulse Counter Interrupt Enable Register (ADC12Bn_PCIRQE0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQE31	PCIRQE30	PCIRQE29	PCIRQE28	PCIRQE27	PCIRQE26	PCIRQE25	PCIRQE24
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQE23	PCIRQE22	PCIRQE21	PCIRQE20	PCIRQE19	PCIRQE18	PCIRQE17	PCIRQE16
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQE15	PCIRQE14	PCIRQE13	PCIRQE12	PCIRQE11	PCIRQE10	PCIRQE9	PCIRQE8
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQE7	PCIRQE6	PCIRQE5	PCIRQE4	PCIRQE3	PCIRQE2	PCIRQE1	PCIRQE0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQE31~0 : Pulse Counter Interrupt Enable bits

bit	Description
0	Pulse counter interrupt is disabled.
1	Pulse counter interrupt is enabled.

Pulse counter interrupt is issued when this bit is "1" and the corresponding interrupt flags ADC12Bn_PCIRQ0.PCIRQ31~0 and ADC12Bn_CHSTAT0~31.PCIRQ are set to "1".

Pulse Counter Interrupt Enable Register (ADC12Bn_PCIRQE1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQE63	PCIRQE62	PCIRQE61	PCIRQE60	PCIRQE59	PCIRQE58	PCIRQE57	PCIRQE56
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQE55	PCIRQE54	PCIRQE53	PCIRQE52	PCIRQE51	PCIRQE50	PCIRQE49	PCIRQE48
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQE47	PCIRQE46	PCIRQE45	PCIRQE44	PCIRQE43	PCIRQE42	PCIRQE41	PCIRQE40
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQE39	PCIRQE38	PCIRQE37	PCIRQE36	PCIRQE35	PCIRQE34	PCIRQE33	PCIRQE32
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQE63~32 : Pulse Counter Interrupt Enable bits

bit	Description
0	Pulse counter interrupt is disabled.
1	Pulse counter interrupt is enabled.

Pulse counter interrupt is issued when this bit is "1" and the corresponding interrupt flags ADC12Bn_PCIRQ1.PCIRQ63~32 and ADC12Bn_CHSTAT32~63.PCIRQ are set to "1".

5.16. Pulse Counter Interrupt Clear Registers (ADC12Bn_PCIRQC0~1)

These registers contain bits for clearing corresponding pulse counter interrupt flags in the ADC12Bn_PCIRQ0~1 registers. The 64 bits are assigned to 64 logical channels

REGISTER_NAME	ADC12Bn_PCIRQCi (i = 0~1)
OFFSET	0x0358 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Pulse Counter Interrupt Clear Register (ADC12Bn_PCIRQC0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQC31	PCIRQC30	PCIRQC29	PCIRQC28	PCIRQC27	PCIRQC26	PCIRQC25	PCIRQC24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQC23	PCIRQC22	PCIRQC21	PCIRQC20	PCIRQC19	PCIRQC18	PCIRQC17	PCIRQC16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQC15	PCIRQC14	PCIRQC13	PCIRQC12	PCIRQC11	PCIRQC10	PCIRQC9	PCIRQC8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQC7	PCIRQC6	PCIRQC5	PCIRQC4	PCIRQC3	PCIRQC2	PCIRQC1	PCIRQC0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQC31~0 : ADC Pulse Counter Interrupt Clear bits

bit	Description
0	No effect.
1	Pulse counter interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_PCIRQ0 register and PCIRQ bit in the corresponding ADC12Bn_CHSTAT0~31 register are cleared.

Additionally, the corresponding positive and negative counter (ADC12Bn_PCCTRL0~31.PCTPCT and ADC12Bn_PCCTRL0~31.PCTNCT) are reloaded with their reload values defined in the ADC12Bn_PCCTRL0~31.PCTPRL and ADC12Bn_PCCTRL0~31.PCTNRL.

Pulse Counter Interrupt Clear Register (ADC12Bn_PCIRQC1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PCIRQC63	PCIRQC62	PCIRQC61	PCIRQC60	PCIRQC59	PCIRQC58	PCIRQC57	PCIRQC56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PCIRQC55	PCIRQC54	PCIRQC53	PCIRQC52	PCIRQC51	PCIRQC50	PCIRQC49	PCIRQC48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCIRQC47	PCIRQC46	PCIRQC45	PCIRQC44	PCIRQC43	PCIRQC42	PCIRQC41	PCIRQC40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCIRQC39	PCIRQC38	PCIRQC37	PCIRQC36	PCIRQC35	PCIRQC34	PCIRQC33	PCIRQC32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PCIRQC63~32 : ADC Pulse Counter Interrupt Clear bits

bit	Description
0	No effect.
1	Pulse counter interrupt cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_PCIRQ1 register and PCIRQ bit in the corresponding ADC12Bn_CHSTAT32~63 register are cleared.

Additionally, the corresponding positive and negative counter (ADC12Bn_PCCTRL32~63.PCTPCT and ADC12Bn_PCCTRL32~63.PCTNCT) are reloaded with their reload values defined in the ADC12Bn_PCCTRL32~63.PCTPRL and ADC12Bn_PCCTRL32~63.PCTNRL.

5.17. A/D Channel Trigger Status Flag Registers (ADC12Bn_TRGST0~1)

A/D Channel Trigger Status Flag Registers ADC12Bn_TRGST0~1 contain the status of conversion requests for all 64 logical channels.

REGISTER_NAME	ADC12Bn_TRGSTi (i = 0~1)
OFFSET	0x0360 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Channel Trigger Status Flag Register (ADC12Bn_TRGST0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGST31	TRGST30	TRGST29	TRGST28	TRGST27	TRGST26	TRGST25	TRGST24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGST23	TRGST22	TRGST21	TRGST20	TRGST19	TRGST18	TRGST17	TRGST16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGST15	TRGST14	TRGST13	TRGST12	TRGST11	TRGST10	TRGST9	TRGST8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGST7	TRGST6	TRGST5	TRGST4	TRGST3	TRGST2	TRGST1	TRGST0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGST31~0 : A/D Channel Trigger Status flags

bit	Description
0	No conversion request.
1	Conversion request is issued.

This bit is set if any conversion request, dedicated to the corresponding logical channel, occurs (software, hardware, trigger by completion of preceding channel or idle trigger). Moreover, the bit is set if:

- The group of the corresponding logical channel is interrupted which configured as "restart" channel (ADC12Bn_CHCTRL0~31.RSMRST = "10")
- The channel is the start channel of the group or the last already converted channel configured as "resume" channel (ADC12Bn_CHCTRL0~31.RSMRST = "01").

This bit is cleared under one of the following conditions:

- Completion of the logical channel conversion (the last conversion in the case of multiple conversion channels), at the same time the conversion done interrupt flag is set
- The corresponding group processing is interrupted and the channel is not configured as "resume" channel (ADC12Bn_CHCTRL0~31.RSMRST = "01"). Instead, the group interrupted interrupt flag is set
- Writing "1" to the corresponding bit in the ADC12Bn_TRGCL0 register or writing "1" to the corresponding ADC12Bn_CHCTRL0~31.TRGCL bit.

When setting and clearing of the bit takes place at the same time, clearing has priority.

This bit is identical to the TRGST bit in the corresponding ADC12Bn_CHSTAT0~31 register.

A/D Channel Trigger Status Flag Register (ADC12Bn_TRGST1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGST63	TRGST62	TRGST61	TRGST60	TRGST59	TRGST58	TRGST57	TRGST56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGST55	TRGST54	TRGST53	TRGST52	TRGST51	TRGST50	TRGST49	TRGST48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGST47	TRGST46	TRGST45	TRGST44	TRGST43	TRGST42	TRGST41	TRGST40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGST39	TRGST38	TRGST37	TRGST36	TRGST35	TRGST34	TRGST33	TRGST32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGST63~32 : A/D Channel Trigger Status flags

bit	Description
0	No conversion request.
1	Conversion request is issued.

This bit is set if any conversion request, dedicated to the corresponding logical channel, occurs (software, hardware, trigger by completion of preceding channel or idle trigger). Moreover, the bit is set if:

- The group of the corresponding logical channel is interrupted which configured as "restart" channel (ADC12Bn_CHCTRL32~63.RSMRST = "10")
- The channel is the start channel of the group or the last already converted channel configured as "resume" channel (ADC12Bn_CHCTRL32~63.RSMRST = "01").

This bit is cleared under one of the following conditions:

- Completion of the logical channel conversion, at the same time the conversion done interrupt flag is set
- The corresponding group processing is interrupted and the channel is not configured as "resume" channel (ADC12Bn_CHCTRL32~63.RSMRST = "01"). Instead, the group interrupted interrupt flag is set

- Writing "1" to the corresponding bit in the ADC12Bn_TRGCL1 register or writing "1" to the corresponding ADC12Bn_CHCTRL32~63.TRGCL bit.

When setting and clearing of the bit takes place at the same time, clearing has priority.

This bit is identical to the TRGST bit in the corresponding ADC12Bn_CHSTAT32~63 register.

5.18. A/D Channel Trigger Clear Registers (ADC12Bn_TRGCL0~1)

These registers contain bits for clearing corresponding A/D channel trigger status flags in the ADC12Bn_TRGST0~1 registers. The 64 bits are assigned to 64 logical channels.

REGISTER_NAME	ADC12Bn_TRGCLi (i = 0~1)
OFFSET	0x0368 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Channel Trigger Clear Register (ADC12Bn_TRGCL0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGCL31	TRGCL30	TRGCL29	TRGCL28	TRGCL27	TRGCL26	TRGCL25	TRGCL24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGCL23	TRGCL22	TRGCL21	TRGCL20	TRGCL19	TRGCL18	TRGCL17	TRGCL16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGCL15	TRGCL14	TRGCL13	TRGCL12	TRGCL11	TRGCL10	TRGCL9	TRGCL8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGCL7	TRGCL6	TRGCL5	TRGCL4	TRGCL3	TRGCL2	TRGCL1	TRGCL0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGCL31~0 : Trigger Status Clear bits

bit	Description
0	No effect.
1	A/D channel trigger status is cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_TRGST0 register and TRGST bit in the corresponding ADC12Bn_CHSTAT0~31 register are cleared.

If the corresponding trigger status is cleared during A/D conversion, the operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

- Forced stop is enabled (ADC12Bn_CTRL.FSMD = "1"): A/D conversion is stopped after interrupt operation.
- Forced stop is disabled (ADC12Bn_CTRL.FSMD = "0"): A/D conversion is not stopped (but A/D conversion result is not updated). Hence, when the next conversion is already requested, the wait time from trigger status flag clear to the start of next conversion can be up to the maximum A/D conversion period.

This bit is identical to the TRGCL bit in the corresponding ADC12Bn_CHCTRL0~31 register.

Note:

- *If forced stop is disabled (ADC12Bn_CTRL.FSMD = "0"), do not set trigger status flag again during the same conversion after the trigger status is cleared.*
  Please set again after the end timing of the cleared conversion.

A/D Channel Trigger Clear Register (ADC12Bn_TRGCL1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGCL63	TRGCL62	TRGCL61	TRGCL60	TRGCL59	TRGCL58	TRGCL57	TRGCL56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGCL55	TRGCL54	TRGCL53	TRGCL52	TRGCL51	TRGCL50	TRGCL49	TRGCL48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGCL47	TRGCL46	TRGCL45	TRGCL44	TRGCL43	TRGCL42	TRGCL41	TRGCL40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGCL39	TRGCL38	TRGCL37	TRGCL36	TRGCL35	TRGCL34	TRGCL33	TRGCL32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGCL63~32 : Trigger Status Clear bits

bit	Description
0	No effect.
1	A/D channel trigger status is cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_TRGST1 register and TRGST bit in the corresponding ADC12Bn_CHSTAT32~63 register are cleared.

If the corresponding trigger status is cleared during A/D conversion, the operation is dependent on the setting of forced stop mode (ADC12Bn_CTRL.FSMD).

- Forced stop is enabled (ADC12Bn_CTRL.FSMD = "1"): A/D conversion is stopped after interrupt operation.
- Forced stop is disabled (ADC12Bn_CTRL.FSMD = "0"): A/D conversion is not stopped (but A/D conversion result is not updated). Hence, when the next conversion is already requested, the wait time from trigger status flag clear to the start of next conversion can be up to the maximum A/D conversion period.

This bit is identical to the TRGCL bit in the corresponding ADC12Bn_CHCTRL32~63 register.

Note:

- *If forced stop is disabled (ADC12Bn_CTRL.FSMD = "0"), do not set trigger status flag again during the same conversion after the trigger status is cleared.*
  Please set again after the end timing of the cleared conversion.

5.19. A/D Channel Trigger Overrun Flag Registers (ADC12Bn_TRGOR0~1)

A/D Channel Trigger Overrun Flag Registers ADC12Bn_TRGOR0~1 register the possible trials to set already active (set to "1") trigger status for all 64 logical channels.

REGISTER_NAME	ADC12Bn_TRGORi (i = 0~1)
OFFSET	0x0378 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Channel Trigger Overrun Flag Register (ADC12Bn_TRGOR0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGOR31	TRGOR30	TRGOR29	TRGOR28	TRGOR27	TRGOR26	TRGOR25	TRGOR24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGOR23	TRGOR22	TRGOR21	TRGOR20	TRGOR19	TRGOR18	TRGOR17	TRGOR16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGOR15	TRGOR14	TRGOR13	TRGOR12	TRGOR11	TRGOR10	TRGOR9	TRGOR8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGOR7	TRGOR6	TRGOR5	TRGOR4	TRGOR3	TRGOR2	TRGOR1	TRGOR0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGOR31~0 : A/D Channel Trigger Overrun flags

bit	Description
0	No trigger overrun happened
1	Trigger overrun occurred

This bit is set to "1" under following conditions:

- Conversion request is issued although the corresponding trigger status bits ADC12Bn_TRGST0.TRGST and ADC12Bn_CHSTAT0~31.TRGST are already set to "1"
- Software and hardware trigger are issued at the same cycle and corresponding trigger type ADC12Bn_CHCTRL0~31.TRGTYP[1:0] is set to "01"

Writing "1" to the corresponding bit in the ADC12Bn_TRGORC0 register clears this bit.

A/D Channel Trigger Overrun Flag Register (ADC12Bn_TRGOR1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGOR63	TRGOR62	TRGOR61	TRGOR60	TRGOR59	TRGOR58	TRGOR57	TRGOR56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGOR55	TRGOR54	TRGOR53	TRGOR52	TRGOR51	TRGOR50	TRGOR49	TRGOR48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGOR47	TRGOR46	TRGOR45	TRGOR44	TRGOR43	TRGOR42	TRGOR41	TRGOR40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGOR39	TRGOR38	TRGOR37	TRGOR36	TRGOR35	TRGOR34	TRGOR33	TRGOR32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGOR63~32 : A/D Channel Trigger Overrun flags

bit	Description
0	No trigger overrun happened
1	Trigger overrun occurred

This bit is set to "1" under following conditions:

- Conversion request is issued although the corresponding trigger status bits ADC12Bn_TRGST1.TRGST and ADC12Bn_CHSTAT32~63.TRGST are already set to "1"
- Software and hardware trigger are issued at the same cycle and corresponding trigger type ADC12Bn_CHCTRL32~63.TRGTYP[1:0] is set to "01"

Writing "1" to the corresponding bit in the ADC12Bn_TRGORC1 register clears this bit.

5.20. A/D Channel Trigger Overrun Clear Registers (ADC12Bn_TRGORC0~1)

These registers contain bits for clearing corresponding A/D channel trigger overrun flags in the ADC12Bn_TRGOR0~1 registers. The 64 bits are assigned to 64 logical channels.

REGISTER_NAME	ADC12Bn_TRGORCi (i = 0~1)
OFFSET	0x0380 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

A/D Channel Trigger Overrun Clear Register (ADC12Bn_TRGORC0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGORC31	TRGORC30	TRGORC29	TRGORC28	TRGORC27	TRGORC26	TRGORC25	TRGORC24
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGORC23	TRGORC22	TRGORC21	TRGORC20	TRGORC19	TRGORC18	TRGORC17	TRGORC16
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGORC15	TRGORC14	TRGORC13	TRGORC12	TRGORC11	TRGORC10	TRGORC9	TRGORC8
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGORC7	TRGORC6	TRGORC5	TRGORC4	TRGORC3	TRGORC2	TRGORC1	TRGORC0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGORC31~0 : Trigger Overrun Clear bits

bit	Description
0	No effect.
1	Trigger overrun flag is cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_TRGOR0 register is cleared.

A/D Channel Trigger Overrun Clear Register (ADC12Bn_TRGORC1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TRGORC63	TRGORC62	TRGORC61	TRGORC60	TRGORC59	TRGORC58	TRGORC57	TRGORC56
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TRGORC55	TRGORC54	TRGORC53	TRGORC52	TRGORC51	TRGORC50	TRGORC49	TRGORC48
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TRGORC47	TRGORC46	TRGORC45	TRGORC44	TRGORC43	TRGORC42	TRGORC41	TRGORC40
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TRGORC39	TRGORC38	TRGORC37	TRGORC36	TRGORC35	TRGORC34	TRGORC33	TRGORC32
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TRGORC63~32 : Trigger Overrun Clear bits

bit	Description
0	No effect.
1	Trigger overrun flag is cleared.

When this bit is set to "1", the corresponding bit in the ADC12Bn_TRGOR1 register is cleared.

5.21. Range Comparator Over Threshold Flag Registers (ADC12Bn_RCOTF0~1)

The flag bits (ADC12Bn_RCOTF0~1) are set when the result of the range comparator is "outside range" and the converted value is above the value of the upper threshold register.

REGISTER_NAME	ADC12Bn_RCOTFi (i = 0~1)
OFFSET	0x0370 + i*4
ACCESS_SIZE	B H W
MULTIPLE	0:1
NUMERIC_TYPE	-
OTHER	-

Range Comparator Over Threshold Flag Register (ADC12Bn_RCOTF0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCOTF31	RCOTF30	RCOTF29	RCOTF28	RCOTF27	RCOTF26	RCOTF25	RCOTF24
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCOTF23	RCOTF22	RCOTF21	RCOTF20	RCOTF19	RCOTF18	RCOTF17	RCOTF16
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCOTF15	RCOTF14	RCOTF13	RCOTF12	RCOTF11	RCOTF10	RCOTF9	RCOTF8
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCOTF7	RCOTF6	RCOTF5	RCOTF4	RCOTF3	RCOTF2	RCOTF1	RCOTF0
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCOTF31~0 : Range Comparator Over Threshold flags

The flag is only applicable in "outside range" mode i.e. while the RCINVSEL bit in the corresponding ADC12Bn_CHCTRL0~31 register is "0". If a range comparator interrupt is signaled (corresponding bits ADC12Bn_RCIRQ0.RCIRQ31~0 = ADC12Bn_CHSTAT0~31.RCIRQ = "1"), this flag has the following meaning:

bit	Description
0	The conversion result is less than or equal to the upper threshold.
1	The conversion result is above the upper threshold.

This bit is updated only in the case the corresponding interrupt flag (ADC12Bn_RCIRQ0.RCIRQ, ADC12Bn_CHSTAT0~31.RCIRQ) has a rising edge.

This bit is identical to the RCOTF bit in the corresponding ADC12Bn_CHSTAT0~31 register.

Range Comparator Over Threshold Flag Register (ADC12Bn_RCOTF1)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RCOTF63	RCOTF62	RCOTF61	RCOTF60	RCOTF59	RCOTF58	RCOTF57	RCOTF56
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RCOTF55	RCOTF54	RCOTF53	RCOTF52	RCOTF51	RCOTF50	RCOTF49	RCOTF48
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RCOTF47	RCOTF46	RCOTF45	RCOTF44	RCOTF43	RCOTF42	RCOTF41	RCOTF40
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCOTF39	RCOTF38	RCOTF37	RCOTF36	RCOTF35	RCOTF34	RCOTF33	RCOTF32
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RCOTF63~32 : Range Comparator Over Threshold flags

The flag is only applicable in "outside range" mode i.e. while the RCINVSEL bit in the corresponding ADC12Bn_CHCTRL32~63 register is "0". If a range comparator interrupt is signaled (corresponding bits ADC12Bn_RCIRQ1.RCIRQ63~32 = ADC12Bn_CHSTAT32~63.RCIRQ = "1"), this flag has the following meaning:

bit	Description
0	The conversion result is less than or equal to the upper threshold.
1	The conversion result is above the upper threshold.

This bit is updated only in the case the corresponding interrupt flag (ADC12Bn_RCIRQ1.RCIRQ, ADC12Bn_CHSTAT32~63.RCIRQ) has a rising edge.

This bit is identical to the RCOTF bit in the corresponding ADC12Bn_CHSTAT32~63 register.

5.22. Conversion Done DMA Select Registers (ADC12Bn_CDDS0~3)

These four registers specify four logical channels whose conversion done interrupt flags can initiate DMA requests to transfer A/D conversion results to memory. Only ADC12Bn_CDDS0 register is described here. Other registers (ADC12Bn_CDDS1, ADC12Bn_CDDS2 and ADC12Bn_CDDS3) have identical bit fields.

REGISTER_NAME	ADC12Bn_CDDSi (i = 0~3)
OFFSET	0x0388 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:3
NUMERIC_TYPE	-
OTHER	-

Conversion Done DMA Select Register (ADC12Bn_CDDS0)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	CDCHEN	CDCHNUM[5]	CDCHNUM[4]	CDCHNUM[3]	CDCHNUM[2]	CDCHNUM[1]	CDCHNUM[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:7] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit6] CDCHEN : Channel conversion done DMA dedicated interrupt enable bit

bit	Description
0	Interrupt for triggering DMA request on conversion done interrupt flag (ADC12Bn_CHSTAT0~63.CDONEIRQ = ADC12Bn_CDONEIRQ0~1.CDONEIRQ = "1") of the logical channel defined by CDCHNUM bits is disabled.
1	Interrupt for triggering DMA request on conversion done interrupt flag (ADC12Bn_CHSTAT0~63.CDONEIRQ = ADC12Bn_CDONEIRQ0~1.CDONEIRQ = "1") of the logical channel defined by CDCHNUM bits is enabled.

[bit5:0] CDCHNUM[5:0] : Number of the logical channel selected for conversion done interrupt for triggering a DMA request

CDCHNUM[5:0]	Description
000000	Logical channel 0 selected.
000001	Logical channel 1 selected.
...	...
111111	Logical channel 63 selected.

The conversion done interrupts of the group last logical channels are good candidates for DMA burst setup, since the group result registers can be read linearly.

5.23. A/D Converter Comparison Time Setting Register (ADC12Bn_CT)

ADC12Bn_CT register specifies the comparison time of the A/D converter. It is not allowed to update the value of this register during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

REGISTER_NAME	ADC12Bn_CT
OFFSET	0x0390
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Comparison Time Setting Register (ADC12Bn_CT)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CT[15]	CT[14]	CT[13]	CT[12]	CT[11]	CT[10]	CT[9]	CT[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CT[7]	CT[6]	CT[5]	CT[4]	CT[3]	CT[2]	CT[1]	CT[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	1	0	0

[bit15:0] CT[15:0] : A/D Converter Comparison Time Setting bits

These bits specify the comparison phase time.

If [CT value < 4]: Comparison time = (CT value x 12 + 4) x Peripheral clock period

If [CT value >= 4]: Comparison time = (CT value x 13) x Peripheral clock period

Do not set CT value to 0.

For specific values of minimum and maximum comparison time, please refer to the Device Data Sheet.

5.24. A/D Converter Resumption Time Setting Register (ADC12Bn_RT)

ADC12Bn_RT register specifies the resumption time of the A/D converter. It is not allowed to update the value of this register during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

REGISTER_NAME	ADC12Bn_RT
OFFSET	0x0392
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Resumption Time Setting Register (ADC12Bn_RT)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RT[7]	RT[6]	RT[5]	RT[4]	RT[3]	RT[2]	RT[1]	RT[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	1	1	1	1	1	1

[bit15:8] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit7:0] RT[7:0] : A/D Converter Resumption Time Setting bits

These bits specify the resumption phase time (power-up wait time).

Please set RT value is longer than the specific values of maximum resumption time.

Do not set RT value to 0.

RT value >= Maximum resumption time / Peripheral clock period.

For specific values of maximum resumption time, please refer to the Device Data Sheet.

5.25. A/D Converter Sampling Time Setting Registers (ADC12Bn_ST0~3)

ADC12Bn_ST0~3 registers specify four different settings for the selection of the sampling time of the A/D converter. It is not allowed to update the value of these registers during A/D sampling operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1"). Only ADC12Bn_ST0 register is described here. Other registers (ADC12Bn_ST1, ADC12Bn_ST2 and ADC12Bn_ST3) have similar bit fields.

REGISTER_NAME	ADC12Bn_STi (i = 0~3)
OFFSET	0x0394 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:3
NUMERIC_TYPE	-
OTHER	-

A/D Converter Sampling Time Setting Register (ADC12Bn_ST0)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	ST[15]	ST[14]	ST[13]	ST[12]	ST[11]	ST[10]	ST[9]	ST[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	ST[7]	ST[6]	ST[5]	ST[4]	ST[3]	ST[2]	ST[1]	ST[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	1	0	1	1	0	0

[bit15:0] ST[15:0] : A/D Converter Sampling Time Setting bits

These bits specify one of the four available settings for A/D converter sampling phase time.

This time is selected for a logical channel conversion by configuring the corresponding logical channel register field ADC12Bn_CHCTRL0~63.SMTIME to "00".

Sampling time = ST value x Peripheral clock period

Do not set ST value below 6.

For specific values of minimum sampling time, please refer to the Device Data Sheet.

5.26. A/D Converter Offset Compensation Setting Register (ADC12Bn_OCV)

The global register ADC12Bn_OCV specifies the setting for offset compensation value.

REGISTER_NAME	ADC12Bn_OCV
OFFSET	0x039C
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Offset Compensation Setting Register (ADC12Bn_OCV)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	OCV[7]	OCV[6]	OCV[5]	OCV[4]	OCV[3]	OCV[2]	OCV[1]	OCV[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	1	0

[bit15:8] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit7:0] OCV[7:0] : Offset Compensation Value

The A/D Converter can be configured to convert the values of its reference voltages AVRH and AVRL in order to calculate the offset compensation value.

After A/D Converter calibration, the calculated offset compensation value must be written to this register.

For further explanation of A/D Converter calibration refer to section "A/D Converter Calibration" in chapter "3 Operation of A/D Converter".

5.27. A/D Converter Gain Compensation Setting Register (ADC12Bn_GCV)

This register ADC12Bn_GCV specifies the setting for gain compensation value.

REGISTER_NAME	ADC12Bn_GCV
OFFSET	0x039E
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Gain Compensation Setting Register (ADC12Bn_GCV)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved			GCV[4]	GCV[3]	GCV[2]	GCV[1]	GCV[0]
ACCESS_TYPE	R0,W0			R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	000			0	0	0	0	0

[bit15:5] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit4:0] GCV[4:0] : Gain Compensation Value

The A/D Converter can be configured to convert the values of its reference voltages AVRH and AVRL in order to calculate the gain compensation value.

After A/D Converter calibration, the calculated gain compensation value must be written to this register.

For further explanation of A/D Converter calibration refer to section "A/D Converter Calibration" in chapter "3 Operation of A/D Converter".

5.28. A/D Converter Global Control Register (ADC12Bn_CTRL)

The register configures the global control settings of the A/D Converter: A/D conversion resolution, enabling of the debug feature and the mode of ADC12Bn_STAT.ACH bits, forced stop mode and request, full range comparator mode, and power-down disable mode.

REGISTER_NAME	ADC12Bn_CTRL
OFFSET	0x03A0
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Global Control Register (ADC12Bn_CTRL)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PDDMD	FSTP	FRCMD	FSMD	ACHMD	DBGE	RES[1]	RES[0]
ACCESS_TYPE	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:8] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit7] PDDMD : Power-down disable mode

bit	Description
0	A/D converter goes to idle (power-down) state after A/D conversion finished (and next conversion is not requested).
1	A/D convertor does not go idle (power-down) state after A/D conversion finished (and next conversion is not requested). It can start A/D sampling without the resumption time.

Note:

- *At the following case, even if the power-down mode is disabled (PDDMD="1"), A/D converter is waited until resumption time.*
  - A/D converter is in idle (power-down) state.
  - PDDMD is change "0" to "1".

[bit6] FSTP : Forced stop

When the forced stop mode (FSMD = "1"):

bit	Description	
	Read	Write
0	The value is always "0"	No effect.
1		Request forced stop of A/D conversion.

When the "not" forced stop mode (FSMD = "0"):

"0" is always read from this bit. Writing "0" or "1" to this bit has no effect.

[bit5] FRCMD : Full Range Comparator mode

bit	Description
0	8-bit range comparator mode. ADC12Bn_RCOH and ADC12Bn_RCOL are used for 8-bit range comparator. ADC12Bn_FRCOH0~7 and ADC12Bn_FRCOL0~7 are not used.
1	12-bit range comparator mode. ADC12Bn_RCOH and ADC12Bn_RCOL are not used. ADC12Bn_FRCOH0~7 and ADC12Bn_FRCOL0~7 are used for 12-bit range comparator.

[bit4] FSMD : Forced stop mode

bit	Description
0	The forced stop mode is disabled. − An active A/D conversion cannot be interrupted. − Request forced stop of A/D conversion is disabled.
1	The forced stop mode is enabled. − An active A/D conversion can be interrupted. − Request forced stop of A/D conversion is enabled.

[bit3] ACHMD : ACH register mode

bit	Description
0	Direct ACH register mode. ADC12Bn_STAT.ACH shows the number of currently converted logical channel.
1	Latched ACH register mode. ADC12Bn_STAT.ACH shows the number of the logical channel whose conversion was finished last.

[bit2] DBGE : Debug Enable bit

bit	Description
0	Debug mode disabled
1	Debug mode enabled.

When this bit is set to "1" and the processor is in debug state, the A/D Converter completes the current conversion, but further conversion is stopped. When the processor leaves debug state or DBGE is set to "0", conversion continues with the next channel from where it had stopped.

[bit1:0] RES[1:0] : Resolution of A/D conversion

RES[1:0]	Description
x0	12-bit resolution
01	10-bit resolution
11	8-bit resolution.

Conversion result is stored in lower 10 bits of ADC12Bn_CD0~63 registers in case of 10-bit resolution and in lower 8 bits of ADC12Bn_CD0~63 registers if the 8-bit resolution is configured.

In case of 10-bit or 8-bit resolution, the lower 2 or 4 bits of the 12-bit conversion result are truncated (not rounded).

5.29. A/D Converter Global Status Register (ADC12Bn_STAT)

The register ADC12Bn_STAT is responsible for storing global status information of A/D Converter, such as currently converted channel and activity indication bit.

REGISTER_NAME	ADC12Bn_STAT
OFFSET	0x03A2
ACCESS_SIZE	B H W
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

A/D Converter Global Status Register (ADC12Bn_STAT)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	RX,WX							
PROT_TYPE								
INITIAL_VALUE	0x00							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	BUSY	ACH[5]	ACH[4]	ACH[3]	ACH[2]	ACH[1]	ACH[0]
ACCESS_TYPE	RX,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:7] Reserved : Reserved bits

Reading this bit returns an undefined.

Writing data to these bits has no effect on the operation.

[bit6] BUSY : A/D Converter Busy flag

bit	Description
0	A/D Converter is not active, i. e. no A/D conversion is ongoing.
1	A/D Converter is active, i. e. A/D conversion is ongoing.

■ Power-down disable mode (ADC12Bn_CTRL.PDDMD = "1"):

This bit set to "1" when first conversion request (trigger status set to "1") occurs (without A/D converter resumption time).
This bit clear to "0" when the all trigger status are cleared.

■ Power-down enable mode (ADC12Bn_CTRL.PDDMD = "0"):

This bit is set to "1" when first conversion request (trigger status set to "1") occurs and A/D Converter resumption time (power-up wait time configured as A/D converter resumption time register (ADC12Bn_RT)) elapses.
This bit clear to "0" when the all trigger status are cleared.

If there is no idle trigger type set for any of logical channels, after all conversion requests are processed A/D Converter goes to idle (power-down) state and BUSY flag becomes "0" until next conversion request appears and resumption time elapses.

In case that idle trigger type is set (ADC12Bn_CHCTRL0~63.TRGTYP[1:0] = "11"), there will be always at least one conversion request active and after BUSY flag is set first time to "1", it will not change.

[bit5:0] ACH[5:0] : Converted logical channel number

This bit field depends on ADC12Bn_CTRL.ACHMD field setting.

- If ACHMD is equal to "0", ACH represents currently converted logical channel number.
- If ACHMD is equal to "1", ACH represents the last logical channel number whose conversion has finished. ACH is updated at the end of the A/D conversion only in the case the corresponding trigger status (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST bits) is still "1".

5.30. Range Comparator Upper Threshold Registers (ADC12Bn_RCOH0~7)

When the 8-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "0"), these registers are used for 8-bit range comparator.

If the 12-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "1"), these registers are not used.

These registers specify the upper threshold values that can be selected for the 8-bit range comparator to which the output of the A/D Converter is compared. Only ADC12Bn_RCOH0 is described here. Other registers (ADC12Bn_RCOH1,..., ADC12Bn_RCOH6 and ADC12Bn_RCOH7) have similar bit fields.

REGISTER_NAME	ADC12Bn_RCOHi (i = 0~7)
OFFSET	0x03B1 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:7
NUMERIC_TYPE	-
OTHER	-

Range Comparator Upper Threshold Register 0 (ADC12Bn_RCOH0)

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCOH[7]	RCOH[6]	RCOH[5]	RCOH[4]	RCOH[3]	RCOH[2]	RCOH[1]	RCOH[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit7:0] RCOH[7:0] : Range Comparator Upper Threshold value

The RCOH bits define the upper comparison threshold of the range comparator 0.

The upper comparator compares the upper 8 bits of the A/D conversion result. If the value is higher than RCOH[7:0], then the conversion result is outside range.

Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields.

5.31. Range Comparator Lower Threshold Registers (ADC12Bn_RCOL0~7)

When the 8-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "0"), these registers are used for 8-bit range comparator.

If the 12-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "1"), these registers are not used.

These registers specify the lower threshold values that can be selected for the 8-bit range comparator to which the output of the A/D Converter is compared. Only ADC12Bn_RCOL0 is described here. Other registers (ADC12Bn_RCOL1,...,ADC12Bn_RCOL6, ADC12Bn_RCOL7) have similar bit fields.

REGISTER_NAME	ADC12Bn_RCOLi (i = 0~7)
OFFSET	0x03B0 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:7
NUMERIC_TYPE	-
OTHER	-

Range Comparator Lower Threshold Register 0 (ADC12Bn_RCOL0)

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RCOL[7]	RCOL[6]	RCOL[5]	RCOL[4]	RCOL[3]	RCOL[2]	RCOL[1]	RCOL[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7:0] RCOL[7:0] : Range Comparator Lower Threshold value

The RCOL bits define the lower comparison threshold of the range comparator 0.

The lower comparator compares the upper 8 bits of the A/D conversion result. If the value is lower than RCOL[7:0], then the conversion result is outside range.

Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields.

5.32. Full Range Comparator Upper Threshold Registers (ADC12Bn_FRCOH0~7)

When the 12-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "1"), these registers are used for 12-bit range comparator.

If the 8-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "0"), these registers are not used.

These registers specify the upper threshold values that can be selected for the 12-bit range comparator to which the output of the A/D Converter is compared. Only ADC12Bn_FRCOH0 is described here. Other registers (ADC12Bn_FRCOH1,..., ADC12Bn_FRCOH6 and ADC12Bn_FRCOH7) have similar bit fields.

REGISTER_NAME	ADC12Bn_FRCOHi (i = 0~7)
OFFSET	0x03F0 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:7
NUMERIC_TYPE	-
OTHER	-

Full Range Comparator Upper Threshold Register 0 (ADC12Bn_FRCOH0)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved				FRCOH[11]	FRCOH[10]	FRCOH[9]	FRCOH[8]
ACCESS_TYPE	R0,W0				R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0000				1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FRCOH[7]	FRCOH[6]	FRCOH[5]	FRCOH[4]	FRCOH[3]	FRCOH[2]	FRCOH[1]	FRCOH[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit15:12] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit11:0] FRCOH[11:0] : Full Range Comparator Upper Threshold value

The FRCOH bits define the upper comparison threshold of the range comparator 0.

The upper comparator compares the 12 bits, 10 bits or 8 bits of the A/D conversion result. ADC12Bn_CTRL.RES[1:0] define this resolution.

- If 12-bit resolution (ADC12Bn_CTRL.RES[1:0] = "x0"):

FRCOH[11:0]	Compares the 12 bits of A/D conversion result.

- If 10-bit resolution (ADC12Bn_CTRL.RES[1:0] = "01"):

FRCOH[11:10]	Not used for 10-bit range comparator.
FRCOH[9:0]	Compares the 10 bits of A/D conversion result.

- If 8-bit resolution (ADC12Bn_CTRL.RES[1:0] = "11"):

FRCOH[11:8]	Not used for 8-bit range comparator.
FRCOH[7:0]	Compares the 8 bits of A/D conversion result.

Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields.

5.33. Full Range Comparator Lower Threshold Registers (ADC12Bn_FRCOL0~7)

When the 12-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "1"), these registers are used for 12-bit range comparator.

If the 8-bit range comparator mode (ADC12Bn_CTRL.FRCMD = "0"), these registers are not used.

These registers specify the lower threshold values that can be selected for the 12-bit range comparator to which the output of the A/D Converter is compared. Only ADC12Bn_FRCOL0 is described here. Other registers (ADC12Bn_FRCOL1,..., ADC12Bn_FRCOL6 and ADC12Bn_FRCOL7) have similar bit fields.

REGISTER_NAME	ADC12Bn_FRCOLi (i = 0~7)
OFFSET	0x03D0 + i*2
ACCESS_SIZE	B H W
MULTIPLE	0:7
NUMERIC_TYPE	-
OTHER	-

Full Range Comparator Lower Threshold Register 0 (ADC12Bn_FRCOL0)

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved				FRCOL[11]	FRCOL[10]	FRCOL[9]	FRCOL[8]
ACCESS_TYPE	R0,W0				R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0000				0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FRCOL[7]	FRCOL[6]	FRCOL[5]	FRCOL[4]	FRCOL[3]	FRCOL[2]	FRCOL[1]	FRCOL[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit15:12] Reserved : Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit11:0] FRCOL[11:0] : Full Range Comparator Lower Threshold value

The FRCOL bits define the lower comparison threshold of the range comparator 0.

The lower comparator compares the 12 bits, 10 bits or 8 bits of the A/D conversion result. ADC12Bn_CTRL.RES[1:0] define this resolution.

- If 12-bit resolution (ADC12Bn_CTRL.RES[1:0] = "x0"):

FRCOL[11:0]	Compares the 12 bits of A/D conversion result.

- If 10-bit resolution (ADC12Bn_CTRL.RES[1:0] = "01"):

FRCOL[11:10]	Not used for 10-bit range comparator.
FRCOL[9:0]	Compares the 10 bits of A/D conversion result.

- If 8-bit resolution (ADC12Bn_CTRL.RES[1:0] = "11"):

FRCOL[11:8]	Not used for 8-bit range comparator.
FRCOL[7:0]	Compares the 8 bits of A/D conversion result.

Selection of one of eight available range comparator threshold values for the logical channel is configured by corresponding ADC12Bn_CHCTRL0~63.RCSEL[2:0] bit fields.

5.34. A/D Multiple Conversion Channel Control Registers (ADC12Bn_MCCTRL0~3)

These registers ADC12Bn_MCCTRL0~3 contain additional configuration bits for the first four logical channels (multiple conversion logical channels). They control number of conversions, multiple conversion interruption and A/D Converter calibration setup. Only ADC12Bn_MCCTRL0 is described here. Other registers (ADC12Bn_MCCTRL1, ADC12Bn_MCCTRL2 and ADC12Bn_MCCTRL3) have identical bit fields.

REGISTER_NAME	ADC12Bn_MCCTRLi (i = 0~3)
OFFSET	0x03C0 + i
ACCESS_SIZE	B H W
MULTIPLE	0:3
NUMERIC_TYPE	-
OTHER	-

A/D Multiple Conversion Channel Control Register (ADC12Bn_MCCTRL0)

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	AVRHSEL	AVRLSEL	ICIRQY	CNVNUM[3]	CNVNUM[2]	CNVNUM[1]	CNVNUM[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit7] Reserved : Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit6] AVRHSEL : A/D reference voltage AVRH selection bit

bit	Description
0	AVRH voltage is not selected for A/D conversion
1	AVRH voltage is selected for A/D conversion.

In case that both bits AVRHSEL and AVRLSEL bits are set to "1", AVRLSEL has higher priority and AVRL voltage is converted.

If any of AVRHSEL/AVRLSEL bits is set to "1", conversion of a regular analog input AN is not possible.

For further explanation of A/D Converter calibration refer to section "A/D Converter Calibration" in chapter "3 Operation of A/D Converter".

[bit5] AVRLSEL : A/D reference voltage AVRL selection bit

bit	Description
0	AVRL voltage is not selected for A/D conversion

1	AVRL voltage is selected for A/D conversion

In case that both bits AVRHSEL and AVRLSEL bits are set to "1", AVRLSEL has higher priority and AVRL voltage is converted.

If any of AVRHSEL/AVRLSEL bits is set to "1", conversion of a regular analog input AN is not possible.

For further explanation of A/D Converter calibration refer to section "A/D Converter Calibration" in chapter "3 Operation of A/D Converter".

[bit4] ICIRQY : Intra-channel interrupt ability for the multiple conversion logical channel

This bit determines if the multiple conversion logical channel can be interrupted between single conversions, in case the conversion request with higher priority is issued.

bit	Description
0	Multiple conversion logical channel can be interrupted between single conversions.
1	Multiple conversion logical channel cannot be interrupted between single conversions.

This bit has no effect if CNVNUM[3:0] is configured to "0000".

[bit3:0] CNVNUM[3:0] : Number of conversions for the multiple conversion channel

This bit field specifies the number of A/D conversions to be performed if the conversion request for the multiple conversion logical channel is issued.

CNVNUM[3:0]	Description
0000	1 conversion. With this setting multiple conversion logical channel behaves like any other logical channel.
0001	2 conversions.
0010	3 conversions.
...	...
1111	16 conversions

CNVNUM[3:0] are not allowed to update during A/D conversion operation (ADC12Bn_TRGST0~1.TRGST and ADC12Bn_CHSTAT0~63.TRGST="1").

5.35. A/D Multiple Conversion Channel Status Registers (ADC12Bn_MCSTAT0~3)

These registers contain the current status of finished conversion number for first four multiple conversion logical channels. Only ADC12Bn_MCSTAT0 is described here. Other registers (ADC12Bn_MCSTAT1, ADC12Bn_MCSTAT2 and ADC12Bn_MCSTAT3) have identical bit fields.

REGISTER_NAME	ADC12Bn_MCSTATi (i = 0~3)
OFFSET	0x03E0 + i
ACCESS_SIZE	B H W
MULTIPLE	0:3
NUMERIC_TYPE	-
OTHER	-

A/D Multiple Conversion Channel Status Register (ADC12Bn_MCSTAT0)

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved			MCCNT[4]	MCCNT[3]	MCCNT[2]	MCCNT[1]	MCCNT[0]
ACCESS_TYPE	RX,WX			R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	000			0	0	0	0	0

[bit7:5] Reserved : Reserved bits

Reading this bit returns an undefined.

Writing data to these bits has no effect on the operation.

[bit4:0] MCCNT[4:0] : Counter of the Multiple conversion

Conversion counter counting the number of already finished A/D conversions of the multiple conversion logical channel.

After the set event of the conversion done interrupt flag, the ADC12Bn_MCSTAT.MCCNT[4:0] are meaningless.

MCCNT[4:0]	Description
00000	No conversion is finished.
00001	1 conversion is finished.
...	...
10000	16 conversions are finished.
10001-11111	Reserved

CHAPTER 23: Partial Wakeup Control

This chapter explains the partial wakeup control function and its operations.

1. Overview
2. Configuration
3. Explanation of Operation
4. Registers
5. Precautions for Using this Device

CODE: PWU-JUPI-E3

1. Overview

This section provides an overview of the partial wakeup control.

Partial wakeup (PWU) mode is implemented in this device.

PWU mode is one of the PSS (Power Saving State) modes. In this mode, only the functions required for voltage monitoring by the A/D converter are operating.

Effective use of this function enables monitoring of voltage, such as in sensor output, with low-power consumption and without starting the CPU.

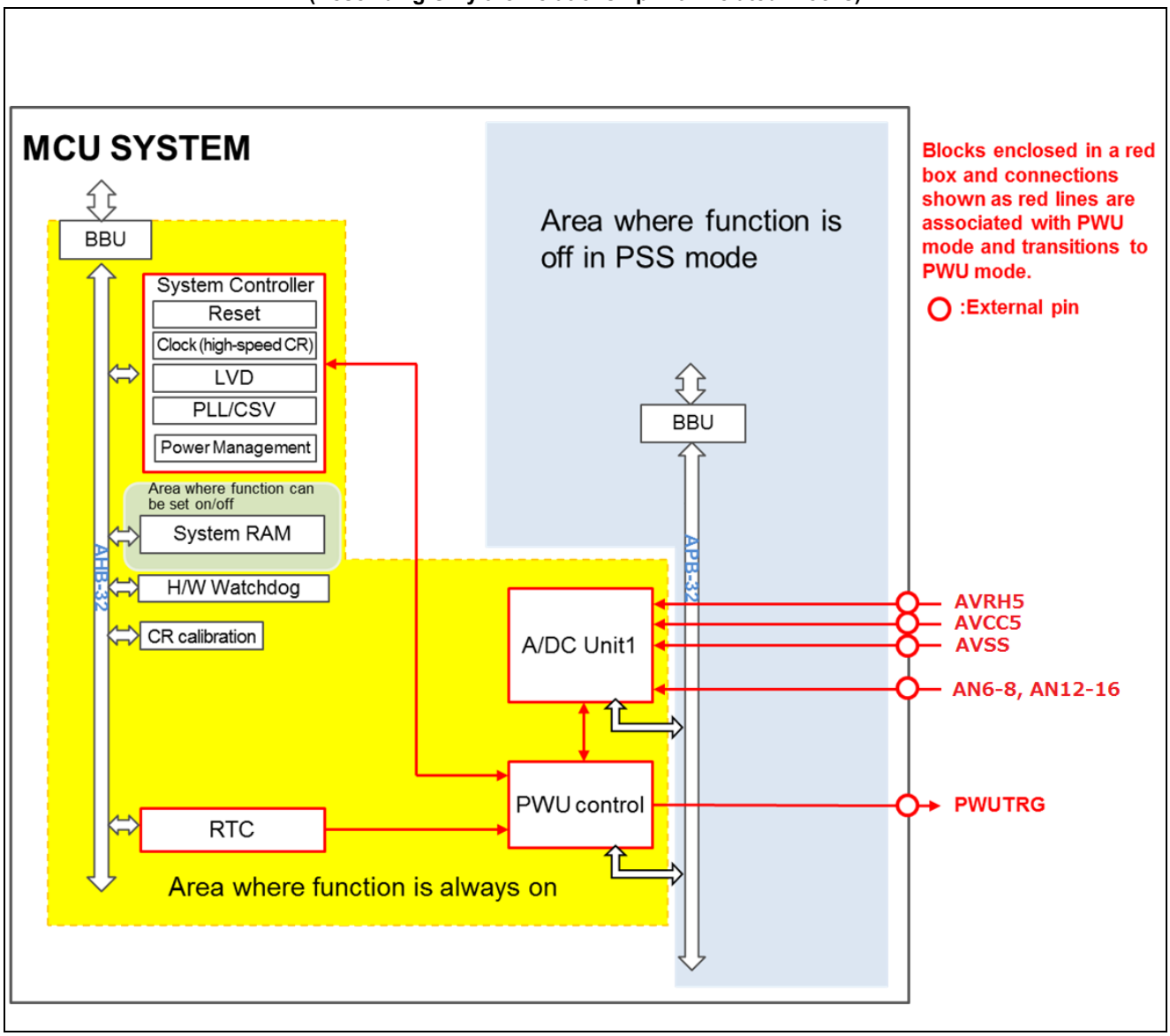
2. Configuration

This section explains the partial wakeup configuration.

Relationship between the entire system and the partial wakeup function

Figure 2-1 shows the relationship between the entire system and the partial wakeup function.

Figure 2-1 Relationship between the Entire System and the Partial Wakeup Function (Describing Only the Relationship with Related Blocks)



■ RTC

This block counts the cycles for transitioning to PWU mode. The cycles can be set in steps of 8 ms, between 8 and 64 ms. For the setting method, see "3. Explanation of Operation."

Note:

The term of 8ms is generated from the following calculation in RTC.
0.25[s] ÷ 32 = 7.8125[ms] (≈ 8[ms])
Moreover as RTC operates with low-speed CR (typical 100KHz) in PWU mode, the actual cycles for transitioning to PWU mode become as the following.

7.81[ms] (The description in this chapter is “8ms”)
15.62[ms] (The description in this chapter is “16ms”)
23.43[ms] (The description in this chapter is “24ms”)
31.24[ms] (The description in this chapter is “32ms”)
39.05[ms] (The description in this chapter is “40ms”)
46.86[ms] (The description in this chapter is “48ms”)
54.67[ms] (The description in this chapter is “56ms”)
62.48[ms] (The description in this chapter is “64ms”)

■ System controller

- This block manages the device state (PSS or RUN).
- The block turns on the high-speed CR oscillator when transitioning to PWU mode.

■ PWU control

- This block controls the high-speed CR oscillator, the "PWUTRG" pin output function, and A/DC Unit1.
- In PWU mode, the block outputs "H" from the "PWUTRG" pin.
- The block starts A/D conversion within a certain time after the PWUTRG pin outputs "H". The time until A/D conversion start can be set in steps of 50 us, between 50 and 5100 us.

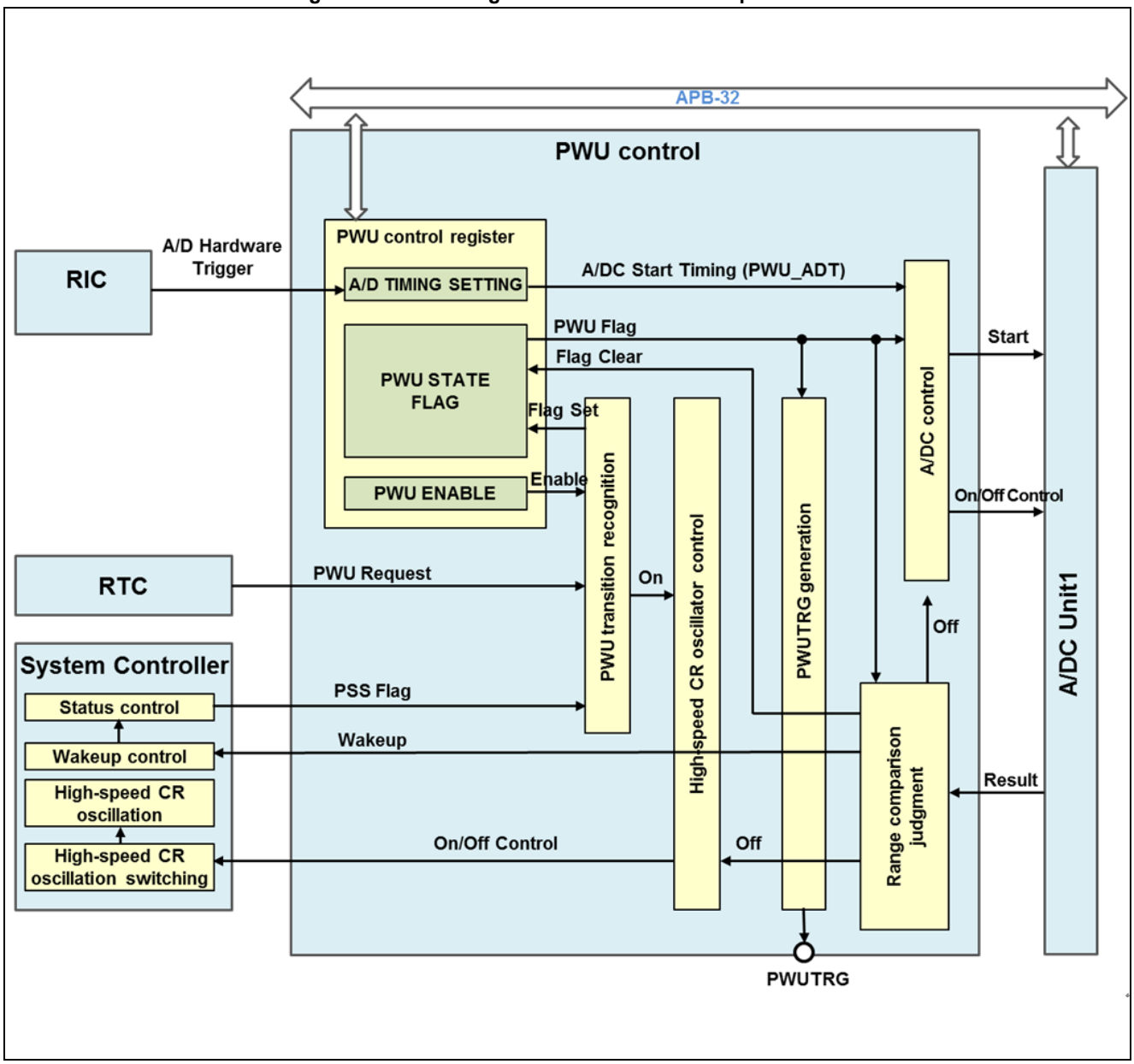
■ A/DC Unit1

- In PWU mode, the A/D converter can be used with the 8 channels from AN6-8 and AN12-16 out of all channels.
- In PWU mode, only the range comparison function of the A/D converter can be used. For details, see the chapter of "12/10/8-BIT Analog To Digital Converter" on this manual.

Block diagram of the partial wakeup function

Figure 2-2 is a block diagram of the partial wakeup function.

Figure 2-2 Block Diagram of the Partial Wakeup Function



■ PWU control register

The configuration includes the "partial wakeup mode enable bit", the "A/D conversion start time setting register", “A/D conversion request trigger control register” and “A/D conversion offset compensation value select register”. For details, see the chapter of “Partial Wakeup Control 4. Registers" on this manual.

■ PWU transition recognition

This block judges whether the conditions for transitioning to PWU mode are met. Also, the block turns on the high-speed CR oscillator.

■ High-speed CR oscillator control

This block is used for On/Off control of the high-speed CR oscillator in PWU mode.

■ PWUTRG generation

In PWU mode, this block outputs "H" from the "PWUTRG" pin.

■ A/DC control

This block is used to turn A/DC on or off and to adjust the timing of conversion start in PWU mode.

The block starts A/D conversion within a certain time after the PWUTRG pin outputs "H".

The time until A/D conversion start can be set in steps of 50 us, between 50 and 5100 us.

■ Range comparison judgment

This block determines, from an A/D conversion range comparison, whether to transition to RUN mode or PSS mode.

To transition to RUN mode, it generates a Wakeup signal.

After the completion of A/D conversion, the block generates a signal for turning off the high-speed CR oscillator and A/DC Unit1.

3. Explanation of Operation

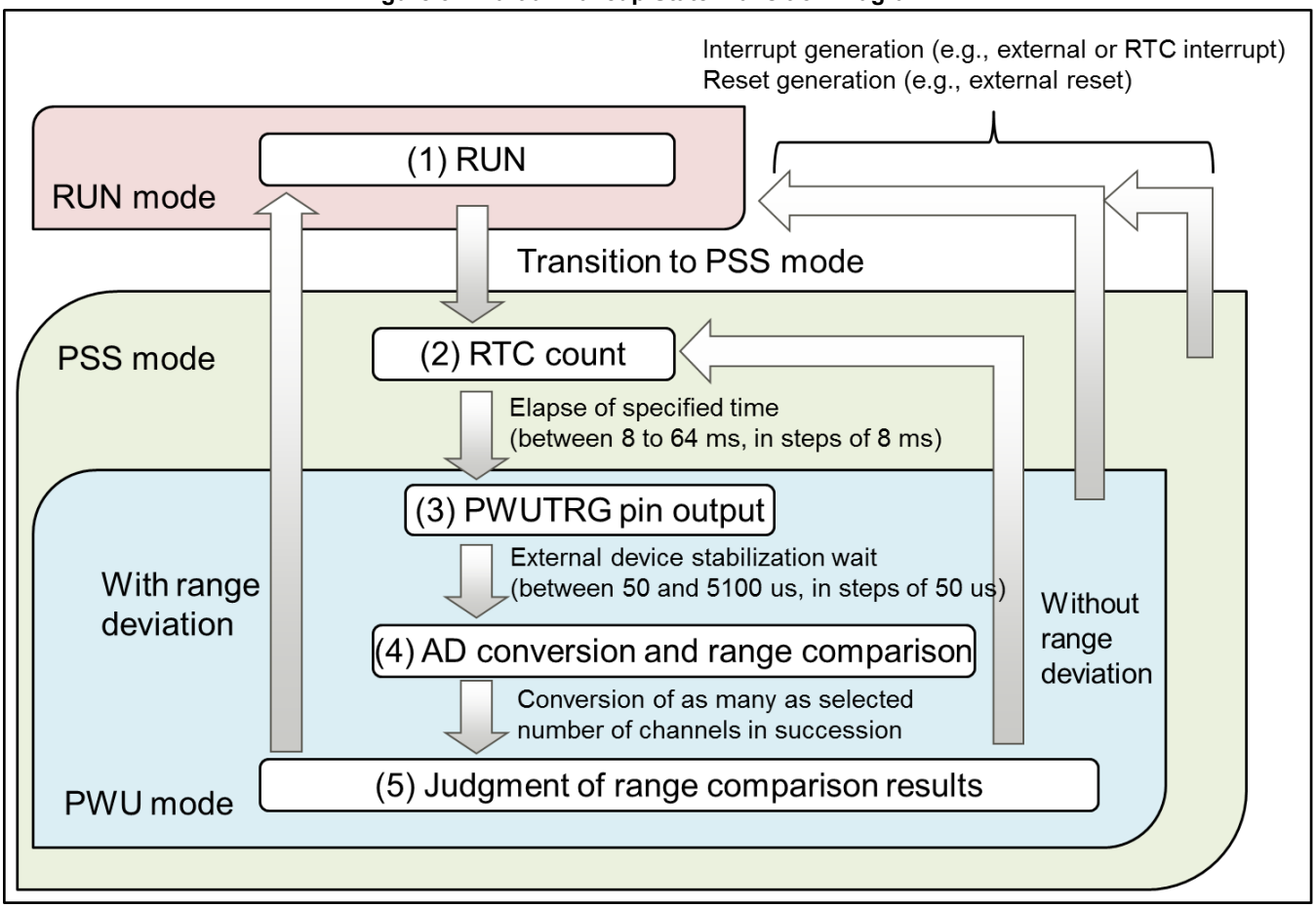
This section explains the operations of partial wakeup control.

3.1. State Transition Diagram

Figure 3-1 shows the relationship between partial wakeup mode and other modes.

The figure outlines the operations. For a more detailed explanation of the operations, see the next section, "Flowchart."

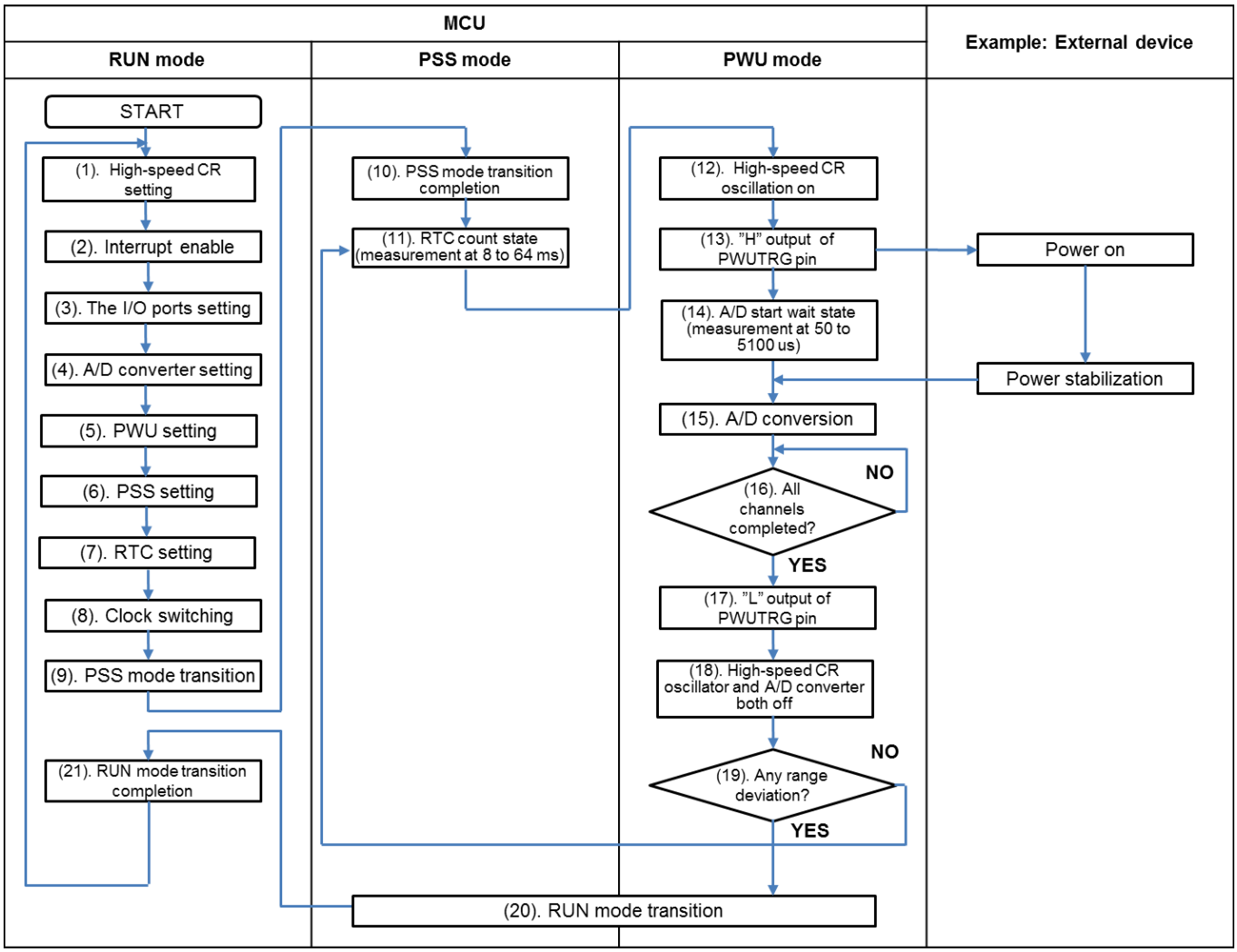
Figure 3-1 Partial Wakeup State Transition Diagram



3.2. Flowchart

This section uses the flowchart in Figure 3-2 to explain the partial wakeup function in detail.

Figure 3-2 Partial Wakeup Function Flowchart



(1) High-speed CR setting (processing by software)

(1)-1. Calibration

Software performs high-speed CR oscillation calibration and corrects any deviation in the A/D conversion time and any deviation in the period from PWUTRG pin output to A/D conversion start (external device stabilization wait time). Such deviations are due to process variations and variations depending on the use conditions.

For the setting method for the above, see the chapter of "CR Calibration" in Traveo™ Platform hardware manual.

(1)-2. Stabilization wait time

Set the PSCL bit and CMPR bit in SYSC_FCRCTCPR register to initial (default) value.

These settings are applied to “(13) High-speed CR oscillation on.”

If stabilization wait time is set to large value and PWU mode time exceeds PWU transition cycle,

This device becomes illegal state.

In this state, the transition to RUN mode can’t occur by A/D converter range deviation.

For the setting method for the above, see the chapter of "Source Clock Timer" in Traveo™ Platform hardware manual.

(2) Interrupt enable (processing by software)

You can treat enabled interrupt factors as return factors from PSS mode and PWU mode.
For any return due to an interrupt, there is a transition to the "(20)." processing.

The following can be specified: the interrupt of a resource not in the Power Down state, even in PSS mode.

Examples include external interrupt, RTC, NMI, low-voltage detection*1.

(*1 When used as an interrupt at the abnormal state detection)

For the method of enabling interrupts, see the chapters on the respective resources.
Note that this section does not cover A/DC-related settings because they are handled in the "(4)." processing.

(3) The I/O ports setting (processing by software)

Here, software makes the port output setting for the PWUTRG function and the analog I/O settings of the A/DC.

You can set the PWUTRG by writing appropriate value to the POF bit in the corresponding PPC_PCFGRijj (i=0 to 4, jj=00 to 31) register.

The target port is used as analog input of A/DC by writing appropriate value to the corresponding PPC_PCFGRijj (i=0 to 4, jj=00 to 31) register and GPIO_DDR register.

For a corresponding register and setting, see the chapter of "Port Configuration" on this manual.

Note that this section does not cover settings related to "retaining the pin state in PSS mode" because they are handled in the "(6)." processing.

(4) A/D converter setting (processing by software)

Table 3-1 lists the setting values for writable registers related to A/D converter unit 1, corresponding to PWU mode.

The following values are degrees of limitations.

3: Always set values according to the "setting value."

2: The recommended setting is based on the "setting value." It can be changed to match the use conditions.

1: Set a value appropriate to the use conditions.

Interpret the values as described.

Notes:

'i' represents the logical channel number. i = 0, 1, 2, …, 63
In PWU mode, the number of activation channel which can be used is maximum 8 in i = 0, 1, 2, …, 31.
In PWU mode, be sure to select the consecutive logical channel to execute the group processing.

For example
If 8 channels are used and the start channel is 0 (i=0), select the following logical channel number.
i = 0, 1, 2, 3, 4, 5, 6 and 7
If the start channel is 29 (i=29), the available logical channel number is the following.
i = 29, 30 and 31.

For a detailed explanation of the registers and the setting method, see the chapter of "12/10/8-Biy Analog To Digital Converter" on this manual.

Table 3-1 A/D Converter Settings

Register Name	Setting Value	Degree of Limitation
ADC12B1_CHCTRLi (i = 0~63)	TRGCL=1'b0	3
	SWTRG=1'b0	3
	Value of RCEN is set to the following value. RCEN bits corresponding to activation channels are set to 1'b1 (Range comparator enabled), and other bits are set to 1'b0 (Range comparator disabled).	3
	RCINVSEL=Set a value appropriate to use conditions.	1
	RCSEL[2:0]=Set a value appropriate to use conditions. Set the register value "i" (i.e. ADC12B1_FRCOHi, ADC12B1_FRCOLi) selected as the upper/lower threshold values.	1
	SMTIME[1:0]=Set a value appropriate to use conditions. Set the register value "i" (i.e. ADC12B1_STi) selected as channel sampling time.	2
	DP=1'b0	3
	RSMRST[1:0]=2'b00	1

Register Name	Setting Value	Degree of Limitation
ADC12B1_CHCTRLi (i = 0~63)	Value of CHPRI[3:0] are set to three kinds of priority. CHPRI[3:0]=4'b0000 (Highest priority) Set the first channel (e.g. corresponding to AN6) of the group to the above value. CHPRI[3:0]=4'b0001 Set the following channels (e.g. corresponding to AN7 - AN8, AN12-AN16) to the above value. CHPRI[3:0]=4'b0010 or more Set the unnecessary channels for PWU mode to the above value. For details and notes on the settings, see the chapter of “12/10/8-Bit Analog To Digital Converter" on this manual.	3
	Value of TRGTYP[1:0] are set to two kinds of the trigger type. TRGTYP[1:0]=2'b01 (Software or hardware trigger.) Set the first channel (e.g. corresponding to AN6) of the group to the above trigger type. TRGTYP[1:0]=2'b10 (Preceding logical channel conversion completion.) Set the following channels (e.g. corresponding to AN7 - AN8, AN12-AN16) to the above trigger type. No other channel (corresponding no used in PWU mode) must be included in the group which is defined for PWU mode. It is prohibited to set TRGTYP[1:0]=2’b11 (Idle trigger) in PWU mode. For details and notes on the settings, see the chapter of “12/10/8-Bit Analog To Digital Converter" on this manual.	3
	Value of ANIN[5:0]=6'b000110 to 6'b001000 or 6’b001100 to 6’b010000 Activation channels and analog input pins are associated. Only analog input pins AN6 to AN8 or AN12 to AN16 are enabled with PWU mode.	3
ADC12B1_PCCTRL i (i = 0~63)	Do not use with the PWU function.	1
ADC12B1_CDONEIRQE0/1	Value of {CDONEIRQE63 to CDONEIRQE0} are set to the following value. Interrupt enable bit (CDONEIRQEx) of the last channel in the group is set to 1'b1 (Conversion done interrupt enabled), and other bits are set to 1'b0 (Conversion done interrupt are disabled). (Do not transition to PWU mode with interrupts left enabled.) The channel set at this register will be the last channel of the group procedure.	3
ADC12B1_CDONEIRQC0/1	Write "0xFFFF_FFFF" to clear all A/D conversion done interrupt flags.	3
ADC12B1_GRPIRQE0/1	Value of {GRPIRQE63 to GRPIRQE0}="0x0000_0000_0000_0000" (Group interrupted interrupt disabled.)	3
ADC12B1_GRPIRQC0/1	These bits are not used in PWU mode.	1

Register Name	Setting Value	Degree of Limitation
ADC12B1_RCIRQE0/1	Value of {RCIRQE63 to RCIRQE0} are set to the following value. Interrupt enable bit (RCIRQEx) of the activation channel for PWU mode is set to 1'b1 (Range comparator interrupt enabled), and other bits are set to 1'b0 (Range comparator interrupt are disabled). (Do not transition to PWU mode with interrupts left enabled.)	3
ADC12B1_RCIRQC0/1	Write "0xFFFF_FFFF" to clear all Range comparator interrupt flags.	3
ADC12B1_PCIRQE0/1	Value of {PCIRQE63 to PCIRQE0}="0x0000_0000_0000_0000" (Pulse counter interrupt is disabled.)	3
ADC12B1_PCIRQC0/1	These bits are not used in PWU mode.	1
ADC12B1_TRGCL0/1	Write "0xFFFF_FFFF" to clear all A/D channel trigger status flags	3
ADC12B1_TRGORC0/1	Write "0xFFFF_FFFF" to clear all A/D channel trigger overrun flags	3
ADC12B1_CDDS0~3	CDCHEN=1'b0 (DMA request is disabled.) CDCHNUM[5:0] have no effect with CHCHEN=1'b0.	3
ADC12B1_RT	These bits are not used in PWU mode, when Power down disable mode (ADC12Bn_CTRL.PDDMD = "1"), these bits have no effect.	1
ADC12B1_CT	CT[15:0]="0x0001" ("0x0006" or more is a prohibited setting.) The built-in high-speed CR oscillation clock (4 MHz) is divided according to the setting, and the comparison time is measured using the divided clock. For details and notes on the settings, see the chapter of “12/10/8-Bit Analog To Digital Converter" on this manual.	2
ADC12B1_STi (i = 0~3) * Select only the registers corresponding to the numbers assigned as the settings for the sampling time	STi[15:0]="0x0007" ("0x0019" or more is a prohibited setting.) The built-in high-speed CR oscillation clock (4 MHz) is divided according to the setting, and the sampling time is measured using the divided clock. For details and notes on the settings, see the chapter of “12/10/8-Bit Analog To Digital Converter" on this manual.	2
ADC12B1_OCV	This register specifies the setting for offset compensation value.	2
ADC12B1_GCV	This register specifies the setting for gain compensation value. For details, see “4.Registers” in this chapter.	2
ADC12B1_CTRL	RES[1:0]=2'b00 (12 bit resolution)	3
	DBGE=1'b0	3
	ACHMD=1'b0	3
	FSMD=1'b0	3
	FRCMD=1'b1 (12 bit range comparator mode)	3
	FSTP=1'b0 (When the "not" forced stop mode (FSMD = "0")	1
	PDDMD=1'b1 (A/D convertor does not go idle state after A/D conversion finished.)	3
ADC12B1_RCOH0~7	Do not use with the PWU function.	1

Register Name	Setting Value	Degree of Limitation
ADC12B1_RCOL0~7	Do not use with the PWU function.	1
ADC12B1_FRCOH0~7 * Select only the registers corresponding to the numbers assigned as the upper threshold values	FRCOH[11:0]=Set a value appropriate the use conditions.	1
ADC12B1_FRCOL0~7 * Select only the registers corresponding to the numbers assigned as the lower threshold values	FRCOL[11:0]=Set a value appropriate the use conditions.	1
ADC12B1_MCCTRL0~3	Value of ADC12B1_MCTRL="0x00"	3

(5) PWU setting (processing by software)

Here, software makes the following settings.

(5)-1. Enable the PWU function.

(5)-2. Set the external device stabilization wait time, which is the period between PWUTRG pin output and A/D conversion start.

You can select stabilization wait times from 50 to 5100 us, in steps of 50 us.

(5)-3. Select the PWU A/D conversion trigger (PWU_ADT) as A/D conversion trigger.

(5)-4. Set the first logical channel number of the group procedure executing in the PWU mode.

For the method of setting the above, see the chapter of “Partial Wakeup Control 4. Registers" on this manual.

(6) PSS setting (processing by software)

Table 3-2 lists the setting values for the writable registers in the PSS profile register group and the writable registers of the system special setting registers. The following values are degrees of limitations.

3: Always set values according to the "setting value."

2: The recommended setting is based on the "setting value." It can be changed to match the use conditions.

1: Set a value appropriate to the use conditions.

Interpret the values as described.

For a detailed explanation of the registers and the setting method, see the chapter Low-Power Consumption" in Traveo™ Platform hardware manual.

Table 3-2 PSS Profile Register Group Settings

Register Name	Setting Value	Degree of Limitation
SYSC0_PSSPDCFGR	PD4_1EN = Set a value appropriate to Backup area in System SRAM use conditions.	1
	PD4_0EN = Set a value appropriate to Backup area in System SRAM use conditions.	1
	PD2EN=1'b0	3

Register Name	Setting Value	Degree of Limitation
SYSC0_PSSCKSRER	SSCG0EN=1'b0	3
	PLL0EN=1'b0	3
	MOSCEN=1'b0	3
	SCROSCEN=1'b1	3
	CROSCEN=1'b0	3
	SOSCEN=1’b0	3
SYSC0_PSSCKSELR	CDMCUCCSL=3'b111	3
SYSC0_PSSCKER	ENCLKMCUCP=1'b0	3
	ENCLKMCUCH=1'b0	3
SYSC0_PSSCKDIVR	MCUCHDIV=5'b00000	3
SYSC0_PSSPLLxCNTR	PLLxISEL=1'b0	2
	PLLxDIVN=8'b00001101	2
	PLLxDIVM=4'b0001	2
	PLLxDIVL=2'b00	2
SYSC0_PSSSSCGxCNTR0	SSCGxISEL=1'b0	2
	SSCGxDIVN=8'b00001101	2
	SSCGxDIVM=4'b0001	2
	SSCGxDIVL=2'b00	2
SYSC0_PSSSSCGxCNTR1	SSCGxSSEN=1'b0	2
	SSCGxFREQ=2'b00	2
	SSCGxMODE=1'b0	2
	SSCGxRATE=10'b0000101001	2
SYSC0_PSSLVDCFGR	LVDL1S = Set a value appropriate to the use conditions.	1
	LVDL1V = Set a value appropriate to the use conditions.	1
	LVDL1E = Set a value appropriate to the use conditions.	1
	LVDH1S = Set a value appropriate to the use conditions.	1
	LVDH1V = Set a value appropriate to the use conditions.	1
	LVDH1E = Set a value appropriate to the use conditions.	1
SYSC0_PSSCSVCFGR	SSCG0CSVE=1'b0	3
	PLL0CSVE=1'b0	3
	SCRCSVE=1’b0	3
	CRCSVE=1’b0	3
	SOCSVE=1’b0	3
	MOCSVE=1'b0	3
SYSC0_PSSREGCFGR	RMSEL=1'b1	3

Register Name	Setting Value	Degree of Limitation
SYSC0_SPECFGR	HOLDIO_PD2=1'b1 The setting by this bit is reflected immediately after writing register, it is not after PSS profile change.	3
	PSSPADCTRL=1'b0	3
	EXVRSTCNT=1'b1 If the setting is EXVRSTRCNT=1’b0, the RAM data of Backup area in System SRAM are not guaranteed after LVDH1 reset.	3
SYSC1_PSSCKSELR0	LAPP1ACSL=1’b0	3
	LAPP0ACSL=1’b0	3
	LCP1ACSL=1’b0	3
	LCP0ACSL=1’b0	3
	CD0CSL=3’b111 If this setting is not made, a PSS profile error occurs at a PSS mode transition.	3

(7) RTC setting (processing by software)

The following shows examples of RTC settings. For details on the setting method, see the chapter "Real-Time Clock" in Traveo™ Platform hardware manual.

Example of settings

(7)-1. Initialization settings

Sequence	Set Contents *1	Set Timing
1-1.	Write "1" to the ST bit in the RTC_WTCR register.	1st setting only after POR
1-2.	Write "0" to the ST bit in the RTC_WTCR register.	1st setting only after POR
1-3.	Write "2'b10" to the RCKSEL bit in the RTC_WTCR register, write "1" to the CSM bit, and select the low-speed CR clock.	1st setting only after POR (RTC Clock = 100KHz)
1-4.	Set each bit in the RTC_WRT register.	1st setting only after POR

*1 Don’t write to the registers by bit level.
In bit-level-write, the no-change-bit have the function which the same value as read data is written to the register. In this case, the bit having difference function between read and write is changed to the other setting.
Therefore the bits no change is intended will be changed.

(7)-2. Calibration setting (for correcting any deviation in the count values due to variations in low-speed oscillation)

Sequence	Set Contents *1	Set Timing
2-1.	Write a value to the DURMW bit in the RTC_DURMW register, and set the calibration period. Normally, set the calibration period to 0.25[s]. To set a time 0.25[s] or shorter, a value of 0.25[s] divided by 1, 2, 4, 8, or 16 can be selected for the period. However, the lower the value set, the lower the accuracy becomes. (Related to the SCAL[2:0] bit) When the main clock is 4 MHz and the calibration period is set to 0.25[s], set DURMW bit value="0xF4240" (1'000'000 as a decimal).	1st setting only after POR

Sequence	Set Contents *1	Set Timing
2-2.	Set the related bits in the RTC_WTCR register as described below.	
	− Set the SCAL [2:0] bit according to the RTC_DURMW value. If the calibration period is set to 0.25[s], set "'3'b000".	1st setting only after POR
	− Set the CCKSEL bit to 1'b1, and select the low-speed CR.	1st setting only after POR (RTC Clock = 100KHz)
	− Set the ENUP bit according to the use conditions. To update the calibration result automatically as hardware, set "1'b1". To update it with software, set "1'b0".	1st setting only after POR
	− Set the ACAL bit to "1'b0", and disable the auto-calibration.	1st setting only after POR
2-3.	Write "1'b1" to the CALDC bit in the RTC_WINC register, and clear the calibration interrupt.	Clear the interrupt flag after interrupt occurs
2-4.	Write "1'b1" to the MTRG bit in the RTC_WTCR register, and start calibration.	Set in the timing you want to calibrate.(e.g. Post-PWU) The calibration is available on real time count.
2-5.	Wait until the CALD bit in the RTC_WINS register becomes "1'b1".	

*1 Refer to the previous table.

(7)-3. Partial wakeup setting - Case 1: When the ENUP bit in (7)-2-2 is set to "1'b1". -

Sequence	Set Contents *1	Set Timing
3-1.	Set the related bits in the RTC_PWUTRGCR register as described below.	
	Set the MD bit to "1'b1".	1st setting only after POR
	For the SEL bit, set the cycles for transitioning to PWU mode in accordance with the purpose. Values ranging from 8 to 64 ms can be selected in units of 8 ms.	The period of PWU can be switched.
3-2.	Wait until the BUSY bit in the RTC_PWUTRGSR register becomes 1'b0.	Set if RTC_PWUTRGCR register is changed.

*1 Refer to the previous table.

(7)-3. Partial wakeup setting - Case 2: When the ENUP bit in (7)-2-2 is set to "1'b0". -

Sequence	Set Contents *1	Set Timing
3-1.	Read the value of RTC_CNTCAL, execute [-1 processing], and then execute [divide-by-32 processing]. The RTC_CNTCAL includes the counter value of 0.25[s]. By dividing by 32, it generates a count value of approximately 8 ms.	Set if the calibration (the setting of “2-3”-”2-5”) is completed.

Sequence	Set Contents *1	Set Timing
3-2.	Read the value of RTC_CNTCAL, and execute [-1 processing]. Write the post processing value to the WTBR bit in the RTC_WTBR register.	
3-3.	Set the related bits in the RTC_PWUTRGCR register as described below.	
	− Set the C8MRL bit to the value calculated in "(7)-3-1."	Set if the calibration (the setting of “2-3”-”2-5”) is completed.
	− Set the MD bit to "1'b0".	1st setting only after POR
	− For the SEL bit, set the cycles for transitioning to PWU mode in accordance with the purpose. Values ranging from 8 to 64 ms can be selected in units of 8 ms.	The period of PWU can be switched.
3-4.	Wait until the BUSY bit in the RTC_PWUTRGSR register becomes 1'b0.	Set if RTC_PWUTRGCR register is changed.

*1 Refer to the previous table.

(7)-4. RTC count start

Sequence	Set Contents *1	Set Timing
4-1.	Set the RTC_WINE register, and enable the required interrupts.	1st setting only after POR
4-2.	Write "1" to each bit in the RTC_WINC register, and clear the interrupt flag.	Clear the interrupt flags after interrupts occur
4-3.	Write "1" to the ST bit in the RTC_WTCR register, and start counting.	1st setting only after POR (RTC Clock = 100KHz)

*1 Refer to the previous table.

(8) Clock switching (processing by software)

Here, software makes the following settings.
(8)-1. Clock gear down operation

If the PLL clock is being used for operating the internal circuit, use the clock gear down function to set the clock frequency lower in stages to reduce the current fluctuations due to clock switching.

For details on how to use the clock gear down function, see the "Clock Gear" section in the chapter of "Clock System" in Traveo™ Platform hardware manual.

(8)-2. Clock adjustment

To transition to the PSS, establish a 1:1:1:1 relationship between the CPU clock, the memory configuration clock, the SCU clock, and the MCUCH clock.

The absence of this relationship between these clocks may cause a malfunction.

Additionally, PLL and SSCG-PLL must be disabled before PSS mode transition.

Example of settings

Because the SCU clock operates with the high-speed CR oscillation clock,

switch the CPU clock (CLK_CPU), memory configuration clock (CLK_MEMC), and MCUCH clock (CLK_SYSC0H) so that they all operate with the high-speed CR oscillation clock.

(8)-2-1. Set the CD0CSL bit in the SYSC1_RUNCKSELR0 register to "All-0".

(8)-2-2. Set the HPMDIV bit and the SYSDIV bit in the SYSC1_RUNCKDIVR0 register to "All-0".

(8)-2-3. Set the CDMCUCCSL bit in the SYSC0_RUNCKSELR register to "All-0".

(8)-2-4. Set the MCUCHDIV bit in the SYSC0_RUNCKDIVR register to "All-0".

(8)-2-5. Set the PLL0EN and SSCG0EN bit in the SYSC0_RUNCKSRER register to "All-0".
(PLL and SSCG-PLL are disabled.)

(8)-2-6. Set the SYSC0_RUNPLL0CNTR, SYSC0_RUNSSCG0CNTR0 and SYSC0_RUNSSCG0CNTR1 register to appropriate value to avoid profile error.

For details on the combination of violation settings,
see "Profile" in the chapter of "Low-Power Consumption" in Traveo™ Platform hardware manual.

(8)-2-7. Set the SYSC1_RUNENR register to "0xAB".

(8)-2-8. Set the SYSC0_TRGRUNCNTR register to "0xAB". (The RUN profile update starts.)

(8)-2-9. Wait until the RUNDF0 bit in the SYSC0_SYSSTSR register becomes "1".

(9) PSS mode transition (processing by software)

See "Operation Procedure" in the chapter of "Low-Power Consumption" in Traveo™ Platform hardware manual and transition to PSS mode.

(10) PSS mode transition completion (processing by hardware)

There has been a transition to PSS mode according to the PSS profile register settings at "(6)."

(11) RTC count state (processing by hardware)

Hardware waits until the RTC count value reaches the cycles set at "(7)."

After the cycles are reached, there is a transition to PWU mode.

For a transition from RUN mode to PSS mode, the RTC is not initialized. For this reason, after the transition from RUN mode to PSS mode, the time until the transition to PWU mode is less than the setting at "(7)."

(The time depends on the RTC count value at the transition to PSS mode.)

(12) High-speed CR oscillation on (processing by hardware)

Hardware turns on high-speed CR oscillation.

Stabilization wait time is inserted to turn on high-speed CR oscillation.

This wait time is set at “(1).”

PWU mode operates with the output clock of the high-speed CR oscillation.

(13) "H" output of the PWUTRG pin (processing by hardware)

Hardware changes PWUTRG to "H".

PWUTRG is "L" before this processing.

(14) A/D start wait state (processing by hardware)

Hardware waits until it arrives at the time set at "(5)."

The power supply of external devices is assumed to be stable during this wait period.

When the time comes, A/D conversion request (PWU_ADT) is issued and A/D conversion is started.

(15) A/D conversion (processing by hardware)

A/D conversion is executed according to the settings at "(4)."

(16) A/D conversion judgment (processing by hardware)

Hardware waits until the conversion of all selected channels (maximum of 8 channels) completes.

(17) "L" output of the PWUTRG pin (processing by hardware)

Hardware switches PWUTRG output from "H" to "L".

(18) High-speed CR oscillator and A/D converter both off (processing by hardware)

Hardware turns off the high-speed CR oscillator and A/D converter.

(19) Range comparison judgment (processing by hardware)

Hardware judges whether a range deviation has arisen as a result of the A/D conversion at "(15)."

If no deviation has arisen, there is a transition to PSS mode again.

If a deviation has arisen, a WAKEUP signal is issued, followed by a transition to RUN mode to wake up the CPU.

(20) RUN mode transition (processing by hardware)

For details on the operation, see "Operation Procedure" in the chapter of "Low-Power Consumption" in Traveo™ Platform hardware manual.

(21) RUN mode transition completion (processing by hardware)

The transition to RUN mode has completed.
There are the following notes to design software.

(21)-1. Regarding PSS enable setting

Set PSSEN0 bit in SYSC0_PSSENR register to "0x00".
Otherwise, written data to registers in SYSC0 is invalid. Bus error will occur.

But the set to PSSEN1 bit in SYSC1_PSSENR register is not required
because of initialization with power down.

(21)-2. Regarding the judgment of wakeup factor

If the following “(21)-2-1” and “(21)-2-2” both are "0" before "(9) PSS mode transition (processing by software)" and "1" after RUN mode transition, you are able to judge RUN mode transition occurred by A/DC range deviation.

(21)-2-1. RCIRQ0 to RCIRQ63 in ADC12B1_RCIRQ0/1 register

(21)-2-2. CDONEIRQ0 to CDONEIRQ63 in ADC12B1_CDONEIRQ0/1 register

Moreover, in case that you set in "(2). Interrupt enable" process,
RUN mode transition might occur by the factor you set.

You need to confirm these interrupt flag.

(21)-3. Regarding A/D Converter use after RUN mode transition

The corresponding A/D Channel Trigger Status flag (ADC12B1_TRGST0/1.TRGST and ADC12B1_CHSTAT0～63.TRGST) is set if A/D conversion request (PWU_ADT) occurs in PWU mode.

It is not allowed to update the A/D converter setting during A/D conversion operation (ADC12B1_TRGST0/1.TRGST and ADC12B1_CHSTAT0～63.TRGST=”1”)

After run mode transition, clear all trigger status flags in case of changing A/D converter setting.

Example of settings
Write "1" to TRGCL0 to TRGCL63 bits in ADC12B1_TRGCL0/1 register.

For a detailed explanation of the A/D Channel Trigger Status flags, see the chapter of "12/10/8-Bit Analog To Digital Converter” on this manual.

When using hardware trigger as active trigger of A/D Converter Unit1 in RUN mode, write “0” to ADHWTS bit in PWU_ADTC register to make the other activation factor available.

For a detailed description of the A/D conversion hardware trigger select, refer to Section4.

3.3. Timing Charts

Figure 3-3, Figure 3-4, and Figure 3-5 are timing charts related to the transition to partial wakeup mode.

Figure 3-3 With a Range Deviation Arising after a Transition from PSS Mode to PWU Mode

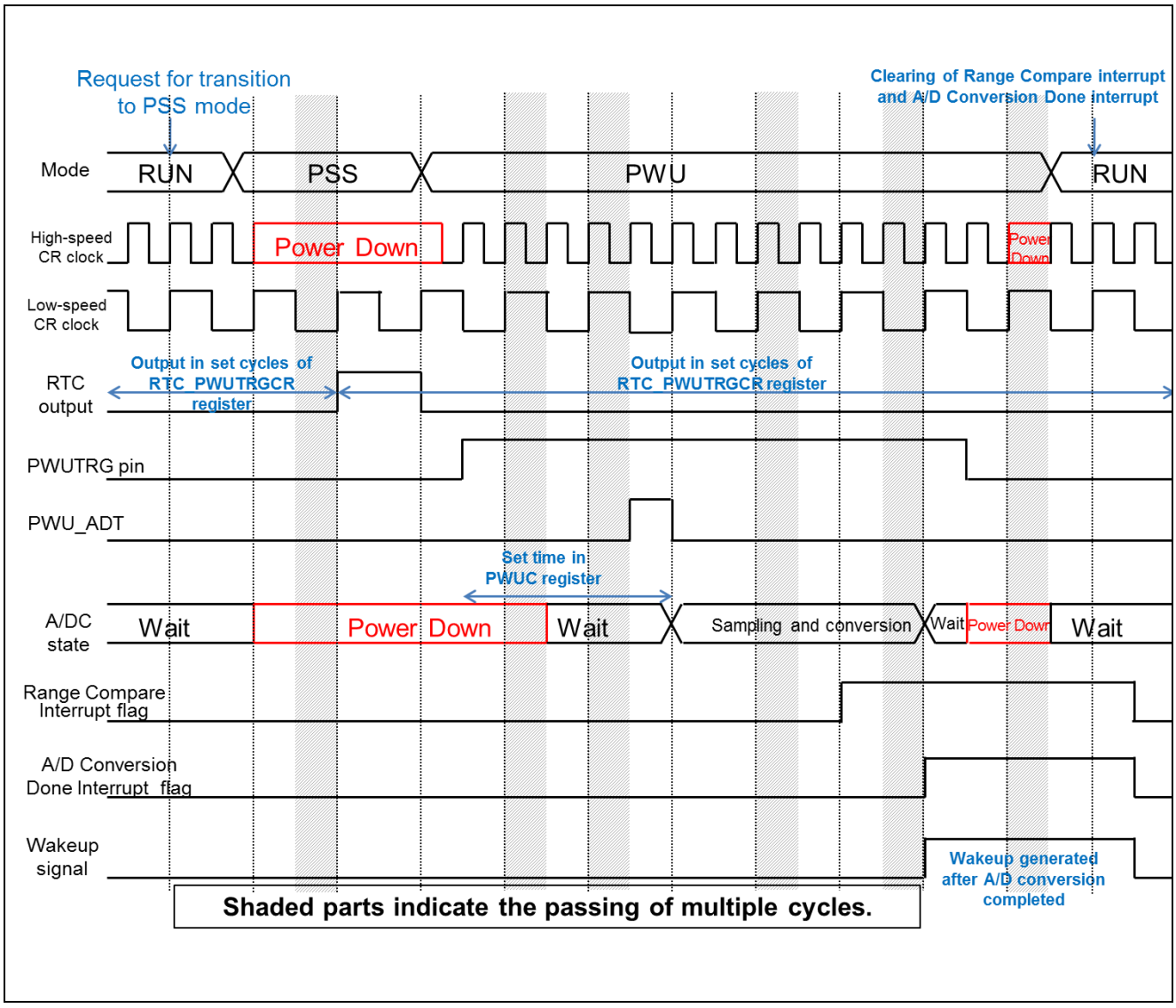


Figure 3-4 Without a Range Deviation Arising after a Transition from PSS Mode to PWU Mode

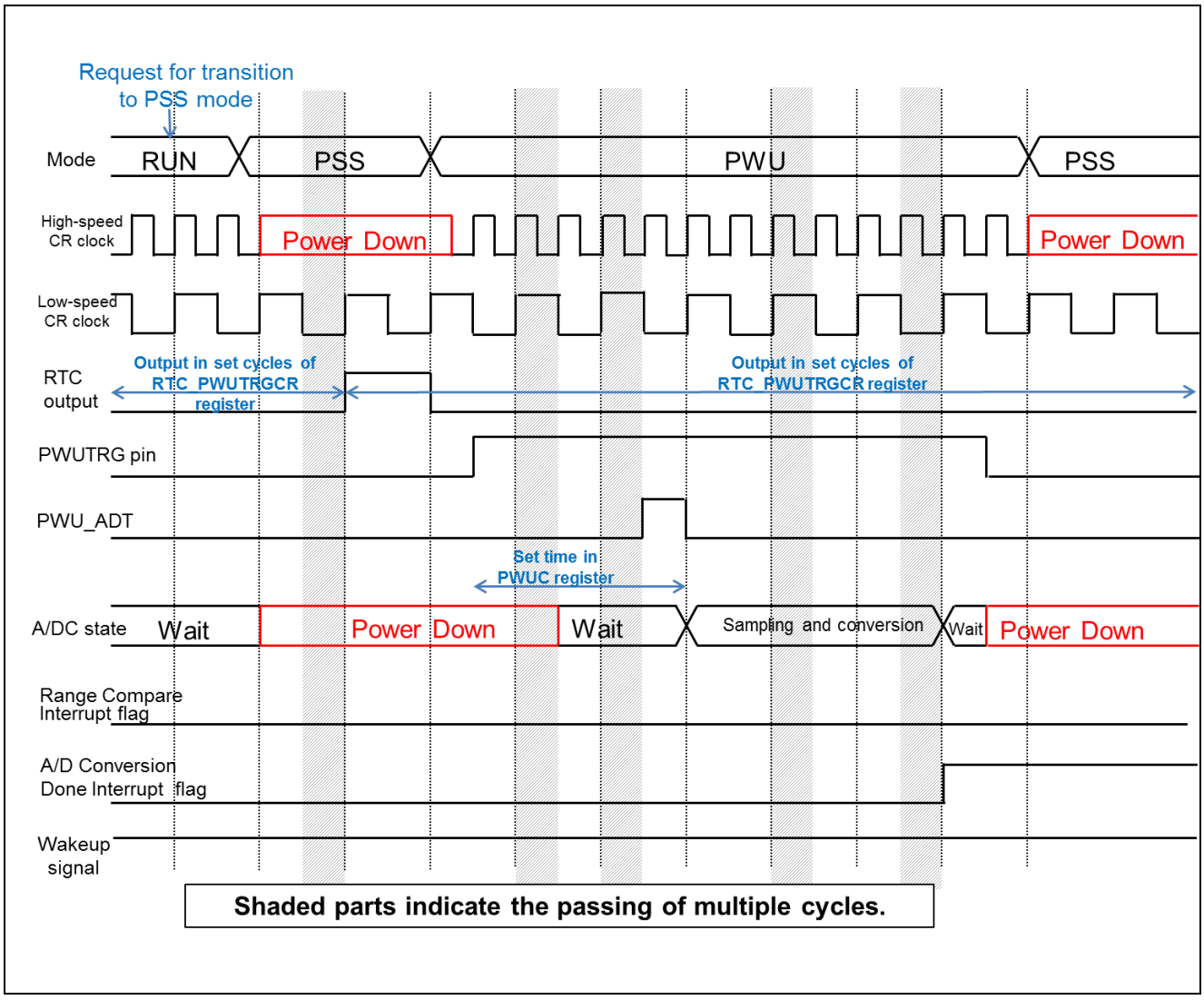
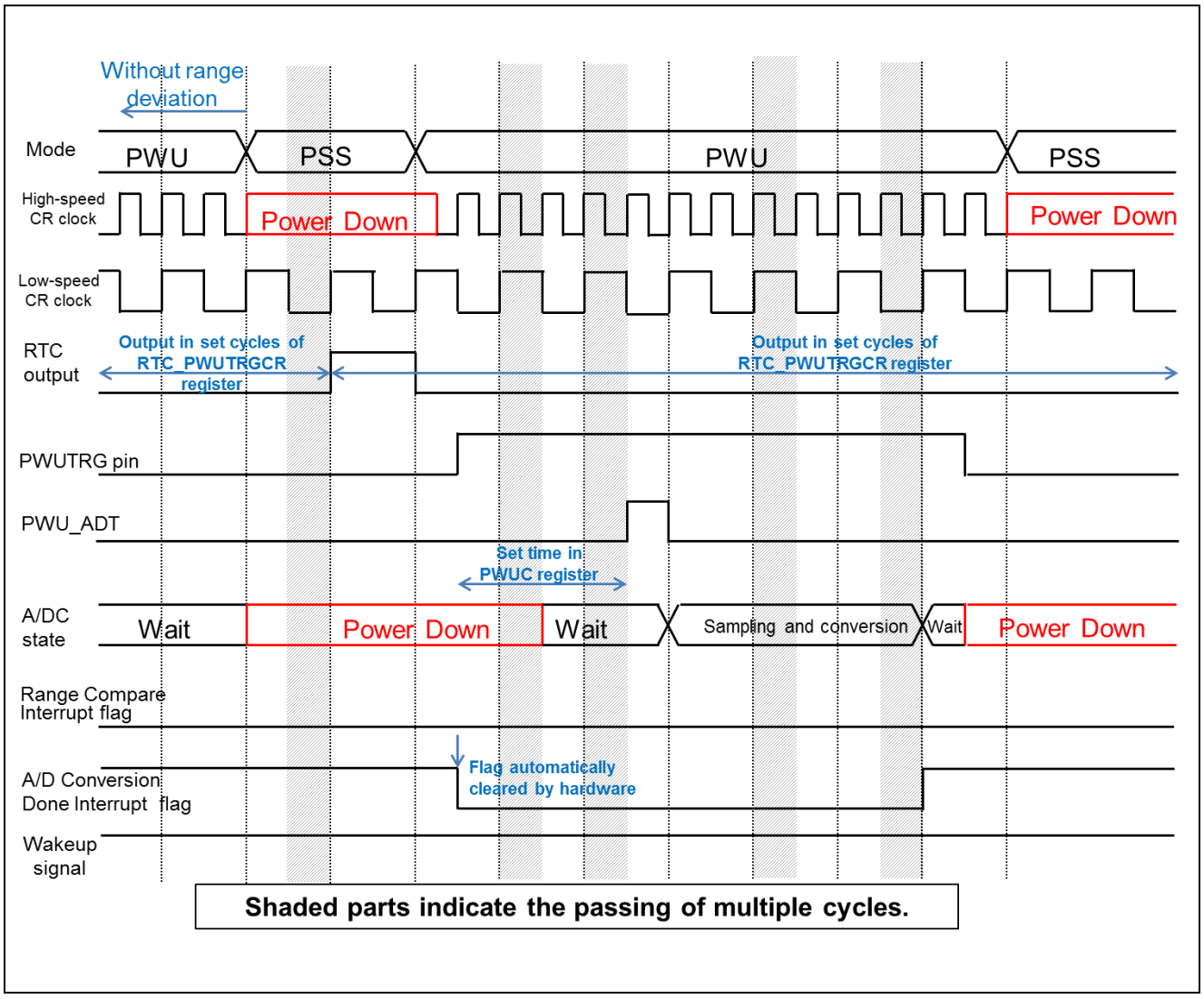


Figure 3-5 Until a Restart of A/D Conversion after Figure 3-4.



Note:

- *The clocks in the figures do not match the actual number of cycles.*

4. Registers

This section explains the partial wakeup control register.

Table 4-1 List of the Partial Wakeup Control Register

Abbreviated Register Name	Register Name	Reference
PWU_PWUC	PWU control register	4.1
PWU_ADTC	A/D Conversion Request Trigger Control Register	4.2

Table 4-2 Memory Map and Initial Values of the Partial Wakeup Control Register

Offset	+3	+2	+1	+0
0x00000000	Reserved	Reserved	PWU_ADTC 00000000	PWU_PWUC 00000000

The offset value is the value for the "Partial Wake Up" area start address.
The partial wakeup register is deployed after the register of A/D converter unit 1.

4.1. PWU Control Register (PWU_PWUC)

The PWU control register (PWUC) is used to enable PWU mode and to set the period between PWUTRG pin output and A/D conversion start.

Bit	7	6	5	4	3	2	1	0
Field	PWUE	ADSTS6	ADSTS5	ADSTS4	ADSTS3	ADSTS2	ADSTS1	ADSTS0
R/W Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit7] PWUE: PWU mode enable bit

bit	Description
0	Disable transition to PWU mode.
1	Enable transition to PWU mode.

- This bit is used to enable transition to PWU mode.
- When this bit is "1", there is a transition from PSS mode to PWU mode.

[bit6:0] ADSTS [6:0]: A/D conversion start time setting bits

ADSTS [6:0]	Setting	Value for High-speed CR Oscillator, 4 MHz
0000000	200 cycles of high-speed CR oscillation cycle	50 us
0000001	400 cycles of high-speed CR oscillation cycle	100 us
0000010	600 cycles of high-speed CR oscillation cycle	150 us
…	…	…
N	(n+1)*200 cycles of high-speed CR oscillation cycle	(n+1)*50 us
…	…	…
1100011	20000 cycles of high-speed CR oscillation cycle	5000 us
1100100	20200 cycles of high-speed CR oscillation cycle	5050 us
1100101	20400 cycles of high-speed CR oscillation cycle	5100 us
110011x	20400 cycles of high-speed CR oscillation cycle	5100 us
1101xxx		
111xxxx		

These bits are used to set the period between PWUTRG pin output and A/D conversion start.

4.2. A/D Conversion Request Trigger Control Register (PWU_ADTC)

The A/D Conversion Request Trigger Control Register configures the logical channel which uses PWU_ADT as A/D Conversion activation trigger.

Bit	15	14	13	12	11	10	9	8
Field	ADHWTS	Reserved	Reserved	ADSTCH[4]	ADSTCH[3]	ADSTCH[2]	ADSTCH[1]	ADSTCH[0]
R/W Attribute	R/W	RX, WX	RX, WX	R/W	R/W	R/W	R/W	R/W
Protection Attribute	-							
Initial Value	0	0	0	0	0	0	0	0

[bit15] ADHWTS: A/D conversion Hardware Trigger Select bit

bit	Description
0	Hardware trigger of A/D conversion request is the other activation factor.
1	Hardware trigger of A/D conversion request is A/D conversion request for PWU (PWU_ADT).

- The other activation factor is configured at RESSEL bit in RIC_RESIN register.
- When this bit is set to “1”, PWU_ADT is selected as hardware trigger of A/D Converter unit 1.
- Before PWU mode transition, set “1” to this bit.
- When using hardware trigger as active trigger of A/D Converter unit1 in RUN mode, write “0” to this bit.

Notes:

- *These bits are not allowed to update during A/D conversion operation (ADC12Bn_TRGST0/1.TRGST and ADC12Bn_CHSTATi.TRGST=”1”).*
- *When this setting is changed, confirm A/D conversion request isn’t issued (ADC12Bn_TRGST0/1.TRGST and ADC12Bn_CHSTATi.TRGST=”0”).*

[bit14:13] Reserved: Reserved bits

- Reading this bit returns an undefined.
- Writing data to these bits has no effect on the operation.

[bit12:8] ADSTCH [4:0]: A/D conversion start channel setting bits

ADSTCH [4:0]	Description
00000	The logical channel 0 is selected.
00001	The logical channel 1 is selected.
…	…
N	The logical channel N is selected.
…	…
11110	The logical channel 30 is selected.
11111	The logical channel 31 is selected.

- Set the first logical channel number of the group procedure executing in the PWU mode.
- PWU_ADT is selected as hardware trigger of the logical channel set at this register.
- When TRGTYP of the logical channel selected at this register is set to Software or Hardware (TRGTYP[1:0] = “01”), A/D conversion request of the corresponding logical channel occurs after PWU mode transition.

5. Precautions for Using this Device

This section provides notes on using the partial wakeup control function.

Time until transition to PWU mode

For a transition from RUN mode to PSS mode, the RTC is not initialized.
For this reason, after the transition from RUN mode to PSS mode, the time until the transition to PWU mode is less than the setting at "(7)."
(The time depends on the RTC count value at the transition to PSS mode.)

Restrictions on A/D conversion data

If an interrupt factor other than a range comparison interrupt causes a return to RUN mode during A/D conversion, the A/D conversion results (values recorded in the A/D data register) are not guaranteed. Examples of this factor include external interrupts and RTC interrupts.

Restrictions on the range comparison function

The range comparison function sets the upper and lower limit thresholds for voltage and judges A/D conversion results.
8 analog input channels can be used in PWU mode.
However, for pairs of the upper limit thresholds and lower limit thresholds that can be set, select from 4 sets.

Restrictions on port settings

- Before transitioning to PSS mode, be sure to use the port function to set for output of the PWUTRG function.
- The state of ports at the generation of various resets is "output Hi-Z" and "input disabled."

Restrictions on software watchdog timer settings

If the settings of the software watchdog timer in PSS mode contradict the contents of the SYSC0_PSSCKSRER register that have been set in "PSS setting (processing by software)" in Section "3.Explanation of Operation", a PSS profile error occurs. It cannot transition to PSS mode.
Be careful when setting the software watchdog timer.

For details on the combination of violation settings, see "Profile" in the chapter of "Low-Power Consumption" in Traveo™ Platform hardware manual.

For details on the setting procedure examples of the software watchdog timer, see " Example of set procedure" in the chapter of "Software Watchdog Timer" in Traveo™ Platform hardware manual.

Restrictions on the cycles for transitioning to PWU mode

The term of 8ms is generated from the following calculation in RTC.

0.25[s] ÷ 32 = 7.8125[ms] (≈ 8[ms])

Moreover as RTC operates with low-speed CR (typical 100KHz) in PWU mode,

the actual cycles for transitioning to PWU mode become as the following.

7.81[ms] (The description in this chapter is “8ms”)

15.62[ms] (The description in this chapter is “16ms”)

23.43[ms] (The description in this chapter is “24ms”)

31.24[ms] (The description in this chapter is “32ms”)

39.05[ms] (The description in this chapter is “40ms”)

46.86[ms] (The description in this chapter is “48ms”)

54.67[ms] (The description in this chapter is “56ms”)

62.48[ms] (The description in this chapter is “64ms”)

Restrictions on the A/D conversion hardware trigger settings

■ Before PSS mode transition, be sure to select the PWU A/D conversion trigger (PWU_ADT) as A/D conversion trigger.

CHAPTER 24: Programmable CRC

This chapter explains the function and operation of the Programmable CRC.

1. Overview
2. Configuration and Block Diagram
3. Operation of the Programmable CRC
4. Registers

CODE: PRGCRC-S6J3200-E1

1. Overview

This section describes the features and the block diagram of the Programmable CRC.

Features of Programmable CRC

The Programmable CRC is a software configurable module with serial CRC calculation logic as hardware implementation. The serial CRC logic works on modulo-2 arithmetic for calculation of checksum. The CRC module can detect errors in data blocks by calculating a checksum.

- Programmable 8-, 16-, 24-, or 32-bit input data width
- Programmable polynomial value (polynomial degree from 2 to 32)
- Programmable initial seed value
- Programmable final checksum XOR value
- Interrupt and DMA trigger capability
- Configurable input/output bit reflection and byte swapping. This facilitates the different settings of common CRC standards and the handling of data organized in little or big endian format
- Supports block/multiple data transfers (more than 32-bit)

Areas of application

- Data security/integrity
- Communication protocols

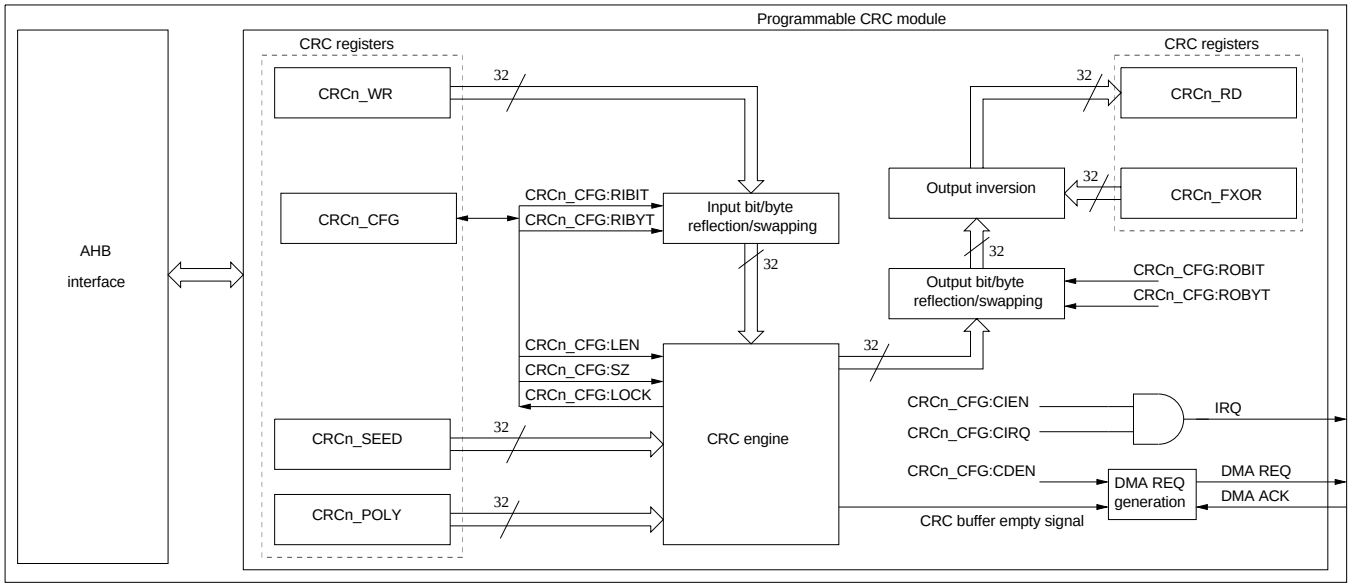
Programmable CRC module can be configured to widely used common CRC standards, some of them are listed below:

- CRC-32-IEEE 802.3
- CRC-16-CCITT
- CRC-8-CCITT
- CRC-5-USB
- CRC-XMODEM
- 12-bit CRC
- 10-bit CRC
- 8-bit CRC

2. Configuration and Block Diagram

This section shows a block diagram of Programmable CRC

Figure 2-1 Block Diagram of Programmable CRC



3. Operation of the Programmable CRC

This section describes operation of Programmable CRC in detail.

For more details on flowcharts for CRC operation see Section 3.1, on CRC calculation flow see Section 3.2, and on an example for CRC calculation see Section 3.3

3.1. CRC Operation Flowcharts

The flowcharts Figure 3-1, Figure 3-2, and Figure 3-3 show the steps to configure CRC registers and to perform a CRC calculation.

Figure 3-1 Polling based CRC Operation

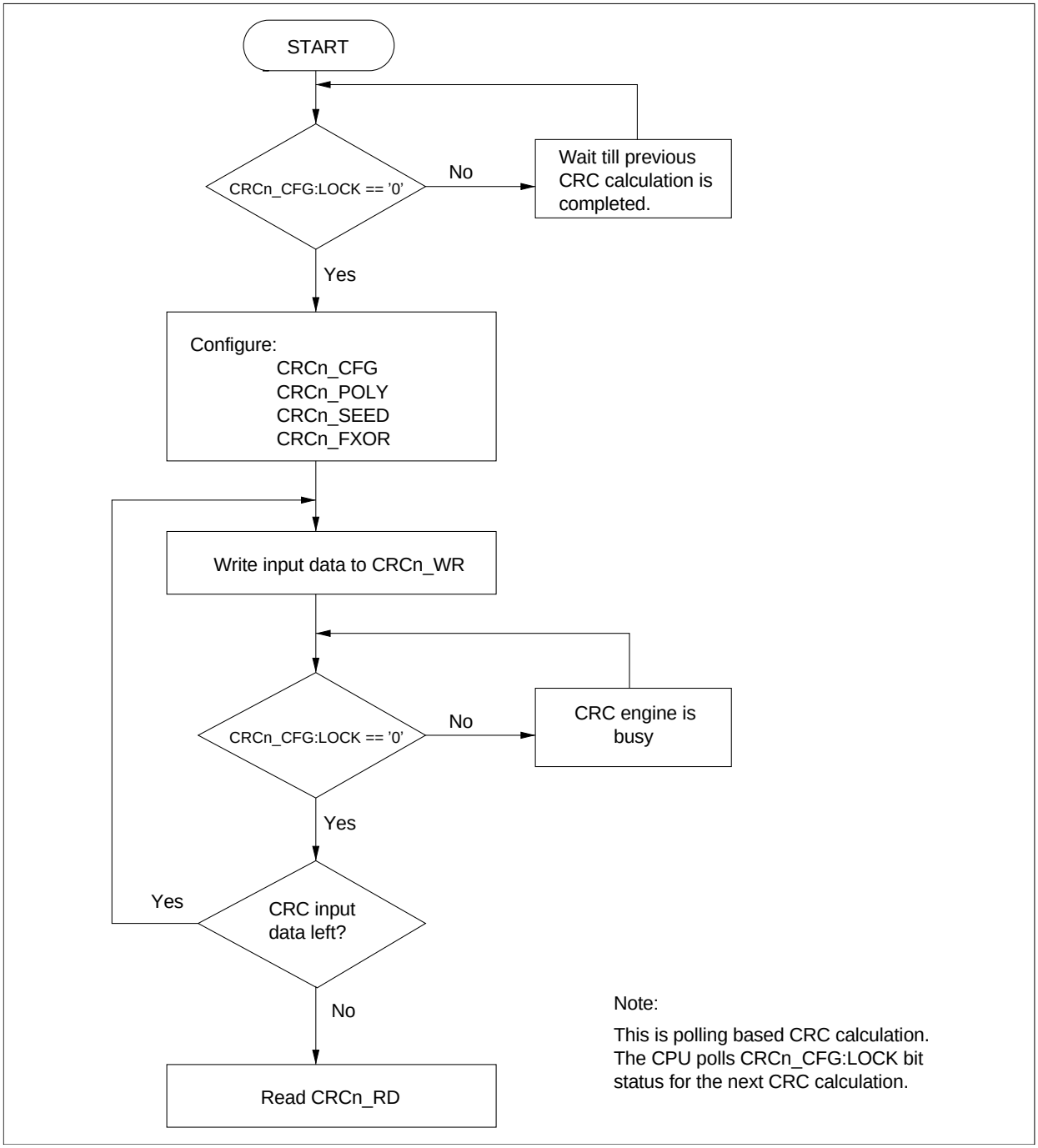


Figure 3-2 CRC Operation with IRQ

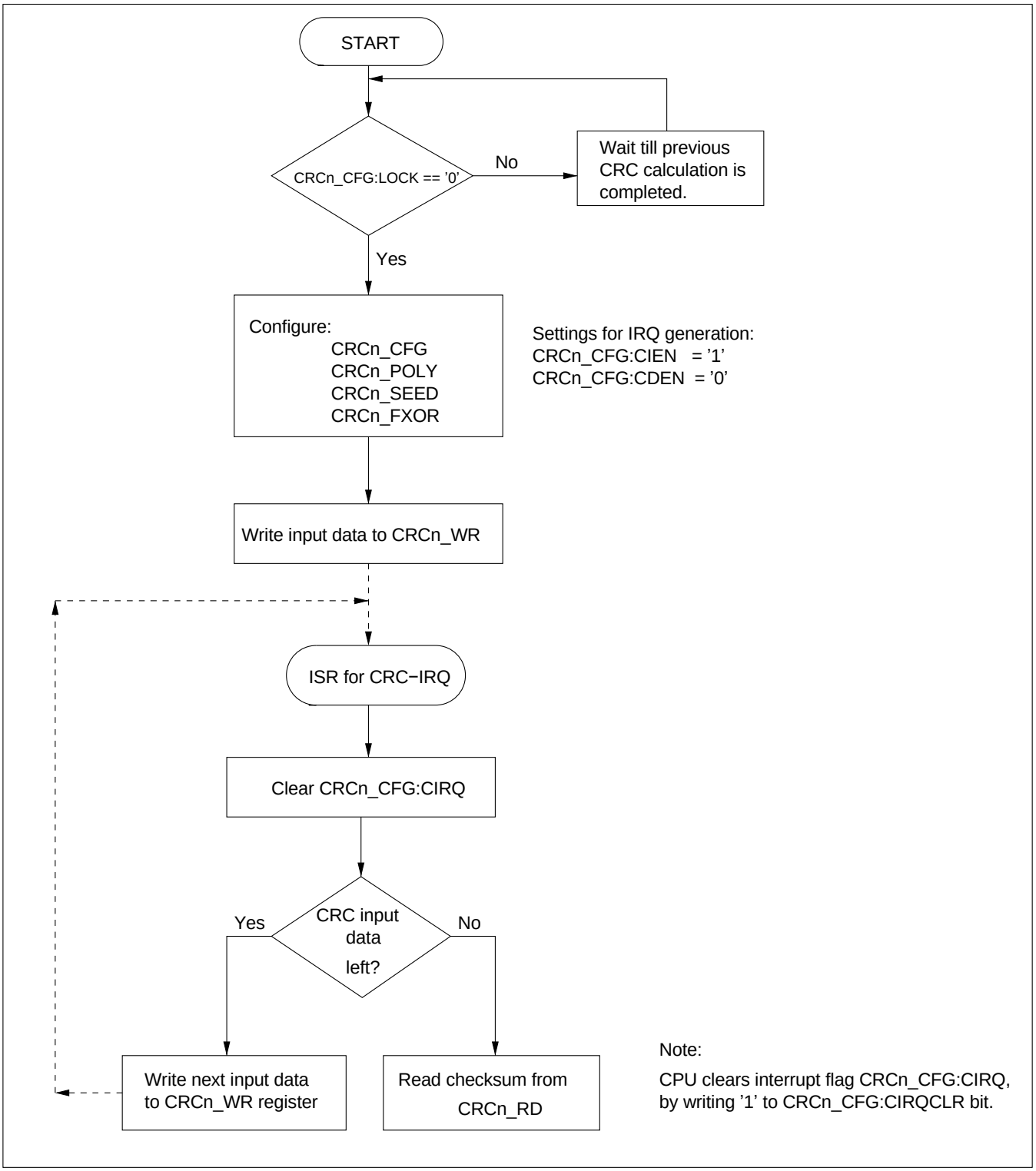
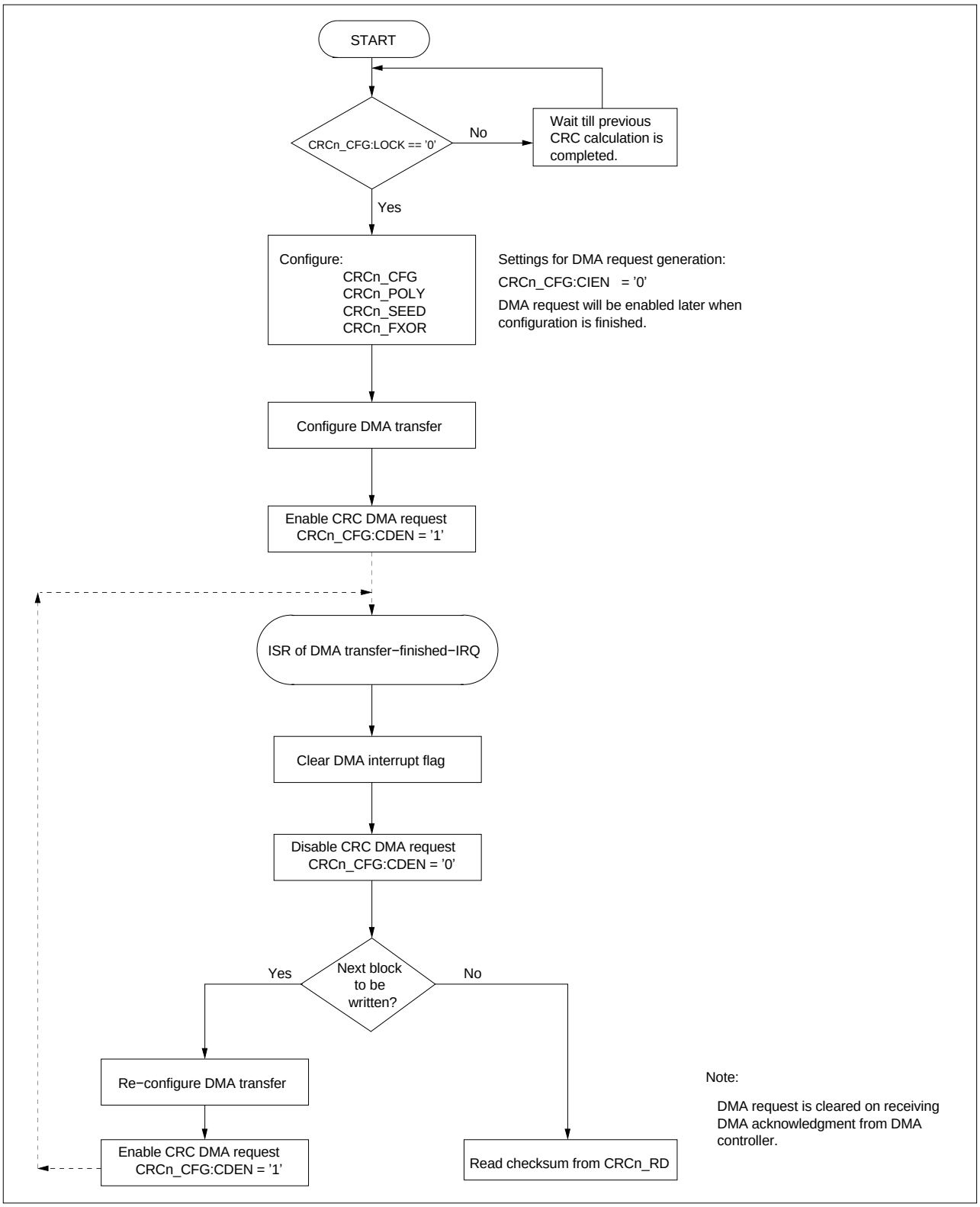
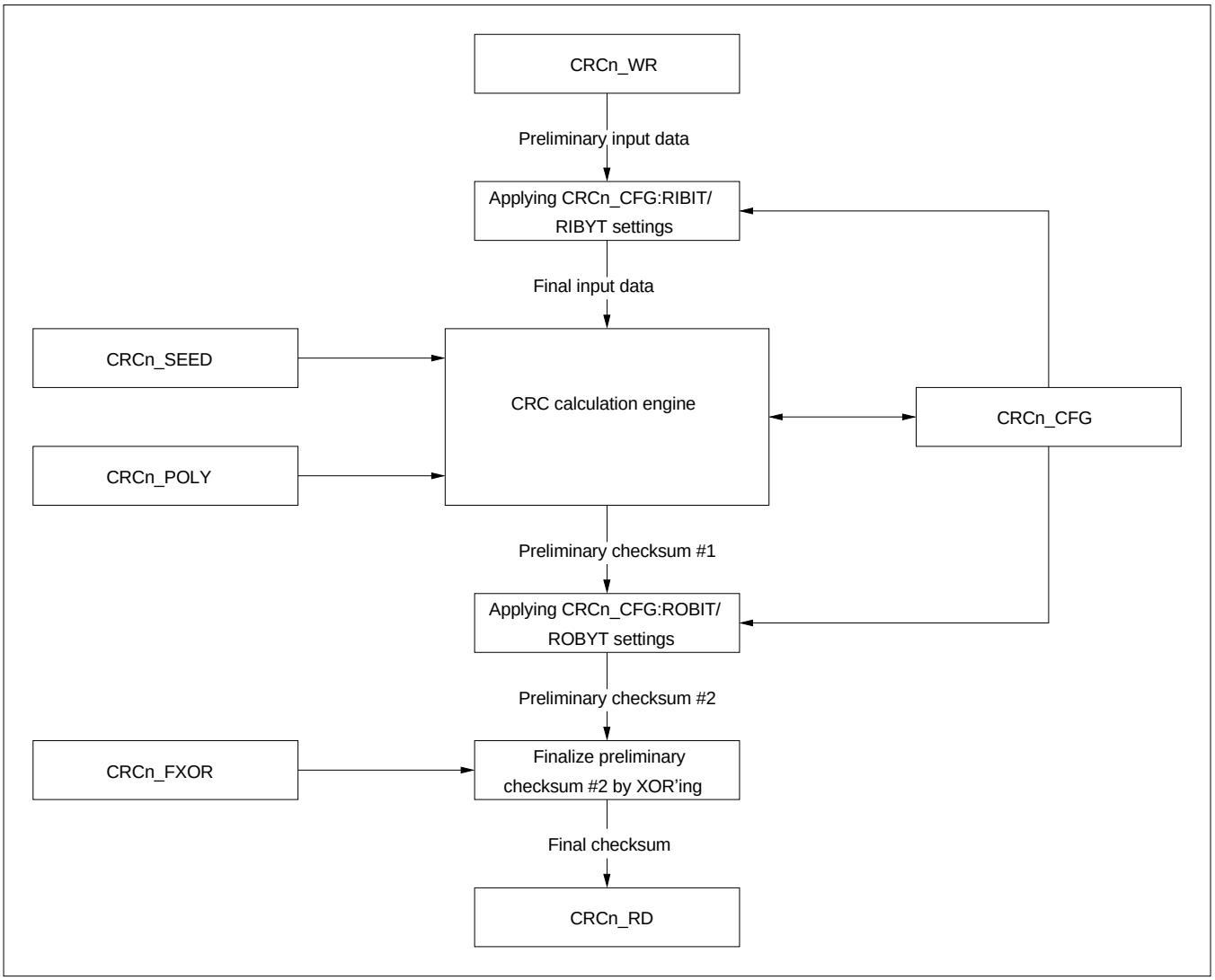


Figure 3-3 CRC Operation with DMA Request



3.2. CRC Input Data and Checksum Calculation Flow

Figure 3-4 Block Diagram of CRC Input Data and Checksum Calculation Flow



1. The input data for which CRC is to be calculated is written to CRCn_WR register. This is the 'preliminary input data'.
2. The 'preliminary input data' bytes can be swapped/reflected bit-wise using CRCn_CFG:RIBIT and/or byte-wise using CRCn_CFG:RIBYT before they enter the CRC engine. The settings are shown below:

'preliminary input data' in CRCn_WR register:

A7----A0 B7----B0 C7----C0 D7----D0

If the input data size is less than 32-bit (SZ < '11'), then the remaining bits (8-,16-, or 24-bit) of the data are considered as don't care (X) as shown in below table.

Table 3-1 Preliminary Input Data Bit-wise and/or Byte-wise Reflection/Swapping

RIBYT	RIBIT	SZ	Final Input Data for CRC Engine			
			+3	+2	+1	+0
‘0’	‘0’	‘00’	XXXX XXXX	XXXX XXXX	XXXX XXXX	D7------- D0
		‘01’	XXXX XXXX	XXXX XXXX	C7--------C0	D7--------D0
		‘10’	XXXX XXXX	B7--------B0	C7--------C0	D7--------D0
		‘11’	A7--------A0	B7--------B0	C7--------C0	D7--------D0
	‘1’	‘00’	XXXX XXXX	XXXX XXXX	XXXX XXXX	D0--------D7
		‘01’	XXXX XXXX	XXXX XXXX	C0--------C7	D0--------D7
		‘10’	XXXX XXXX	B0--------B7	C0--------C7	D0--------D7
		‘11’	A0---------A7	B0--------B7	C0--------C7	D0--------D7
‘1’	‘0’	‘00’	XXXX XXXX	XXXX XXXX	XXXX XXXX	D7--------D0
		‘01’	XXXX XXXX	XXXX XXXX	D7--------D0	C7--------C0
		‘10’	XXXX XXXX	D7--------D0	C7--------C0	B7--------B0
		‘11’	D7---------D0	C7--------C0	B7--------B0	A7--------A0
	‘1’	‘00’	XXXX XXXX	XXXX XXXX	XXXX XXXX	D0--------D7
		‘01’	XXXX XXXX	XXXX XXXX	D0--------D7	C0--------C7
		‘10’	XXXX XXXX	D0--------D7	C0--------C7	B0--------B7
		‘11’	D0---------D7	C0--------C7	B0--------B7	A0--------A7

3. The 'preliminary input data' after applying the settings of CRCn_CFG:RIBIT/RIBYT results in the 'final input data', which is sent to the CRC engine for checksum calculation.
4. The CRCn_SEED register provides the initial value to the CRC engine. The required polynomial is provided by CRCn_POLY register. The CRC engine starts its operation once CRCn_WR register is written with the input data.
5. CRC engine performance: The performance of CRC engine for CRC checksum calculation is based on the input data size and number of clock cycles (bus clock) required to complete a calculation. The shows number of clock cycles required to get final checksum at CRCn_RD register with respect to input data size.

Table 3-2 Clock Cycles Requirement for Checksum Calculation

Input Data Size	Number of Clock Cycles Required for Final Checksum at CRCn_RD
8-bit	Input data size (8-bit) + 2 = 10 clock cycles.
16-bit	Input data size (16-bit) + 2 = 18 clock cycles.
24-bit	Input data size (24-bit) + 2 = 26 clock cycles.
32-bit	Input data size (32-bit) + 2 = 34 clock cycles.

6. The 'preliminary checksum #1' bytes can be swapped/reflected bit-wise using CRCn_CFG:ROBIT and/or byte-wise using CRCn_CFG:ROBYT. shows at which positions the checksum bits of 'preliminary checksum #1' S[(LEN-1):0] will be located in 'preliminary checksum #2' after CRCn_CFG:ROBIT/ROBYT settings have been applied. Only some examples of different CRCn_CFG:LEN configurations are shown.

Note:

- *Only some examples for CRCn_CFG:LEN are shown.*

Table 3-3 Preliminary Checksum #1 Bit-wise and/or Byte-wise Reflection/Swapping

ROBYT	ROBIT	LEN	Preliminary Checksum #2				
			+3	+2	+1	+0	Action
‘0’	‘0’	32	S31---S24	S23---S16	S15---S8	S7---S0	No swapping/reflection. The checksum is aligned with the polynomial degree/length.
		21	‘0000 0000’	‘000 S20---S16’	S15---S8	S7---S0	No swapping/reflection. The checksum is aligned with the polynomial degree/length. The bits S21 to S31 are ’0’.
		16	‘0000 0000’	‘0000 0000’	S15---S8	S7---S0	No swapping/reflection. The checksum is aligned with the polynomial degree/length. The bits S16 to S31 are ’0’.
		3	‘0000 0000’	‘0000 0000’	‘0000 0000’	‘00000 S2--S0’	No swapping/reflection. The checksum is aligned with the polynomial degree/length. The bits S3 to S31 are ’0’.
	‘1’	32	S24---S31	S16---S23	S8---S15	S0---S7	Byte aligned checksum reflection.
		21	‘0000 0000’	‘S16---S20 000’	S8---S15	S0---S7	Bit reflection. The checksum is byte aligned. Bit S21-S23 and S24-S31 are ‘0’.
		16	‘0000 0000’	‘0000 0000’	S8---S15	S0---S7	Bit reflection. The checksum is byte aligned. Bit S16-S31 are ‘0’.
		3	‘0000 0000’	‘0000 0000’	‘0000 0000’	‘S0---S2 00000’	Bit reflection. The checksum is byte aligned. Bit S3-S7 and S8-S31 are ‘0’.

ROBYT	ROBIT	LEN	Preliminary Checksum #2				
			+3	+2	+1	+0	Action
‘1’	‘0’	32	S7---S0	S15---S8	S23---S16	S31---S24	Byte aligned checksum swapping.
		21	‘0000 0000’	S7---S0	S15---S8	‘000 S20---S16’	Byte swapping. The checksum is byte aligned. Bit S21-S23 and S24-S31 are ‘0’.
		16	‘0000 0000’	‘0000 0000’	S7---S0	S15---S8	Byte aligned checksum swapping. Bit S16-S31 are ‘0’.
		3	‘0000 0000’	‘0000 0000’	‘0000 0000’	‘00000 S2_S0’	No byte swapping. Bit S3-S7 and S8-S31 are ‘0’.
	‘1’	32	S0---S7	S8---S15	S16---S23	S24---S31	Bit reflection and byte swapping aligned with polynomial degree/length.
		21	‘0000 0000’	‘000 S0---S4’	S5---S12	S13---S20	Bit reflection and byte swapping. The checksum is aligned with polynomial length/degree. Bit S21-S23 and S24-S31 are ‘0’.
		16	‘0000 0000’	‘0000 0000’	S0---S7	S8---S15	Bit reflection and byte swapping. The checksum is aligned with polynomial length/degree. Bit S16-S31 are ‘0’.
		3	‘0000 0000’	‘0000 0000’	‘0000 0000’	‘00000 S0---S2’	Bit reflection and byte swapping. The checksum is aligned with polynomial length/degree. Bit S3-S7 and S8-S31 are ‘0’.

7. The checksum after applying settings of CRCn_CFG:ROBIT/ROBYT is 'preliminary checksum #2'.
8. The 'preliminary checksum #2' is XOR’ed with the contents of CRCn_FXOR register to get the 'final checksum'.
9. The 'final checksum' gets available at CRCn_RD register.

3.3. CRC Calculation Example

Consider the following values for calculating 8-bit CRC checksum value.

Input data = 0x0F (Hex)

Polynomial = $x^8 + x^2 + x + 1$

Seed = 0xFF (Hex)

Final XOR = 0x00 (Hex)

The coefficients of the polynomial are arranged in .

Table 3-4 Coefficients of the Polynomial

x8	x7	x6	x5	x4	x3	x2	x1	x0
1	0	0	0	0	0	1	1	1

The highest order coefficient x8 provides the degree of the CRC polynomial and the checksum length, respectively. It must not be set to ’1’ while configuring CRCn_POLY register, instead CRCn_CFG:LEN should be configured (here it is CRCn_CFG:LEN = 8). Therefore, the value of the polynomial that is written to CRCn_POLY register in accordance with above coefficients is 0x07 (Hex).

The input/output bit reflection is disabled in this example.

The Programmable CRC registers should be configured as follows for the given values:

The CRC configuration register is configured by considering 8-bit input data size and 8-bit polynomial/checksum length as follows.

CRCn_CFG = 0x00080000 (Hex)

CRCn_POLY = 0x00000007 (Hex)

CRCn_SEED = 0x000000FF (Hex)

CRCn_FXOR = 0x00000000 (Hex)

CRCn_WR = 0x0000000F (Hex)

The final result of CRC checksum calculation is 0xDE (Hex), which gets available after 10 clock cycles (once CRCn_WR is written) in the CRCn_RD register.

If another input data is given to the CRC module, then ’preliminary checksum #1’ (0xDE) is used as the initial seed value.

If the new CRC calculation should start from the seed value instead of from the last CRC result, then the CRCn_SEED register needs to be re-written (even if it is the same seed value as before).

4. Registers

All Programmable CRC registers are explained in this section.

The suffix 'n' in the register name indicates that the register is an instance 'n' of the module.

Registers of Programmable CRC

The following registers are available for each instance of Programmable CRC:

- CRC Polynomial Register (CRCn_POLY)
- CRC Seed Register (CRCn_SEED)
- CRC Final XOR Register (CRCn_FXOR)
- CRC Configuration Register (CRCn_CFG)
- CRC Write Register (CRCn_WR)
- CRC Read Register (CRCn_RD)

Memory layout of Programmable CRC registers

Figure 4-1 Memory Layout of Programmable CRC Registers

Offset	+3	+2	+1	+0
0x00000000	CRCn_POLY 00000100 11000001 00011101 10110111			
0x00000004	CRCn_SEED 11111111 11111111 11111111 11111111			
0x00000008	CRCn_FXOR 11111111 11111111 11111111 11111111			
0x0000000C	CRCn_CFG 00000000 11100000 00000000 00000000			
0x00000010	CRCn_WR 00000000 00000000 00000000 00000000			
0x00000014	CRCn_RD 00000000 00000000 00000000 00000000			

4.1. CRC Polynomial Register (CRCn_POLY)

The CRC Polynomial Register (CRCn_POLY) defines the polynomial value for the CRC checksum calculation.

CRC Polynomial Register (CRCn_POLY)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	POLY[31]	POLY[30]	POLY[29]	POLY[28]	POLY[27]	POLY[26]	POLY[25]	POLY[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	1	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	POLY[23]	POLY[22]	POLY[21]	POLY[20]	POLY[19]	POLY[18]	POLY[17]	POLY[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	0	0	0	0	0	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	POLY[15]	POLY[14]	POLY[13]	POLY[12]	POLY[11]	POLY[10]	POLY[9]	POLY[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	1	1	1	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	POLY[7]	POLY[6]	POLY[5]	POLY[4]	POLY[3]	POLY[2]	POLY[1]	POLY[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	0	1	1	0	1	1	1

[bit31:0] POLY[31:0] : CRC Polynomial

The CRCn_POLY contains the CRC polynomial. The degree of the polynomial must be between 2 to 32. Initial value is the common CRC-32 polynomial 0x04C11DB7 (x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1). The highest degree of coefficient should be used to configure polynomial/checksum length (CRCn_CFG:LEN).

Note:

- *If the polynomial length as defined by CRCn_CFG:LEN is less than 32-bit, then the upper bits [31:LEN] must be written to '0' by the programmer. The highest order degree must not be set to '1' while configuring CRCn_POLY register, as it is implicitly defined by CRCn_CFG:LEN.*

4.2. CRC Seed Register (CRCn_SEED)

The CRC Seed Register (CRCn_SEED) defines the initial value for the CRC checksum calculation.

CRC Seed Register (CRCn_SEED)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	SEED[31]	SEED[30]	SEED[29]	SEED[28]	SEED[27]	SEED[26]	SEED[25]	SEED[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SEED[23]	SEED[22]	SEED[21]	SEED[20]	SEED[19]	SEED[18]	SEED[17]	SEED[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	SEED[15]	SEED[14]	SEED[13]	SEED[12]	SEED[11]	SEED[10]	SEED[9]	SEED[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	SEED[7]	SEED[6]	SEED[5]	SEED[4]	SEED[3]	SEED[2]	SEED[1]	SEED[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:0] SEED[31:0] : CRC SEED

CRCn_SEED contains the initial value for checksum calculation. If the seed value is not initialized for the next operation (even if the seed value is identical to the previous operation), then calculation is continued from the last state with the current checksum value as initial value. Initial value for seed register is 0xFFFFFFFF.

Note:

- *The CRCn_SEED register should be configured with respect to the polynomial length (CRCn_CFG:LEN). If the polynomial length is less than 32-bit, then the upper bits [31:LEN] must be written to '0' by the programmer.*

4.3. CRC Final XOR Register (CRCn_FXOR)

The CRC Final XOR register (CRCn_FXOR) contains the values to be XOR'ed with the preliminary checksum to finalize the CRC calculation

CRC Final XOR Register (CRCn_FXOR)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FXOR[31]	[30]	FXOR[29]	FXOR[28]	FXOR[27]	FXOR[26]	FXOR[25]	FXOR[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FXOR[23]	FXOR[22]	FXOR[21]	FXOR[20]	FXOR[19]	FXOR[18]	FXOR[17]	FXOR[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FXOR[15]	FXOR[14]	FXOR[13]	FXOR[12]	FXOR[11]	FXOR[10]	FXOR[9]	FXOR[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FXOR[7]	FXOR[6]	FXOR[5]	FXOR[4]	FXOR[3]	FXOR[2]	FXOR[1]	FXOR[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	1	1	1	1	1

[bit31:0] FXOR [31:0] : CRC XOR Data

The contents of CRCn_FXOR register are XOR'ed with the preliminary checksum data (after CRCn_CFG:ROBIT/ROBYT settings have been applied) and then the final checksum is moved to CRCn_RD register. Initial value for final XOR register is 0xFFFFFFFF.

Note:

- *The bits of this register affect the corresponding bits of the CRCn_RD register. Therefore, the bits not belonging to the checksum should be written to '0'. For the position of the checksum bits depending on the used output bit/byte reflection refer to Table 3-3.*

4.4. CRC Configuration Register (CRCn_CFG)

The CRC Configuration Register (CRCn_CFG) is used to set the operation mode of the CRC module. CRCn_CFG register describes the polynomial/checksum length, input data size, and input/output bit/byte reflection. It indicates the status of CRC operation. Interrupt and DMA requests are also configured using CRCn_CFG register.

CRC Configuration Register (CRCn_CFG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	LOCK	read0	CDEN	CIEN	CIRQ
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R0,WX	R/W	R/W	R,WX
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	SZ[1]	SZ[0]	LEN[5]	LEN[4]	LEN[3]	LEN[2]	LEN[1]	LEN[0]
ACCESS_TYPE	R/W	R7W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	1	1	1	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	RIBIT	RIBYT	ROBIT	ROBYT
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	CIRQCLR
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] read0 : -

[bit28] LOCK : CRC Engine Status bit

This bit indicates the status of the CRC engine.

bit	Description
0	CRC engine is ready, new data can be written to CRC registers
1	CRC engine is busy and writing to CRC registers is not possible. If the data is written to the CRC registers when LOCK bit is '1', then an error response is generated

[bit27] read0 : -

[bit26] CDEN : DMA Request Enable bit

This bit enables/disables the DMA request

bit	Description
0	Disable the DMA request
1	Enable the DMA request

DMA request is generated when CRC is in buffer empty state and CRCn_CFG:CDEN is set. DMA ISR should clear this bit after final transfer. Clearing this bit by CPU at any other time might lead to unwanted behavior.

[bit25] CIEN : CRC Interrupt Enable bit to CPU

This bit enables/disables the interrupt request.

bit	Description
0	Disable the interrupt request
1	Enable the interrupt request

The IRQ is triggered when CRCn_CFG:CIEN bit is enabled and CRC interrupt flag (CRCn_CFG:CIRQ) is set.

[bit24] CIRQ : CRC Interrupt Flag

This bit indicates the interrupt status of CRC.

bit	Description
0	No interrupt request Note: CPU clears interrupt flag by writing CRCn_CFG:CIRQCLR bit to '1':
1	Interrupt request

Checksum has been calculated by CRC engine and is available in CRCn_RD register.

[bit23:22] SZ[1:0] : CRC Input Data Size Configuration bits

These bits are used to configure the input data size as follows

bit[1:0]	Description
00	8-bit
01	16-bit
10	24-bit
11	32-bit

[bit21:16] LEN[5:0] : CRC Polynomial/Checksum Length Configuration bits

These bits are used to configure the length (degree) of CRC polynomial/checksum as follows:

bit[5:0]	Description
100000	32
011111	31
.....	.....
000010	2

Notes:

- *The following settings are not supported:*
- *CRCn_CFG:LEN > 32*
- *CRCn_CFG:LEN < 2*

[bit15:12] read0 : -

[bit11] RIBIT : Reflect Input Bits

bit	Description
0	Disable input bit reflection
1	Enable input bit reflection

When the input data in the CRCn_WR register is passed to the CRC engine, the bit ordering of each byte within input data is reversed. For more details refer to Section 'Operation of the Programmable CRC'.

[bit10] RIBYT : Reflect Input Bytes

bit	Description
0	Disable input byte reflection (swapping)
1	Enable input byte reflection (swapping)

When the input data in the CRCn_WR register is passed to the CRC engine, the byte ordering of input data is reversed. Only the bytes of the configured input data size CRCn_CFG:SZ are affected. For more details refer to Section ''Operation of the Programmable CRC'.

Note:

- *For 8-bit input data, this setting has no effect.*

[bit9] ROBIT : Reflect Output Bits

bit	Description
0	Disable output bit reflection
1	Enable output bit reflection

The bit ordering of each byte within checksum is reversed, before passing the checksum to final XOR'ing stage. For more details refer to Section ''Operation of the Programmable CRC'.

[bit8] ROBYT : Reflect Output Bytes

bit	Description
0	Disable output byte reflection (swapping)
1	Enable output byte reflection (swapping)

The byte ordering of checksum data is reversed, before passing the byte aligned polynomial length of checksum data to final XOR'ing stage. For more details refer to Section ''Operation of the Programmable CRC'.

Note:

- *For the checksum data less than or equal to 8 bits, this setting has no effect.*

[bit7:1] read0 : -

[bit0] CIRQCLR : Interrupt Clear

This bit clears the CRC interrupt flag.

bit	Description
0	Write '0' is ignored, reading this bit always returns '0'
1	Clear CRC interrupt flag (CRCn_CFG:CIRQ)

4.5. CRC Write Register (CRCn_WR)

The input data for the CRC checksum calculation must be written to the CRC Write Register (CRCn_WR).

CRC Write Register (CRCn_WR)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CRCWR[31]	CRCWR[30]	CRCWR[29]	CRCWR[28]	CRCWR[27]	CRCWR[26]	CRCWR[25]	CRCWR[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CRCWR[23]	CRCWR[22]	CRCWR[21]	CRCWR[20]	CRCWR[19]	CRCWR[18]	CRCWR[17]	CRCWR[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CRCWR[15]	CRCWR[14]	CRCWR[13]	CRCWR[12]	CRCWR[11]	CRCWR[10]	CRCWR[9]	CRCWR[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CRCWR[7]	CRCWR[6]	CRCWR[5]	CRCWR[4]	CRCWR[3]	CRCWR[2]	CRCWR[1]	CRCWR[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CRCWR[31:0] : CRC Write Register

The CRCn_WR register contains the input data, for which the CRC checksum is to be calculated. Writing to this register starts the CRC calculation process. After pre-processing (bit/byte reflection/swapping) the contents of the CRCn_WR register are passed to the CRC engine (the contents of CRCn_SEED register are also provided). There it is divided by the content of CRCn_POLY register in modulo-2 arithmetic to get the final checksum after post-processing (bit/byte reflection/swapping and XOR'ing).

Note:

- *The size of input data is configured by CRCn_CFG:SZ, where 8, 16, 24, and 32 bits are only supported as data size. If the input data size is less than 32-bit (i.e. 8, 16, or 24 bits), then the invalid/unused bits are considered as don't care (X).*

4.6. CRC Read Register (CRCn_RD)

The CRC Read Register (CRCn_RD) contains the final checksum of the data written to the CRCn_WR register.

CRC Read Register (CRCn_RD)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CRCRD[31]	CRCRD[30]	CRCRD[29]	CRCRD[28]	CRCRD[27]	CRCRD[26]	CRCRD[25]	CRCRD[24]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CRCRD[23]	CRCRD[22]	CRCRD[21]	CRCRD[20]	CRCRD[19]	CRCRD[18]	CRCRD[17]	CRCRD[16]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	CRCRD[15]	CRCRD[14]	CRCRD[13]	CRCRD[12]	CRCRD[11]	CRCRD[10]	CRCRD[9]	CRCRD[8]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CRCRD[7]	CRCRD[6]	CRCRD[5]	CRCRD[4]	CRCRD[3]	CRCRD[2]	CRCRD[1]	CRCRD[0]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] CRCRD[31:0]: CRC Read Data

The CRC Read Register contains the result (final checksum) of the CRC calculation after post-processing (applying CRCn_CFG:ROBIT, CRCn_CFG:ROBYT, and CRCn_FXOR settings). Writing any value on CRCn_RD register changes its content and it does not affect CRC calculation.

Note:

- *The polynomial length (CRCn_CFG:LEN) provides the length of the final checksum. If the polynomial length is less than 32-bit, then bits not belonging to the checksum are '0' (in case the invalid bits of the CRCn_FXOR register are not programmed to '0', then these bits in CRCn_RD register might also be '1'). The bit/byte reflection settings (CRCn_CFG:ROBIT/ROBYT) can influence the checksum in CRCn_RD register. For the position of the checksum bits refer to Table 3-3.*

CHAPTER 25: Clock Monitor

This chapter explains the clock monitor.

1. Overview
2. Configuration and Block Diagram
3. Operation of clock monitor
4. Registers
5. Precautions for Using This Device

CODE: CLKMON-JUPI-E1

1. Overview

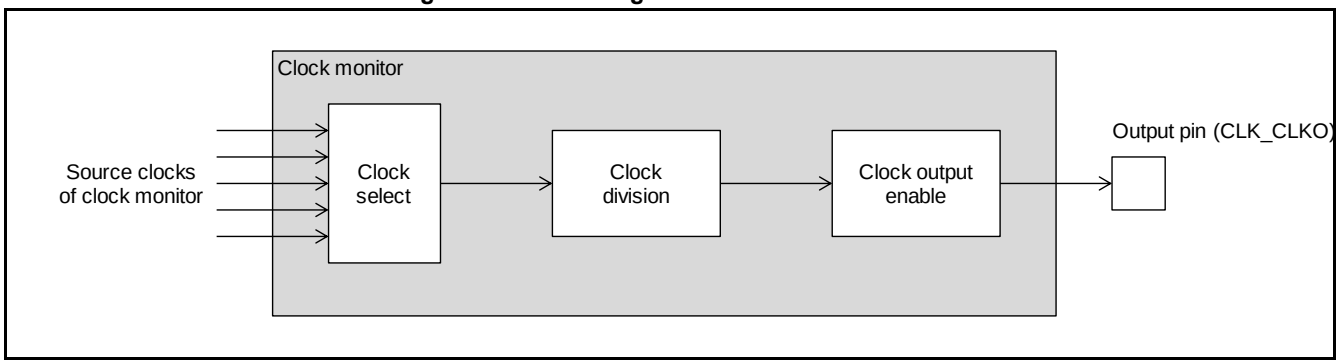
The clock monitor is a macro that outputs internal clock signals to external pin. The clock monitor has a function for dividing the frequency of a clock signal before output to the pin.

Features

- Format: Divide the internal clock signal and output to a pin (CLK_CLKO)
- Channel: 1
- Division ratio: CLK/1 to CLK/128
- Monitoring clock: Fast-CR clock, Slow-CR clock, Main clock, Sub clock, PLL0 clock, SSCG PLL0 clock and other internal clocks (CLK_CAN, CLK_LCP0, CLK_LCP0A, CLK_LCP1, CLK_LCP1A and CLK_SYSC0H)

2. Configuration and Block Diagram

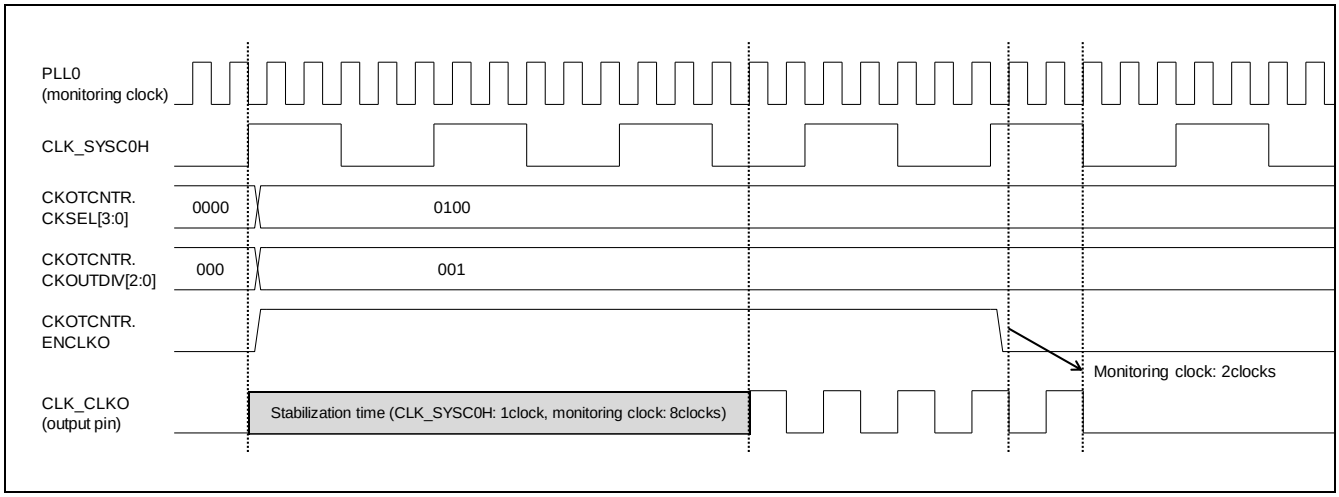
Figure 2-1 Block Diagram of the Clock Monitor



3. Operation of Clock Monitor

This section explains the operations of the clock monitor.

Figure 3-1 Example of Clock Monitor Timing Chart (Monitoring Clock: PLL0)



4. Registers

Table 4-1 Register List

Abbreviated Register Name	Register Name	Reference
CKOTCNTR	Clock output function control register	4.1

Table 4-2 Clock Monitor Memory Map

Offset Address	Register			
	+3	+2	+1	+0
0x00000000	CKOTCNTR 00000000 00000000 00000000 00000000			

4.1. Clock Output Function Control Register (CKOTCNTR)

Clock output function control register (CKOTCNTR) is used to control clock output function.

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ENCLKO
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	CKOUTDIV[2]	CKOUTDIV[1]	CKOUTDIV[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	CKSEL[3]	CKSEL[2]	CKSEL[1]	CKSEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:25] Reserved

This is a reserved bit. Writing data to these bits has no effect on operation.

[bit24] ENCLKO: Clock output enable bit

This bit controls enable/disable clock output function.

bit	Description
0	Clock output function disable
1	Clock output function enable

[bit23:11] Reserved

This is a reserved bit. Writing data to these bits has no effect on operation.

[bit10:8] CKOUTDIV[2:0]: Clock division bits

These bits configure the clock output divider.

bit10:8	Description
000	Output clock is not divided
001	Output clock is divide by 2
010	Output clock is divide by 4
011	Output clock is divide by 8
100	Output clock is divide by 16
101	Output clock is divide by 32
110	Output clock is divide by 64
111	Output clock is divide by 128

[bit7:4] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit3:0] CKSEL[3:0]: Clock select bits

These bits select the source clock for clock output function.

bit3:0	Description
0000	Fast-CR clock is selected
0001	Slow-CR clock is selected
0010	Main clock is selected
0011	Sub clock is selected
0100	PLL0 clock is selected
0101	CLK_CAN is selected
0110	CLK_LCP0 is selected
0111	CLK_LCP0A is selected
1000	SSCG PLL0 clock is selected
1001	CLK_LCP1 is selected
1010	CLK_LCP1A is selected
1011	CLK_SYSC0H is selected
1100	Prohibit (Fast-CR clock is selected)
1101	Prohibit (Fast-CR clock is selected)
1110	Prohibit (Fast-CR clock is selected)
1111	Clock is tied to low

5. Precautions

The following shows the precautions when using the clock monitor.

- The frequency which can output as monitor output has a limitation and see the datasheet. When monitoring clock is higher than the limitation frequency, always divide clock.
- Monitoring clock cannot change when selected clock or next-selected clock or CLK_SYSC0H stops.

CHAPTER 26: DDR High Speed SPI Controller

This chapter explains the functionality and operation of the DDR High Speed SPI Controller (DDRHSSPI).

1. Overview
2. Configuration
3. Serial Interface
4. Operations of the DDRHSSPI
5. Registers

CODE: DDRHSSPI-E01.92

1. Overview

The DDRHSSPI provides various operating modes for interfacing to serial peripheral devices that use the de-facto standard SPI protocol. It supports also the Quad and Octal transfer modes to access multi-bit Serial Flash Memories. The DDRHSSPI's features are described in this section.

Features of the DDR High Speed SPI Controller

- In case of DDRHSSPIn_MID.MID=0x00000001
  - Supports Legacy Mode and Quad Mode
  - Allows the use of up to 4 Serial Flash Memories
- In case of DDRHSSPIn_MID.MID=0x00000100 or 0x00000300
  - Supports Legacy Mode, Quad Mode and Octal Mode (Dual Quad, Dual Legacy)
  - Allows the use of up to 8 Serial Flash Memories
- Features a programmable transfer rate of the serial clock
- External Serial Flash Memories can be memory-mapped to the address-space of the MCU, in Command Sequencer Mode
- In Command Sequencer Mode, memory accesses initiated by the MCU and other masters are automatically converted to Serial Flash Memory read commands by the DDRHSSPI.
- In Command Sequencer Mode, access to the SPI Flash Memory is done through the Prefetch Buffer (PB)
- Direct Mode allows the DDRHSSPI to be used as a standard SPI master through a FIFO interface

Abbreviations

This section lists the terms and abbreviations used in this chapter.

Table 1-1 Terms and Abbreviations

Term	Meaning
Byte Time	Byte Time is the time required for transmission of 8 bits of data, over the SPI interface In SDR Mode, one byte time is: 1 cycle of SCLK in Octal Mode, 2 cycles of SCLK in Quad Mode and 8 cycles of SCLK in Legacy Mode. In DDR Mode one byte time is: 1/2 cycle of SCLK in Octal Mode, 1 cycle of SCLK in Quad Mode and 4 cycles of SCLK in Legacy Mode.
CSR	Control and Status Registers of DDRHSSPI.
DAP	Debug Access Port
DDR	Double Data Rate
DLP	Data Learning Pattern
DMA	Direct Memory Access
FIFO	First-In, First-Out
Half Word	16-bits of data
MCU	Micro Control Unit
on-the-fly	Switching modes of serial interface, during transferring data there
PB	Prefetch Buffer
RX	Receive
SDR	Standard Data Rate
SPI	Serial Peripheral Interface
SCLK	Serial Clock on the external port of the chip
SSEL	Serial Slave Select on the external port of the chip
SDATA	Serial Data on the external port of the chip
SW	Software
TX	Transmit
Word	32-bits of data

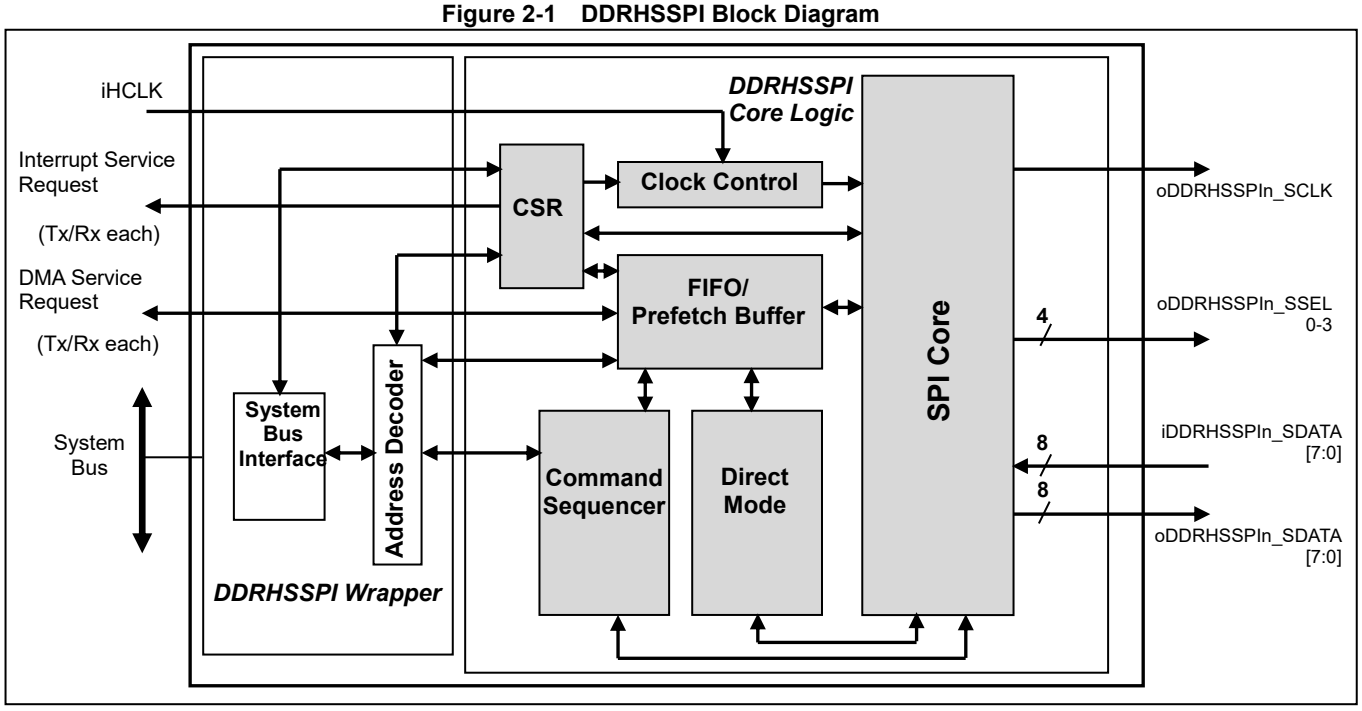
2. Configuration

This section describes a block diagram of DDRHSSPI.

Block Diagram

Figure 2-1 shows the block diagram of the DDRHSSPI, followed by the explanations of internal blocks.

Figure 2-1 DDRHSSPI Block Diagram



CSR

The CSR block holds the configuration and status registers, which are used to control and monitor the status of the DDRHSSPI.

Please refer to section 4.3 Address Map of DDRHSSPI for more detailed information about the CSR.

SPI Core and Clock Control

The SPI Core block contains the SPI protocol engine, which handles the read and write operations from/to the external serial Flash Memory. The SPI communication-related attributes like Serial Clock Frequency and Serial Clock Mode, etc., are configured via the CSR. Depending on whether the DDRHSSPI is operating in Direct Mode or in Command Sequencer Mode, the SPI Core connects with the FIFOs. The DDRHSSPI can initiate serial transfers with Serial Flash Memories, which are connected to the four Slave Select lines: oDDRHSSPIn_SSEL0-3. An internal clock divider is used to derive the serial clock output (oDDRHSSPIn_SCLK).

System Bus and Address Decoding

These two blocks are used as the interface to the System Bus. The masters such as MCUs are able to access the DDRHSSPI through the System Bus Interface and Address Decoder.

The Address Decoder performs the decoding of the address bus. If the System Bus access is targeted to a command or a status register, the request is sent to the CSR block. If the access is for the Command Sequencer or the Direct Mode blocks, then also access relevant info is routed to Command Sequencer or Direct Mode. When in Command Sequencer Mode, if the System Bus access is for a Serial Flash Memory which is mapped onto one of the four Slave Select lines, then the memory address is passed to the Command Sequencer block.

FIFO / Prefetch Buffer

This block can be used in two different modes based on DDRHSSPI configured mode, Command Sequencer Mode or Direct Mode. In Direct Mode, this block is used as two separate FIFOs: One for TX and one for RX. Each FIFO is 24-locations deep and has a data-width of 32 bits.

In Command Sequencer Mode this block is used as Prefetch Buffer for only receiving data from Flash Memory. The Prefetch Buffer is 48-locations deep and has data width of 32 bits.

Direct Mode

In Direct Mode DDRHSSPI internally uses FIFO/Prefetch buffer block configured as two FIFOs for temporary storage: One holds the data to be transmitted (TX-FIFO) and one stores the data to be received (RX-FIFO).

TX-FIFO and RX-FIFO are used by DDRHSSPI only in Direct Mode.

The Direct Mode block has following functions:

- Transmits data onto the Serial Interface via TX-FIFO.
- Receives data from the Serial Interface via RX-FIFO.
- DMA is available both for transmission and reception.
- Controls the data format to transmit.
  - Tri-state with variable length
  - SPI protocol on each data
  - Data Rate Mode (SDR Mode or DDR Mode) on each data

Command Sequencer

The Command Sequencer maps the external Serial Flash Memories on the address space of MCU.

In this mode the FIFO/Prefetch Buffer block is configured as Prefetch Buffer (PB) available for temporary storage and prefetch accesses, and only read operations of the Flash Memory are possible. The read data are output to the System Bus via internal Prefetch Buffer.

The Command Sequencer block has following functions:

- Watches continuity of memory access on the System Bus and determines whether it should keep current serial transfer or start a new one.
- Generates a serial transaction with
  - Decoding a Command Sequence table in CSR to output a series of data to the SPI Core.
  - Generating a serial address: it concatenates address bits from System Bus and a register value in CSR.
- Controls Prefetch Buffer to flush.
- Runs an Idle Timer (ITIMER) to control whether to close current serial transaction.

3. Serial Interface

This section describes the Serial Interface of DDRHSSPI.

3.1. Serial Clock Modes and Data Rate Modes

DDRHSSPI supports two Data Rate Modes for Serial Flash Interface, namely Single Data Rate (SDR) and Double Data Rate (DDR). In Direct Mode, the DDR is allowed only at TX-Only Mode.

Maximum output serial clock frequency in both SDR and DDR Modes, is up to 1/2 peripheral bus frequency.

Based on the programmed values of the DDRHSSPIn_PCC0-3.ACES bits (valid only in SDR Mode), each peripheral can have up to 2 clock modes in SDR Mode and only 1 clock mode in DDR Mode. DDRHSSPIn_PCC0-3.ACES bit setting in DDR Mode is not allowed. These bits decide the serial data input and output timings of DDRHSSPI, with respect to the serial SPI clock. This is explained in Table 3-1 and Table 3-2.

Table 3-1 Clock Modes in SDR Mode

MODE	ACES (Active Clock Edges are Same on Peripheral)	Description
Mode 0	0	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one half-cycle before the first positive edge of serial clock (oDDRHSSPIn_SCLK) and on the subsequent **negative** edges of oDDRHSSPIn_SCLK.
		Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **positive** edges of oDDRHSSPIn_SCLK.
Mode 4	1	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one half-cycle before the first positive edge of serial clock (oDDRHSSPIn_SCLK) and on the subsequent **negative** edges of oDDRHSSPIn_SCLK.
		Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **negative** edges of oDDRHSSPIn_SCLK.

Figure 3-1 SDR Clock Mode 0

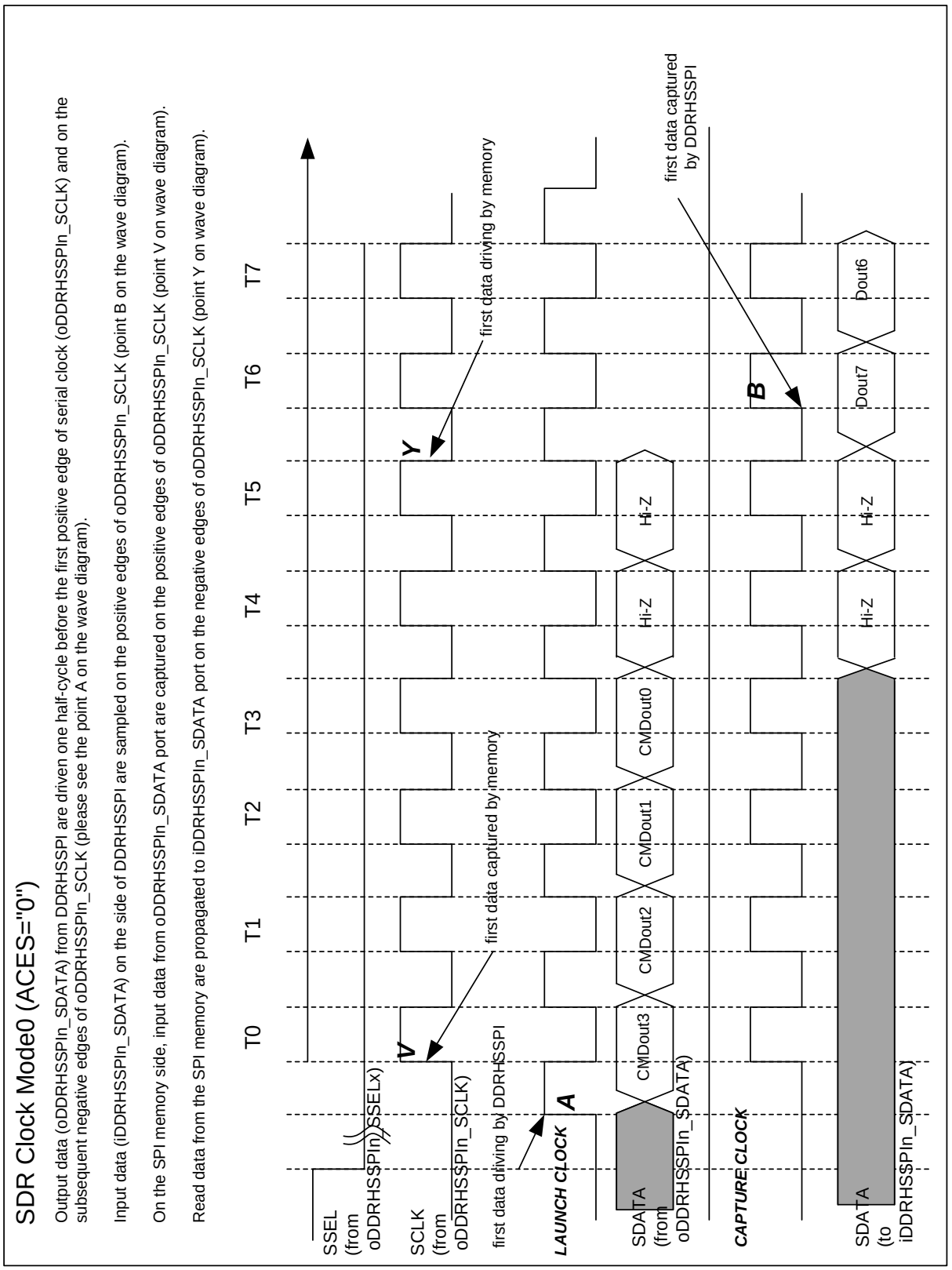


Figure 3-2 SDR Clock Mode 4

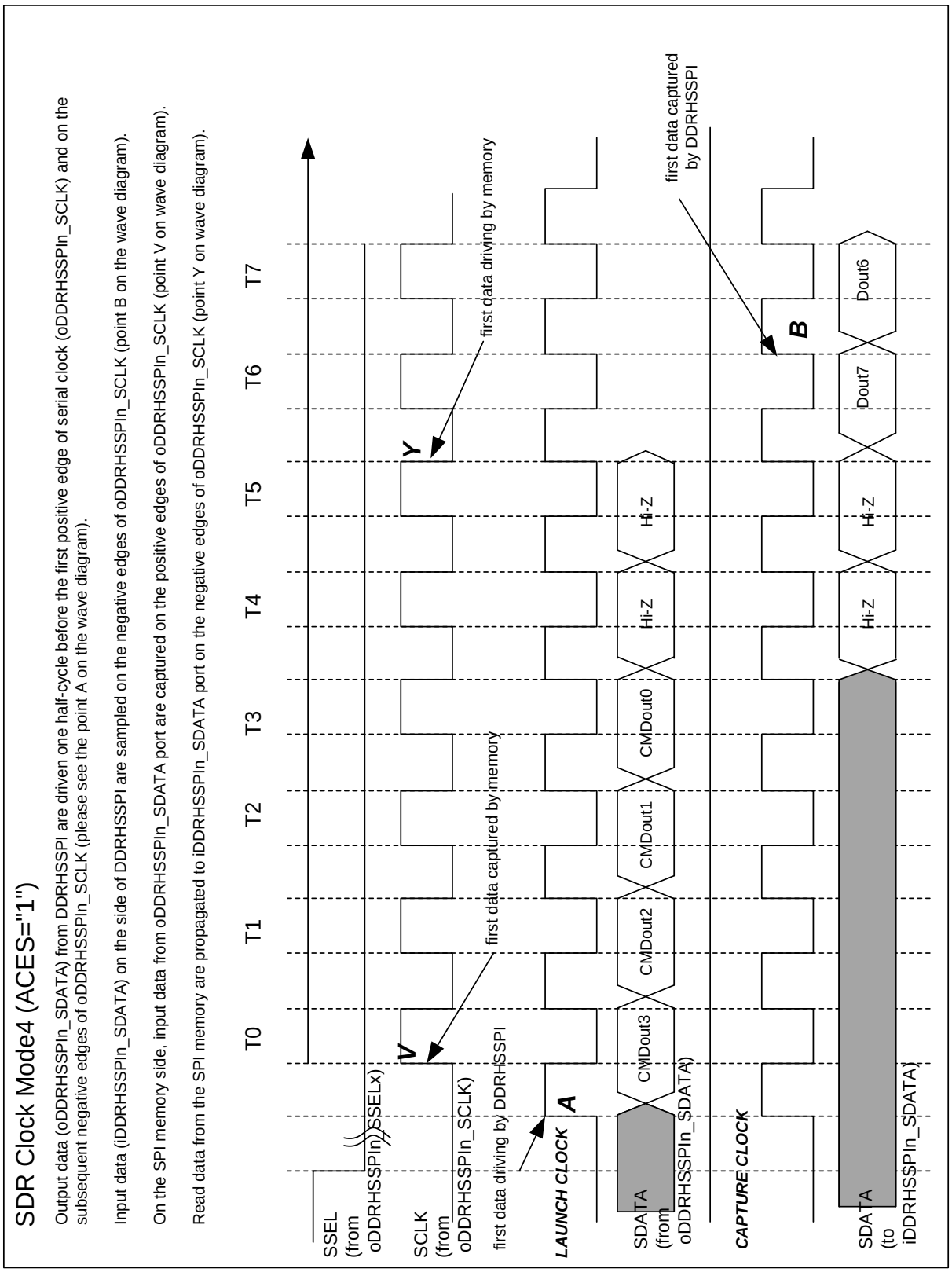


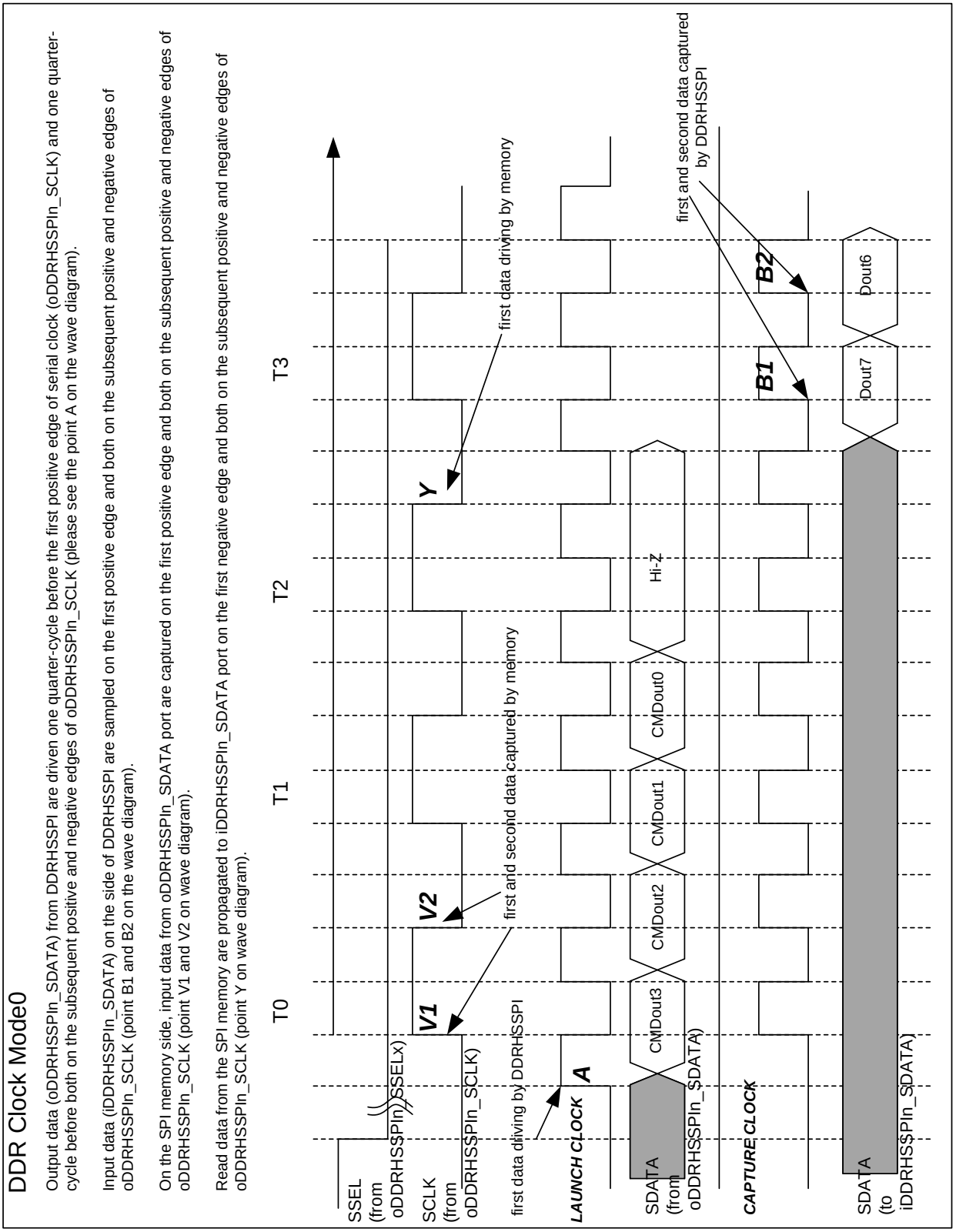
Table 3-2 Clock Modes in DDR Mode

MODE	Description
Mode 0	Output data (oDDRHSSPIn_SDATA) from DDRHSSPI are driven one quarter-cycle before the **first positive edge** of serial clock (oDDRHSSPIn_SCLK) and one quarter-cycle before both on the **subsequent positive and negative edges** of oDDRHSSPIn_SCLK.
	Input data (iDDRHSSPIn_SDATA) on the side of DDRHSSPI are sampled on the **first positive edge,** and both on the **subsequent positive and negative edges** of oDDRHSSPIn_SCLK.

Note:

- *In Direct Mode, the DDR is allowed only at TX-Only Mode.*

Figure 3-3 DDR Clock Mode 0



3.2. SPI Clock Frequency

oDDRHSSPIn_SCLK clock is internally generated by dividing the System Bus Clock (iHCLK). The clock division ratio of the resulting internal clock-divider can be programmed in the DDRHSSPIn_PCC0-3 Registers. The DDRHSSPIn_PCC0-3.CDRS (4 bit wide) decides the clock division ratio for output serial clock frequency. Maximum clock division rate supported is 32.

For CDRS value 0 output serial clock frequency SCLK is 1/2 of iHCLK

For CDRS value 1 output serial clock frequency SCLK is 1/4 of iHCLK.

.....

For CDRS value 15 output serial clock frequency SCLK is 1/32 of iHCLK.

Output clock frequency set by CDRS is independent of Data Rate Mode (SDR Mode or DDR Mode) settings.

3.3. SPI Input Data Sampling Point

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The delay of internal clock which samples iDDRHSSPIn_SDATA is adjustable, comparing to oDDRHSSPIn_SCLK clock. By this feature, the sampling timing of input data from Flash Memory can be finely tuned. The SPI clock sampling point per each bit line of iDDRHSSPIn_SDATA port can be programmed in the DDRHSSPIn_SDATASAMPLEPTCNT0-7 Registers.

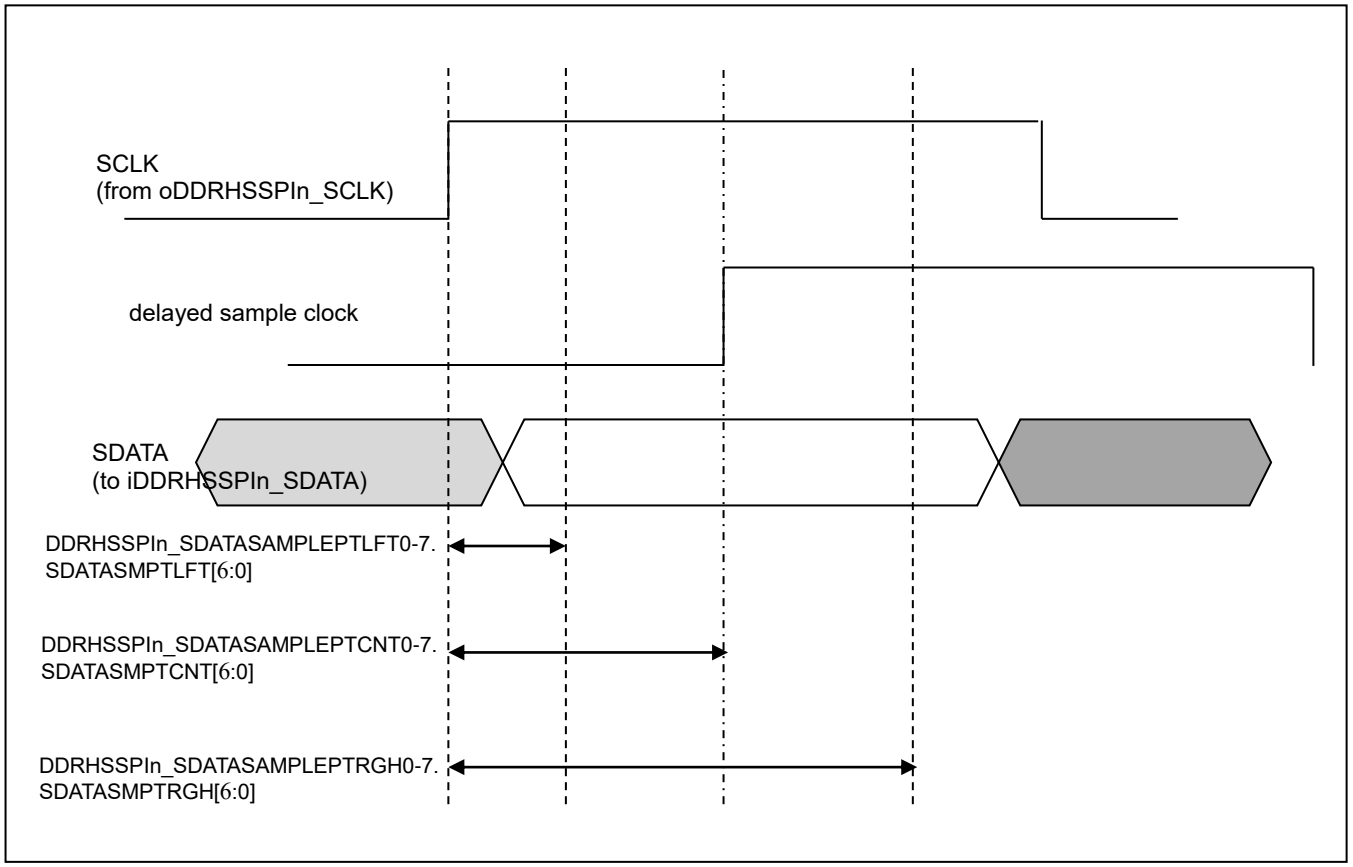
The resolution for the fine-tuning of the sampling point is fixed and it is equivalent to one buffer delay. Please consult physical implementation library datasheet for exact delay inserted with each buffer.

Sampling clock fine tuning is implemented for each iDDRHSSPIn_SDATA[7:0] input line with buffers forming delay chain (on the internal clock) with programmable delay selection. It is up to SW to choose appropriate value of iDDRHSSPIn_SDATA clock sample delay point in order to provide consistency and compensate for input delay per data lines.

When in using DLP, it is possible to set additional two clock sampling points per each iDDRHSSPIn_SDATA input (so called left and right sampling point) for checking purpose only. These values are set in DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7 Registers.

Please refer to Figure 3-4 explaining this in wave diagrams.

Figure 3-4 SDATA Sampling Clock Settings



Please note that setting delay is always done comparing to the rising edge of oDDRHSSPIn_SCLK clock and it is not possible to set this to be "negative" i.e. to be before oDDRHSSPIn_SCLK rising edge.

It is recommended to set values of corresponding registers such that SDATASMPTLFT <= SDATASMPTCNT<= SDATASMPTRGH.

Note:

- *When in DDRHSSPIn_MID.MID is 0x00000001, all of SDATASMPTLFT, SDATASMPTCNT and SDATASMPTRGH must be fixed to 0.*

3.4. SPI Data Learning Pattern

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

In case DDRHSSPI is connected with Flash Memory that has a feature outputting a Data Learning Pattern (DLP) during the dummy cycles, DDRHSSPI can check if the sampling point setting is correct.

This feature can be used only in Command Sequencer Mode and DDR Mode.

This feature is enable, when DDRHSSPIn_MCTRL.DLPEN bit is "1". This feature must be also enabled on the Flash Memory side. Please program the same value of DLP to both registers, in the Flash Memory and in DDRHSSPI (DDRHSSPIn_DLP Register). Once this feature is enabled, Flash Memory will output a Data Learning Pattern during dummy cycles, and DDRHSSPI will receive data on iDDRHSSPIn_SDATA at the same time.

Since DDRHSSPI has individual sampling points on each line of iDDRHSSPIn_SDATA, during the DLP pattern, DDRHSSPI can test neighboring (left-center-right) sampling points by registers DDRHSSPIn_SDATASAMPLEPTCNT0-7, DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7.

All checks in iDDRHSSPIn_SDATA sampling clock settings (left-center-right) are comparing to the value of DDRHSSPIn_DLP Register as reference.

3.5. SPI Data Protocol

DDRHSSPI supports Legacy (1-bit), Quad(4-bit) and Octal (8-bit) SPI protocols. The Quad and Octal Protocols are used for interfacing with Serial Flash Memories like Cypress FL-S series.

Legacy Protocol

The Legacy Protocol is a full-duplex protocol. When DDRHSSPI is configured with Legacy Protocol, the data can be received on a single input port (i.e. iDDRHSSPIn_SDATA[1]) and simultaneously, the data can also be transmitted on a single output port (i.e. oDDRHSSPIn_SDATA[0]). While Legacy Protocol is being used, the unused data lines (i.e. oDDRHSSPIn_SDATA[7:2]) are tri-stated by DDRHSSPI.

In Direct Mode, when DDRHSSPIn_DMTRP.TRP is configured for TX-and-RX Mode, it is allowed to use only the Legacy Mode.

In Command Sequencer Mode, the Legacy Protocol is not supported.

Quad Protocol

In Quad Protocol, four serial data lines (i.e. SDATA[3:0]) are used and SDATA[7:4] are tri-stated, in a half-duplex manner for accessing single device. Data transmission and reception cannot happen simultaneously.

In Direct Mode, when DDRHSSPIn_DMTRP.TRP is configured for "TX-Only in Quad Mode", the Quad Protocol is used.

In Command Sequencer Mode, the Quad Protocol is supported.

Octal Protocol

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

In Octal Protocol, eight serial data lines (i.e. SDATA[7:0]) are used, in a half-duplex manner for accessing two Serial Flash Memories. Data transmission and reception cannot happen simultaneously.

In this mode DDRHSSPI is attached to two identical Flash Memories with 4 data bits each. Both of these memories also must be configured with same configuration parameters. In this mode, one memory will be attached to SDATA[3:0] and the other to SDATA[7:4]. These memories will be connected to the same SSEL signal.

The data transfer of Octal Protocol is classified as below.

- Dual Legacy

This transfer type is available only in Direct Mode (TX-Only Mode and TX-and-RX Mode). The output ports oDDRHSSPIn_SDATA[0] and oDDRHSSPIn_SDATA[4] are used at Legacy Protocol each. And the input ports iDDRHSSPIn_SDATA[1] and iDDRHSSPIn_SDATA[5] are used at Legacy Protocol each, as well. Please refer to Table 4-1 for more details.

- Dual Quad

This transfer type is available both in Direct Mode (TX-Only Mode) and Command Sequencer Mode. In Direct Mode, this transfer type is not available in TX-and-RX Mode.

The output ports oDDRHSSPIn_SDATA[3:0] and oDDRHSSPIn_SDATA[7:4] are used at Quad Protocol each. And the input ports iDDRHSSPIn_SDATA[3:0] and iDDRHSSPIn_SDATA[7:4] are used at Quad Protocol each, as well. Please refer to Table 4-1 for more details.

In Command Sequencer Mode, the value of address field is divided in half, since the data are located in each Serial Flash Memory at every 2 bytes.

3.6. Shift Direction

Most Significant Bit in the Shift Register is transmitted first and the first received data is shifted into the Least Significant Bit in the Shift Register. i.e. the Shift Register is shifted left. The read or write accesses to the data-registers always have Least Significant Bit of the data in bit 0.

The following figuresFigure 4-7 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines. The waveforms assume DDRHSSPIn_DMFIFOCFG.FWIDTH = "00" in Direct Mode. The Figures depict that the transmit data is loaded into the Shift Register from the TX-FIFO. However, the source of transmit data can be other registers, such as DDRHSSPIn_RDCSDC0-11.RDCSDATA.

When DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI is available for Dual Legacy Mode and Dual Quad Mode. Figure 3-11 and Figure 3-12 depict the direction in which the data in the Shift Register is shifted to/from the serial data lines, under Dual Legacy Mode and Dual Quad Mode.

In Command Sequencer Mode, byte ordering is always LSB first.

In Direct Mode, byte ordering is always MSB first.

Figure 3-5 Quad Mode / byte ordering - Command Sequencer Mode

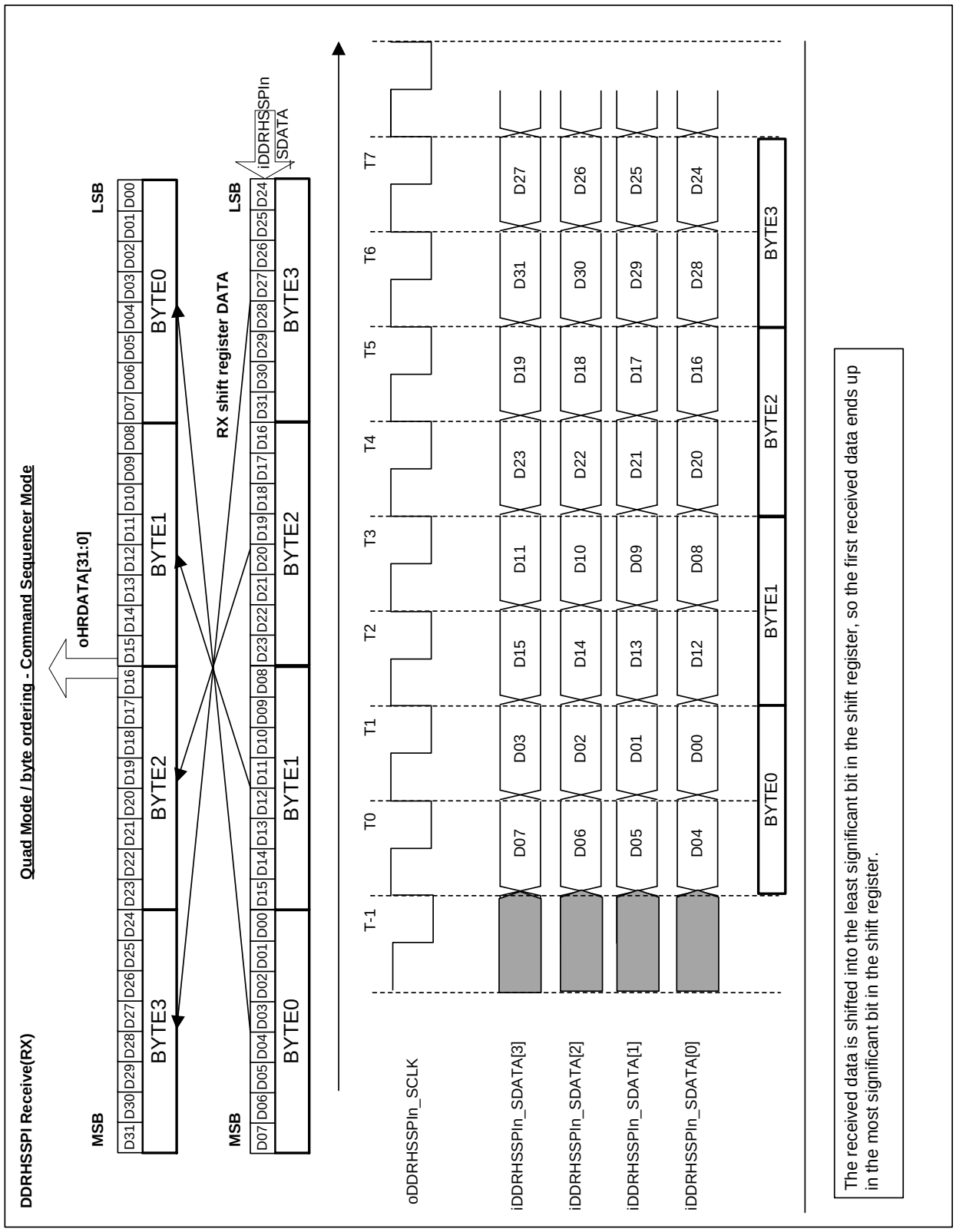


Figure 3-6 Quad Mode / Byte Ordering - Command Sequencer Mode (RX Register - In Time)

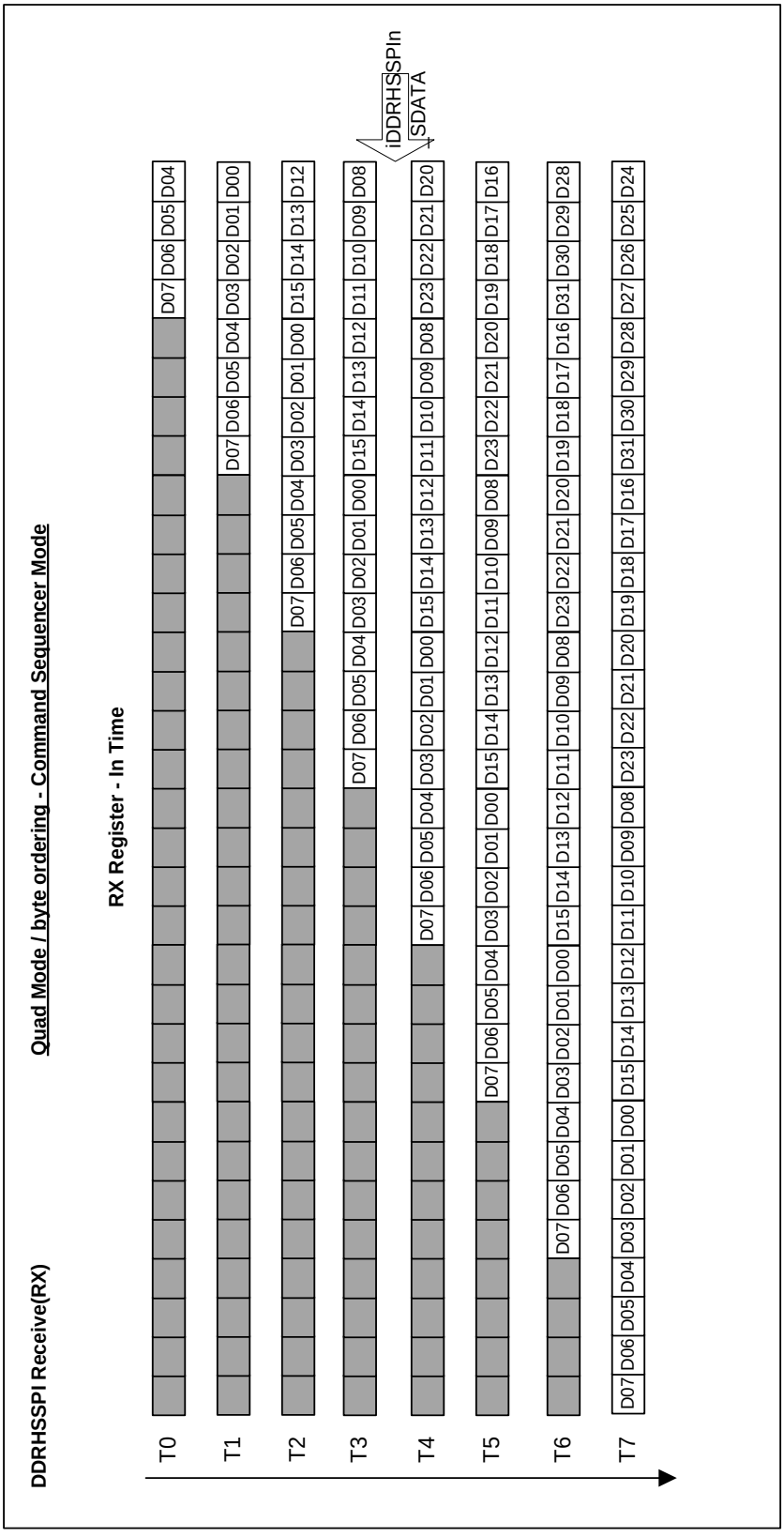


Figure 3-7 Quad Mode / Byte Ordering - Direct Mode

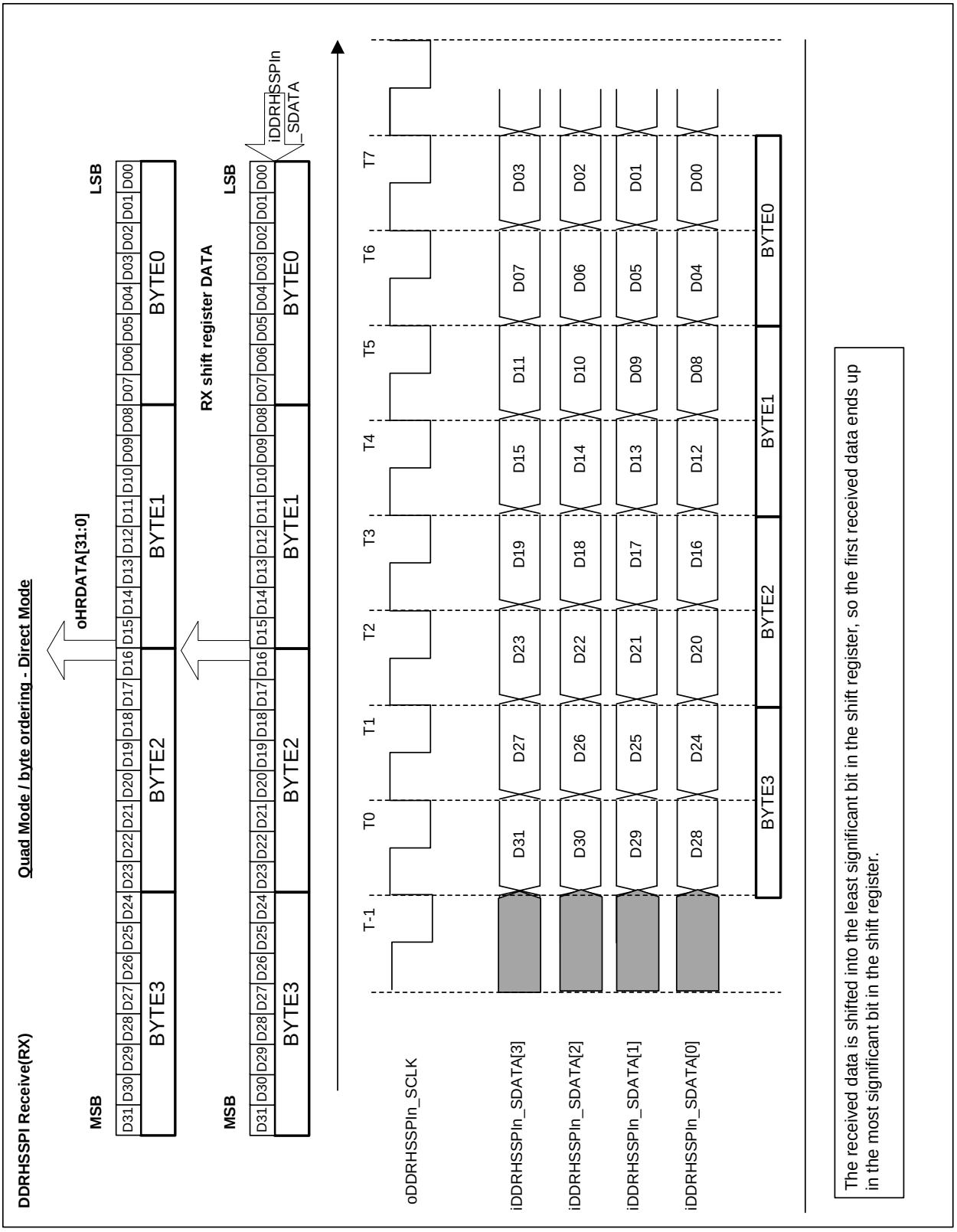


Figure 3-8 Quad Mode / Byte Ordering - Direct Mode (RX Register - In Time)

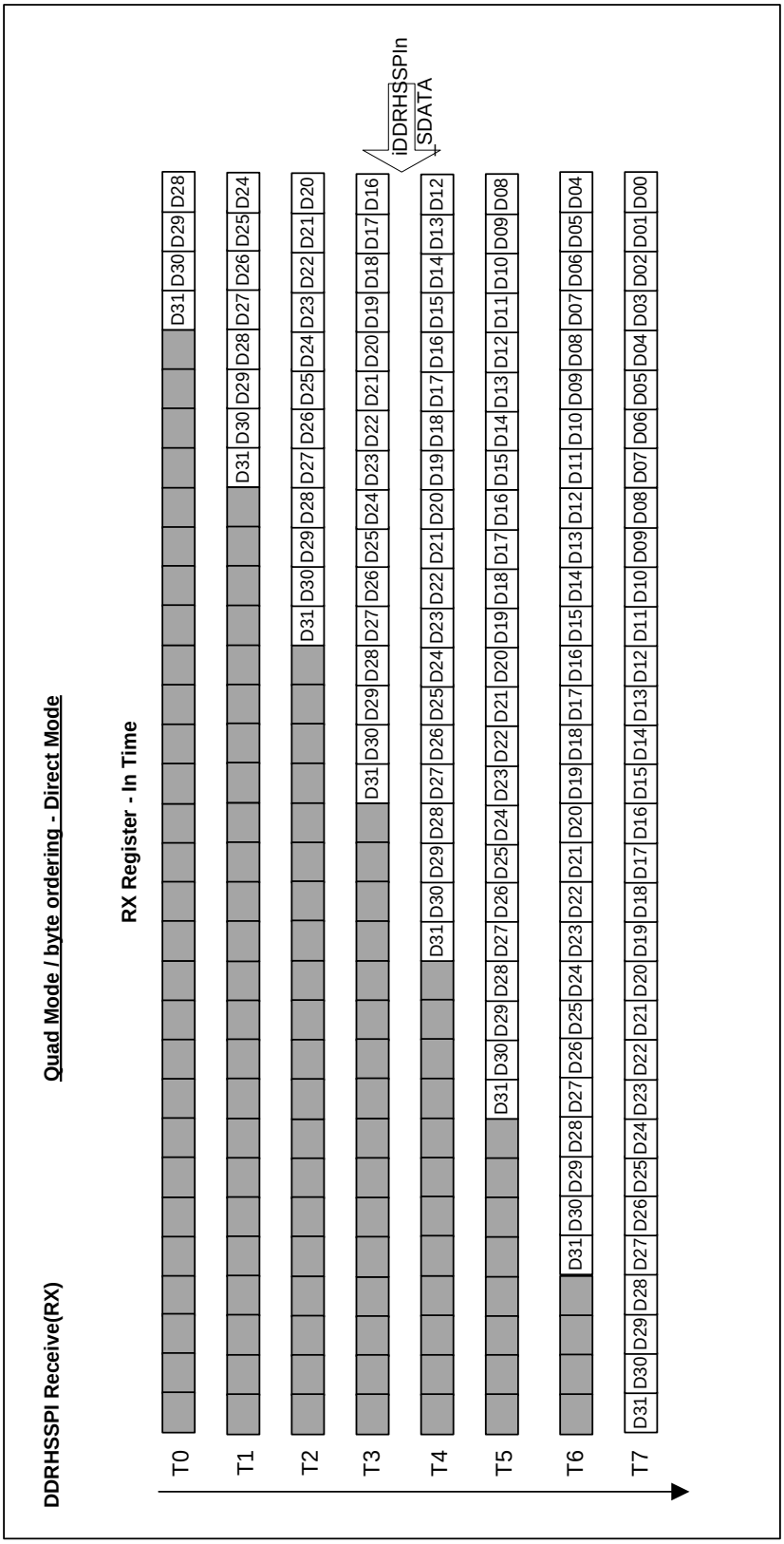


Figure 3-9 Shift Direction (Legacy Mode and Quad Mode in Direct Mode)
(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00")

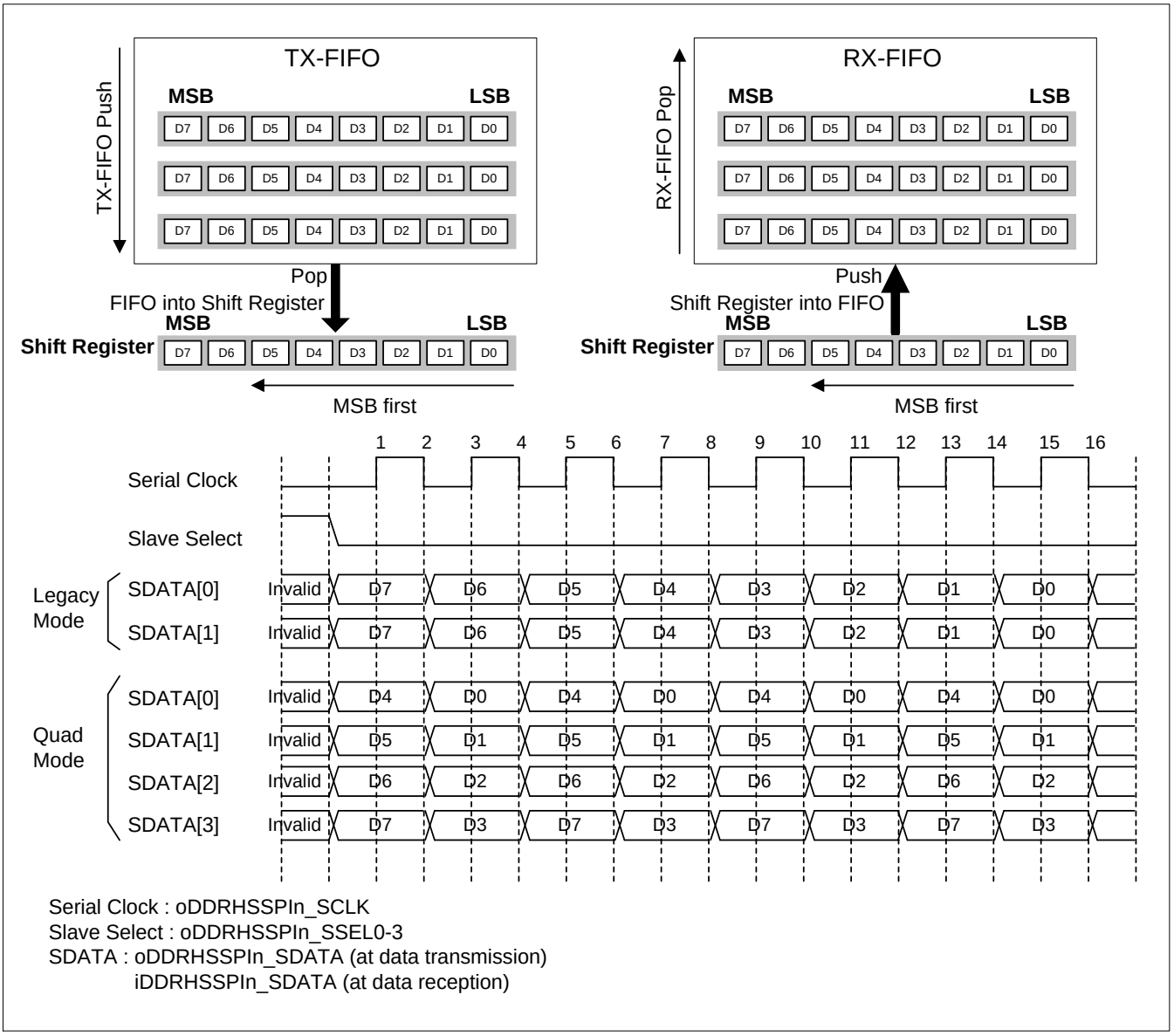
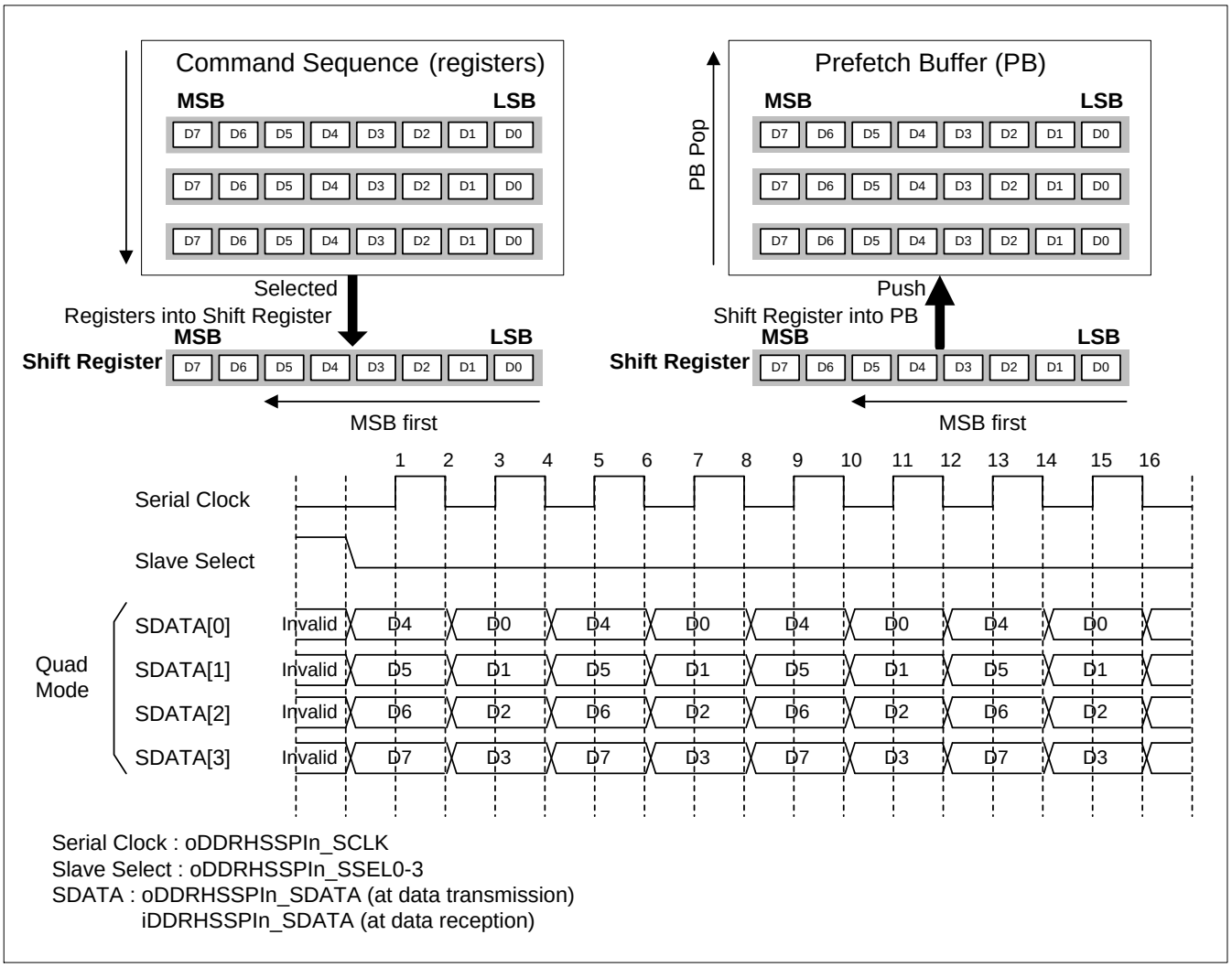
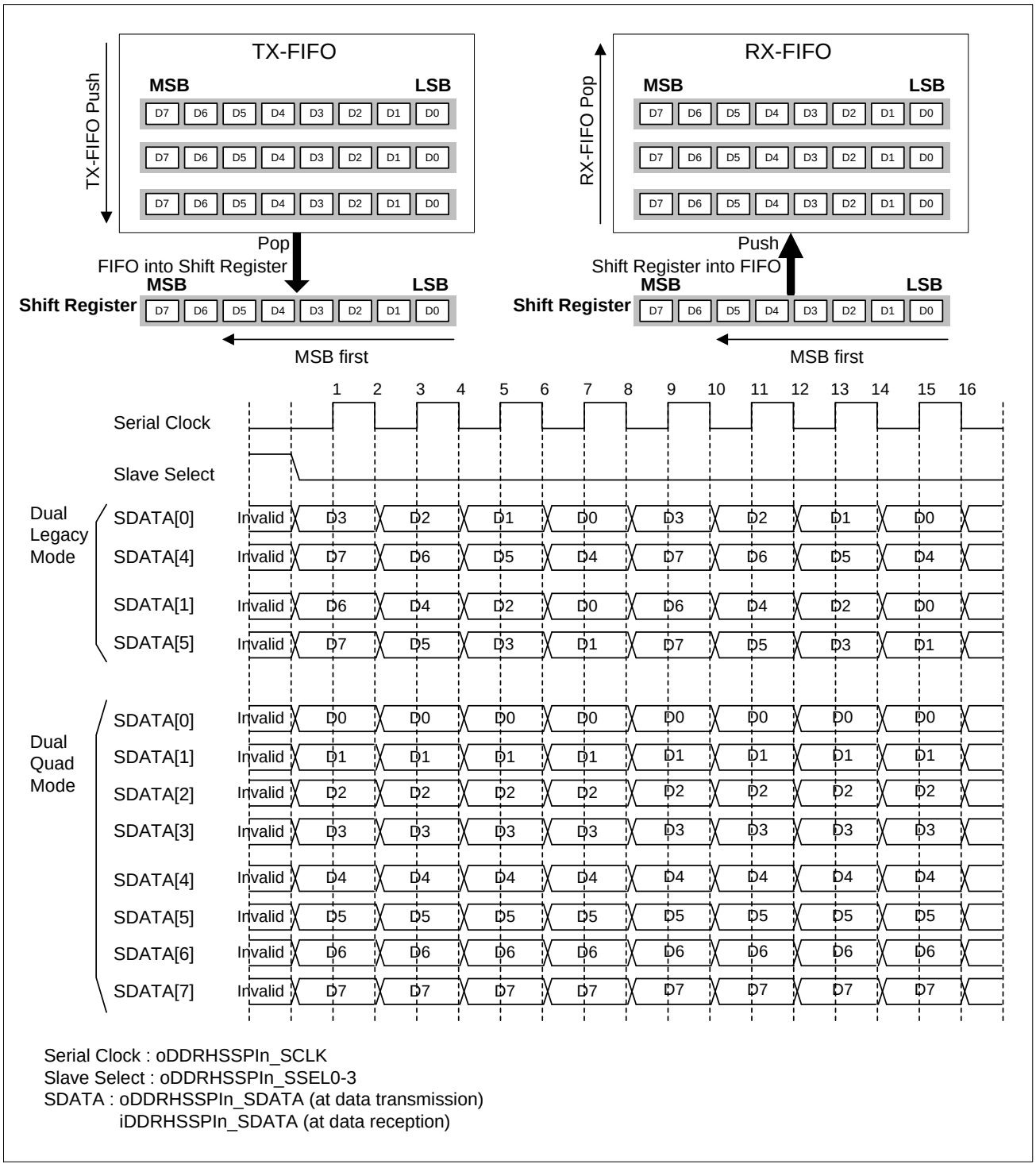


Figure 3-10 Shift Direction (Quad Mode in Command Sequencer Mode)



(Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.)

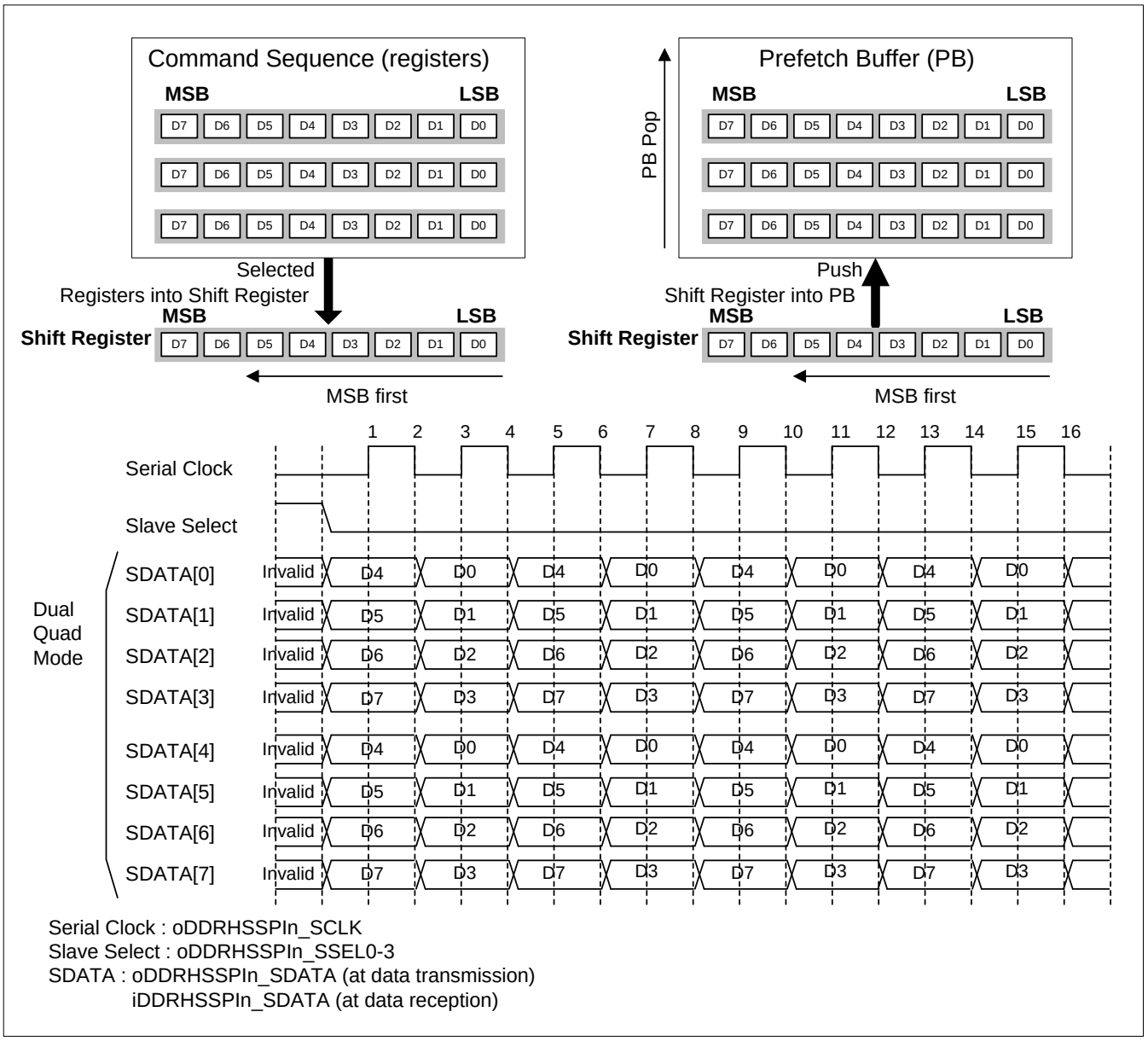
Figure 3-11 Shift Direction (Dual Legacy Mode and Dual Quad Mode in Direct Mode)
(Assumptions: DDRHSSPIn_DMFIFOCFG.FWIDTH="00")



Note:

- *The bit alignment is different between TX and RX in Dual Legacy Mode.*

Figure 3-12 Shift Direction (Dual Quad Mode in Command Sequencer Mode)



(Each line in the PB has 32-bit width, and this diagram is simplified by showing the lower 8 bits.)

4. Operations of the DDRHSSPI

This section describes the operation of DDRHSSPI.

DDRHSSPI can be configured in one of the two operating modes: Direct Mode and Command Sequencer Mode.

In Direct Mode, the MCU can directly write data into the TX-FIFO to be transmitted to the Serial Flash Memory. Similarly, the MCU can directly read the data received from Serial Flash Memory, over the Serial Interface - from the RX-FIFO.
The SPI Core transfers the data to/from the FIFOs over the Serial Interface. The Direct Mode is described in section 4.1 Direct Mode. In Command Sequencer Mode, DDRHSSPI maps the external Serial Flash Memory onto the address space of the MCU.

In Command Sequencer Mode, only Flash Memory reads are supported. If the MCU (or any other master) initiates a System Bus transfer to access any of the mapped Serial Flash Memories, the DDRHSSPI initiates serial transfer for the corresponding memory read operation. Till the time DDRHSSPI accesses the external device, the System Bus transfer is stalled. The Command Sequencer Mode is described in section 4.2 Command Sequencer Mode.

If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, DDRHSSPI can be connected with up to 8 Serial Flash Memories. At the connection with 8 Serial Flash Memories, each of the 4 Slave Select outputs is connected with 2 Serial Flash Memories.

4.1. Direct Mode

In Direct Mode, the MCU (or the DMA controller) is responsible to directly control the serial transfer on the Serial Interface. Direct Mode of transfer can be enabled using the DDRHSSPIn_MCTRL.CSEN bit.

Direct Mode supports the following modes.

- TX-Only Mode

  This is only to transmit data to serial interface, and supports Legacy Mode and Quad Mode. If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, Dual Legacy Mode and Dual Quad Mode are added. This mode supports both SDR and DDR, and is used in the following cases.

  (1)Mode setting of Serial Flash Memory

  (2)Page programming of Serial Flash Memory

  (3)Initialing Memory Read access to Serial Flash Memory

- TX-and-RX Mode

  This is to receive data from serial interface with transmitting data (full-duplex), and supports Legacy Mode and Dual Legacy Mode. The transmission of one TX-FIFO level will involve receiving one RX-FIFO level. This mode does not support DDR, and is used in the following case.

  (1)Status reading from Serial Flash Memory

In Direct Mode, DDRHSSPI uses its internal TX-FIFO and RX-FIFO, for temporary storage of the data to be transmitted and the data received over the Serial Interface.

4.1.1. Internal FIFOs

In Direct Mode, DDRHSSPI internally has two FIFOs for temporary storage: One for the data to be transmitted (TX-FIFO) and one for the data to be received (RX-FIFO).

Based on whether the serial transfers in DDRHSSPI are configured as TX-Only Mode or TX-and-RX Mode in the DDRHSSPIn_DMTRP.TRP, only one or both FIFOs are used by DDRHSSPI. If DDRHSSPI is configured for TX-Only Mode, the TX-FIFO is used. If DDRHSSPI is configured for TX-and-RX Mode, then both TX-FIFO and RX-FIFO are used.

4.1.2. FIFO Size

Each FIFO is 24-locations deep and has a data-width of 32 bits. However, the software can configure the valid data-width of the TX-FIFO and the RX-FIFO in DDRHSSPIn_DMFIFOCFG.FWIDTH.

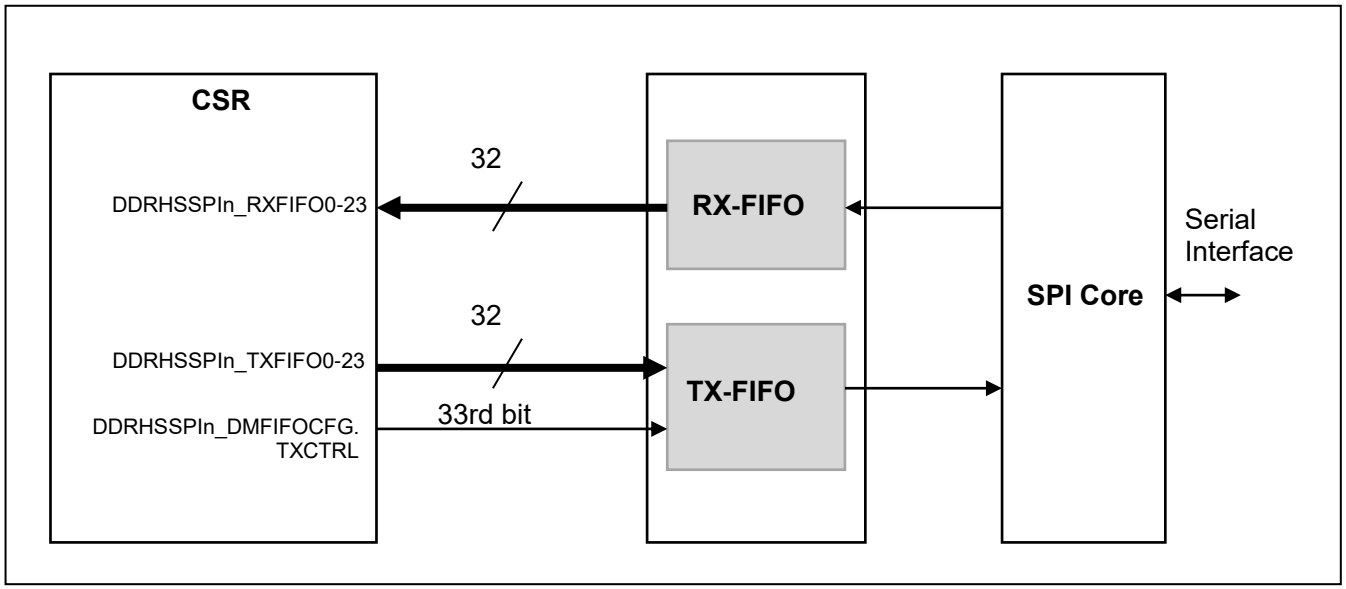
The Shift Register in the SPI Core is 32-bit wide. When the width of the FIFO is changed in the DDRHSSPIn_DMFIFOCFG.FWIDTH, the usable width of the Shift Register also changes accordingly.

Please refer to Figure 4-1, for details.

Note:

- *When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be programmed along the boundary of bytes according to DDRHSSPIn_DMFIFOCFG.FWIDTH, e.g. when FWIDTH is "11", the BCC should be a multiple of 4.*

Figure 4-1 DDRHSSPI FIFO's in Direct Mode



In addition to the 32-bit of data-width, each location in TX-FIFO has a 33rd control bit (known as DDRHSSPIn_DMFIFOCFG.TXCTRL bit), which decides whether

- the data from the TX-FIFO is to be transmitted by the SPI Core, OR
- the serial data lines are to be tri-stated.

4.1.3. TX-FIFO Control

If the DDRHSSPIn_DMFIFOCFG.TXCTRL bit is set to "1", the DDRHSSPI further decodes the bits[12:11] of the data in the corresponding TX-FIFO location. All possible value combinations of the DDRHSSPIn_DMFIFOCFG.TXCTRL bit and the bits[12:11] of TX-FIFO data (DDRHSSPIn_TXFIFO0-23.TXDATA) are shown in Table 4-1.

Table 4-1 Serial Data Output Control

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
0	Don't Care	Don't Care	Serial data output lines are neither tri-stated nor on-the-fly while transmitting the corresponding data. Data Rate Mode (SDR Mode or DDR Mode) and SPI data width (Legacy Mode, Quad Mode or Octal Mode) are as set in DDRHSSPIn_DMTRP Register. Plus, DDRHSSPIn_DMFIFOCFG.FWIDTH affects the valid bit field of TXDATA as below. (1) DDRHSSPIn_DMTRP.TRP[1:0]≠"11" FWIDTH TRP[1:0] SDATA "00" "00" TXDATA[7:0] will be sent on SDATA[0]. "10" TXDATA[7:0] will be sent on SDATA[3:0]. "01" "00" TXDATA[15:8][7:0] will be sent on SDATA[0]. "10" TXDATA[15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) "11" "00" TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[0]. "10" TXDATA[31:24][23:16][15:8][7:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) In TX-and-RX Mode, it supports only Legacy Mode. So, RX-FIFO receives data from SDATA[1]. (2) DDRHSSPIn_DMTRP.TRP[1:0]="11" (Only when DDRHSSPIn_MID.MID is 0x00000100) **(2-1) SDR Mode:** FWIDTH SDATA "00" TXDATA[7:4] will be sent on SDATA[7:4]. TXDATA[3:0] will be sent on SDATA[3:0]. "01" TXDATA[15:12][7:4] will be sent on SDATA[7:4]. TXDATA[11: 8][3: 0] will be sent on SDATA[3:0]. (in the order of TXDATA[15:8]->[7:0], along the SCLK edge) "11" TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) (Continued)
0	Don't Care	Don't Care	**(2-2) DDR Mode:** FWIDTH SDATA "00" Not allowed. "01" TXDATA[31:28][23:20] will be sent on SDATA[7:4]. TXDATA[27:24][19:16] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16], along the SCLK edge) "11" TXDATA[31:28][23:20][15:12][7:4] will be sent on SDATA[7:4]. TXDATA[27:24][19:16][11: 8][3:0] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16]->[15:8]->[7:0] along the SCLK edge) When in TXCTRL="0", DDRHSSPI cannot transfer data at Dual Legacy Mode, since the SPI protocol is controlled just by DDRHSSPIn_DMTRP.TRP. Therefore, TX-and-RX Mode is not available for Dual Legacy Mode under this condition. **Common in (1) and (2):** The actual valid byte lane of TXDATA varies according to the lower address bits for TX-FIFO, e.g. if FWIDTH="00" AND HSIZE=0x0 AND HADDR[1:0]="01", TXDATA[15:8] is valid instead of TXDATA[7:0].

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
1	0	0	Serial data output lines are tri-stated for defined number of SCLK cycles . The number of dummy cycles is defined in TXDATA[7:4] bits in following way: "0000" : Serial data output lines are tri-stated for 1 SCLK cycle "0001" : Serial data output lines are tri-stated for 2 SCLK cycles .... "1111" : Serial data output lines are tri-stated for 16 SCLK cycles Please note that maximum number of dummy cycles programmed per one TX-FIFO entry, is limited to be equivalent to sending one TX-FIFO level (with defined DDRHSSPIn_DMFIFOCFG.FWIDTH). The length of dummy cycles is limited as below. [Max Dummy Cycles] = [Bit number of 1 FIFO entry] / [1 Byte Time (cycle number)] For example, under the following conditions: − Quad Mode (by DDRHSSPIn_DMTRP.TRP[1:0]) − SDR data rate − number of dummy cycles is 4 − DDRHSSPIn_DMFIFOCFG.FWIDTH is "01" (16 bit-wide). In this example, the number of dummy cycle is within the limit. If the number of dummy cycles is larger than 4 in this example, the output serial data will be tri-stated for only 4 SCLK cycles and the rest will be ignored. In TX-and-RX Mode, data on the input port iDDRHSSPIn_SDATA is sampled along the output of oDDRHSSPIn_SDATA. The number of received byte data is calculated as below. [Rx Byte Count] = [Dummy Cycles] / [1 Byte Time (cycle number)] The number of dummy cycles shall be a multiple of 1 Byte Time. RX-FIFO is being filled also during the dummy cycles in TX-and-RX Mode, so the software needs to read RX-FIFO to remove these received data.

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
1	1	Don't Care	**SDR Mode** In SDR Mode (DDRHSSPIn_DMTRP.DDRM = "0"), TXDATA[10] is ignored, and TXDATA[7:0] will be sent in SDR Mode, depending on TXDATA[9:8]: "00": Legacy Mode − If DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300, and DDRHSSPIn_DMTRP. TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]= "11"), this works as Dual Legacy Mode, and a byte data will be sent as below (4 SCLK cycles): − TXDATA[7:4] will be sent on SDATA[4]. − TXDATA[3:0] will be sent on SDATA[0]. − On the RX side of TX-and-RX Mode, a byte data will be received as below (4 SCLK cycles): − RXDATA[7:6] is received from {SDATA[5], SDATA[1]}. (1st cycle) − RXDATA[5:4] is received from {SDATA[5], SDATA[1]}. (2nd cycle) − RXDATA[3:2] is received from {SDATA[5], SDATA[1]}. (3rd cycle) − RXDATA[1:0] is received from {SDATA[5], SDATA[1]}. (4th cycle) − The actual valid byte lane of RXDATA varies according to the lower address bits for RX-FIFO, − e.g. if FWIDTH="00" AND HSIZE=0x0 AND HADDR[1:0]="01", RXDATA[15:8] is valid instead of RXDATA[7:0]. − − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Legacy Mode (8 SCLK cycles) on SDATA[0]. − On the RX side of TX-and-RX Mode, a byte data will be received from SDATA[1] (8 SCLK cycles). "01": Not allowed. "10": Quad Mode − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Quad Mode (2 SCLK cycles) on SDATA[3:0]. − If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "10". "11": Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) − If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), this works as Dual Quad Mode, and a byte data will be sent as below (1 SCLK cycles): − TXDATA[7:4] will be sent on SDATA[7:4]. − TXDATA[3:0] will be sent on SDATA[3:0]. − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then it is not allowed to set TXDATA[9:8] to "11". (Continued)

DDRHSSPIn_DMFIFOCFG.TXCTRL	Bit 12 of TX-FIFO Data (TXDATA[12])	Bit 11 of TX-FIFO Data (TXDATA[11])	Description
1	1	Don't Care	**DDR Mode** In DDR Mode (DDRHSSPIn_DMTRP.DDRM = "1"), then the behavior depends on TXDATA[10]: (1)SDR on-the-fly If TXDATA[10] = "0", then TXDATA[7:0] will be sent same as SDR Mode above. (2)DDR on-the-fly If TXDATA[10] = "1", then the behavior depends on bits TXDATA[9:8]: "00": Legacy Mode − If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "00". − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Legacy Mode (4 SCLK cycles) on SDATA[0]. "01": Not allowed. "10": Quad Mode − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then TXDATA[7:0] will be sent in Quad Mode (1 SCLK cycles) on SDATA[3:0]. − If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), then it is not allowed to set TXDATA[9:8] to "10". "11": Dual Quad Mode (Only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300) − If DDRHSSPIn_DMTRP.TRP is configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]="11"), this works as Dual Quad Mode, and two bytes of data will be sent as below (1 SCLK cycles): − TXDATA[31:28][23:20] will be sent on SDATA[7:4]. − TXDATA[27:24][19:16] will be sent on SDATA[3:0]. (in the order of TXDATA[31:24]->[23:16] along the SCLK edge) − If DDRHSSPIn_DMTRP.TRP is not configured to Octal Mode (DDRHSSPIn_DMTRP.TRP[1:0]≠"11"), then it is not allowed to set TXDATA[9:8] to "11".

4.1.4. Available Command Formats of Serial Flash Memory

In order to apply to the command format of Serial Flash Memory, DDRHSSPI has the following features.

- Dynamic control of SPI Protocol
- Dynamic control of Data Rate Mode (SDR Mode or DDR Mode)
- Inserting dummy cycles

To realize these features, DDRHSSPI uses partial bits of TXDATA to control them. When DDRHSSPIn_DMFIFOCFG.TXCTRL bit is "1", these features are available by decoding some bits of TXDATA. By this control, DDRHSSPI is capable of Serial Flash command formats shown in Table 4-2 and Table 4-3.

Note:

- *When in writing TXDATA[12:8] for SPI control on-the-fly, it is allowing regardless the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH.*

The available command formats of Serial Flash are different according to the value of DDRHSSPIn_MID. MID.

(1)SDR (DDRHSSPIn_DMTRP.DDRM="0")

Table 4-2 The available Serial Flash Command Format (SDR)

DDRHSSPIn_MID		Command	Address	Mode (*1)	Dummy
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	Legacy (1byte)	none	none	none
Available	Available	Legacy (1byte)	Legacy (3 / 4bytes)	none / 1byte	none / greater than 0
Available	Available	Legacy (1byte)	Quad (3 / 4bytes)	none / 1byte	none / greater than 0
N/A	Available	Dual Legacy (1byte each)	none	none	none
N/A	Available	Dual Legacy (1byte each)	Dual Legacy (3 / 4bytes each)	none / 1byte each	none
N/A	Available	Dual Legacy (1byte each)	Dual Quad (3 / 4bytes each)	none / 1byte each	none / greater than 0

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(2)DDR (DDRHSSPIn_DMTRP.DDRM="1")

Table 4-3 The available Serial Flash Command Format (DDR)

DDRHSSPIn_MID		Command	Address	Mode (*1)	Dummy
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	Legacy (1byte), SDR	Quad (3 / 4bytes), DDR	1byte, DDR	greater than 0
N/A	Available	Dual Legacy (1byte each), SDR	Dual Quad (3 / 4bytes each), DDR	1byte each, DDR	greater than 0

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

For above Serial Flash command formats, DDRHSSPI supports the following SPI protocols on-the-fly.

Table 4-4 The Available Switching SPI Protocols On-the-fly

DDRHSSPIn_MID		DDRHSSPIn_ DMTRP. TRP[1:0]	TXDATA[9:8] *1	Rx	Tx
0x0000 0001	0x0000 0100 or 0x00000 300				
Available	Available	"00" (Legacy)	"00" (Legacy)	SDATA[1]	SDATA[0]
N/A	N/A	"00" (Legacy)	"10" (Quad)	Not allowed	Not allowed
Available	Available	"10" (Quad)	"00" (Legacy)	SDATA[1]	SDATA[0]
Available	Available	"10" (Quad)	"10" (Quad)	Not allowed	SDATA[3:0]
N/A	Available	"11" (Dual Quad)	"00" (Dual Legacy)	SDATA[1], SDATA[5]	SDATA[0], SDATA[4]
N/A	Available	"11" (Dual Quad)	"10" (Quad)	Not allowed	Not allowed
N/A	Available	"11" (Dual Quad)	"11" (Dual Quad)	Not allowed	SDATA[7:0]

*1 With conditions of DDRHSSPIn_DMFIFOCFG.TXCTRL="1" and TXDATA[12]="1"

4.1.5. FIFO Accesses in Direct Mode

Depending on the configured width of the FIFOs, 8-bit, 16-bit or 32-bit accesses are allowed to the DDRHSSPIn_RXFIFO0-23 and DDRHSSPIn_TXFIFO0-23 Registers.

If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "11" (32-bit wide mode), then only 32-bit access to the FIFO is allowed.

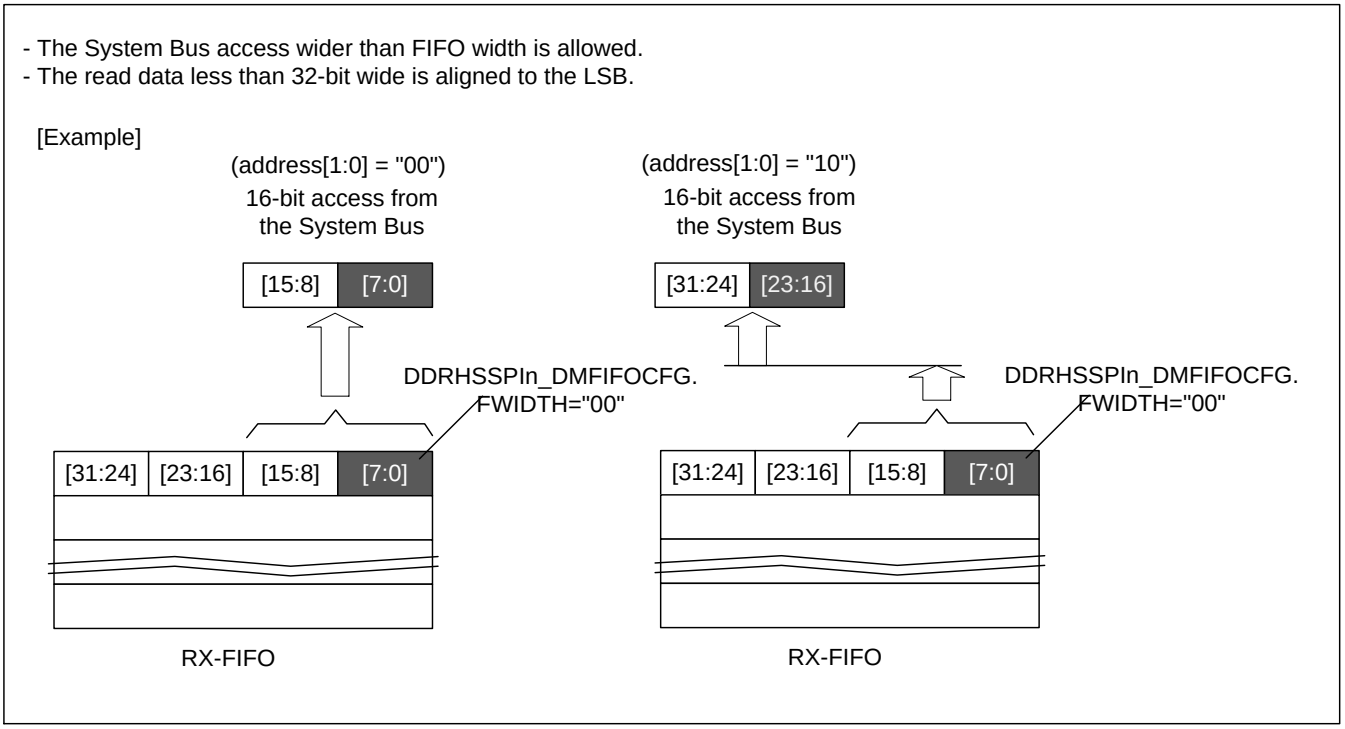
If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "01" (16-bit wide mode), then 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry.

If DDRHSSPIn_DMFIFOCFG.FWIDTH is set to "00" (8-bit wide mode), then 8-bit, 16-bit and 32-bit accesses to the FIFO are allowed. A 32-bit access covers a whole word line for one FIFO entry. A 16-bit access with any alignment covers lower 16 bits of one FIFO entry. An 8-bit access with any alignment covers the lower 8 bits of one FIFO entry.

When in writing data into TX-FIFO (to any of DDRHSSPIn_TXFIFO0-23), the written data go along with DDRHSSPIn_DMFIFOCFG.TXCTRL bit, and they are written into TX-FIFO together. And when the data width is not a word size (32 bits), the data is aligned to the LSB, before written into TX-FIFO.

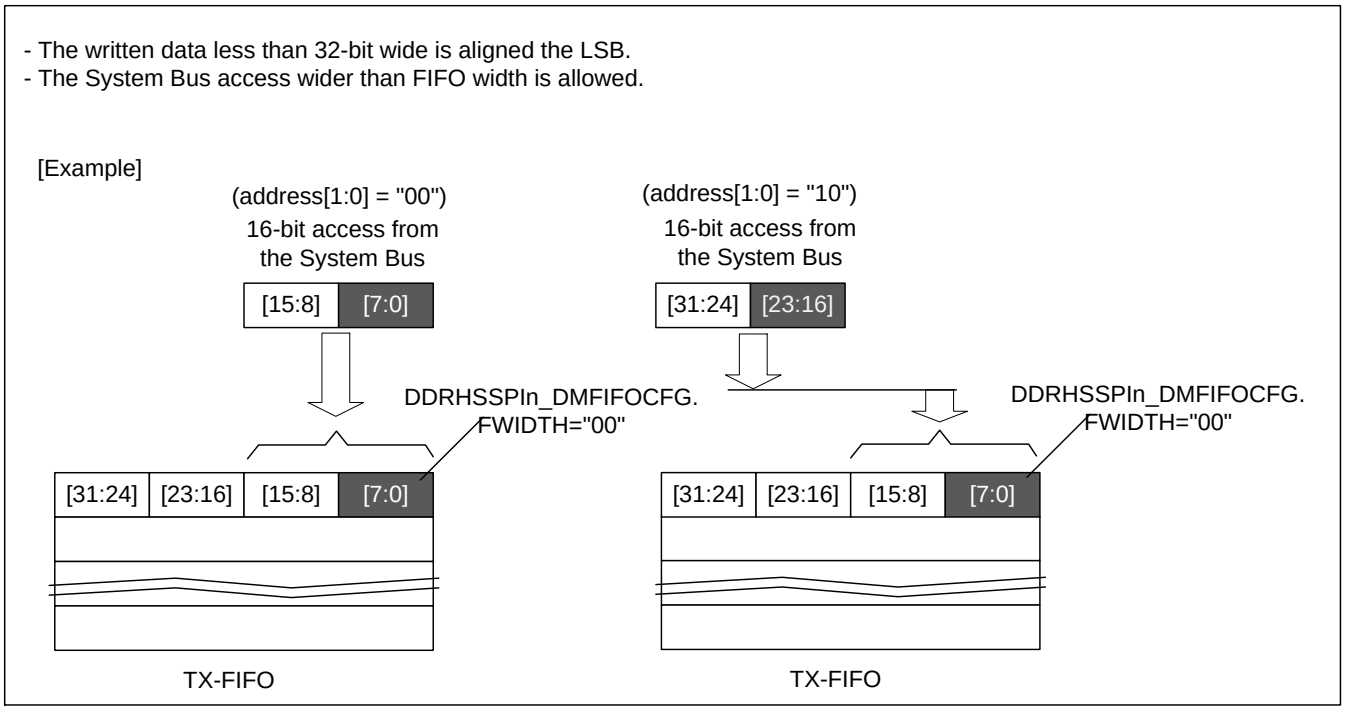
When the System Bus access is wider than RX-FIFO width (configured by DDRHSSPIn_DMFIFOCFG. FWIDTH), the width of read data follows FWIDTH value, and the read data is aligned to the LSB. Figure 4-2 shows this alignment and a notice of the System Bus address.

Figure 4-2 RX-FIFO Access at Greater Width



When the System Bus access is wider than TX-FIFO width (at DDRHSSPIn_DMFIFOCFG.TXCTRL="0"), it is allowed as well. In this case, the valid byte lane follows the System Bus address, as shown in Figure 4-3.

Figure 4-3 TX-FIFO Access at Greater Width (DDRHSSPIn_DMFIFOCFG.TXCTRL="0")



When in DDRHSSPIn_DMFIFOCFG.TXCTRL="1", the setting of DDRHSSPIn_DMFIFOCFG.FWIDTH is ignored at writing into TX-FIFO, and up to 32 bits could be relevant. (please refer to the Table 4-1).

Note:

- *In case that DDRHSSPI is configured in:*
  - DDR Mode, AND
  - Dual Quad Mode

  setting DDRHSSPIn_DMFIFOCFG.FWIDTH to "00" (8-bit wide mode) is not allowed.

4.1.6. Service Requests

When operating in Direct Mode, Interrupt Service Requests to the MCU are triggered based on the current fill-levels of the TX-FIFO and the RX-FIFO, and their configured threshold values. Alternatively, the external DMA controller can be used for data transfers. When operating in Direct Mode, the DDRHSSPI has an interface with the DMA controller in the MCU for block transfers of data to/from its TX-FIFO and the RX-FIFO.

Interrupt flags are set when the current SPI transfer finishes.

Assertion of Interrupt Service Requests Based on FIFO Levels

The fill levels of both FIFOs are accessible to the system through the DDRHSSPIn_DMFIFOSTATUS. TXFLEVEL and the DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL. The Interrupt Service Requests are generated by the DDRHSSPI based on (1)the FIFO fill levels and (2)their threshold values configured by the MCU.

The DDRHSSPIn_TXF.TFLETS is set if the DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH.

The DDRHSSPIn_RXF.RFMTS is set if DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.RXFTH.

If the DDRHSSPI is configured for TX-Only operation, the RX-FIFO is not used.

Assertion of DMA Service Requests Based on FIFO Levels

The DDRHSSPI supports block transfer mechanism in the DMA controller. The DMA Service Requests are generated by the DDRHSSPI based on (1)the FIFO fill levels and (2)their threshold values configured by the MCU. To keep track of the number of successful data transfers to/from the TX-FIFO and/or the RX-FIFO, the DDRHSSPI internally uses two down-counters: the DDRHSSPI RX Block Counter and the DDRHSSPI TX Block Counter. Each of these counters is a 5 bits down counter, which is reloaded with the DMA Block size (for the respective channel) whenever the DMA Service Request for that channel is asserted. The counters are decremented with every successful read or write accesses to the RX-FIFO or TX-FIFO. In case of the RX-FIFO accesses, the RX Block Counter is decremented only if the access was from a System Bus master other than the DAP Controller. The block counters do not underflow (i.e. the counter value remains 0 even if it is decremented while it is already 0).

DMA Write Channel has DDRHSSPIn_FAULTF.DWCBSFS bit, and DMA Read Channel has DDRHSSPIn_FAULTF.DRCBSFS bit. A DMA Block Size Fault is triggered if all of the following conditions are satisfied:

- The DMA Block Counter is decremented (due to a valid System Bus access) while it is already 0, AND
- DDRHSSPIn_DMAEN.RXDMAEN or DDRHSSPIn_DMAEN.TXDMAEN for the corresponding DMA Channel is set to "1", AND
- DDRHSSPIn_MCTRL.MES="1", AND
- DDRHSSPIn_MCTRL.CSEN="0"

The DMA Read Channel must be setup to perform a block transfer of "DDRHSSPIn_DMFIFOCFG.RXFTH + 1" transfers. The DMA Write Channel must be setup to perform a block transfer of "24 - DDRHSSPIn_DMFIFOCFG.TXFTH" transfers. These values are reloaded into the DDRHSSPI's internal Block Counters, whenever the DMA Service Request is asserted.

The DMA Block Counter is cleared to "0" in either of the following conditions:

- The corresponding DMA channel is disabled (in DDRHSSPIn_DMAEN Register), OR
- DDRHSSPIn_MCTRL.MES="0", OR
- The mode of operation is switched from Direct Mode to Command Sequencer Mode.

The RX DMA Service Request (for the DMA Read Channel) is asserted if all of the following conditions are satisfied:

- DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is more than DDRHSSPIn_DMFIFOCFG.RXFTH. This condition is the same as the DDRHSSPIn_RXF.RFMTS="1", AND
- The DDRHSSPI RX Block Counter value is 0, AND
- The DMA Read Channel acknowledgement signal is deasserted by the DMA controller, AND
- A previous DMA Read Service Request is not pending, AND
- DDRHSSPIn_DMAEN.RXDMAEN="1", AND
- DDRHSSPIn_FAULTF.DRCBSFS="0", AND
- DDRHSSPIn_MCTRL.MES="1", AND
- DDRHSSPIn_MCTRL.CSEN="0", AND
- DDRHSSPIn_DMTRP.TRP[3:2]="00"

The RX DMA Service Request (for DMA Read Channel) is deasserted if any of the following conditions is satisfied:

- The DMA Read Channel Service Request has been acknowledged by the DMA controller, OR
- DDRHSSPIn_DMAEN.RXDMAEN="0", OR
- DDRHSSPIn_MCTRL.MES="0", OR
- The mode of operation is switched from Direct Mode to Command Sequencer Mode

The TX DMA Service Request (for DMA Write Channel) is asserted if all of the following conditions are satisfied:

- DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH. This condition is the same as the DDRHSSPIn_TXF.TFLETS="1", AND
- The DDRHSSPI TX Block Counter value is 0, AND
- The DMA Write Channel acknowledgement signal is deasserted by the DMA controller, AND
- The Previous DMA Write Service Request is not pending, AND
- DDRHSSPIn_DMAEN.TXDMAEN="1", AND
- DDRHSSPIn_FAULTF.DWCBSFS="0", AND
- DDRHSSPIn_MCTRL.MES="1", AND
- DDRHSSPIn_MCTRL.CSEN="0", AND
- DDRHSSPIn_DMTRP.TRP[2]="0"

The TX DMA Service Request (for DMA Write Channel) is deasserted if any of the following conditions is satisfied:

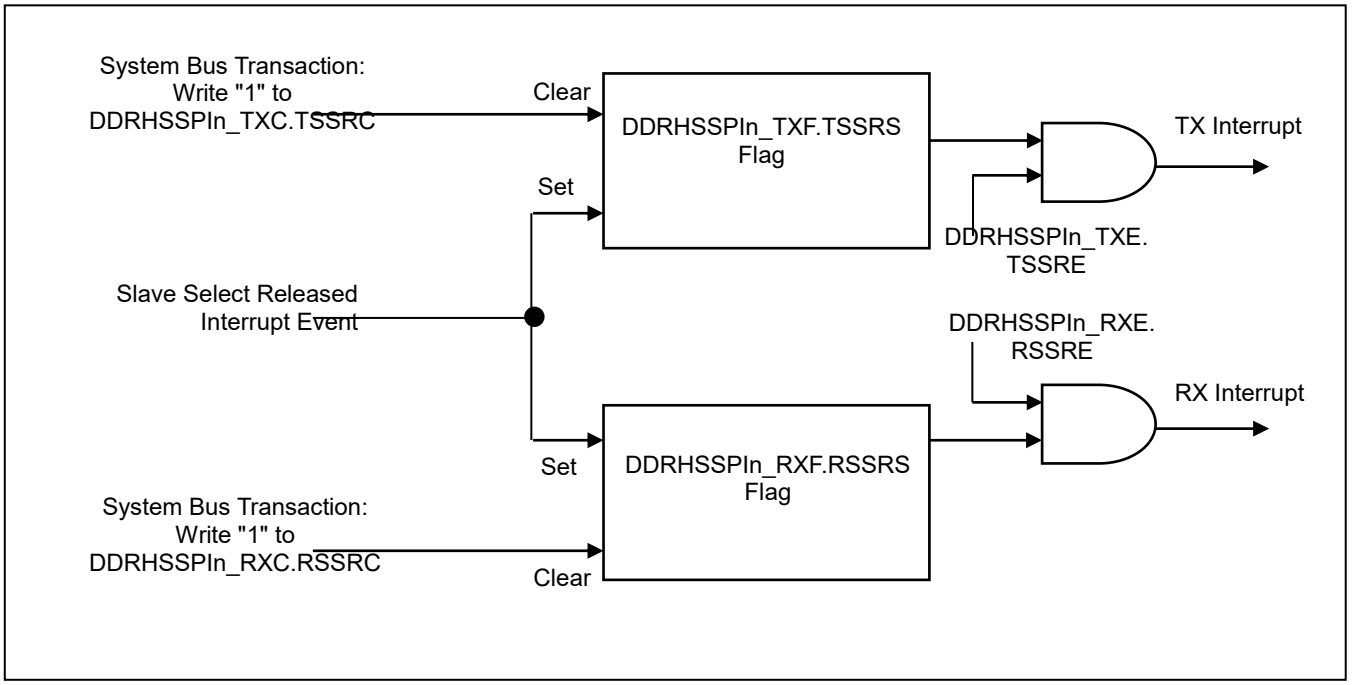
- The DMA Write Channel Service Request has been acknowledged by the DMA controller, OR
- DDRHSSPIn_DMAEN.TXDMAEN="0", OR
- DDRHSSPIn_MCTRL.MES="0", OR
- The mode of operation is switched from Direct Mode to Command Sequencer Mode

It is highly recommended that either value of DDRHSSPIn_DMFIFOCFG.TXFTH or DDRHSSPIn_DMFIFOCFG.RXFTH should not exceed 23.

Assertion of Service Requests on End of Transfer

While operating in Direct Mode, the DDRHSSPI also triggers interrupts when the Slave Select line is deasserted. The Slave Select Deassertion event is routed onto two Interrupt Flags: DDRHSSPIn_TXF.TSSRS and DDRHSSPIn_RXF.RSSRS, which have separate Interrupt-Clear and Interrupt-Enable bits. The interrupt flags are routed onto separate Interrupt Signals. This logic is indicated in Figure 4-4.

Figure 4-4 Routing of the "Slave Select Released" Interrupt Event



4.1.7. SPI Transfers

The DDRHSSPI can initiate the transfers onto one of the four Slave Select lines, selected by the DDRHSSPIn_DMPSEL.PSEL. The following are major steps and other relevant information for initiating SPI transfers:

■ Communication Attributes of the DDRHSSPI

Communication over the Serial Interface has several attributes, like: Frequency of the Serial Interface Clock, etc. These communication attributes should be same among Serial Flash Memories. When the DDRHSSPI is working in Direct Mode, it can be interfaced with up to 4 Serial Flash Memories.

These device-specific communication attributes can be configured in the DDRHSSPIn_PCC0-3 Registers in the CSR.

■ Configuration parameter change

It is possible to switch between different bit widths and between SDR Mode and DDR Mode on-the-fly, without releasing the Slave Select line. However, it is up to the SW to take care of the timing of the switching from different SPI widths or clock modes so that data in TX-FIFO or RX-FIFO does not get lost or corrupted because of the switch.

The ability to change clock mode and access width on-the-fly is implemented in such way that the DDRHSSPI can support various Serial Flash Memories access protocols (such as command and address).

■ Initiating the Serial Transfers:

When DDRHSSPIn_MCTRL.MES="1" and DDRHSSPIn_MCTRL.CSEN="0", serial transfers are initiated by DDRHSSPI when the DDRHSSPIn_DMSTART.START bit is set to "1".

If the DDRHSSPIn_DMTRP.TRP is programmed such that transmission is enabled, and if the TX-FIFO is empty when the DDRHSSPIn_DMSTART.START bit is set to "1"; then the DDRHSSPI delays the initiation of the serial transfer until the TX-FIFO is written by the software.

If Byte Counter Mode is used for controlling the transfer length, then the DDRHSSPIn_DMBCS Register will be loaded with the value in DDRHSSPIn_DMBCC Register immediately when DDRHSSPIn_DMBCC is written.

Once the DDRHSSPIn_DMSTART.START bit is set to "1", it cannot be reset by the software. The DDRHSSPI resets the bit after it has finished the Serial Transfer.

Writing a "1" to the DDRHSSPIn_DMSTART.START bit while SSEL is asserted has no effect.

■ Halting a transfer due to lack of TX-DATA or due to lack of RX-FIFO space:

In TX-Only Mode, an ongoing transfer can be halted by keeping the Slave Select asserted and by halting the Serial Clock. The DDRHSSPI automatically halts the Serial Clock, while it is waiting for the TX-FIFO to be written. The condition to halt a serial transfer in TX-Only Mode is as below.

- the TX-FIFO and the TX Shift Register are empty

And, when above condition is not satisfied while in halting, DDRHSSPI gets out of the halting status and resume the serial transfer (with starting the toggling of the Serial Clock).

In TX-and-RX Mode, it is necessary to avoid the halting of serial transfer. The condition to halt a serial transfer is as below.

- the TX-FIFO and the TX Shift Register are empty, OR
- the RX-FIFO and the RX Shift Register are full

■ Controlling the transfer length:

The transfer length is controlled as follows.

The MCU is supposed to initialize the DDRHSSPIn_DMBCC.BCC with the number of bytes to be transferred over the Serial Interface, during the Slave Select output is asserted. When the DDRHSSPI transfers are initiated, the DDRHSSPI counts the number of bytes that are transferred and releases the Slave Select line after the number of bytes indicated in DDRHSSPIn_DMBCC.BCC have been transferred.

When the value of DDRHSSPIn_DMFIFOCFG.FWIDTH is greater than 0, the value of DDRHSSPIn_DMBCC.BCC should be the multiple of "DDRHSSPIn_DMFIFOCFG.FWIDTH + 1".

In TX-Only Mode, and SW sets number of dummy cycles more than 1 FIFO entry, the actual length of dummy cycles on the Serial Interface become same as 1 FIFO entry (the excessive number is ignored).

In TX-and-RX Mode, when transmitted data is for dummy cycles by DDRHSSPIn_DMFIFOCFG.TXCTRL="1", the received data is input according to the dummy length. Please refer to Table 4-1 for details. The length of dummy cycles shall not be more than 1 FIFO entry.

Note:

- *When in TX-and-RX Mode, please avoid halting the serial transfer.*

4.2. Command Sequencer Mode

In Command Sequencer Mode, the DDRHSSPI acts as an SPI master for interfacing with the external Serial Flash Memories. Each of the 4 Slave Select lines can be used for mapping uniform type of Serial Flash Memories such as Cypress FL-S series. Memory accesses initiated by the MCU and the other System Bus masters on the System Bus, are automatically converted to the Serial Flash Memory read commands by the DDRHSSPI.

This section describes the Command Sequencer Mode of the DDRHSSPI.

Notes:

- *To use the Command Sequencer Mode, please make sure that the "Serial Flash" has a "Continuous Read Command Sequence", that omits the instruction bit field and starts with address field from second SPI transaction. DDRHSSPI needs to work on the first SPI transaction at the Direct Mode (TX-Only Mode) to control the variable bit width on the SPI, and this feature covers the first SPI transaction. Then, the Command Sequencer Mode can be used from the second SPI transaction that continues from the first one. Since the first SPI transaction is controlled under TX-Only Mode, the read data from the Serial Flash are ignored. Valid reading of Serial Flash is run at the Command Sequencer Mode.*
- *In memory reading, the bit width on the SPI from the address field should be constantly Quad.*

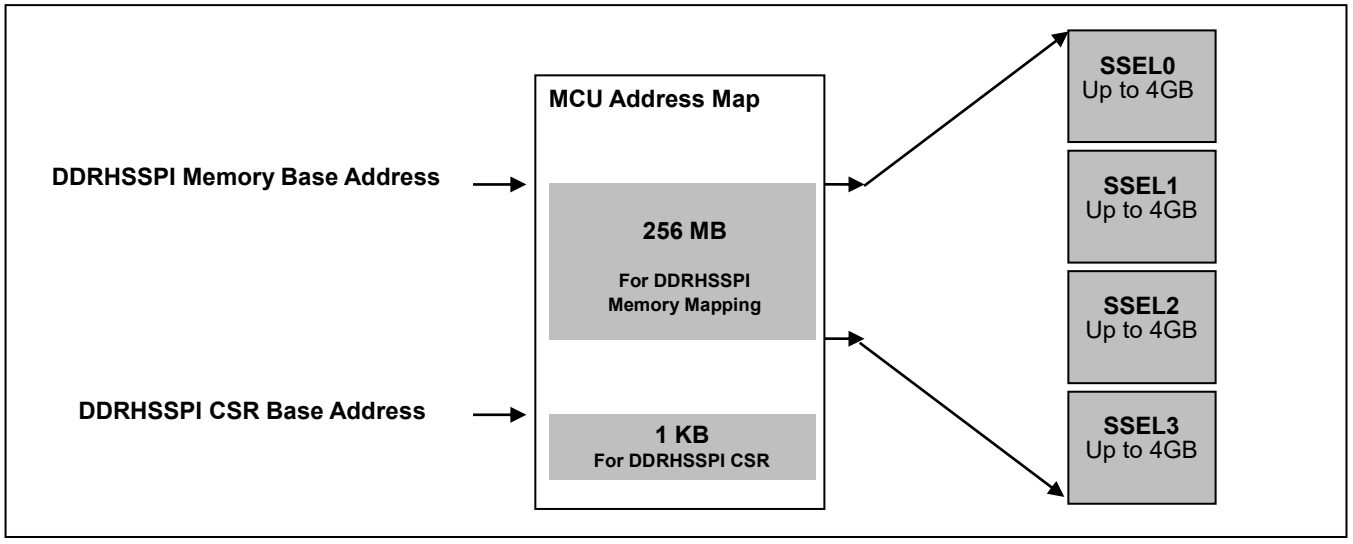
Memory Mapping

The Command Sequencer Mode can be used for memory mapping of up to 4 Slave Selects. All mapped Serial Flash Memories shall be of the same family and with the same internal configuration.

In Command Sequencer Mode, the DDRHSSPI allocates a memory space of 256MB, for mapping up to 4 external Serial Flash Memories. By using the Address Extension mechanism in Command Sequencer, each Slave Select can address a memory of up to 4GB (i.e. 32-bit address bus). The Address Extension mechanism allows concatenation of the most significant bits from a 19-bit Address Extension Register (i.e. the DDRHSSPIn_CSAEXT Register) with the few bits from the System Bus address, to form a 32-bit address to be accessed on each Slave Select. This feature is explained in detail in the subsequent sub-sections of this chapter.

Thus, the 256MB of the MCU address-space is virtually mapped to 16GB of external Serial Flash Memories, as shown in Figure 4-5.

Figure 4-5 Mapping of Serial Flash Memories on the Slave Select lines



Selection of Slaves

The DDRHSSPIn_CSCFG.MSEL indicates the size of the System Bus address space associated with each Slave Select line.

Based on the value of the DDRHSSPIn_CSCFG.MSEL and the address placed by the MCU (or the other System Bus Master, like the DMA Controller) on the Address Bus, the DDRHSSPI Command Sequencer decides which of the 4 Slave Select lines is to be asserted. Please refer to Table 4-5 for details.

As an example, in case DDRHSSPIn_CSCFG.MSEL indicates that the System Bus address space associated with each Slave Select is of 8KB. If the System Bus address is between "the DDRHSSPI Memory Base Address" and "the DDRHSSPI Memory Base Address + 8KB", then the Slave Select 0 is asserted. If the System Bus address is between "the DDRHSSPI Memory Base Address + 8KB" and "the DDRHSSPI Memory Base Address + 16KB", then the Slave Select 1 is asserted, and so on. If the System Bus Address is beyond "the DDRHSSPI Memory Base Address + 32KB", then the address is out of range and the DDRHSSPIn_FAULTF.UMAFS flag is set.

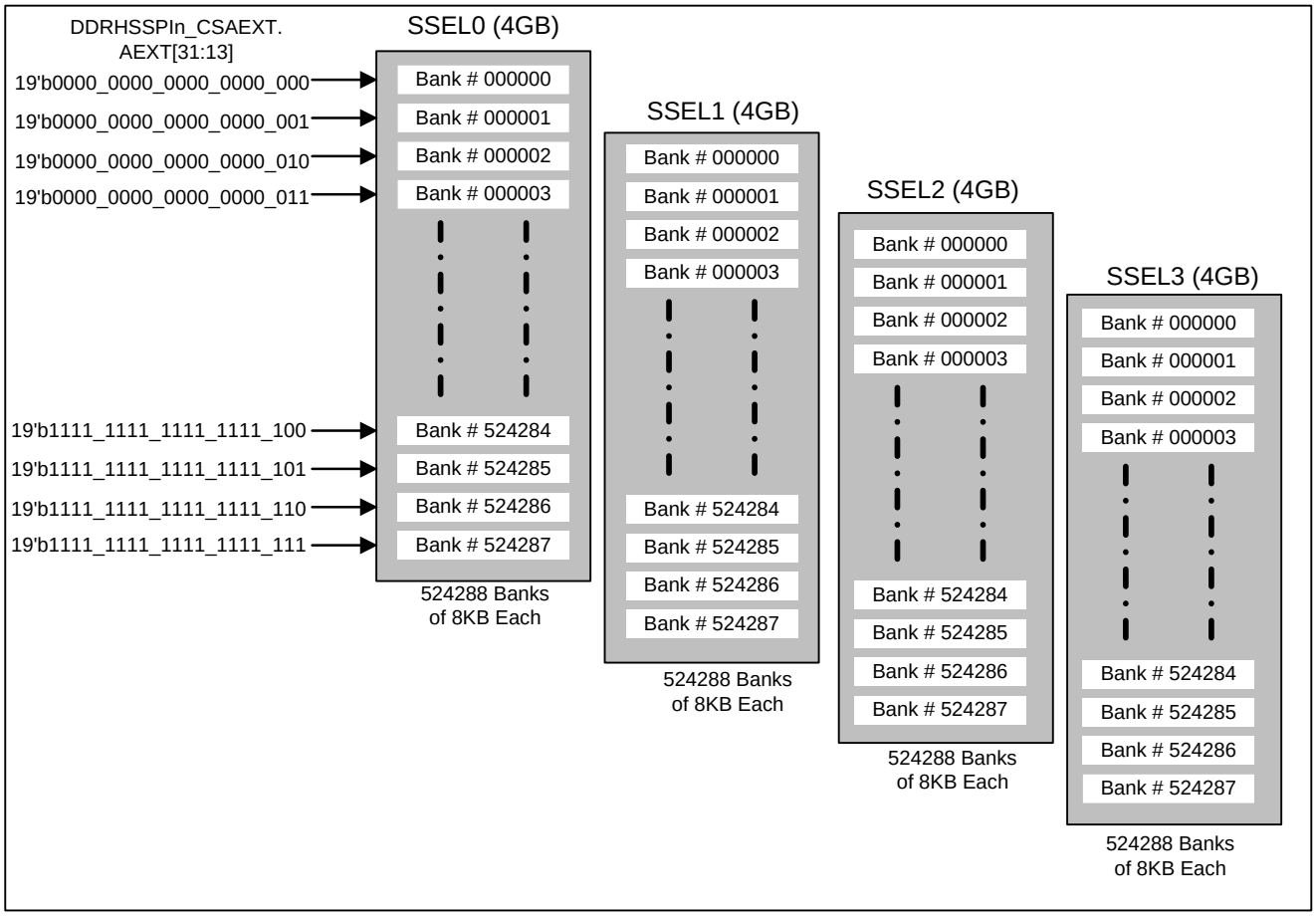
Even if the address of the System Bus access is in the valid range, if the target Slave Select is not set valid, the DDRHSSPIn_FAULTF.UMAFS flag is set as well. For example, if DDRHSSPIn_CSCFG.SSEL0EN is "0", any memory access corresponding to the Slave Select 0 ends up asserting an error of DDRHSSPIn_FAULTF.UMAFS = "1".

Generation of 32-bit Memory Address

The Address Extension Mechanism allows the mapping of 256MB of the MCU address space to 4GB of address space on a Slave Select line. Every Serial Flash Memory can be visualized as entity, consisting of several memory banks. The size of each bank can be programmed in the DDRHSSPIn_CSCFG.MSEL. Each bank can be selected by changing the value in the DDRHSSPIn_CSAEXT Register. By reprogramming the DDRHSSPIn_CSAEXT Register, each time a new bank in the selected Serial Flash Memory is accessed. Through different banks, it is possible to address a Serial Flash Memory of up to 4GB size.

Please refer to Figure 4-6. It shows how each 4GB Serial Flash Memory consists of 524288 banks when the DDRHSSPIn_CSCFG.MSEL is programmed to "0000".

Figure 4-6 Addressing 4GB Serial Flash Memories on Each Slave Select, Through Different Banks (DDRHSSPIn_CSCFG.MSEL="0000")



The Least Significant Bits of the System Bus Address received by the DDRHSSPI on the System Bus are used as the offset within the bank selected by the Address Extension bits.

The concatenation of the appropriate number of bits from the Address Extension Register with the appropriate number of bits from the address bus, determine the 32-bit address of the memory to be accessed on the Serial Interface. Please see Table 4-5.

Table 4-5 MCU Address Space to Memory Address Mapping

DDRHSSPIn_CSCFG.MSEL	Size of a memory bank on each Slave Select / Size of the System Bus address range associated with each Slave Select	Number of Slave Select lines that can be activated	Number of bits used from DDRHSSPIn_CSAEXT Register, for selection of the memory bank on a Slave Select	Number of bits used from address bus for addressing the memory location within a bank
0000	8K Bytes	SSEL0, SSEL1, SSEL2 and SSEL3	AEXT[31:13]	HADDR[12:0]
0001	16K Bytes		AEXT[31:14]	HADDR[13:0]
0010	32K Bytes		AEXT[31:15]	HADDR[14:0]
0011	64K Bytes		AEXT[31:16]	HADDR[15:0]
0100	128K Bytes		AEXT[31:17]	HADDR[16:0]
0101	256K Bytes		AEXT[31:18]	HADDR[17:0]
0110	512K Bytes		AEXT[31:19]	HADDR[18:0]
0111	1M Bytes		AEXT[31:20]	HADDR[19:0]
1000	2M Bytes		AEXT[31:21]	HADDR[20:0]
1001	4M Bytes		AEXT[31:22]	HADDR[21:0]
1010	8M Bytes		AEXT[31:23]	HADDR[22:0]
1011	16M Bytes		AEXT[31:24]	HADDR[23:0]
1100	32M Bytes		AEXT[31:25]	HADDR[24:0]
1101	64M Bytes		AEXT[31:26]	HADDR[25:0]
1110	128M Bytes	SSEL0 and SSEL1 Only	AEXT[31:27]	HADDR[26:0]
1111	256M Bytes	SSEL0 Only	AEXT[31:28]	HADDR[27:0]

Last two columns in Table 4-5 indicate which bits from the DDRHSSPIn_CSAEXT.AEXT and the address bus (i.e. HADDR) are concatenated, to get the final 32-bit address of the Serial Flash Memory.

Although, the final memory address generated in this way is a 32-bit address, it must be noted, that the software can choose the number of bytes (from this 32-bit address) to be sent to the Serial Flash Memory during the address phase of a memory read Command Sequence.

Internal Prefetch Buffer

In Command Sequencer Mode the DDRHSSPI internally has one Prefetch Buffer for temporary storage, of the data to be received (Prefetch Buffer).

This Prefetch Buffer consists of 48 locations each 32-bit wide.

Prefetch Buffer shall support 32 bits, 16 bits and 8 bits reads using the following mechanism:

In generation of the Command Sequence the two least significant address bits shall be ignored.

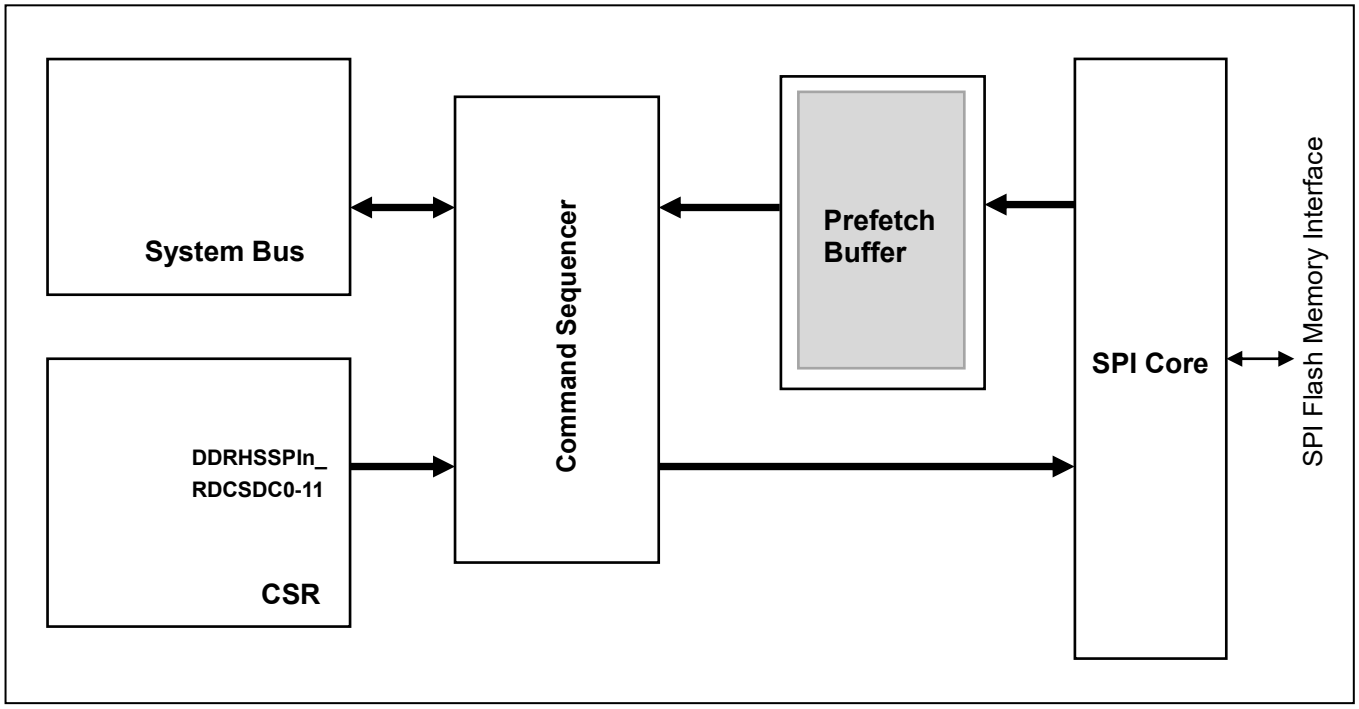
When a byte or halfword is read at a random (non-sequential) address, the Command Sequence shall be generated at the aligned word address which contains this byte or halfword. In any access scenario where the required data can be read from the top most position of the Prefetch Buffer, there will be no additional Command Sequence.

When the Prefetch Buffer is read (as FIFO), only if the read includes the last byte within the actual Prefetch Buffer word, it will be popped. Following are typical read scenarios with explanation when the Prefetch Buffer pops:

- When the whole aligned word is read, FIFO will be popped
- When the upper aligned halfword (word address + 2) is read, FIFO will be popped
- When the most significant byte (word address + 3) is read, FIFO will be popped
- In all other cases, FIFO will be read and will not be popped

In order to obtain maximum throughput, 32-bit access is highly recommended.

Figure 4-7 DDRHSSPI in Command Sequencer Mode



In order to see if the requested data is already available on the top of the Prefetch Buffer, the SW can read the DDRHSSPIn_CSPREFETCHADDR Register which keeps the first available address at the top of the Prefetch Buffer.

Initiation of Command Sequence

Whenever the Command Sequencer receives a System Bus read access for the memory mapped Serial Flash Memory, it initiates a corresponding memory read command on one of the four Slave Select lines, assembles the data it has received, and responds with the memory read-data. Only Flash Memory read commands are supported in the Command Sequencer Mode.

While the DDRHSSPI initiates a memory-read command and receives the read-data from the Serial Flash Memory, the System Bus Slave port of the DDRHSSPI inserts WAIT states on the System Bus. The DDRHSSPI keeps track of the previous address and the System Bus transfer type issued by the System Bus master. If the new transaction address is not contiguous, then a new serial transaction is issued on the Serial Interface.

Idle Timeout

After a Serial Flash Memory is accessed in the Command Sequencer Mode, the DDRHSSPI keeps asserting the Slave Select line even if the System Bus transaction is over. And DDRHSSPI can deassert the Slave Select automatically after waiting for a certain period of time. To use this feature, it is necessary to set DDRHSSPIn_CSCFG.ITIMEREN bit to "1".

If this feature is disabled, Slave Select remains asserted after any transfer on the Serial Flash Memory.

If this feature is enabled, the following part explains the Idle Timer behavior:

During idle cycles on the System Bus, prefetch reads from the Serial Flash Memories to the Prefetch Buffer are performed. As soon as the Prefetch Buffer is full, the Slave Select shall be kept asserted, and the Idle Timer shall start running. If there are no subsequent System Bus accesses to the consecutive memory address during the idle time, then after the Idle Timer expires, the DDRHSSPI deasserts the Slave Select, indicating the termination of the transfer and flushes the Prefetch Buffer.

Whenever a prefetch is performed before the Idle Timer expires, the Idle Timer shall be cleared again. Within the predefined time period, defined by the DDRHSSPIn_CSITIME.ITIME, the DDRHSSPI determines whether to extend the current serial transaction. If the following conditions are satisfied:

- A new System Bus transaction is detected on the System Bus, AND
- The new address is contiguous with the previous transaction

the DDRHSSPI extends the current serial transfer, instead of initiating a new Command Sequence. Thus, it can reduce the access time on the Serial Interface.

If the following conditions are satisfied:

- There is a subsequent memory access during the idle time, AND

- The access is to a non-consecutive memory address, AND

- The requested read data is not ready on the top of the Prefetch Buffer

the DDRHSSPI deasserts the Slave Select (indicating the termination of the current transfer), even before the Idle Timer expires. Then, the DDRHSSPI flushes the Prefetch Buffer and initiates a new Command Sequence.

Thus, the DDRHSSPIn_CSITIME.ITIME is used to enhance the overall performance of the memory accesses by extending the serial transaction for a programmed period.

The value of the DDRHSSPIn_CSITIME.ITIME is based on the System Bus Clock (iHCLK).

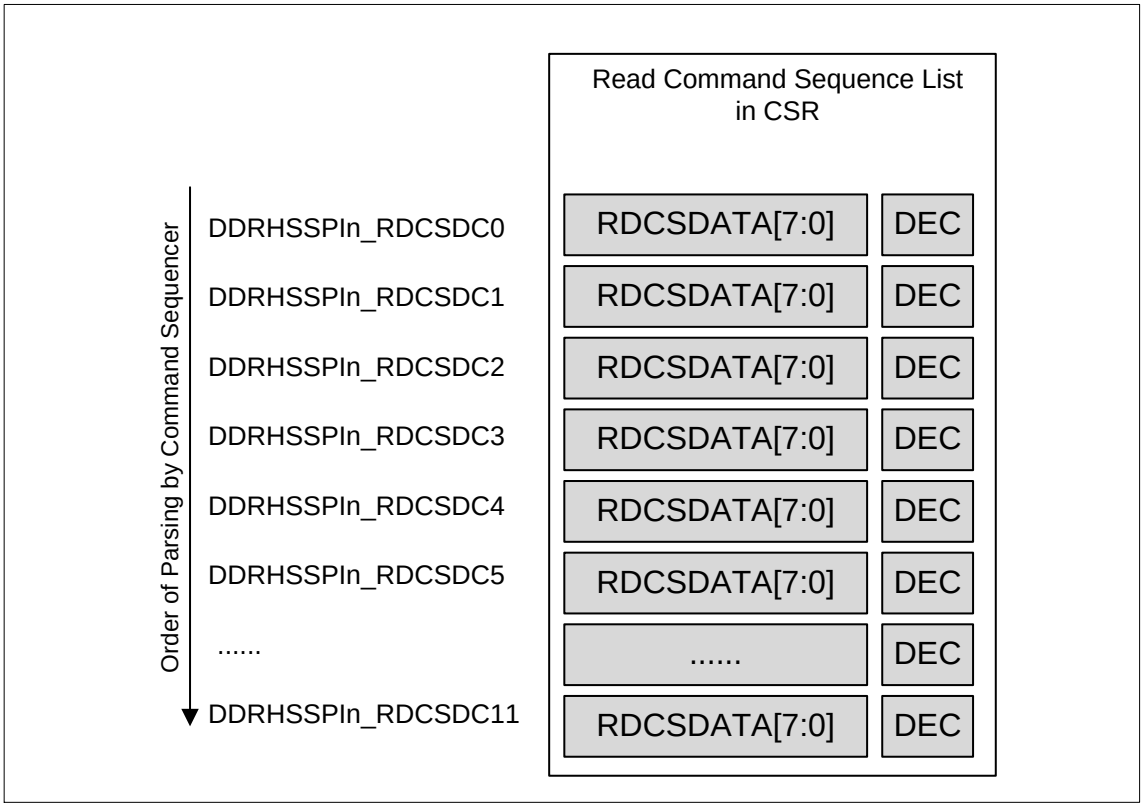
Configuration of Command Sequencer Mode in the CSR

The Command Sequencer supports memory read accesses. The sequence of command phases (i.e. command-phase, address-phase and data-phase) by the combination of Direct Mode and Command Sequencer Mode, is configured by the software (during initialization of the DDRHSSPI) in the CSR.

- Generation of Serial Flash Memory Read Command Sequence

For memory read transactions, the sequence of command phases (address-phase and data-phase) can be configured in the list of 12 registers: DDRHSSPIn_RDCSDC0-11. Starting from the DDRHSSPIn_RDCSDC0, up to the DDRHSSPIn_RDCSDC11 each of the 12 registers in the list is parsed. Please refer to Figure 4-8.

Figure 4-8 Memory Read Command Sequence List



The DEC bit in each of these registers determines whether to decode RDCSDATA[2:0] (as shown in Table 4-6). If DEC bit is "0", the data byte in RDCSDATA[7:0] is transmitted as it is.

Table 4-6 Decoding of the Read Command Sequence List

DEC	RDCSDATA [7:0]	Description
0	Don't Care	Transmit RDCSDATA[7:0] as it is.
1	0000_0000	Transmit address bits [7:0] of the Serial Flash Memory to be accessed
1	0000_0001	Transmit address bits [15:8] of the Serial Flash Memory to be accessed
1	0000_0010	Transmit address bits [23:16] of the Serial Flash Memory to be accessed
1	0000_0011	Transmit address bits [31:24] of the Serial Flash Memory to be accessed
1	XXXX_X100	Dummy cycle(s). The bit field RDCSDATA [7:3] means the length of dummy cycles: SDATA[7:3]="00000" -> Generates 1 dummy cycle of the SCLK SDATA[7:3]="00001" -> Generates 2 dummy cycles of the SCLK ..... Bits at "X" are for the value to control the length of dummy cycles.
1	0000_0111	End of List

The Command Sequencer switches to data-read cycles by the following conditions:

- If It gets "End of List", OR
- After the DDRHSSPIn_RDCSDC11 Register

In response to the System Bus Read transaction, during data-read cycles, the serial data on the SDATA lines is sampled, and the assembled data is returned to the System Bus master.

The actual patterns allowed in this Command Sequence List is limited in the following formats. This is based on using a Continuous Read Command Sequence, and the Command Sequence begins with Address field for the Serial FLASH Memory.

(1)24-bit Address

Table 4-7 Command Sequence List for 24-bit Address

Register	RDCSDATA [7:0]	DEC	Description
DDRHSSPIn_ RDCSDC0	0000_0002	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC1	0000_0001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC2	0000_0000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC3	byte data	0	Mode code (*1)
DDRHSSPIn_ RDCSDC4	XXXX_X100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_ RDCSDC5	0000_0111	1	End of List

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

(2)32-bit Address

Table 4-8 Command Sequence List for 32-bit Address

Register	RDCSDATA [7:0]	DEC	Description
DDRHSSPIn_ RDCSDC0	0000_0003	1	address bits [31:24] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC1	0000_0002	1	address bits [23:16] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC2	0000_0001	1	address bits [15:8] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC3	0000_0000	1	address bits [7:0] of the Serial Flash Memory
DDRHSSPIn_ RDCSDC4	byte data	0	Mode code (*1)
DDRHSSPIn_ RDCSDC5	XXXX_X100	1	Dummy cycles Bits at "X" are for the value to control the length of dummy cycles.
DDRHSSPIn_ RDCSDC6	0000_0111	1	End of List

(*1)The "Mode" field is a control code following Address field, and the actual code of Mode field depends on the Serial Flash products.

The Operation of Memory Read Access

To perform memory read accesses to the Serial Flash Memories, it is necessary to run the following combination of Direct Mode and Command Sequencer Mode, because the Command Sequencer Mode cannot change SPI width on-the-fly.

1. Switch to Direct Mode, if it is in Command Sequencer Mode. Clear DDRHSSPIn_MCTRL.CSEN bit to "0".
2. Write a byte count of the transmission to the DDRHSSPIn_DMBCC.BCC. The byte count shall be a sum of command, address, and mode code. For example, if the address has 4 bytes, the BCC is 6.
3. Set the DDRHSSPIn_DMTRP.TRP to "1010" (TX-Only Mode and Quad Mode).
4. When in DDR Mode, set the DDRHSSPIn_DMTRP.DDRM bit to "1". Otherwise, clear the DDRHSSPIn_DMTRP.DDRM bit to "0".
5. Set the DDRHSSPIn_DMFIFOCFG.FWIDTH to "00".
6. Set the DDRHSSPIn_DMFIFOCFG.TXCTRL bit to "1".

7. Write a byte for the command field, into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "01" (Legacy Mode on-the-fly). TXDATA[10] = "0" (SDR Mode on-the-fly). TXDATA[7:0] is a byte for the command. Please refer to the data sheet of the Serial Flash Memory about the command code.
8. Write a byte for the address field [31:24], into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "10" (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the address[31:24].
9. Write bytes for the address field [23:16], [15:8] and [7:0] into TX-FIFO as well.
10. Write a byte for the mode field following the address field, into TX-FIFO. TXDATA[12] = "1", TXDATA[9:8] = "10" (Quad Mode on-the-fly). TXDATA[10] = DDRHSSPIn_DMTRP.DDRM bit value. TXDATA[7:0] is a byte for the mode. Please refer to the data sheet of the Serial Flash Memory about the mode code.
11. Write "1" to the DDRHSSPIn_TXC.TSSRC bit.
12. Set the DDRHSSPIn_TXE.TSSRE bit to "1".
13. Select a Serial Flash Memory by setting the DDRHSSPIn_DMPSEL.PSEL value.
14. Set the DDRHSSPIn_DMSTART.START bit to "1".
15. Wait for the TX Interrupt Request, or run a polling of DDRHSSPIn_TXF Register until the DDRHSSPIn_TXF.TSSRS bit to be "1".
16. Write "1" to the DDRHSSPIn_TXC.TSSRC bit.
17. Clear the DDRHSSPIn_TXE.TSSRE bit to "0".
18. Switch to Command Sequencer Mode. Set DDRHSSPIn_MCTRL.CSEN bit to "1".
19. Set the DDRHSSPIn_CSCFG.MBM same as DDRHSSPIn_DMTRP.TRP[1:0] value. Set DDRHSSPIn_CSCFG.MSEL to appropriate value. Set at least one of DDRHSSPIn_CSCFG.SSEL0EN, DDRHSSPIn_CSCFG.SSEL1EN, DDRHSSPIn_CSCFG.SSEL2EN or DDRHSSPIn_CSCFG.SSEL3EN to "1", in order to contain the Serial Flash Memory selected by DDRHSSPIn_DMPSEL.PSEL. Set DDRHSSPIn_CSCFG.DDRMODE bit same as DDRHSSPIn_DMTRP.DDRM bit.
20. Set the appropriate value to DDRHSSPIn_CSITIME.ITIME.
21. Set the appropriate value to DDRHSSPIn_CSAEXT.AEXT.
22. Write appropriate values to DDRHSSPIn_RDCSDC0-11 Registers according to Table 4-7 or Table 4-8.
23. Perform memory read accesses to the address where Serial Flash Memories are allocated. The memory accesses on the System Bus are translated to serial transactions on the Serial Interface.
24. As long as performing the memory read accesses, there is no need to change the register settings. When in issuing a new set of command, it is necessary to stop current Continuous Read Command Sequence. To stop it, go to the next step.
25. Operate steps from 1 to 17 in Direct Mode to get out of current Continuous Read Command Sequence. In this operation, please write a different code from the step 10. About the exact code, please refer to the data sheet of the Serial Flash Memory about the mode code.
26. The SW can issue a new set of command.

Prefetch memory access in Command Sequencer Mode

After the DDRHSSPI is enabled and before the first access, no prefetch accesses to the Serial Flash Memory shall be performed. Once the first System Bus accesses are completed and if there are no new System Bus requests, the DDRHSSPI will initiate the continuous prefetch Flash Memory reads, on the Serial Interface until the Prefetch Buffer is full.

In case of new System Bus transfers for the Serial Flash Memory reads (in Command Sequencer Mode), two scenarios are possible as below:

a) Prefetch Buffer miss

In case of a Prefetch Buffer miss, this means that the address requested at the System Bus transaction is not same as the one available on the top of the Prefetch Buffer. This address is also readable at the DDRHSSPIn_CSPREFETCHADDR Register.

In this case, the following sequence occurs:

1. A Prefetch Buffer miss is detected
2. The Prefetch Buffer is flushed
3. A new Command Sequence to the Serial Flash Memory is generated, with a new aligned address

b) Prefetch Buffer hit

In case of a Prefetch Buffer hit, this means that the address requested at the System Bus transaction is same as the one available on the top of the Prefetch Buffer. In this case, the DDRHSSPI can immediately respond with valid data on the System Bus.

In both scenarios, when the Prefetch Buffer is full, the SCLK will be halted and the Slave Select will work according to the following conditions.

- DDRHSSPIn_CSCFG.ITIMEREN bit = "1": The Slave Select will be released after the Idle Timer expires.
- DDRHSSPIn_CSCFG.ITIMEREN bit = "0": The Slave Select will remain asserted.

In general, PB keeps its content as long as it is not flushed or popped.

There is difference in terms for PB pop and PB read as below:

- PB pop:

  Access to the PB top word results in reading-and-popping PB (as FIFO).

  The internal read pointer of PB increments.

- PB read:

  Access to the PB top word results in just reading.

  The internal read pointer of PB does not increment.

When in reading PB, if the last byte of a word on the PB top is read, it causes a PB pop, otherwise it causes a PB read. The position of the last byte depends on the access width on the System Bus.

4.3. Address Map of DDRHSSPI

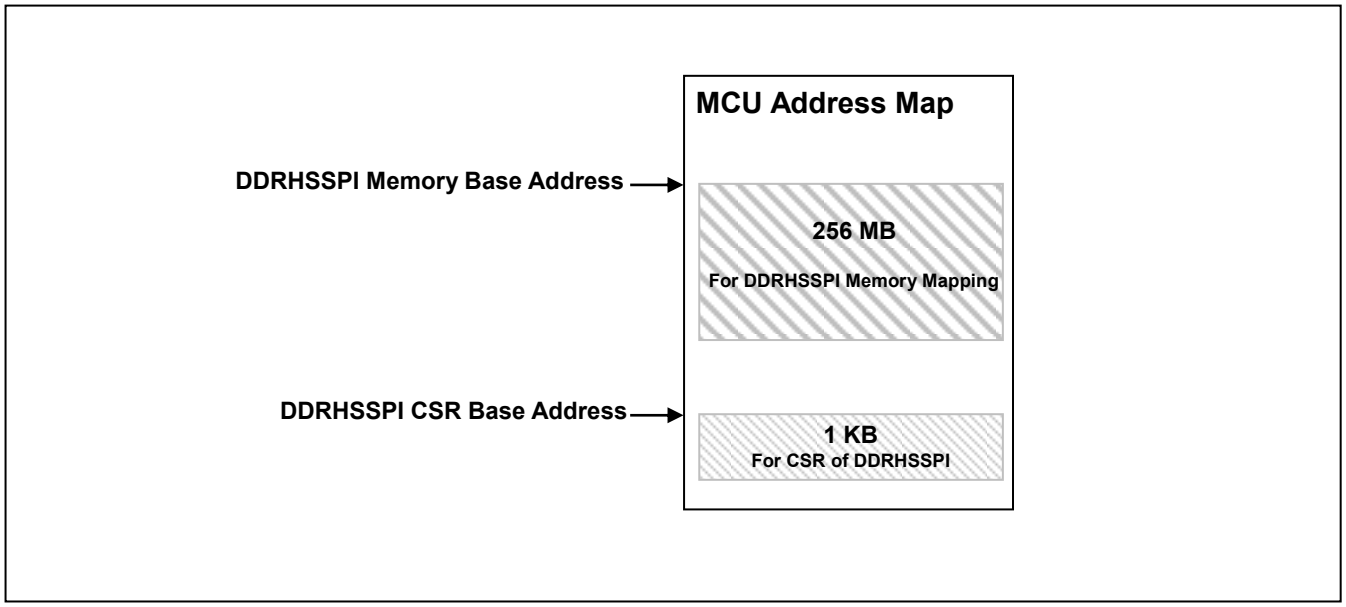
The DDRHSSPI is allocated 256MB of MCU's address space for memory mapping of the external Serial Flash Memories, using Command Sequencer Mode. An additional 1KB of MCU's address space is reserved for mapping the internal Configuration and Status Registers (i.e. CSRs) of the DDRHSSPI.

The address area allocated to the DDRHSSPI is explained in this section.

Arrangement of DDRHSSPI Address Space in Memory

Figure 4-9 shows the allocation of DDRHSSPI address space in the MCU's address space.

Figure 4-9 Address Map of DDRHSSPI



- Allocation for Serial Flash Memories

The 256MB of memory space, starting from "DDRHSSPI Memory Base Address" is reserved for memory mapping of the external Serial Flash Memories onto the MCU's address space.

This space is used in Command Sequencer Mode.

- Allocation for CSRs

The 1KB of memory space, starting from "DDRHSSPI CSR Base Address" is reserved for memory mapping of the Configuration and Status Registers of DDRHSSPI on to the MCU's address space.

4.4. General Use Case Guidelines for DDRHSSPI

This section lists up the guidelines for programming the DDRHSSPI. It is strongly recommended to read these guidelines before programming.

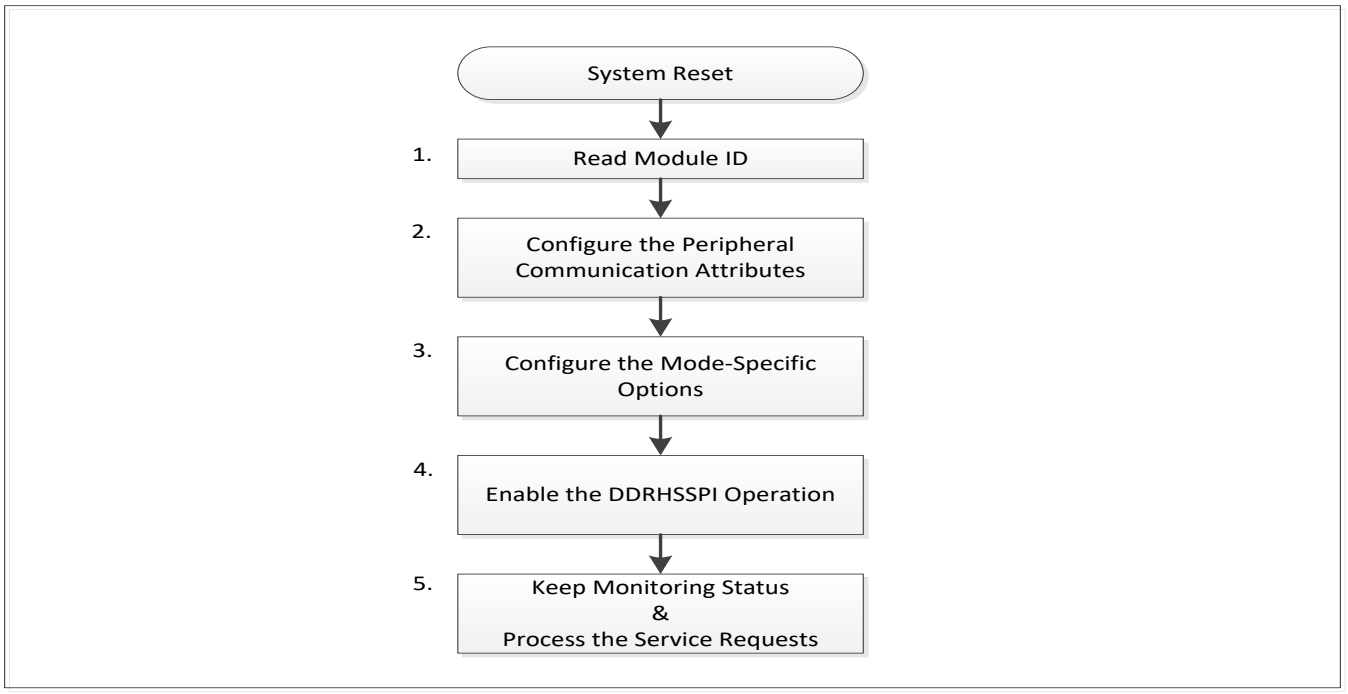
Guidelines on Typical Use Cases of DDRHSSPI

- Any serial-transaction related parameters and control bits (DDRHSSPIn_DMCFG, DDRHSSPIn_DMBCC, DDRHSSPIn_DMBCS, DDRHSSPIn_DMTRP, DDRHSSPIn_DMPSEL and DDRHSSPIn_DMFIFOCFG) should not be performed while a serial transaction is in progress. Any such changes should be performed only after the current serial transfer has ended (i.e. DDRHSSPIn_TXF.TSSRS="1" or DDRHSSPIn_RXF.RSSRS="1"). To ensure that the DDRHSSPI has finished all of its transfers, the software can read the DDRHSSPIn_DMFIFOSTATUS.SSACTIVE.
- Only one configurable FIFO is used, with different configuration settings, for each Direct Mode and Command Sequencer Mode. Hence, when switching from one mode to another, all FIFO data will be lost.
- When the Serial Flash Memory has a command format with changing SPI protocols, DDRHSSPI supports it by Direct Mode. In the Direct Mode, it can transfer the initial set of byte(s) in Legacy Mode using SDR clock, and then transfer the remaining sets of data bytes using the Quad Mode. In such cases, software needs to use DDRHSSPIn_DMFIFOCFG.TXCTRL bit and write TX-FIFO data with additional control such as Data Rate Mode (SDR Mode or DDR Mode) and SPI data width (Legacy Mode, Quad Mode or Octal Mode). Thus, for example, it is possible to transmit the first byte in Legacy Mode with SDR clock, and the following bytes in Quad Mode with DDR clock.
- When Direct Mode is used, the software shall be responsible to take care that the internal FIFOs of DDRHSSPI do not get overrun or underrun. In case the FIFOs get overrun or underrun, the FIFO fill-levels (i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL and DDRHSSPIn_DMFIFOSTATUS. RXFLEVEL) are no longer pertinent and the software would have to flush the FIFOs.
- The DDRHSSPIn_DMFIFOCFG.RXFLSH and DDRHSSPIn_DMFIFOCFG.TXFLSH bits shall be used by the software to flush the corresponding FIFOs and Shift Registers before using them for any serial transfer. Flushing a FIFO and Shift Registers ensures that they are not pre-loaded with any garbage data (possibly from the previous transfer).
- In Direct Mode, whenever the transfer ends, the data from the RX Shift Register is pushed into the RX-FIFO; provided that the RX-FIFO is not full.
  - If the RX-FIFO is already full while a serial transfer is terminating, the serial transfer halts and the remainder data remains in the RX Shift Register as long as DDRHSSPI is halted. As soon as the RX-FIFO is not full anymore, DDRHSSPI comes out of the halt state and the remainder data from the RX Shift Register is pushed into the RX-FIFO before DDRHSSPI releases the Slave Select line. The remainder data is pushed into the RX-FIFO irrespective of whether the RX Shift Register is filled completely or partially. Thus, in Direct Mode, remainder data never remains in the RX Shift Register.
  - In Direct Mode when DDRHSSPI is configure in SDR Mode (when DDRHSSPIn_DMTRP. DDRM bit is set to "0"), it is not allowed to switch to DDR Mode while module is enabled.

Steps in Programming the DDRHSSPI Module

Figure 4-10 gives the general steps a programmer shall follow while using the DDRHSSPI.

Figure 4-10 Programmer's Flowchart: General Steps

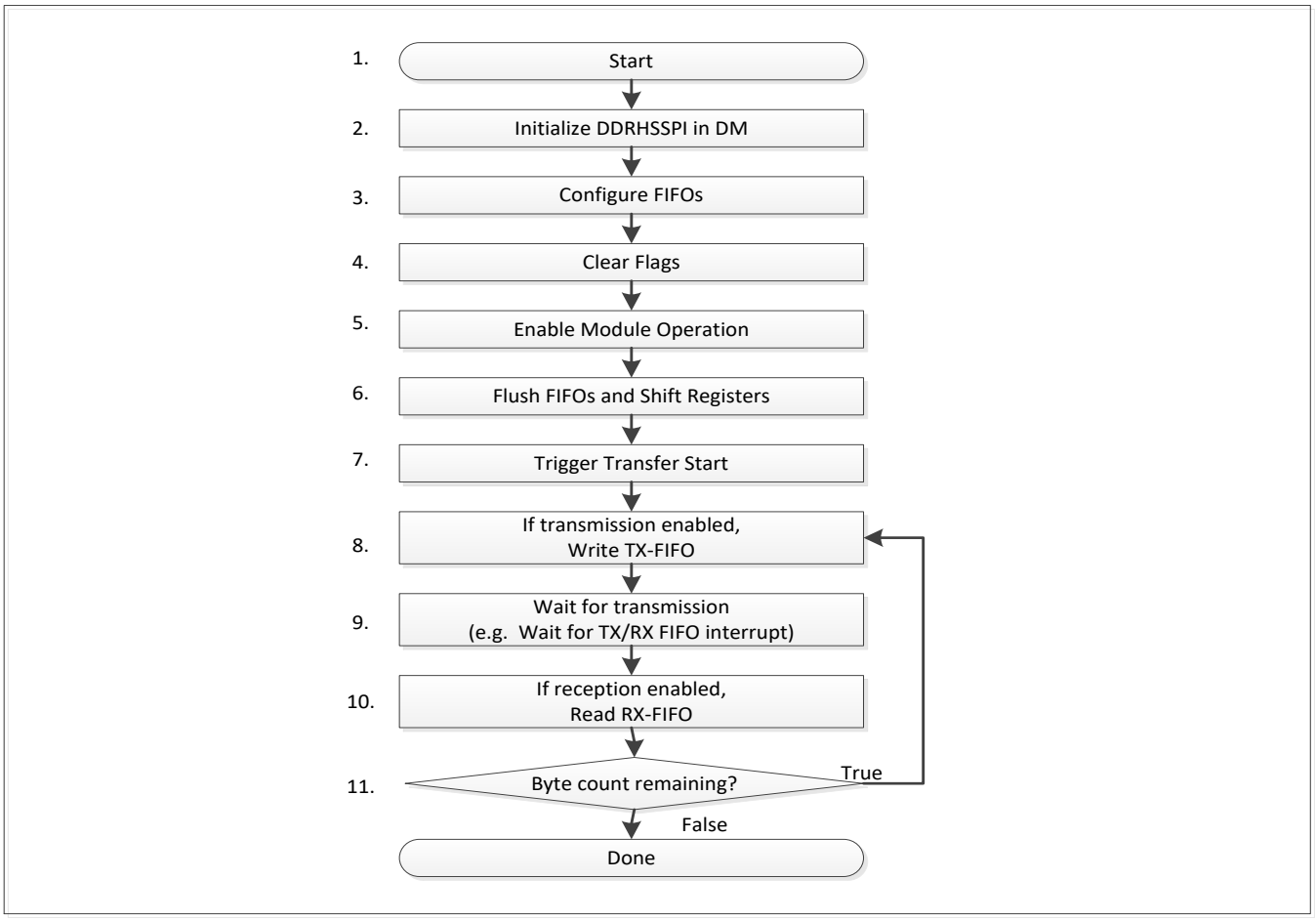


1. After the System Reset, the software shall detect the Module ID number of DDRHSSPI, by reading the DDRHSSPIn_MID Register. This would help it in identifying the attributes and capabilities supported by the DDRHSSPI.
2. The next step is to configure the Attributes related to the Peripheral Communication with the Serial Flash Memory(Memories) connected with DDRHSSPI. DDRHSSPI can be interfaced with up to 4 Serial Flash Memories. Serial communication related attributes like Transfer Frequency (i.e. Clock Division Ratio bits), etc. shall be configured in the registers: DDRHSSPIn_PCC0-3. It is very important that these attributes shall be the same as the Serial Flash Memory, which is connected with DDRHSSPI. These configurations shall not be modified while the DDRHSSPI is active. In case the software has to re-program any of these values, the software shall first disable the DDRHSSPI and wait until the current serial transfer is finished.
3. DDRHSSPI can be configured either in Direct Mode or in Command Sequencer Mode, through the DDRHSSPIn_MCTRL.CSEN bit. Depending on which mode is to be used, the software shall configure the mode-specific registers. The registers specific to the Direct Mode are: (DDRHSSPIn_DMCFG, DDRHSSPIn_DMBCC, DDRHSSPIn_DMBCS, DDRHSSPIn_DMTRP, DDRHSSPIn_DMPSEL and DDRHSSPIn_DMFIFOCFG) and the registers specific to the Command Sequencer Mode are: (DDRHSSPIn_RDCSDC0-11, DDRHSSPIn_CSCFG, DDRHSSPIn_CSITIME, DDRHSSPIn_CSAEXT and DDRHSSPIn_CSPBUFFERCFG).
4. Only after all module-specific configurations are programmed, the DDRHSSPI shall be enabled (by setting the DDRHSSPIn_MCTRL.MEN to "1").
5. Once the DDRHSSPI is enabled, its normal working begins. The software shall keep monitoring the status of the DDRHSSPI using the various status bits. If the DDRHSSPI is configured for initiating the service requests, it would periodically trigger the service requests (i.e. Interrupts and/or DMA Service Requests). The software would service those requests, in order to ensure the normal working of DDRHSSPI.

Using the DDRHSSPI in Direct Mode of Operation

Figure 4-11 gives the general steps which the SW shall follow while using the DDRHSSPI in Direct Mode.

Figure 4-11 Programmer's Flowchart: DDRHSSPI in Direct Mode of Operation



1. After the System Reset, the software shall initialize the DDRHSSPI by reading the DDRHSSPIn_MID Register and setting the Peripheral Communication related attributes in the DDRHSSPIn_PCC0, DDRHSSPIn_PCC1, DDRHSSPIn_PCC2 and DDRHSSPIn_PCC3 Registers. Please make sure that the DDRHSSPIn_MCTRL.CSEN bit is cleared to "0".
2. The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Legacy or the Quad Protocol and whether the DDRHSSPI would be used only for transmission, or both for transmission and reception) in the DDRHSSPIn_DMTRP.TRP. DDRHSSPI loads the DDRHSSPIn_DMBCC.BCC with the number of bytes to be serially transferred.
3. Configure the DDRHSSPIn_DMFIFOCFG Register, to set the FIFO threshold levels. By programming these levels, the assertion of the service requests can be controlled. Also configure the DDRHSSPIn_DMFIFOCFG. FWIDTH, to select the width of the FIFOs. Configure the service requests: DDRHSSPI supports both Interrupt Service Request and DMA Service Request, for the normal data read operations from RX-FIFO or write operations to TX-FIFO. For normal operation, either the Interrupt Service Requests or the DMA Service Requests shall be enabled by the software. To enable the Interrupt Service Requests for writing TX-FIFO, please program the bits in the DDRHSSPIn_TXE Register. To enable the Interrupt Service Requests for reading RX-FIFO, please program the bits in the DDRHSSPIn_RXE Register. To enable the DMA Service Request (for writing and/or reading), please program either/both of the DDRHSSPIn_DMAEN.TXDMAEN and the DDRHSSPIn_DMAEN.RXDMAEN bits. The DMA Read Channel must be setup to perform a block transfer of "DDRHSSPIn_DMFIFOCFG.RXFTH + 1" transfers. The DMA Write Channel must be

setup to perform a block transfer of "24 - DDRHSSPIn_DMFIFOCFG.TXFTH" transfers. Select the peripheral (in DDRHSSPIn_DMPSEL.PSEL) on which DDRHSSPI shall initiate the transfer
4. Clear all relevant flags. This finishes the steps in initialization of DDRHSSPI for Direct Mode.
5. Set the DDRHSSPIn_MCTRL.MEN bit, to enable the module.
6. Flush FIFOs to ensure data consistency and avoid any data corruption from previous transfers.
7. When DDRHSSPI is configured, setting the DDRHSSPIn_DMSTART.START bit triggers the start of the serial transaction. Once the serial transaction starts, if transmission is enabled in the DDRHSSPIn_DMTRP.TRP, the DDRHSSPI reads data from TX-FIFO and loads them to the Shift Register. The Shift Register is shifted left and the transmit data is shifted-out onto the Serial Interface. If DDRHSSPI is enabled for Receive operation (in DDRHSSPIn_DMTRP.TRP), the DDRHSSPI receives the serial data with shifting the Shift Register. The received data assembled in the Shift Register is pushed into the RX-FIFO.
8. Write the data to be transmitted into the TX-FIFO via DDRHSSPIn_TXFIFO0-23 Register address. Before writing to the DDRHSSPIn_TXFIFO0-23 Register, modify the value of the DDRHSSPIn_DMFIFOCFG. TXCTRL bit appropriately. Generally (i.e. when the data being written to the TX-FIFO is to be transmitted as it is), the DDRHSSPIn_DMFIFOCFG.TXCTRL bit shall be "0". Only when in adding some kind of controls such as dummy cycles, the DDRHSSPIn_DMFIFOCFG.TXCTRL bit shall be set to "1". The write access to DDRHSSPIn_TXFIFO0-23 shall be performed after the control of the DDRHSSPIn_DMFIFOCFG.TXCTRL bit.
9. Service Requests are asserted by DDRHSSPI whenever the TX-FIFO level is below the threshold or whenever the DDRHSSPI RX-FIFO level is above the threshold. The software shall write TX-FIFO or read RX-FIFO, to ensure the serial data transfer of DDRHSSPI. After writing or reading the relevant FIFO, the software shall clear the Interrupt Service Requests by writing the DDRHSSPIn_TXC or the DDRHSSPIn_RXC Register. DMA Service Requests are cleared by the DMA Controller.
10. If reception is enabled in DDRHSSPIn_DMTRP Register, then the software fetches the received data from the RX-FIFO.
11. Software judges if current serial transfer has finished, by checking (1)the DDRHSSPIn_TXF.TSSRS bit to be "1" or (2)the DDRHSSPIn_DMBCS Register value to be 0x0000. In the normal course of operation, the software usually keeps repeating steps from 8 to 11 until the end of serial transfer.

When the software initiates a new serial transfer again, it starts this flow from step 2.

To switch between the Direct Mode and Command Sequencer Mode, or to re-program any of the parameters that directly affect the serial transfer, the software shall first stop the current transfer and disable the DDRHSSPI (by resetting DDRHSSPIn_MCTRL.MEN bit to "0"). The software can check the status bit DDRHSSPIn_TXF.TSSRS, to see if the current transfer has finished.

Using the DDRHSSPI in Command Sequencer Mode of Operation

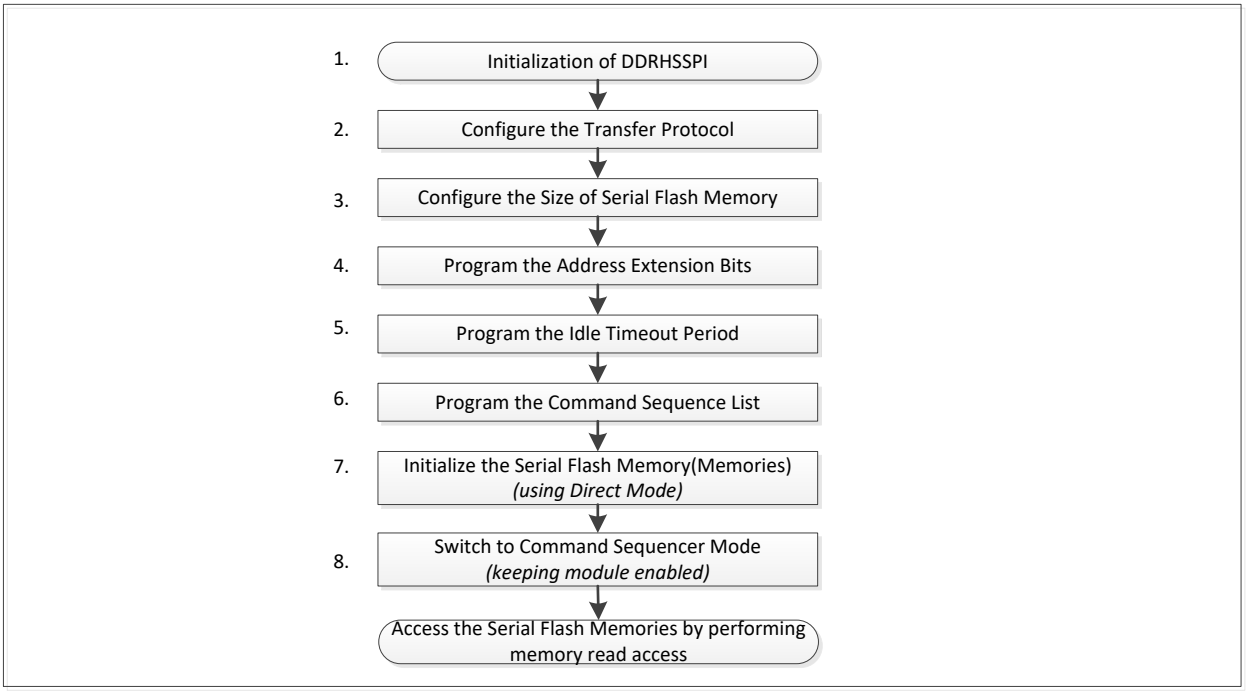
- Using the Memory Mapped Memories

Following usage rules shall be followed, when interfacing Serial Flash Memories, for memory-mapped accesses in Command Sequencer Mode.

- Usage Rules and Notes
  - In Command Sequencer Mode, all Serial Flash Memories interfaced with DDRHSSPI shall be of same family. Do not mix Serial Flash Memories from different vendor families.
  - If Serial Flash Memories of same family, but with different memory sizes are to be interfaced, then while deciding the suitable value of the DDRHSSPIn_CSCFG.MSEL, the Serial Flash Memory with maximum size must be considered. However, it shall be noted here, that the number of bytes from the final memory address that will be transmitted by DDRHSSPI to the Serial Flash Memory is programmed in the Command Sequence lists (in DDRHSSPIn_RDCSDC0-11). Interfacing of one Serial Flash Memory which requires 32-bit addressing and other Serial Flash Memory (also of same family, but) which requires only 24-bit addressing in Command Sequencer Mode is not possible. This is because a Serial Flash Memory which has 24-bit addressing cannot be used with a bit-stuffed 32-bit address - its address phase is of 3 bytes only.
  - In Command Sequencer Mode, it is impossible to change the SPI width while in the serial transfer. For this reason, for some types of the Serial Flash Memories - the Command Sequencer can be enabled only after the Serial Flash Memory has transited to work in the "Continuous Read Command Sequence". The Serial Flash Memory can be programmed in the Continuous Read Command Sequence, using the Direct Mode operation of DDRHSSPI.
  - When Command Sequencer Mode is used, internal Prefetch Buffer will be used. Since in Command Sequencer Mode only memory reads are supported there is only Prefetch Buffer available in Command Sequencer Mode. This buffer is 48 location deep, 32 bit each.
  - In Command Sequencer Mode - prefetch of read data is used in order to streamline data reads. This means that during System Bus idle cycles - DDRHSSPI is performing speculative reads from Flash Memory until it gets internal Prefetch Buffer full. There is separate register DDRHSSPIn_CSPREFETCHADDR [31:0] which stores next available prefetch address. In case there is address miss with next System Bus transfer then Prefetch Buffer flush is performed.
  - On-the-fly switching of programmed Data Rate Mode (SDR Mode or DDR Mode) is not allowed in Command Sequencer Mode while module is enabled.
- Programmer’s Flowchart

Figure 4-12 shows the general steps how to allocate the Serial Flash Memories onto the address space of MCU.

Figure 4-12 Programmer's Flowchart: Memory Mapping of Serial Flash Memories



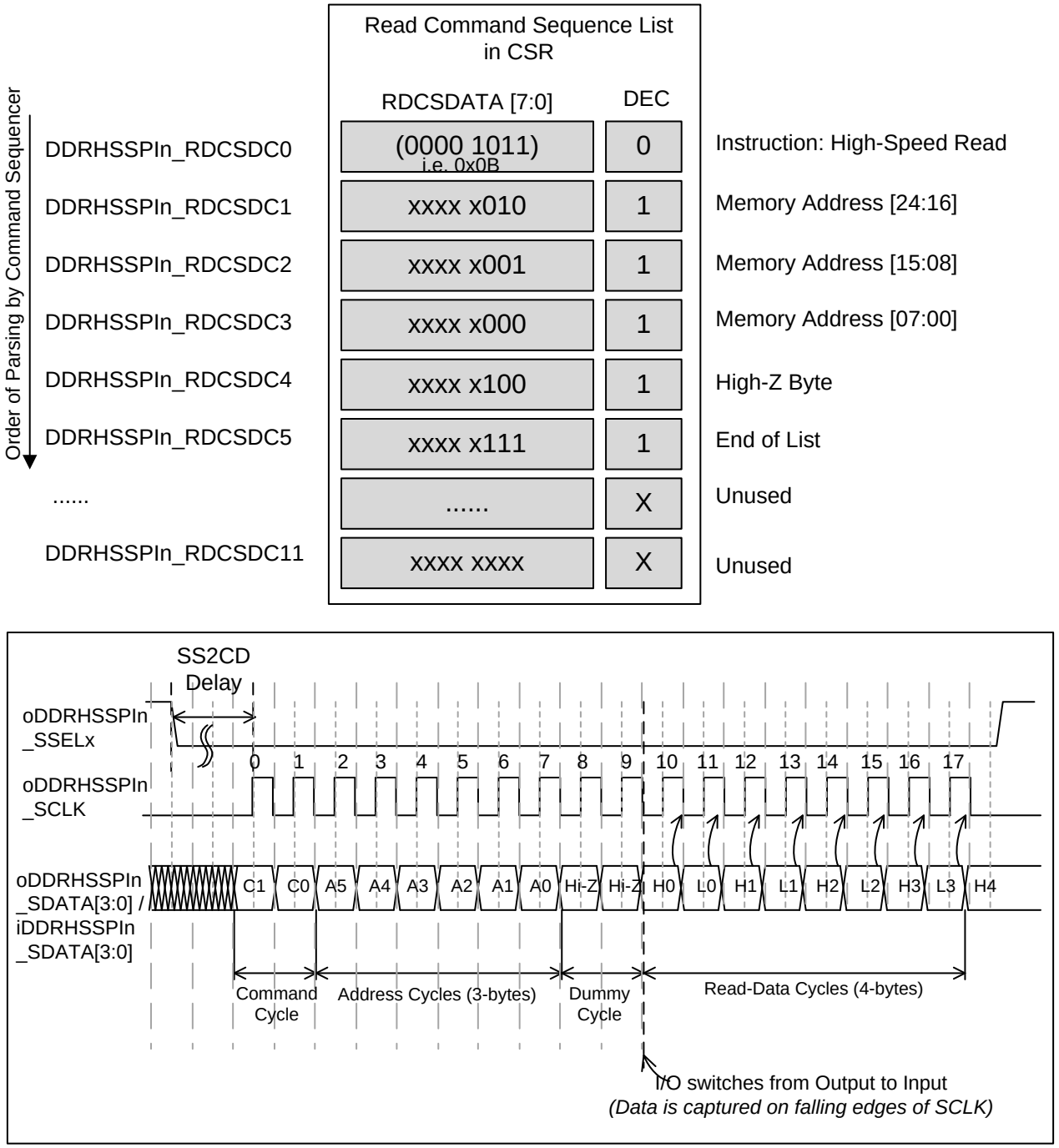
1. After the System Reset, the software shall initialize the DDRHSSPI by setting the Peripheral Communication related attributes in the DDRHSSPIn_PCC0-3 Registers. It is very important that these attributes shall be the same as being used by the Serial Flash Memory with which DDRHSSPI is interfaced. When Serial Flash Memories are to be memory-mapped using Command Sequencer Mode, all Serial Flash Memories shall be of same family. Therefore, all of DDRHSSPIn_PCC0-3 Registers shall have same configuration values.
2. The next step is to configure the transfer protocol (i.e. whether the DDRHSSPI serial transfers use the Quad or Octal Protocol in the DDRHSSPIn_CSCFG.MBM). The DDRHSSPIn_CSCFG. DDRMODE bit shall be set same as DDRHSSPIn_DMTRP.DDRM bit.
3. Program the DDRHSSPIn_CSCFG.MSEL, with the size of the System Bus address space which must be used in selection of the Serial Flash Memory on which the serial transfer must be initiated. Please refer to Section 4.2 for details of the Slave Select.
4. If the addresses generated for the memory-mapped accesses are to be virtually extended to cover a memory range of virtually 16GB, the DDRHSSPIn_CSAEXT Register value gives the upper bits of the address. Please refer to Section 4.2 for details of address generation.
5. The DDRHSSPIn_CSITIME.ITIME helps DDRHSSPI enhance the performance of memory accesses, by continuing previous serial transfer. If DDRHSSPI detects a consecutive memory access during ITIMER period (Slave Select is kept asserted and SCLK is halted), it extends the data transfer without deasserting current Slave Select. This feature reduces the access time by omitting a new Command Sequence. Program the DDRHSSPIn_CSITIME.ITIME with appropriate idle time-out value.
6. Program the list of Read Command Sequence Registers (i.e. DDRHSSPIn_RDCSDC0-11) with the sequence of the memory read command for the Serial Flash Memory which is interfaced. Please refer to the data sheet of the Serial Flash Memory for details of the Read Command Sequence.
7. The next step is to initialize the Serial Flash Memory that is to be memory mapped. The initialization is specific for the Serial Flash Memory, including the setting of some control or status bits in its register set. e.g. To use a Serial Flash Memory in a high-performance Quad Mode. Please refer to the data sheet of the Serial Flash Memory to be interfaced. This initialization of the Serial Flash Memory shall be performed using Direct Mode of DDRHSSPI.
8. With this, DDRHSSPI has been configured for accessing the memory-mapped devices. Switch the DDRHSSPI to Command Sequencer Mode, so that it starts generating the Read Command Sequences on the Serial Interface, by mapping the System Bus accesses to the memory-mapped locations.

- Timing Diagram for Command Sequencer

Figure 4-13 illustrates with an example, how the Command Sequencer generates the Serial Flash Memory Read Command Sequence. Assuming that the Read Command Sequence list is programmed in DDRHSSPIn_RDCSDC0-11 Registers, as shown in the figure, the Command Sequencer parses the list, starting from DDRHSSPIn_RDCSDC0 Register, and executes the commands as explained in Section 4.2 Command Sequencer Mode.

Figure 4-13 shows also the corresponding timing diagram for a Read Command Sequence, for Mode 4.

Figure 4-13 Read Command Sequence Illustration with Timing Diagram (*Mode4*)



Note:

- *The above example is to illustrate the relation between the Command Sequence List and the serial transmission, and it does not mean the actual example of the SPI transaction for Serial Flash Memory.*

4.4.1. Notice on Document Designations

Cypress issues documents and data sheets with Advance Information or Preliminary designations to advise readers of product information or intended specifications throughout the product life cycle, including development, qualification, initial production, and full production. In all cases, however, readers are encouraged to verify that they have the latest information before finalizing their design. The following descriptions of Cypress data sheet designations are presented here to highlight their presence and definitions.

4.4.2. Advance Information

The Advance Information designation indicates that Cypress is developing one or more specific products, but has not committed any design to production. Information presented in a document with this designation is likely to change, and in some cases, development on the product may discontinue. Cypress therefore places the following conditions upon Advance Information content:

This document contains information on one or more products under development at Cypress.

The information is intended to help you evaluate this product. Do not design in this product without contacting the Cypress.

Cypress reserves the right to change or discontinue work on this proposed product without notice.

5. Registers

The DDRHSSPI contains various registers to configure its operation, to monitor its status and to read or write the data to be transferred over the Serial Interface.

All registers in DDRHSSPI are explained in this section.

List of DDRHSSPI Registers in Memory

Abbreviation	Register Name	See
DDRHSSPIn_MCTRL	DDRHSSPI Module Control Register	5.1
DDRHSSPIn_PCC0	DDRHSSPI Peripheral Communication Configuration Register 0	5.2
DDRHSSPIn_PCC1	DDRHSSPI Peripheral Communication Configuration Register 1	
DDRHSSPIn_PCC2	DDRHSSPI Peripheral Communication Configuration Register 2	
DDRHSSPIn_PCC3	DDRHSSPI Peripheral Communication Configuration Register 3	
DDRHSSPIn_TXF	DDRHSSPI TX Interrupt Flag Register	5.3
DDRHSSPIn_TXE	DDRHSSPI TX Interrupt Enable Register	5.4
DDRHSSPIn_TXC	DDRHSSPI TX Interrupt Clear Register	5.5
DDRHSSPIn_RXF	DDRHSSPI RX Interrupt Flag Register	5.6
DDRHSSPIn_RXE	DDRHSSPI RX Interrupt Enable Register	5.7
DDRHSSPIn_RXC	DDRHSSPI RX Interrupt Clear Register	5.8
DDRHSSPIn_FAULTF	DDRHSSPI Fault Status Flag Register	5.9
DDRHSSPIn_FAULTC	DDRHSSPI Fault Status Clear Register	5.10
DDRHSSPIn_DMCFG	DDRHSSPI Direct Mode Configuration Register	5.11
DDRHSSPIn_DMAEN	DDRHSSPI DMA Enable Register	5.12
DDRHSSPIn_DMSTART	DDRHSSPI Direct Mode Start Register	5.13
DDRHSSPIn_DMPSEL	DDRHSSPI Direct Mode Peripheral Select Register	5.14
DDRHSSPIn_DMTRP	DDRHSSPI Direct Mode Transfer Protocol Register	5.15
DDRHSSPIn_DMBCC	DDRHSSPI Byte Count Control Register	5.16
DDRHSSPIn_DMBCS	DDRHSSPI Byte Count Status Register	5.17
DDRHSSPIn_DMFIFOSTATUS	DDRHSSPI Direct Mode FIFO Status Register	5.18
DDRHSSPIn_DMFIFOCFG	DDRHSSPI Direct Mode FIFO Configuration Register	5.19
DDRHSSPIn_TXFIFO0	DDRHSSPI TX-FIFO Register 0	5.20
DDRHSSPIn_TXFIFO1	DDRHSSPI TX-FIFO Register 1	
DDRHSSPIn_TXFIFO2	DDRHSSPI TX-FIFO Register 2	
DDRHSSPIn_TXFIFO3	DDRHSSPI TX-FIFO Register 3	
DDRHSSPIn_TXFIFO4	DDRHSSPI TX-FIFO Register 4	
DDRHSSPIn_TXFIFO5	DDRHSSPI TX-FIFO Register 5	
DDRHSSPIn_TXFIFO6	DDRHSSPI TX-FIFO Register 6	
DDRHSSPIn_TXFIFO7	DDRHSSPI TX-FIFO Register 7	
DDRHSSPIn_TXFIFO8	DDRHSSPI TX-FIFO Register 8	
DDRHSSPIn_TXFIFO9	DDRHSSPI TX-FIFO Register 9	
DDRHSSPIn_TXFIFO10	DDRHSSPI TX-FIFO Register 10	
DDRHSSPIn_TXFIFO11	DDRHSSPI TX-FIFO Register 11	
DDRHSSPIn_TXFIFO12	DDRHSSPI TX-FIFO Register 12	
DDRHSSPIn_TXFIFO13	DDRHSSPI TX-FIFO Register 13	
DDRHSSPIn_TXFIFO14	DDRHSSPI TX-FIFO Register 14	
DDRHSSPIn_TXFIFO15	DDRHSSPI TX-FIFO Register 15	
DDRHSSPIn_TXFIFO16	DDRHSSPI TX-FIFO Register 16	
DDRHSSPIn_TXFIFO17	DDRHSSPI TX-FIFO Register 17	
DDRHSSPIn_TXFIFO18	DDRHSSPI TX-FIFO Register 18	
DDRHSSPIn_TXFIFO19	DDRHSSPI TX-FIFO Register 19	
DDRHSSPIn_TXFIFO20	DDRHSSPI TX-FIFO Register 20	
DDRHSSPIn_TXFIFO21	DDRHSSPI TX-FIFO Register 21	5.20
DDRHSSPIn_TXFIFO22	DDRHSSPI TX-FIFO Register 22	
DDRHSSPIn_TXFIFO23	DDRHSSPI TX-FIFO Register 23	
DDRHSSPIn_RXFIFO0	DDRHSSPI RX-FIFO Register 0	5.21
DDRHSSPIn_RXFIFO1	DDRHSSPI RX-FIFO Register 1	
DDRHSSPIn_RXFIFO2	DDRHSSPI RX-FIFO Register 2	
DDRHSSPIn_RXFIFO3	DDRHSSPI RX-FIFO Register 3	
DDRHSSPIn_RXFIFO4	DDRHSSPI RX-FIFO Register 4	
DDRHSSPIn_RXFIFO5	DDRHSSPI RX-FIFO Register 5	
DDRHSSPIn_RXFIFO6	DDRHSSPI RX-FIFO Register 6	
DDRHSSPIn_RXFIFO7	DDRHSSPI RX-FIFO Register 7	
DDRHSSPIn_RXFIFO8	DDRHSSPI RX-FIFO Register 8	

Abbreviation	Register Name	See
DDRHSSPIn_RXFIFO9	DDRHSSPI RX-FIFO Register 9	
DDRHSSPIn_RXFIFO10	DDRHSSPI RX-FIFO Register 10	
DDRHSSPIn_RXFIFO11	DDRHSSPI RX-FIFO Register 11	
DDRHSSPIn_RXFIFO12	DDRHSSPI RX-FIFO Register 12	
DDRHSSPIn_RXFIFO13	DDRHSSPI RX-FIFO Register 13	
DDRHSSPIn_RXFIFO14	DDRHSSPI RX-FIFO Register 14	
DDRHSSPIn_RXFIFO15	DDRHSSPI RX-FIFO Register 15	
DDRHSSPIn_RXFIFO16	DDRHSSPI RX-FIFO Register 16	
DDRHSSPIn_RXFIFO17	DDRHSSPI RX-FIFO Register 17	
DDRHSSPIn_RXFIFO18	DDRHSSPI RX-FIFO Register 18	
DDRHSSPIn_RXFIFO19	DDRHSSPI RX-FIFO Register 19	
DDRHSSPIn_RXFIFO20	DDRHSSPI RX-FIFO Register 20	
DDRHSSPIn_RXFIFO21	DDRHSSPI RX-FIFO Register 21	
DDRHSSPIn_RXFIFO22	DDRHSSPI RX-FIFO Register 22	
DDRHSSPIn_RXFIFO23	DDRHSSPI RX-FIFO Register 23	
DDRHSSPIn_RDCSDC0	DDRHSSPI Read Command Sequencer Data/Control Register 0	5.22
DDRHSSPIn_RDCSDC1	DDRHSSPI Read Command Sequencer Data/Control Register 1	
DDRHSSPIn_RDCSDC2	DDRHSSPI Read Command Sequencer Data/Control Register 2	
DDRHSSPIn_RDCSDC3	DDRHSSPI Read Command Sequencer Data/Control Register 3	
DDRHSSPIn_RDCSDC4	DDRHSSPI Read Command Sequencer Data/Control Register 4	
DDRHSSPIn_RDCSDC5	DDRHSSPI Read Command Sequencer Data/Control Register 5	
DDRHSSPIn_RDCSDC6	DDRHSSPI Read Command Sequencer Data/Control Register 6	
DDRHSSPIn_RDCSDC7	DDRHSSPI Read Command Sequencer Data/Control Register 7	
DDRHSSPIn_RDCSDC8	DDRHSSPI Read Command Sequencer Data/Control Register 8	
DDRHSSPIn_RDCSDC9	DDRHSSPI Read Command Sequencer Data/Control Register 9	
DDRHSSPIn_RDCSDC10	DDRHSSPI Read Command Sequencer Data/Control Register 10	
DDRHSSPIn_RDCSDC11	DDRHSSPI Read Command Sequencer Data/Control Register 11	
DDRHSSPIn_MID	DDRHSSPI Module ID Register	5.23
DDRHSSPIn_CSPREFETCHADDR	DDRHSSPI Command Sequencer Prefetch Address Register	5.24
DDRHSSPIn_SDATASAMPLEPTCNT0	DDRHSSPI SDATA Center Clock Sample Point Register 0	5.25
DDRHSSPIn_SDATASAMPLEPTCNT1	DDRHSSPI SDATA Center Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTCNT2	DDRHSSPI SDATA Center Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTCNT3	DDRHSSPI SDATA Center Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTCNT4	DDRHSSPI SDATA Center Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTCNT5	DDRHSSPI SDATA Center Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTCNT6	DDRHSSPI SDATA Center Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTCNT7	DDRHSSPI SDATA Center Clock Sample Point Register 7	
DDRHSSPIn_SDATASAMPLEPTLFT0	DDRHSSPI SDATA Left Clock Sample Point Register 0	5.26
DDRHSSPIn_SDATASAMPLEPTLFT1	DDRHSSPI SDATA Left Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTLFT2	DDRHSSPI SDATA Left Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTLFT3	DDRHSSPI SDATA Left Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTLFT4	DDRHSSPI SDATA Left Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTLFT5	DDRHSSPI SDATA Left Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTLFT6	DDRHSSPI SDATA Left Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTLFT7	DDRHSSPI SDATA Left Clock Sample Point Register 7	
DDRHSSPIn_SDATASAMPLEPTRGH0	DDRHSSPI SDATA Right Clock Sample Point Register 0	5.27
DDRHSSPIn_SDATASAMPLEPTRGH1	DDRHSSPI SDATA Right Clock Sample Point Register 1	
DDRHSSPIn_SDATASAMPLEPTRGH2	DDRHSSPI SDATA Right Clock Sample Point Register 2	
DDRHSSPIn_SDATASAMPLEPTRGH3	DDRHSSPI SDATA Right Clock Sample Point Register 3	
DDRHSSPIn_SDATASAMPLEPTRGH4	DDRHSSPI SDATA Right Clock Sample Point Register 4	
DDRHSSPIn_SDATASAMPLEPTRGH5	DDRHSSPI SDATA Right Clock Sample Point Register 5	
DDRHSSPIn_SDATASAMPLEPTRGH6	DDRHSSPI SDATA Right Clock Sample Point Register 6	
DDRHSSPIn_SDATASAMPLEPTRGH7	DDRHSSPI SDATA Right Clock Sample Point Register 7	
DDRHSSPIn_DLP	DDRHSSPI Data Learning Pattern Register	5.28
DDRHSSPIn_DLPSAMPLESTATUS	DDRHSSPI Data Learning Pattern Sample Status Register	5.29
DDRHSSPIn_CSCFG	DDRHSSPI Command Sequencer Configuration Register	5.30
DDRHSSPIn_CSITIME	DDRHSSPI Command Sequencer Idle Time Register	5.31
DDRHSSPIn_CSAEXT	DDRHSSPI Command Sequencer Address Extension Register	5.32
DDRHSSPIn_CSPBUFFERCFG	DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register	5.33
DDRHSSPIn_CSPBUFFERSTATUS	DDRHSSPI Command Sequencer Prefetch Buffer Status Register	5.34

5.1. DDRHSSPI Module Control Register (DDRHSSPIn_MCTRL)

The DDRHSSPI Module Control Register controls the DDRHSSPI. It contains vital bits like the Module Enable bit, the Command Sequencer Mode Enable bit and Data Learning Pattern Enable bit.

The Software can enable/disable the DDRHSSPI operation by using this register.

REGISTER_NAME	DDRHSSPIn_MCTRL
OFFSET	0x0000
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	DLPEN	MES	read0	read0	CSEN	MEN
ACCESS_TYPE	R0,W0	R0,W0	R/W	R,WX	R0,W0	R0/W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:6] read0 : Reserved

[bit5] DLPEN : Data Learning Pattern Enable

This bit enables when the sampling point of receiving data is checked to use the command with DLP.

This bit shall be fixed to "0" when DDRHSSPIn_MID.MID is 0x00000001.

This bit is valid in DDR Mode of the Command Sequencer Mode.

bit	Description
0	DLP is disabled.
1	DLP is enabled.

[bit4] MES : Module Enable Status

This bit shows the status of the DDRHSSPI.

bit	Description
0	DDRHSSPI is completely disabled and it has entered the power saving mode.
1	DDRHSSPI is enabled.

[bit3:2] read0 : Reserved

[bit1] CSEN : Command Sequencer Mode Enable

This bit selects the operating mode(Direct Mode or Command Sequencer Mode).

bit	Description
0	Direct Mode is enabled. Command Sequencer Mode is disabled.
1	Command Sequencer Mode is enabled. Direct Mode is disabled.

[bit0] MEN : Module Enable

This bit enables DDRHSSPI operation.

bit	Description
0	DDRHSSPI is disabled. DDRHSSPI enters power saving mode.
1	DDRHSSPI is enabled.

After configuring the DDRHSSPI, Software must set this bit to "1", to enable DDRHSSPI in operating mode.

When Software resets this bit:

a) In Direct Mode: DDRHSSPI aborts current SPI transfer and Slave Select is released. After the Slave Select is released, it internally enters a power saving mode.

b) In Command Sequencer Mode: DDRHSSPI generates an unmapped memory access fault interrupt if any further access to memory mapped Serial Flash Memories is received. It aborts ongoing transaction on the Serial Interface and release Slave Select. After the Slave Select has been released, it internally enters a power saving mode.

5.2. DDRHSSPI Peripheral Communication Configuration Registers (DDRHSSPIn_PCC0-3)

The DDRHSSPI Peripheral Communication Configuration Registers 0-3 control the attributes related to the serial communication on Slave Select 0-3. The Software must initialize these registers with the attributes that match the communication attributes of the Serial Flash Memory that is to be interfaced on the corresponding Slave Select line 0-3 of DDRHSSPI. When DDRHSSPI is configured in Direct Mode or in Command Sequencer Mode, each of the 4 registers is used and must be same value.

Only DDRHSSPIn_PCC0 Register is described here. Other registers (i.e. DDRHSSPIn_PCC1-3) have same bit fields.

REGISTER_NAME	DDRHSSPIn_PCCi (i = 0~3)
OFFSET	0x0004 + i*4
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	SSELDEASRT[4]	SSELDEASRT[3]	SSELDEASRT[2]	SSELDEASRT[1]	SSELDEASRT[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	CDRS[3]	CDRS[2]	CDRS[1]	CDRS[0]	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	SS2CD[1]	SS2CD[0]	read0	read0	ACES	read0	read0
ACCESS_TYPE	R0/W0	R/W	R/W	R0/W0	R0,W0	R/W	R0/W0	R0/W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:21] read0 : Reserved

[bit20:16] SSELDEASRT[4:0] : Slave Selection Deassertion Timeout

It defines the time period after SSEL deassertion when a new SSEL assertion can not occur.

SSELDEASRT[4:0]	Description	
	SDR Mode	DDR Mode
00000	Setting this value is prohibited.	
00001	Deassertion time = 4 x SCLK cycle	Deassertion time = 3.5 x SCLK cycle
00010	Deassertion time = 5 x SCLK cycle	Deassertion time = 4.0 x SCLK cycle
...	...	
00111	Deassertion time = 10 x SCLK cycle	Deassertion time = 6.5 x SCLK cycle
...	...	
11111	Deassertion time = 34 x SCLK cycle	Deassertion time = 18.5 x SCLK cycle

[bit15:13] read0 : Reserved

[bit12:9] CDRS[3:0] : Clock Division Ratio Select for Peripheral

This field decides the Clock Division Ratio for the SCLK.

CDRS[3:0]	Description
0000	Divide by 2
0001	Divide by 4.
0010	Divide by 6.
...	...
0111	Divide by 16.
...	...
1111	Divide by 32.

[bit8:7] read0 : Reserved

[bit6:5] SS2CD[1:0] : Slave Select to Clock Delay

It defines a setup time for the Serial Flash Memory. By delaying the toggling of SCLK, DDRHSSPI delays the start of data transfer from the Slave Select active edge by a multiple of SCLK cycles.

The delay between the assertion of Slave Select and the first edge on the SCLK is given by:

(SS2CD + 0.50) x SCLK cycle [SDR mode]

(SS2CD + 0.75) x SCLK cycle [DDR mode]

When the Slave Select becomes active, the Serial Flash Memory has to prepare data transfer within the delay time defined by SS2CD bits.

Note:
SS2CD[1:0] = "00" must not be used when all the following conditions are met:

- *DDRHSSPIn_DMTRP.TRP[3:0] = "1011"*
- *DDRHSSPIn_DMTRP.DDRM = "1"*
- *DDRHSSPIn_DMFIFOCFG.TXCTRL = "1"*
- *DDRHSSPIn_TXFIFOx.TXDATA[12][10][9:8] = "1000"*

[bit4:3] read0 : Reserved

[bit2] ACES : Active Clock Edges are Same on Peripheral

This bit decides whether the active edges of the clock used for launching of data and for capturing of data are same or not. This bit takes effect both in Direct Mode and Command Sequencer Mode. This bit is only valid in SDR Mode.

bit	Description
0	Launching of data and capturing of data is done on alternate (i.e. opposite) edges of clock.
1	Launching of data and capturing of data is done on same edges of clock.

[bit1:0] read0 : Reserved

5.3. DDRHSSPI TX Interrupt Flag Register (DDRHSSPIn_TXF)

The DDRHSSPI TX Interrupt Flag Register indicates the status of the TX interrupt flags. These interrupt flags except TSSRS flag are set in Direct Mode only. When Command Sequencer Mode is selected, these flags except TSSRS flag are automatically cleared. The interrupt flags except TSSRS flag can be active also while the DDRHSSPI is disabled (DDRHSSPIn_MCTRL.MES = "0").

Software can enable these interrupts and wait for their assertion, or it can also use them in polling mode.

REGISTER_NAME	DDRHSSPIn_TXF
OFFSET	0x0014
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	TSSRS	TFMTS	TFLETS	read0	TFOS	TFES	TFFS
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:7] read0 : Reserved

[bit6] TSSRS : Slave Select Released

This interrupt flag indicates that the Slave Select line is released by the DDRHSSPI.

bit	Description
0	DDRHSSPI does not generate the event which SSEL signal is deasserted.
1	DDRHSSPI generates the event which SSEL signal is deasserted.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TSSRE.

This interrupt flag is cleared when DDRHSSPIn_TXC.TSSRC bit is written "1".

[bit5] TFMTS : TX-FIFO Fill Level is More Than Threshold

This interrupt flag indicates whether the TX-FIFO Fill Level is more than the configured TX-FIFO threshold value or not.

bit	Description
0	The TX-FIFO fill level is not more than the configured TX-FIFO threshold value.
1	The TX-FIFO fill level is more than the configured TX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.TXFTH.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFMTE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFMTC bit is written "1".

[bit4] TFLETS : TX-FIFO Fill Level is Less Than or Equal to Threshold

This interrupt flag indicates whether the TX-FIFO Fill Level is less than or equal to the configured TX-FIFO threshold value or not.

bit	Description
0	The TX-FIFO fill level is not less than or equal to the configured TX-FIFO threshold value.
1	The TX-FIFO fill level is less than or equal to the configured TX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.TXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.TXFTH.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFLETE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFLETC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit3] read0 : Reserved

[bit2] TFOS : TX-FIFO Overrun

This interrupt flag indicates that the TX-FIFO is overrun.

bit	Description
0	Any of DDRHSSPIn_TXFIFO0-23 Registers is not written by the Software when the TX-FIFO is full.
1	Any of DDRHSSPIn_TXFIFO0-23 Registers is written by the Software when the TX-FIFO is full.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFOE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFOC bit is written "1".

[bit1] TFES : TX-FIFO and Shift Register are Empty

This interrupt flag indicates whether the TX-FIFO and TX Shift Register are empty or not.

bit	Description
0	The TX-FIFO or the TX Shift Register (in SPI Core) are not empty.
1	The TX-FIFO and the TX Shift Register (in SPI Core) are empty. This bit is always set while this condition is satisfied.

This interrupt flag triggers the TX Interrupt Service Request if it is enabled in DDRHSSPIn_TXE.TFEE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFEC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit0] TFFS : TX-FIFO Full

This interrupt flag indicates whether the TX-FIFO is full or not.

bit	Description
0	The TX-FIFO is not full.
1	The TX-FIFO is full. This bit is always set while this condition is satisfied.

This interrupt flag triggers the TX Interrupt Service Request, if it is enabled in DDRHSSPIn_TXE.TFFE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_TXC.TFFC bit is written "1".

5.4. DDRHSSPI TX Interrupt Enable Register (DDRHSSPIn_TXE)

The DDRHSSPI TX Interrupt Enable Register decides whether the interrupt flags in DDRHSSPIn_TXF Register trigger the TX Interrupt Service Request or not.

The Software must enable these bits if it wants to wait for the assertion of the TX Interrupt Service Request.

REGISTER_NAME	DDRHSSPIn_TXE
OFFSET	0x0018
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	TSSRE	TFMTE	TFLETE	read0	TFOE	TFEE	TFFE
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:7] read0 : Reserved

[bit6] TSSRE : Slave Select Released Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TSSRS interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TSSRS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TSSRS interrupt flag triggers the TX Interrupt Service Request.

[bit5] TFMTE : TX-FIFO Fill Level is More Than Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFMTS interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TFMTS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFMTS interrupt flag triggers the TX Interrupt Service Request.

[bit4] TFLETE : TX-FIFO Fill Level is Less Than or Equal To Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFLETS interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TFLETS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFLETS interrupt flag triggers the TX Interrupt Service Request.

[bit3] read0 : Reserved

[bit2] TFOE : TX-FIFO Overrun Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFOS interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TFOS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFOS interrupt flag triggers the TX Interrupt Service Request.

[bit1] TFEE : TX-FIFO Empty Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFES interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TFES interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFES interrupt flag triggers the TX Interrupt Service Request.

[bit0] TFFE : TX-FIFO Full Interrupt Enable

This bit decides whether the DDRHSSPIn_TXF.TFFS interrupt flag is routed on TX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_TXF.TFFS interrupt flag does not trigger the TX Interrupt Service Request.
1	The DDRHSSPIn_TXF.TFFS interrupt flag triggers the TX Interrupt Service Request.

5.5. DDRHSSPI TX Interrupt Clear Register (DDRHSSPIn_TXC)

The DDRHSSPI TX Interrupt Clear Register is used to clear the interrupt flags in the DDRHSSPIn_TXF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_TXF Register.

REGISTER_NAME	DDRHSSPIn_TXC
OFFSET	0x001C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	TSSRC	TFMTC	TFLETC	read0	TFOC	TFEC	TFFC
ACCESS_TYPE	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

[bit31:7] read0 : Reserved

[bit6] TSSRC : Slave Select Released Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TSSRS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TSSRS interrupt flag

The read value is "0".

[bit5] TFMTC : TX-FIFO Fill Level More Than Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFMTS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFMTS interrupt flag.

The read value is "0".

[bit4] TFLETC : TX-FIFO Fill Level Less Than or Equal to Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFLETS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFLETS interrupt flag.

The read value is "0".

[bit3] read0 : Reserved

[bit2] TFOC : TX-FIFO Overrun Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFOS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFOS interrupt flag.

The read value is "0".

[bit1] TFEC : TX-FIFO Empty Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFES interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFES interrupt flag.

The read value is "0".

[bit0] TFFC : TX-FIFO Full Interrupt Clear

This bit is used to clear the DDRHSSPIn_TXF.TFFS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_TXF.TFFS interrupt flag.

The read value is "0".

5.6. DDRHSSPI RX Interrupt Flag Register (DDRHSSPIn_RXF)

The DDRHSSPI RX Interrupt Flag Register indicates the status of the RX interrupt flags. The interrupt flags can be active also while the DDRHSSPI is disabled (DDRHSSPIn_MCTRL.MES = "0").

Software can enable these interrupts and wait for their assertion, or it can also use them in polling mode.

REGISTER_NAME	DDRHSSPIn_RXF
OFFSET	0x0020
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	TEST
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	RX,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERR	RSSRS	RFMTS	RFLETS	RFUS	read0	RFES	RFFS
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:9] read0 : Reserved

[bit8] TEST : Test

This is a test flag. The read value is not defined. Writing to this bit does not influence other functions.

[bit7] DLPERR : Data Learning Pattern Reception Error

This bit is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

This interrupt flag indicates that the one or more SDATA input ports received incorrect DLP from Flash Memory which is caused by wrong sample clock settings in one or more registers among DDRHSSPIn_SDATASAMPLEPTCNT0-7, DDRHSSPIn_SDATASAMPLEPTLFT0-7 and DDRHSSPIn_SDATASAMPLEPTRGH0-7.

bit	Description
0	DDRHSSPIn_DLP.DLP matches the received DLP data, or DDRHSSPI does not check the DLP data.
1	DDRHSSPIn_DLP.DLP does not match the received DLP data.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.DLPERRE.

DDRHSSPIn_DLPSAMPLESTATUS Register has information which SDATA input pins detected problems with data sampling. This flag can be used only in DDR Mode of the Command Sequencer Mode.

This interrupt flag is cleared when DDRHSSPIn_RXC.DLPERRC bit is written "1".

[bit6] RSSRS : Slave Select Released

This interrupt flag indicates that the Slave Select is released by the DDRHSSPI.

bit	Description
0	DDRHSSPI does not generate the event which SSEL signal is deasserted.
1	DDRHSSPI generates the event which SSEL signal is deasserted.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RSSRE.

This interrupt flag is cleared when DDRHSSPIn_RXC.RSSRC bit is written "1".

[bit5] RFMTS : RX-FIFO Fill Level is More Than Threshold

This interrupt flag indicates whether the RX-FIFO Fill Level is more than the configured RX-FIFO threshold value or not.

bit	Description
0	The RX-FIFO fill level is not more than the configured RX-FIFO threshold value.
1	The RX-FIFO fill level is more than the configured RX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is greater than DDRHSSPIn_DMFIFOCFG.RXFTH.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.RFMTE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFMTC bit is written "1".

[bit4] RFLETS : RX-FIFO Fill Level is Less Than or Equal to Threshold

This interrupt flag indicates whether the RX-FIFO Fill Level is less than or equal to the configured RX-FIFO threshold value or not.

bit	Description
0	The RX-FIFO fill level is not less than or equal to the configured RX-FIFO threshold value.
1	The RX-FIFO fill level is less than or equal to the configured RX-FIFO threshold value. This bit is always set while this condition is satisfied.

i.e. DDRHSSPIn_DMFIFOSTATUS.RXFLEVEL is less than or equal to DDRHSSPIn_DMFIFOCFG.RXFTH.

This interrupt flag triggers the RX Interrupt Service Request, if it is enabled in DDRHSSPIn_RXE.RFLETE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFLETC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit3] RFUS : RX-FIFO Underrun

This interrupt flag indicates that the RX-FIFO is underrun.

bit	Description
0	Any of DDRHSSPIn_RXFIFO0-23 Registers is not read by the Software when the RX-FIFO is empty.
1	Any of DDRHSSPIn_RXFIFO0-23 Registers is read by the Software when the RX-FIFO is empty.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFUE.

This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFUC bit is written "1".

Note:

- *This flag is not set when the DAP controller reads the DDRHSSPIn_RXFIFO0-23 Registers while the RX-FIFO is empty.*

[bit2] read0 : Reserved

[bit1] RFES : RX-FIFO Empty

This interrupt flag indicates whether the RX-FIFO is empty or not.

bit	Description
0	The RX-FIFO is not empty.
1	The RX-FIFO is empty. This bit is always set while this condition is satisfied.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFEE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFEC bit is written "1".

Note:

- *After System Reset, this bit is set to "1".*

[bit0] RFFS : RX-FIFO Full

This interrupt flag indicates whether the RX-FIFO is full or not.

bit	Description
0	The RX-FIFO is not full.
1	The RX-FIFO is full. This bit is always set while this condition is satisfied.

This interrupt flag triggers the RX Interrupt Service Request if it is enabled in DDRHSSPIn_RXE.RFFE. This interrupt flag shall be used in Direct Mode only. If Command Sequencer Mode is selected, this flag is automatically cleared.

This interrupt flag is cleared when DDRHSSPIn_RXC.RFFC bit is written "1".

5.7. DDRHSSPI RX Interrupt Enable Register (DDRHSSPIn_RXE)

The DDRHSSPI RX Interrupt Enable Register decides whether the interrupt flags in DDRHSSPIn_RXF Register trigger the RX Interrupt Service Request or not.

The Software must enable these bits if it wants to wait for the assertion of the RX Interrupt Service Request.

REGISTER_NAME	DDRHSSPIn_RXE
OFFSET	0x0024
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERRE	RSSRE	RFMTE	RFLETE	RFUE	read0	RFEE	RFFE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:8] read0 : Reserved

[bit7] DLPERRE : Data Learning Pattern Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.DLPERR interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.DLPERR interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.DLPERR interrupt flag triggers the RX Interrupt Service Request.

[bit6] RSSRE : Slave Select Released Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RSSRS interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RSSRS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RSSRS interrupt flag triggers the RX Interrupt Service Request.

[bit5] RFMTE : RX-FIFO Fill Level is More Than Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFMTS interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RFMTS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFMTS interrupt flag triggers the RX Interrupt Service Request.

[bit4] RFLETE : RX-FIFO Fill Level is Less Than or Equal To Threshold Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFLETS interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RFLETS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFLETS interrupt flag triggers the RX Interrupt Service Request.

[bit3] RFUE : RX-FIFO Underrun Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFUS interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RFUS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFUS interrupt flag triggers the RX Interrupt Service Request.

[bit2] read0 : Reserved

[bit1] RFEE : RX-FIFO Empty Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFES interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RFES interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFES interrupt flag triggers the RX Interrupt Service Request.

[bit0] RFFE : RX-FIFO Full Interrupt Enable

This bit decides whether the DDRHSSPIn_RXF.RFFS interrupt flag is routed on RX Interrupt Service Request or not.

bit	Description
0	The DDRHSSPIn_RXF.RFFS interrupt flag does not trigger the RX Interrupt Service Request.
1	The DDRHSSPIn_RXF.RFFS interrupt flag triggers the RX Interrupt Service Request.

5.8. DDRHSSPI RX Interrupt Clear Register (DDRHSSPIn_RXC)

The DDRHSSPI RX Interrupt Clear Register is used to clear the interrupt flags in the DDRHSSPIn_RXF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_RXF Register.

REGISTER_NAME	DDRHSSPIn_RXC
OFFSET	0x0028
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLPERRC	RSSRC	RFMTC	RFLETC	RFUC	read0	RFEC	RFFC
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

[bit31:8] read0 : Reserved

[bit7] DLPERRC : DLP Error Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.DLPERR interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.DLPERR interrupt flag.

The read value is "0".

[bit6] RSSRC : Slave Select Released Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RSSRS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RSSRS interrupt flag.

The read value is "0".

[bit5] RFMTC : RX-FIFO Fill Level More Than Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFMTS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFMTS interrupt flag.

The read value is "0".

[bit4] RFLETC : RX-FIFO Fill Level Less Than or Equal to Threshold Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFLETS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFLETS interrupt flag.

The read value is "0".

[bit3] RFUC : RX-FIFO Underrun Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFUS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFUS interrupt flag.

The read value is "0".

[bit2] read0 : Reserved

[bit1] RFEC : RX-FIFO Empty Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFES interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFES interrupt flag.

The read value is "0".

[bit0] RFFC : RX-FIFO Full Interrupt Clear

This bit is used to clear the DDRHSSPIn_RXF.RFFS interrupt flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_RXF.RFFS interrupt flag.

The read value is "0".

5.9. DDRHSSPI Fault Status Flag Register (DDRHSSPIn_FAULTF)

The DDRHSSPI Fault Status Flag Register indicates the status of the FAULT flag.

Once a fault occurs, the Software needs to take a corrective action. Whenever one of these flags gets set, the corresponding bus access returns with a bus error response.

REGISTER_NAME	DDRHSSPIn_FAULTF
OFFSET	0x002C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	DLPFS	read0	DRCBSFS	DWCBSFS	PVFS	read0	UMAFS
ACCESS_TYPE	R0,WX	R,WX	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:7] read0 : Reserved

[bit6] DLPFS : DLP Error Fault

This status flag indicates that the DLP error fault has occurred on the data received on Serial Interface.

bit	Description
0	DDRHSSPI does not detect the DLP error.
1	DDRHSSPI detects the DLP error.

The DLP error fault occurs if the DDRHSSPI receives 8 bit DLP value that is not matched DDRHSSPIn_DLP.DLP.

This status flag is cleared when DDRHSSPIn_FAULTC.DLPFC bit is written "1".

[bit5] read0 : Reserved

[bit4] DRCBSFS : DMA Read Channel Block Size Fault

This status flag indicates that the block size fault has occurred on DMA read channel.

bit	Description
0	DDRHSSPI does not detect the DMA read channel block size fault.
1	DDRHSSPI detects the DMA read channel block size fault.

The DMA read channel block size fault occurs if the DDRHSSPI RX block counter is 0 and there is a valid read access to the RX-FIFO (except for from the DAP controller).
This status flag is cleared when DDRHSSPIn_FAULTC.DRCBSFC bit is written "1".

[bit3] DWCBSFS : DMA Write Channel Block Size Fault

This status flag indicates that the block size fault has occurred on DMA write channel.

bit	Description
0	DDRHSSPI does not detect the DMA write channel block size fault.
1	DDRHSSPI detects the DMA write channel block size fault.

The DMA write channel block size fault occurs if the DDRHSSPI TX block counter is 0 and there is a valid write access to the TX-FIFO.

This status flag is cleared when DDRHSSPIn_FAULTC.DWCBSFC bit is written "1".

[bit2] PVFS : Protection Violation Fault

This status flag indicates that a protection violation fault has occurred.

bit	Description
0	DDRHSSPI does not detect the protection violation fault.
1	DDRHSSPI detects the protection violation fault.

The protection violation fault includes following cases:

- Access to a reserved register except some reserved registers.
- Write access to a read-only register.

This status flag is cleared when DDRHSSPIn_FAULTC.PVFC bit is written "1".

[bit1] read0 : Reserved

[bit0] UMAFS : Unmapped Memory Access Fault

This status flag indicates that an unmapped memory access fault has occurred.

This bit is set by DDRHSSPI when any of the following event occurs.

- In Direct Mode (i.e. DDRHSSPIn_MCTRL.CSEN = "0"), a System Bus access within the 256 MB address range starting from the DDRHSSPI base address is detected.
- In Command Sequencer Mode (i.e. DDRHSSPIn_MCTRL.CSEN = "1"), an access to a Serial Flash Memory which is not enabled (in DDRHSSPIn_CSCFG.SSEL0EN-SSEL3EN bits) is detected.
- In Command Sequencer Mode (i.e. DDRHSSPIn_MCTRL.CSEN = "1"), a System Bus access to a memory location which is outside the memory range being mapped onto the four Slave Selects (configured through the DDRHSSPIn_CSCFG.MSEL) is detected.
- While the module is disabled (i.e. DDRHSSPIn_MCTRL.MES = "0"), an access to a mapped memory is detected.

This status flag is cleared when DDRHSSPIn_FAULTC.UMAFC bit is written "1".

5.10. DDRHSSPI Fault Status Clear Register (DDRHSSPIn_FAULTC)

The DDRHSSPI Fault Status Clear Register is used to clear the status flags in the DDRHSSPIn_FAULTF Register.

By writing "1" to a bit in this register, the Software can clear the corresponding flag in the DDRHSSPIn_FAULTF Register.

REGISTER_NAME	DDRHSSPIn_FAULTC
OFFSET	0x0030
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	DLPFC	read0	DRCBSFC	DWCBSFC	PVFC	read0	UMAFC
ACCESS_TYPE	R0,W0	R0,W	R0,W0	R0,W	R0,W	R0,W	R0,W0	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

[bit31:7] read0 : Reserved

[bit6] DLPFC : DLP Error Fault Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DLPFS status flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DLPFS interrupt flag.

The read value is "0".

[bit5] read0 : Reserved

[bit4] DRCBSFC : DMA Read Channel Block Size Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DRCBSFS status flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DRCBSFS interrupt flag.

The read value is "0".

[bit3] DWCBSFC : DMA Write Channel Block Size Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.DWCBSFS status flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.DWCBSFS status flag.

The read value is "0".

[bit2] PVFC : Protection Violation Fault Status Clear

This bit is used to clear the DDRHSSPIn_FAULTF.PVFS status flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.PVFS status flag.

The read value is "0".

[bit1] read0 : Reserved

[bit0] UMAFC : Unmapped Memory Access Fault Interrupt Clear

This bit is used to clear the DDRHSSPIn_FAULTF.UMAFS status flag.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit clears the DDRHSSPIn_FAULTF.UMAFS status flag.

The read value is "0".

5.11. DDRHSSPI Direct Mode Configuration Register (DDRHSSPIn_DMCFG)

The DDRHSSPI Direct Mode Configuration Register configures the following operational parameter of DDRHSSPI:

- Byte Counter Mode of Slave Select Deassertion

This register is used only when DDRHSSPI is in Direct Mode.

REGISTER_NAME	DDRHSSPIn_DMCFG
OFFSET	0x0034
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	SSDC	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7:2] read0 : Reserved

[bit1] SSDC : Slave Select Deassertion Control

The SSDC bit decides how the Slave Select is deasserted:

bit	Description
0	This is prohibited when DDRHSSPI is operated in Direct Mode.
1	Byte Counter Mode. DDRHSSPIn_DMBCC.BCC is used to decide when to deassert the Slave Select.

[bit0] read0 : Reserved

5.12. DDRHSSPI DMA Enable Register (DDRHSSPIn_DMAEN)

The DDRHSSPI DMA Enable Register can be used by the Software, for enabling/disabling of the DMA Service Requests generated by DDRHSSPI.

This register is used in Direct Mode only.

REGISTER_NAME	DDRHSSPIn_DMAEN
OFFSET	0x0035
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	TXDMAEN	RXDMAEN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0/W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7:2] read0 : Reserved

[bit1] TXDMAEN : TX DMA Enable

This bit enable or disable the TX DMA channel (DMA write channel side).

bit	Description
0	TX DMA channel is disabled.
1	TX DMA channel is enabled.

This bit is valid in Direct Mode.

[bit0] RXDMAEN : RX DMA Enable

This bit enable or disable the RX DMA channel (DMA read channel side).

bit	Description
0	RX DMA channel is disabled.
1	RX DMA channel is enabled.

This bit is valid in Direct Mode.

5.13. DDRHSSPI Direct Mode Start Register (DDRHSSPIn_DMSTART)

The DDRHSSPI Direct Mode Start Register can be used by the Software, for triggering the start of the serial transfer.

This register is used only in Direct Mode.

REGISTER_NAME	DDRHSSPIn_DMSTART
OFFSET	0x0038
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	START
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	0

[bit7:1] read0 : Reserved

[bit0] START : Start Transfer

This bit is a trigger to start a new serial transfer in Direct Mode.

bit	Description
0	Writing "0" to this bit has no effect.
1	Writing "1" to this bit sets this bit. This bit can be only set by Software and it is cleared by Hardware.

DDRHSSPI resets this bit to "0" in any of following conditions:

- Serial transfer ends.
- Module is disabled (DDRHSSPIn_MCTRL.MES="0")
- DDRHSSPI is switched from Direct Mode to Command Sequencer Mode.

Writing "1" to this bit when the bit is already set to "1" has no effect on the current serial transfer (in any cases).

5.14. DDRHSSPI Direct Mode Peripheral Select Register (DDRHSSPIn_DMPSEL)

The DDRHSSPI Direct Mode Peripheral Select Register can be used by the Software to select one of 4 Slave Selects for initiating the serial transfer.

This register is used only when DDRHSSPI is in Direct Mode.

REGISTER_NAME	DDRHSSPIn_DMPSEL
OFFSET	0x003A
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	PSEL[1]	PSEL[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7:2] read0 : Reserved

[bit1:0] PSEL[1:0] : Peripheral Select

The PSEL bits decide which of the 4 Slave Selects in oDDRHSSPIn_SSEL3-0 is asserted for the current serial transfer.

PSEL[1:0]	Description
00	oDDRHSSPIn_SSEL0 is asserted.
01	oDDRHSSPIn_SSEL1 is asserted.
10	oDDRHSSPIn_SSEL2 is asserted.
11	oDDRHSSPIn_SSEL3 is asserted.

5.15. DDRHSSPI Direct Mode Transfer Protocol Register (DDRHSSPIn_DMTRP)

The DDRHSSPI Direct Mode Transfer Protocol Register configures the transfer protocol of the serial transfer.

This register is used only when DDRHSSPI is in Direct Mode.

REGISTER_NAME	DDRHSSPIn_DMTRP
OFFSET	0x003B
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	DDRM	read0	TRP[3]	TRP[2]	TRP[1]	TRP[0]
ACCESS_TYPE	R0,W0	R0/W0	R/W	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7:6] read0 : Reserved

[bit5] DDRM : DDR Mode

This bit indicates whether DDRHSSPI is in DDR Mode or SDR Mode, in Direct Mode.

bit	Description
0	DDRHSSPI is in Single Data Rate (SDR) Mode.
1	DDRHSSPI is in Dual Data Rate (DDR) Mode.

[bit4] read0 : Reserved

[bit3:0] TRP[3:0] : Transfer Protocol

Bits TRP[3:2] indicate the Duplex Configuration : TX-and-RX or TX-Only.

Bits TRP[1:0] indicate the using protocol : Legacy, Quad or Octal.

TRP[3:0]	Description
0000	TX-and-RX in Legacy Mode is configured.
0011	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. TX-and-RX in Octal Mode is configured.
1000	TX-Only in Legacy Mode is configured.
1010	TX-Only in Quad Mode is configured.
1011	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. TX-Only in Octal Mode is configured.

All other combinations are RESERVED and prohibited.

5.16. DDRHSSPI Byte Count Control Register (DDRHSSPIn_DMBCC)

The DDRHSSPI Byte Count Control Register configures the number of bytes that would be transferred in the serial transfer.

This register is used in Direct Mode only.

REGISTER_NAME	DDRHSSPIn_DMBCC
OFFSET	0x003C
ACCESS_SIZE	H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	BCC[15]	BCC[14]	BCC[13]	BCC[12]	BCC[11]	BCC[10]	BCC[9]	BCC[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	BCC[7]	BCC[6]	BCC[5]	BCC[4]	BCC[3]	BCC[2]	BCC[1]	BCC[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit15:0] BCC[15:0] : Byte Count Control

This field is used by DDRHSSPI in Direct Mode and when DDRHSSPIn_DMCFG.SSDC="1".

BCC must be programmed by the Software with the number of bytes to be transmitted/received/both.

The value in this field is loaded in a down-counter immediately after it is written in this register, and the counter is decremented when a byte is serially transferred. DDRHSSPI completes the transaction and deasserts the Slave Select when this down-counter reaches zero.

5.17. DDRHSSPI Byte Count Status Register (DDRHSSPIn_DMBCS)

The DDRHSSPI Byte Count Status Register is a read-only register, which can be used by the Software, to know how many bytes are yet to be transferred, in the current serial transfer.

This register is valid only when DDRHSSPI is configured so that Byte Counter Mode is selected (i.e. if DDRHSSPIn_DMCFG.SSDC="1" in Direct Mode).

REGISTER_NAME	DDRHSSPIn_DMBCS
OFFSET	0x003E
ACCESS_SIZE	H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	BCS[15]	BCS[14]	BCS[13]	BCS[12]	BCS[11]	BCS[10]	BCS[9]	BCS[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	BCS[7]	BCS[6]	BCS[5]	BCS[4]	BCS[3]	BCS[2]	BCS[1]	BCS[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit15:0] BCS[15:0] : Byte Count Status

This read-only field is valid only when DDRHSSPI is in Direct Mode and when DDRHSSPIn_DMCFG.SSDC="1".

BCS indicates the number of bytes in the current serial transfer that are not yet serially transmitted/received/both. Counter is updated with a new value only when current byte is considered to be transferred i.e. when last bit has been sent to memory or last bit from the memory has been received and put into FIFO.

For each FIFO entry where the number of cycles is less than a number of complete bytes, the BCS will be decremented by upper rounded value. For each FIFO entry where the number of cycles is less than a byte, the BCS will be decremented by one.

For FIFO entries where the number of cycles is more than one byte and less than two bytes, then the BCS will be decremented by two etc.

5.18. DDRHSSPI Direct Mode FIFO Status Register (DDRHSSPIn_DMFIFOSTATUS)

The DDRHSSPI Direct Mode FIFO Status Register contains the status bits like current fill-level of the TX/RX-FIFOs and whether the TX/RX path is active/idle i.e. Slave Select is asserted. This register is used only in Direct Mode.

REGISTER_NAME	DDRHSSPIn_DMFIFOSTATUS
OFFSET	0x0040
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	SSACTIVE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	TXF LEVEL[4]	TXF LEVEL[3]	TXF LEVEL[2]	TXF LEVEL[1]	TXF LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	RXF LEVEL[4]	RXF LEVEL[3]	RXF LEVEL[2]	RXF LEVEL[1]	RXF LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:17] read0 : Reserved

[bit16] SSACTIVE : Slave Select Active

This flag indicates whether a Slave Select is active.

bit	Description
0	Slave Select is not active.
1	Slave Select is active.

[bit15:13] read0 : Reserved

[bit12:8] TXFLEVEL[4:0] : Current Fill Level of TX-FIFO

This field indicates the current fill level of the TX-FIFO.

[bit7:5] read0 : Reserved

[bit4:0] RXFLEVEL[4:0] : Current Fill Level of RX-FIFO

This field indicates the current fill level of the RX-FIFO.

5.19. DDRHSSPI Direct Mode FIFO Configuration Register (DDRHSSPIn_DMFIFOCFG)

The DDRHSSPI FIFO Configuration Register configures the operation of the TX-FIFO and the RX-FIFO. The Software can configure the FIFO threshold levels and the FIFO width.

The Software can also initialize the FIFOs using the TXFLSH and RXFLSH bits in this register.

REGISTER_NAME	DDRHSSPIn_DMFIFOCFG
OFFSET	0x0044
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	TXFLSH	RXFLSH	TXCTRL	FWIDTH[1]	FWIDTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	TXFTH[4]	TXFTH[3]	TXFTH[2]	TXFTH[1]	TXFTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	RXFTH[4]	RXFTH[3]	RXFTH[2]	RXFTH[1]	RXFTH[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	1	1	1
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:21] read0 : Reserved

[bit20] TXFLSH : TX-FIFO Flush

This bit can be used by the Software to flush the TX-FIFO and TX Shift Register.

bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the TX-FIFO and TX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_DMFIFOSTATUS.SSACTIVE bit is "1".

The read value is "0".

[bit19] RXFLSH : RX-FIFO Flush

This register can be used by the Software to flush the RX-FIFO and RX Shift Register.

bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the RX-FIFO and RX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_DMFIFOSTATUS.SSACTIVE bit is "1".

This read value is "0".

[bit18] TXCTRL : TXCTRL bit to be written to TX-FIFO

When one of the DDRHSSPIn_TXFIFO0-23 Registers is written, the 33rd bit (TXCTRL bit) in TX-FIFO word takes this value.

Before writing to one of the DDRHSSPIn_TXFIFO0-23 Registers, the Software must update this bit, to control the next entry of TX-FIFO. Please refer to Section 4.1.3 for details of TXCTRL.

[bit17:16] FWIDTH[1:0] : FIFO Width

This field indicates the FIFO Width. Depending on the configured width of the FIFO, the available size of the Shift-Register in the SPI-Core also changes.

FWIDTH[1:0]	Description
00	TX-FIFO, RX-FIFO and Shift-Register are 8-bit wide.
01	TX-FIFO, RX-FIFO and Shift-Register are 16-bit wide.
11	TX-FIFO, RX-FIFO and Shift-Register are 32-bit wide.

All other combinations are RESERVED and prohibited.

[bit15:13] read0 : Reserved

[bit12:8] TXFTH[4:0] : TX-FIFO Threshold Level

Software must program this field with the threshold level of the TX-FIFO. The Maximum allowed value is 24 that is equal to TX-FIFO depth.

[bit7:5] read0 : Reserved

[bit4:0] RXFTH[4:0] : RX-FIFO Threshold Level

Software must program this field with the threshold level of the RX-FIFO. The Maximum allowed value is 24 that is equal to RX-FIFO depth.

5.20. DDRHSSPI TX-FIFO Registers (DDRHSSPIn_TXFIFO0-23)

The DDRHSSPI TX-FIFO Registers are used to push the data into the TX-FIFO. There are 24 registers for the input of the TX-FIFO, and they are all consecutively placed in the register-map. Each of these 24 registers is identical in function and fields. This is because the System Bus protocol does not support burst transfers to the same address. Only DDRHSSPIn_TXFIFO0 Register is described here. Other registers (i.e. DDRHSSPIn_TXFIFO1-23) have same bit fields.

Only 8-bit, 16-bit or 32-bit accesses are allowed to these registers. A write access to these registers by the Software pushes 33-bits into the TX-FIFO. This 33rd bit corresponds to the DDRHSSPIn_DMFIFOCFG. TXCTRL bit. These registers are used only in Direct Mode.

REGISTER_NAME	DDRHSSPIn_TXFIFOi (i = 0~23)
OFFSET	0x0048 + i*4
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TXDATA[31]	TXDATA[30]	TXDATA[29]	TXDATA[28]	TXDATA[27]	TXDATA[26]	TXDATA[25]	TXDATA[24]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TXDATA[23]	TXDATA[22]	TXDATA[21]	TXDATA[20]	TXDATA[19]	TXDATA[18]	TXDATA[17]	TXDATA[16]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TXDATA[15]	TXDATA[14]	TXDATA[13]	TXDATA[12]	TXDATA[11]	TXDATA[10]	TXDATA[9]	TXDATA[8]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TXDATA[7]	TXDATA[6]	TXDATA[5]	TXDATA[4]	TXDATA[3]	TXDATA[2]	TXDATA[1]	TXDATA[0]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

[bit31:0] TXDATA[31:0] : TX-FIFO Data

Writing to this 32-bit register pushes the data into the next location of TX-FIFO, and increments the write pointer of TX-FIFO. In addition, the DDRHSSPIn_DMFIFOCFG.TXCTRL bit is internally written as the 33rd bit of a TX-FIFO entry.

When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="11"), only 32-bit access is possible.

When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="01"), both 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.

When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), then 8-bit, 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 8 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO, but only the lower 8 bits are valid.

c) 8-bit access with any alignment shall access the lower 8 bits of the FIFO.

Other conditions (e.g. 8-bit write access in 32-bit wide setting) will cause a bus error response.

While the TX-FIFO is full, a write access to this register pushes the new data into the TX-FIFO and triggers a TX-FIFO overrun(DDRHSSPIn_TXF.TFOS) event. When the TX-FIFO overrun condition occurs, the integrity of the data transmitted over the Serial Interface is not guaranteed. Before writing to this register, the Software must ensure that the TX-FIFO is not full to avoid an overrun.

5.21. DDRHSSPI RX-FIFO Registers (DDRHSSPIn_RXFIFO0-23)

The DDRHSSPI RX-FIFO Registers are used to pop the data out of the RX-FIFO. There are 24 registers for the output of the RX-FIFO, and they are all consecutively placed in the register-map. Each of these 24 registers is identical in function and fields. This is because the System Bus protocol does not support burst transfers to the same address. Only DDRHSSPIn_RXFIFO0 Register is described here. Other registers (i.e. DDRHSSPIn_RXFIFO1-23) have same bit fields.

8-bit, 16-bit and 32-bit read accesses are allowed to these registers. By reading these registers, the Software can pop the data out of the RX-FIFO. These registers are used in Direct Mode.

REGISTER_NAME	DDRHSSPIn_RXFIFOi (i = 0~23)
OFFSET	0x00A8 + i*4
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RXDATA[31]	RXDATA[30]	RXDATA[29]	RXDATA[28]	RXDATA[27]	RXDATA[26]	RXDATA[25]	RXDATA[24]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RXDATA[23]	RXDATA[22]	RXDATA[21]	RXDATA[20]	RXDATA[19]	RXDATA[18]	RXDATA[17]	RXDATA[16]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RXDATA[15]	RXDATA[14]	RXDATA[13]	RXDATA[12]	RXDATA[11]	RXDATA[10]	RXDATA[9]	RXDATA[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RXDATA[7]	RXDATA[6]	RXDATA[5]	RXDATA[4]	RXDATA[3]	RXDATA[2]	RXDATA[1]	RXDATA[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	N	N	N	N	N	N	N	N

[bit31:0] RXDATA[31:0] : RX-FIFO Data

Reading this register returns a word of data from the RX-FIFO location pointed by the RX-FIFO read pointer. After a read access to this register, the RX-FIFO read pointer is incremented, if the read cycle was initiated by the System Bus master (not the DAP controller). If the DAP controller reads this register, the RX-FIFO read pointer is not incremented.

When the Shift Register width is 32 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="11"), only 32-bit access is possible.

When the Shift Register width is 16 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="01"), both 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 16 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO.

When the Shift Register width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), then 8-bit, 16-bit and 32-bit access are possible.

a) 32-bit access shall access the full width of FIFO, but only the lower 8 bits are valid.

b) 16-bit access with any alignment shall access the lower 16 bits of the FIFO, but only the lower 8 bits are valid.

c) 8-bit access with any alignment shall access the lower 8 bits of the FIFO.

Other conditions (e.g. 8-bit write access in 32-bit wide setting) will cause a bus error response.

When the bus width to access the FIFO is greater than the width of FIFO, the upper redundant bits of the read data on the System Bus are filled with "0". For example, if configured FIFO width is 8 bits (DDRHSSPIn_DMFIFOCFG.FWIDTH="00"), and the read access of the FIFO is on 32 bits, then

- Only the bits RXDATA[7:0] are valid, AND
- The other bits RXDATA[31:8] are all "0".

The Software must not use any data from the unused most significant bits.

While the RX-FIFO is empty, a read access to this register pops invalid data out of the RX-FIFO. A RX-FIFO underrun interrupt (DDRHSSPIn_RXF.RFUS) event is triggered if the read cycle was initiated by the System Bus master other than the DAP controller. If the DAP controller reads this register while the RX-FIFO is empty, the DDRHSSPIn_RXF.RFUS flag is not set.

5.22. DDRHSSPI Read Command Sequencer Data/Control Registers (DDRHSSPIn_RDCSDC0-11)

The DDRHSSPI Read Command Sequencer Data/Control Registers 0-11 are the list of Data/Control registers which configure the phases of the serial transaction generated by the Command Sequencer for Memory Read operations. These registers are used only in Command Sequencer Mode.

Only the DDRHSSPIn_RDCSDC0 Register is explained here. Other registers have identical fields.

REGISTER_NAME	DDRHSSPIn_RDCSDCi (i = 0~11)
OFFSET	0x0108 + i*2
ACCESS_SIZE	H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RDC SDATA[7]	RDC SDATA[6]	RDC SDATA[5]	RDC SDATA[4]	RDC SDATA[3]	RDC SDATA[2]	RDC SDATA[1]	RDC SDATA[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	DEC
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit15:8] RDCSDATA[7:0] : Command Sequencer Data or Control byte for Memory-Read Transactions

This field contains either a command code or a control code according to the setting of DEC bit.

- When DEC bit is "0":
  The RDCSDATA contains the 8-bit data to be transmitted on the Serial Interface.
- When DEC bit is "1":
  The RDCSDATA[7:0] is decoded as follows.
  RDCSDATA[2:0] = "000": Transmit address bits [7:0] of the Serial Flash Memory address.
  RDCSDATA[2:0] = "001": Transmit address bits [15:8] of the Serial Flash Memory address.
  RDCSDATA[2:0] = "010": Transmit address bits [23:16] of the Serial Flash Memory address.
  RDCSDATA[2:0] = "011": Transmit address bits [31:24] of the Serial Flash Memory address.
  RDCSDATA[2:0] = "100": Dummy cycle(s) on the SDATA output lines for RDCSDATA [7:3] number of SCLK cycles.
  - RDCSDATA[7:3]="00000" -> Dummy cycle for 1 SCLK cycle.
  - RDCSDATA[7:3]="00001" -> Dummy cycles for 2 SCLK cycles.

  ....

  RDCSDATA[2:0] = "111": End of list.
  All other values of RDCSDATA[2:0] are reserved and must not be used.

[bit7:1] read0 : Reserved

[bit0] DEC : Decode

This bit controls whether to decode RDCSDATA[2:0].

bit	Description
0	Transmit the RDCSDATA[7:0] as it is.
1	Decode the RDCSDATA[2:0] to decide the further action.

Notes:

- *The Command Sequencer Mode can use only a Continuous Read Command Sequence of the Serial FLASH Memory.*
- *The transmitted address bits of the Serial Flash Memory address must be registered in the DDRHSSPIn_RDCSDC0 Register.*

5.23. DDRHSSPI Module ID Register (DDRHSSPIn_MID)

This is a read-only register with a unique Module Identification Number which identifies the version of the DDRHSSPI used in the MCU.

REGISTER_NAME	DDRHSSPIn_MID
OFFSET	0x0120
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0 or 1	0 or 1
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0 or 1
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:0] MID[31:0] : Module ID

This read-only register gives the unique module identification number of DDRHSSPI.

The unique module ID number identifies the version of the DDRHSSPI used in the MCU.

MID[31:0]	Description
0x00000001	This version has limitations as below. − Octal protocol (Dual Quad and Dual Legacy) is not supported. ➢ DDRHSSPIn_DMTRP.TRP[1:0] shall not be "11". ➢ DDRHSSPIn_CSCFG.MBM[1:0] shall not be "11". − Sample Point Control function is not supported. ➢ DDRHSSPIn_SDATASAMPLEPTCNT0-7.SDATASMPTCNT shall be fixed to 0. ➢ DDRHSSPIn_SDATASAMPLEPTLFT0-7.SDATASMPTLFT shall be fixed to 0. ➢ DDRHSSPIn_SDATASAMPLEPTRGH0-7.SDATASMPTRGH shall be fixed to 0. − DLP function is not supported. ➢ DDRHSSPIn_MCTRL.DLPEN shall be fixed to "0".
0x00000100	This version supports the following features. − Octal protocol (Dual Quad and Dual Legacy) − Sample Point Control function − DLP function
0x00000300	The DDRHSSPIn_SDATASAMPLEPT* registers have been extended from six ([5:0]) to seven ([6:0]) bits, All other features are exactly the same as MID = 0x00000100.

5.24. DDRHSSPI Command Sequencer Prefetch Address Register (DDRHSSPIn_CSPREFETCHADDR)

The DDRHSSPI Command Sequencer Prefetch Address Register is used to keep the next available address of memory that is available in Command Sequencer Mode Prefetch Buffer.

This register is used only in Command Sequencer Mode.

REGISTER_NAME	DDRHSSPIn_CSPREFETCHADDR
OFFSET	0x0124
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PRF ADDR[31]	PRF ADDR[30]	PRF ADDR[29]	PRF ADDR[28]	PRF ADDR[27]	PRF ADDR[26]	PRF ADDR[25]	PRF ADDR[24]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PRF ADDR[23]	PRF ADDR[22]	PRF ADDR[21]	PRF ADDR[20]	PRF ADDR[19]	PRF ADDR[18]	PRF ADDR[17]	PRF ADDR[16]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PRF ADDR[15]	PRF ADDR[14]	PRF ADDR[13]	PRF ADDR[12]	PRF ADDR[11]	PRF ADDR[10]	PRF ADDR[9]	PRF ADDR[8]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PRF ADDR[7]	PRF ADDR[6]	PRF ADDR[5]	PRF ADDR[4]	PRF ADDR[3]	PRF ADDR[2]	PRF ADDR[1]	PRF ADDR[0]
ACCESS_TYPE	R	R	R	R	R	R	R	R
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	0	0	0	0	0	0	0	0

[bit31:0] PRFADDR[31:0] : Prefetch Buffer Address

This register is used in Command Sequencer Mode.

Reading this register returns address available at the top of the Prefetch Buffer.

Writing to this register is not allowed.

5.25. DDRHSSPI SDATA Center Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTCNT0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Center Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock center sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTCNT0 Register is explained here. Other registers have identical fields.

REGISTER_NAME	DDRHSSPIn_SDATASAMPLEPTCNTi (i = 0~7)
OFFSET	0x0128 + i
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	SDATASMP TCNT[6]	SDATASMP TCNT[5]	SDATASMP TCNT[4]	SDATASMP TCNT[3]	SDATASMP TCNT[2]	SDATASMP TCNT[1]	SDATASMP TCNT[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7] read0 : Reserved

[bit6:0] SDATASMPTCNT[6:0] : SDATA Sample Point Center Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

5.26. DDRHSSPI SDATA Left Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTLFT0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Left Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock left sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTLFT0 Register is explained here. Other registers have identical fields.

REGISTER_NAME	DDRHSSPIn_SDATASAMPLEPTLFTi (i = 0~7)
OFFSET	0x0130 + i
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	SDATASMPTLFT[6]	SDATASMPTLFT[5]	SDATASMPTLFT[4]	SDATASMPTLFT[3]	SDATASMPTLFT[2]	SDATASMPTLFT[1]	SDATASMPTLFT[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7] read0 : Reserved

[bit6:0] SDATASMPTLFT[6:0] : SDATA Left Clock Sample Point Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

5.27. DDRHSSPI SDATA Right Clock Sample Point Registers (DDRHSSPIn_SDATASAMPLEPTRGH0-7)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI SDATA Right Clock Sample Point Registers 0-7 are used for fine tuning and configuring of the clock right sampling point for all incoming SDATA ports. Each SDATA[7:0] input port has one register.

Only the DDRHSSPIn_SDATASAMPLEPTRGH0 Register is explained here. Other registers have identical fields.

REGISTER_NAME	DDRHSSPIn_SDATASAMPLEPTRGHi (i = 0~7)
OFFSET	0x0138 + i
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	SDATASMPTRGH[6]	SDATASMPTRGH[5]	SDATASMPTRGH[4]	SDATASMPTRGH[3]	SDATASMPTRGH[2]	SDATASMPTRGH[1]	SDATASMPTRGH[0]
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit7] read0 : Reserved

[bit6:0] SDATASMPTRGH[6:0] : SDATA Right Clock Sample Point Control

This field defines, for an incoming SDATA port, sample clock delay.

Clock is delayed through a series of buffers and each buffer output from delay chain, can be selected by putting corresponding value into this register.

5.28. DDRHSSPI Data Learning Pattern Register (DDRHSSPIn_DLP)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI Data Learning Pattern Register is used for programming value of Data Learning Pattern supported by Serial Flash Memories.

REGISTER_NAME	DDRHSSPIn_DLP
OFFSET	0x0144
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DLP[7]	DLP[6]	DLP[5]	DLP[4]	DLP[3]	DLP[2]	DLP[1]	DLP[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:8] read0 : Reserved

[bit7:0] DLP[7:0] : Data Learning Pattern

This field is used to specify the data pattern of DLP function, and can be used only in Command Sequencer Mode.

Programming this register is possible only if DDRHSSPIn_MCTRL.DLPEN="1" and DLP is enabled on the Serial Flash Memory side.

It is necessary to write an expected DLP value to this field, before receiving a DLP from the Serial Flash Memory.

Results of comparing DLP data received from the Serial Flash Memory with DLP Register will be stored in DDRHSSPIn_DLPSAMPLESTATUS Register.

5.29. DDRHSSPI Data Learning Pattern Sample Status Register (DDRHSSPIn_DLPSAMPLESTATUS)

This feature is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300.

The DDRHSSPI Data Learning Pattern Status Register indicates status of DLP patterns recorded on each SDATA[7:0] input ports.

REGISTER_NAME	DDRHSSPIn_DLPSAMPLESTATUS
OFFSET	0x0148
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	DLPSMP LST7R	DLPSMP LST7C	DLPSMP LST7L	read0	DLPSMP LST6R	DLPSMP LST6C	DLPSMP LST6L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	DLPSMP LST5R	DLPSMP LST5C	DLPSMP LST5L	read0	DLPSMP LST4R	DLPSMP LST4C	DLPSMP LST4L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	DLPSMP LST3R	DLPSMP LST3C	DLPSMP LST3L	read0	DLPSMP LST2R	DLPSMP LST2C	DLPSMP LST2L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	DLPSMP LST1R	DLPSMP LST1C	DLPSMP LST1L	read0	DLPSMP LST0R	DLPSMP LST0C	DLPSMP LST0L
ACCESS_TYPE	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31] read0 : Reserved

[bit30] DLPSMPLST7R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH7, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH7, WITH − DDRHSSPIn_DLP[7:0] value

[bit29] DLPSMPLST7C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT7, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT7, WITH − DDRHSSPIn_DLP[7:0] value

[bit28] DLPSMPLST7L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT7, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[7] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT7, WITH − DDRHSSPIn_DLP[7:0] value

[bit27] read0 : Reserved

[bit26] DLPSMPLST6R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH6, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH6, WITH − DDRHSSPIn_DLP[7:0] value

[bit25] DLPSMPLST6C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT6, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT6, WITH − DDRHSSPIn_DLP[7:0] value

[bit24] DLPSMPLST6L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT6, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[6] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT6, WITH − DDRHSSPIn_DLP[7:0] value

[bit23] read0 : Reserved

[bit22] DLPSMPLST5R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH5, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH5, WITH − DDRHSSPIn_DLP[7:0] value

[bit21] DLPSMPLST5C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT5, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT5, WITH − DDRHSSPIn_DLP[7:0] value

[bit20] DLPSMPLST5L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT5, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[5] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT5, WITH − DDRHSSPIn_DLP[7:0] value

[bit19] read0 : Reserved

[bit18] DLPSMPLST4R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH4, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH4, WITH − DDRHSSPIn_DLP[7:0] value

[bit17] DLPSMPLST4C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT4, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT4, WITH − DDRHSSPIn_DLP[7:0] value

[bit16] DLPSMPLST4L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT4, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[4] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT4, WITH − DDRHSSPIn_DLP[7:0] value

[bit15] read0 : Reserved

[bit14] DLPSMPLST3R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH3, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH3, WITH − DDRHSSPIn_DLP[7:0] value

[bit13] DLPSMPLST3C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT3, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT3, WITH − DDRHSSPIn_DLP[7:0] value

[bit12] DLPSMPLST3L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT3, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[3] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT3, WITH − DDRHSSPIn_DLP[7:0] value

[bit11] read0 : Reserved

[bit10] DLPSMPLST2R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH2, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH2, WITH − DDRHSSPIn_DLP[7:0] value

[bit9] DLPSMPLST2C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT2, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT2, WITH − DDRHSSPIn_DLP[7:0] value

[bit8] DLPSMPLST2L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT2, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[2] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT2, WITH − DDRHSSPIn_DLP[7:0] value

[bit7] read0 : Reserved

[bit6] DLPSMPLST1R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH1, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH1, WITH − DDRHSSPIn_DLP[7:0] value

[bit5] DLPSMPLST1C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT1, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT1, WITH − DDRHSSPIn_DLP[7:0] value

[bit4] DLPSMPLST1L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT1, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[1] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT1, WITH − DDRHSSPIn_DLP[7:0] value

[bit3] read0 : Reserved

[bit2] DLPSMPLST0R

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH0, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTRGH0, WITH − DDRHSSPIn_DLP[7:0] value

[bit1] DLPSMPLST0C

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT0, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTCNT0, WITH − DDRHSSPIn_DLP[7:0] value

[bit0] DLPSMPLST0L

bit	Description
0	Indicates 8-bit data compare match: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT0, WITH − DDRHSSPIn_DLP[7:0] value
1	Indicates 8-bit data compare miss: − Sampled data on the SDATA[0] port, at the sampling point DDRHSSPIn_SDATASAMPLEPTLFT0, WITH − DDRHSSPIn_DLP[7:0] value

5.30. DDRHSSPI Command Sequencer Configuration Register (DDRHSSPIn_CSCFG)

The DDRHSSPI Command Sequencer Configuration Register configures the Command Sequencer in DDRHSSPI.

This register must be programmed by the Software before enabling the Command Sequencer Mode. Attributes like Transfer Protocol, Slave Select enable/disable and Size of Serial Flash Memory interfaced with DDRHSSPI can be configured in this register.

REGISTER_NAME	DDRHSSPIn_CSCFG
OFFSET	0x014C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	ITIMEREN	read0	read0	read0	MSEL[3]	MSEL[2]	MSEL[1]	MSEL[0]
ACCESS_TYPE	R/W	R0,W0	R0,W0	R0/W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	SSEL3EN	SSEL2EN	SSEL1EN	SSEL0EN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	DDRMODE	MBM[1]	MBM[0]	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:24] read0 : Reserved

[bit23] ITIMEREN : Idle Timer Enable

This bit is set whether Idle Timer is enabled or not.

bit	Description
0	ITIMER is disabled. SSEL remains asserted after transfers are finished. The programmed value in DDRHSSPIn_CSITIME Register is ignored.
1	ITIMER is enabled. SSEL is deasserted after the time expiration, that has been programmed in DDRHSSPIn_CSITIME Register.

[bit22:20] read0 : Reserved

[bit19:16] MSEL[3:0] : Memory Selection bits

This field indicates the range of the address space associated with each Slave Select. It also indicates the size of each memory banks in the selected Serial Flash Memory.

This field is used by Command Sequencer Mode for two things:

- To assert one of 4 Slave Selects for the memory mapped serial transfer
- To select the size of each memory bank in the selected Serial Flash Memory

Please refer to the Section 4.2 for more details.

[bit15:12] read0 : Reserved

[bit11] SSEL3EN : Slave Select 3 Enable

This bit enables Slave Select 3 for Command Sequencer Mode.

bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 3 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 3 is enabled.

[bit10] SSEL2EN : Slave Select 2 Enable

This bit enables Slave Select 2 for Command Sequencer Mode.

bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 2 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 2 is enabled.

[bit9] SSEL1EN : Slave Select 1 Enable

This bit enables Slave Select 1 for Command Sequencer Mode.

bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 1 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 1 is enabled.

[bit8] SSEL0EN : Slave Select 0 Enable

This bit enables Slave Select 0 for Command Sequencer Mode.

bit	Description
0	Access to the Serial Flash Memory mapped on Slave Select 0 is disabled.
1	Access to the Serial Flash Memory mapped on Slave Select 0 is enabled.

[bit7:4] read0 : Reserved

[bit3] DDRMODE : DDR Mode

This bit indicates whether DDRHSSPI is in DDR Mode or SDR Mode, in Command Sequencer Mode.

bit	Description
0	DDRHSSPI is in Single Data Rate (SDR) Mode.
1	DDRHSSPI is in Dual Data Rate (DDR) Mode.

[bit2:1] MBM[1:0] : Multi Bit Mode

This field decides the bit width on the Serial Interface in Command Sequencer Mode.

MBM[1:0]	Description
00	This setting is prohibited.
01	This setting is prohibited.
10	Serial transfer works at Quad Protocol. Read data is sampled on SDATA[3:0]. Memory instruction, address, and other control information are transmitted on SDATA[3:0].
11	This value is available only when DDRHSSPIn_MID.MID is 0x00000100 or 0x00000300. Serial transfer works at Dual Quad Protocol. Read data is sampled on SDATA[7:0]. Memory instruction, address, and other control information are transmitted on SDATA[7:0].

[bit0] read0 : Reserved

5.31. DDRHSSPI Command Sequencer Idle Time Register (DDRHSSPIn_CSITIME)

The DDRHSSPI Command Sequencer Idle Time Register configures the idle-time-out period of Command Sequencer Mode in the DDRHSSPI.

The Software must program this timeout value before enabling the Command Sequencer Mode. This register is used only in Command Sequencer Mode.

REGISTER_NAME	DDRHSSPIn_CSITIME
OFFSET	0x0150
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	ITIME[15]	ITIME[14]	ITIME[13]	ITIME[12]	ITIME[11]	ITIME[10]	ITIME[9]	ITIME[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	1	1	1	1	1	1	1	1
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	ITIME[7]	ITIME[6]	ITIME[5]	ITIME[4]	ITIME[3]	ITIME[2]	ITIME[1]	ITIME[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	1	1	1	1	1	1	1	1
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:16] read0 : Reserved

[bit15:0] ITIME[15:0] : Idle Time

This field decides the idle timeout period after a memory access in Command Sequencer Mode.

The idle timeout is available when DDRHSSPIn_CSCFG.ITIMEREN is "1".

Once DDRHSSPI completes the required number of memory read access on the Serial Interface and fills Prefetch Buffer, it keeps the Slave Select asserted. If no more access to the Serial Flash Memory is detected within the idle timeout period, then DDRHSSPI deasserts the Slave Select. The value of ITIME is based on the system clock (iHCLK).

5.32. DDRHSSPI Command Sequencer Address Extension Register (DDRHSSPIn_CSAEXT)

The DDRHSSPI Command Sequencer Address Extension Register is used to extend the usable size of Serial Flash Memory, which is mapped by the Command Sequencer.

The Software must program this register if the Address Extension feature is required, in order to virtually access a Serial Flash Memory up to 16GB. If Address Extension is not to be used, the Software must reset all bits in this register to "0".

REGISTER_NAME	DDRHSSPIn_CSAEXT
OFFSET	0x0154
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	AEXT[18]	AEXT[17]	AEXT[16]	AEXT[15]	AEXT[14]	AEXT[13]	AEXT[12]	AEXT[11]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	AEXT[10]	AEXT[9]	AEXT[8]	AEXT[7]	AEXT[6]	AEXT[5]	AEXT[4]	AEXT[3]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	AEXT[2]	AEXT[1]	AEXT[0]	read0	read0	read0	read0	read0
ACCESS_TYPE	R/W	R/W	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:13] AEXT[18:0] : Address Extension bits

This field is used to extend the address of Serial Flash Memory in Command Sequencer Mode.

The DDRHSSPIn_CSAEXT Register contains the 19 most significant bits [31:13] of the memory address which are generated by the Command Sequencer. The generated memory address on each Slave Select is a concatenation of the appropriate number of bits from the DDRHSSPIn_CSAEXT Register and bits from the address bus. Please refer to the Section 4.2 for more details.

If the address extension is not used, the Software should reset this field to 0x00000.

[bit12:0] read0 : Reserved

5.33. DDRHSSPI Command Sequencer Prefetch Buffer Configuration Register (DDRHSSPIn_CSPBUFFERCFG)

The DDRHSSPI Prefetch Buffer Configuration Register configures the operation of the Prefetch Buffer. The Software can also flush the Prefetch Buffer using the PBFLSH bit in this register.

REGISTER_NAME	DDRHSSPIn_CSPBUFFERCFG
OFFSET	0x0158
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	PBFLSH	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read1	read1	read1	read1
ACCESS_TYPE	R0,W0	R0,W0	R0/W0	R0/W0	R1/W1	R1/W1	R1/W1	R1/W1
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	1	1	1	1
NEUTRAL_TYPE	R	R	R	R	R	R	R	R

[bit31:20] read0 : Reserved

[bit19] PBFLSH : Prefetch Buffer Flush

This bit can be used by the Software to flush the Prefetch Buffer and RX Shift Register.

bit	Description
0	Writing "0" has no effect.
1	Writing "1" flushes the Prefetch Buffer and RX Shift Register. Please do not write "1" to this bit, while DDRHSSPIn_CSPBUFFERSTATUS.SSACTIVE bit is "1".

The read value is "0".

[bit18:4] read0 : Reserved

[bit3:0] read1 : Reserved

5.34. DDRHSSPI Command Sequencer Prefetch Buffer Status Register (DDRHSSPIn_CSPBUFFERSTATUS)

The DDRHSSPI Command Sequencer Prefetch Buffer Status Register contains the status bits like current fill-level of the Prefetch Buffer or whether the Serial Flash Memory access is in progress or not, i.e. Slave Select is still asserted. This register is used only in Command Sequencer Mode.

REGISTER_NAME	DDRHSSPIn_CSPBUFFERSTATUS
OFFSET	0x015C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	SSACTIVE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	PB LEVEL[5]	PB LEVEL[4]	PB LEVEL[3]	PB LEVEL[2]	PB LEVEL[1]	PB LEVEL[0]
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0
NEUTRAL_TYPE	*	*	*	*	*	*	*	*

[bit31:17] read0 : Reserved

[bit16] SSACTIVE : Slave Select Active

This flag indicates whether a Slave Select is active.

bit	Description
0	Slave Select is not active.
1	Slave Select is active.

[bit15:6] read0 : Reserved

[bit5:0] PBLEVEL[5:0] : Current Fill Level of Prefetch Buffer

This field indicates the current fill level of the Prefetch Buffer.

CHAPTER 27: Stepper Motor Controller

This chapter explains the functions and operations of the Stepper Motor Controller (SMC).

1. Overview
2. Configuration and Block Diagram
3. Operation of the Stepper Motor Controller
4. Registers

CODE: FIP021-E1.1

1. Overview

This section lists the main features for Stepper Motor Controller.

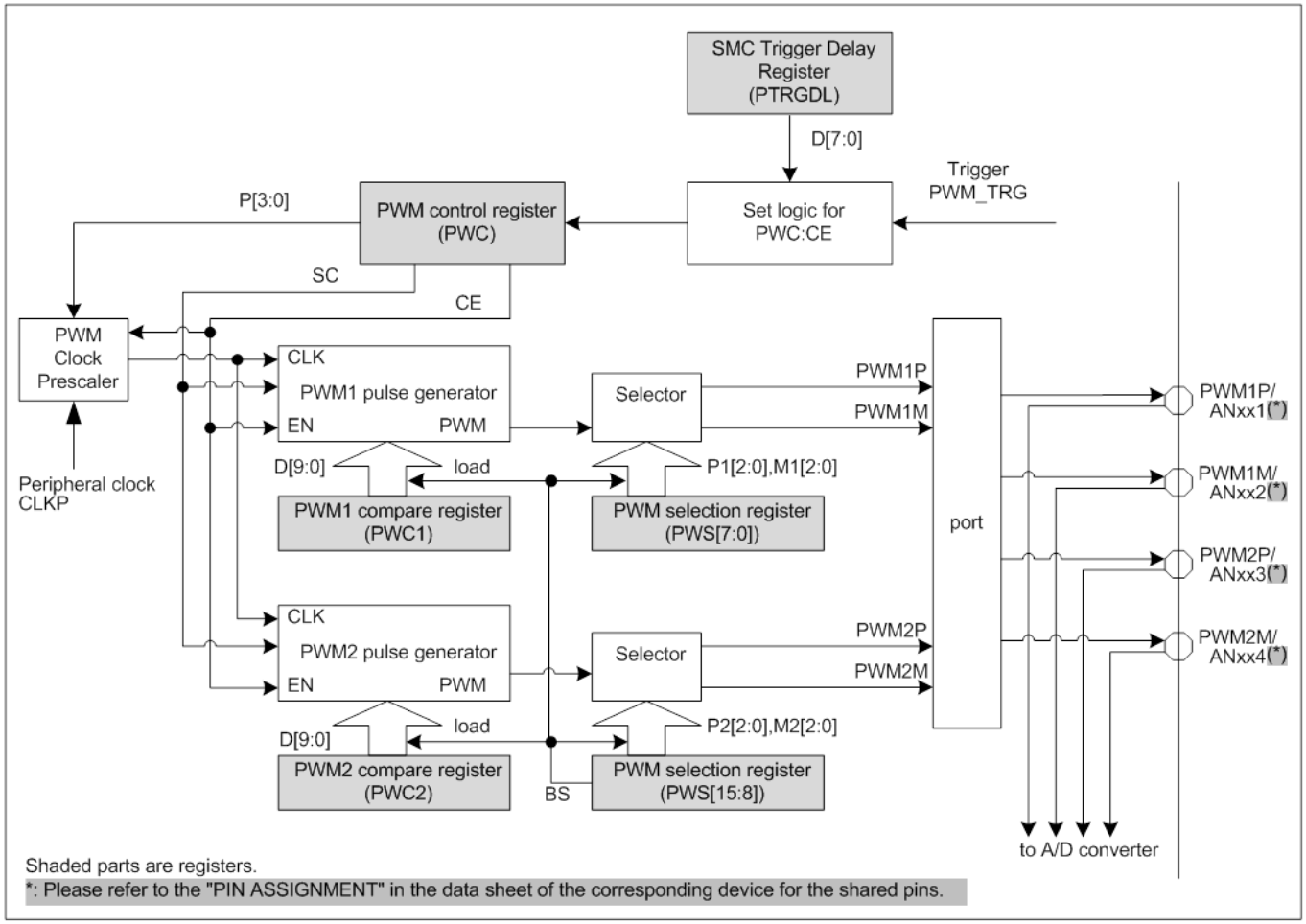
Features of the Stepper Motor Controller:

- Four high-current output drivers: Stepper Motor Controller provides four driver outputs with high current driving capability. Two motor coils can be connected directly to four stepper motor controller pins. The motor rotation is controlled by a combination of the PWM Pulse Generator and selector logic circuits.
- Two PWM Pulse Generators with 10-bit/8-bit operation mode (selected by software):
  - For the PWM pulse width in 8-bit operation mode, the duty can be specified within the range from 0% to 99.6%.
  - For the PWM pulse width in 10-bit operation mode, the duty can be specified within the range from 0% to 99.9%.
- A synchronization mechanism that enables synchronous operation of PWM Pulse Generators.
- A variable operating Clock Prescaler (the division ratio from a Peripheral clock can be selected by software).
- Output selector logic ("High", "Low", PWM pulse and "Hi-Z" signal levels can be selected).

2. Configuration and Block Diagram

This section shows block diagrams of the Stepper Motor Controller.

Figure 2-1 Block Diagram of the Stepper Motor Controller



PWM Clock Prescaler

The PWM Clock Prescaler divides the Peripheral clock (CLKP) and generates the PWM operating clock. The PWM operating clock is used by the PWM Pulse Generators.

PWM1 Pulse Generator / PWM2 Pulse Generator

The PWM Pulse Generators generate the pulse-width modulated signals according to the waveform configuration and the selected operation mode.

Selector

This sub-module, based on configuration, selects the level of the signal that will be sent to the Stepper Motor Controller output pins.

Port

The Port controls the I/O port for the Stepper Motor Controller pins.

PWM Control Register (PWC)

The PWM Control Register (PWC) starts/stops the Stepper Motor Controller, sets the PWM Prescaler and selects the operation mode for the PWM Pulse Generators.

PWM1 and PWM2 Compare Registers (PWC1, PWC2)

The PWM1 and PWM2 Compare Registers determine the PWM pulse width.

PWM Selection Register (PWS)

The PWM Selection Register selects the signal levels ("H" and "L", PWM pulse and "Hi-Z") to be output to the Stepper Motor Controller pins. It also sets the update instruction for the system to synchronize the PWM Pulse Generators.

SMC Trigger Delay Register (PTRGDL)

The SMC Trigger Delay Register (PTRGDL) is used to delay the start of the operation by 0 to 255 clock cycles.

Set Logic for PWC.CE

The Set logic for PWC.CE receives the trigger and waits for a configured delay time and sets automatically PWC.CE bit.

3. Operation of the Stepper Motor Controller

This section describes the operation of the Stepper Motor Controller.

3.1 Operation of PWM Pulse Generator
3.2 Selection of Motor Drive Signals
3.3 Synchronization System of Stepper Motor Controller
3.4 Operation of the SMC Trigger Delay Function

3.1. Operation of PWM Pulse Generator

The PWM period is determined by the selected division ratio from a Peripheral clock (CLKP) and the operation mode (8-bit/10-bit) of the PWM Pulse Generator:

- During configuration, software can select the PWM operating clock by specifying the division ratio for the Peripheral clock (CLKP). The division ratio for the Peripheral clock is defined by setting the PWM operating clock prescaler bits (PWC.P[3:0]).
- Software can also configure the operation mode for the counter that controls the PWM period. When the Operation Mode Switching bit (PWC.SC) is set to "0", 8-bit operation mode is selected (PWM period = 256 counts). If this bit is set to "1", 10-bit operation mode is selected (PWM period = 1024 counts).

Examples of operating clock values for different configurations are given in Table 3-1.

Table 3-1 Relation between PWM Operating Clock and PWM Period for Different Operation Modes

PWC P[3:0]	PWM Operating Clock (CLK)	PWM Period [µs] (CLKP = 16 MHz)		PWM Period [µs] (CLKP = 60 MHz)	
		PWC SC = "0"	PWC SC = "1"	PWC SC = "0"	PWC SC = "1"
0000	CLKP	16	64	4.267	17.067
0001	CLKP/4	64	256	17.067	68.267
0010	CLKP/5	80	320	21.333	85.333
0011	CLKP/6	96	384	25.6	102.4
0100	CLKP/8	128	512	34.133	136.533
0101	CLKP/10	160	640	42.667	170.667
0110	CLKP/12	192	768	51.2	204.8
0111	CLKP/16	256	1024	68.267	273.067
1000	CLKP/2	32	128	8.533	34.133
1001	CLKP/8	128	512	34.133	136.533
1010	CLKP/10	160	640	42.667	170.667
1011	CLKP/12	192	768	51.2	204.8
1100	CLKP/16	256	1024	68.267	273.067
1101	CLKP/20	320	1280	85.333	341.333
1110	CLKP/24	384	1536	102.4	409.6
1111	CLKP/32	512	2048	136.533	546.133

CLKP: Peripheral clock

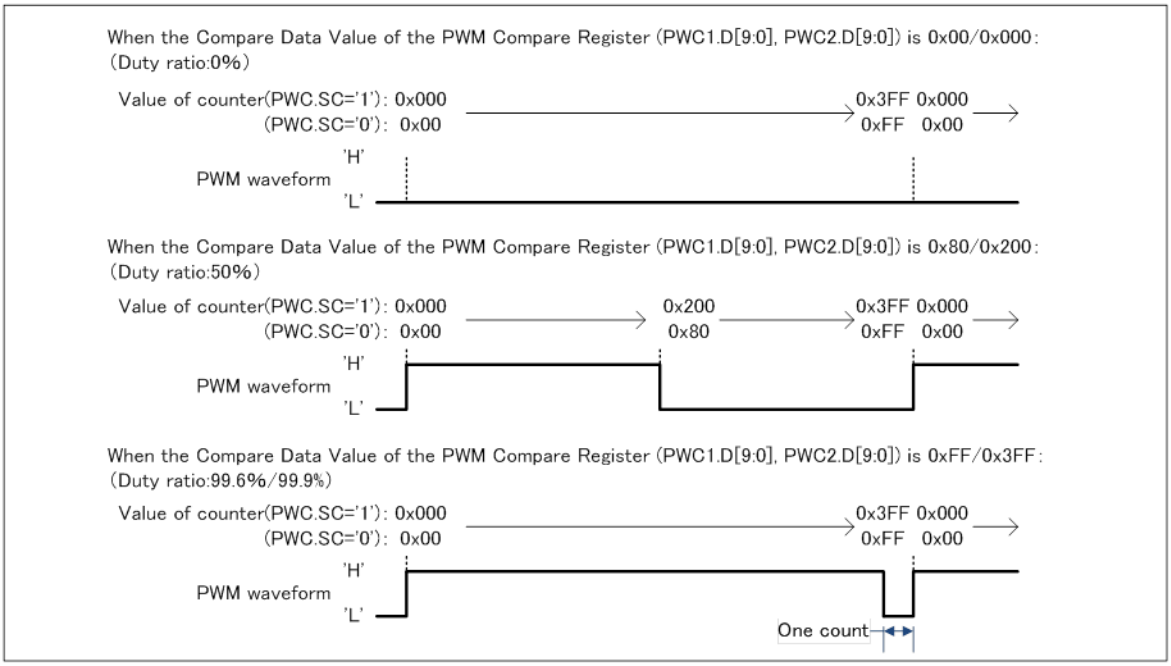
The PWM pulse width (duty cycle) is determined by the PWM Compare Data Value (PWC1.D[9:0], PWC2.D[9:0]). In 8-bit operation mode (PWC.SC = "0"), the PWM Compare Data Value is taken from the PWC1.D[7:0], PWC2.D[7:0] bits and bits PWC1.D[9:8], PWC2.D[9:8] are "don't care". In 10-bit operation mode (PWC.SC = "1"), the PWM Compare Data Value is taken from the PWC1.D[9:0], PWC2.D[9:0] bits. Below are the ranges in which PWM Compare Data Values (PWC1.D[9:0], PWC2.D[9:0]) can be set:

- In 8-bit operation mode (PWC.SC = "0") 0% to 99.6% (0x00 to 0xFF)
- In 10-bit operation mode (PWC.SC = "1") 0% to 99.9% (0x000 to 0x3FF)

The PWM Pulse Generator is started/stopped using the Count Enable bit (PWC.CE). When this bit is set to "1", the PWM Pulse Generator starts the counter that controls the PWM period. The counter increments its value from 0x00 (0x000) at the rising edge of the PWM operating clock. The PWM Pulse Generator output remains at the "H" level until the counter value matches the PWM Compare Data Value. Once the PWM Compare Data Value is reached, the PWM Pulse Generator output changes to "L" and remains at the "L" level until the counter value overflows: 0xFF ➔ 0x00 (0x3FF ➔ 0x000).

Figure 3-1 shows the PWM waveforms generated by the PWM Pulse Generator.

Figure 3-1 Examples of PWM1 and PWM2 Output Waveforms



If the PWM Pulse Generator is stopped (PWC.CE = "0"), the setting of the PWM pulse width (duty cycle) and the output pin selection settings are not reflected on the Stepper Motor Controller outputs.

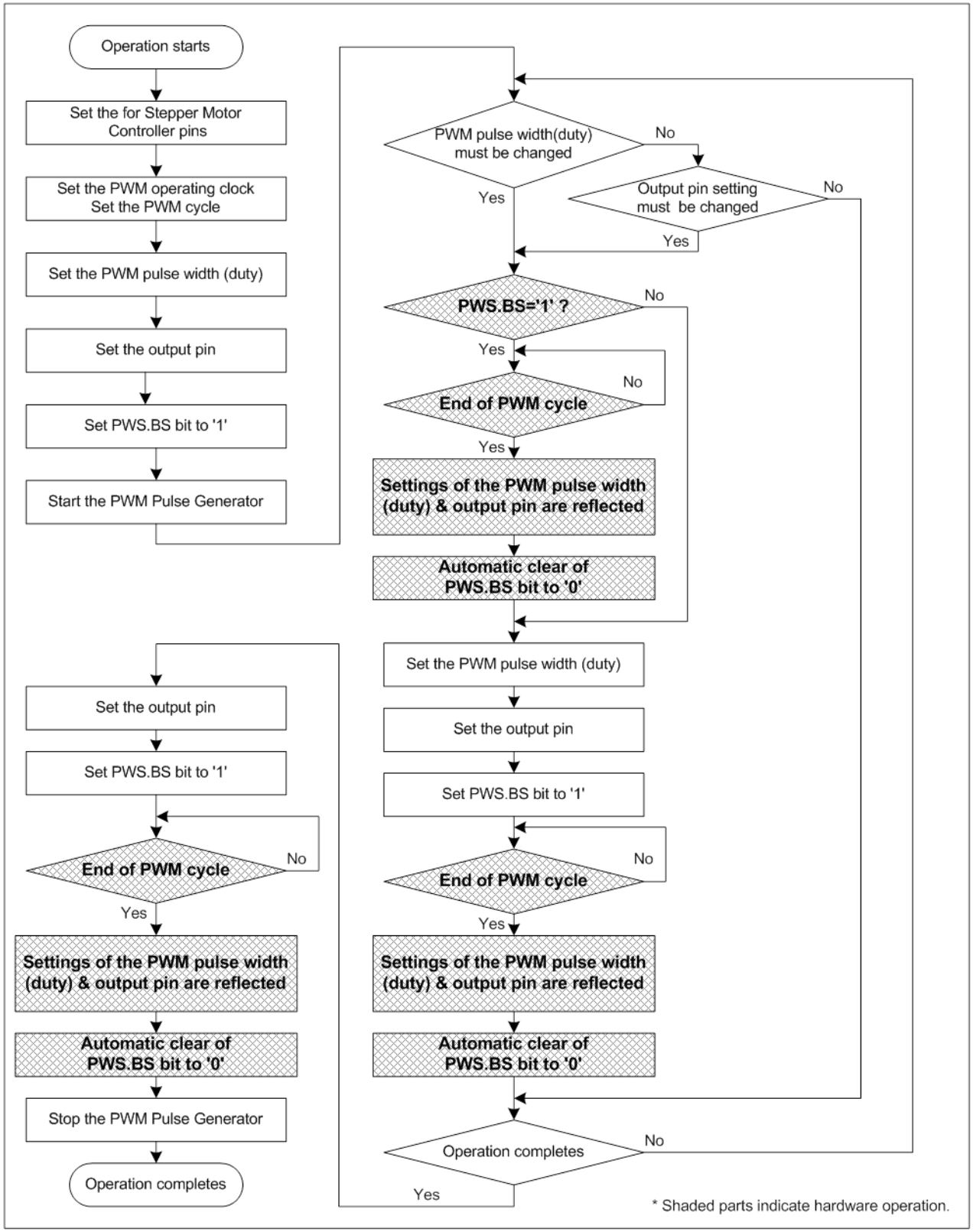
Notes:

- *If the PWM Pulse Generator is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(PWC.CE="1")
- *If the PWM Pulse Generator is stopped when the Output Update bit (PWS.BS) is "1", the PWS.BS bit will retain the value "1"until the PWM Pulse Generator is started and the beginning of the new PWM cycle is detected. As a result, the setting of the PWM pulse width (duty cycle) and the output pin selection settings are reflected on the Stepper Motor Controller outputs. To avoid this scenario, clear PWC.CE only after PWS.BS is automatically cleared by the SMC.*
- *If the PWS.BS bit is "1", keep the following registers value because the setting of the PWM pulse width (duty cycle) and the output pin selection settings are not reflected at the same time.*

- *PWM1 Compare Register.Compare Data Value(PWC1.D[9:0])*
- *PWM2 Compare Register.Compare Data Value(PWC2.D[9:0])*
- *PWM Selection Register.Output Selection bits(PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])*

An example for the Stepper Motor Controller setting procedure is given on the Figure 3 2.

Figure 3-2 Example of the Setting Procedure for the Stepper Motor Controller



3.2. Selection of Motor Drive Signals

The user can select the signal at each output pin of the Stepper Motor Controller according to the Table 3-2 and Table 3-3. The selection of output pin is done by configuring the PWM Selection Register (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).

Table 3-2 Selection of Motor Drive Signals and Setting of PWM1 Selection Bits

PWS.P1[2:0]	PWM1P
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

PWS.M1[2:0]	PWM1M
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

X: don't care

Table 3-3 Selection of Motor Drive Signals and Setting of PWM2 Selection Bits

PWS.P2[2:0]	PWM2P
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

PWS.M2[2:0]	PWM2M
000	L
001	H
01X	PWM pulse
1XX	Hi-Z

X: don't care

3.3. Synchronization System of Stepper Motor Controller

The Stepper Motor Controller contains a dedicated system developed to enable synchronization of PWM1 and PWM2 Pulse Generators, as well as Stepper Motor Controller output pin selection. The synchronization system is controlled by the PWM Selection Register Output Update bit (PWS.BS). Until the Output Update bit (PWS.BS) is set to "1", changes made to the PWM Compare Registers (PWC1, PWC2) and the PWM Selection Register (PWS) are not reflected on the output. Configuring the Stepper Motor Controller output pin selection and setting the Output Update bit (PWS.BS) simultaneously (in the same software access) is allowed.

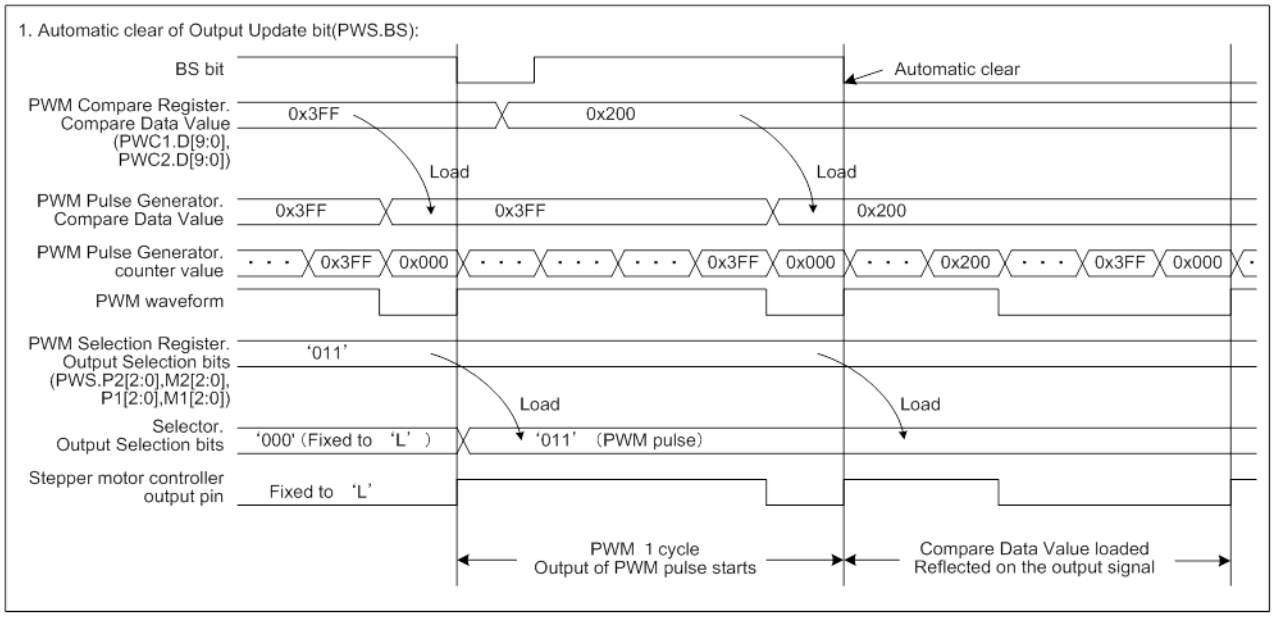
When the Output Update bit (PWS.BS) is set to "1"by software, and the end of current PWM cycle is detected, PWM Pulse Generator and motor drive selection logic will load values of the corresponding registers. The Output Update bit (PWS.BS) is automatically cleared to "0" at the beginning of the next PWM cycle.

If the Output Update bit (PWS.BS) is equal to "1"and the software access is initiated to set this bit to "1"at the beginning of the PWM cycle (at the same time when an automatic clear of Output Update bit (PWS.BS) is issued), the Output Update bit (PWS.BS) will retain the value "1"(i.e. no change is made to the PWS.BS bit) and the automatic clearing is canceled for this PWM cycle.

The following scenario is shown on Figure 3-3:

1. Automatic clear of the Output Update bit (PWS.BS): load operation is executed and reflected on the output.

Figure 3-3 Load Timing of PWM Compare Register and PWM Selection Register



3.4. Operation of the SMC Trigger Delay Function

After the Stepper Motor Controller receives a trigger input (PWM_TRG), it waits for a configured delay time to elapse before the Count Enable bit (PWC.CE) is automatically set and the Stepper Motor Controller starts operating. The delay time is configured in the SMC Trigger Delay Register (PTRGDL). The Stepper Motor Controller will continue to operate until the Count Enable bit (PWC.CE) is reset by software.

Figure 3-4 demonstrates the software flow for receiving a trigger input (PWM_TRG). An example, illustrating how the trigger starts Stepper Motor Controller, is shown on Figure 3-5.

Note:

- *The trigger should only be received when the Stepper Motor Controller are not running (i.e. the Count Enable bit (PWC.CE) is cleared to "0").*

Figure 3-4 Flowchart for Triggering Stepper Motor Controller

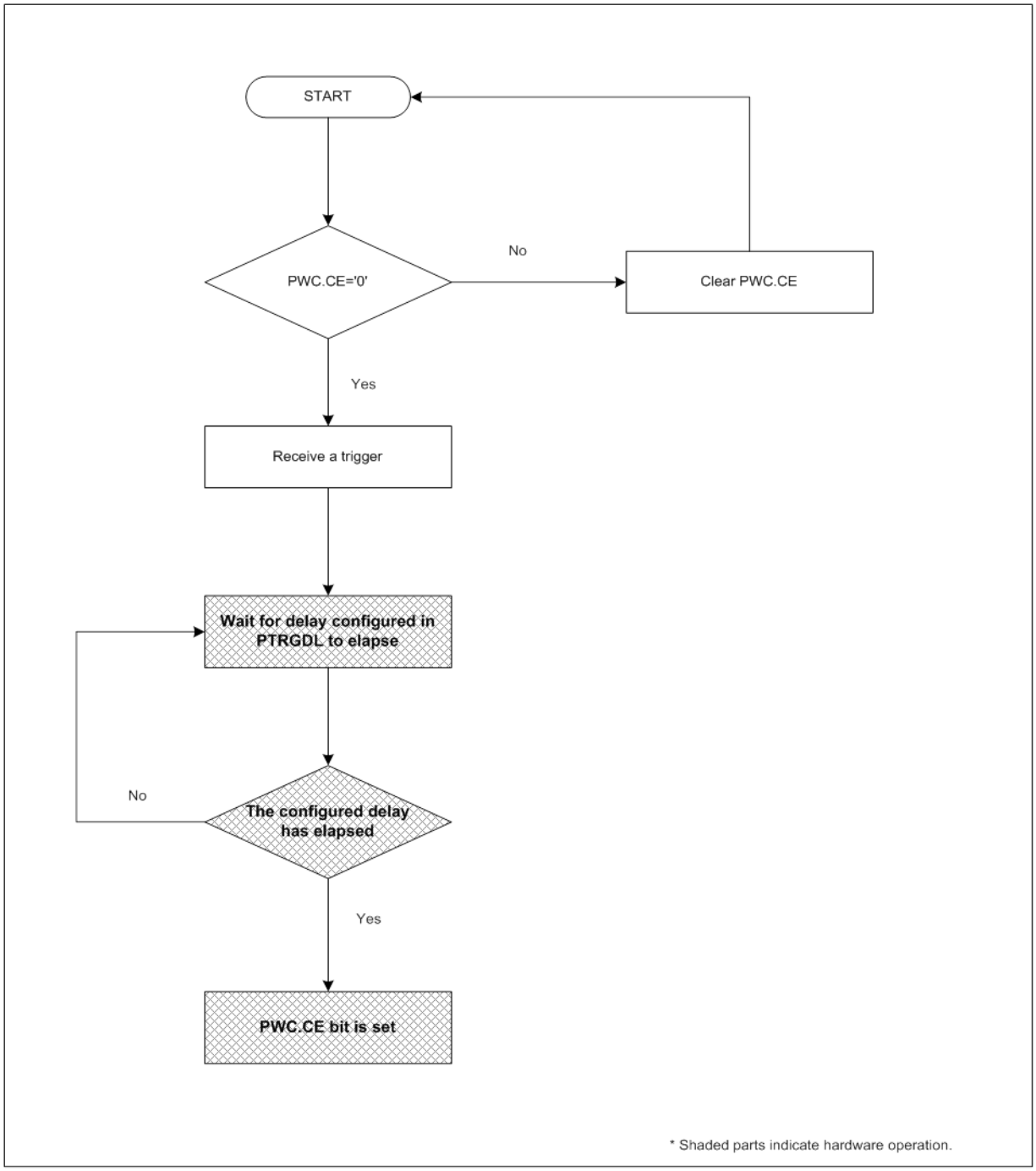


Figure 3-5 Timing Diagram for Triggering of Stepper Motor Controller

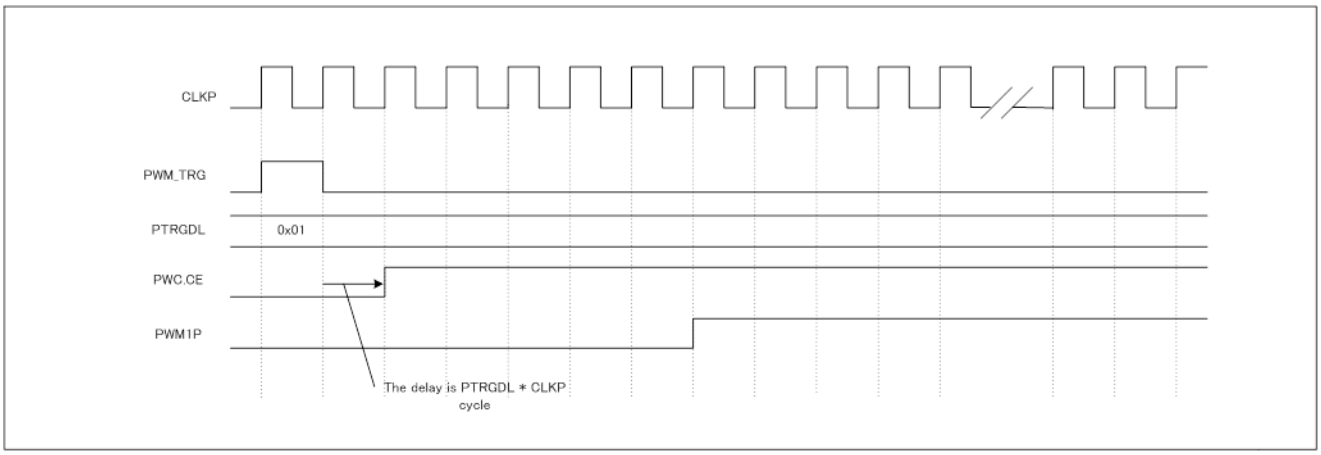


Figure 3-5 shows how the trigger starts Stepper Motor Controller (PTRGDL=0x01). If the SMC Trigger Delay Register (PTRGDL) setting is 0x00, PWC.CE is set immediately after receiving the trigger input (PWM_TRG). If the SMC Trigger Delay Register (PTRGDL) setting is 0x01, PWC.CE is set with a delay of one clock cycle after receiving the trigger input (PWM_TRG).

4. Registers

This section describes the registers of the Stepper Motor Controller.

■ **The Following Registers are Available for One Stepper Motor Controller:**

PWM Control Register (PWC)

PWM1 and PWM2 Compare Registers (PWC1, PWC2)

PWM Selection Register (PWS)

PWM Selection Set Register (PWSS)

SMC Trigger Delay Register (PTRGDL)

Table 4-1 List of Registers for the Stepper Motor Controller

Abbreviation	Register Name	Reference
PWC	PWM Control Register	4.1
PWC1	PWM1 Compare Register	4.2.1
PWC2	PWM2 Compare Register	4.2.2
PWS	PWM Selection Register	4.3
PWSS	PWM Selection Set Register	4.4
PTRGDL	SMC Trigger Delay Register	4.5

4.1. PWM Control Register (PWC)

The PWM Control Register starts/stops the Stepper Motor Controller, sets the PWM prescaler and selects the operation mode for the generation of the PWM pulse.

REGISTER_NAME	SMCi_PWC
OFFSET	0x00000000
ACCESS_SIZE	B, H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved						P[3]	Reserved
ACCESS_TYPE	R0,W0						R/W	R0/W0
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0						0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	P[2:0]			CE	SC	Reserved	Reserved
ACCESS_TYPE	R0/W0	R/W			R,W	R/W	R0,W0	R0/W0
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0			0	0	0	0

[bit15:10] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit9] P[3]: PWM Operating Clock Prescaler bit

P[3] bit, in combination with P[2:0] bits, specifies the clock input signal to the PWM Pulse Generators.

[bit8:7] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit6:4] P[2:0]: PWM Operating Clock Prescaler bits

P[2:0] bits, in combination with P[3] bit, specify the clock input signal to the PWM Pulse Generators.

P[3]	P[2:0]	Description
0	000	PWM Operating clock is equal to the Peripheral clock (CLKP)
0	001	PWM Operating clock is equal to CLKP/4
0	010	PWM Operating clock is equal to CLKP/5

P[3]	P[2:0]	Description
0	011	PWM Operating clock is equal to CLKP/6
0	100	PWM Operating clock is equal to CLKP/8
0	101	PWM Operating clock is equal to CLKP/10
0	110	PWM Operating clock is equal to CLKP/12
0	111	PWM Operating clock is equal to CLKP/16
1	000	PWM Operating clock is equal to CLKP/2
1	001	PWM Operating clock is equal to CLKP/8
1	010	PWM Operating clock is equal to CLKP/10
1	011	PWM Operating clock is equal to CLKP/12
1	100	PWM Operating clock is equal to CLKP/16
1	101	PWM Operating clock is equal to CLKP/20
1	110	PWM Operating clock is equal to CLKP/24
1	111	PWM Operating clock is equal to CLKP/32

CLKP: Peripheral clock

Note:

- *Configuration of the PWM Operating Clock Prescaler bits (P[3:0]) shall be done while counting operation is disabled (PWC.CE = "0").*

[bit3] CE: Count Enable bit

The CE bit enables operation of the PWM Pulse Generators. When CE bit is set to "1", the PWM Pulse Generator starts its operating. The PWM2 Pulse Generator starts with the delay of one Peripheral clock cycle (CLKP) after the PWM1 Pulse Generator is started.

The CE bit can be set and cleared by the software. This bit also can be set to "1" by receiving trigger. To receive trigger, the CE bit must first be cleared to "0".

Bit	Description
0	Initializes and stops the operation of the PWM Pulse Generators
1	Starts the operation of the PWM Pulse Generators

Notes:

- *If the PWM Pulse Generator is started(CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", P2[2:0], M2[2:0], P1[2:0], M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(CE="1")
- *If the PWM Pulse Generator is stopped (CE="0") when the PWM pulse is selected by the Output Selection bits (PWS.P2[2:0], M2[2:0], P1[2:0], M1[2:0]), the pin for which the PWM pulse is selected is fixed to the "L" level.*
- *If the trigger input is used, write "0" to the CE bit only when the CE bit is "1". Because the CE bit can be cleared at the same time when the CE bit is set by the trigger.*
- *The CE bit shall be set to "1" after the setting of the PWM Operating Clock Prescaler bits (PWC.P[3:0]) and the Operation Mode Switching bit (PWC.SC) is completed.*

[bit2] SC: Operation Mode Switching bit

This bit specifies the operation mode of the PWM Pulse Generators

Bit	Description
0	8-bit operation mode is selected
1	10-bit operation mode is selected

Note:

- *The PWC.CE bit shall be set to "1" after the setting of the PWM Operating Clock Prescaler bits (PWC.P[3:0]) and the Operation Mode Switching bit (SC) is completed.*

[bit1:0] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

4.2. PWM1 and PWM2 Compare Registers (PWC1, PWC2)

The value of the 8/10 bit PWM1 and PWM2 Compare Registers determines the width of the PWM pulse. A value of 0x00 (0x000) indicates that the PWM duty cycle is 0%, while a value of 0xFF (0x3FF) indicates that the PWM duty cycle is 99.6% (99.9%).

4.2.1. PWM1 Compare Register(PWC1)

REGISTER_NAME	SMCi_PWC1
OFFSET	0x00000002
ACCESS_SIZE	B, H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved						D[9:8]	
ACCESS_TYPE	R0,W0						R/W	
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0						X	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	D[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	X							

[bit15:10] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit9:0] D[9:0]: PWM1 Compare Data Value

The PWM1 Compare Data Value D[9:0] are used to set the PWM1 duty cycle. When the Operation Mode Switching bit is set to 8-bit operation mode (PWC.SC= "0"), the compare data value is taken from D[7:0] bits and D[9:8] bits are "don't care". When the Operation Mode Switching bit is set to 10-bit operation mode (PWC.SC= "1"), the compare data value is taken from D[9:0] bits.

D[9:0]	Description
0x(0)00	0%
…	Set the PWM pulse width
0x(3)FF	(99.9%) 99.6%

Notes:

- *PWM Pulse Generator.Compare Data Value is updated at the end of PWM cycle, after PWS.BS bit is set to "1".*
- *When PWC.SC bit is set to "0" (PWM Pulse Generators are operating in 8-bit mode), D[9:8], PWC2.D[9:8] bits are "don't care".*
- *The software must always make a 16-bit write access to PWM1 and PWM2 Compare Registers (PWC1, PWC2) to ensure consistency of data. However, 8-bit write access to PWM1 or PWM2 Compare Register (PWC1, PWC2) is possible and will not result in error response.*
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

4.2.2. PWM2 Compare Register(PWC2)

REGISTER_NAME	SMCi_PWC2
OFFSET	0x00000004
ACCESS_SIZE	B, H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved						D[9:8]	
ACCESS_TYPE	R0,W0						R/W	
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0						X	

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	D[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	X							

[bit15:10] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit9:0] D[9:0]: PWM2 Compare Data Value

The PWM2 Compare Data Value D[9:0] bits are used to set the PWM2 duty cycle. When the Operation Mode Switching bit is set to 8-bit operation mode (PWC.SC= "0"), the compare data value is taken from D[7:0] bits and D[9:8] bits are "don't care". When the Operation Mode Switching bit is set to 10-bit operation mode (PWC.SC= "1"), the compare data value is taken from D[9:0] bits.

D[9:0]	Description
0x(0)00	0%
…	Set the PWM pulse width
0x(3)FF	(99.9%) 99.6%

Notes:

- *PWM Pulse Generator.Compare Data Value is updated at the end of PWM cycle, after PWS.BS bit is set to "1".*
- *When PWC.SC bit is set to "0" (PWM Pulse Generators are operating in 8-bit mode), PWC1.D[9:8], D[9:8] bits are "don't care".*

- *The software must always make a 16-bit write access to PWM1 and PWM2 Compare Registers (PWC1, PWC2) to ensure consistency of data. However, 8-bit write access to PWM1 or PWM2 Compare Register (PWC1, PWC2) is possible and will not result in error response.*
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

4.3. PWM Selection Register (PWS)

The PWM Selection Register (PWS) sets the output state of the Stepper Motor Controller pins ("L", "H", PWM pulse or "Hi-Z") and allows the update of the PWM duty cycle or output pin states.

REGISTER_NAME	SMCi_PWS
OFFSET	0x00000006
ACCESS_SIZE	B, H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	BS	P2[2:0]			M2[2:0]		
ACCESS_TYPE	R0,W0	R/+W	R/W			R/W		
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0			0		

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		P1[2:0]			M1[2:0]		
ACCESS_TYPE	R0/W0		R/W			R/W		
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0		0			0		

[bit15] Reserved: Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit14] BS: Output Update bit

The Output Update bit (BS) synchronously updates the Selector.Output Selection bits and PWM Pulse Generator.Compare Data Value (See Figure3-3). To change the settings for the PWM duty cycle or output pin states, the BS bit must be set to "1"after (or at the same time) a new configuration is written to the PWM1 and PWM2 Compare Registers (PWC1, PWC2) and PWM Selection Register (PWS). If the BS bit is set to "1"while PWM Pulse Generators are operating, the new settings are loaded at the end of the current PWM cycle. The BS bit is cleared automatically at the beginning of the following PWM cycle. As long as BS bit is "0", the PWM duty cycle and output pin states are not changed.

This bit can also be set by the PWSS.BSS bit.

Bit	Description
0	Disables the update of the setting for the PWM output
1	Enables the update of the setting for the PWM output

Notes:

- *If the PWM Pulse Generator is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*

1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
2. Set the PWM Selection Register.Output Selection bits(BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
3. Set the PWM Control Register:Count Enable bit(PWC.CE="1")

- *If the BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit13:11] P2[2:0]: Plus Output 2 Selection bits

The P2[2:0] bits select the output signal for PWM2P.

Bits	Description
000	"L": PWM2P output is fixed to low level
001	"H": PWM2P output is fixed to high level
01X	PWM pulse is driven on the PWM2P output
1XX	"Hi-Z": PWM2P output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit10:8] M2[2:0]: Minus Output 2 Selection bits

The M2[2:0] bits selects the output signal for PWM2M.

Bits	Description
000	"L": PWM2M output is fixed to low level
001	"H": PWM2M output is fixed to high level
01X	PWM pulse is driven on the PWM2M output
1XX	"Hi-Z": PWM2M output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit7:6] Reserved: Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit5:3] P1[2:0]: Plus Output 1 Selection bits

The P1[2:0] bits select the output signal for PWM1P.

Bits	Description
000	"L": PWM1P output is fixed to low level
001	"H": PWM1P output is fixed to high level
01X	PWM pulse is driven on the PWM1P output
1XX	"Hi-Z": PWM1P output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9.0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], P1[2:0], PWS.M1[2:0]).*

[bit2:0] M1[2:0]: Minus Output 1 Selection bits

The M1[2:0] bits select the output signal for PWM1M.

Bits	Description
000	"L": PWM1M output is fixed to low level
001	"H": PWM1M output is fixed to high level
01X	PWM pulse is driven on the PWM1M output
1XX	"Hi-Z": PWM1M output is set to high impedance

X: don't care

Note:

- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], M1[2:0]).*

4.4. PWM Selection Set Register (PWSS)

The PWM Selection Set Register is used to set the value of the PWS.BS bit.

REGISTER_NAME	SMCi_PWSS
OFFSET	0x00000008
ACCESS_SIZE	B,H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	BSS	Reserved					
ACCESS_TYPE	R0,W0	R0,W	R0,W0					
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0	0	0					

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0							

[bit15] Reserved: Reserved bit

When writing, always write "0".

When reading, "0" is always read.

[bit14] BSS: Set bit for the Output Update bit

This bit sets the value of PWS.BS bit.

Bit	Description
0	No effect
1	Sets the PWS.BS bit to "1". Reading of this bit returns "0".

Notes:

- *If the PWM Pulse Generators is started(PWC.CE="1"), operate following steps because the PWM Pulse Generator.Compare Data Value must be initialized.*
  1. Set the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0])
  2. Set the PWM Selection Register.Output Selection bits(PWS.BS="1", PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0])
  3. Set the PWM Control Register.Count Enable bit(PWC.CE="1")
- *If the PWS.BS bit is "1" , don't change the PWM Compare Register.Compare Data Value(PWC1.D[9:0], PWC2.D[9:0]) and the PWM Selection Register.Output Selection bits (PWS.P2[2:0], PWS.M2[2:0], PWS.P1[2:0], PWS.M1[2:0]).*

[bit13:0] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

4.5. SMC Trigger Delay Register (PTRGDL)

The SMC Trigger Delay Register (PTRGDL) is used to delay the start of the operation by 0 to 255 clock cycles.

REGISTER_NAME	SMCi_PTRGDL
OFFSET	0x0000000A
ACCESS_SIZE	B, H
MULTIPLE	1
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,W0							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	D[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0							

[bit15:8] Reserved: Reserved bits

When writing, always write "0".

When reading, "0" is always read.

[bit7:0] D[7:0]: Trigger Delay bits

The Trigger Delay bits (D[7:0]) configure a delay of the trigger input that starts the PWM operation from 0 to 255 clock cycles.

Bits	Description
0x00	No delay (delay disabled)
0x01	Delay of 1 clock cycle between the trigger and the clock cycle in which the PWM generation starts
…	…
0xFF	Delay of 255 clock cycles between the trigger and the clock cycle in which the PWM generation starts

CHAPTER 28: Trigger Configuration of Stepper Motor Controller

This chapter explains the trigger configuration of stepper motor controller.

1. Overview
2. Configuration and Block Diagram
3. Operation
4. Registers

CODE: SMCTRG-S6J3200-E1

1. Overview

The chapter describes the function and its operation of both SMC trigger selection register and SMC trigger register. See the chapter of Stepper motor controller with this chapter.

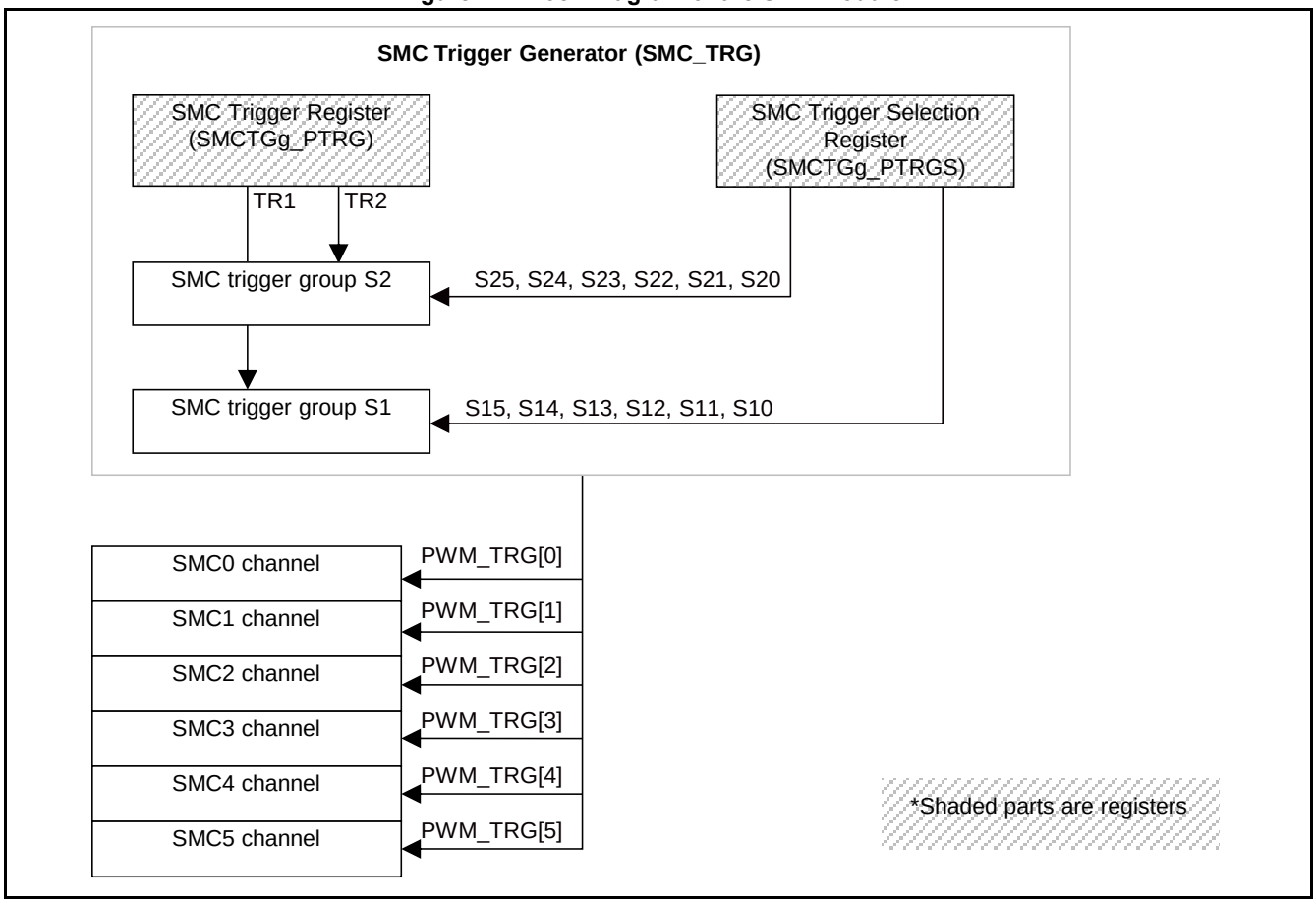
2. Configuration and Block Diagram

See the block diagram below.

Notes:

- *The suffix 'g' in the register name indicates that the register is an instance 'g' of the SMC Trigger Generator.*
- *The suffix 'i' in the register name indicates that the register is an instance 'i' of the SMC channel.*

Figure 2-1 Block Diagram of the SMC Module



3. Operation

The trigger configuration and its operation is described in 4.

4. Registers

Offset	Register Name
0x00000000	SMCTGg_PTRGS
0x00000002	SMCTGg_PTRG

4.1. SMC Trigger Selection Register (SMCTGg_PTRGS)

The SMC Trigger Selection Register (SMCTGg_PTRGS) selects multiple SMC channels that shall be triggered by the SMC Trigger Register to start operation synchronously or with a delay. The SMC channels can be combined into two trigger groups.

REGISTER_NAME	SMCTGg_PTRGS
OFFSET	0x00000000
ACCESS_SIZE	B, H
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved		S25	S24	S23	S22	S21	S20
ACCESS_TYPE	R0,WX		R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0		0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved		S15	S14	S13	S12	S11	S10
ACCESS_TYPE	R0,WX		R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0		0	0	0	0	0	0

[bit13] S25 : Trigger Enable for Operation of SMC<6*g+5>

This bit selects SMC<6*g+5> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+5> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+5> is triggered by SMCTGg_PTRG:TR2

[bit12] S24 : Trigger Enable for Operation of SMC<6*g+4>

This bit selects SMC<6*g+4> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+4> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+4> is triggered by SMCTGg_PTRG:TR2

[bit11] S23 : Trigger Enable for Operation of SMC<6*g+3>

This bit selects SMC<6*g+3> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+3> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+3> is triggered by SMCTGg_PTRG:TR2

[bit10] S22 : Trigger Enable for Operation of SMC<6*g+2>

This bit selects SMC<6*g+2> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+2> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+2> is triggered by SMCTGg_PTRG:TR2

[bit9] S21 : Trigger Enable for Operation of SMC<6*g+1>

This bit selects SMC<6*g+1> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+1> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+1> is triggered by SMCTGg_PTRG:TR2

[bit8] S20 : Trigger Enable for Operation of SMC<6*g+0>

This bit selects SMC<6*g+0> for the SMC trigger group S2 (triggered by SMCTGg_PTRG:TR2).

Bit	Description
0	SMC<6*g+0> is not triggered by SMCTGg_PTRG:TR2
1	SMC<6*g+0> is triggered by SMCTGg_PTRG:TR2

[bit5] S15 : Trigger Enable for Operation of SMC<6*g+5>

This bit selects SMC<6*g+5> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+5> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+5> is triggered by SMCTGg_PTRG:TR1

[bit4] S14 : Trigger Enable for Operation of SMC<6*g+4>

This bit selects SMC<6*g+4> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+4> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+4> is triggered by SMCTGg_PTRG:TR1

[bit3] S13 : Trigger Enable for Operation of SMC<6*g+3>

This bit selects SMC<6*g+3> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+3> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+3> is triggered by SMCTGg_PTRG:TR1

[bit2] S12 : Trigger Enable for Operation of SMC<6*g+2>

This bit selects SMC<6*g+2> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+2> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+2> is triggered by SMCTGg_PTRG:TR1

[bit1] S11 : Trigger Enable for Operation of SMC<6*g+1>

This bit selects SMC<6*g+1> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+1> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+1> is triggered by SMCTGg_PTRG:TR1

[bit0] S10 : Trigger Enable for Operation of SMC<6*g+0>

This bit selects SMC<6*g+0> for the SMC trigger group S1 (triggered by SMCTGg_PTRG:TR1).

Bit	Description
0	SMC<6*g+0> is not triggered by SMCTGg_PTRG:TR1
1	SMC<6*g+0> is triggered by SMCTGg_PTRG:TR1

4.2. SMC Trigger Register (SMCTGg_PTRG)

The SMC Trigger Register is used to generate trigger signals for the two trigger groups configured in the SMC Trigger Selection Register (SMCTGg_PTRGS). The trigger group can be used to synchronously start several SMC channels. The operation (count enable) starts depending on the delay configured in the SMCi_PTRGDL register.

REGISTER_NAME	SMCTGg_PTRG
OFFSET	0x00000002
ACCESS_SIZE	B, H
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved						TR2	TR1
ACCESS_TYPE	R0,WX						R0,W	R0,W
PROT_TYPE	Rp/Wp							
INITIAL_VALUE	0						0	0

[bit1] TR2 : SMC Trigger 2

This bit triggers the SMC trigger group S2 of SMCTGg. The configuration in SMCTGg_PTRGS:S2x determines which SMC channels are triggered. The trigger signal starts the delay counters (if enabled in SMCi_PTRGDL). If the delay has elapsed or the delay is disabled, the Count Enable (SMCi_PWC:CE) bits of the selected SMC channels will be set and the operation starts (PWM generation and output control).

Bit	Description
0	No effect
1	Triggers selected SMC channels via the Set logic for SMCi_PWC:CE

Notes:

- *This bit is cleared automatically to "0" after one clock cycle.*
- *A second trigger should not be applied to a group which has already been triggered. If this happens, the running delay counter will reset its value.*

[bit0] TR1 : SMC Trigger 1

This bit triggers the SMC trigger group S1 of SMCTGg. The configuration in SMCTGg_PTRGS:S1x determines which SMC channels are triggered. The trigger signal starts the delay counters (if enabled in SMCi_PTRGDL). If the delay has elapsed or the delay is disabled, the Count Enable (SMCi_PWC:CE) bits of the selected SMC channels will be set and the operation starts (PWM generation and output control).

Bit	Description
0	No effect
1	Triggers selected SMC channels via the Set logic for SMCi_PWC:CE

Notes:

- *This bit is cleared automatically to "0" after one clock cycle.*
- *A second trigger should not be applied to a group which has already been triggered. If this happens, the running delay counter will reset its value.*

CHAPTER 29: Sound Generator

This chapter explains the functions and operations of the Sound Generator(SG).

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

CODE: FS12C-E03.02

1. Overview

The Sound Generator generates and outputs a tone pulse signal (or mixed signal of a tone pulse and a PWM pulse) and a PWM pulse signal according to the setting by MCU. It is enabled to set the frequency of the tone pulse signal, sound volume (amplitude of PWM pulse), and length of sound. The Sound Generator consists of the following registers and counters.

- DMA Transfer Update Enable Register (SGDER)
- Sound Control Register (SGCR)
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Time Cycle Register (SGTCR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
- Extended Frequency Data Register (SGEFR)
- DMA Transfer Intermediate Register (SGDMAR)
- Interrupt Clear Register (SGCCR)
- PWM pulse generation counter
- Frequency counter
- Decrement counter
- Tone pulse counter

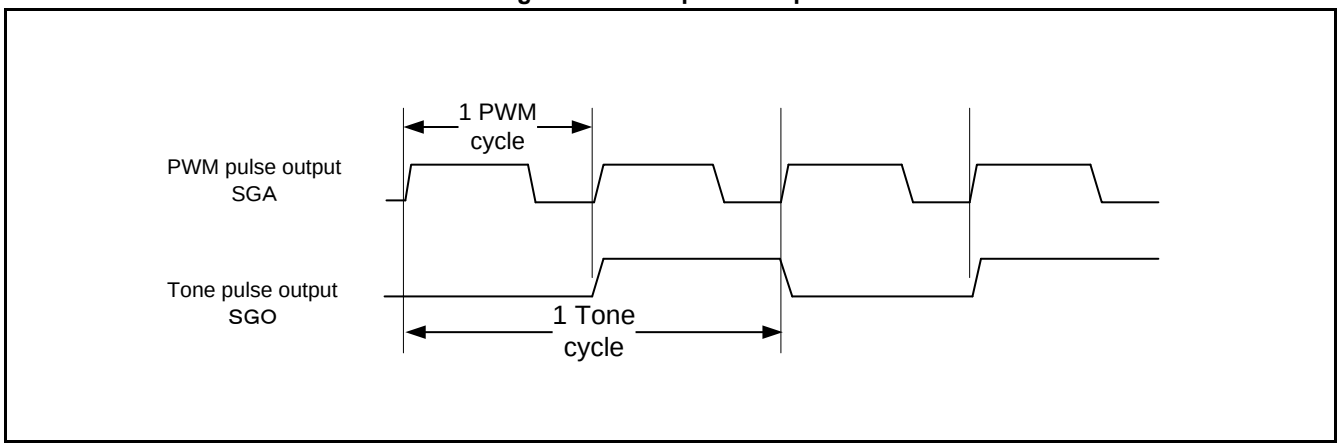
Functions of Sound Generator

Table 1-1 Functions of Sound Generator

No	Item	Function
1	Operation clock	Bus clock (BCLK)
2	Clock input	For clock input of the Sound Generator, divide the bus clock (BCLK) and use it. - 1 x BCLK - 1/2 x BCLK - 1/4 x BCLK - 1/8 x BCLK
3	Waveform	Rectangular wave for sound (tone pulse output, sound output from SGO pin)
4	Volume of sound	It is enabled to set their arbitrary level (PWM pulse output). (Amplitude output from SGA pin)
5	Frequency	It is enabled to set their arbitrary level for sound signal frequency. (Frequency setting and PWM cycle setting) The duty ratio (high side) of sound signal is programmable up to 50%. The high width of sound signal is controlled by the PWM cycle unit. (Extended Frequency setting)
6	Length of sound	It is enabled to set their arbitrary level.
7	Interrupt	- It is enabled to generate interrupt requests at the end of outputting sound at programmed length (An overflow of Tone pulse counter). - It is enabled to generate interrupts by writing "1" to the Start bit (SGCR.ST) when in DMA mode (SGCR.DMA="1").

Output of Sound Generator

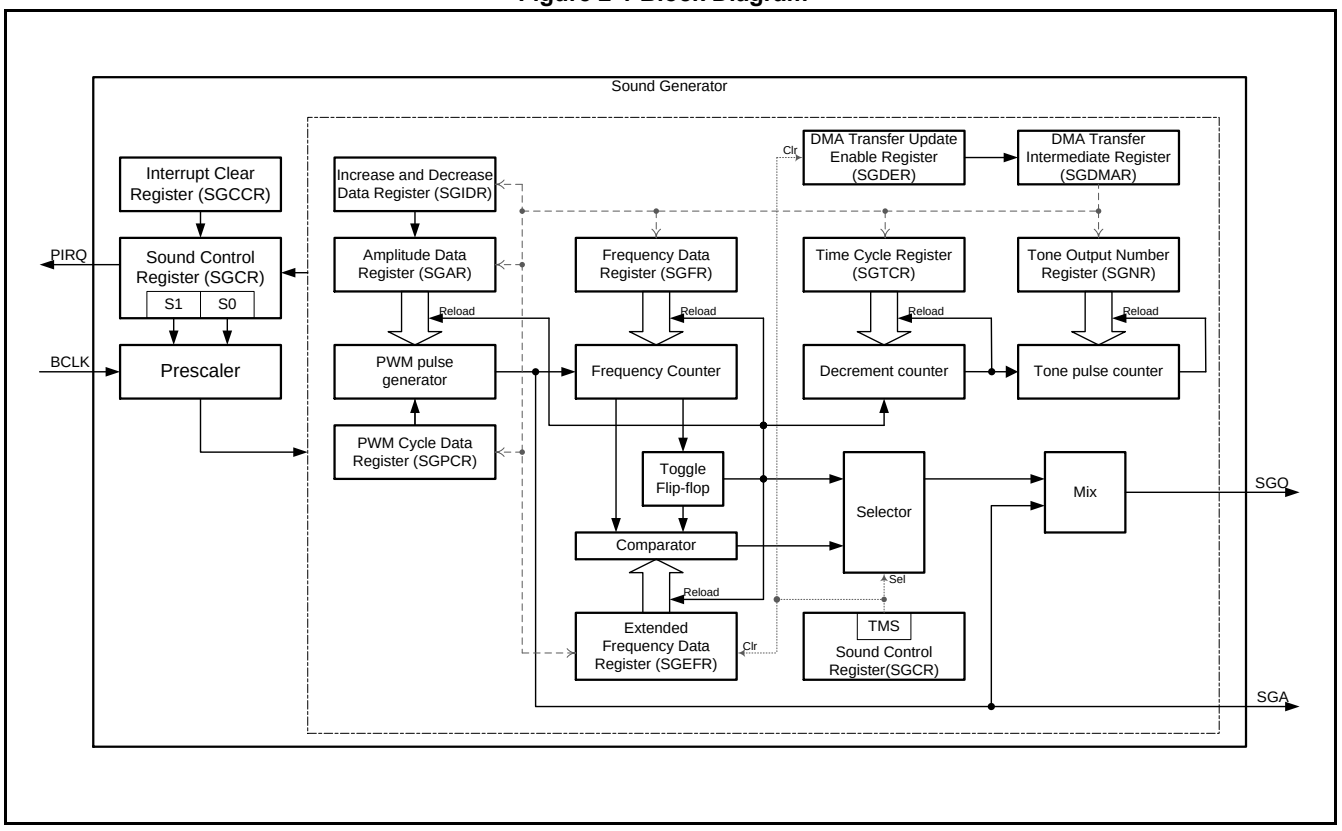
Figure 1-1 Example of Output



2. Configuration

This chapter shows the block diagram of Sound Generator.

Figure 2-1 Block Diagram

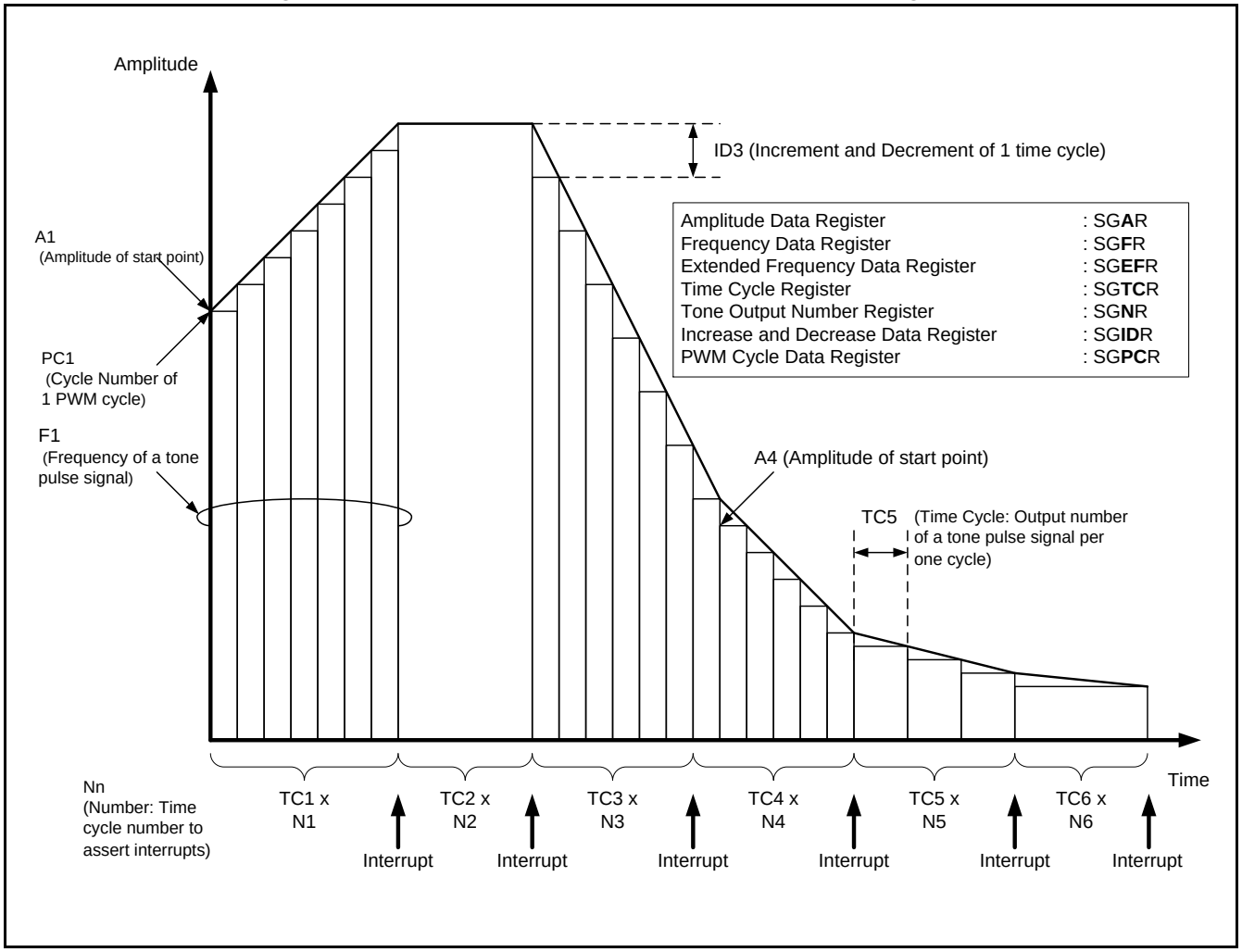


3. Operations

This chapter describes the Sound Generator operation.

The concept of Sound Generator operation is shown in the diagram as below.

Figure 3-1 The Operation of Sound Generator (conceptual diagram)



Set the various values in the register as follows:

Amplitude of start point (An)	- to the Amplitude Data Register (SGAR)
Frequency of a tone pulse signal (Fn)	- to the Frequency Data Register (SGFR)
Duty ratio of a tone pulse signal	- to the Extended Frequency Data Register (SGEFR)
Output number of a tone pulse signal per one cycle (TCn)	- to the Time Cycle Register (SGTCR)
Time cycle number to assert interrupts (Nn)	- to the Tone Output Number Register (SGNR)

Increment and Decrement of 1 time cycle (IDn)	- to the Increase and Decrease Data Register (SGIDR)
Cycle Number of 1 PWM cycle (PCn)	- to the PWM Cycle Data Register (SGPCR)
Other information about Sound Generator control	- to the Sound Control Register (SGCR)

The Sound Generator outputs the tone pulse signal and the amplitude data according to the setting above.

In the conceptual diagram shown above, it outputs 6 steps of signal.

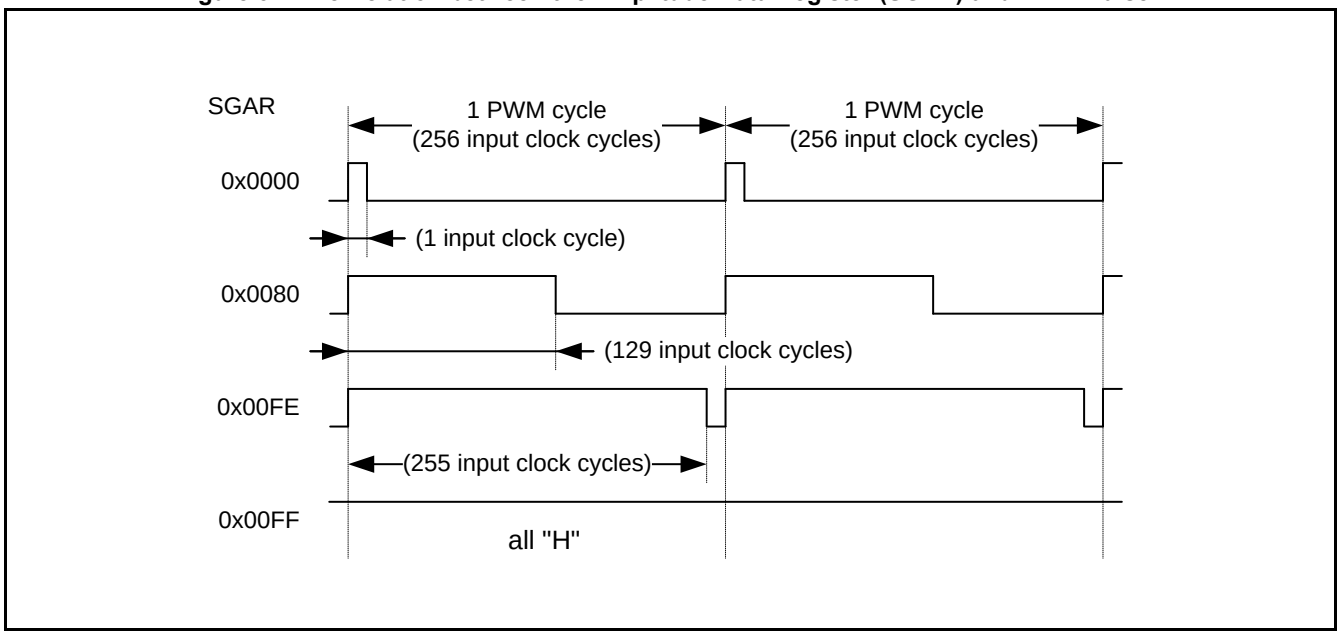
Each register, such as Amplitude Data Register, is supposed to be written at the following timing.

- Before starting signal output
- Each time after asserting an interrupt

3.1. Relation between the Amplitude Data Register (SGAR) and PWM Pulse

This section describes the relation between the Amplitude Data Register (SGAR) and a PWM pulse.

Figure 3-2 The Relation between the Amplitude Data Register (SGAR) and PWM Pulse



The amplitude data is output as a PWM (Pulse Width Modulation) waveform on the SGA pin.

The length of a PWM cycle is programmable in the PWM Cycle Data Register (SGPCR). This length is based on the count of input clock. The initial value is 256 input clock cycle (SGPCR=0x00FF).

The value [Amplitude Data Register (SGAR) + 1] means the number of input clock cycles in which the SGA pin is "H" during 1 PWM cycle. Moreover, when the Amplitude Data Register (SGAR) is greater than or equal to the PWM Cycle Data Register (SGPCR), the SGA pin output is always "H".

3.2. Relation between the Frequency Data Register (SGFR), the Extended Frequency Data Register (SGEFR) and Tone Pulse Signal

This section describes the relation between the Frequency Data Register (SGFR), the Extended Frequency Data Register (SGFR) and a tone pulse signal.

Sound generator has two modes of Normal mode and Extended mode for generation of a tone pulse signal.

The mode is selected by the Tone output mode select bit (TMS) of the Sound Control Register (SGCR).

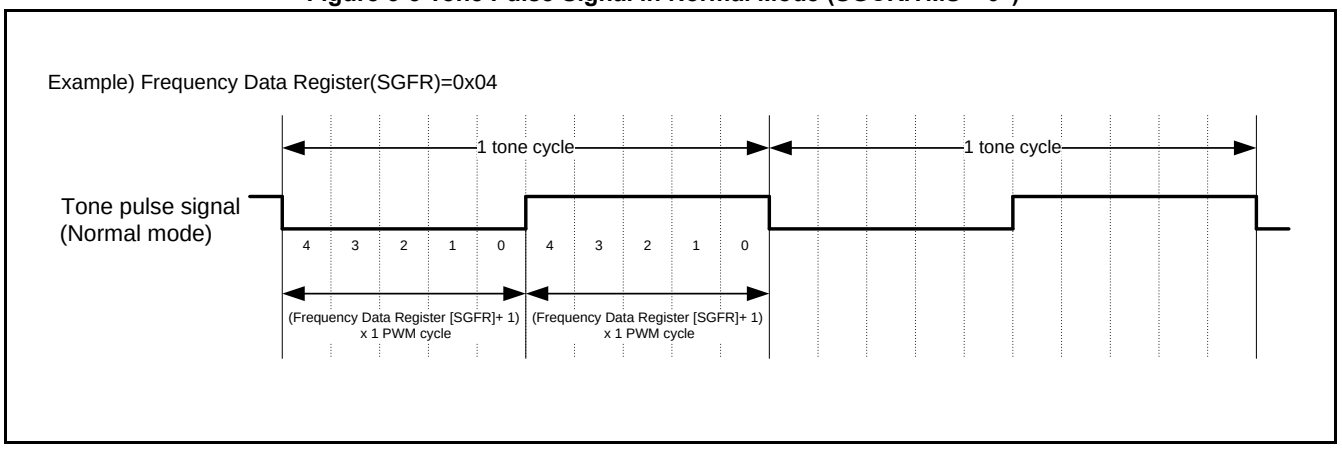
When the Tone output mode select bit (TMS) of the Sound Control Register (SGCR) is "0" (Normal mode), the duty ratio of a tone pulse signal is fixed to 50%.

When the Tone output mode select bit (TMS) of the Sound Control Register (SGCR) is "1" (Extended mode), the duty ratio (high side) of a tone pulse signal is programmable up to 50%.

3.2.1. Generation of a Tone Pulse Signal in Normal Mode

When in Normal mode (SGCR.TMS="0"), a sound generator generates a tone pulse signal by the Frequency Data Register (SGFR).

Figure 3-3 Tone Pulse Signal in Normal Mode (SGCR.TMS="0")



A tone cycle of a tone pulse signal is controlled by the Frequency Data Register (SGFR) and the PWM cycle.

A tone cycle is "(Frequency Data Register [SGFR] + 1) x 1 PWM cycle x 2".

The tone pulse signal transits between "L" and "H" at every cycle of "(Frequency Data Register [SGFR] + 1) x 1 PWM cycle)". This signal is generated by an internal toggle flip-flop.

The duty ratio of a tone pulse signal of Normal mode (SGCR.TMS="0") is fixed to 50%.

When the Tone output bit (SGCR.TONE) of the Sound Control Register (SGCR) is "0", the tone pulse signal is mixed with a PWM pulse, and is output from the SGO pin. And when the Tone output bit (SGCR.TONE) of the Sound Control Register (SGCR) is "1", the tone pulse signal is output without being mixed.

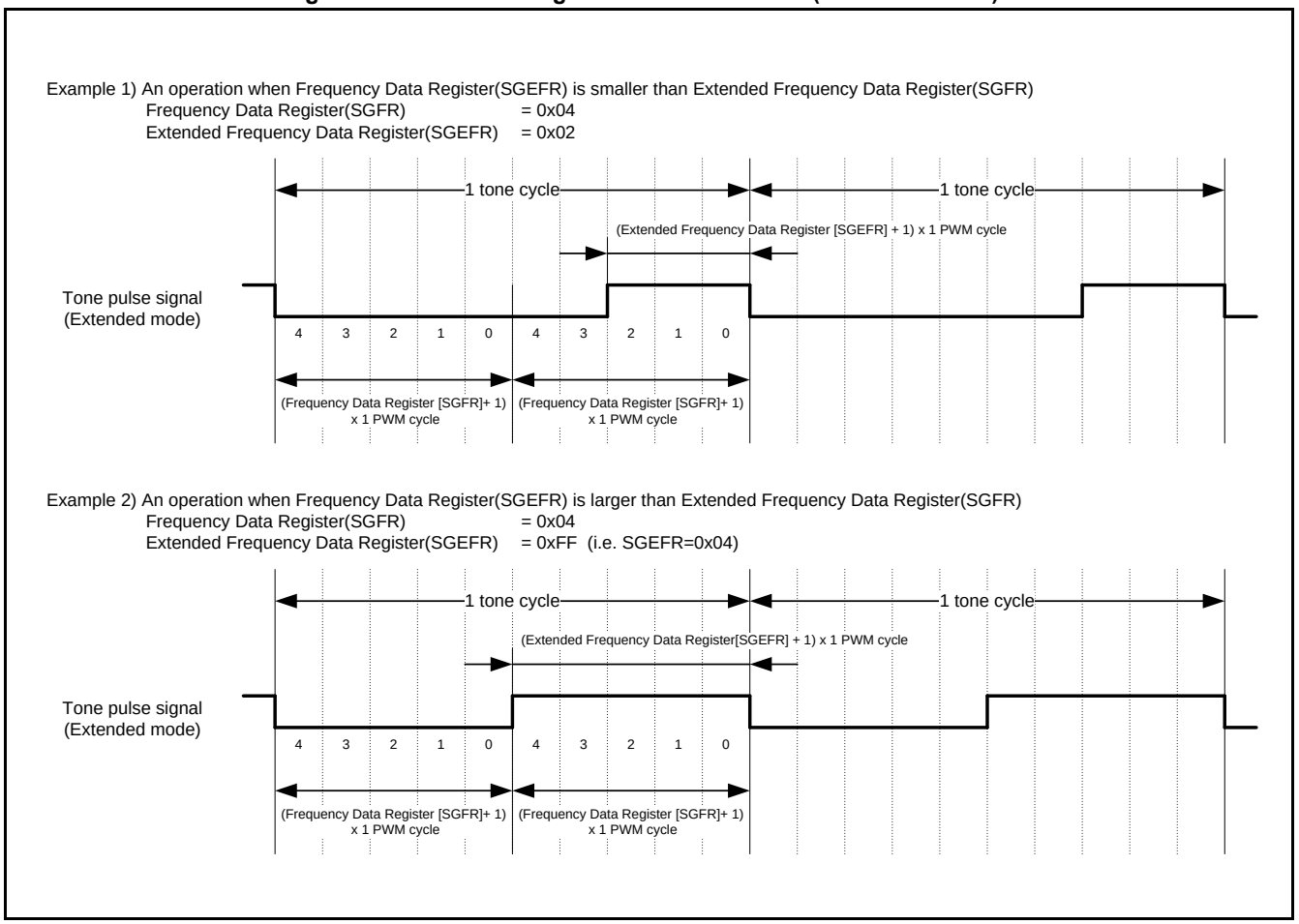
Note:

- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

3.2.2. Generation of a Tone Pulse Signal in Extended Mode

When in Extended mode (SGCR.TMS="1"), a sound generator generates a tone pulse signal by the Frequency Data Register (SGFR) and the Extended Frequency Data Register (SGEFR).

Figure 3-4 Tone Pulse Signal in Extended Mode (SGCR.TMS="1")



A tone cycle of a tone pulse signal is controlled by the Frequency Data Register (SGFR) and the PWM cycle.

A tone cycle is "(Frequency Data Register [SGFR] + 1) x 1 PWM cycle x 2".

The duty ratio (high side) of a tone pulse signal of Extended mode (SGCR.TMS="1") is programmable up to 50%.

The high width of a tone pulse signal is controlled by the PWM cycle unit, with the value of Extended Frequency Data Register (SGEFR). The high width is "(Extended Frequency Data Register [SGEFR] + 1) x1 PWM cycle".

When the Tone output bit (SGCR.TONE) of the Sound Control Register (SGCR) is "0", the tone pulse signal is mixed with a PWM pulse, and is output from the SGO pin. And when the Tone output bit (SGCR.TONE) of the Sound Control Register (SGCR) is "1", the tone pulse signal is output without being mixed.

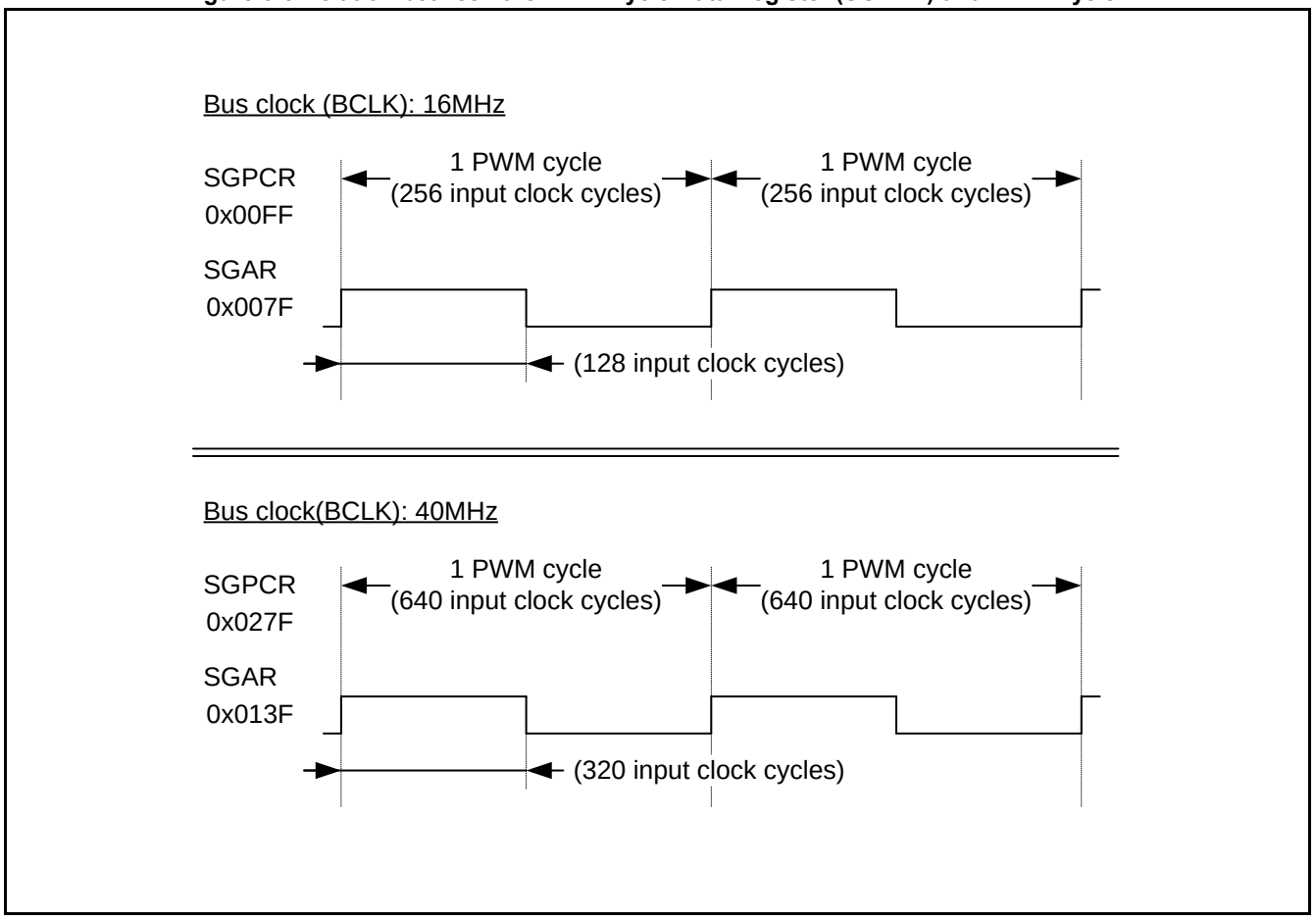
Notes:

- *When the Extended Frequency Data Register value (SGEFR) is larger than the Frequency Data Register value (SGFR) in Extended mode (SGCR.TMS="1"), the duty ratio is 50%.*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

3.3. Relation between the PWM Cycle Data Register (SGPCR) and PWM Cycle

This section describes the relation between the PWM Cycle Data Register (SGPCR) and PWM cycle.

Figure 3-5 Relation between the PWM Cycle Data Register (SGPCR) and PWM Cycle



The length of a PWM cycle is programmable in the PWM Cycle Data Register (SGPCR). This length is based on the count of input clock, and defined as "PWM Cycle Data Register [SGPCR] + 1". The input clock is created by dividing the bus clock.

The PWM cycle is a reference clock for a tone pulse signal (or a mixed signal of the tone pulse signal and a PWM pulse signal), and a PWM pulse signal.

To generate the same sound output both by the bus clock at 16 MHz and 40 MHz, for example, it can be made by changing the values of the PWM Cycle Data Register (SGPCR) and the Amplitude Data Register (SGAR). Because the bus clock proportion of 16 MHz and 40 MHz is 1:2.5, the same sound can be made by setting the values of these two registers to 1:2.5.

3.4. Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings

This section describes the relation between the DMA Transfer Update Enable Register (SGDER) and DMA settings.

3.4.1. The Number of DMA Transfers

DMA Transfer Update Enable Register (SGDER) determines the number of DMA transfer.

Table 3-1 The Relations between the Setting of the DMA Transfer Update Enable Register (SGDER) and the Number of DMA Transfer

No.	Setting of SGDER			The Number of DMA Transfers
	EFRE	ARE1, ARE0, FRE, NRE	TCRE, IDRE,PCRE1,PCRE0	
1	0	all"0"	all"0"	0
2	0	all"0"	except all"0"	1
3	0	except all"0"	all"0"	1
4	0	except all"0"	except all"0"	2
5	1	all"0"	all"0"	1
6	1	all"0"	except all"0"	2
7	1	except all"0"	all"0"	2
8	1	except all"0"	except all"0"	3

Note:

- *In Normal mode (SGCR.TMS="0"), the maximum number of DMA transfer is two. (i.e. SGDER.EFRE is fixed to "0".)*

3.4.2. DMA Transfer Size

The DMA transfer size (1 byte, 2 bytes or 4 bytes) depends on the setting of the DMA Transfer Update Enable Register (SGDER).

The greater value of the following three becomes the DMA transfer size:

The value of SGDER.EFRE,

The value of SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE,

The value of SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0

The transfer size of 3 bytes is regarded as 4 bytes.

3.4.3. Transfer Byte Position in the DMA Transfer Intermediate Register (SGDMAR)

The DMA transfer byte position in the "DMA Transfer Intermediate Register (SGDMAR)" depends on the setting of the "DMA Transfer Update Enable Register (SGDER)" and the DMA transfer size.

If the size of a DMA transfer is less than 4 bytes, the byte position of DMA transfer is left-aligned.

The table below shows the relations between
"the setting of the DMA Transfer Update Enable Register (SGDER)"
and
"the byte position in the DMA Transfer Intermediate Register (SGDMAR)".
The byte position in the DMA Transfer Intermediate Register (SGDMAR) corresponds to the selection of the following registers.

- Amplitude Data Register (SGAR [15:0])
- Frequency Data Register (SGFR [7:0])
- Tone Output Number Register (SGNR [7:0])

The transfer size #1 is calculated by the setting of {SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE} in the "DMA Transfer Update Enable Register (SGDER)". When this transfer size #1 is not 4 bytes, the transfer byte position is left-aligned.

Table 3-2 The Relations between the Setting of the DMA Transfer Update Enable Register (SGDER) and the Transfer Byte Position of SGDMAR #1

No.	Setting of SGDER				Transfer size #1 *1	Transfer Byte Position of SGDMAR			
	ARE1	ARE0	FRE	NRE		SGAR [15:8]	SGAR [7:0]	SGFR [7:0]	SGNR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

*1: The transfer size which is calculated by the setting of SGDER.ARE1, SGDER.ARE0, SGDER.FRE, and SGDER.NRE.

- : Do not care.

The table below shows the relations between
"the setting of the DMA Transfer Update Enable Register (SGDER)"
and
"the byte position in the DMA Transfer Intermediate Register (SGDMAR)".
The byte position in the DMA Transfer Intermediate Register (SGDMAR) corresponds to the selection of the following registers.
- Time Cycle Register (SGTCR[7:0])
- Increase and Decrease Data Register (SGIDR[7:0])
- PWM Cycle Data Register (SGPCR[15:0])

The transfer size #2 is calculated by the setting of {SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0} in the DMA Transfer Update Enable Register (SGDER). When this transfer size #2 is not 4 bytes, the transfer byte position is left-aligned.

Table 3-3 The Relations between the Setting of the DMA Transfer Update Enable Register (SGDER) and Transfer Byte Position of SGDMAR #2

No.	Setting of SGDER				Transfer size #2 *1	Transfer Byte Position of SGDMAR			
	TCRE	IDRE	PCRE1	PCRE0		SGTCR [7:0]	SGIDR [7:0]	SGPCR [15:8]	SGPCR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

*1: The transfer size which is calculated by the setting of SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, and SGDER.PCRE0.

- : Do not care.

The table below shows the relations between
"the setting of the DMA Transfer Update Enable Register (SGDER)"
and
"the byte position in the DMA Transfer Intermediate Register (SGDMAR)".
The byte position in the DMA Transfer Intermediate Register (SGDMAR) corresponds to the selection of the following register.

- Extended Frequency Data Register (SGEFR [7:0])

The transfer size #3 is calculated by the setting of {SGDER.EFRE} in the "DMA Transfer Update Enable Register (SGDER)". When this transfer size #3 is not 4 bytes, the transfer byte position is left-aligned.

Table 3-4 The Relations between the Setting of the DMA Transfer Update Enable Register (SGDER) and the Transfer Byte Position of SGDMAR #3

No.	Setting of SGDER				Transfer size #3 *1	Transfer Byte Position of SGDMAR			
	-	-	-	EFRE		-	-	-	SGEFR [7:0]
1	0	0	0	0	0	-	-	-	-
2	0	0	0	1	1	-	-	-	SGDMAR [31:24]

*1: The transfer size which is calculated by the setting of SGDER.EFRE.

- : Do not care.

3.4.4. DMA Transfer Image

This section shows an example of DMA transfer image, under the setting of DMA Transfer Update Enable Register (SGDER):

[SGDER.EFRE]	="1"
[SGDER.ARE1, SGDER.ARE0, SGDER.FRE, SGDER.NRE]	="1001"
[SGDER.TCRE, SGDER.IDRE, SGDER.PCRE1, SGDER.PCRE0]	="0100"

Condition:

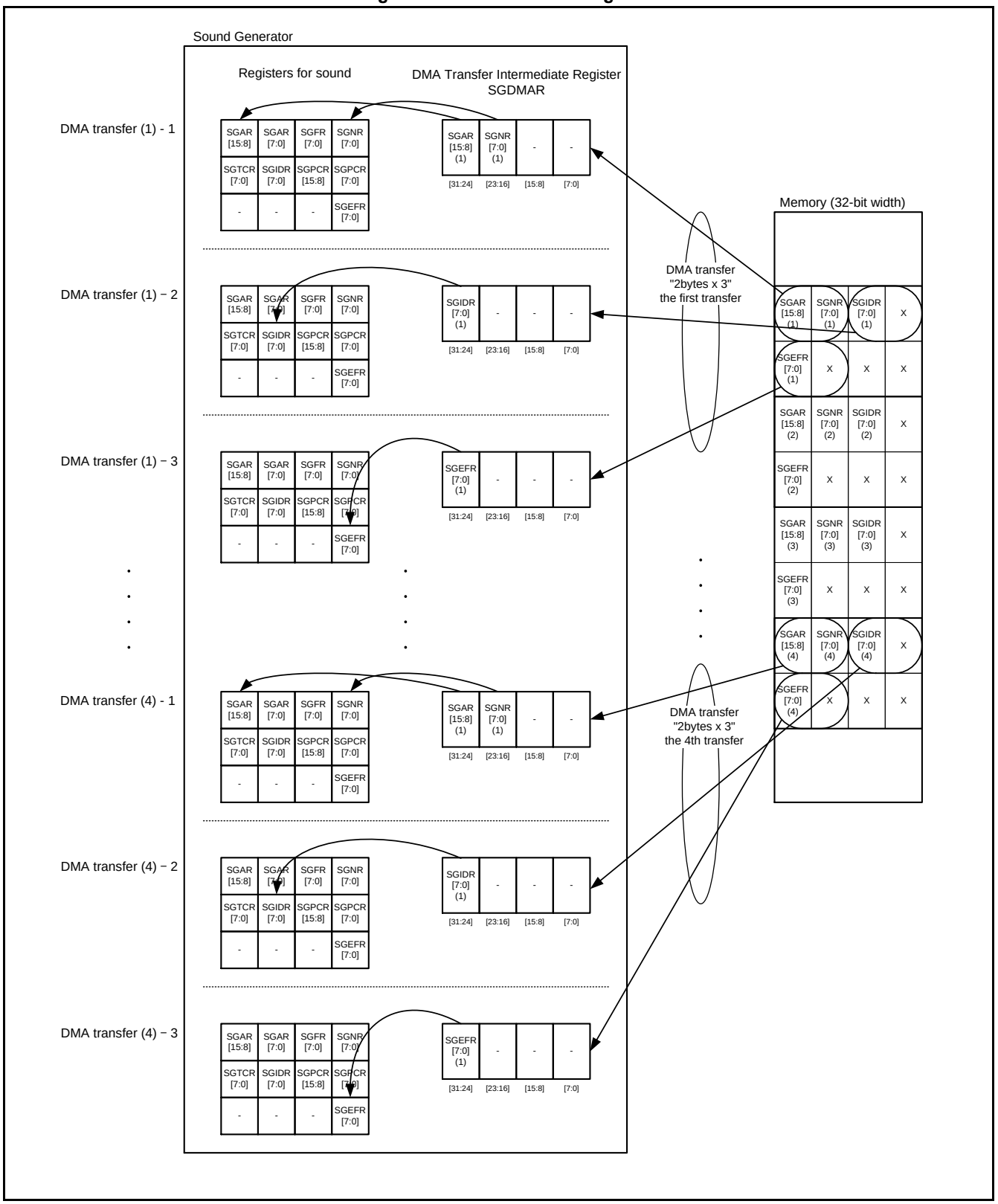
The number of times of DMA transfer : 3 times

DMA transfer size : 2 bytes

Transfer byte position of the DMA Transfer Intermediate Register (SGDMAR):

The first	SGDMAR[31:24]	← The amplitude data (upper byte) / SGAR[15:8]
	SGDMAR[23:16]	← The tone output number / SGNR[7:0]
	SGDMAR[15:0]	← Do not care
The second	SGDMAR[31:24]	← The increment and decrement data / SGIDR[7:0]
	SGDMAR[23:0]	← Do not care
The third	SGDMAR[31:24]	← The extended frequency data / SGEFR[7:0]
	SGDMAR[23:0]	← Do not care

Figure 3-6 DMA Transfer Image

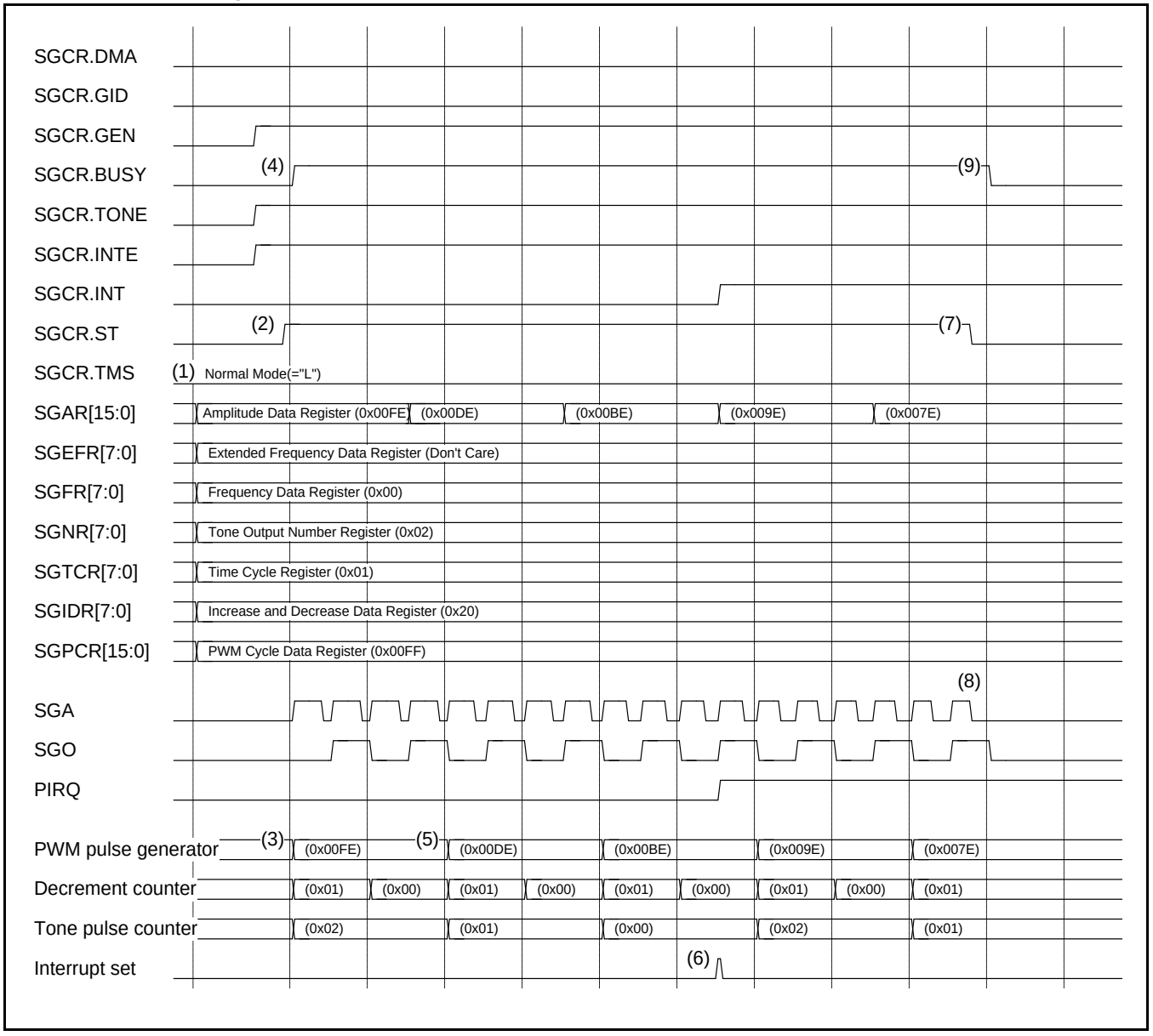


3.5. Sound Generator Operation

This section describes the Sound Generator operation as below

3.5.1. Sound Generator Operation in Normal Mode

Figure 3-7 Sound Generator Operation in Normal Mode (SGCR.TMS="0")



(1) The reload values are written to the Amplitude Data Register (SGAR), the Frequency Data Register (SGFR), the Tone Output Number Register (SGNR), and the Time Cycle Register (SGTCR) by software. Also the selection of increment and

decrement of the amplitude is written to the Increase and Decrease Data Register (SGIDR) by software as well as the number of cycles in 1 PWM cycle to the PWM Cycle Data Register (SGPCR).

Moreover, set other information to the Sound Control Register (SGCR) to control the Sound Generator. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Write "1" to the Start bit (SGCR.ST).

(3) By setting "1" to the Start bit (SGCR.ST), the Amplitude Data Register (SGAR) value is loaded into the PWM pulse generator, the Frequency Data Register (SGFR) value into the Frequency counter, the Tone Output Number Register (SGNR) value into the Tone pulse counter, the Time Cycle Register (SGTCR) value into the Decrement counter.

(4) The operation flag (SGCR.BUSY) is automatically set.

(5) Due to the operation that the Decrement counter counts the number of tone pulses until it reaches the reload value, the Amplitude Data Register (SGAR) value decreases according to the setting of the Automatic increase/decrease enable bit (SGCR.GEN) and the Increase/decrease setting bit (SGCR.GID).

(6) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, it sets the Interrupt status bit (SGCR.INT) and asserts the Interrupt request (PIRQ).
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(7) Write "0" to the Start bit (SGCR.ST). The Sound Generator keeps operating until the Busy status bit (SGCR.BUSY) turns "0".

(8) The Sound Generator stops when the current tone cycle finishes.

(9) The Busy status bit (SGCR.BUSY) turns "0".

DMA transfer start interrupt setting enable bit (SGCR.DMA)

To assert the first Interrupt request (PIRQ) after starting (SGCR.ST="1"), there are two modes for the request. And the selection depends on the setting of the "DMA transfer start interrupt setting enable bit (SGCR.DMA)".

- Normal mode :
  When Sound Generator outputs the tone pulses, to the number programmed in the Time Cycle Register (SGTCR)

- DMA mode :
  Immediately after the setting of Start bit (writing "1" to SGCR.ST)
  (In this case, the Interrupt request (PIRQ) is regarded as the DMA transfer request.)

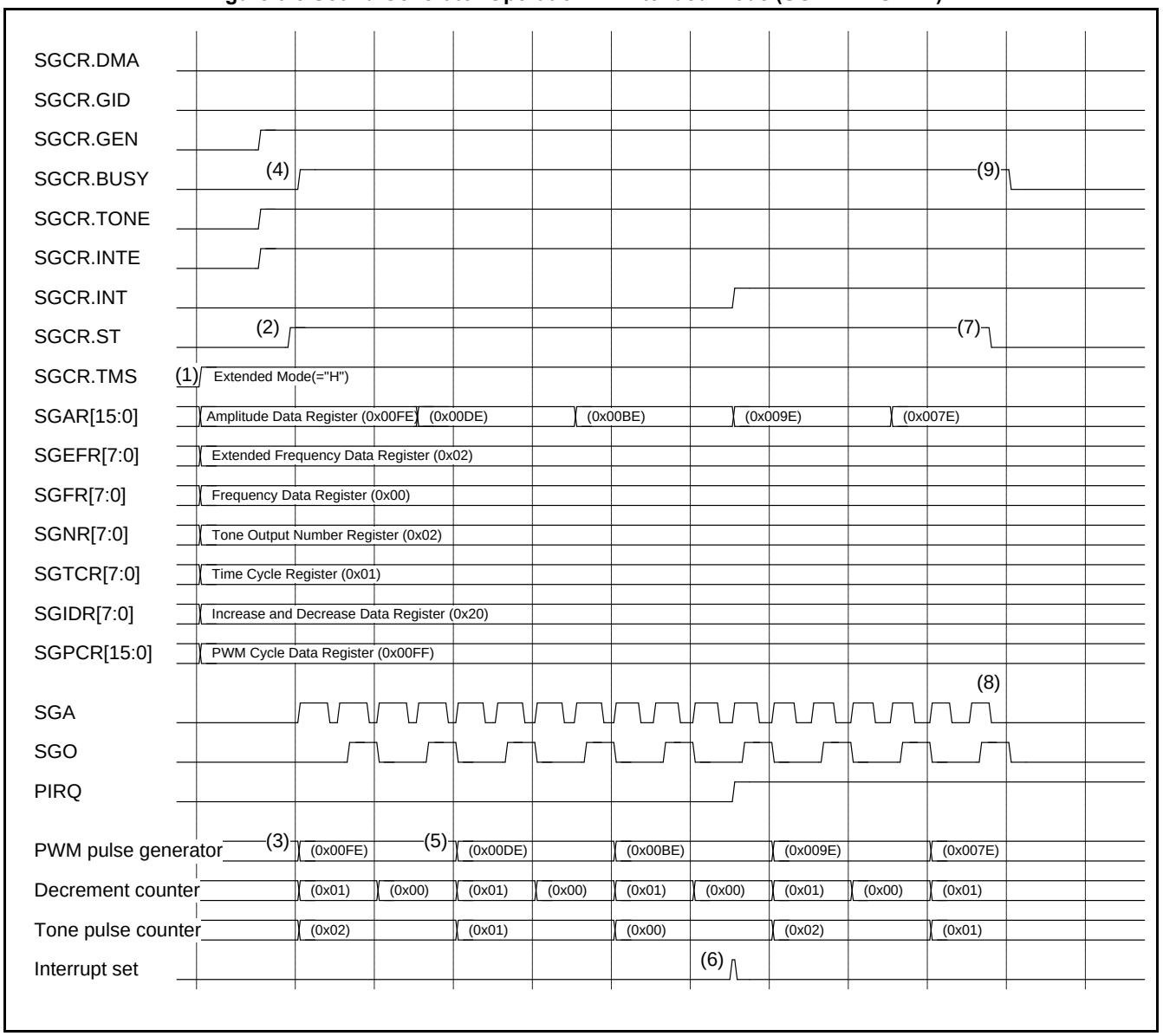
DMA transfer

The following registers are set through the DMA Transfer Intermediate Register (SGDMAR).
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Time Cycle Register (SGTCR)

- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

3.5.2. Sound Generator Operation in Extended Mode

Figure 3-8 Sound Generator Operation in Extended Mode (SGCR.TMS="1")



(1) The reload values are written to the Amplitude Data Register (SGAR), the Frequency Data Register (SGFR), the Tone Output Number Register (SGNR), and the Time Cycle Register (SGTCR) by software. Comparison value of high width of a tone pulse signal is written in the Extended Frequency Data Register (SGEFR) by software. Also the selection of increment and decrement of the amplitude is written to the Increase and Decrease Data Register (SGIDR) by software as well as the number of cycles in 1 PWM cycle to the PWM Cycle Data Register (SGPCR).

Moreover, set other information to the Sound Control Register (SGCR) to control the Sound Generator. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Write "1" to the Start bit (SGCR.ST).

(3) By setting "1" to the Start bit(SGCR.ST), the Amplitude Data Register (SGAR) value is loaded into the PWM pulse generator, the Frequency Data Register (SGFR) value into the Frequency counter, the Extended Frequency Data Register (SGEFR) value into the comparison register of high width of a tone pulse signal, the Tone Output Number Register (SGNR) value into the Tone pulse counter, the Time Cycle Register (SGTCR) value into the Decrement counter.

(4) The operation flag (SGCR.BUSY) is automatically set.

(5) Due to the operation that the Decrement counter counts the number of tone pulses until it reaches the reload value, the Amplitude Data Register (SGAR) value decreases according to the setting of the Automatic increase/decrease enable bit (SGCR.GEN) and the Increase/decrease setting bit (SGCR.GID).

(6) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, it sets the Interrupt status bit (SGCR.INT) and asserts the Interrupt request (PIRQ).
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(7) Write "0" to the Start bit (SGCR.ST). The Sound Generator keeps operating until the Busy status bit (SGCR.BUSY) turns "0".

(8) The Sound Generator stops when the current tone cycle finishes.

(9) The Busy status bit (SGCR.BUSY) turns "0".

DMA transfer start interrupt setting enable bit (SGCR.DMA)

To assert the first Interrupt request (PIRQ) after starting (SGCR.ST="1"), there are two modes for the request. And the selection depends on the setting of the "DMA transfer start interrupt setting enable bit (SGCR.DMA)".

- Normal mode :
  When Sound Generator outputs the tone pulses, to the number programmed in the Time Cycle Register (SGTCR)

- DMA mode :
  Immediately after the setting of Start bit (writing "1" to SGCR.ST)
  (In this case, the Interrupt request (PIRQ) is regarded as the DMA transfer request.)

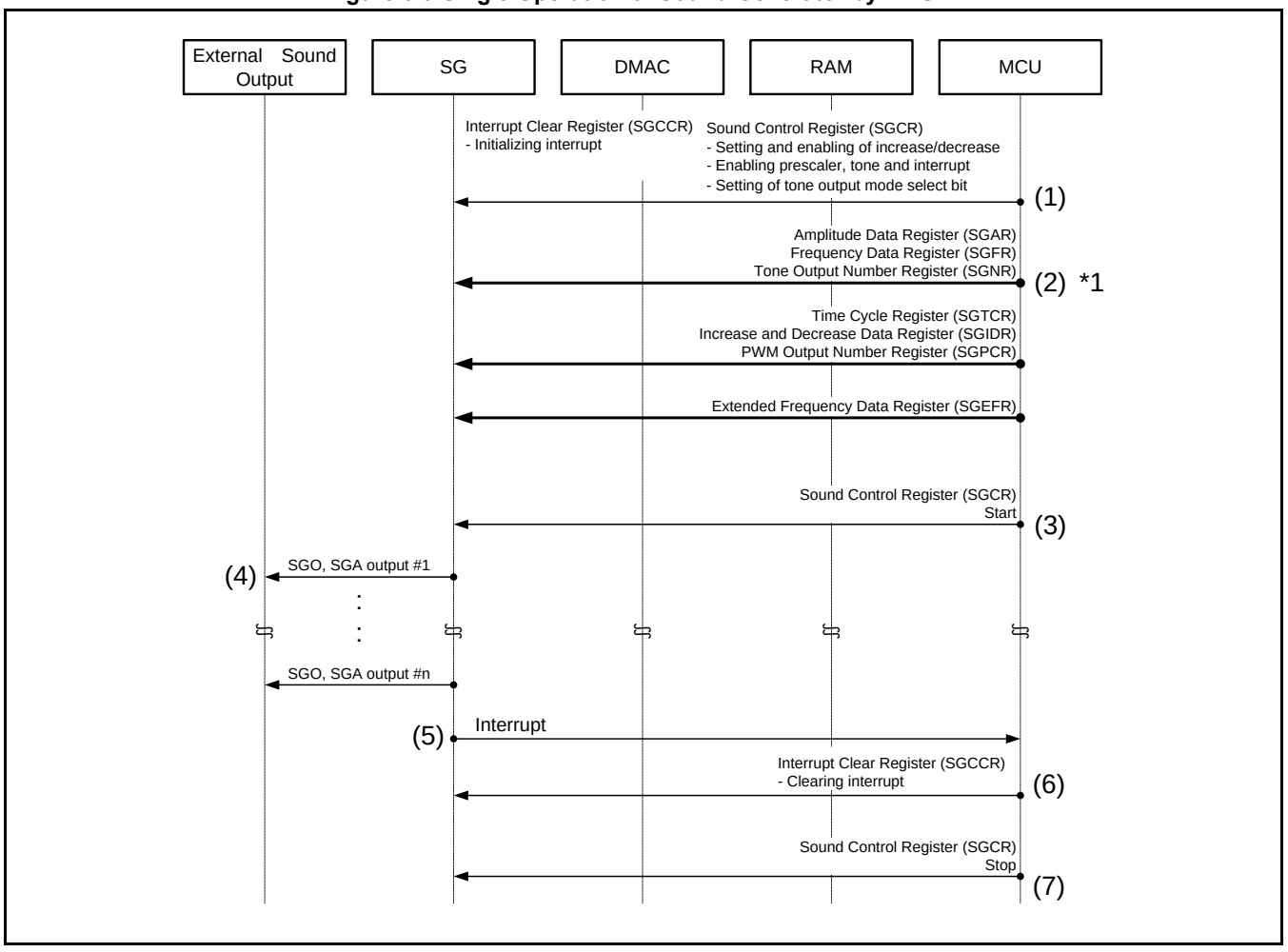
DMA transfer

The following registers are set through the DMA Transfer Intermediate Register (SGDMAR).
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Time Cycle Register (SGTCR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
- Extended Frequency Data Register (SGEFR)

3.5.3. Single Operation of Sound Generator by MCU

This section shows the single operation flow of the Sound Generator by MCU.

Figure 3-9 Single Operation of Sound Generator by MCU



(1) Set the information to control the Sound Generator to the Sound Control Register (SGCR) by software. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)", "PWM Cycle Data Register (SGPCR)" and "Extended Frequency Data Register (SGEFR)" by software. (*1: Set only the necessary registers.)

(3) Write "1" to the Start bit (SGCR.ST).

(4) The outputs of SGO and SGA start.

(5) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00

- Decrement counter is 0x00
- At the rising edge of SGO

(6) MCU clears the interrupt.

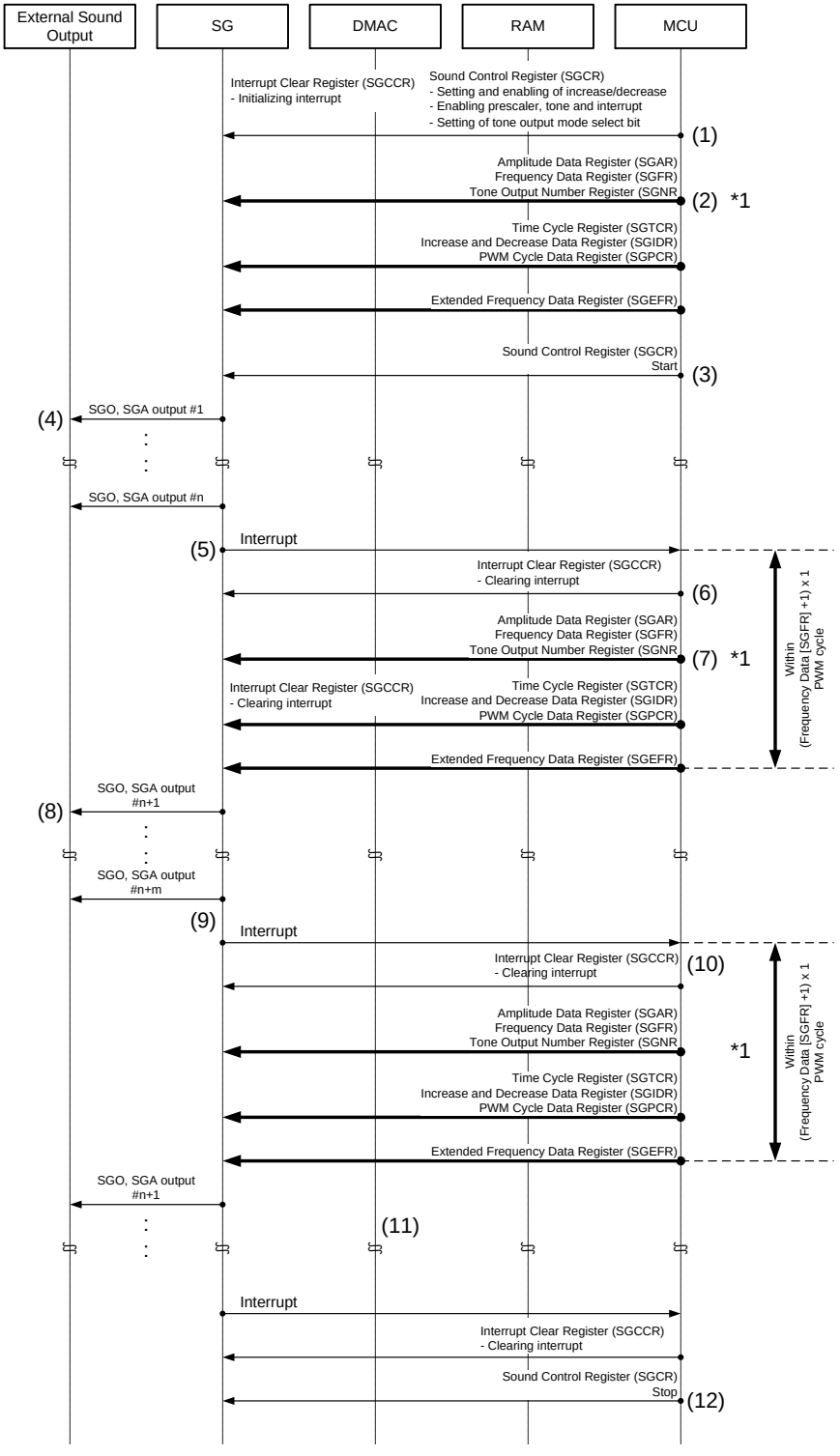
(7) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

3.5.4. Continuous Operation of Sound Generator by MCU

This section shows the continuous operation flow of the Sound Generator by MCU.

The flow from (7) is different from that of the single operation.

Figure 3-10 Continuous Operation of Sound Generator by MCU



(1) Set the information to control the Sound Generator to the Sound Control Register (SGCR) by software. Initialize the Interrupt status bit (SGCR.INT) and set the Interrupt enable bit (SGCR.INTE).

(2) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)", "PWM Cycle Data Register (SGPCR)" and "Extended Frequency Data Register (SGEFR)" by software. (*1: Set only the necessary registers.)

(3) Write "1" to the Start bit (SGCR.ST).

(4) The outputs of SGO and SGA start.

(5) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(6) MCU clears the interrupt.

(7) Set registers "Amplitude Data Register (SGAR)", "Frequency Data Register (SGFR)", "Tone Output Number Register (SGNR)", "Time Cycle Register (SGTCR)", "Increase and Decrease Data Register (SGIDR)", "PWM Cycle Data Register (SGPCR)" and "Extended Frequency Data Register (SGEFR)" by software. (*1: Set only the necessary registers.)

(8) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) MCU clears the interrupt.

(11) Repeat the flow from 7 to 11 to continue outputting the sound.

(12) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

Notes:

- *The software must finish the procedure from step 5 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *In case of switching the selection of increase/decrease, it is needed to write "Increase/decrease setting bit (SGCR.GID)" and the "Automatic increase/decrease enable bit (SGCR.GEN)" in the Sound Control Register (SGCR) within above-mentioned limit time.*

3.5.5. Sound Generator Operation with DMA

This section shows the flow of the Sound Generator operation with DMA.

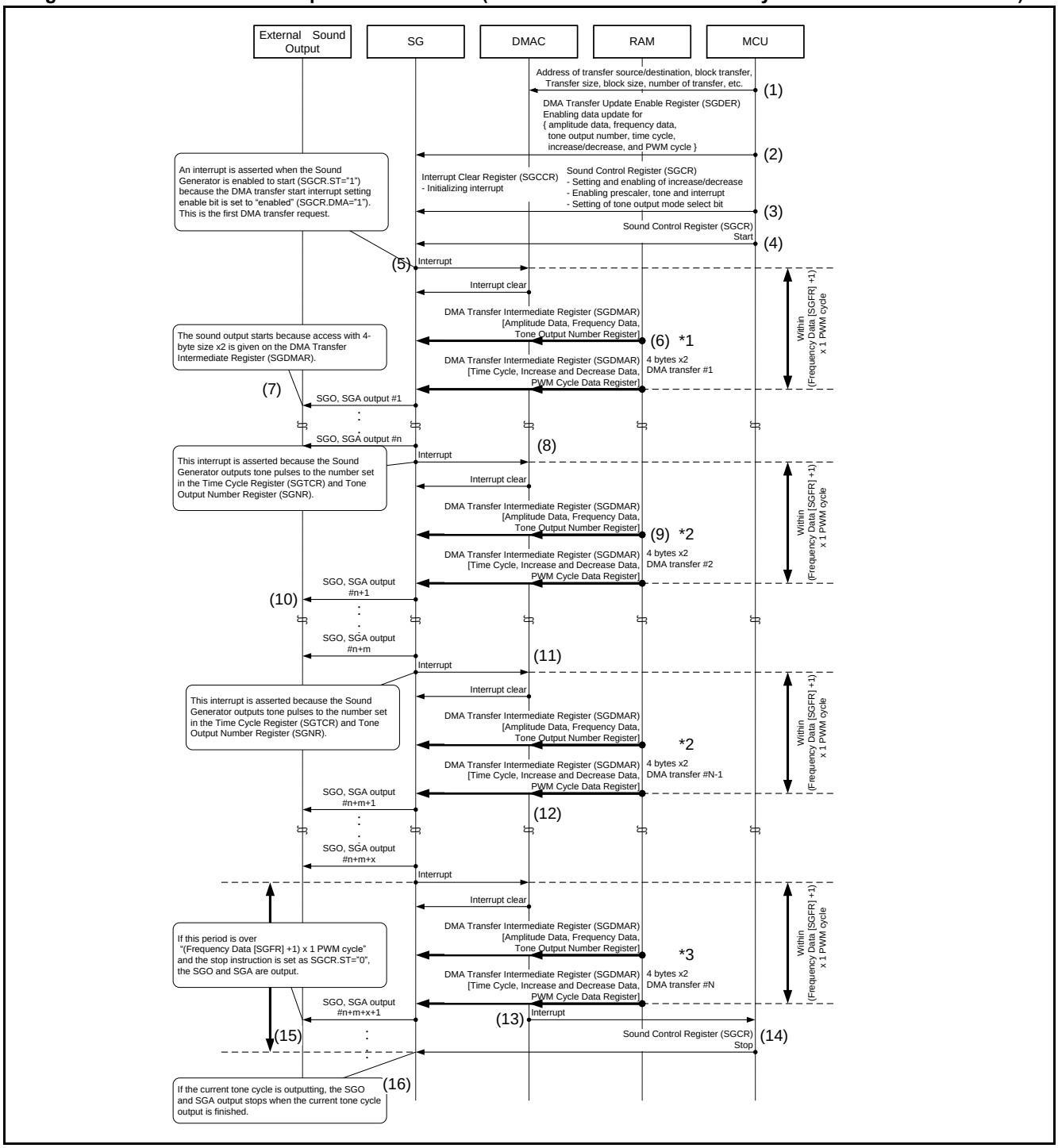
DMAC writes a register related to DMA in the Sound Generator. The timing of the first interrupt with DMA is different from the flow by MCU. The DMAC writes functional registers in the Sound Generator, through the DMA Transfer Intermediate Register (SGDMAR) which works as a window register.

Note:

- *Software needs to set the Interrupt enable bit (SGCR.INTE="1") in order to use the Interrupt request (PIRQ) as a DMA transfer request.*

3.5.5.1 In Case of DMA Transfer with 4-byte Size x 2 is Made N times

Figure 3-11 Sound Generator Operation with DMA (when in DMA transfer with 4-byte Size x 2 is Made N times)



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)
- Tone output mode select bit (SGCR.TMS) to "0"(Normal mode)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*1: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 4-byte size x2, N times).

(12) Repeat the flow from 9 to 11 to continue outputting the sound.

(13) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(14) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.
(15) When above operation (14) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*3: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

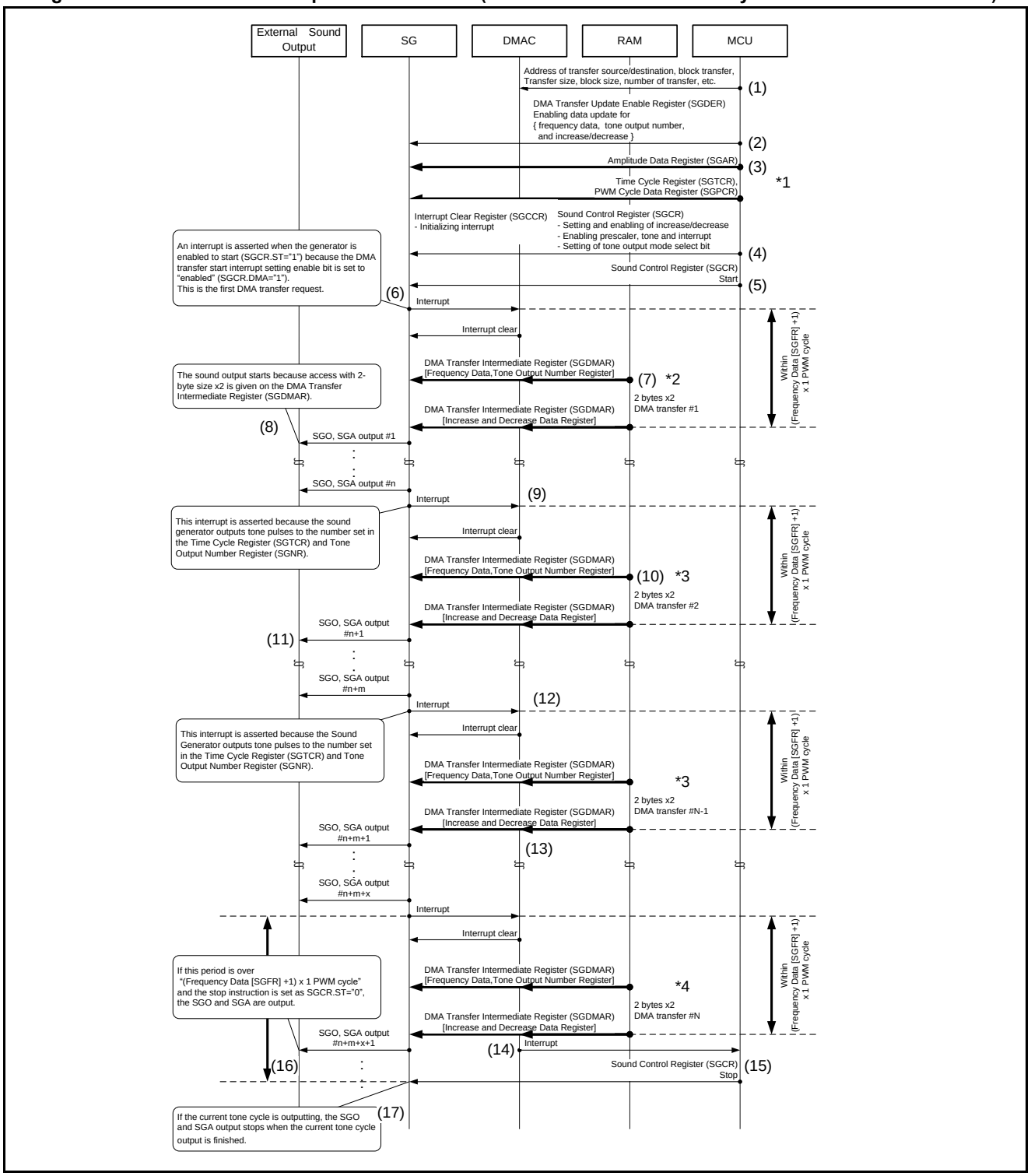
(16) If above (15) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*3: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.5.5.2 In Case of DMA Transfer with 2-byte Size x 2 is Made N Times

Figure 3-12 Sound Generator Operation with DMA (When DMA Transfer with 2-byte Size x 2 is Made N Times)



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "2-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Increase and Decrease Data Register (SGIDR)

(3) Software configures to the following registers by a DMA transfer that are not updated.
- Amplitude Data Register (SGAR)
- Time Cycle Register (SGTCR)
- PWM Cycle Data Register (SGPCR)
(*1: Set values to these registers which are not updated by the DMA Transfer Update Enable Register (SGDER).)

(4) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)
- Tone output mode select bit (SGCR.TMS) to "0"(Normal mode)

(5) Write "1" to the Start bit (SGCR.ST).

(6) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(7) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
[2nd step]
- Increase and Decrease Data Register (SGIDR)
(*2: DMA block size must to be "2-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(8) The outputs of SGO and SGA start, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*3: DMA block size must to be "2-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(11) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(12) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO
DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 2-byte size x2, N times).

(13) Repeat the flow from 10 to 12 to continue outputting the sound.

(14) When DMAC completes all DMA transfer (with 2-byte size x2, N times), it asserts an interrupt to MCU.

(15) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(16) When above operation (15) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

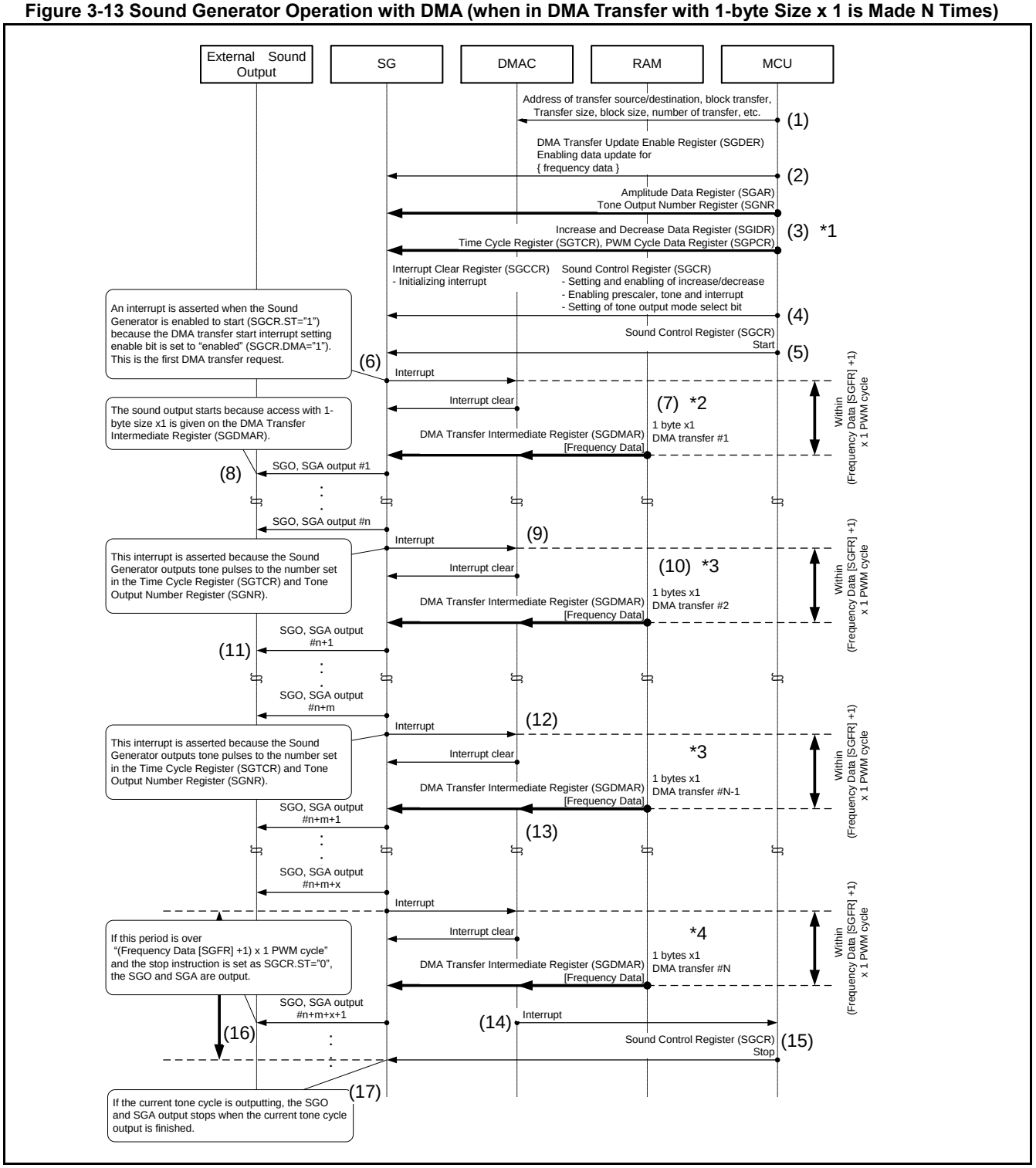
(17) If above (16) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 6 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.5.5.3 In Case of DMA Transfer with 1-byte Size x 1 is Made N Times

Figure 3-13 Sound Generator Operation with DMA (when in DMA Transfer with 1-byte Size x 1 is Made N Times)



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "1-byte size x 1", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)

(3) Software configures to the following registers by a DMA transfer that are not updated.
- Amplitude Data Register (SGAR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
(*1: Set values to these registers which are not updated by the DMA Transfer Update Enable Register (SGDER).)

(4) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)
- Tone output mode select bit (SGCR.TMS) to "0"(Normal mode)

(5) Write "1" to the Start bit (SGCR.ST).

(6) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(7) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following register.
- Frequency Data Register (SGFR)
(*2: DMA block size must to be "1-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(8) The outputs of SGO and SGA start, according to the register settings above.

(9) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(10) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*3: DMA block size must to be "1-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(11) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(12) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00

- At the rising edge of SGO
DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 1-byte size x1, N times).

(13) Repeat the flow from 10 to 12 to continue outputting the sound.

(14) When DMAC completes all DMA transfer (with 1-byte size x1, N times), it asserts an interrupt to MCU.

(15) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(16) When above operation (15) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

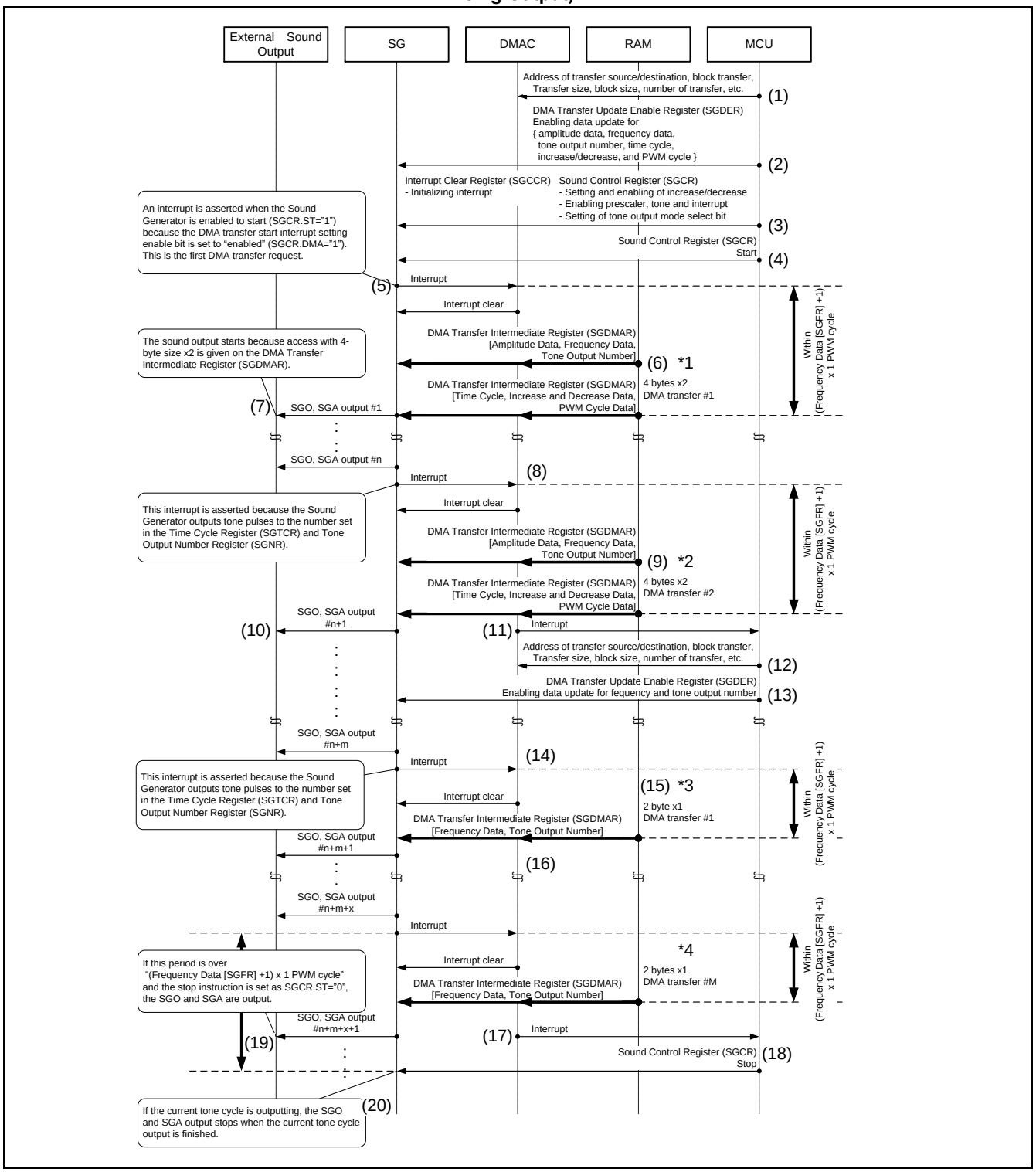
(17) If above (16) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 6 to 7, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.5.5.4 In Case of DMA Transfer with 4-byte Size x 2 is Made N Times and DMA Transfer with 2-byte Size x 1 is Made M Times (The Case the Transfer Byte Size is Changed While Sound Outputting)

Figure 3-14 Sound **Generator Operation with DMA (The Case the Transfer Byte Size is Changed While Sound is Being Output)**



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)
- Tone output mode select bit (SGCR.TMS) to "0"(Normal mode)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*1: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(12) Software makes settings to those registers which are needed in DMA transfer. The DMA transfer is based on a block data of "2-byte size x 1", and this block can be repeated M times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(13) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

(14) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

(15) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*3: DMA block size must to be "2-byte size x 1" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(16) Repeat the flow from 14 to 15 to continue outputting the sound.

(17) When DMAC completes all DMA transfer (with 2-byte size x1, M times), it asserts an interrupt to MCU.

(18) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(19) When above operation (18) was made within the following time, the Mth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*4: The data of the Mth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Mth DMA transfer only to assert an interrupt toward the MCU.)

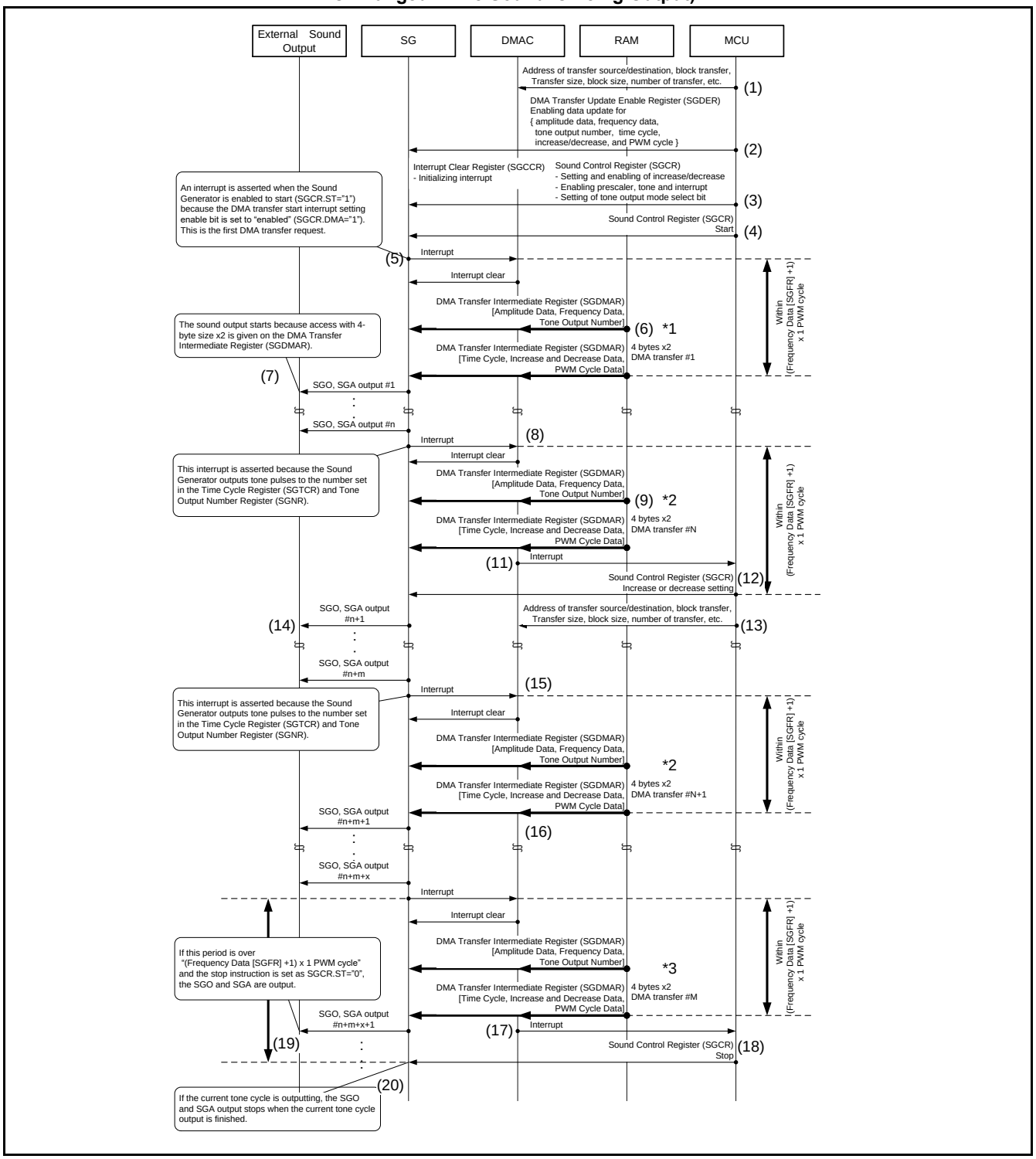
(20) If above (19) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Mth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*4: The data of the Mth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.5.5.5 In Case of DMA Transfer with 4-byte Size x 2 is Made N Times and DMA Transfer with 4-byte Size x 2 is Made M Times (The Case the Transfer Byte Size and the Increase/Decrease Setting is Changed While Sound Outputting)

Figure 3-15 Sound Generator Operation with DMA (The Case the Transfer Byte Size and Increase/Decrease Setting is Changed While Sound is Being Output)



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 2", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(3) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.
- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)
- Interrupt enable bit (SGCR.INTE) to "1"(enabled)
- Tone output mode select bit (SGCR.TMS) to "0"(Normal mode)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 2 steps.
[1st step]
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)

[2nd step]
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)

(*1: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00

- Decrement counter is 0x00
- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) When DMAC completes all DMA transfer (with 4-byte size x2, N times), it asserts an interrupt to MCU.

(12) Change the increase/decrease setting of the Sound Control Register (SGCR) by software (SGCR.GID).

(13) Software makes settings to the DMAC. The DMA transfer is based on a block data of "4-byte size x 2 cycle", and this block can be repeated M times.

(14) The Sound Generator keeps outputting SGO and SGA.

(15) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.
- Tone pulse counter is 0x00
- Decrement counter is 0x00
- At the rising edge of SGO

Then, DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".
(*2: DMA block size must to be "4-byte size x 2" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(16) Repeat the step 15 to continue outputting the sound.

(17) When DMAC completes all DMA transfer (with 4-byte size x2, M times), it asserts an interrupt to MCU.

(18) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(19) When above operation (18) was made within the following time, the Mth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.
The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
(*3: The data of the Mth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Mth DMA transfer only to assert an interrupt toward the MCU.)

(20) If above (19) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Mth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.
(*3: The data of the Mth DMA transfer are written to the Sound Generator, and they are output to the end.)

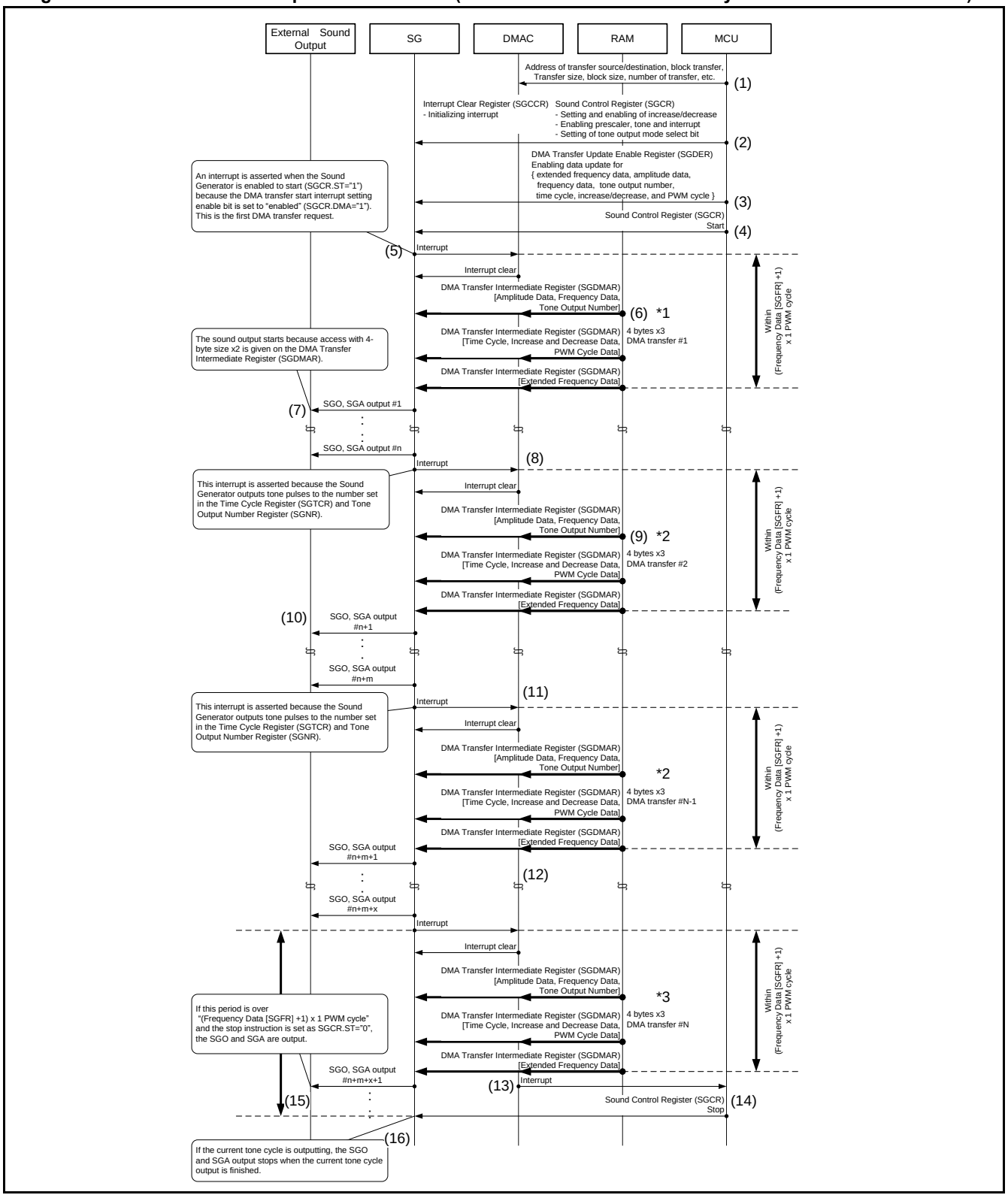
Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

- *The DMAC must finish the sequence from step 8 to 12 (including the time for changing "the increase/decrease setting"), within the following time. The increase/decrease setting change comes into effect on the sound output from step 14, using the data of the Mth transfer with 4-byte size x 2.*
  The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle

3.5.5.6 In Case of DMA Transfer with 4-byte Size x 3 is Made N Times

Figure 3-16 Sound Generator Operation with DMA (when in DMA Transfer with 4-byte Size x 3 is Made N Times)



(1) Software makes initial settings to DMAC which are needed in DMA transfer. The DMA transfer is based on a block data of "4-byte size x 3", and this block can be repeated N times. DMAC sets following registers to prepare for a DMA transfer, through the "DMA Transfer Intermediate Register (SGDMAR)".

- Amplitude Data Register (SGAR)

- Frequency Data Register (SGFR)

- Tone Output Number Register (SGNR)

- Time Cycle Register (SGTCR)

- Increase and Decrease Data Register (SGIDR)

- PWM Cycle Data Register (SGPCR)

- Extended Frequency Data Register (SGEFR)

The destination address of the DMA transfer is a fixed one on the "DMA Transfer Intermediate Register (SGDMAR)".

(2) Software initialize Interrupt status bit (SGCR.INT) by writing "1" to Interrupt status clear bit (SGCCR.INTC). Then, the software configures the Sound Control Register (SGCR) in the needed mode, and this must include the following bit operations.

- DMA transfer start interrupt setting enable bit (SGCR.DMA) to "1"(enabled)

- Interrupt enable bit (SGCR.INTE) to "1"(enabled)

- Tone output mode select bit (SGCR.TMS) to "1"(Extended mode)

(3) Software configures the "DMA Transfer Update Enable Register (SGDER)" to enable the automatic update of the following registers during DMA transfer.

- Amplitude Data Register (SGAR)

- Frequency Data Register (SGFR)

- Tone Output Number Register (SGNR)

- Time Cycle Register (SGTCR)

- Increase and Decrease Data Register (SGIDR)

- PWM Cycle Data Register (SGPCR)

- Extended Frequency Data Register (SGEFR)

(4) Write "1" to the Start bit (SGCR.ST).

(5) The interrupt occurs immediately after setting Start bit (SGCR.ST), since the Sound Generator is enabled on DMA transfer (SGCR.DMA="1"). An Interrupt request (PIRQ) is asserted, and this interrupt is used as a DMA transfer request.

(6) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)". This operation configures the following registers by 3 steps.

[1st step]

- Amplitude Data Register (SGAR)

- Frequency Data Register (SGFR)

- Tone Output Number Register (SGNR)

[2nd step]

- Time Cycle Register (SGTCR)

- Increase and Decrease Data Register (SGIDR)

- PWM Cycle Data Register (SGPCR)

[3rd step]

- Extended Frequency Data Register (SGEFR)

(*1: DMA block size must to be "4-byte size x 3" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(7) The outputs of SGO and SGA start, according to the register settings above.

(8) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.

- Tone pulse counter is 0x00

- Decrement counter is 0x00

- At the rising edge of SGO

(9) DMAC clears the interrupt, and write registers in the Sound Generator through the "DMA Transfer Intermediate Register (SGDMAR)".

(*2: DMA block size must to be "4-byte size x 3" for the access to "DMA Transfer Intermediate Register (SGDMAR)")

(10) The Sound Generator keeps outputting SGO and SGA, according to the register settings above.

(11) The Tone pulse counter counts the number of tone pulses. When the following conditions are satisfied, the interrupt is generated.

- Tone pulse counter is 0x00

- Decrement counter is 0x00

- At the rising edge of SGO

DMAC doesn't assert an interrupt to MCU until the DMAC completes all DMA transfer (with 4-byte size x3, N times).

(12) Repeat the flow from 9 to 11 to continue outputting the sound.

(13) When DMAC completes all DMA transfer (with 4-byte size x3, N times), it asserts an interrupt to MCU.

(14) Software writes "0" to the Start bit (SGCR.ST) to stop outputting the sound.

(15) When above operation (14) was made within the following time, the Nth DMA transfer doesn't come to the output of SGO and SGA. The Sound Generator stops driving SGO and SGA just before outputting the data of Nth DMA transfer.

The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle

(*3: The data of the Nth DMA transfer are written to the Sound Generator, however, they are not output. DMAC issues this Nth DMA transfer only to assert an interrupt toward the MCU.)

(16) If above (15) is not done within the limit time, the Sound Generator keeps driving SGO and SGA in order to output the data of the Nth DMA transfer. Then, the Sound Generator stops driving SGO and SGA after the end of all data.

(*3: The data of the Nth DMA transfer are written to the Sound Generator, and they are output to the end.)

Notes:

- *The DMAC must finish the sequence from step 5 to 6, within the following time. The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle*
- *The DMA transfer error means the occurrence of delay in the sound data setting. It causes unsteady sound output. In that case, please fix the priority of DMA transfer in the system to finish all data transfer within the limit time.*

3.6. Interrupt of Sound Generator

The interrupt from Sound Generator comes from the judgment of counting tone pulses.

Interrupt of Sound Generator

The interrupt control bit and its factor is shown in the following table.

Table 3-5 The Interrupt Bit of Sound Generator and its Factor

	Interrupt Bit of Sound Generator (singular)
Interrupt status bit	Register: Sound Control Register (SGCR) Bit: Interrupt status bit (INT: bit1)
Interrupt enable bit	Register: Sound Control Register (SGCR) Bit: Interrupt enable bit (INTE: bit2)
Interrupt factor	Tone pulse count is greater than or equal to ((Time Cycle Register [SGTCR] + 1) x (Tone Output Number Register [SGNR] + 1))

During the tone pulses are counted, when the count reaches the multiple of "Time Cycle Register [SGTCR] + 1" and "Tone Output Number Register [SGNR] + 1", an interrupt flag (SGCR.INT) is set to"1". When the interrupt is enabled by the setting of the Interrupt enable bit (SGCR.INTE), an Interrupt request (PIRQ) is subsequently asserted (to be "H").

4. Registers

This chapter shows the list of registers in Sound Generator.

Registers of Sound Generator

Table 4-1 The List of Registers of Sound Generator

Contracted Name	Register Name	Reference
SGDER	DMA Transfer Update Enable Register	4.1
SGCR	Sound Control Register	4.2
SGAR	Amplitude Data Register	4.3
SGFR	Frequency Data Register	4.4
SGNR	Tone Output Number Register	4.5
SGTCR	Time Cycle Register	4.6
SGIDR	Increase and Decrease Data Register	4.7
SGPCR	PWM Cycle Data Register	4.8
SGEFR	Extended Frequency Data Register	4.9
SGDMAR	DMA Transfer Intermediate Register	4.10
SGCCR	Interrupt Clear Register	4.11

The Meaning of the Register Bit Property Code

Table 4-2 The Meaning of the Register Bit Property Code

R	Read-Only Register. Only Read access is available, and Write access is ignored.
W	Write-Only Register. Only Write access is available, and Read access makes nothing.
R/W	Read/Write Register. Both Read access and Write access are available.
-	Both Read access and Write access make nothing.

Table 4-3 Register Offset Address Map

OFFSET_ADDRESS	REGISTER_NAME				ACCESS_SIZE
	+3	+2	+1	+0	
0x00000000	Reserved	SGDER	SGCR		B, H, W
0x00000004	SGAR		SGFR	SGNR	B, H, W
0x00000008	SGTCR	SGIDR	SGPCR		B, H, W
0x0000000C	Reserved *1			SGEFR	B, H, W
0x00000010	SGDMAR				B, H, W
0x00000014	Reserved		SGCCR		H, W

*1: 0x000000 is read out from the reserved bit field of 0x0000000C.

4.1. DMA Transfer Update Enable Register (SGDER)

The DMA Transfer Update Enable Register (SGDER) is to enable the update of the registers (SGAR, SGFR, SGNR, SGTCR, SGIDR, SGPCR) which can be updated when in DMA transfer.

This setting is made for respective registers.

The Sound Generator identifies the register to be updated by DMA transfer according to the setting of this register.

It also recognizes the number of times of DMA transfer and the transfer byte size according to the setting of this register, as well as the valid byte position of the DMA Transfer Intermediate Register (SGDMAR).

bit	15	14	13	12	11	10	9	8
Field	Reserved							EFRE
Attribute	-							R/W
Initial value	0x00							0

bit	7	6	5	4	3	2	1	0
Field	ARE1	ARE0	FRE	NRE	TCRE	IDRE	PCRE1	PCRE0
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

[bit15:9] Reserved : Reserved bits

Always write "0" to this bit. The read value is "0".

[bit8] EFRE : Extended frequency data update enable bit

This bit is to enable the update of the extended frequency data of the Extended Frequency Data Register (SGEFR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the extended frequency data is disabled.
1	The update of the extended frequency data is enabled.

Notes:

- *When the Tone output mode select bit (SGCR.TMS) is in Extended mode (SGCR.TMS="1"), it is allowed to write the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR).*
- *The Tone output mode select bit (SGCR.TMS) and the Extended frequency data update enable bit (SGDER.EFRE) are allowed to be written at one time.*
- *When the Tone output mode select bit (SGCR.TMS) was changed from Extended mode (SGCR.TMS="1") to Normal mode (SGCR.TMS="0"), the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR) are cleared to the initial value.*

[bit7] ARE1 : Amplitude data (upper byte) update enable bit

This bit is to enable the update of the amplitude data (upper byte) of the Amplitude Data Register (SGAR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the amplitude data (upper byte) is disabled.
1	The update of the amplitude data (upper byte) is enabled.

[bit6] ARE0 : Amplitude data (lower byte) update enable bit

This bit is to enable the update of the amplitude data (lower byte) of the Amplitude Data Register (SGAR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the amplitude data (lower byte) is disabled.
1	The update of the amplitude data (lower byte) is enabled.

[bit5] FRE : Frequency data update enable bit

This bit is to enable the update of the frequency data of the Frequency Data Register (SGFR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the frequency data is disabled.
1	The update of the frequency data is enabled.

[bit4] NRE : Tone output number update enable bit

This bit is to enable the update of the tone output number of the Tone Output Number Register (SGNR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the tone output number is disabled.
1	The update of the tone output number is enabled.

[bit3] TCRE : Time cycle update enable bit

This bit is to enable the update of the time cycle of the Time Cycle Register (SGTCR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the time cycle is disabled.
1	The update of the time cycle is enabled.

[bit2] IDRE : Increase and decrease data update enable bit

This bit is to enable the update of the increase and decrease data of the Increase and Decrease Data Register (SGIDR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the increase and decrease data is disabled.
1	The update of the increase and decrease data is enabled.

[bit1] PCRE1 : PWM cycle data (upper byte) update enable bit

This bit is to enable the update of the PWM cycle data (upper byte) of the PWM Cycle Data Register (SGPCR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the PWM cycle data (upper byte) is disabled.
1	The update of the PWM cycle data (upper byte) is enabled.

[bit0] PCRE0 : PWM cycle data (lower byte) update enable bit

This bit is to enable the update of the PWM cycle data (lower byte) of the PWM Cycle Data Register (SGPCR) through the DMA Transfer Intermediate Register (SGDMAR) when in DMA transfer.

bit	Description
0	The update of the PWM cycle data (lower byte) is disabled.
1	The update of the PWM cycle data (lower byte) is enabled.

4.2. Sound Control Register (SGCR)

The Sound Control Register (SGCR) is to control interrupts and operation status of the Sound Generator.

bit	15	14	13	12	11	10	9	8
Field	Reserved	SRST	DMA	GID	GEN	Reserved	BUSY	TMS
Attribute	-	W	R/W	R/W	R/W	-	R	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	S1	S0	TONE	SGOOE	SGAOE	INTE	INT	ST
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

[bit15] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

[bit14] SRST : Software reset bit

This bit is used to issue a software reset in the Sound Generator.

When in the read access, the read value is always "0". Writing "0" to this bit has no effect.

bit	Description
	Write
0	No effect.
1	Writing "1" issues a software reset.

[bit13] DMA : DMA transfer start interrupt setting enable bit

This bit is to enable DMA transfer start interrupt.

bit	Description
0	Writing "1" to the Start bit (SGCR.ST) does not make the Interrupt status bit (SGCR.INT) set.
1	Writing "1" to the Start bit (SGCR.ST) makes the Interrupt status bit (SGCR.INT) set.

Note:

- *Do not change this setting while operating (SGCR.ST="1" or SGCR.BUSY="1").*

[bit12] GID : Increase/decrease setting bit

This bit is for automatic increase/decrease setting of the sound amplitude, working with the Time Cycle Register (SGTCR), the Increase and Decrease Data Register (SGIDR), and the automatic increase/decrease enable bit (GEN).

This bit determines whether the value of Amplitude Data Register (SGAR) increases or decreases.

bit	Description
0	The value stored in the Amplitude Data Register (SGAR) decreases.
1	The value stored in the Amplitude Data Register (SGAR) increases.

Notes:

- *This bit is valid only when the automatic increase/decrease enable bit is set to "enabled"(SGCR.GEN="1").*
- *If this bit is changed during the operation, it is reflected at the time when the value of "Amplitude Data Register (SGAR)" is automatically updated.*

[bit11] GEN : Automatic increase/decrease enable bit

This bit is for automatic increase/decrease setting of the sound, working with the Time Cycle Register (SGTCR), the Increase and Decrease Data Register (SGIDR), and the Increase/decrease setting bit (GID).

bit	Description
0	The automatic increase or decrease is disabled.
1	The automatic increase or decrease is enabled.

Note:

- *When this bit turns from "enabled" to "disabled", the Amplitude Data Register (SGAR) holds the value at that time.*

[bit10] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

[bit9] BUSY : Busy status bit

This bit is to indicate whether the Sound Generator is in operation or not.

This bit is only for read access, and the write access to this bit has no effect.

bit	Description
	Read
0	Sound Generator is inactive. This bit is cleared under any of following conditions. - Software reset is set (SGCR.SRST="1"). - Start bit(SGCR.ST) is cleared by writing "0". And the operation is completed after 1 tone cycle (current tone cycle) has finished.
1	Sound Generator is active. This bit is set when Start bit(SGCR.ST) is written "1".

[bit8] TMS: Tone output mode select bit

This bit selects output mode of tone pulse signal of SGO.

bit	Description
0	Normal mode (initial value) - Duty ratio of a tone pulse signal is fixed to 50%.
1	Extended mode - Duty ratio (high side) of a tone pulse signal is programmable up to 50%.

Notes:

- *Do not change this setting while operating (SGCR.ST="1" or SGCR.BUSY="1")*
- *When the Tone output mode select bit (SGCR.TMS) is in Extended mode (SGCR.TMS="1"), it is allowed to write the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR).*
- *The Tone output mode select bit (SGCR.TMS) and the Extended frequency data update enable bit (SGDER.EFRE) are allowed to be written at one time.*
- *When the Tone output mode select bit (SGCR.TMS) was changed from Extended mode (SGCR.TMS="1") to Normal mode (SGCR.TMS="0"), the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR) are cleared to the initial value.*

[bit7:6] S1, S0 : Operation clock select bits

The combination of these bits controls the internal clock division, to determine the bit rate on the output sound.

S1	S0	Description
0	0	The division ratio at the Prescaler is 1.
0	1	The division ratio at the Prescaler is 2.
1	0	The division ratio at the Prescaler is 4.
1	1	The division ratio at the Prescaler is 8..

[bit5] TONE : Tone output bit

This bit controls the type of SGO output signal.

bit	Description
0	SGO outputs the mixed signal of tone pulse and PWM pulse.
1	SGO outputs a simple square wave (tone pulse)

[bit4] SGOOE : SGO signal output enable bit

This bit controls whether to enable the output of SGO.

bit	Description
0	SGO output is disabled.
1	SGO output is enabled.

[bit3] SGAOE : SGA signal output enable bit

This bit controls whether to enable the output of SGA.

bit	Description
0	SGA output is disabled.
1	SGA output is enabled.

[bit2] INTE : Interrupt enable bit

This bit enables the interrupt request of the Sound Generator.

bit	Description
0	The interrupt by Interrupt status bit (SGCR.INT) is masked, and the status of Interrupt status bit (SGCR.INT) is not reflected to the output on PIRQ.
1	The interrupt by Interrupt status bit (SGCR.INT) is not masked, and the status of Interrupt status bit (SGCR.INT) is reflected to the output on PIRQ.

[bit1] INT : Interrupt status bit

This is an interrupt flag of the Sound Generator.

This bit is only for read access, and the write access to this bit has no effect.

bit	Description	
	Read	Write
0	The tone pulse counter hasn't detected the number of tone pulses set as Tone Output Number Register (SGNR) and Time Cycle Register (SGTCR). This bit is cleared to "0" under any of following conditions. - Software reset is set (SGCR.SRST="1"). - DMA transfer - Interrupt Clear bit is set (SGCCR.INTC="1")	No effect.
1	The tone pulse counter has detected the number of tone pulses set as Tone Output Number Register (SGNR) and Time Cycle Register (SGTCR). This bit is set to "1" under the following condition. - The number of tone pulse count is greater than or equal to ((Time Cycle Register [SGTCR] + 1) x (Tone Output Number Register [SGNR] + 1))	No effect.

[bit0] ST : Start bit

This bit is a start trigger of the Sound Generator.

bit	Description
0	Writing "0" to this bit stops Sound Generator.
1	Writing "1" to this bit starts Sound Generator.

Notes:

- *Whether the Sound Generator is stopped or not can be seen on the status bit BUSY.*
- *This bit is cleared also when a software reset is issued (SGCR.SRST = "1").*

4.3. Amplitude Data Register (SGAR)

The Amplitude Data Register (SGAR) stores the reload value for a PWM pulse generator.

The register value represents sound amplitude and is reloaded into the PWM pulse generator at the end of every tone cycle.

bit	15	14	13	12	11	10	9	8
Field	SGAR[15:8]							
Attribute	R/W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGAR[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit15:0] SGAR : Amplitude data bits

These bits store the reload value for the PWM pulse generator,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

When in decreasing the value of this register, 0x0000 is the lower limit and the value doesn't roll over to 0xFFFF from it.

When in increasing the value of this register, 0xFFFF is the upper limit and the value doesn't roll over to 0x0000 from it.

In any case, the Sound Generator keeps operating until the Start bit (SGCR.ST) is cleared.

Notes:

- *The number of clock cycles in high width of 1 PWM is equal to "SGAR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.4. Frequency Data Register (SGFR)

The Frequency Data Register (SGFR) stores the reload value for a Frequency counter.

This register setting is half cycle of a tone pulse signal.

bit	7	6	5	4	3	2	1	0
Field	SGFR[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit7:0] SGFR : Frequency data bits

These bits store the reload value for the Frequency counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of PWM pulses is equal to "SGFR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.5. Tone Output Number Register (SGNR)

The Tone Output Number Register (SGNR) stores the reload value for a Tone pulse counter.

The Tone pulse counter accumulates the number of tone pulses (or the number of sound increment/decrement). When the accumulated value reaches the reload value, it sets the Interrupt status bit (SGCR.INT). This operation is to reduce the frequency of interrupts.

bit	7	6	5	4	3	2	1	0
Field	SGNR[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit7:0] SGNR : Tone output number bits

These bits store the reload value for the Tone pulse counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of pulses is equal to "SGNR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.6. Time Cycle Register (SGTCR)

The Time Cycle Register (SGTCR) stores the reload value for a Decrement counter.

bit	7	6	5	4	3	2	1	0
Field	SGTCR[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit7:0] SGTCR : Time cycle bits

These bits store the reload value for the Decrement counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of pulses is equal to "SGTCR + 1".*
- *When SGTCR is 0x00, the automatic increase or decrease operation on the Amplitude Data Register (SGAR) occurs at each tone cycle.*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.7. Increase and Decrease Data Register (SGIDR)

The Increase and Decrease Data Register (SGIDR) stores the increment or decrement for the Amplitude Data Register (SGIDR).

bit	7	6	5	4	3	2	1	0
Field	SGIDR[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit7:0] SGIDR : Increase and decrease data bits

These bits store the increment or decrement for the Amplitude Data Register (SGAR) ,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Note:

- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.8. PWM Cycle Data Register (SGPCR)

The PWM Cycle Data Register (SGPCR) stores the number of cycles in 1 PWM cycle.

bit	15	14	13	12	11	10	9	8
Field	SGPCR[15:8]							
Attribute	R/W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGPCR[7:0]							
Attribute	R/W							
Initial value	0xFF							

[bit15:0] SGPCR : PWM cycle number data bits

These bits store the number of cycles in 1PWM cycle,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of cycles in 1 PWM is equal to "SGPCR + 1".*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.9. Extended Frequency Data Register (SGEFR)

The Extended Frequency Data Register (SGEFR) stores the comparison value of Frequency counter. This register setting is high width of a tone pulse signal in Extended mode (SGCR.TMS="1").

bit	7	6	5	4	3	2	1	0
Field	SGEFR[7:0]							
Attribute	R/W							
Initial value	0xFF							

[bit7:0] SGEFR : Extended Frequency data bits

These bits store the comparison value of Frequency counter,
and the value is written either in the following ways.
(1)Software writes a value to this register.
(2)DMAC writes a value through DMA Transfer Intermediate Register (SGDMAR).

Notes:

- *The number of PWM pulses is equal to "SGEFR + 1".*
- *When the Tone output mode select bit (SGCR.TMS) is in Extended mode (SGCR.TMS="1"), it is allowed to write the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR).*
- *The Tone output mode select bit (SGCR.TMS) and the Extended frequency data update enable bit (SGDER.EFRE) are allowed to be written at one time.*
- *When the Tone output mode select bit (SGCR.TMS) was changed from Extended mode (SGCR.TMS="1") to Normal mode (SGCR.TMS="0"), the Extended frequency data update enable bit (SGDER.EFRE) and the Extended Frequency Data Register (SGEFR) are cleared to the initial value.*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse. (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.10. DMA Transfer Intermediate Register (SGDMAR)

The DMA Transfer Intermediate Register (SGDMAR) is available in DMA transfer, and works as a window to access several registers. The details are described in "3.4 Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings".

bit	31	30	29	28	27	26	25	24
Field	SGDMAR[31:24]							
Attribute	W							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	SGDMAR[23:16]							
Attribute	W							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	SGDMAR[15:8]							
Attribute	W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	SGDMAR[7:0]							
Attribute	W							
Initial value	0x00							

[bit31:0] SGDMAR : DMA Transfer data bits

This register is used to write the following registers:
- Amplitude Data Register (SGAR)
- Frequency Data Register (SGFR)
- Tone Output Number Register (SGNR)
- Time Cycle Register (SGTCR)
- Increase and Decrease Data Register (SGIDR)
- PWM Cycle Data Register (SGPCR)
- Extended Frequency Data Register (SGEFR)

The details are described in "3.4 Relation between the DMA Transfer Update Enable Register (SGDER) and DMA Settings".

The read value is always 0x00000000.

Notes:

- *If the number of DMA transfer is greater than "the number determined by the DMA Transfer Update Enable Register (SGDER)", the value of this register is updated.*

- *When in reading corresponding registers (not through this register), please make sure the setting of "DMA Transfer Update Enable Register (SGDER)" is available for the target registers.*
- *As for the enabled setting register of the DMA Transfer Update Enable Register (SGDER), access is enabled to both the DMA Transfer Intermediate Register (SGDMAR) and individual registers (SGAR, SGFR, SGNR, SGTCR, SGIDR, SGPCR, SGEFR).*
- *During operation, in the case of changing a register value, meet the following either conditions.*
  (1)Sound generator is in the stop state (SGCR.ST="0" and SGCR.BUSY="0").
  (2)The changing comes between an interrupt occurrence and falling edge of the first tone pulse.
  (The limit time = (Frequency Data Register [SGFR] + 1) x 1 PWM cycle)
  When it meets neither condition, the sound output cannot guarantee the expected duty ratio.

4.11. Interrupt Clear Register (SGCCR)

The Interrupt Clear Register (SGCCR) clears the Interrupt status bit (SGCR.INT) in the Sound Control Register(SGCR).

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved						INTC	Reserved
Attribute	-						W	-
Initial value	0x00						0	0

Note:

- *Use 16- or 32-bit data access for the SGCCR register.*

[bit15:2] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit1] INTC : Interrupt status clear bit

When this bit is set to "1", Interrupt status bit (SGCR.INT) is cleared.

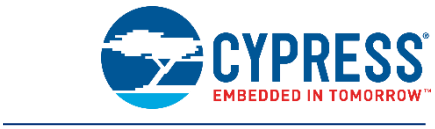
The read value is "0".

bit	Description
	Write
0	No effect.
1	Interrupt status bit (SGCR.INT) is cleared.

[bit0] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

CHAPTER 30: Sound Waveform Generator



This chapter explains the sound waveform generator.

1. Overview
2. Configuration and Block Diagram
3. Operation of the Sound Waveform Generator
4. Registers
5. Appendix

CODE: SOUNDWFG-E3.25-0

1. Overview

This section gives a brief overview of the sound waveform generator (SWFG).

Also refer to the chapter of "Sound System configuration" how to configure the SWFG..

The SWFG is a function that generates and outputs a Pulse Code Modulation (PCM) sound source using simple software settings.

SWFG generates signed 16bit PCM. Maximum value of the PCM is 0x7FFF in hexadecimal and minimum value is 0x8001 in hexadecimal. In decimal, both values are 32767 and -32767 respectively. In figure on this document, both values are normalized and these are 1 and -1 respectively.

Table 1-1

Feature	Description
Output frequency	20 Hz to 20 kHz (0.5 Hz step) settable by software.
Output waveform	Sine wave, triangle wave, and square wave selectable by software. Duty is variable for triangle wave and square wave.
Volume control (attack and release functions)	Seamless fade-in (attack control) and fade-out (release control) by software are supported immediately after generation starts and immediately before generation ends. Support exponential attack/release and linear attack/release.
Sound time	A time from 0.5 ms to 8191.5 ms (0.5 ms step) selectable by software.
Number of output channels	Simultaneous output of up to five different sound sources is supported.
Interrupt	Completion of sound source output generates an interrupt. SWFG AHB Master interface bus error interrupt.
Consecutive setting	While a sound source is being generated, register settings for generating the next sound source can be configured. Support continuous generation and number of continuous times is selectable (1 to 15, or infinite).
Output data pattern	Generates a 16-bit PCM (Pulse Code Modulation) sound source XXXX internally by the SWFG, expands it to 32 bits, and outputs it (AHB master interface). The output data pattern can be selected by register settings. − XXXX_XXXX: Upper and lower 16 bits are the same data. − 0000_XXXX: Only lower 16 bits of data (Upper 16 bits are zero.) − XXXX_0000: Only upper 16 bits of data (Lower 16 bits are zero.)
Output sampling rate	48 kHz (fixed).
FIR filter	A Low Pass Filter is provided to eliminate triangle and square waveform overtones. Support variable filter coefficients which are defined by software.
Initial gain	Initial gain is settable by software. Its unit is linear or dB and it is selectable by software.
Gain fluctuating value	Gain fluctuating value is settable by software. Its unit is linear or dB and it is selectable by software.

Notes:

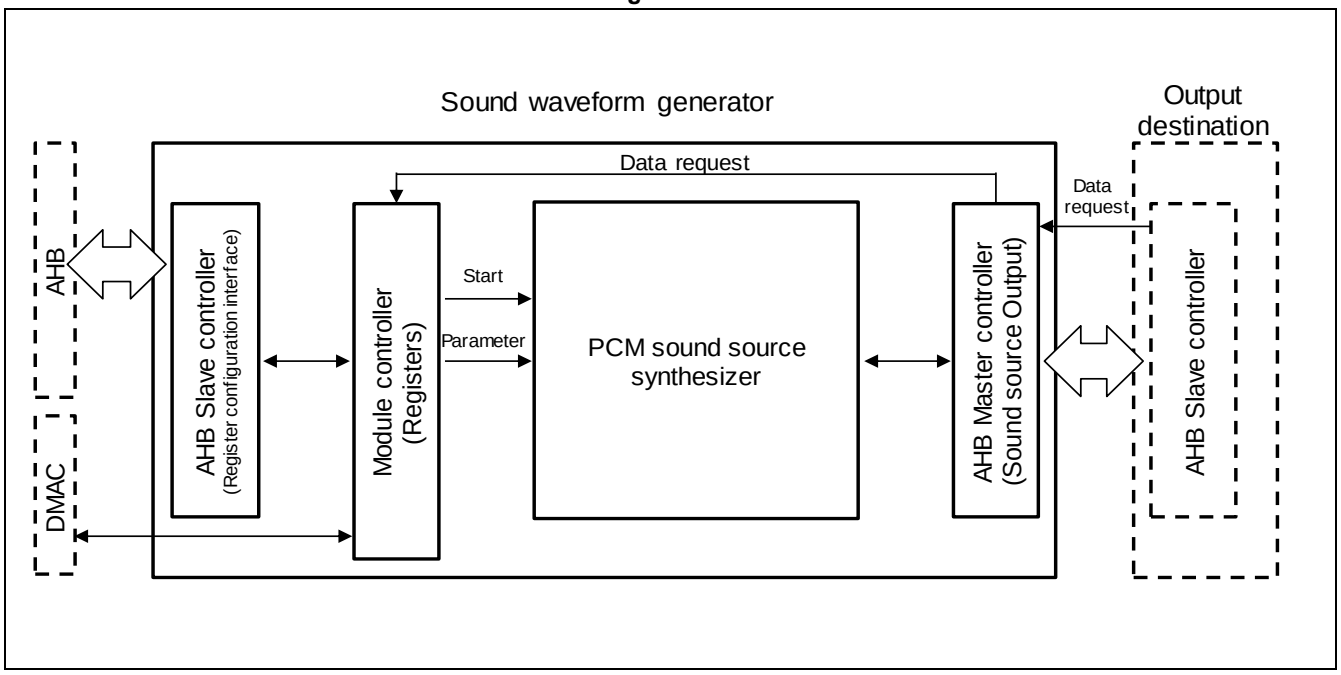
− *Theoretically, system noise tends not to be a factor in the case of a sine wave that contains single frequency components. Use of a sine wave is particularly recommended when sound source mixing or other processes are performed by other functions.*

- *Interrupts due to factors other than sound source output completion are also supported. For details, see 3.3.*
- *Support 2 types of continuous sound generation method. For details, see 3.6.*
- *Use of the Low Pass Filter can be disabled. When using the Low Pass Filter, the optimum cutoff frequency should be decided upon sufficient evaluation of sound quality on the system.*

2. Configuration and Block Diagram

This section shows a block diagram of the sound waveform generator.

Figure 2-1



3. Operation of the Sound Waveform Generator

3.1. Sound Source Specifications

This section describes the specifications of a sound source generated by SWFG.

Figure 3-1

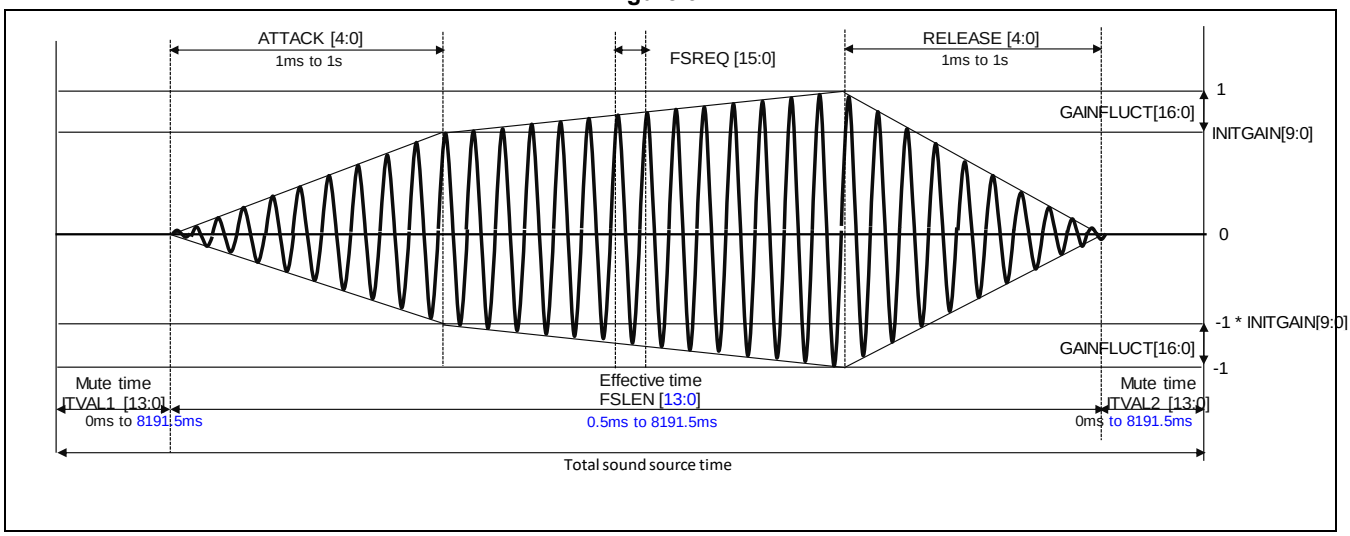


Table 3-1

Element	Description	Relation
Anterior mute time (ITVAL1)	Silence generation time (ALL 0 data generation) immediately following sound source generation start.	4.24 WGCHnCTRL3 ITVAL1[13:0]
Posterior mute time (ITVAL2)	Silence period just before sound source generation completion. All-0 data is generated in this period.	4.24 WGCHnCTRL3 ITVAL2[13:0]
Waveform (FSINF)	Sound source waveform (sine, triangle, squire wave)	4.23 WGCHnCTRL2 FSINF[1:0]
Frequency (FSFREQ)	Sound source frequency	4.25 WGCHnCTRL4 FSFREQ[15:0]
Run time (FSLEN)	Time to generate a sound source of a specified frequency	4.25 WGCHnCTRL4 FSLEN[13:0]
Attack time (ATTACK)	Fade in time immediately following run time start (included in run time)	4.23 WGCHnCTRL2 ATTACK[4:0]
Release time (RELEASE)	Fade out time immediately before run time end (included in run time)	4.23 WGCHnCTRL2 RELEASE[4:0]
Initial gain (INITGAIN)	Initial amplitude of waveform (Ratio for maximum amplitude)	4.22 WGCHnCTRL1 INITGAIN[9:0]
Gain fluctuating value (GAINFLUCT)	Gain increasing value or decreasing value	4.22 WGCHnCTRL1 GAINFLUCT[16:0]
Total sound source time	Total time of the specified sound source, which is the total of mute time and effective time	Figure 3-1

3.1.1. Mute Time (ITVAL1/2)

There are anterior mute time and posterior mute time.

Anterior mute time is silence generation time immediately following sound source generation start. Posterior mute time is silence generation time just before sound source generation completion.

As for both mute time, 0 to 8191.5 ms (0.5 ms step) is selectable by software.

Notes:

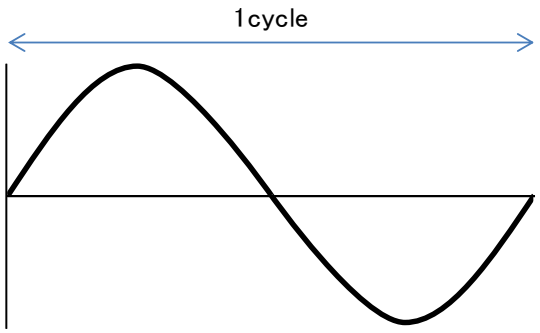
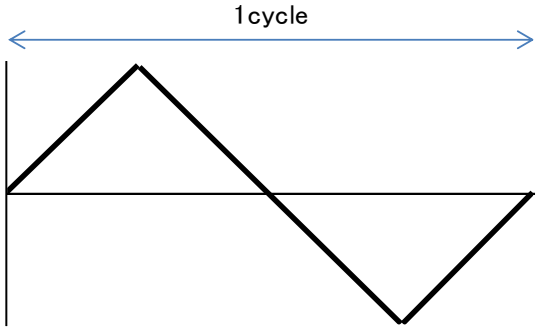
- *Control of WGCHEN under 4.1 causes ALL 0 to be output at the point that SWFG operation is enabled, but this is not included in mute time. It should be noted that anterior mute time indicates the period from the start of sound source generation in accordance with control by WGCHSTART under 4.2, up to generation of ALL 0.*
- *See also to 3.1.*

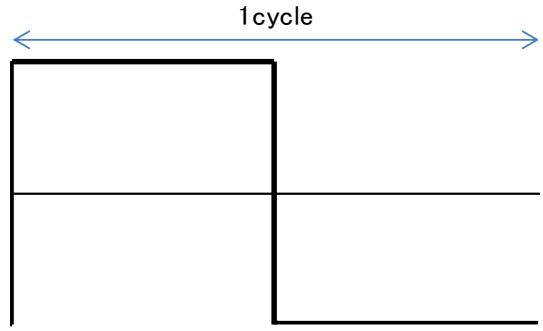
3.1.2. Waveform (FSINF)

The waveform is the fundamental wave that makes up the sound source.

One of 3 waveforms (sine wave, triangle wave, square wave) is selectable by software.

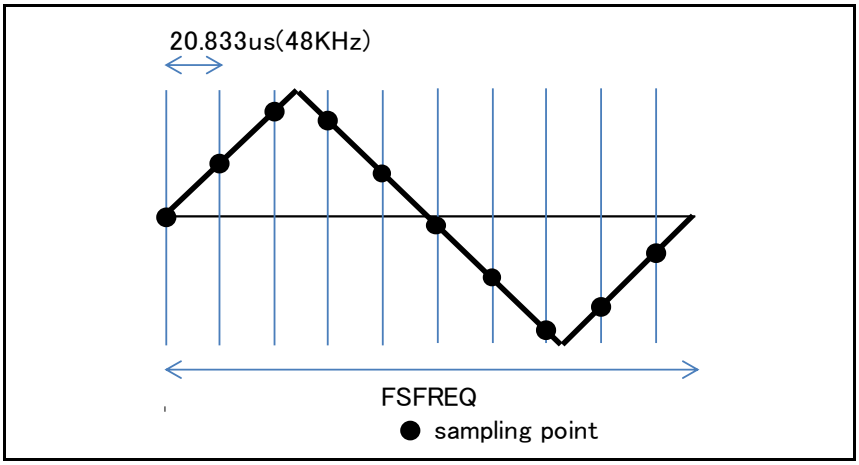
Table 3-2

Feature	Description
Sine wave	
Triangle wave	

Feature	Description
Squire wave	

Sampling rate is 48 kHz for all waveforms. PCM data is sampled from ideal waveform. In case of triangle wave, sampling point is not on peak of triangle shape at some frequency. (See Figure 3-2)

Figure 3-2



In case of triangle or squire wave, duty is settable with 1% resolution. (See 4.23)

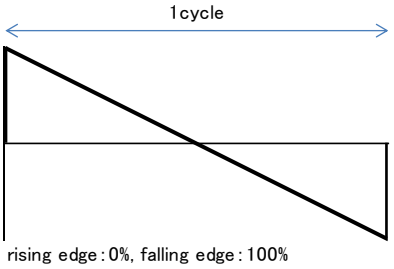
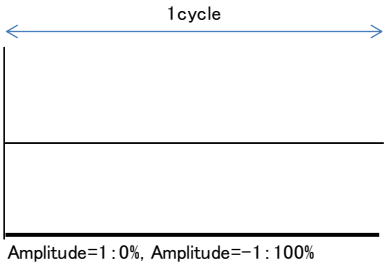
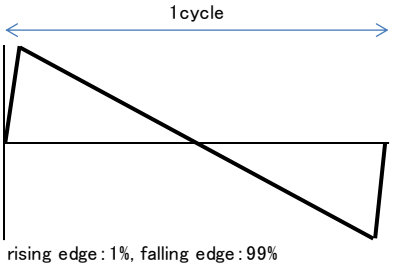
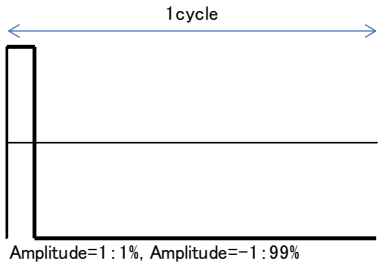
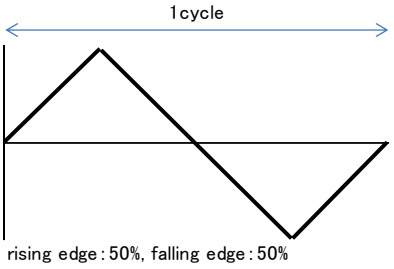
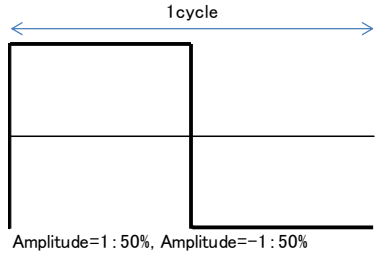
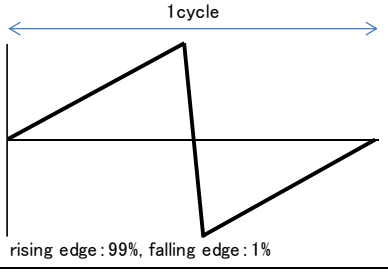
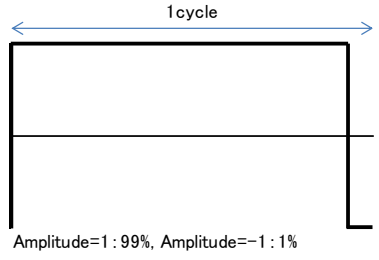
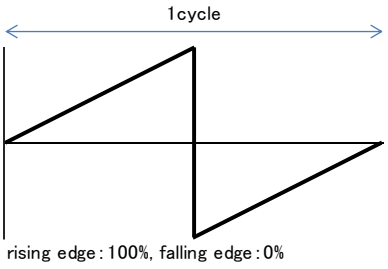
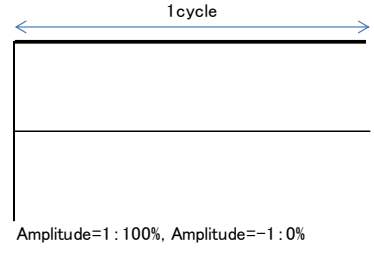
In case of triangle wave, waveform goes to sawtooth when duty setting is equal to 0% or 100%. (See Table 3-3)

In case of squire wave, waveform goes to DC level silence signal when duty setting is equal to 0% or 100%. (See Table 3-3)

At 0% : DC level is equal to minimum value (0x8001 in hexadecimal) times gain ratio.

At 100% : DC level is equal to maximum value (0x7FFF in hexadecimal) times gain ratio.

Table 3-3

DUTY[6:0]	FSINF[1:0]	
	Triangle Wave	Square Wave
0000000 (0d0)		
0000001 (0d1)		
.....	.....	.....
0110010 (0d50)		
.....	.....	.....
1100011 (0d99)		
1100100 (0d100) to 1111111 (0d127)		

3.1.3. Frequency (FSFREQ)

Frequency is a sound source component that corresponds to the pitch of the sound.

20 Hz to 20 kHz (0.5 Hz step) is selectable by software. For details, see 4.25.

3.1.4. Run Time (FSLEN)

Run time is the time to generate a sound source of the specified frequency.

0.5 ms to 8191.5 ms (0.5 ms step) is selectable by software. For details, see 4.25.

Notes:

- *Neither the anterior nor posterior mute times are included in the run time.*
- *Attack time and release time are both included in the run time.*

3.1.5. Attack Time (ATTACK)

Gain value during attack fluctuates to initial gain from start gain within setting time period. Start gain is final gain value on previous waveform and is "0" in case of just after reset.

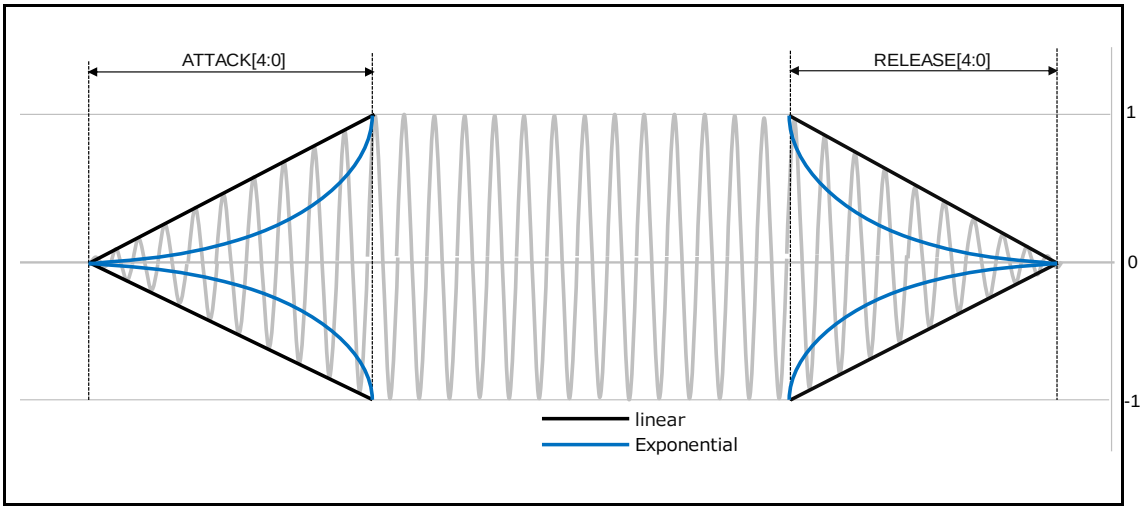
In case of "FSLEN < ATTACK", this specification is not applicable. See 5.1.7.

This operation depends on gain transfer setting. See 3.1.10.

One of 24 attack times from 1 to 1000 ms is selectable by software.

Waveform during attack control is selectable by software (See 4.23). Support linear shape and exponential shape. For detail, see Figure 3-3.

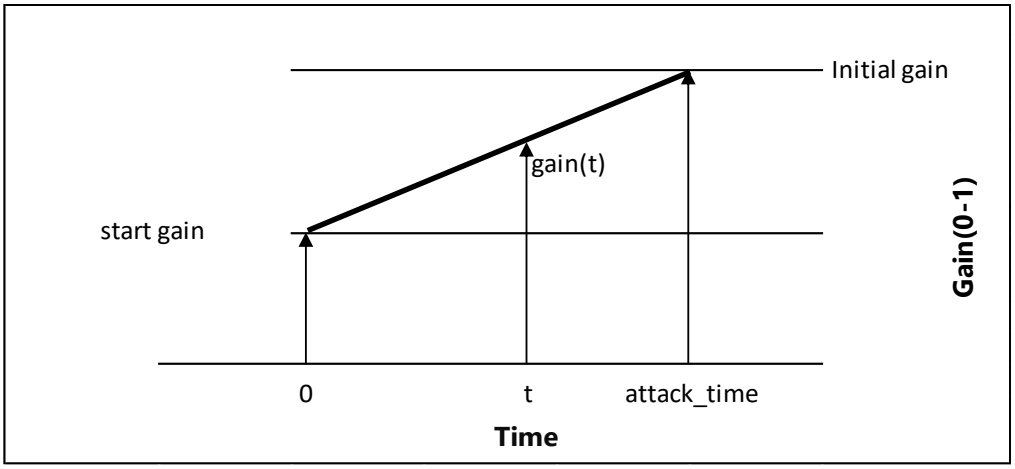
Figure 3-3



3.1.5.1 Attack with Linear Waveform Setting

Waveform is on Figure 3-4.

Figure 3-4



Gain(t) is calculated by following formula.

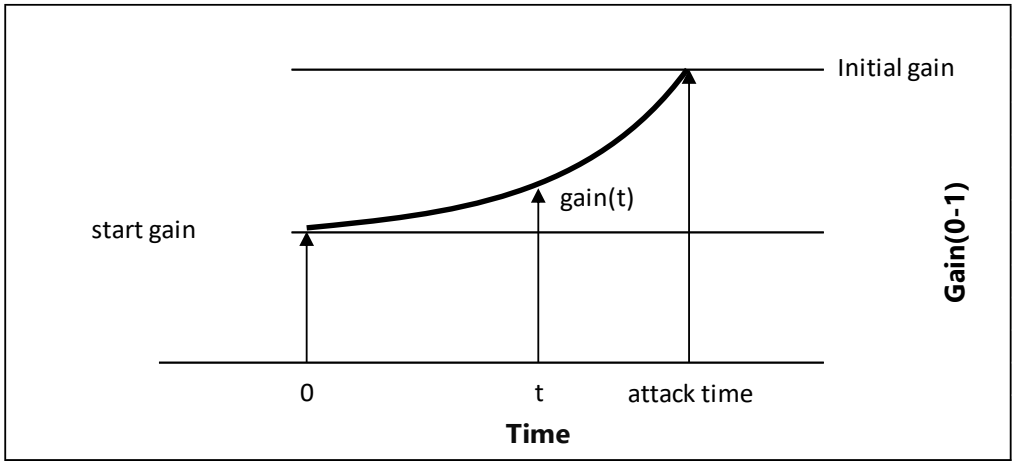
$$gain_slope=(initial_gain-start_gain)/attack_time$$
$$gain(t)=start_gain+gain_slope*t$$

gain_slope :	Gain increase/decrease value at unit period (s).
initial_gain :	Initial gain
start_gain :	Start gain of the attack operation
attack_time :	Time period of attack (s)

3.1.5.2 Attack with Exponential Waveform Setting

Waveform is Figure 3-5.

Figure 3-5



Gain(t) is calculated by following formula.

$$gain_slope_db=(initial_gain_db-start_gain_db)/attack_time$$
$$gain_db(t)=start_gain_db+gain_slope_db*t$$
$$gain(t)=10^{gain_db(t)/20}$$

gain_slope :	Gain increase/decrease value at unit period (s).
initial_gain :	Initial gain
start_gain :	Start gain of the attack operation
gain_db(t) :	gain (dB) at time t.
attack_time :	Time period of attack (s)

3.1.6. Release Time (RELEASE)

Gain value during release fluctuates to "0" from final gain within setting time period. Final gain is gain value before stating the release operation.

In case of "FSLEN < (ATTACK + RELEASE)", this specification is not applicable. See 5.1.5, 5.1.6 and 5.1.7.

One of 24 release times from 1 to 1000 ms is selectable by software.

Waveform during release control is selectable by software (Refer to 4.23). Support linear shape and exponential shape. For detail, see Figure 3-3.

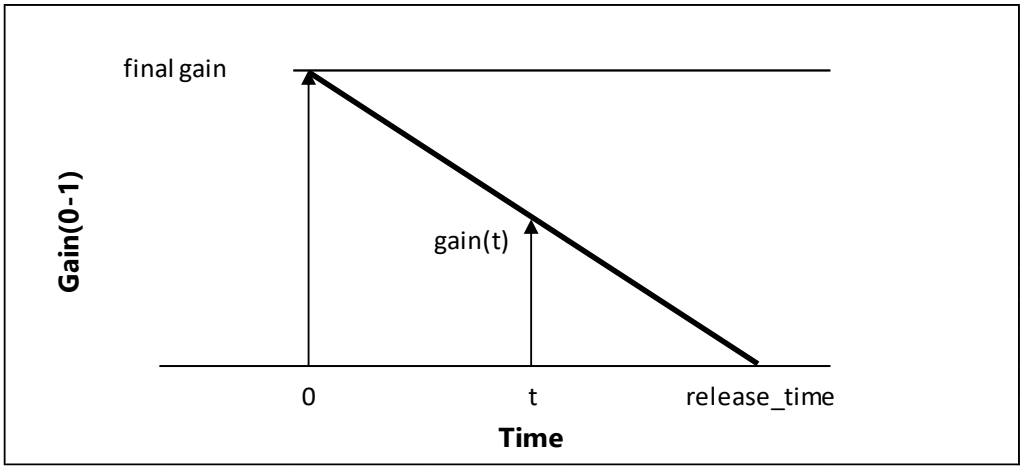
Notes:

- *Attack time and release time are both included in the run time.*

3.1.6.1 Release with Linear Waveform Setting

Waveform is Figure 3-6.

Figure 3-6



Gain(t) is calculated by following formula.

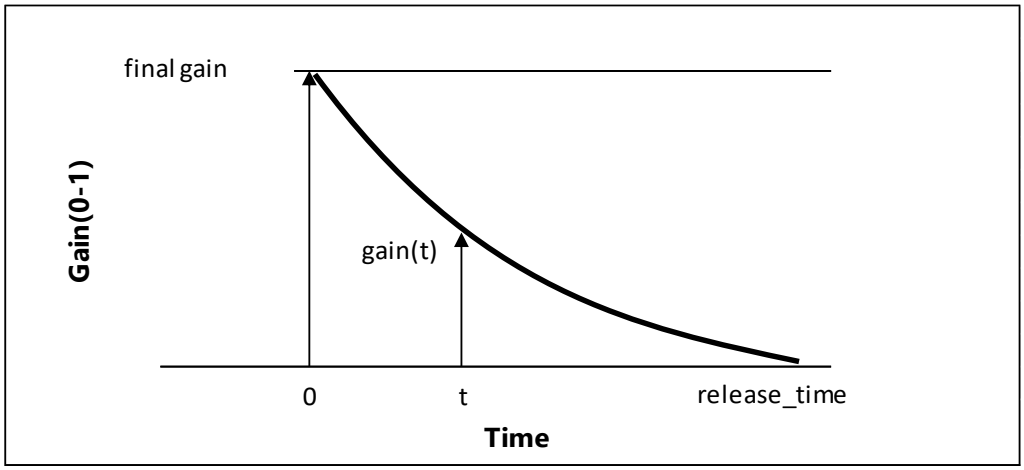
$$gain_slope = final_gain / release_time$$
$$gain(t) = final_gain - gain_slope * t$$

gain_slope :	Gain decrease value at unit period (1 s).
final_gain :	Gain value before stating the release operation
release_time :	Time period of release (s)

3.1.6.2 Release with Exponential Waveform Setting

Waveform is Figure 3-7.

Figure 3-7



Gain(t) is calculated by following formula.

$$gain_slope_db = final_gain_db / release_time$$
$$gain_db(t) = final_gain_db - gain_slope_db * t$$
$$gain(t) = 10^{gain_db(t)/20}$$

gain_slope_db :	Gain decrease value (dB) at unit period (1 s).
final_gain_db :	Gain value (dB) before stating the release operation
gain_db(t) :	Gain value (dB) at time t
release_time :	Time period of release (s)

3.1.7. Initial Gain

Initial gain is set by 4.22 WGCHnCTRL1-INITGAIN[9:0].
Setting value is multiplying factor.
Available amplitude is between 0x7FFF and 0x8001 in hexadecimal (between 32767 and -32767 in decimal).
Initial amplitude is calculated from initial gain and this amplitude range.

Unit of initial gain is ratio or dB and selected by software.
In case of ratio, settable initial gain is 1.000 to 0.000 (0.001 step).
In case of dB, settable initial gain is 0.0 dB to -96.0 dB (0.1 dB step).

Gain transfer function is defined in 3.1.8.
If gain transfer is enabled, initial gain of next sound source generation is come from previous sound.
If gain transfer is not enabled, initial gain of next sound source is equal to this setting value.

For details, see 3.1.10 and 4.22.

Notes:

Because of following reasons, volume of the PCM data is different according to the kind of the waveform or duty even though setting value of the initial gain is same.

- *Power of sound depends on waveform shape (sine/ triangle/ squire), even though these frequencies are same.*
- *Harmonic overtone happens on triangle and square waveform but does not happen on sine waveform.*
- *Square waveform with duty setting 0% and 100% becomes DC level waveform and it does not make any sound.*
- *Even if the minimum gain is applied to the sound source, the SWFG does not always output ALL 0 data. In that case, positive data become ALL 0, but negative data become ALL 1 (-1 in decimal). Due to this, slight sound may be heard in accordance with your output environment.*

3.1.8. Gain Transfer

Initial gain of sound source generation is transferred from gain of previous sound source generation or set by register as initial gain.
It is selected by software (4.22 WGCHnCTRL1-GAINSUCEN).

In case of using gain of previous sound source generation, transferred gain is the value at the previous sound source generation completion. Therefore gain is smoothly connected to next sound source generation.
In case of using setting register value, gain is not smoothly connected. However with attack function, gain can be smoothly connected from previous sound source generation to setting register value as initial gain.

If gain transfer is enabled just after reset, transferred gain for 1st sound source generation is "0".
Previous sound source generation is completed with the condition that gain value is "0", transferred gain is also "0".
In the above 2 cases, generated sound source become silence (PCM value is "0") with following condition.

- Gain increase/decrease setting is "decrease".
  (GAINUDSEL = decrease)
- Gain increase/decrease setting is "increase" and gain fluctuating value is "0".
  (GAINUDSEL = increase and GAINFLUCT[16:0] = 0)

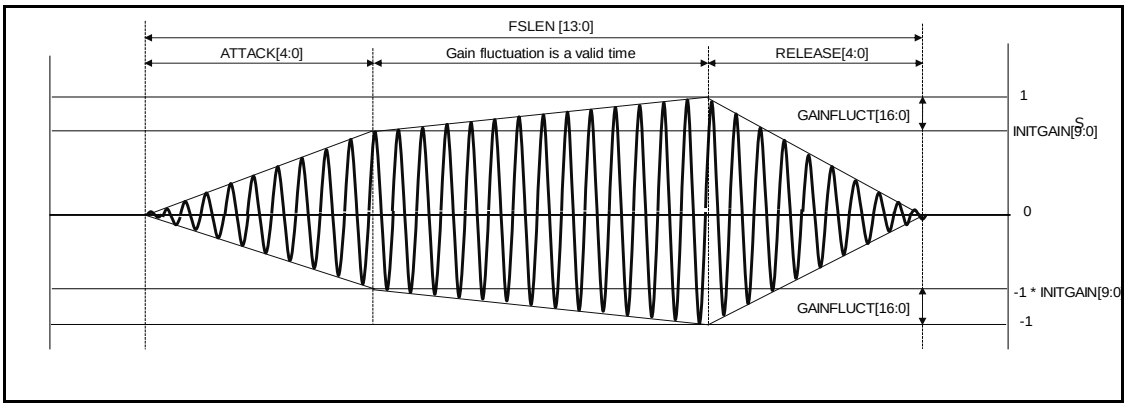
For details, see 3.1.10 and 4.22.

3.1.9. Gain Increase/Decrease

Gain increase/decrease value per 10 ms can be set by software (4.22 WGCHnCTRL1-GAINFLUCT[16:0]).
Unit of this value is ratio or dB and selected by software (4.22 WGCHnCTRL1-DBEN).
In case of ratio, settable value is 0.00000 to 1.00000 (0.00001 step).
In case of dB, settable value is 0.000 dB to -96.000 dB (0.001 dB step).

Gain increase/decrease function is applied on period except ATTACK/RELEASE in FSLEN period (see Figure 3-8).

Figure 3-8



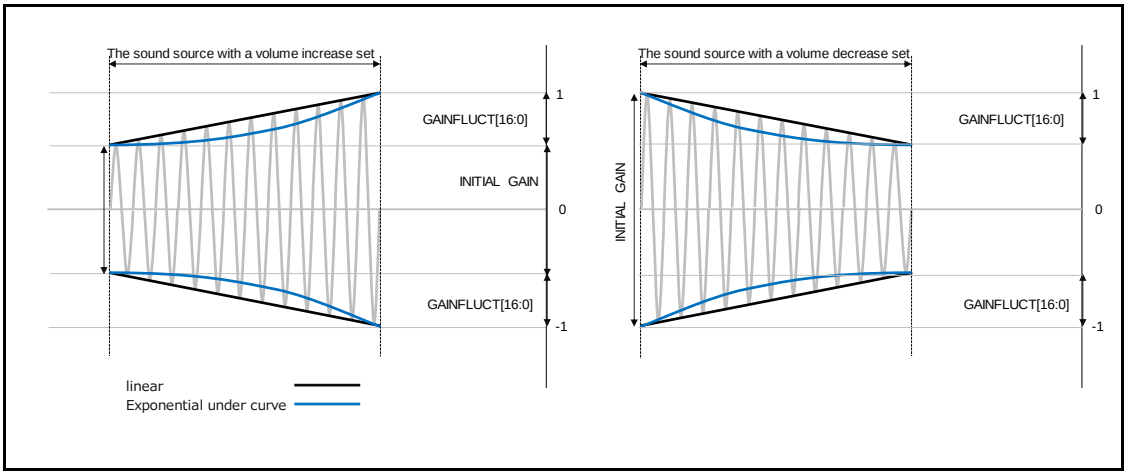
Waveform shape of gain increase/decrease function is as follows and selected by software. (See Figure 3-9)

Shape becomes linear if ratio is selected as unit.

Shape becomes exponential if dB is selected as unit.

Selection of increase/decrease is alternative for each sound source.

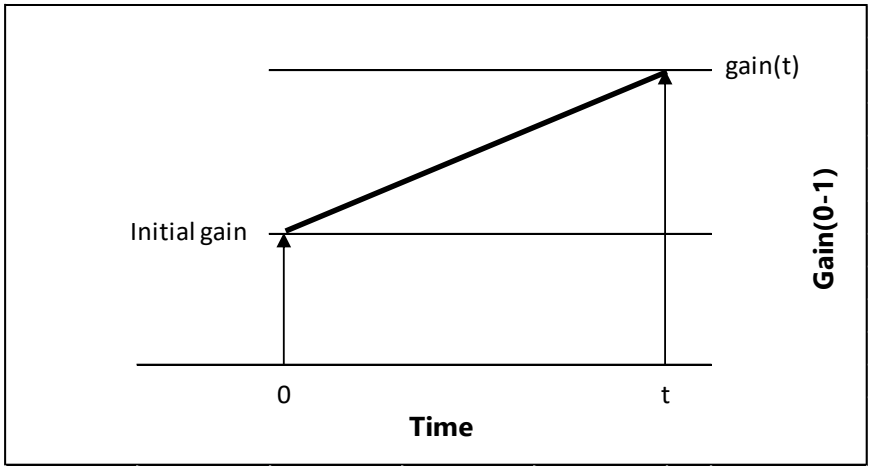
Figure 3-9



3.1.9.1 Gain Increasing with Ratio Setting

Waveform is Figure 3-10.

Figure 3-10



Gain(t) is calculated by following formula.

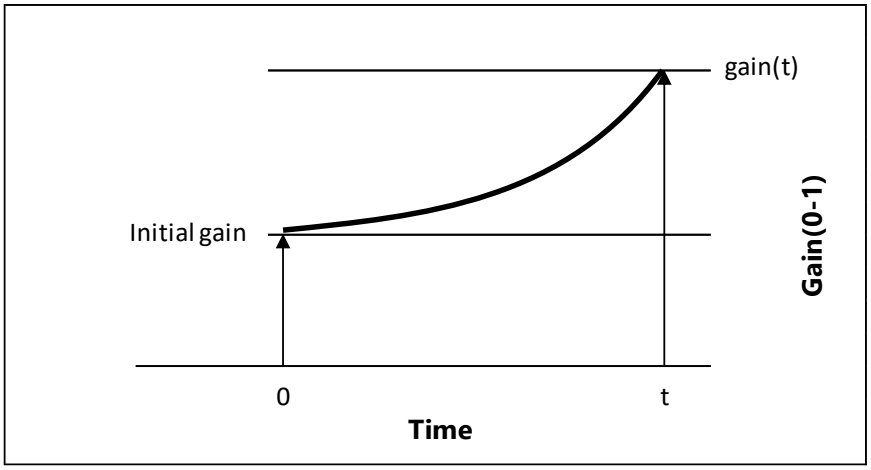
$$gain_slope = gain_fluctuation * 0.001$$
$$gain(t) = initial_gain + gain_slope * t$$

gain_slope :	gain increasing value within 1s period.
gain_fluctuation :	register setting value (4.22 WGCHnCTRL1-GAINFLUCT[16:0])
initial_gain :	initial gain (register setting value or transferred gain)

3.1.9.2 Gain Increasing with dB Setting

Waveform is Figure 3-11.

Figure 3-11



Gain(t) is calculated by following formula.

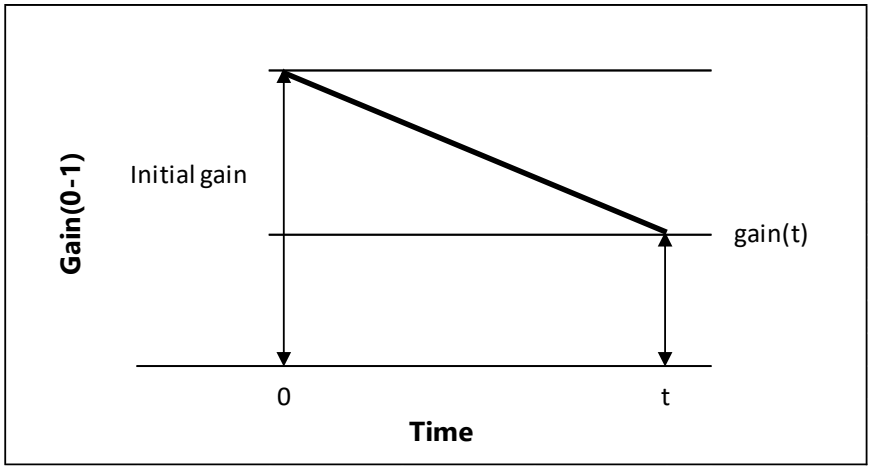
$$gain_slope_db = gain_fluctuation * 0.1$$
$$initial_gain_db = 20 * log10(initial_gain)$$
$$gain_db(t) = initial_gain_db + gain_slope_db * t$$
$$gain(t) = 10^{gain_db(t)/20}$$

gain_slope_db :	gain increasing value in dB within 1s period.
gain_fluctuation :	register setting value (4.22 WGCHnCTRL1-GAINFLUCT[16:0])
initial_gain_db :	initial gain in dB (register setting value or transferred gain in dB)

3.1.9.3 Gain Decreasing with Ratio Setting

Waveform is Figure 3-12.

Figure 3-12



Gain(t) is calculated by following formula.

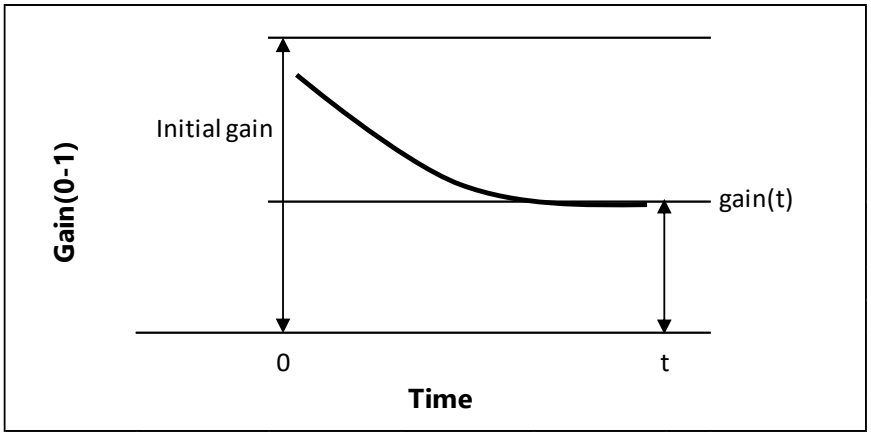
$$gain_slope=(-1.0)*gain_fluctuation*0.001$$
$$gain(t)=initial_gain+gain_slope*t$$

gain_slope :	gain decreasing value within 1s period.
gain_fluctuation :	register setting value (4.22 WGCHnCTRL1-GAINFLUCT[16:0])
initial_gain :	initial gain (register setting value or transferred gain)

3.1.9.4 Gain Decreasing with dB Setting

Waveform is Figure 3-13.

Figure 3-13



Gain(t) is calculated by following formula.

$$gain_slope_db=(-1.0)*gain_fluctuation*0.1$$
$$initial_gain_db=20*log10(initial_gain)$$
$$gain_db(t)=initial_gain_db+gain_slope_db*t$$
$$gain(t)=10^{gain_db(t)/20}$$

gain_slope_db : gain decreasing value in dB within 1s period.
gain_fluctuation : register setting value (4.22 WGCHnCTRL1-GAINFLUCT[16:0])
initial_gain_db : initial gain in dB (register setting value or transferred gain in dB)

3.1.10. Combinational Usage of Each Functions

Operation of attack function varies according to setting of gain transfer. (See Table 3-4)

In case of gain transfer disabled, gain fluctuates from gain at completion of previous sound source generation to initial gain. Its period is attack time which is set by software.
In case of gain transfer enabled, gain keeps its value at completion of previous sound source generation during attack time which is set by software. In this case, attack function does not work.

Table 3-4

ATTACK	Gain Transfer (GAINSUCEN)	ATTACK Operation	Remarks
Unused	Disabled	Gain immediately goes to initial gain which is set by software. Attack function does not work.	Figure 3-14
Used	Disabled	Gain fluctuates from gain at completion of previous sound source generation to initial gain.	Figure 3-15
Used/Unused	Enabled	Gain keeps its value at completion of previous sound source generation. Attack function does not work and initial gain register value is ignored.	Figure 3-16

Figure 3-14

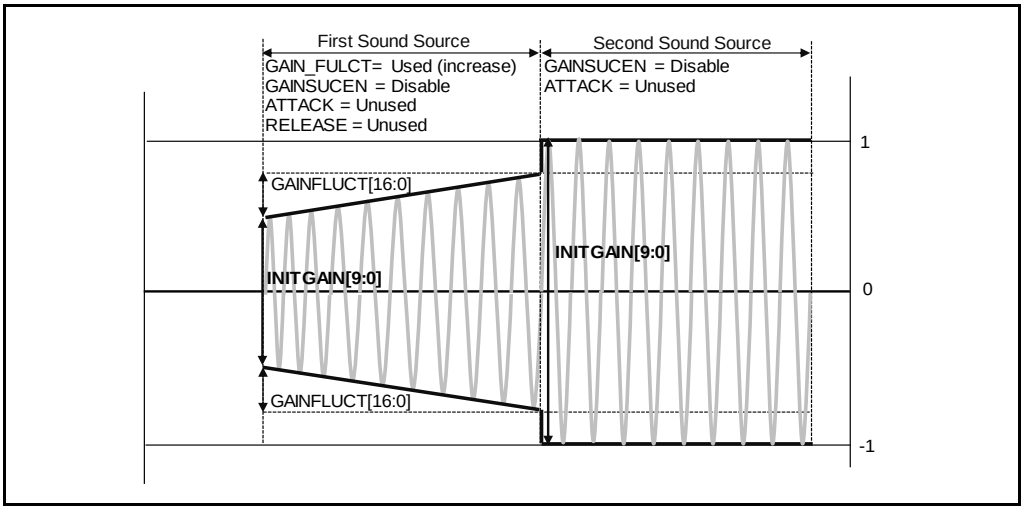


Figure 3-15

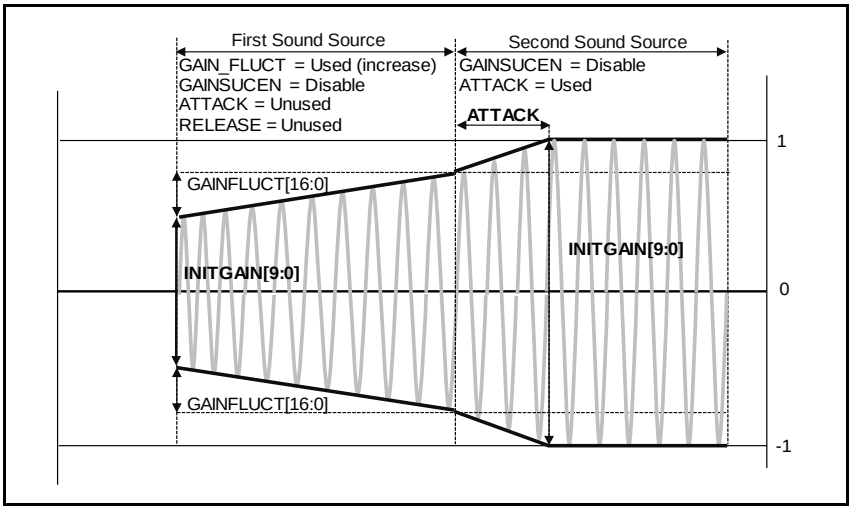
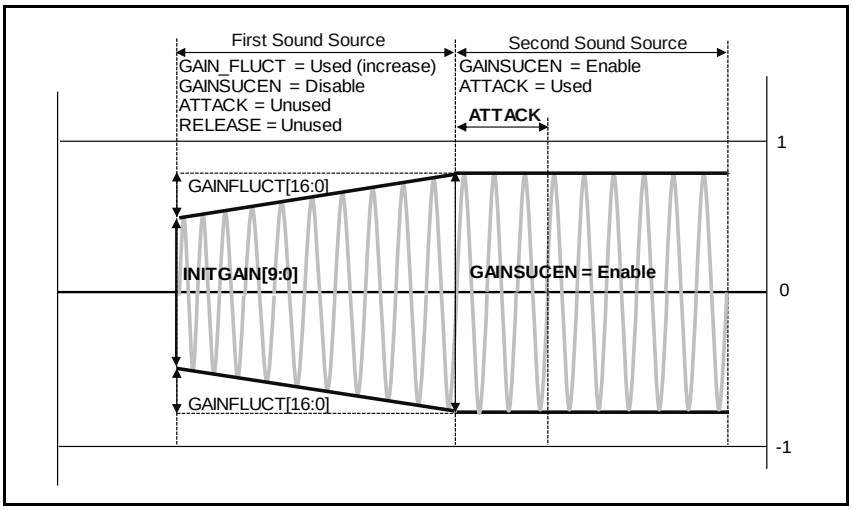


Figure 3-16

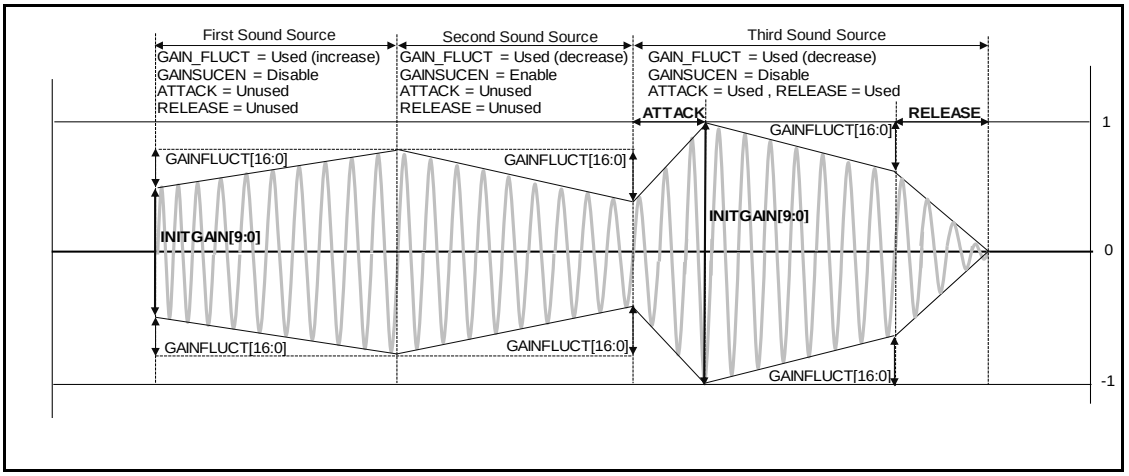


Waveform in Figure 3-17 can be generated in combination with functions in 3.1.5 to 3.1.9. Figure 3-17 is composed by 3 sound source generations. Each setting of 3 sound source generations are as follows.

Sound Source	Gain Increase/Decrease	Gain Transfer	ATTACK	RELEASE
1st	Increase	Disabled	Unused	Unused
2nd	Decrease	Enabled	Unused	Unused
3rd	Decrease	Disabled	Used	Used

As described in the figure, gain can be smoothly changed in combination with gain transfer, attack, and release.

Figure 3-17



Notes:

Calculation of gain has error because of limited bit length for calculation registers.
Therefore 3 usage notes must be taken care in following 2 cases.

- *Both of ratio control and dB control are used in attack/release and gain increase/decrease function in one sound source.*
- *Both of ratio control and dB control are used in gain increase/decrease function between 2 sound sources.*

3 usage notes are followings.

- *When ratio control and dB control are switched, error after switching may appear large. It may happen when gain increase/decrease value is valuated in previous unit.*
- *Figure 3-18 is example of switching from ratio control to dB control. Figure 3-19 is example of switching from dB control to ratio control.*
- *It does not happen in following 3 cases because gain transfer does not work at ratio/dB control switching.*
  1. *Sound source after switching does not use both functions of gain transfer and ATTACK.*
  2. *Sound source before switching uses RELEASE.*
     (Except the case that FSLEN is shorter than ATTACK+RELEASE.)
  3. *Sound source before switching are terminated by forced termination function.*

Figure 3-18

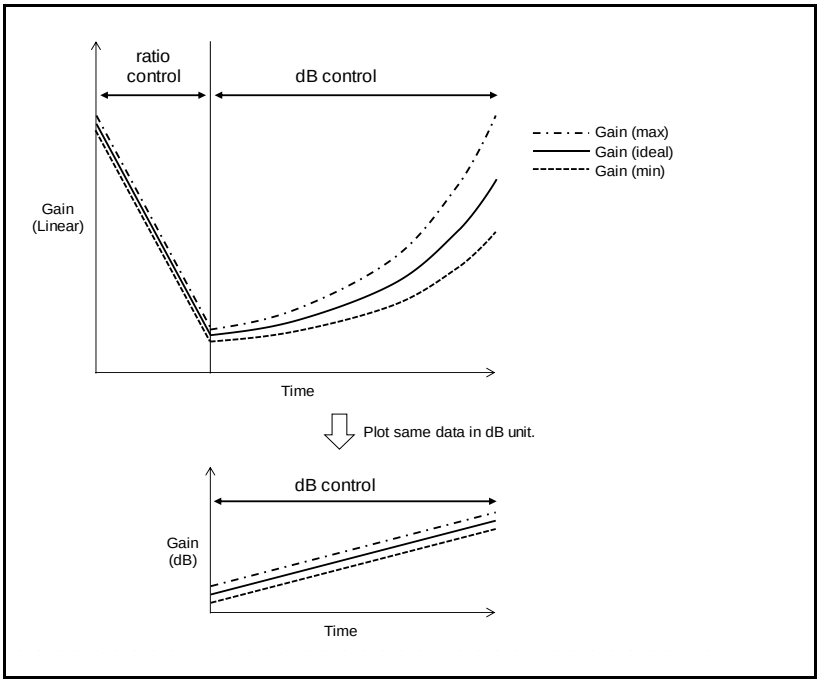
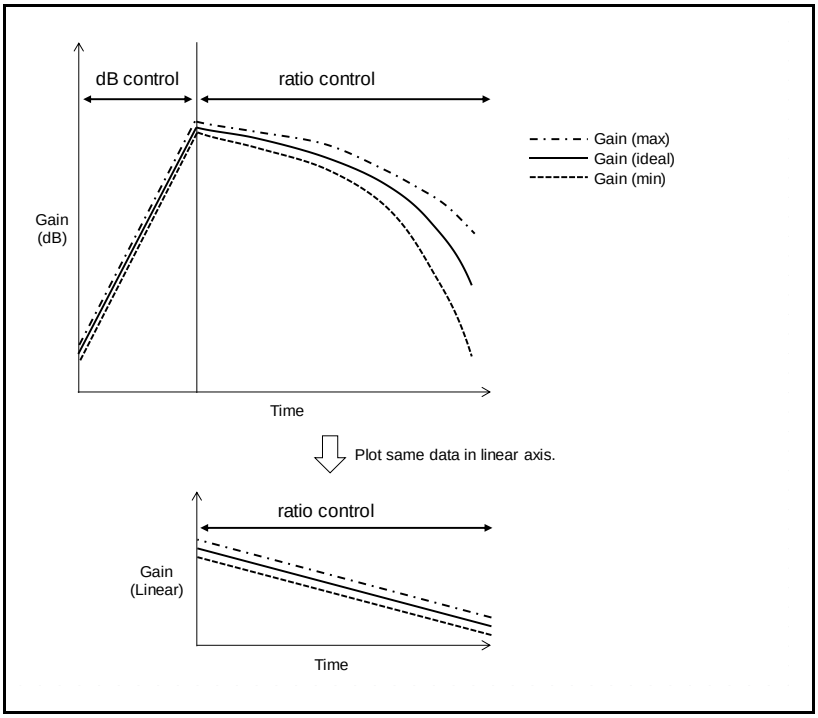


Figure 3-19



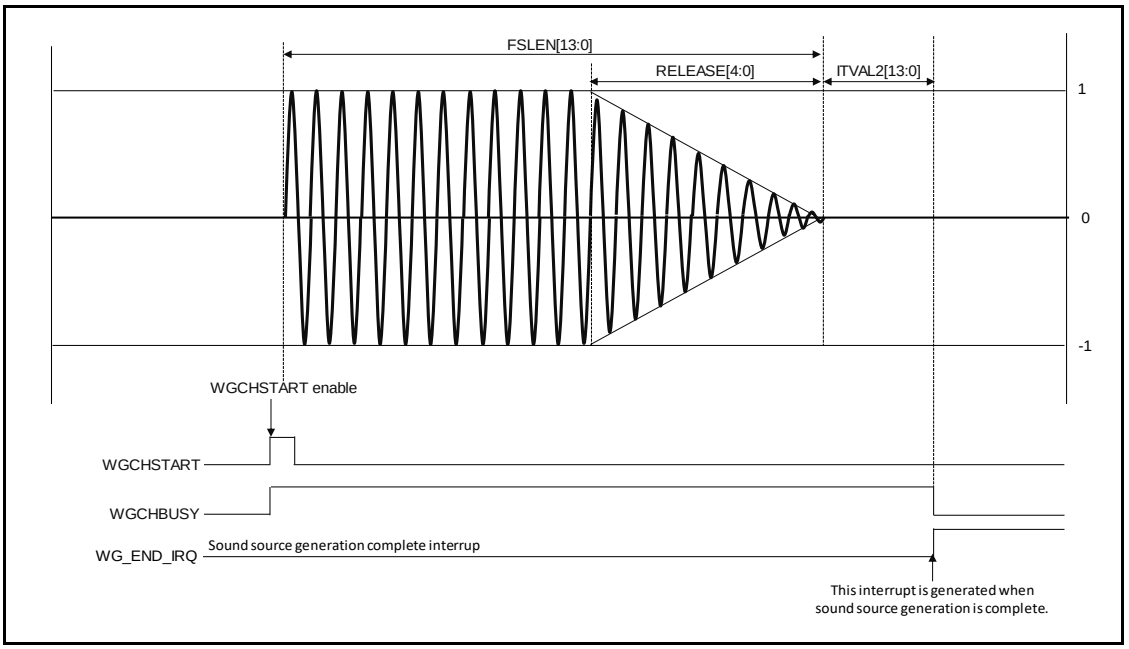
3.2. Status Indicator

3.2.1. Indicate Sound Source Generation Status

SWFG has register bit which indicates that sound source generator is operating. Software can determine if sound source generator is operation by reading this bit. (4.3 WGCHBUSY-CHnBUSY)

- SWFG has dedicated bit for each channel. Initial value is "0". It automatically becomes "1" when sound source generation is initiated by software which writes "1" on WGCHSTART or WGCHnCTRL4-CHnSTART. (See Figure 3-20)
- It automatically goes back to "0" when SWFG completes sound source generation. Posterior mute period is included in sound source generation period.
- In case of forced termination, it automatically goes back to "0" when RELEASE after forced termination is completed. (See Figure 3-27)
- In case of continuous generation, it keeps "1" between 1st sound source generation and final sound source generation. It does not go back to "0" in the middle. (Figure 3-28, Figure 3-29)

Figure 3-20



3.2.2. Indicate Gain Value at Time of Gain Increase/Decrease Completion

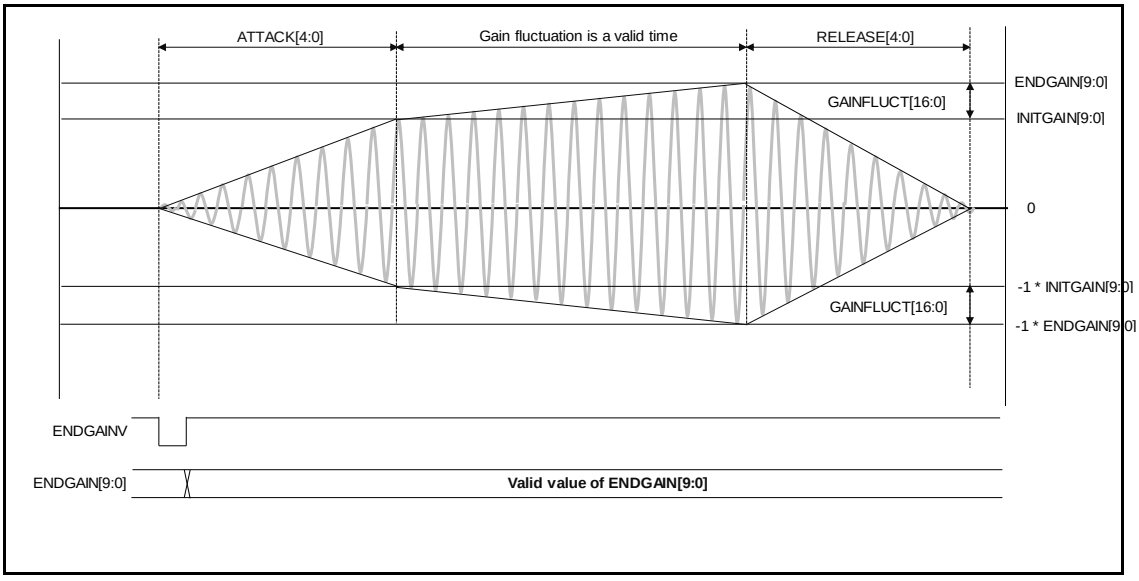
SWFG has register which indicates gain value at time of gain increase/decrease completion (4.26 WGCHnSTATUS-ENDGAIN[9:0]). Software can determine gain value at time of gain increase/decrease completion on current sound source generation by reading this register.

- SWFG has register bit which indicates validity of this gain value (4.26 WGCHnSTATUS-ENDGAINV). This bit is cleared when sound source generation is initiated. This clear function works regardless of whether sound source generations is single operation or continuous operation. (See Figure 3-21)
- Gain value is final value of gain increase/decrease period. It is not reflected final one of ATTACK/RELEASE period. (See Figure 3-21)
- Gain value indicates the value on Table3-6.
- Gain value is not PCM value. It is multiplying factor and its unit is same as initial gain (ratio or dB). Unit is selected by software (4.22 WGCHnCTRL1-DBEN).
- When sound source generation is forcibly terminated by forced termination, gain value is not the value at time of termination but it is the gain value at time of gain increase/decrease completion when not forcibly terminated.

Table 3-5

No	Gain Fluctuation Value	Gain Transfer	Gain Value Indicated by Register
1	0	Disabled	Initial gain
2	0	Enabled	Initial gain (transferred gain)
3	Except 0	Do not care	Increased/decreased gain value

Figure 3-21



3.3. Interrupt

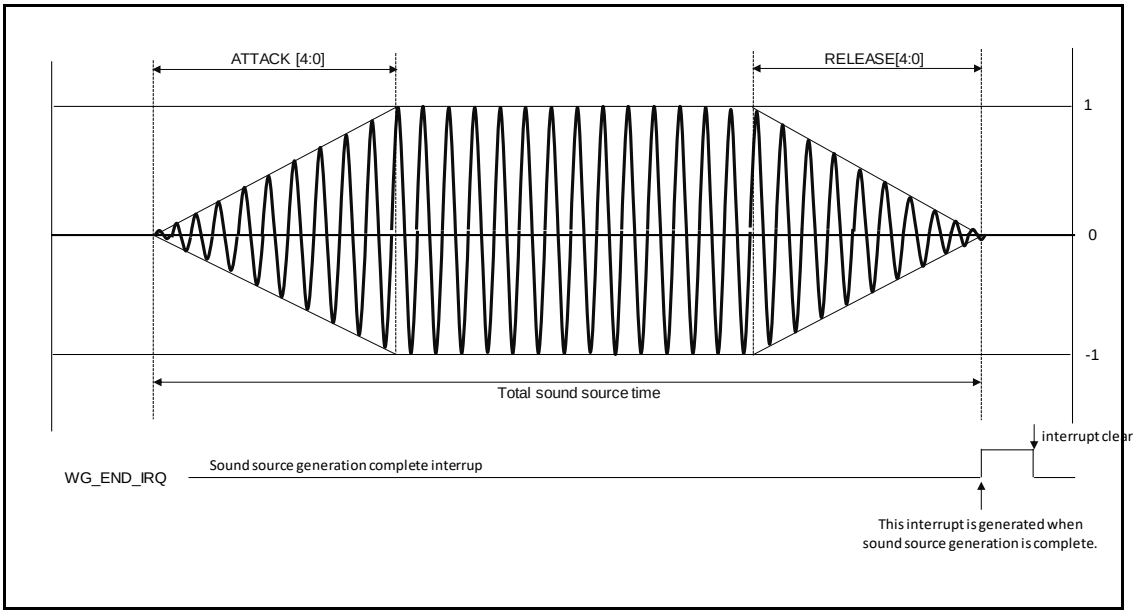
SWFG supports the interrupts described below.

■ Sound source generation complete interrupt (WG_END_IRQ)
This interrupt is generated when sound source generation is complete. SWFG performs interrupt enable setting, interrupt clear control and status indication.
This interrupt factor has register bit which indicates its interrupt status (4.7 WGINTSTATE-CHnEND).
Following 2 cases must be taken care.

1. Completion of final sound source generation on continuous generation.
   Interrupt is generated when final sound source generation is completed on continuous generation which is controlled by continuous number setting register. For the details, see 3.6.
2. Completion of forced termination.
   In case of forced termination during execution of sound source generation, complete interrupt is generated after RELEASE operation. If RELEASE period is set as 0 ms, it is generated just after forced termination. If sound source generation is not initiated, forced termination does not generate forced termination complete interrupt. For the details, see 3.5.

This interrupt factor is informed to CPU if interrupt enable bit is set as "enable".
This interrupt factor is cleared when clear bit is written by "1".

Figure 3-22



■ AHB Master Interface bus errors interrupt.
This interrupt is generated when an error response is received during sound source output to an output destination on the AHB Master Interface of SWFG. SWFG performs interrupt enable setting, interrupt clear control and status indication.
This interrupt factor has register bit which indicates interrupt status (4.7 WGINTSTATE-AHBERR).

This interrupt factor is informed to CPU if interrupt enable bit is set as "enable".
This interrupt factor is cleared when clear bit is written by "1".

3.4. Sound Source Generation Start

The procedure for starting sound source generation is provided below. Each channel can be controlled independently.

1. Set the output destination address of the sound source. See WGCHADD0-2 under 4.10, 4.11 and 4.12.
2. Configure interrupt enable/disable settings. See WGINTREN under 4.6.
3. Configure filter coefficients. See WGCHnFILCOEF0-7 under 4.13 through 4.20.
4. Configure sound source specification settings. See WGCHxCTRL0-4(x=0 through 4) under 4.21 through 4.25.
5. Performs startup control. See WGCHEN under 4.1.
6. Start sound source generation. See WGCHSTART under 4.2.

WGCHSTART has mirror region on 4.25 WGCHnCTRL4-CHnSTART because of convenience for DMA transfer (See 3.11).

Notes:

- *Steps 1 through 4 of the above procedure can be performed in any sequence.*
- *Enabling operation in step 5 of the above procedure causes SWFG to output ALL 0 data. After that, step 6 starts generation of the configured sound source. If SWFG is not enabled, the sound source is not generated when generation start is executed. Changing destination address which is set on step.1 after SWFG is enabled, data transfer does not correctly work because of address mismatch.*
- *Even if operation is enabled and sound source generation startup control is performed, it will not start without a data request from the SWFG output destination. The output destination needs to be set beforehand.*
- *WGCHnFILCOEF0-7 on 4.13 to 4.20 and WGCHxCTRL0-4 on 4.21 to 4.25 are parameters for sound source generation. Sound source will be generated based on these register settings when sound source generation is initiated on step.6. After initiation, updating these register is not reflected on initiated sound source generation.*

3.5. Sound Source Generation Stop

SWFG operation that started sound source generation can be stopped on a channel-by-channel basis. See WGCHEN under 4.1 and WGCHSTOP under 4.2.

Sound source generation stopped by WGCHEN under 4.1 can be resumed by performing startup control on the same bit, which will resume from the point where generation was stopped. There is no sound source output, including ALL 0, from channels that are stopped by this control. The effect on sound source output destinations should be kept firmly in mind when stopping sound source generation.

Sound source generation can be forcibly terminated by 4.4 WGCHSTOP-CHnSTOP. This operation is called as "Forced Termination" in TRM. Followings are details about forced termination.

- WGCHSTOP is automatically cleared to "0" after completion of forced termination. (Figure 3-23)
- It is forbidden to write "1" to WGCHSTART and WGCHnCTRL4-CHnSTART on corresponding channel from initiation of forced termination (writing "1" to WGCHSTOP) to completion of forced termination (automatic clearing WGCHSTOP to "0").
- Forced termination to sound source generation with DMA transfer has limitation that forced termination must be initiated after DMA transfer is stopped. Following procedure must be applied.
    1. Send stop request to DMA and confirm that DMA is stopped.
    2. Write "0" to 4.5 WGCHDMAEN-CHnDMAEN on corresponding channel.
    3. Initiate forced termination.
- Internal retained values of gain/ phase/ filters are initialized after forced termination. Initialized values are same as one just after reset.
- Forced termination does not initialize any registers except WGCHSTART and WGCHnCTRL4-CHnSTART.
- Sound source generation which is terminated by forced termination cannot be resumed. If initiating terminated sound source generation again after completion of forced termination, sound source generation is initiated but it is not resuming. It is newly initiated.
- Forced termination can stop continuous sound source generation (3.6 Continuous SWFG Sound Source Generation) also. In this case, sound source generation at the timing of forced termination execution becomes final one. After completion of that, continuous sound source generation is terminated.
- The channel which is termination by forced termination outputs silence sound that PCM data is "0" if WGCHEN is set as enable. Time period to output silence sound is defined as below.
    1. In case of that release time of forced termination defined by 4.21 WGCHnCTRL0-STOPREL[2:0] is not equal to 0 ms.
       After release operation within the timer period defined by STOPREL[2:0], PCM data goes to "0". After "0" data is forcibly outputted 31 times (approx. 0.6 ms), operation of the forced termination is completed. That is for cleaning of filter pipe line (Figure 3-23).
    2. In case of that release time of forced termination defined by 4.21 WGCHnCTRL0-STOPREL[2:0] is equal to 0 ms.
       PCM data goes to "0" just after WGCHSTOP bit is enabled. Sound source generation does not operate release and forcible 31 "0" data outputs. (Figure 3-24)
- Shape of release waveform is linear only at forced termination and exponential cannot be selected.

Notes:

- *Remaining time of total sound source time is overwritten by release time of forced termination. Therefore total sound source time becomes longer than the time defined by setting register in certain case even though the motivation of forced termination is earlier terminating sound source*

generation (See 3-23). If forced termination is executed during RELEASE period of the sound source, release time of forced termination is applied at that moment. (Figure 3-25)

- *When forced termination is executed during anterior mute period or posterior mute period of the sound source, sound source generation is terminated after RELEASE operation which is defined by 4.21.WGCHnCTRL0-STOPREL[2:0] and forcible output of 32 "0"s (See 3-26). In case of that RELEASE time of forced termination (4.21) is equal to 0 ms, sound source generation is terminated just after WGCHSTOP is enabled.*

Figure 3-23

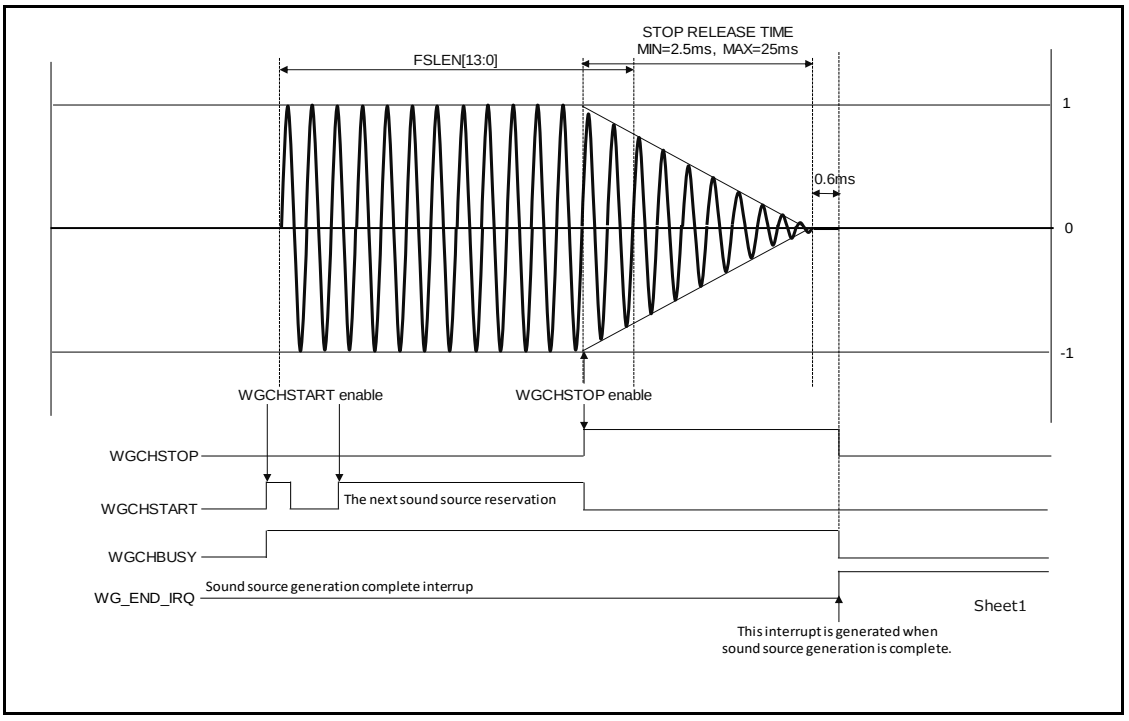


Figure 3-24

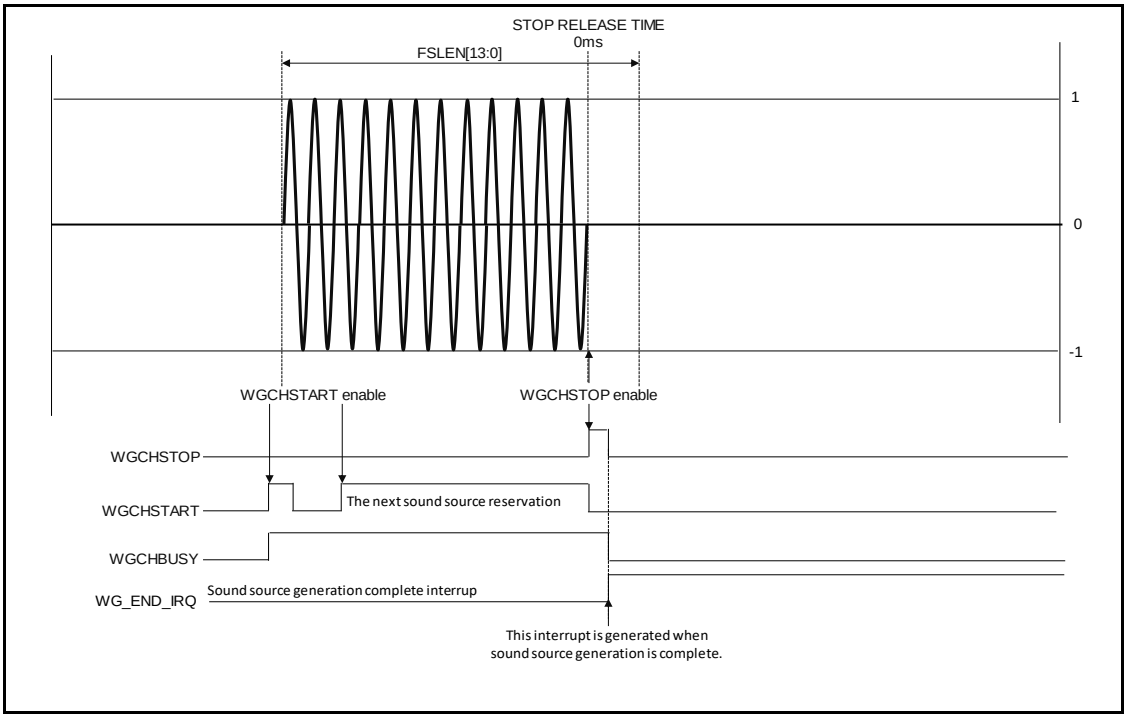


Figure 3-25

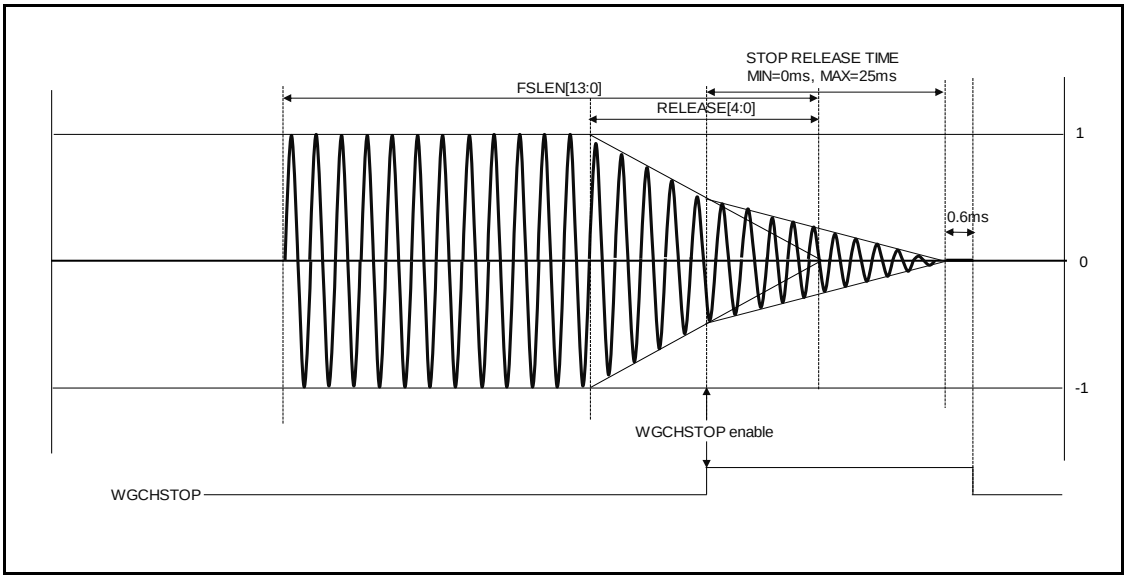
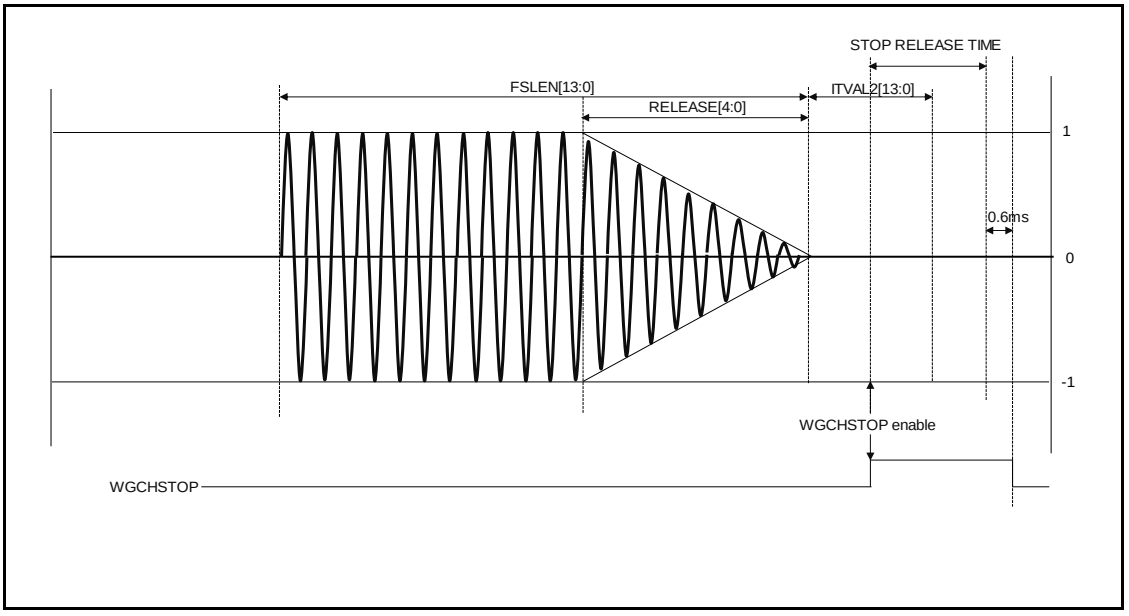


Figure 3-26


STOP RELEASE TIME
FSLEN[13:0]
RELEASE[4:0]
ITVAL2[13:0]
1
0.6ms
0
-1
WGCHSTOP enable
WGCHSTOP

3.6. Continuous SWFG Sound Source Generation

There are 2 methods for continuous sound source generation.

1. Software controlled generation
2. Hardware controlled generation

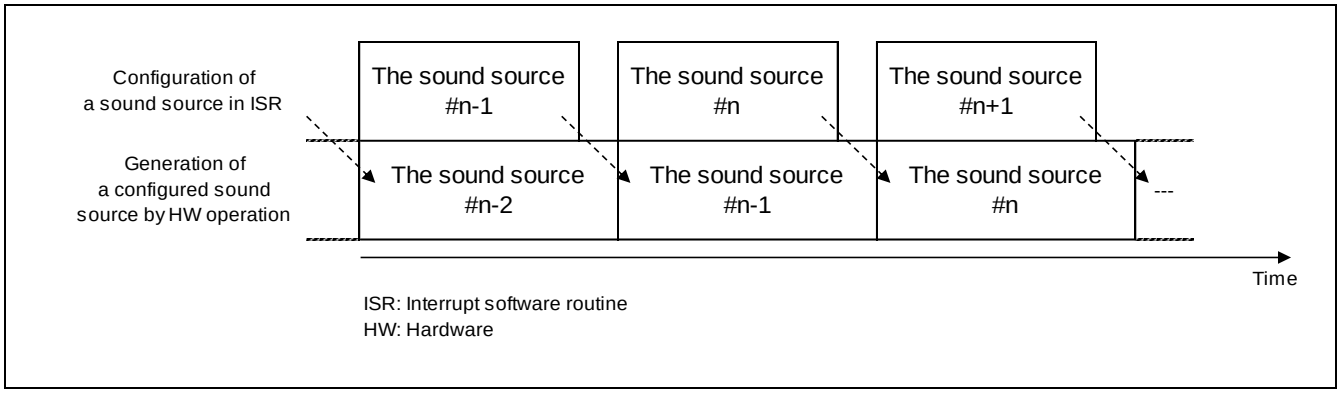
Note:

- *In case of same type of waveform (See 3.1.2), next sound source generation takes over its phase from previous sound source generation. If the type of waveform is changed, phase cannot be taken over.*

3.6.1. Software Controlled Continuous Sound Source Generation

Setting of the next sound source specification can be performed by starting sound source generation using the procedure under 3.1, even if a previous sound source generation operation is in progress. After setting is complete, sound source generation starts and the corresponding WGCHSTART bit or WGCHnCTRL4-CHnSTART bit is set to 1, so generation of the next sound source starts as soon as generation of the current sound source is complete.

Figure 3-27



Notes:

- *Setting the next sound's sound source specifications after the previous sound source generation is complete can cause an audible break in the sound. Consecutive sound generation that uses a sound source generation complete interrupt as shown in Figure 3-27 is recommended.*
- *Using attack and release control between the previous generation and the current generation when consecutively generating sound sources can cause an audible break in the sound. When performing consecutive generation, application of attack and release, control is recommended only when setting initial and final sound source specifications.*

3.6.2. Hardware Controlled Continuous Sound Source Generation

How many times SWFG continues sound source generation is set by software (4.21 WGCHnCTRL1-CONTNUM[3:0]). Available number is 1 through 15 and infinite. In case of infinite, forced termination can stop it (3.5).

Procedure for this function is as follows.

1. Set number except "0" to WGCHnCTRL1-CONTNUM[3:0].
2. Set "1" to WGCHSTART or WGCHnCTRL4-CHnSTART on corresponding channel.

In this operation, settings at initiation of continuous sound source generation are applied on whole generations. Settings cannot be changed in the middle.
4.26 WGCHnSTATUS - CONTNUMR[3:0] shows remaining number of times.

3.6.2.1 Example 1

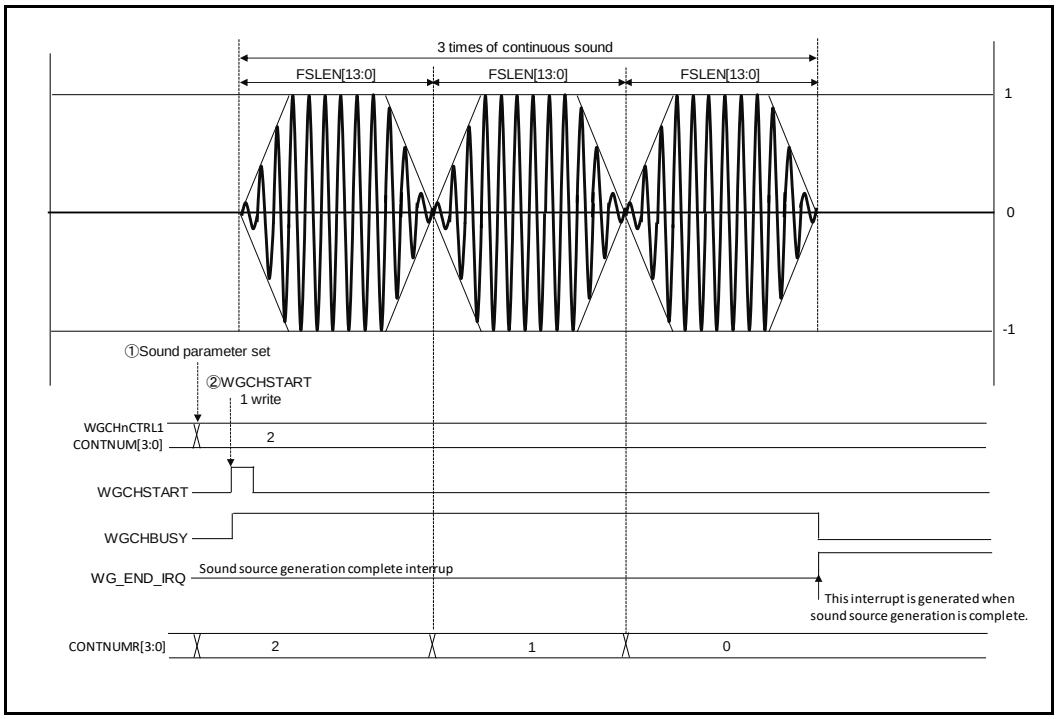
Same waveform is only required for continuous sound source generation as Figure 3-28.
In this case, following procedure is required.

1. Set the number of times to setting register with any other waveform setting registers.

2. Write "1" to WGCHSTART or WGCHnCTRL4-CHnSTART on corresponding channel.

Completion interrupt (WG_END_IRQ) is asserted when all of sound source generation is completed.
WGCHSTART/ WGCHnCTRL4-CHnSTART are automatically cleared to "0" when sound source generation is initiated.

Figure 3-28



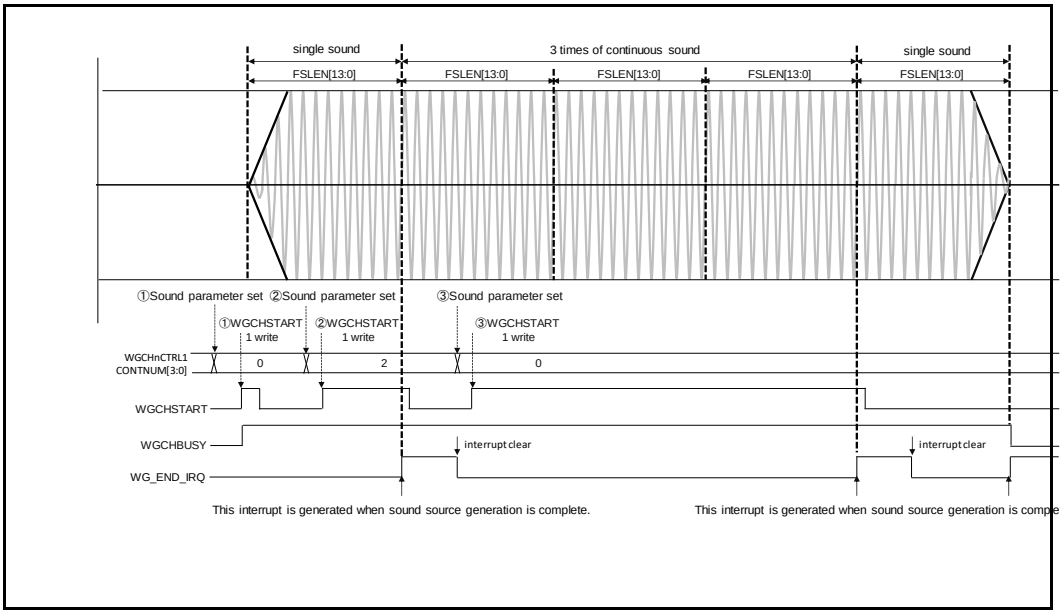
3.6.2.2 Example 2

It is combination of continuous sound source generation and single sound source generation as Figure 3-29.

In this case, following procedure is required.

1. Generate 1st sound source as single sound source generation.
2. Set the parameters for 2nd sound source as continuous sound source generation during 1st sound source generation period. (Interrupt happens when next parameter setting is available.)
3. Write "1" to WGCHSTART or WGCHnCTRL4-CHnSTART during 1st sound source generation period.
4. Set the parameters for 3rd sound source as single sound source generation during 2nd sound source generation period. (Interrupt happens when next parameter setting is available.)
5. Write "1" to WGCHSTART or WGCHnCTRL4-CHnSTART during 2nd sound source generation period.

Figure 3-29



3.7. Output Data Format

SWFG generates 16bit monaural PCM data stream. Output PCM data format is 32bit stereo PCM. Therefore lacked 16bit data is complemented by SWFG. How to complement can be selected by software.

In stereo mode, lacked 16bit is filled by same 16bit monaural PCM and becomes 32bit stereo PCM.
In monaural mode, lacked 16bit is filled by "0".

Filled region can be selected by software. For details, see Table 3-6 and 4.23 WGCHnCTRL2-MONO[1:0].

Table 3-6

WGCHnCTRL2-MONO[1:0]	OUTPUT PCM DATA[31:0]	
	[31:16]	[15:0]
00	WFG Generate Data[15:0]	WFG Generate Data[15:0]
01	"00000000_00000000"	WFG Generate Data[15:0]
10	WFG Generate Data[15:0]	"00000000_00000000"
11	Prohibited	

3.8. Filter

Harmonic overtones generated by triangle or square waveform can be filtered.

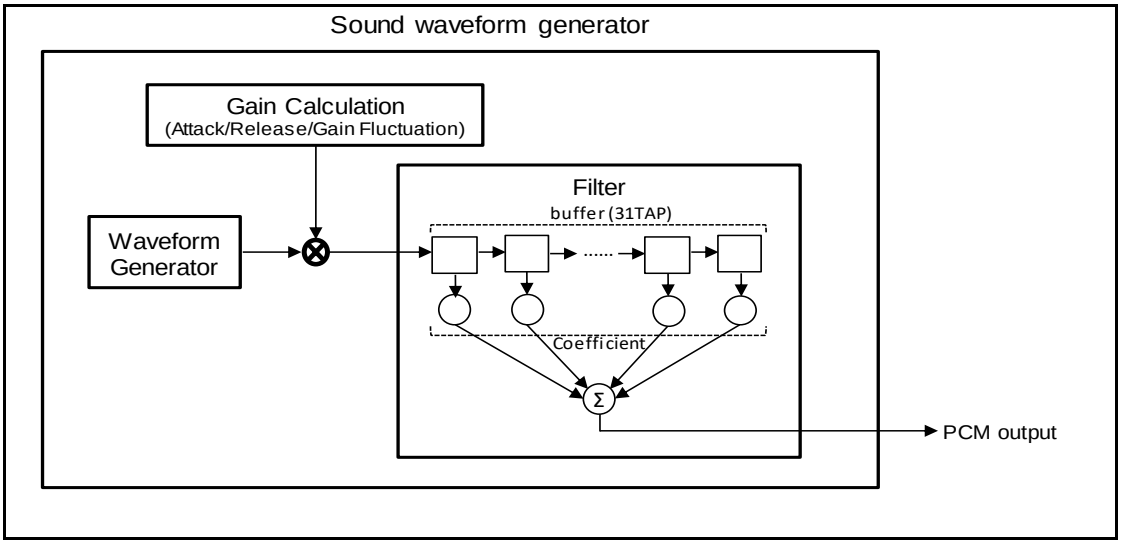
- It is FIR filter and it has 31 taps.
- Cut off frequency can be defined by software because coefficients are flexibly defined by software (See 4.13 to 4.20).
- SWFG has 16 coefficients for each channel.
- Coefficient is signed 16bit length. It is less than 1.0 and more than -1.0. (Refer to FILCOEF0[15:0] to FILCOEF15[15:0])
- It is symmetrical FIR filter. Center of coefficients is FILCOEF0. Convolution formula is as follow.

$$f(\mathrm{x}) = \mathrm{buff}[15] * FILCOEF[0] + \sum_{n=1}^{15}(\mathrm{buff}[15-\mathrm{n}] + \mathrm{buff}[15+\mathrm{n}]) * FILCOEF[n]$$

buff[n] : input PCM data
FILCOEF[n]: filer coefficients

- FIR filter can be disabled by software (4.21 WGCHnCTRL0-FILEN).
- Several examples are shown in Table3-10 to 3-14 and Figure3-31 to 3-35.
- Filtering operation is applied on waveform which is adjusted its gain by ATTACK, RELEASE, or gain increase/decrease operation (Figure 3-30).
- Internal buffers for convolution are cleared to silence ("0") after forced termination.

Figure 3-30



Note:

- *PCM data streaming is buffered by internal buffer for convolution calculation. If coefficients are updated without buffer clear, SWFG may output noise. To avoid it, internal buffer must be filled by silence PCM data before update of coefficients.*

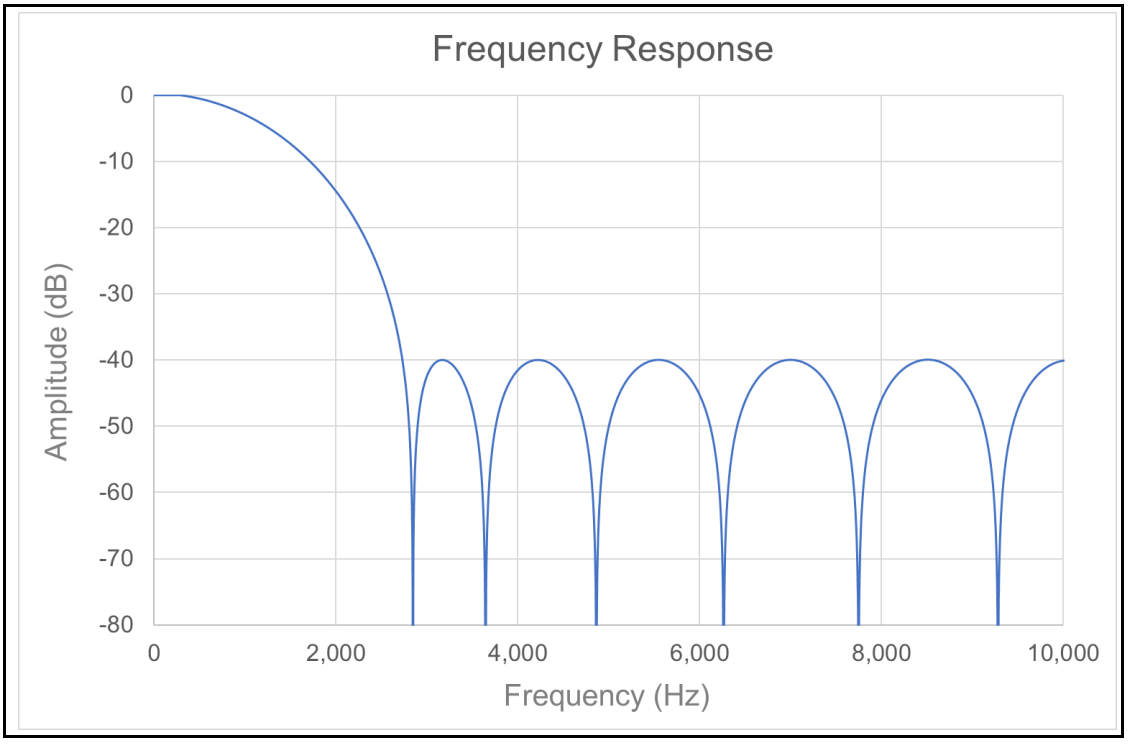
3.8.1. Example.1

Figure 3-31 shows frequency response when the filter coefficients are set as Table 3-7.

Table 3-7

FILTER SETTING (n=0 to 4)	Details	
WGCHnFILCOEF0	FILCOEF1[15:0] = 0x0771	FILCOEF0[15:0] = 0x0785
WGCHnFILCOEF1	FILCOEF3[15:0] = 0x06EE	FILCOEF2[15:0] = 0x0743
WGCHnFILCOEF2	FILCOEF5[15:0] = 0x05FD	FILCOEF4[15:0] = 0x0687
WGCHnFILCOEF3	FILCOEF7[15:0] = 0x04C5	FILCOEF6[15:0] = 0x0572
WGCHnFILCOEF4	FILCOEF9[15:0] = 0x0373	FILCOEF8[15:0] = 0x0431
WGCHnFILCOEF5	FILCOEF11[15:0] = 0x0230	FILCOEF10[15:0] = 0x02F4
WGCHnFILCOEF6	FILCOEF13[15:0] = 0x0118	FILCOEF12[15:0] = 0x01E9
WGCHnFILCOEF7	FILCOEF15[15:0] = 0x00BD	FILCOEF14[15:0] = 0x0131

Figure 3-31



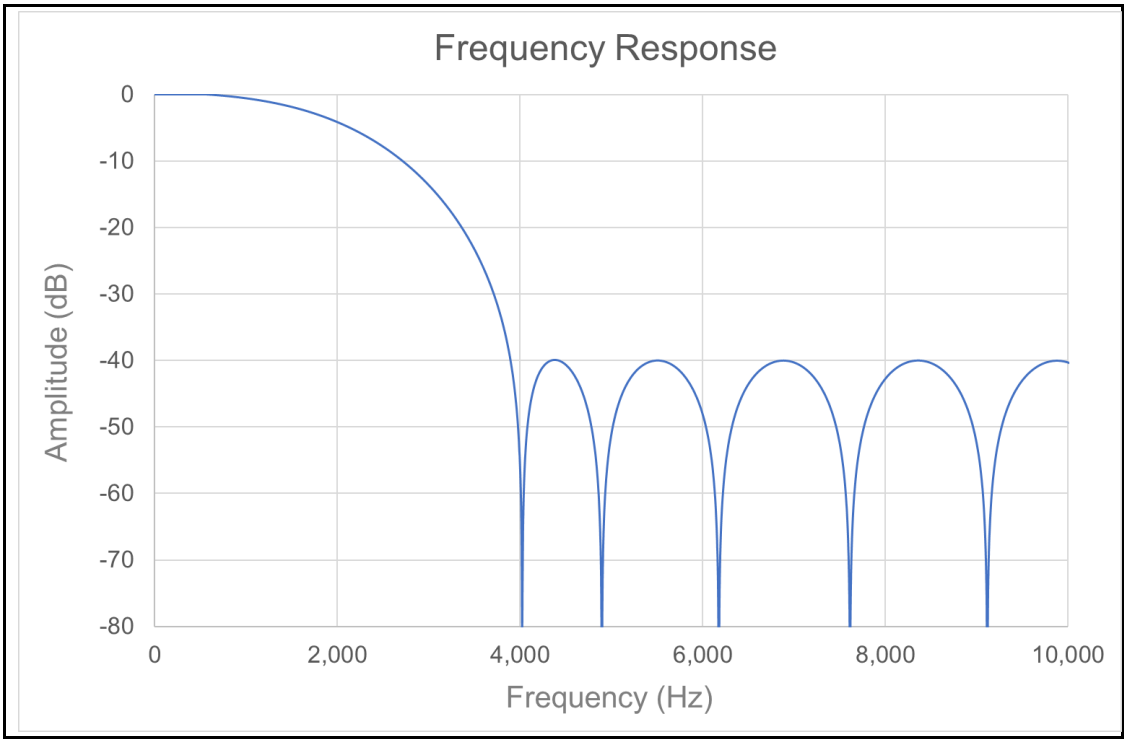
3.8.2. Example.2

Figure 3-32 shows frequency response when the filter coefficients are set as Table 3-8.

Table 3-8

FILTER SETTING (n=0 to 4)	Details	
WGCHnFILCOEF0	FILCOEF1[15:0] = 0x0BEC	FILCOEF0[15:0] = 0x0C2A
WGCHnFILCOEF1	FILCOEF3[15:0] = 0x0A1C	FILCOEF2[15:0] = 0x0B37
WGCHnFILCOEF2	FILCOEF5[15:0] = 0x071D	FILCOEF4[15:0] = 0x08B4
WGCHnFILCOEF3	FILCOEF7[15:0] = 0x03E4	FILCOEF6[15:0] = 0x0578
WGCHnFILCOEF4	FILCOEF9[15:0] = 0x014F	FILCOEF8[15:0] = 0x027B
WGCHnFILCOEF5	FILCOEF11[15:0] = 0xFFD1	FILCOEF10[15:0] = 0x006C
WGCHnFILCOEF6	FILCOEF13[15:0] = 0xFF5B	FILCOEF12[15:0] = 0xFF7A
WGCHnFILCOEF7	FILCOEF15[15:0] = 0xFF15	FILCOEF14[15:0] = 0xFF64

Figure 3-32



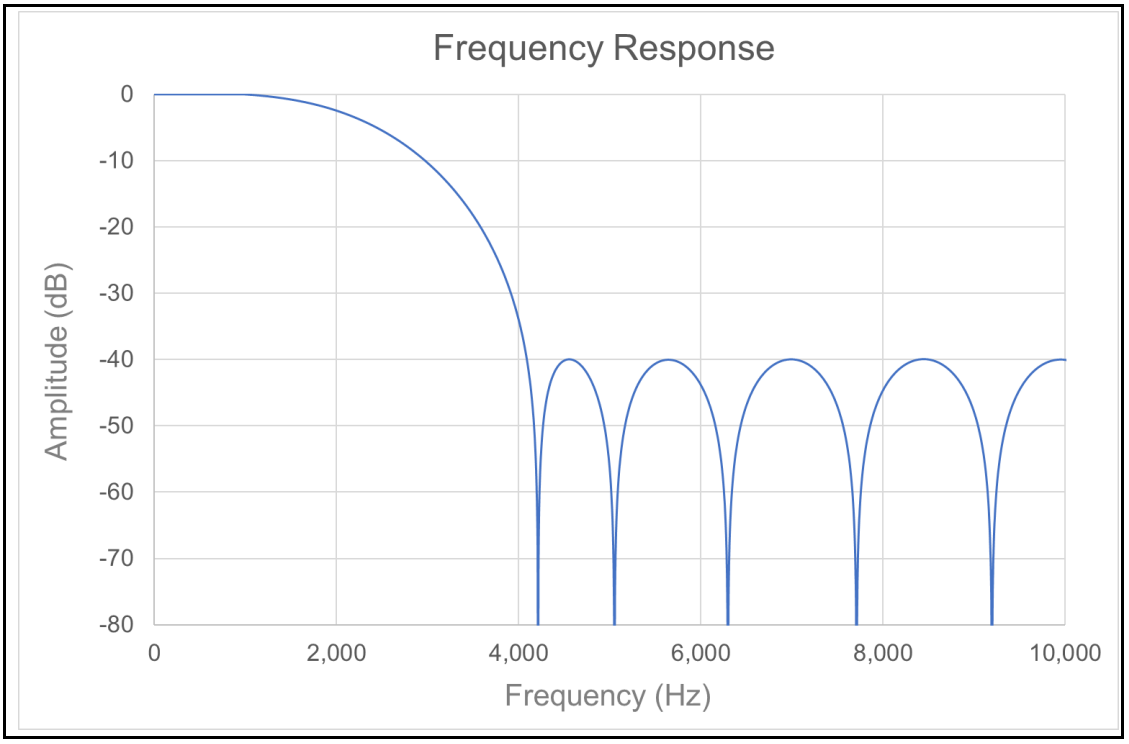
3.8.3. Example.3

Figure 3-33 shows frequency response when the filter coefficients are set as Table 3-9.

Table 3-9

FILTER SETTING (n=0 to 4)	Details	
WGCHnFILCOEF0	FILCOEF1[15:0] = 0x0D77	FILCOEF0[15:0] = 0x0DC9
WGCHnFILCOEF1	FILCOEF3[15:0] = 0x0B1B	FILCOEF2[15:0] = 0x0C8A
WGCHnFILCOEF2	FILCOEF5[15:0] = 0x074A	FILCOEF4[15:0] = 0x094C
WGCHnFILCOEF3	FILCOEF7[15:0] = 0x035C	FILCOEF6[15:0] = 0x0541
WGCHnFILCOEF4	FILCOEF9[15:0] = 0x0075	FILCOEF8[15:0] = 0x01BB
WGCHnFILCOEF5	FILCOEF11[15:0] = 0xFF0D	FILCOEF10[15:0] = 0xFF91
WGCHnFILCOEF6	FILCOEF13[15:0] = 0xFEE7	FILCOEF12[15:0] = 0xFEDB
WGCHnFILCOEF7	FILCOEF15[15:0] = 0xFEFA	FILCOEF14[15:0] = 0xFF1A

Figure 3-33



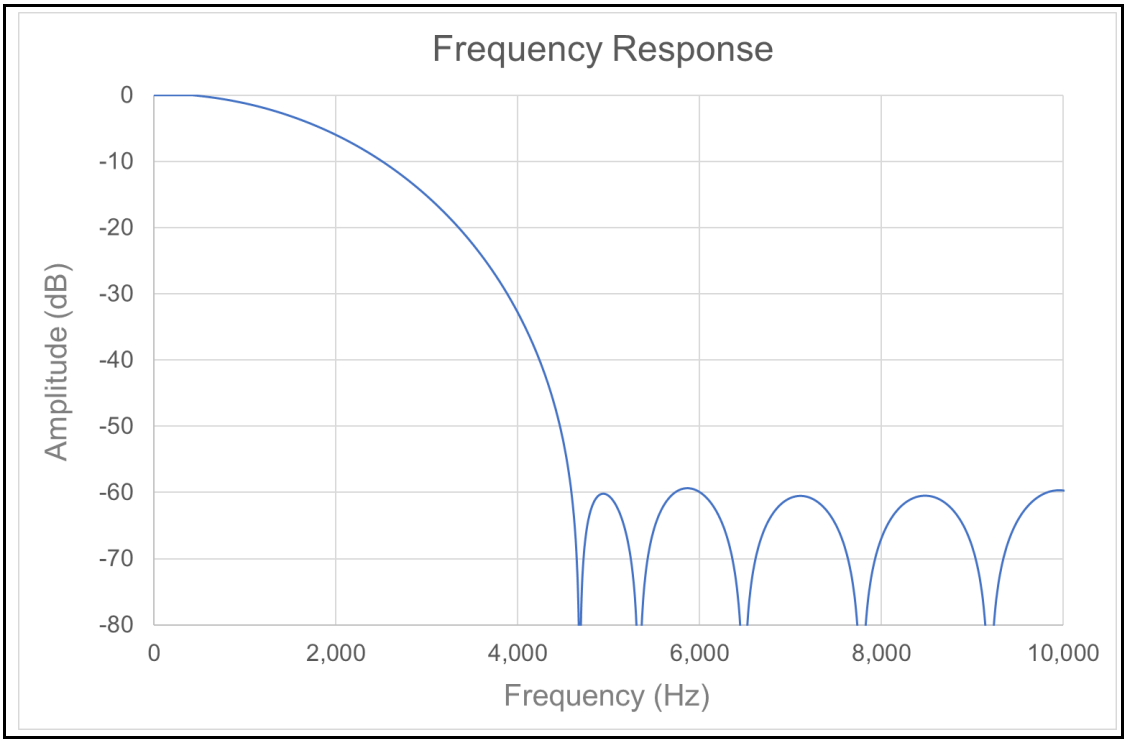
3.8.4. Example.4

Figure 3-34 shows frequency response when the filter coefficients are set as Table 3-10.

Table 3-10

FILTER SETTING (n=0 to 4)	Details	
WGCHnFILCOEF0	FILCOEF1[15:0] = 0x0ADB	FILCOEF0[15:0] = 0x0B12
WGCHnFILCOEF1	FILCOEF3[15:0] = 0x0941	FILCOEF2[15:0] = 0x0A3A
WGCHnFILCOEF2	FILCOEF5[15:0] = 0x06A5	FILCOEF4[15:0] = 0x0805
WGCHnFILCOEF3	FILCOEF7[15:0] = 0x03EA	FILCOEF6[15:0] = 0x053E
WGCHnFILCOEF4	FILCOEF9[15:0] = 0x01C6	FILCOEF8[15:0] = 0x02BD
WGCHnFILCOEF5	FILCOEF11[15:0] = 0x0085	FILCOEF10[15:0] = 0x0109
WGCHnFILCOEF6	FILCOEF13[15:0] = 0x0006	FILCOEF12[15:0] = 0x0032
WGCHnFILCOEF7	FILCOEF15[15:0] = 0xFFE8	FILCOEF14[15:0] = 0xFFF5

Figure 3-34



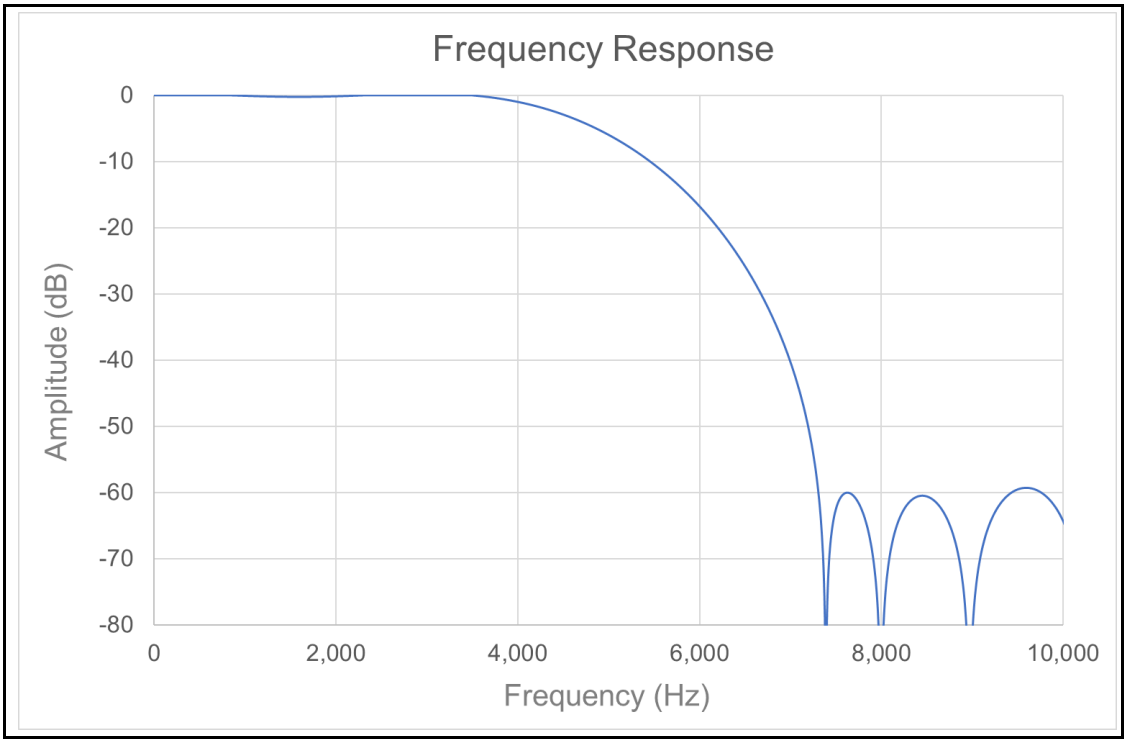
3.8.5. Example.5

Figure 3-35 shows frequency response when the filter coefficients are set as Table 3-11.

Table 3-11

FILTER SETTING (n=0 to 4)	Details	
WGCHnFILCOEF0	FILCOEF1[15:0] = 0x18E5	FILCOEF0[15:0] = 0x1AF7
WGCHnFILCOEF1	FILCOEF3[15:0] = 0x0BCF	FILCOEF2[15:0] = 0x134D
WGCHnFILCOEF2	FILCOEF5[15:0] = 0xFEE1	FILCOEF4[15:0] = 0x0470
WGCHnFILCOEF3	FILCOEF7[15:0] = 0xFBAA	FILCOEF6[15:0] = 0xFC00
WGCHnFILCOEF4	FILCOEF9[15:0] = 0xFEFA	FILCOEF8[15:0] = 0xFD09
WGCHnFILCOEF5	FILCOEF11[15:0] = 0x016C	FILCOEF10[15:0] = 0x009A
WGCHnFILCOEF6	FILCOEF13[15:0] = 0x0112	FILCOEF12[15:0] = 0x0180
WGCHnFILCOEF7	FILCOEF15[15:0] = 0x002C	FILCOEF14[15:0] = 0x009A

Figure 3-35



3.9. Phase Synchronizer

"SELF CH" and "TARGET CH" are used in this chapter.
"SELF CH" means channel which is adjusted its phase by this function.
"TARGET CH" means channel which has target phase.

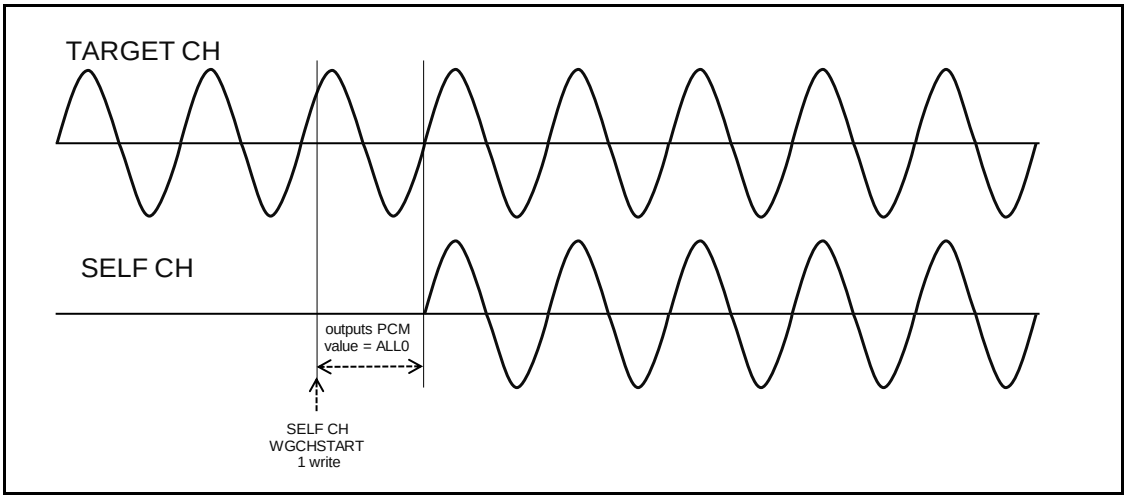
If phase synchronizer of "SELF CH" is activated, "SELF CH" waits the timing which "TARGET CH" outputs first PCM data of its unit cycle. After this waiting time, "SELF CH" is initiated. "SELF CH" outputs silence PCM data during waiting period. Waiting time is not included in FSLEN (Figure 3-8).
See Figure 3-36 for this timing chart.
See Table 3-2 for definition of unit cycle for each type of waveform.
"TARGET CH" is selected by software (4.21 WGCHnCTRL0-ADPHASECH[2:0]).

- Phase synchronizer can be independently enabled for each channel by software (4.21 WGCHnCTRL0-ADPHASEEN).
- Channel should be enabled phase synchronizer is "SELF CH".
- If "SELF CH" is equal to "TARGET CH", phase synchronizer is disabled.
- If "TARGET CH" is in anterior mute period, phase synchronizer is initiated just after anterior mute is completed.
- If "TARGET CH" is in posterior mute period, "SELF CH" is initiated immediately.
- If "SELF CH" has anterior mute period, phase synchronizer is initiated just after anterior mute is completed.
- If "SELF CH" has anterior mute period and "TARGET CH" is in anterior mute period, phase synchronizer is initiated just after both of anterior mute is completed.
- If "TARGET CH" does not generate any waveform (WGCHBUSY=0), phase synchronizer is ignored and "SELF CH" is initiated immediately.

Notes:

- *Phase synchronizer has error. It is +/-3 sampling period.*
- *After initiation of phase synchronizer, "TARGET CH" must not be disabled (WGCHEN=0). If disabled, "SELF CH" may be not initiated because "TARGET CH" is stopped. Therefore, if "TARGET CH" must be stopped, forced termination for "TARGET CH" must be initiated. Then, "TARGET CH" can be stopped by WGCHEN after WGCHSTOP on "TARGET CH" becomes "0".*

Figure 3-36


TARGET CH
SELF CH
outputs PCM
value = ALL0
SELF CH
WGCHSTART
1 write

3.10. Smooth Connection

In the following 3 cases, if smooth connection is activated, SWFG can smoothly connect 2 sound source generations.

- 2 sound source generations are continuously generated.
- Sound source generation becomes silence after completion.
- Current sound source generation has posterior mute period.

Smooth connection is enabled by software (4.21 WGCHnCTRL0-PCMZEROEN). It is applicable for Sine/ Triangle/ Square waveforms.

When the smooth connection is enabled, phase of the next sound source is started from 0 degree regardless of end phase of the previous sound source. This may cause a noise by discontinuity of sound between the sound sources.

If gain is changed between the sound sources, the sound sources are connected at the point that phase of sound source generation is almost 0 degree by using the smooth connection function. This can decrease a noise by discontinuity of sound by avoiding that the amplitude is dynamically changed at the connection point of the sound sources. But, it cannot avoid a noise because discontinuity of sound is remaining between the sound sources. Attack and release functions are recommended to avoid a noise when gain is changed between the sound sources. Please see 3.1.5 Attack Time, 3.1.6 Release Time and 3.1.8 Gain transfer for the details.

If parameter of frequency between the sound sources is changed, a noise by discontinuity of sound cannot be decreased by this function. Because, the discontinuity of sound cannot be improved even if the sound sources are connected at 0 degree of phase of the sound source.

Because of smooth connection, time period of sound source generation may become long in certain cases. If sound source generations are continuously generated, extended time period is deleted from next sound source generation. In this case, the next sound source becomes 2 PCM data (2/48000 sec) longer than calculated sound source time. This operation is illustrated in Figure 3-37.

Without smooth connection, parameters of sound source generations are switched at point A in Figure 3-37 and disconnection happens because amplitudes of 2 sound source generations are different.

With smooth connection, parameters of sound source generations are switched at point B in Figure 3-37. At point B, phase of sound source generation is almost 0 degree and PCM data is almost "0" also. Therefore, 2 sound source generations are smoothly connected. In this case, total sound source time of second sound source generation is shortened by "B-A".
If second sound source generation has anterior mute period, extended time period is deleted from anterior mute period (ITVAL1) as illustrated in Figure 3-38.
If sound source generation has posterior mute period, extended time period is deleted from posterior mute period (ITVAL2) as illustrated in Figure 3-39.

Smooth connection is set on "First Sound source" in Figure 3-37.

Figure 3-37

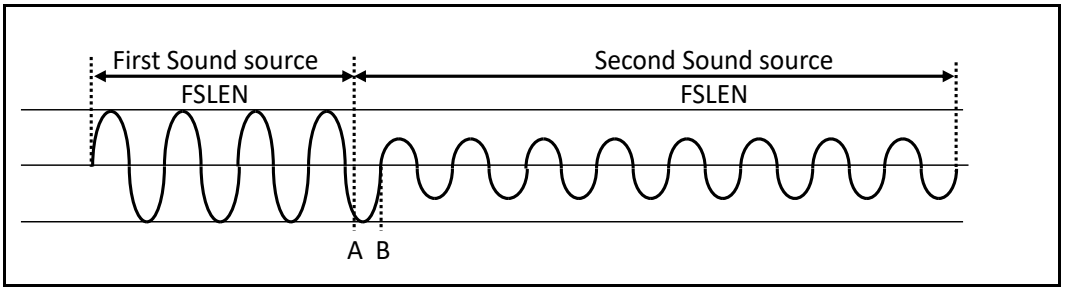


Figure 3-38

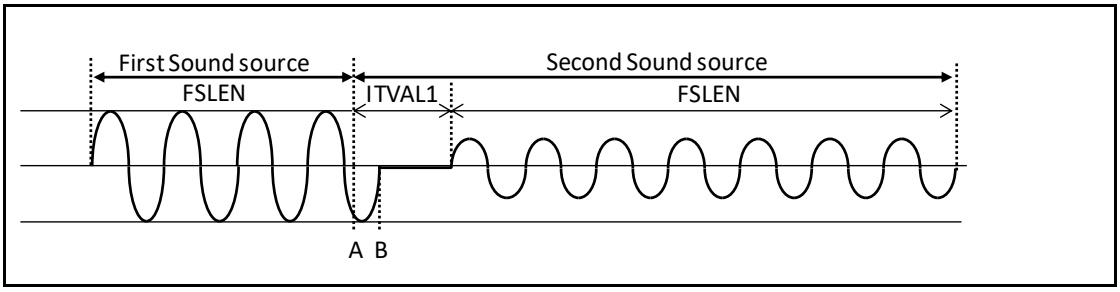
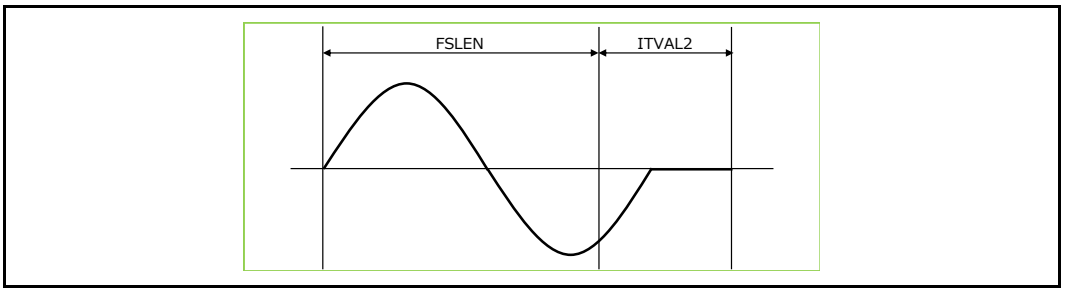
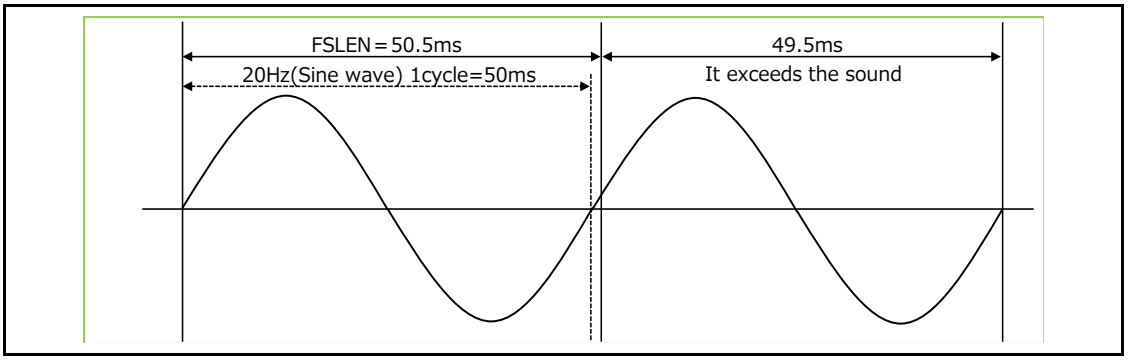


Figure 3-39



"B-A" in Figure 3-38 is error on total sound source time. Maximum number of this error is unit cycle period of first sound source generation. If unit cycle period comes close to total sound source time, the effect of error cannot be ignored. For example, Figure 3-40 shows the case that parameters of sound source generation are "FSFREQ=20Hz" and "FSLEN=50.5ms". In this case, extended time period becomes 49.5ms.

Figure 3-40



Note:

- *Sound source time of the second sound source must be longer than one period of frequency of the first sound source.*
  "One period of 1st sound source defined by FSREQ" < "2nd sound source time (FSLEN)"

3.11. DMA Interface

SWFG has DMA interface to generate sound source without CPU operation.

- DMA interface can be independently enabled for each channel (4.5 WGCHDMAEN-CHnDMAEN n=channel number).
- When parameter setting of sound source generation is available, SWFG requests data transfer to DMA.
- DMA transfers parameters into SWFG and initiates sound source generation by 4.25 WGCHnCTRL4-CHnSTART.
- Registers, which are not frequently updated, are placed on upper side of memory map.
  For example, if parameter updating does not need filter coefficients, DMA can avoid transferring un-necessary filter relevant registers and can limit updating registers to WGCHnCTRL0-4 only.
- Start bit for sound source generation (4.2 WGCHSTART-CHnSTART) can be controlled via 4.25 WGCHnCTRL4-CHnSTART also. In case of DMA operation, DMA must write "1" to 4.25 WGCHnCTRL4-CHnSTART. This bit is in control register. Therefore DMA can do it by DMA transfer operation.

Note:

- *In case of DMA operation, forced termination needs following steps.*
  1. *Request DMA controller to stop DMA operation.*
  2. *Confirm DMA actually stops.*
  3. *Disable DMA interface by 4.5.WGCHDMAEN-CHnDMAEN.*
  4. *Terminate SWFG by forced termination.*

4. Registers

This section describes the registers of Sound Waveform Generator

4.1. SWFG Channel Enable Register (WGCHEN)

REGISTER_NAME	SWFG Channel Enable Register (WGCHEN)
OFFSET	0x000
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.

Always write "0" to this bit. The read value is "0".

[bit4] CH4EN: Channel 4 operation enable/disable

[bit3] CH3EN: Channel 3 operation enable/disable

[bit2] CH2EN: Channel 2 operation enable/disable

[bit1] CH1EN: Channel 1 operation enable/disable

[bit0] CH0EN: Channel 0 operation enable/disable

These bits enable SWFG operation.

CHnEN (n=0 to 4)	Description
0	Stop operation.
1	Enable operation.

Notes:

- *When operation is stopped, SWFG stops sound source generation and output.*
- *There is no sound source output, including ALL 0, from channels that are stopped by this control. The effect on sound source output destinations should be kept firmly in mind when stopping sound source generation.*

4.2. SWFG Channel Start Register (WGCHSTART)

REGISTER_NAME	SWFG Channel Start Register (WGCHSTART)
OFFSET	0x004
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4 START	CH3 START	CH2 START	CH1 START	CH0 START
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] CH4START: Channel 4 sound source generation start

[bit3] CH3START: Channel 3 sound source generation start

[bit2] CH2START: Channel 2 sound source generation start

[bit1] CH1START: Channel 1 sound source generation start

[bit0] CH0START: Channel 0 sound source generation start

These bits initiate sound source generation.

CHnSTART (n=0 to 4)	Description	
	Read	Write
0	Enable sound source generation.	Does not have any effect on operation.
1	Sound source generation start wait.	Start sound source generation.

Notes:

- *When "1" is written to this bit, SWFG starts sound source generation in accordance with sound source specification settings.*
- *SWFG is when "1" is written to this bit while a previous sound source generation is in progress, consecutive sound source generation starts when generation of the previous sound source is complete.*
- *This bit is automatically cleared when the desired sound source generation starts.*
- *Writing "0" in this bit does not stop sound source generation. For information about stopping sound source generation, see 4.1.*
- *Sound source generation start wait cannot be canceled by writing "0" in this bit.*
- *Changing sound source specifications during sound source generation start standby can result in sound source generation starting while implementation of changes is in progress, resulting in unintended sound being produced. Set sound source specifications while sound source generation is enabled.*

4.3. SWFG Channel Busy Register (WGCHBUSY)

REGISTER_NAME	SWFG Channel Busy Register (WGCHBUSY)
OFFSET	0x008
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4BUSY	CH3BUSY	CH2BUSY	CH1BUSY	CH0BUSY
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit4] CH4BUSY: Channel 4 Busy

[bit3] CH3BUSY: Channel 3 Busy

[bit2] CH2BUSY: Channel 2 Busy

[bit1] CH1BUSY: Channel 1 Busy

[bit0] CH0BUSY: Channel 0 Busy

These bits indicate sound source generation is initiated and busy.

CHnBUSY (n=0 to 4)	Description
0	SWFG is not busy
1	SWFG is busy

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

4.4. SWFG Channel Stop Register (WGCHSTOP)

REGISTER_NAME	SWFG Channel Stop Register (WGCHSTOP)
OFFSET	0x00C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4STOP	CH3STOP	CH2STOP	CH1STOP	CH0STOP
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] CH4STOP: Forced termination for channel 4 sound source generation

[bit3] CH3STOP: Forced termination for channel 3 sound source generation

[bit2] CH2STOP: Forced termination for channel 2 sound source generation

[bit1] CH1STOP: Forced termination for channel 1 sound source generation

[bit0] CH0STOP: Forced termination for channel 0 sound source generation

These bits initiate forced termination for each channel.

CHnSTOP (n=0 to 4)	Description	
	Read	Write
0	Forced termination is available	No effects
1	Forced termination is under operating	Initiate forced termination

Notes:

- *CHnSTOP which is written to "1" is automatically cleared to "0" after forced termination is completed.*
- *Forced termination is completed after SWFG outputs silence PCM data. If sound source output is disabled by WGCHEN-CHnEN=0, SWFG cannot output silence PCM data. Therefore forced termination cannot be completed and CHnSTOP keeps "1".*
  If sound source output is enabled at this situation, SWFG outputs silence PCM data and clear CHnSTOP to "0".

4.5. SWFG Channel DMA Enable Register (WGCHDMAEN)

REGISTER_NAME	SWFG Channel DMA Enable Register (WGCHDMAEN)
OFFSET	0x010
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4DMAEN	CH3DMAEN	CH2DMAEN	CH1DMAEN	CH0DMAEN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] CH4DMAEN: Channel 4 Support DMA I/F for setting Enable

[bit3] CH3DMAEN: Channel 3 Support DMA I/F for setting Enable

[bit2] CH2DMAEN: Channel 2 Support DMA I/F for setting Enable

[bit1] CH1DMAEN: Channel 1 Support DMA I/F for setting Enable

[bit0] CH0DMAEN: Channel 0 Support DMA I/F for setting Enable

These bits enable DMA interface.

CHnDMAEN (n=0 to 4)	Description
0	Disable DMA I/F
1	Enable DMA I/F

4.6. SWFG Interrupt Enable Register (WGINTREN)

REGISTER_NAME	SWFG Interrupt Enable Register (WGINTREN)
OFFSET	0x014
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AHBERR
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4END	CH3END	CH2END	CH1END	CH0END
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:17] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit16] AHBERR: AHB MASTER INTERFACE bus error interrupt enable

This bit enables AHB master interface bus error interrupt.

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

[bit15:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] CH4END: Channel 4 sound source generation end interrupt enable

[bit3] CH3END: Channel 3 sound source generation end interrupt enable

[bit2] CH2END: Channel 2 sound source generation end interrupt enable

[bit1] CH1END: Channel 1 sound source generation end interrupt enable

[bit0] CH0END: Channel 0 sound source generation end interrupt enable

These bits enable sound source generation completion interrupt.

CHnEND (n=0 to 4)	Description
0	Disable interrupts.
1	Enable interrupts.

4.7. SWFG Interrupt State Register (WGINTRSTATE)

REGISTER_NAME	SWFG Interrupt State Register (WGINTRSTATE)
OFFSET	0x018
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AHBERR
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4END	CH3END	CH2END	CH1END	CH0END
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:17] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit16] AHBERR: SWFG AHB MASTER INTERFACE bus error interrupts status indication

This bit indicates AHB master interface bus error interrupt status.

Bit	Description
0	No interrupt
1	Interrupt

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

[bit15:5] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit4] CH4END: Channel 4 sound source generation end interrupt status indication

[bit3] CH3END: Channel 3 sound source generation end interrupt status indication

[bit2] CH2END: Channel 2 sound source generation end interrupt status indication

[bit1] CH1END: Channel 1 sound source generation end interrupt status indication

[bit0] CH0END: Channel 0 sound source generation end interrupt status indication

These bits indicate completion interrupt status of sound source generation.

CHnEND (n=0 to 4)	Description
0	No interrupt
1	Interrupt

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

4.8. SWFG Interrupt Clear Register (WGINTRCLR)

REGISTER_NAME	SWFG Interrupt Clear Register (WGINTRCLR)
OFFSET	0x01C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AHBERR
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	CH4END	CH3END	CH2END	CH1END	CH0END
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:17] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit16] AHBERR: SWFG AHB MASTER INTERFACE bus error interrupt clear

This bit clears AHB master interface bus error interrupt.

Bit	Description
0	Does not have any effect on operation
1	Clear interrupt.

Note:

- *If interrupt generation and interrupt clear occur simultaneously, interrupt generation is given priority.*

[bit15:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] CH4END: Channel 4 sound source generation end interrupt clear

[bit3] CH3END: Channel 3 sound source generation end interrupt clear

[bit2] CH2END: Channel 2 sound source generation end interrupt clear

[bit1] CH1END: Channel 1 sound source generation end interrupt clear

[bit0] CH0END: Channel 0 sound source generation end interrupt clear

These bits clear completion interrupt of sound source generation.

CHnEND (n=0 to 4)	Description
0	Does not have any effect on operation.
1	Clear interrupt.

4.9. SWFG AHB Bus Error Register (WGAHBERR)

REGISTER_NAME	SWFG AHB Bus Error Register (WGAHBERR)
OFFSET	0x020
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AHBSERR [1:0]	
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:2] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit1:0] AHBSERR: SWFG AHB Slave Interface access error information indication

These bits indicate AHB slave interface access error information.

Bits	Description
00	There is no error.
01	Address error
10	Write access to Read Only register
11	Access size error

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

4.10. SWFG Channel Address0 Register (WGCHADD0)

REGISTER_NAME	SWFG Channel Address0 Register (WGCHADD0)
OFFSET	0x024
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	CH1ADD[12:8]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CH1ADD[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	1	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	CH0ADD[12:8]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CH0ADD[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:16] CH1ADD[12:0]: Channel 1 output destination address

Sets the Channel 1 output destination address as 13 bits.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:0] CH0ADD[12:0]: Channel 0 output destination address

Sets the Channel 0 output destination address as 13 bits.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

4.11. SWFG Channel Address1 Register (WGCHADD1)

REGISTER_NAME	SWFG Channel Address1 Register (WGCHADD1)
OFFSET	0x028
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	CH3ADD[12:8]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	CH3ADD[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	1	1	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	CH2ADD[12:8]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CH2ADD[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	1	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:16] CH3ADD[12:0]: Channel 3 output destination address

Sets the Channel 3 output destination address as 13 bits.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:0] CH2ADD[12:0]: Channel 2 output destination address

Sets the Channel 2 output destination address as 13 bits.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

4.12. SWFG Channel Address2 Register (WGCHADD2)

REGISTER_NAME	SWFG Channel Address2 Register (WGCHADD2)
OFFSET	0x02C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	CH4ADD[12:8]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	CH4ADD[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	1	0	0	0	0

[bit31:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:0] CH4ADD[12:0]: Channel 4 output destination address

Sets the Channel 4 output destination address as 13 bits.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

4.13. SWFG Channel n Filter Coefficient 0 Register (WGCHnFILCOEF0, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 0 Register (WGCHnFILCOEF0)
OFFSET	n=0:0x080, n=1:0x0C0, n=2:0x100, n=3:0x140, n=4:0x180
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF1[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF1[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF0[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF0[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF1[16:0]: CHn Setting of the filter coefficient 1

This field is filter coefficient 1.

[bit15:0] FILCOEF0[16:0]: CHn Setting of the filter coefficient 0

This field is filter coefficient 0.

4.14. SWFG Channel n Filter Coefficient 1 Register (WGCHnFILCOEF1, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 1 Register (WGCHnFILCOEF1)
OFFSET	n=0:0x084, n=1:0x0C4, n=2:0x104, n=3:0x144, n=4:0x184
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF3[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF3[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF2[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF2[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF3[16:0]: CHn Setting of the filter coefficient 3

This field is filter coefficient 3.

[bit15:0] FILCOEF2[16:0]: CHn Setting of the filter coefficient 2

This field is filter coefficient 2.

4.15. SWFG Channel n Filter Coefficient 2 Register (WGCHnFILCOEF2, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 2 Register (WGCHnFILCOEF2)
OFFSET	n=0:0x088, n=1:0x0C8, n=2:0x108, n=3:0x148, n=4:0x188
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF5[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF5[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF4[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF4[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF5[16:0]: CHn Setting of the filter coefficient 5

This field is filter coefficient 5.

[bit15:0] FILCOEF4[16:0]: CHn Setting of the filter coefficient 4

This field is filter coefficient 4.

4.16. SWFG Channel n Filter Coefficient 3 Register (WGCHnFILCOEF3, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 3 Register (WGCHnFILCOEF3)
OFFSET	n=0:0x08C, n=1:0x0CC, n=2:0x10C, n=3:0x14C, n=4:0x18C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF7[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF7[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF6[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF6[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF7[16:0]: CHn Setting of the filter coefficient 7

This field is filter coefficient 7.

[bit15:0] FILCOEF6[16:0]: CHn Setting of the filter coefficient 6

This field is filter coefficient 6.

4.17. SWFG Channel n Filter Coefficient 4 Register (WGCHnFILCOEF4, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 4 Register (WGCHnFILCOEF4)
OFFSET	n=0:0x090, n=1:0x0D0, n=2:0x110, n=3:0x150, n=4:0x190
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF9[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF9[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF8[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF8[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF9[16:0]: CHn Setting of the filter coefficient 9

This field is filter coefficient 9.

[bit15:0] FILCOEF8[16:0]: CHn Setting of the filter coefficient 8

This field is filter coefficient 8.

4.18. SWFG Channel n Filter Coefficient 5 Register (WGCHnFILCOEF5, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 5 Register (WGCHnFILCOEF5)
OFFSET	n=0:0x094, n=1:0x0D4, n=2:0x114, n=3:0x154, n=4:0x194
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF11[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF11[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF10[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF10[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF11[16:0]: CHn Setting of the filter coefficient 11

This field is filter coefficient 11.

[bit15:0] FILCOEF10[16:0]: CHn Setting of the filter coefficient 10

This field is filter coefficient 10.

4.19. SWFG Channel n Filter Coefficient 6 Register (WGCHnFILCOEF6, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 6 Register (WGCHnFILCOEF6)
OFFSET	n=0:0x098, n=1:0x0D8, n=2:0x118, n=3:0x158, n=4:0x198
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF13[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF13[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF12[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF12[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF12[16:0]: CHn Setting of the filter coefficient 13

This field is filter coefficient 13.

[bit15:0] FILCOEF13[16:0]: CHn Setting of the filter coefficient 12

This field is filter coefficient 12.

4.20. SWFG Channel n Filter Coefficient 7 Register (WGCHnFILCOEF7, n=0 to 4)

REGISTER_NAME	SWFG Filter Coefficient 7 Register (WGCHnFILCOEF7)
OFFSET	n=0:0x09C, n=1:0x0DC, n=2:0x11C, n=3:0x15C, n=4:0x19C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILCOEF15[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FILCOEF15[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FILCOEF14[15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FILCOEF14[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] FILCOEF15[16:0]: CHn Setting of the filter coefficient 15

This field is filter coefficient 15.

[bit15:0] FILCOEF14[16:0]: CHn Setting of the filter coefficient 14

This field is filter coefficient 14.

4.21. SWFG Channel n Control 0 Register (WGCHnCTRL0, n=0 to 4)

REGISTER_NAME	SWFG Channel n Control 0 Register (WGCHnCTRL0)
OFFSET	n=0:0x0A0, n=1:0x0E0, n=2:0x120, n=3:0x160, n=4:0x1A0
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	FILEN	Reserved	Reserved	Reserved	Reserved	STOPREL [2:0]		
ACCESS_TYPE	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	CONTNUM [3:0]			
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PCM ZEROEN
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	ADPHASE EN	ADPHASECH [2:0]		
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] FILEN: FILTER Effective/invalidity Enable

This bit enables filter function.

Bit	Description
0	Filter Invalidity
1	Filter Effective

[bit30:27] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit26:24] STOPREL[2:0]: Stop release Time

This field is RELEASE time on forced termination.

Bit[2:0]	Description
000	0 ms
001	2.5 ms
010	5.0 ms
011	7.5 ms
100	10.0 ms
101	15.0 ms
110	20.0 ms
111	25.0 ms

[bit23:20] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit19:16] CONTNUM[3:0]: Continuous number

This field defines number of times for continuous sound source generation.

Bits[3:0]	Description
0000	Playback continuous sound x1
0001	Playback continuous sound x2
0010	Playback continuous sound x3
0011	Playback continuous sound x4
0100	Playback continuous sound x5
0101	Playback continuous sound x6
0110	Playback continuous sound x7
0111	Playback continuous sound x8
1000	Playback continuous sound x9
1001	Playback continuous sound x10
1010	Playback continuous sound x11
1011	Playback continuous sound x12
1100	Playback continuous sound x13
1101	Playback continuous sound x14
1110	Playback continuous sound x15
1111	Playback continuous sound in infinity

[bit15:9] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit8] PCMZEROEN: PCM ZERO phase enable

This bit enables smooth connection.

Bit	Description
0	Disable Connect Sound Sources with 0 value of PCM
1	Enable Connect Sound Sources with 0 value of PCM

[bit7:4] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit3] ADPHASEEN: Adjust phase enable

This bit enables phase synchronizer.

Bit	Description
0	Disable Additional function to adjust phase among channels
1	Enable Additional function to adjust phase among channels

[bit2:0] ADPHASECH[2:0]: Adjust phase channel

This field defines target channel of phase synchronizer.

Bits[2:0]	Description
000	Channel 0
001	Channel 1
010	Channel 2
011	Channel 3
100	Channel 4
101～111	Channel 4

4.22. SWFG Channel n Control 1 Register (WGCHnCTRL1, n=0 to 4)

REGISTER_NAME	SWFG Channel n Control 1 Register (WGCHnCTRL1)
OFFSET	n=0:0x0A4, n=1:0x0E4, n=2:0x124, n=3:0x164, n=4:0x1A4
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	DBEN	Reserved	GAINUD SEL	GAINFLUCT [16:12]				
ACCESS_TYPE	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	GAINFLUCT [11:4]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	GAINFLUCT [3:0]				Reserved	GAIN SUCEN	INITGAIN [9:8]	
ACCESS_TYPE	R/W	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	INITGAIN [7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] DBEN: Decibel enable

This bit defines unit of following registers.

WGCHnCTRL1-GAINFLUCT[16:0], WGCHnCTRL1-INITGAIN[7:0], WGCHnSTATUS-ENDGAIN[9:0]

Bit	Description
0	Magnification
1	Decibel

[bit30] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit29] GAINUDSEL: Gain up/down selector

This bit selects gain increase/decrease direction.

Bit	Description
0	Gain increase
1	Gain decrease

[bit28:12] GAINFLUCT[16:0]: Gain fluctuation

This field defines gain fluctuation value.

Unit depends on setting of DBEN bit. It is ratio in case of DBEN=0 and dB in case of DBEN=1.

Bits[16:0] (in case of DBEN=0)	Description
	The following 10ms gain
0x00000	0.00000
0x00001	0.00001
0x00002	0.00002
0x00003	0.00003
0x00004	0.00004
…	…
0x1869C	0.99996
0x1869D	0.99997
0x1869E	0.99998
0x1869F	0.99999
0x186A0 - 0x1FFFF	1.00000
0x1869C	0.99996

Bits[16:0] (in case of DBEN=1)	Description
	The following 10ms gain
0x00000	0.000 dB
0x00001	0.001 dB
0x00002	0.002 dB
0x00003	0.003 dB
0x00004	0.004 dB
…	...
0x176FC	95.996 dB
0x176FD	95.997 dB
0x176FE	95.998 dB
0x176FF	95.999 dB
0x17700 - 0x1FFFF	96.000 dB

[bit11] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit10] GAINSUCEN: Gain transfer enable

This bit enables gain transfer.

Bit	Description
0	Disable gain transfer
1	Enable gain transfer

[bit9:0] INITGAIN[9:0]: Initial gain

This field selects initial gain.

Unit depends on setting of DBEN bit. It is ratio in case of DBEN=0 and dB in case of DBEN=1.

Bits[9:0] (in case of DBEN=0)	Description
0x000	1.000
0x001	0.999
0x002	0.998
0x003	0.997
0x004	0.996
0x005	0.995
...	...
0x3E3	0.005
0x3E4	0.004
0x3E5	0.003
0x3E6	0.002
0x3E7	0.001
0x3E8 – 0x3FF	0.000

Bits[9:0] (in case of DBEN=1)	Description
0x000	0 dB
0x001	-0.1 dB
0x002	-0.2 dB
0x003	-0.3 dB
0x004	-0.4 dB
0x005	-0.5 dB
...	...
0x3BB	-95.5 dB
0x3BC	-95.6 dB
0x3BD	-95.7 dB
0x3BE	-95.8 dB
0x3BF	-95.9 dB
0x3C0 – 0x3FF	-96.0 dB

4.23. SWFG Channel n Control 2 Register (WGCHnCTRL2, n=0 to 4)

REGISTER_NAME	SWFG Channel n Control 2 Register (WGCHnCTRL2)
OFFSET	n=0:0x0A8, n=1:0x0E8, n=2:0x128, n=3:0x168, n=4:0x1A8
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RELCV SEL	Reserved	Reserved	RELEASE [4:0]				
ACCESS_TYPE	R/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	ATTCV SEL	Reserved	Reserved	ATTACK [4:0]				
ACCESS_TYPE	R/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	DUTY[6:0]						
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	FSINF[1:0]		Reserved	Reserved	MONO[1:0]	
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R0,W0	R0,W0	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] RELCVSEL: Release curve selector

This bit selects waveform for RELEASE function.

Bit	Description
0	Linear
1	Exponential

[bit30:29] Reserved

This bit is reserved.

Always write "0" to this bit. The read value is "0".

[bit28:24] RELEASE[4:0]: Release Time

This field selects RELEASE time.

Bits[4:0]	Description
00000	release control not used.
00001	1 ms
00010	2 ms
00011	3 ms
00100	4 ms
00101	5 ms
00110	7.5 ms
00111	10 ms
01000	15 ms
01001	20 ms
01010	25 ms
01011	30 ms
01100	40 ms
01101	50 ms
01110	75 ms
01111	100 ms
10000	125 ms
10001	150 ms
10010	200 ms
10011	250 ms
10100	300 ms
10101	400 ms
10110	500 ms
10111	750 ms
11000 to 11111	1000 ms

[bit23] ATTCVSEL: Attack curve selector

This bit selects waveform for ATTACK function.

Bit	Description
0	Linear
1	Exponential

[bit22:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] ATACK[4:0]: Attack Time

This field selects RELEASE time.

Bits[4:0]	Description
00000	attack control not used.
00001	1 ms
00010	2 ms
00011	3 ms
00100	4 ms
00101	5 ms
00110	7.5 ms
00111	10 ms
01000	15 ms
01001	20 ms
01010	25 ms
01011	30 ms
01100	40 ms
01101	50 ms
01110	75 ms
01111	100 ms
10000	125 ms
10001	150 ms
10010	200 ms
10011	250 ms
10100	300 ms
10101	400 ms
10110	500 ms
10111	750 ms
11000 to 11111	1000 ms

[bit15] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit14:8] DUTY[6:0]: DUTY setting

This field selects DUTY ratio.
This setting is valid in case of triangle and square waveforms.
This setting is ignored in case of sine waveform.

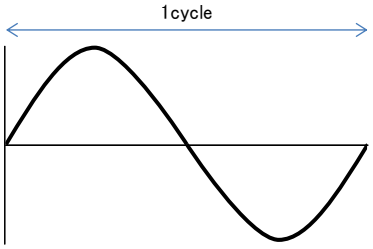
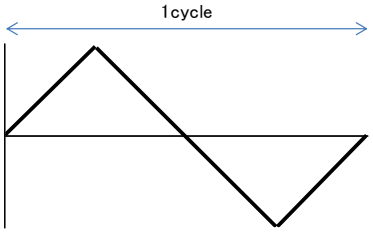
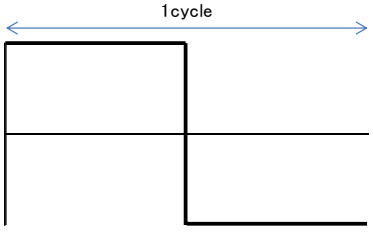
Bits[6:0]	Description
0000000	0 %
0000001	1 %
…	…
0110010	50 %
…	…
1100011	99 %
1100100 to 1111111	100 %

[bit7:6] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit5:4] FSINF[1:0]: Type of generation sound waveform

This field selects waveform.

Bits[1:0]	Description	
00	Sign waveform	
01	Triangle waveform	
10 / 11	Square waveform	

[bit3:2] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit1:0] MONO[1:0]: Channel output data pattern

This field selects output PCM format.

Bits[1:0]	Description
00	Both of upper and lower 16bits field are filled by same 16bits PCM data (XXXX). Result is "XXXX_XXXX".
01	Lower 16bits are filled by 16bits PCM data (YYYY) and upper 16bits are filled by "0". Result is "0000_YYYY".
10	Upper 16bits are filled by 16bits PCM data (ZZZZ) and lower 16bits are filled by "0". Result is "ZZZZ_0000".
11	Setting prohibited

4.24. SWFG Channel n Control 3 Register (WGCHnCTRL3, n=0 to 4)

REGISTER_NAME	SWFG Channel n Control 3 Register (WGCHnCTRL3)
OFFSET	n=0:0x0AC, n=1:0x0EC, n=2:0x12C, n=3:0x16C, n=4:0x1AC
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	ITVAL2 [13:8]					
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	ITVAL2 [7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	ITVAL1 [13:8]					
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	ITVAL1 [7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:30] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit29:16] ITVAL2[13:0]: Posterior mute time

This field selects posterior mute time.

Bits[13:0]	Description
0x0000	0.0 ms
0x0001	0.5 ms
0x0002	1.0 ms
0x0003	1.5 ms
0x0004	2.0 ms
0x0005	2.5 ms
0x0006	3.0 ms
…	…
0x1000	2,048.0 ms
0x1001	2,048.5 ms
0x1002	2,049.0 ms
0x1003	2,049.5 ms
0x1004	2,050.0 ms
0x1005	2,050.5 ms
0x1006	2,051.0 ms
…	…
0x2000	4,096.0 ms
0x2001	4,096.5 ms
0x2002	4,097.0 ms
0x2003	4,097.5 ms
0x2004	4,098.0 ms
0x2005	4,098.5 ms
0x2006	4,099.0 ms
…	…
0x3FF9	8,188.5 ms
0x3FFA	8,189.0 ms
0x3FFB	8,189.5 ms
0x3FFC	8,190.0 ms
0x3FFD	8,190.5 ms
0x3FFE	8,191.0 ms
0x3FFF	8,191.5 ms

[bit15:14] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit13:0] ITVAL1[13:0]: Anterior mute time

This field selects anterior mute time.

Bits[13:0]	Description
0x0000	0.0 ms
0x0001	0.5 ms
0x0002	1.0 ms
0x0003	1.5 ms
0x0004	2.0 ms
0x0005	2.5 ms
0x0006	3.0 ms
…	…
0x1000	2,048.0 ms
0x1001	2,048.5 ms
0x1002	2,049.0 ms
0x1003	2,049.5 ms
0x1004	2,050.0 ms
0x1005	2,050.5 ms
0x1006	2,051.0 ms
…	…
0x2000	4,096.0 ms
0x2001	4,096.5 ms
0x2002	4,097.0 ms
0x2003	4,097.5 ms
0x2004	4,098.0 ms
0x2005	4,098.5 ms
0x2006	4,099.0 ms
…	…
0x3FF9	8,188.5 ms
0x3FFA	8,189.0 ms
0x3FFB	8,189.5 ms
0x3FFC	8,190.0 ms
0x3FFD	8,190.5 ms
0x3FFE	8,191.0 ms
0x3FFF	8,191.5 ms

4.25. SWFG Channel n Control 4 Register (WGCHnCTRL4, n=0 to 4)

REGISTER_NAME	SWFG Channel n Control 4 Register (WGCHnCTRL4)
OFFSET	n=0:0x0B0, n=1:0x0F0, n=2:0x130, n=3:0x170, n=4:0x1B0
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CHn START	Reserved	FSLEN [13:8]					
ACCESS_TYPE	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	FSLEN [7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FSFREQ [15:8]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FSFREQ [7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] CHnSTART: Channel n start mirror bit

Mirror of Start Bit set to WGCHSTART.

[bit30] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit29:16] FSLEN[13:0]: Run time

This field selects run time.
Available run time is from 0.5 ms to 8191.5 ms.

Bits[13:0]	Description
0x0000	0.5 ms
0x0001	0.5 ms
0x0002	1.0 ms
0x0003	1.5 ms
0x0004	2.0 ms
…	…
0x1000	2048.0 ms
0x1001	2048.5 ms
0x1002	2049.0 ms
0x1003	2049.5 ms
0x1004	2050.0 ms
…	…
0x2000	4096.0 ms
0x2001	4096.5 ms
0x2002	4097.0 ms
0x2003	4097.5 ms
0x2004	4098.0 ms
…	…
0x3FF7	8187.5 ms
0x3FF8	8188.0 ms
0x3FF9	8188.5 ms
0x3FFA	8189.0 ms
0x3FFB	8189.5 ms
0x3FFC	8190.0 ms
0x3FFD	8190.5 ms
0x3FFE	8191.0 ms
0x3FFF	8191.5 ms

[bit15:0] FSFREQ[15:0]: Frequency setting

This field selects frequency of sound source.

Bits[15:0]	Description
0x0000 – 0x0028	20.0 Hz
0x0029	20.5 Hz
0x002A	21.0 Hz
0x002B	21.5 Hz
0x002C	22.0 Hz
…	…
0x9C3A	19997.0 Hz
0x9C3B	19997.5 Hz
0x9C3C	19998.0 Hz
0x9C3D	19998.5 Hz
0x9C3E	19999.0 Hz
0x9C3F	19999.5 Hz
0x9C40 – 0xFFFF	20000.0 Hz

4.26. SWFG Channel n Status Register (WGCHnSTATUS, n=0 to 4)

REGISTER_NAME	SWFG Channel n Status Register (WGCHnSTATUS)
OFFSET	n=0:0x0B4, n=1:0x0F4, n=2:0x134, n=3:0x174, n=4:0x1B4
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	CONTNUMR[3:0]			
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	END GAINV	Reserved	Reserved	Reserved	Reserved	Reserved	ENDGAIN [9:8]	
ACCESS_TYPE	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	ENDGAIN [7:0]							
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:20] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit19:16] CONTNUMR[3:0]: Continuous Remaining Number

Number of remainder counts of CONTNUM of WGCHnCTRL1.

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

[bit15] ENDGAINV: End Gain value Status

This bit shows validity of ENDGAIN bits on this register.

Bit	Description
0	ENDGAIN[9:0] is invalid.
1	ENDGAIN[9:0] is valid.

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

[bit14:10] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a SWFG AHB Slave interface access error.

[bit9:0] ENDGAIN[9:0]: End gain at completion of sound source generation

This field shows final gain value at completion of sound source generation.
Unit depends on setting of DBEN bit of WGCHnCTRL1. It is ratio in case of DBEN=0 and dB in case of DBEN=1.

Bits[9:0] (in case of DBEN=0)	Description
0x000	1.000
0x001	0.999
0x002	0.998
0x003	0.997
0x004	0.996
0x005	0.995
…	…
0x3E3	0.005
0x3E4	0.004
0x3E5	0.003
0x3E6	0.002
0x3E7	0.001
0x3E8	0.000
0x3E9 – 0x3FF	Undefined

Bits[9:0] (in case of DBEN=1)	Description
0x000	0 dB
0x001	-0.1 dB
0x002	-0.2 dB
0x003	-0.3 dB
0x004	-0.4 dB
0x005	-0.5 dB
…	…
0x3BB	-95.5 dB
0x3BC	-95.6 dB
0x3BD	-95.7 dB
0x3BE	-95.8 dB
0x3BF	-95.9 dB
0x3C0	-96.0 dB
0x3C1 - 0x3FF	Undefined

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a SWFG AHB Slave Interface access error.*

5. Appendix

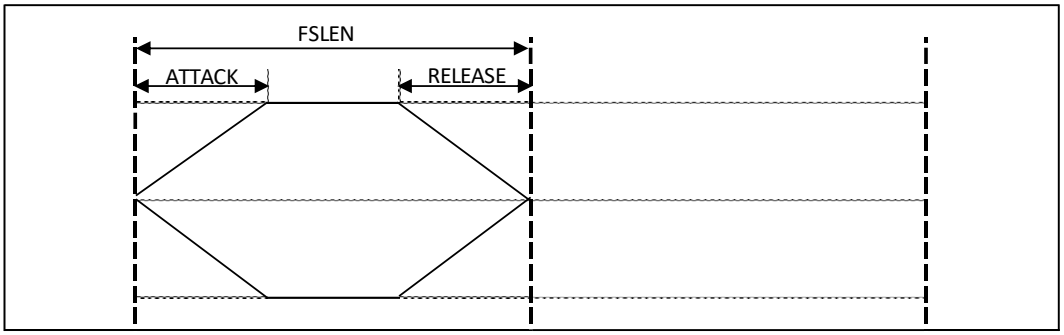
5.1. Example of sound source generation

This chapter shows examples of combinational operation (ATTACK/ RELEASE/ FSLEN).

Assumptions are "Gain transfer is not enabled", "Initial gain is 1 (ratio)", and "Gain increase/decrease is not enabled".

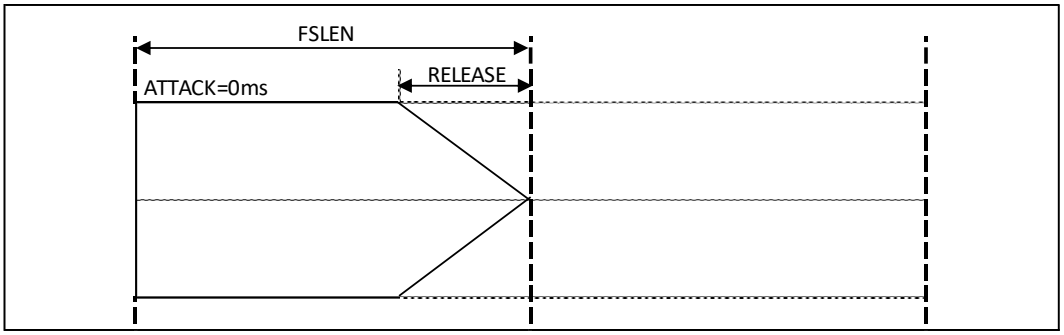
5.1.1. FSLEN > ATTACK + RELEASE

Figure 5-1



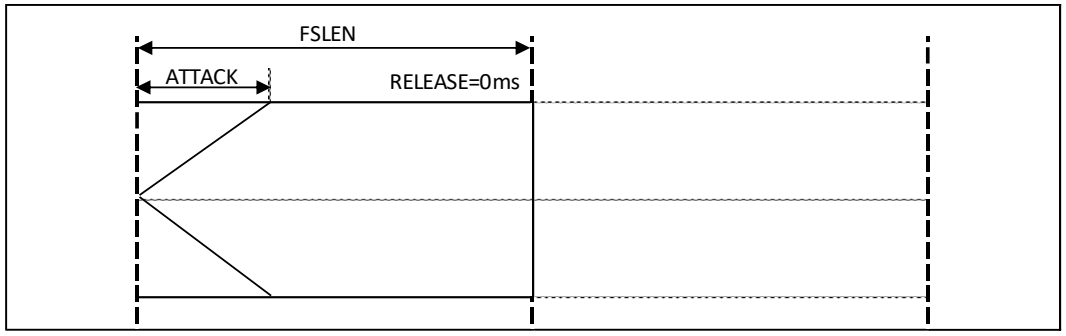
5.1.2. FSLEN > RELEASE (ATTACK=0ms)

Figure 5-2



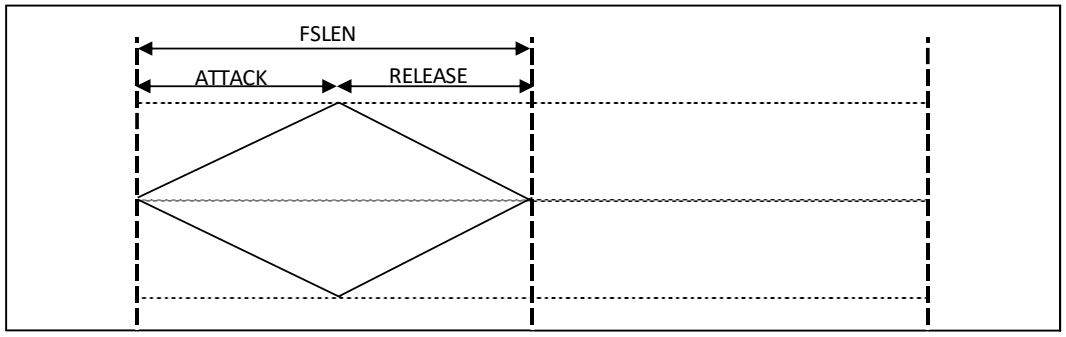
5.1.3. FSLEN > ATTACK (RELEASE=0ms)

Figure 5-3



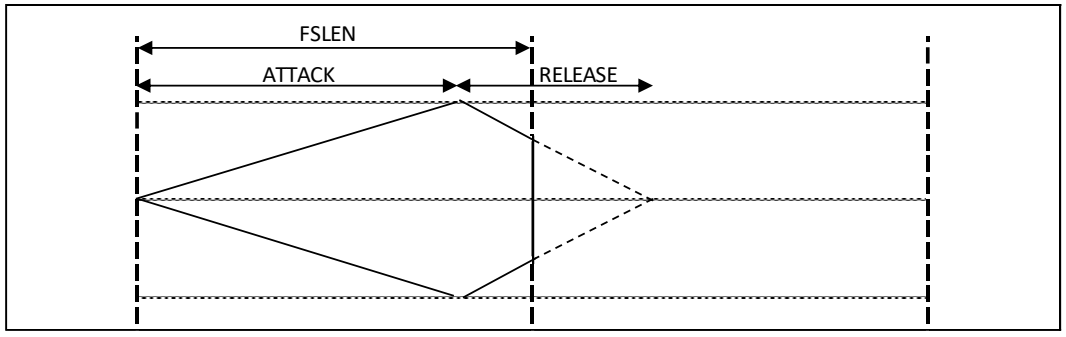
5.1.4. FSLEN = ATTACK + RELEASE

Figure 5-4



5.1.5. FSLEN < ATTACK + RELEASE (FSLEN > ATTACK)

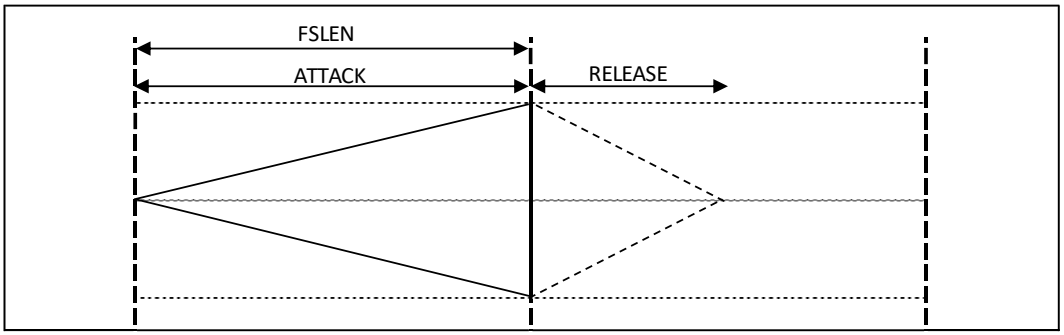
Figure 5-5



5.1.6. FSLEN < ATTACK + RELEASE (FSLEN = ATTACK)

RELEASE setting is "Don't care" in case of "FSLEN = ATTACK".

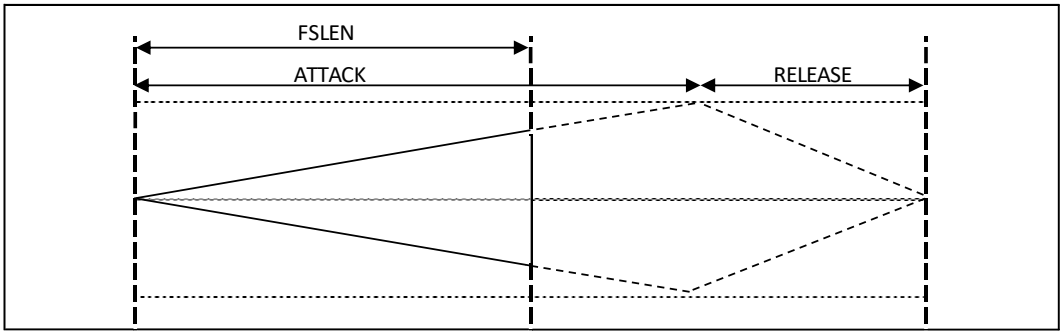
Figure 5-6



5.1.7. FSLEN < ATTACK + RELEASE (FSLEN < ATTACK)

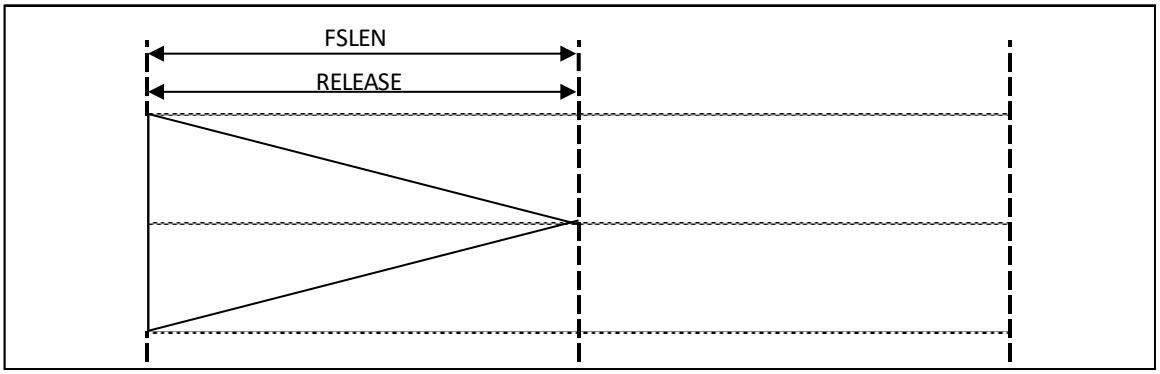
RELEASE setting is "Don't care" in case of "FSLEN < ATTACK".

Figure 5-7



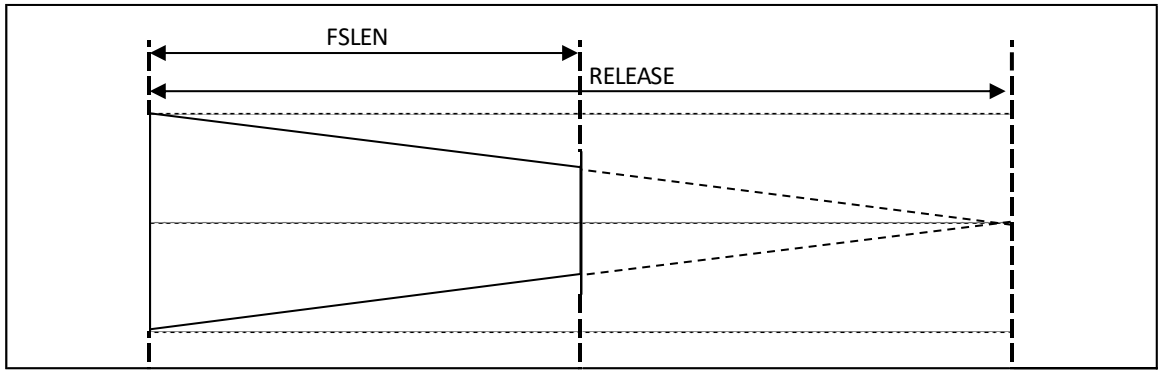
5.1.8. FSLEN = RELEASE (ATTACK=0ms)

Figure 5-8



5.1.9. FSLEN < RELEASE (ATTACK=0ms)

Figure 5-9



CHAPTER 31: Sound Mixer

This chapter explains the sound mixer.

1. Overview
2. Configuration and Block Diagram
3. Operation of the Sound Mixer
4. Registers
5. Appendix

CODE: SOUNDMIX-E3.16-0

1. Overview

This section gives a brief overview of the sound mixer.

Also see the chapter of "Sound System configuration" how to configure the sound mixer.

The sound mixer provides functions for mixing multiple input PCM (pulse code modulation) sound sources.

Table 1-1

Feature	Description
Input sound source	16-bit or 32-bit PCM sound source
Output sound source	32-bit PCM sound source
Number of input channels	Up to 10 channels of sound source input are supported.
Number of output channels	1 ch
Input sampling rate	Mixing of sound sources with different sampling rates is supported. − WFG0 to 4 (five channels total): Input channels for fixed sampling rates: 48 kHz − PMIS0 to 4 (five channels total): Input channels for selectable sampling rates: 4, 8, 12, 24, 48, 96, and 44.1 kHz
Output sampling rate	48 kHz
Mixing System	Saturation calculation operation
FIR filter	Prevention of sound quality change by sampling rate conversion
Volume effects	The volume and the mute, fade in, and fade out effects of the input sound source and output sound source are controllable.

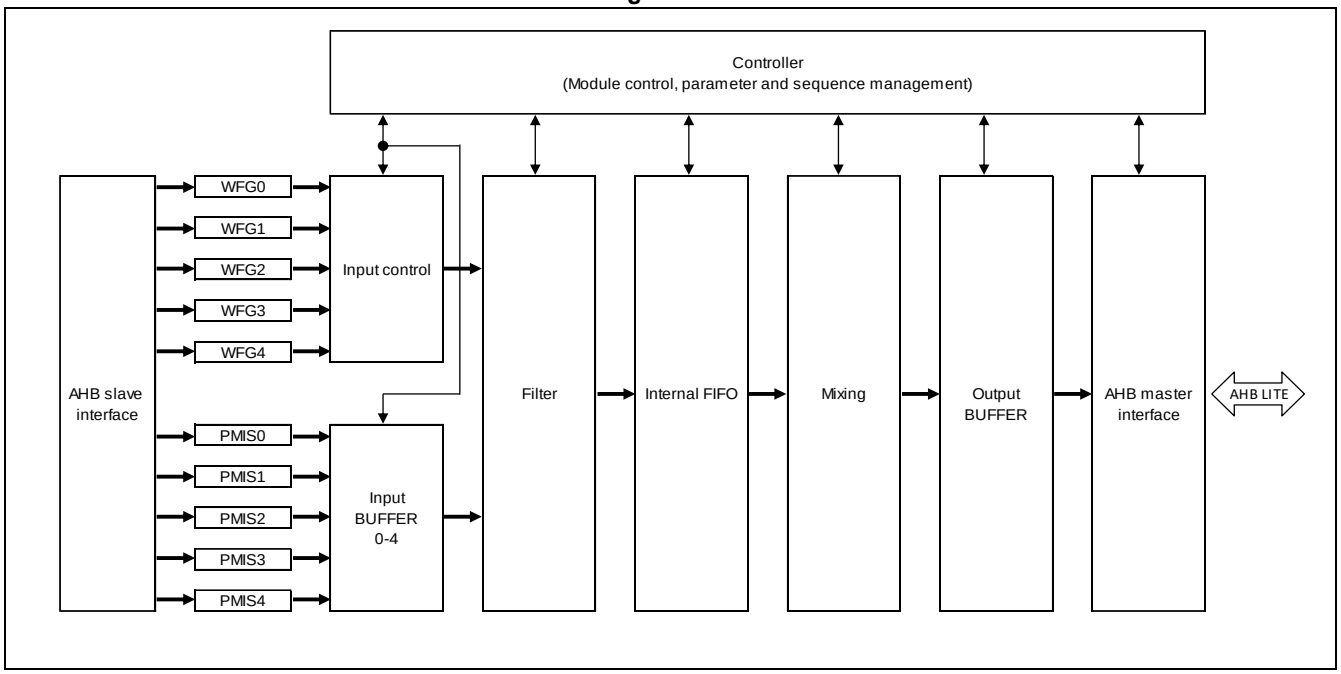
Notes:

- *The input sound source is input in 32-bit units, regardless of the sound source format. A monaural sound source is input in 16-bit and 32-bit units.*
- *16-bit monaural sound source output is not supported.*
- *Sound waveform generator output may have a fixed connection to WFG0 to 4. It should be noted that there may be specification limits on specific models.*
- *The 44.1 kHz input sampling rate can be applied to only one of the input channels PMIS0 to 4. Operation when 44.1 kHz is applied to multiple channels cannot be guaranteed from the viewpoint of the mixing process load imposed by the sound mixer. It should be noted that there may limits on specific models concerning the selection of other frequencies.*
- *In the case of a sound source with a sampling rate that is not 48 kHz or 96 kHz, the sampling rate is converted internally by the mixer to 48 kHz. Noise removal by an FIR noise filter is performed before mixing to reduce change in sound quality due to distortion generated by this conversion. However, noise removal by FIR filtering is not performed in the case of down sampling of a 96 kHz sound source to 48 kHz. Because of this, there is the risk of sound quality being affected by aliasing generated at that time. The sampling rate must be determined with sufficient sound quality evaluation being implemented by the system.*
- *Mixed sound sources are calculated by the sound mixer's signed saturation calculation. Saturation calculation uses a pre-defined minimum value or maximum value when a calculation operation generates overflow. Mixing multiple high volume (large amplitude) sound sources creates the risk of sound quality being affected by saturation calculation rounding error. Sound mixing supports individual volume control for each channel, so adjustments can be made while evaluating the result on the system.*

2. Configuration and Block Diagram

This section shows a block diagram of the sound mixer.

Figure 2-1



Notes:

- *The mixing section is equipped with functions for use by volume effects (volume control, mute, fade in/fade out effects). Volume of each input channel and volume effects of output are applied before and after saturation calculation operations within each mixing operation.*
- *The AHB master outputs the mixed sound source to the address specified as the output destination.*
- *Every channel of WFG0 to 4 has an input buffer for a word (32bit) in the part of "input control".*

3. Operation of the Sound Mixer

3.1. Basic Mixing Operation Procedure

3.1.1. Mixing Start

The mixing procedure started by sound mixing is shown below.

1. Disable input of the sound source. See MXCH under 4.1.

2. Set the output destination address and number of transfers of the mix sound source. See MXOCTRL under 4.2.

3. Configure sound source input channel PMIS0 through 4 sampling rate settings. See MXICTRL under 4.4.

4. Configure sound source input channel PMIS0 through 4 sound source process mode settings. See MXCHMONO under 4.5.

5. Initialize the sound source input channel PMIS0 through 4 input buffers. Also, initialize the mixed sound source output buffer. See MXBUFFCLR under 4.18.

6. Enable sound source input. See MXCH under 4.1.

7. Configure DMA transfer request and data transfer request interrupt threshold value settings for sound input channels PMIS0 through 4. See MXDRQCTRL under 4.3.

8. Configure interrupt enable/disable settings. See MXINTREN under 4.20.

9. Start sound source input.

Notes:

- *The settings in steps 3, 4 and 7 are not required for sound source input channels WFG0 through 4. Also, other settings equivalent to these settings are not configured.*
- *Configure the input channels with the above settings in sequence from 1 through 9 in order to add new sound source input to an ongoing sound mixer operation and perform consecutive mixing. However, the output destination setting in step 2 is not required.*
- *The number of transfers of mixed sound sources from the sound mixer must match the number of output destination transfer requests. When the number of transfers on the sound mixer side is greater, the output destination will generate a DMA transfer error. When the number of transfer requests on the output destination side is greater, the output destination will not issue a DMA transfer request.*
- *If a sound source is not input on the actual device a sound source input is enabled, mixing is stopped for all channels (not only on the applicable channel).*
- *In the step 5, completion of the initialization is required before executing the next step. Completion can be confirmed by reading the applicable bit of the MXBUFFCLR register, because the bit is automatically cleared to 0 following initialization.*

3.1.2. Mixing Stop

Stopping mixing stops input of all channels. Removing a specific channel from mixing stops input of that channel. See MXCH under 4.1.

3.1.3. Fade In

- This setting sets the initial fade in time. See MXCHFADE1 to 5 under 4.10 to 4.14.
- Next, initialize the fade state. See MXFADECLR under 4.19.
- After that, enable fade in. See MXCHFADEEN under 4.16.

Notes:

- *The fade in effect assumes that settings are configured before mixing and applied simultaneously with the start of mixing. However, they can also be applied while the mixing operation is in progress. In that case, the fade in process starts immediately.*
- *Be sure to initialize the fade state before fade in. Fade in following initialization will perform fade in control of the fade gain from -96 dB to 0 dB. However, this initialization operation is not necessarily required for fade in following fade out. This is because fade out causes gain to become -96 dB.*

3.1.4. Fade Out

Fade out can be applied only after fade in.

First, set the fade out time. See MXCHFADE1 to 5 under 4.10 to 4.14. After that, enable fade out. See MXCHFADEEN under 4.16.

Notes:

- *The fade out effect assumes that settings are configured before the completion of mixing and applied simultaneously with the completion of mixing. However, they can also be applied while the mixing operation is in progress. In that case, the fade out process starts immediately.*
- *Be sure to initialize the fade state before fade out. Fade out following initialization will perform fade out control of the fade gain from 0 dB to -96 dB. However, this initialization operation is not necessarily required for fade out following fade in. This is because fade in causes gain to become 0 dB.*

3.2. Volume Effects

There are three volume effects: volume control, mute, and fade in/fade out.

These effects can be applied individually to sound sources being input/output.

In the case of an input sound source in particular, volume effect settings can be configured for each individual input channel (WFG0 to 4 and PMIS0 to 4).

Table 3-1

Volume Effects	Description
Volume control	Volume is controlled in accordance with the gain set by software.
Mute	Mute outputs ALL 0 or ALL 1, regardless of the input/output sound source set software.
Fade in/fade out effects	Phased fade in and fade out are automatically controlled over the time set by software.

Note:

- *Even if the minimum gain or mute is applied to the sound source, the mixer does not always output ALL 0 data. In that case, positive data become ALL 0, but negative data become ALL 1 (-1 in decimal). Due to this, slight sound may be heard in accordance with your output environment.*

3.2.1. Volume Control

Volume of the input/output sound source is controlled in accordance with the volume control gain set by software. This setting can be configured while operation is in progress, and settings are immediately reflected in operation.

For information about register settings, see 4.6, 4.7 and 4.8. For details about gain for volume control, see 5.1.

3.2.2. Mute

Outputs ALL 0 or ALL 1, regardless of the input/output sound source set software. This setting can be configured while operation is in progress, and settings are immediately reflected in operation.

For information about register settings, see 4.9.

3.2.3. Fade In/Fade Out Effects

Seamless fade in and fade out are controlled over the time set by software. The hardware internal fade in/out gain is changed automatically.

For information about the register setting procedure, see 3.1.3 and 3.1.4. For details about fade in and fade out time settings for volume control, see 5.2.

Notes:

- *Volume control gain and fade in/gain are independent of each other. The product of both gains is applied as the overall volume effect.*
- *The fade in/out gain is 0 dB for both the fade in ending point and the fade out starting point. The gain of the overall volume effect is determined in accordance with the volume control gain set by software.*

3.3. Sound Source Input and Internal Mixing Process Mode

Input the input sound source to PMIS0 to 4 in 32-bit units, regardless of the sound source format. Also input using 32-bit units in the case of 16-bit monaural sound source input.

The sound source will be mixed internally by the mixer as a 32-bit stereo sound source. In other words, the upper 16 of the 32 bits is considered to be the left sound source, with the lower 16 bits are considered to be the right sound source, and this is mixed with the upper 16 bits and lower 16 bits of sound sources that are input from other channels.

For input of 16-bit sound sources, the n-th sound source is input as the upper 16 bits and the n+1 sound source is input as the lower 16 bits for 32-bit unit input. Note that n is a positive integer.

A processing mode can be selected from among those shown in Table 3-2 for monaural sound source internal processing.

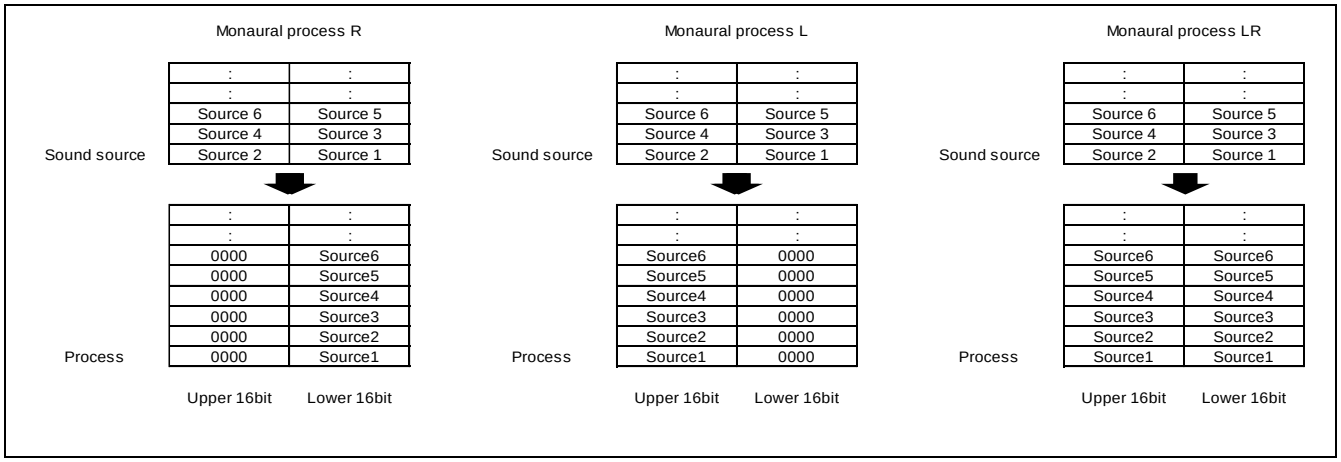
For information about configuring settings by software, see MXCHMONO under 4.5.

Table 3-2

Processing Mode	Application	Internal Process Overview
Stereo processing	Stereo processing	Mixes input 32-bit stereo sound source.
Monaural processing R	Right-side mixing	Mixes by converting to 32-bit data (Upper 16 bits = ALL 0, Lower 16 bits = Monaural sound source).
Monaural processing L	Left-side mixing	Mixes by converting to 32-bit data ('Upper 16 bits = Monaural sound source, Lower 16 bits = ALL 0).
Monaural processing LR	Both side mixing (Left=Right)	Mixes by converting to 32-bit data (Upper 16 bits = Lower 16 bits = Monaural sound source).

An example of monaural processing is shown in Figure 3-1.

Figure 3-1



Notes:

- *A change in processing mode made while sound source is transferred to the sound source input channels PMIS0 through PMIS4 is prohibited.*
- *Monaural processing cannot be used on a 32-bit stereo sound source. Attempting to do so will result in improper sound.*

3.4. Interrupt

The sound mixer supports the interrupts described below.

- Data transfer request interrupt
  This interrupt is generated when a state is detected in which free space in the sound mixer PMIS0 to 4 input buffer 0 to 4 is greater than the FEST value, set in the MXDRQCTRL register, + 1. The interrupt is generated when the value is equal to or greater than the threshold value set by software. The sound mixer interrupt enable setting, interrupt clear control and status indication.
- Input FIFO overflow interrupt
  This interrupt is generated when an overflow is detected in the sound mixer PMIS0 to 4 input buffer 0 to 4. The sound mixer interrupt enable setting, interrupt clear control and status indication.
- DMA transfer error
  This interrupt is generated when a state is detected in which data input to the PMIS0 to 4 input buffer 0 to 4 is greater than the FEST value, set in the MXDRQCTRL register, + 1. The sound mixer interrupt enable setting, interrupt clear control and status indication.
- AHB Master Interface bus error interrupt
  This interrupt is generated when an error response is received during sound source output to an output destination on the AHB Master Interface of the sound mixer. The sound mixer interrupt enable setting, interrupt clear control and status indication.

3.5. Data Request Control

Mixer macros generate two types of data requests for WFG (WFGDATAREQ[4:0]) and for DMAC (MX_DMA_REQ[4:0]).

Data request protocol specifications

• Request assert conditions are described below (1 and 2).

• When ACK is received in response to a request, the request is negated.

• Data transfer is performed following request transfer.

• The data transfer volume is counted internally, and the sequence of transfer protocols is ended after the requested volume of data is complete.

(The next sequential request cannot be asserted until the sequence of transfer protocols is complete.)

1. Data request for WFG
   There is a WFGDATAREQ[4:0] (for WFG) for each channel. WFGDATAREQ[0] corresponds to WFG Channel 0 while WFGDATAREQ[4] corresponds to WFG Channel 4. WFG is notified by a level signal when data receive is possible. HIGH level indicates that receive is possible.

2. Data request for DMAC
   Data request for DMAC (MX_DMA_REQ[4:0]) is a data request to DMAC that asserts the REQ signal of the applicable input channel when there is [threshold value + 1] of space in the input buffer. MX_DMA_REQ[0] is for PMIS Channel 0, while MX_DMA_REQ[4] is for PMIS Channel 4.

Note:

- *If the DMAC is used for the sound source transmission, the corresponding DMAEN bit of MXDRQCTRL is set to 0 after the DMA-transmission is completed.*

3.6. Data Output Control

Subordinate macro data transfer settings are regarded as DMAC INF.

A DMA request is received from a subordinate macro (DAC, PCMPWM, I2S), and [DATATN setting value + 1] data items are transferred to the macro specified by the MACRO field of the MXOCTRL register.

3.6.1. Function Details (Data Protocol Specifications)

1. Receive DMA request.
2. When [MXOCTRL register DATATN setting value + 1] or more of data is accumulated in the output buffer, send an ACK signal.
3. Following ACK transfer, send data. After data send is complete, end sequence of transfer protocols.
   (The next request cannot be accepted until the sequence of transfer protocols is complete.)

3.7. AHB Slave Interface

Mixer macros provide two AHB slave interface groups: for WFG interfacing and for CPU interfacing. AHB Slave interface specifications are shown in Table 3-3 and Table 3-4.

3.7.1. AHB Slave Interface (CPU/DMA)

Table 3-3 AHB Slave Interface (CPU/DMA)

Item	Descriptions
Burst transfer	Mixer macros support burst transfer. However, all burst transfer is processed as a single transfer.
Protection control	not support
Response	OKAY and ERROR only
Access size	8, 16, 32bit only
Width of address	13bit

Notes:

- *The following accesses cause the slave error response.*
- *1) When the access destination is an address that does not exist in the memory map*
- *2) The write access was done to the read only register.*
- *3) When the access size is not supported.*
- *(For example it accessed MXPMISnDADR with 8bit or 16bit) n=0 to 4*

3.7.2. WFG Dedicated AHB Slave Interface

Table 3-4 WFG AHB Slave Interface

Item	Descriptions
Burst transfer	not support
Protection control	not support
Response	OKAY and ERROR only
Access size	8, 16, 32bit only
Width of address	10bit

Notes:

- *The following accesses cause the slave error response.*
- *1) When the access destination is other than MXWFGnDADR.*
- *2) When the access size is not supported.*
- *(For example it accessed MXWFGnDADR with 8bit or 16bit) n=0 to 4*

3.8. AHB Master Interface

Mixer macros provide an AHB master interface for interfacing with subordinate macros. AHB Master interface specifications are shown in Table 3-5.

Table 3-5 AHB Master Interface

Item	Descriptions
Burst transfer	not support
Protection control	not support
Response	OKAY and ERROR only
Access size	8, 16, 32bit only
Width of address	13bit

Notes:

- *When an ERROR response is received by transfer control from the CPU interface, an error response is transferred to the CPU interface.*
- *Information about the operation when an error response is received during mixer macro self-operation is stored in the information register and interrupt occurs.*

3.9. Input Buffer (FIFO)

Mixer macros provide a 5-channel input buffer for PCM/IS2 input. The input buffer is configured with one Dual Port RAM, and a one-channel 32-word area is configured by address segmentation.

The input buffer is accessed via the channel-use MXPMISnDADR0-15(n=0 to 4) register. Since access of the MXPMISnDADR0-15(n=0 to 4) register results in FIFO operation of the internal buffer, storage is in access sequence regardless of the access address. The configuration of the input buffer is shown in Table 3-6.

Table 3-6 Outline of Input Buffer

Item	Description
Width of data	32bit
Width of address	8bit
Depth	160(32×5ch)
Composition	Dual Port RAM

3.9.1. Complement of silence data during buffer empty

Mixing operation stops if mixer cannot take input data at necessary timing. To avoid such case, mixer can complement silence PCM data (all "0") when PMIS input buffer is empty.

Complement function can be independently enabled for each channel. Complement function is enabled or disabled immediately when it is set by software.

Table 3-7

PMIS Input BUFFER(n) Condition	MXPMISIPDC PMISIPDCn Setting	Description
Not Empty	**0**	Not Insert
	1	Not Insert
Empty	**0**	Not Insert
	1	Insert the PCM Data (ALL0)

Note:

- *When complement function is disabled, mixing operation stops during input buffer empty even though other channel has PCM data.*

3.10. Sound Source Sampling Rate

3.10.1. Input Sampling Rate

Sound source input/output sampling rates are shown in Table 3-8.

Table 3-8

Input or Output	Frequency						
	96	**48**	**24**	**12**	**8**	**4**	**44.1**
WFG0 to 4	-	Supported	-	-	-	-	-
PMIS0 to 4	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Mixer output	-	Supported	-	-	-	-	-

3.10.2. Sampling Rate Conversion and FIR Filter

The sound mixer uses saturation calculation to output a mixed sound source with a sampling rate of 48 kHz. When a sound source with a sampling rate that is not 48 kHz is input to PMIS0 to 4, the sound mixer converts the sampling rate.

Conversion is by up-sampling, along with processing by an FIR filter to dampen high frequency noise and prevent changes in sound quality. However, down-sampling is performed in the case of a 96 kHz sampling rate, without filtering.

A list of filter processes for input sampling is shown below.

Table 3-9

Sampling Rate (kHz) (Input Sound Source)	Sampling Rate Change	FIR Filter Cut-Off Frequency (kHz)
96	Perform sound sampling.	-
48	Conversion not performed.	-
24	Perform up-sampling and FIR filter to prevent changes in sound quality.	11.0
12		5.0
8		3.0
4		1.2
44.1		17.0

Notes:

- *However, down-sampling is performed in the case of 96 kHz → 48 kHz, without filtering sampling rate, without filtering, which creates the risk of aliasing. Sufficient evaluation with the actual device is recommended.*
- *Filter characteristics have been selected based on Cypress rules.*
- *See 5.3 in detail.*

4. Registers

This section describes the registers of Sound Mixer

4.1. Mixer Channel Register (MXCH)

REGISTER_NAME	Mixer Channel Register (MXCH)
OFFSET	0x000
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PMIS4	PMIS3
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS2	PMIS1	PMIS0	WFG4	WFG3	WFG2	WFG1	WFG0
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:10] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit9] PMIS4: PMIS4 input setting

[bit8] PMIS3: PMIS3 input setting

[bit7] PMIS2: PMIS2 input setting

[bit6] PMIS1: PMIS1 input setting

[bit5] PMIS0: PMIS0 input setting

PMISn (n=0 to 4)	Description
0	PMISn input invalid
1	PMISn input valid

Note:

- *PMISn bit must not be changed from 0 to 1 when DMAENCHn bit of MXDRQCTRL register is 1.*

[bit4] WFG4: WFG4 input setting

[bit3] WFG3: WFG3 input setting

[bit2] WFG2: WFG2 input setting

[bit1] WFG1: WFG1 input setting

[bit0] WFG0: WFG0 input setting

WFGn (n=0 to 4)	Description
0	WFGn input invalid
1	WFGn input valid

Note:

- *Configuring invalid, the channel will be out of mixing operation not depending on the left data on buffer.*

4.2. Mixer Output Control Register (MXOCTRL)

REGISTER_NAME	Mixer Output Control Register (MXOCTRL)
OFFSET	0x004
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	DATATN[3]	DATATN[2]	DATATN[1]	DATATN[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	MACRO[2]	MACRO[1]	MACRO[0]
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:12] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit11:8] DATATN[3:0]: Number of transfers to output destination

Bits[3:0]	Description
0000 to 1111	Transfers the [value written here + 1] number of data items to the output destination.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

[bit7:3] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit2:0] MACRO[2:0]: Output destination address

Bits[2:0]	Description
000 to 111	Specifies [12:10] of the AHB bus address.

Note:

- *These bits can be set at the initialization. After that, it should not be changed.*

4.3. Mixer Date Request Control Register (MXDRQCTRL)

REGISTER_NAME	Mixer Date Request Control Register (MXDRQCTRL)
OFFSET	0x008
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	DMAEN CH4	DMAEN CH3	DMAEN CH2	DMAEN CH1	DMAEN CH0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	FESTCH4[3:0]			
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	FESTCH3[3:0]				FESTCH2[3:0]			
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	FESTCH1[3:0]				FESTCH0[3:0]			
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28] DMAENCH4: DMA transfer to PMIS4 request setting

[bit27] DMAENCH3: DMA transfer to PMIS3 request setting

[bit26] DMAENCH2: DMA transfer to PMIS2 request setting

[bit25] DMAENCH1: DMA transfer to PMIS1 request setting

[bit24] DMAENCH0: DMA transfer to PMIS0 request setting

DMAENCHn (n=0 to 4)	Description
0	Disable
1	Enable

[bit23:20] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit19:16] FESTCH4[3:0]: Data transfer to PMIS4 request assert threshold setting

[bit15:12] FESTCH3[3:0]: Data transfer to PMIS3 request assert threshold setting

[bit11:8] FESTCH2[3:0]: Data transfer to PMIS2 request assert threshold setting

[bit7:4] FESTCH1[3:0]: Data transfer to PMIS1 request assert threshold setting

[bit3:0] FESTCH0[3:0]: Data transfer to PMIS0 request assert threshold setting

FESTCHn[3:0] (n=0 to 4)	Description
0000 to 1111	A data transfer request is asserted when the input buffer free space becomes [value written here + 1] word.

Notes:

- *See Figure 3-1.*
- *When a 16-bit monaural sound source is input in 32-bit units into PMIS0 to 4, the FESTCHn upper limit setting value is 7. This is because twice the buffer capacity is needed to perform the internal process shown in 3.3 on the monaural sound source.*
- *FESTCHn bits must not be changed when DMAENCHn bit is 1.*

4.4. Mixer Input Control Register (MXICTRL)

REGISTER_NAME	Mixer Input Control Register (MXICTRL)
OFFSET	0x00C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	PMIS4FREQ[2:0]		
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	PMIS3FREQ[2:0]			Reserved	PMIS2FREQ[2:0]		
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	1	0	0	0	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	PMIS1FREQ[2:0]			Reserved	PMIS0FREQ[2:0]		
ACCESS_TYPE	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	1	0	0	0	1

[bit31:19] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit18:16] PMIS4FREQ: Input sampling frequency to PMIS4

See descriptions in [bit2:0].

[bit15] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit14:12] PMIS3FREQ: Input sampling frequency to PMIS3

Refer to descriptions in [bit2:0].

[bit11] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit10:8] PMIS2FREQ: Input sampling frequency to PMIS2

Refer to descriptions in [bit2:0].

[bit7] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit6:4] PMIS1FREQ: Input sampling frequency to PMIS1

See descriptions in [bit2:0].

[bit3] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit2:0] PMIS0FREQ[2:0]: Input sampling frequency to PMIS0

Bits[2:0]	Description
000	96 kHz
001	48 kHz
010	24 kHz
011	12 kHz
100	8 kHz
101	4 kHz
110	44.1 kHz
111	Reserved

Notes:

- *Note that the initial value is 001 (48 kHz).*
- *The 44.1 kHz input sampling rate can be applied to only one of the input channels PMIS0 to 4. Operation when 44.1 kHz is applied to multiple channels cannot be guaranteed from the viewpoint of the mixing process load imposed by the sound mixer. It should be noted that there may limits on specific models concerning the selection of other frequencies.*

4.5. Mixer Channel Monaural Register (MXCHMONO)

REGISTER_NAME	Mixer Channel Monaural Register (MXCHMONO)
OFFSET	0x010
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PMIS4MONO[1:0]	
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS3MONO[1:0]		PMIS2MONO[1:0]		PMIS1MONO[1:0]		PMIS0MONO[1:0]	
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:10] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit9:8] PMIS4MONO[1:0]: PMIS4 sound source processing mode setting

[bit7:6] PMIS3MONO[1:0]: PMIS3 sound source processing mode setting

[bit5:4] PMIS2MONO[1:0]: PMIS2 sound source processing mode setting

[bit3:2] PMIS1MONO[1:0]: PMIS1 sound source processing mode setting

[bit1:0] PMIS0MONO[1:0]: PMIS0 sound source processing mode setting

PMISnMONO[1:0] (n=0 to 4)	Description
00	Stereo processing
01	Monaural process R (Upper 16 bits = ALL 0, Lower 16 bits = Monaural sound source).
10	Monaural process L (Upper 16 bits = Monaural sound source, Lower 16 bits = ALL 0).
11	Monaural process LR (Upper 16 bits = Lower 16 bits = Monaural sound source).

Note:

- *These bits must not be changed during sound source transmission of the channel.*

4.6. Mixer Channel Volume1 Register (MXCHVOL1)

REGISTER_NAME	Mixer Channel Volume1 Register (MXCHVOL1)
OFFSET	0x020
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	WFG3VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	WFG2VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	WFG1VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WFG0VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

[bit31:24] WFG3VOL[7:0]: Gain setting for WFG3 volume control

[bit23:16] WFG2VOL[7:0]: Gain setting for WFG2 volume control

[bit15:8] WFG1VOL[7:0]: Gain setting for WFG1 volume control

[bit7:0] WFG0VOL[7:0]: Gain setting for WFG0 volume control

For details about Gain settings for volume control, see 5.1.

4.7. Mixer Channel Volume2 Register (MXCHVOL2)

REGISTER_NAME	Mixer Channel Volume2 Register (MXCHVOL2)
OFFSET	0x024
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PMIS2VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PMIS1VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMIS0VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WFG4VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

[bit31:24] PMIS2VOL[7:0]: Gain setting for PMIS2VOL volume control

[bit23:16] PMIS1VOL[7:0]: Gain setting for PMIS1VOL volume control

[bit15:8] PMIS0VOL[7:0]: Gain setting for PMIS0VOL volume control

[bit7:0] WFG4VOL[7:0]: Gain setting for WFG4 volume control

For details about Gain settings for volume control, see 5.1.

4.8. Mixer Channel Volume3 Register (MXCHVOL3)

REGISTER_NAME	Mixer Channel Volume3 Register (MXCHVOL3)
OFFSET	0x028
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	MXDVOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMIS4VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS3VOL[7:0]							
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	1	1	1	0	0	1	1	1

[bit31:24] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit23:16] MXDVOL[7:0]: Gain setting for mixed sound source volume control

[bit15:8] PMIS4VOL[7:0]: Gain setting for PMIS4VOL volume control

[bit7:0] PMIS3VOL[7:0]: Gain setting for PMIS3VOL volume control

For details about Gain settings for volume control, see 5.1.

4.9. Mixer Channel Mute Register (MXCHMUTE)

REGISTER_NAME	Mixer Channel Mute Register (MXCHMUTE)
OFFSET	0x02C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	MXD MUTE	PMIS4 MUTE	PMIS3 MUTE
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS2 MUTE	PMIS1 MUTE	PMIS0 MUTE	WFG4 MUTE	WFG3 MUTE	WFG2 MUTE	WFG1 MUTE	WFG0 MUTE
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:11] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit10] MXDMUTE: Mixed sound source MUTE setting

Bit	Description
0	Disable mute.
1	Enable mute.

[bit9] PMIS4MUTE: PMIS4 input sound source MUTE setting

[bit8] PMIS3MUTE: PMIS3 input sound source MUTE setting

[bit7] PMIS2MUTE: PMIS2 input sound source MUTE setting

[bit6] PMIS1MUTE: PMIS1 input sound source MUTE setting

[bit5] PMIS0MUTE: PMIS0 input sound source MUTE setting

PMISnMUTE (n=0 to 4)	Description
0	Disable mute.
1	Enable mute.

[bit4] WFG4MUTE: WFG4 input sound source MUTE setting

[bit3] WFG3MUTE: WFG3 input sound source MUTE setting

[bit2] WFG2MUTE: WFG2 input sound source MUTE setting

[bit1] WFG1MUTE: WFG1 input sound source MUTE setting

[bit0] WFG0MUTE: WFG0 input sound source MUTE setting

WFGnMUTE (n=0 to 4)	Description
0	Disable mute.
1	Enable mute.

4.10. Mixer Channel Fade_In/Out1 Register (MXCHFADE1)

REGISTER_NAME	Mixer Channel Fade_in/out1 Register (MXCHFADE1)
OFFSET	0x030
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	WFG1FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	WFG1FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	WFG0FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	WFG0FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:24] WFG1FADEOUT[4:0]: WFG1 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit23:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] WFG1FADEIN[4:0]: WFG1 fade in time setting

For details about fade in time settings for volume control, see 5.2.

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] WFG0FADEOUT[4:0]: WFG0 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] WFG0FADEIN[4:0]: WFG0 fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.11. Mixer Channel Fade_In/Out2 Register (MXCHFADE2)

REGISTER_NAME	Mixer Channel Fade_in/out2 Register (MXCHFADE2)
OFFSET	0x034
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	WFG3FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	WFG3FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	WFG2FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	WFG2FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:24] WFG3FADEOUT[4:0]: WFG3 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit23:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] WFG3FADEIN[4:0]: WFG3 fade in time setting

For details about fade in time settings for volume control, see 5.2.

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] WFG2FADEOUT[4:0]: WFG2 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] WFG2FADEIN[4:0]: WFG2 fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.12. Mixer Channel Fade_In/Out3 Register (MXCHFADE3)

REGISTER_NAME	Mixer Channel Fade_in/out3 Register (MXCHFADE3)
OFFSET	0x038
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	PMIS0FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	PMIS0FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	WFG4FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	WFG4FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:24] PMIS0FADEOUT[4:0]: PMIS0 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit23:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] PMIS0FADEIN[4:0]: PMIS0 fade in time setting

For details about fade in time settings for volume control, see 5.2.

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] WFG4FADEOUT[4:0]: WFG4 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] WFG4FADEIN[4:0]: WFG4 fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.13. Mixer Channel Fade_In/Out4 Register (MXCHFADE4)

REGISTER_NAME	Mixer Channel Fade_in/out4 Register (MXCHFADE4)
OFFSET	0x03C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	PMIS2FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	PMIS2FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	PMIS1FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS1FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:24] PMIS2FADEOUT[4:0]: PMIS2 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit23:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] PMIS2FADEIN[4:0]: PMIS2 fade in time setting

For details about fade in time settings for volume control, see 5.2.

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] PMIS1FADEOUT[4:0]: PMIS1 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] PMIS1FADEIN[4:0]: PMIS1 fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.14. Mixer Channel Fade_in/out5 Register (MXCHFADE5)

REGISTER_NAME	Mixer Channel Fade_in/out5 Register (MXCHFADE5)
OFFSET	0x040
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	PMIS4FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	PMIS4FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	PMIS3FADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS3FADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28:24] PMIS4FADEOUT[4:0]: PMIS4 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit23:21] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit20:16] PMIS4FADEIN[4:0]: PMIS4 fade in time setting

For details about fade in time settings for volume control, see 5.2.

[bit15:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] PMIS3FADEOUT[4:0]: PMIS3 fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] PMIS3FADEIN[4:0]: PMIS3 fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.15. Mixer Mixed Fade_In/Out Register (MXMXDFADE)

REGISTER_NAME	Mixer Mixed Fade_in/out Register (MXMXDFADE)
OFFSET	0x044
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	MXDFADEOUT[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	MXDFADEIN[4:0]				
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12:8] MXDFADEOUT[4:0]: Mixed sound source fade out time setting

For details about fade out time settings for volume control, see 5.2.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4:0] MXDFADEIN[4:0]: Mixed sound source fade in time setting

For details about fade in time settings for volume control, see 5.2.

4.16. Mixer Channel Fade_In/Out Enable Register (MXCHFADEEN)

REGISTER_NAME	Mixer Channel Fade_in/out Enable Register(MXCHFADEEN)
OFFSET	0x048
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	MXDFADEEN[1:0]		PMIS4FADEEN[1:0]		PMIS3FADEEN[1:0]	
ACCESS_TYPE	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMIS2FADEEN[1:0]		PMIS1FADEEN[1:0]		PMIS0FADEEN[1:0]		WFG4FADEEN[1:0]	
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WFG3FADEEN[1:0]		WFG2FADEEN[1:0]		WFG1FADEEN[1:0]		WFG0FADEEN[1:0]	
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:22] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit21:20] MXDFADEEN[1:0]: Mixed sound source fade in/out operation setting

Bits[1:0]	Description
00	No fade in/fade out
01	Setting prohibited
10	Enable fade in operation start.
11	Enable fade out operation start.

[bit19:18] PMIS4FADEEN[1:0]: PMIS4 fade in/out operation setting

[bit17:16] PMIS3FADEEN[1:0]: PMIS3 fade in/out operation setting

[bit15:14] PMIS2FADEEN[1:0]: PMIS2 fade in/out operation setting

[bit13:12] PMIS1FADEEN[1:0]: PMIS1 fade in/out operation setting

[bit11:10] PMIS0FADEEN[1:0]: PMIS0 fade in/out operation setting

PMISnFADEEN[1:0] (n=0 to 4)	Description
00	No fade in/fade out
01	Setting prohibited
10	Enable fade in operation start.
11	Enable fade out operation start.

[bit9:8] WFG4FADEEN[1:0]: WFG4 fade in/out operation setting

[bit7:6] WFG3FADEEN[1:0]: WFG3 fade in/out operation setting

[bit5:4] WFG2FADEEN[1:0]: WFG2 fade in/out operation setting

[bit3:2] WFG1FADEEN[1:0]: WFG1 fade in/out operation setting

[bit1:0] WFG0FADEEN[1:0]: WFG0 fade in/out operation setting

WFGnFADEEN[1:0] (n=0 to 4)	Description
00	No fade in/fade out
01	Setting prohibited
10	Enable fade in operation start.
11	Enable fade out operation start.

4.17. Mixer Insert the PCM Data Control Register (MXPMISIPDC)

REGISTER_NAME	Mixer Insert the PCM Data Control Register (MXPMISIPDC)
OFFSET	0x04C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS IPDC4	PMIS IPDC3	PMIS IPDC2	PMIS IPDC1	PMIS IPDC0
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] PMISIPDC4: PMISIPDC4 input setting

[bit3] PMISIPDC3: PMISIPDC3 input setting

[bit2] PMISIPDC2: PMISIPDC2 input setting

[bit1] PMISIPDC1: PMISIPDC1 input setting

[bit0] PMISIPDC0: PMISIPDC0 input setting

These bits enable complement silence (all 0) data function during buffer empty at PMIS input buffer.

PMISIPDCn (n=0 to 4)	Description
0	Disable to insert the silence (all 0) data
1	Enable to insert the silence (all 0) data

4.18. Mixer Buffer Clear Register (MXBUFFCLR)

REGISTER_NAME	Mixer Buffer Clear Register (MXBUFFCLR)
OFFSET	0x050
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	OUTBCLR	PMIS4 BCLR	PMIS3 BCLR
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS2 BCLR	PMIS1 BCLR	PMIS0 BCLR	WFG4 BCLR	WFG3 BCLR	WFG2 BCLR	WFG1 BCLR	WFG0 BCLR
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:11] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit10] OUTBCLR: Output buffer initialization

Bit	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize buffer (This bit automatically cleared to 0 following initialization.)

[bit9] PMIS4BCLR: PMIS4 input buffer initialization

[bit8] PMIS3BCLR: PMIS3 input buffer initialization

[bit7] PMIS2BCLR: PMIS2 input buffer initialization

[bit6] PMIS1BCLR: PMIS1 input buffer initialization

[bit5] PMIS0BCLR: PMIS0 input buffer initialization

PMISnBCLR (n=0 to 4)	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize buffer (This bit automatically cleared to 0 following initialization.)

Note:

- *PMISn input buffer initialization must not be executed when DMAENCHn bit of MXDRQCTRL register is 1.*

[bit4] WFG4BCLR: WFG4 input buffer initialization

[bit3] WFG3BCLR: WFG3 input buffer initialization

[bit2] WFG2BCLR: WFG2 input buffer initialization

[bit1] WFG1BCLR: WFG1 input buffer initialization

[bit0] WFG0BCLR: WFG0 input buffer initialization

WFGnBCLR (n=0 to 4)	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize buffer (This bit automatically cleared to 0 following initialization.)

Note:

Every channel of WFG0 to 4 has an input buffer for a word (32bit) in the part of "input control".

4.19. Mixer FADE_In/Out Clear Register (MXFADECLR)

REGISTER_NAME	Mixer FADE_in/out Clear Register (MXFADECLR)
OFFSET	0x054
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	MIXCLR	PMIS4 CLR	PMIS3 CLR
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMIS2 CLR	PMIS1 CLR	PMIS0 CLR	WFG4 CLR	WFG3 CLR	WFG2 CLR	WFG1 CLR	WFG0 CLR
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:11] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit10] MIXCLR: Mixed data FADE state initialization

Bit	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize fade state (This bit automatically cleared to 0 following initialization.)

[bit9] PMIS4CLR: PMIS4 FADE state initialization

[bit8] PMIS3CLR: PMIS3 FADE state initialization

[bit7] PMIS2CLR: PMIS2 FADE state initialization

[bit6] PMIS1CLR: PMIS1 FADE state initialization

[bit5] PMIS0CLR: PMIS0 FADE state initialization

PMISnCLR (n=0 to 4)	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize fade state (This bit automatically cleared to 0 following initialization.)

[bit4] WFG4CLR: WFG4 FADE state initialization

[bit3] WFG3CLR: WFG3 FADE state initialization

[bit2] WFG2CLR: WFG2 FADE state initialization

[bit1] WFG1CLR: WFG1 FADE state initialization

[bit0] WFG0CLR: WFG0 FADE state initialization

WFGnCLR (n=0 to 4)	Description	
	Read	Write
0	Not under initialization	Does not have any effect on operation.
1	Initialization wait	Initialize fade state (This bit automatically cleared to 0 following initialization.)

Note:

- *"1" writing to initialize should be done after all the bits of MXFADECLR to be "0".*

4.20. Mixer Interrupt Enable Register (MXINTREN)

REGISTER_NAME	Mixer Interrupt Enable Register (MXINTREN)
OFFSET	0x060
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	AHBERR	Reserved	Reserved	PMIS4 DMAERR	PMIS3 DMAERR	PMIS2 DMAERR	PMIS1 DMAERR	PMIS0 DMAERR
ACCESS_TYPE	R/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFOVFL	PMIS3 BUFOVFL	PMIS2 BUFOVFL	PMIS1 BUFOVFL	PMIS0 BUFOVFL
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFDREQ	PMIS3 BUFDREQ	PMIS2 BUFDREQ	PMIS1 BUFDREQ	PMIS0 BUFDREQ
ACCESS_TYPE	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] AHBERR: AHB Master interface bus error interrupt setting

Bit	Description
0	Disable interrupts.
1	Enable interrupts.

[bit30:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28] PMIS4DMAERR: PMIS4DMA transfer error interrupt setting

[bit27] PMIS3DMAERR: PMIS3DMA transfer error interrupt setting

[bit26] PMIS2DMAERR: PMIS2DMA transfer error interrupt setting

[bit25] PMIS1DMAERR: PMIS1DMA transfer error interrupt setting

[bit24] PMIS0DMAERR: PMIS0DMA transfer error interrupt setting

PMISnBUFOVFL (n=0 to 4)	Description
0	Disable interrupts.
1	Enable interrupts.

[bit23:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12] PMIS4BUFOVFL: PMIS4 input buffer overflow interrupt setting

[bit11] PMIS3BUFOVFL: PMIS3 input buffer overflow interrupt setting

[bit10] PMIS2BUFOVFL: PMIS2 input buffer overflow interrupt setting

[bit9] PMIS1BUFOVFL: PMIS1 input buffer overflow interrupt setting

[bit8] PMIS0BUFOVFL: PMIS0 input buffer overflow interrupt setting

PMISnBUFOVFL (n=0 to 4)	Description
0	Disable interrupts.
1	Enable interrupts.

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] PMIS4BUFDREQ: PMIS4 data transfer request interrupt setting

[bit3] PMIS3BUFDREQ: PMIS3 data transfer request interrupt setting

[bit2] PMIS2BUFDREQ: PMIS2 data transfer request interrupt setting

[bit1] PMIS1BUFDREQ: PMIS1 data transfer request interrupt setting

[bit0] PMIS0BUFDREQ: PMIS0 data transfer request interrupt setting

PMISnBUFDREQ (n=0 to 4)	Description
0	Disable interrupts.
1	Enable interrupts.

Note:

- *When use of DMA request is disabled by the setting of MSDRQCTRL under 4.3, be sure to disable DMA transfer error interrupt. Failure to do so can result in generation of unexpected interrupts.*

4.21. Mixer Interrupt Status Register (MXINTRSTATE)

REGISTER_NAME	Mixer Interrupt Status Register (MXINTRSTATE)
OFFSET	0x064
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	AHBERR	Reserved	Reserved	PMIS4 DMAERR	PMIS3 DMAERR	PMIS2 DMAERR	PMIS1 DMAERR	PMIS0 DMAERR
ACCESS_TYPE	R,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFOVFL	PMIS3 BUFOVFL	PMIS2 BUFOVFL	PMIS1 BUFOVFL	PMIS0 BUFOVFL
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFDREQ	PMIS3 BUFDREQ	PMIS2 BUFDREQ	PMIS1 BUFDREQ	PMIS0 BUFDREQ
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] AHBERR: AHB master interface bus error indication

Bit	Description
0	No error/no transfer request
1	Error/transfer request present

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

[bit30:29] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit28] PMIS4DMAERR: PMIS4 DMA transfer error indication

[bit27] PMIS3DMAERR: PMIS3 DAM transfer error indication

[bit26] PMIS2DMAERR: PMIS2DMA transfer error indication

[bit25] PMIS1DMAERR: PMIS1DMA transfer error indication

[bit24] PMIS0DMAERR: PMIS0DMA transfer error indication

PMISnDMAERR (n=0 to 4)	Description
0	No error/no transfer request
1	Error/transfer request present

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

[bit23:13] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit12] PMIS4BUFOVFL: PMIS4 input buffer overflow error indication

[bit11] PMIS3BUFOVFL: PMIS3 input buffer overflow error indication

[bit10] PMIS2BUFOVFL: PIS2 input buffer overflow error indication

[bit9] PMIS1BUFOVFL: PMIS1 input buffer overflow error indication

[bit8] PMIS0BUFOVFL: PMIS0 input buffer overflow error indication

PMISnBUFOVFL (n=0 to 4)	Description
0	No error/no transfer request
1	Error/transfer request present

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

[bit7:5] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit4] PMIS4BUFDREQ: PMIS4 data transfer request indication

[bit3] PMIS3BUFDREQ: PMIS3 data transfer request indication

[bit2] PMIS2BUFDREQ: PMIS2 data transfer request indication

[bit1] PMIS1BUFDREQ: PMIS1 data transfer request indication

[bit0] PMIS0BUFDREQ: PMIS0 data transfer request indication

PMISnBUFDREQ (n=0 to 4)	Description
0	No error/no transfer request
1	Error/transfer request present

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

4.22. Mixer Interrupt Clear Register (MXINTRCLR)

REGISTER_NAME	Mixer Interrupt Clear Register (MXINTRCLR)
OFFSET	0x068
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	AHBERR	Reserved	Reserved	PMIS4 DMAERR	PMIS3 DMAERR	PMIS2 DMAERR	PMIS1 DMAERR	PMIS0 DMAERR
ACCESS_TYPE	RX,W	RX,W0	RX,W0	RX,W	RX,W	RX,W	RX,W	RX,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFOVFL	PMIS3 BUFOVFL	PMIS2 BUFOVFL	PMIS1 BUFOVFL	PMIS0 BUFOVFL
ACCESS_TYPE	RX,W0	RX,W0	RX,W0	RX,W	RX,W	RX,W	RX,W	RX,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	PMIS4 BUFDREQ	PMIS3 BUFDREQ	PMIS2 BUFDREQ	PMIS1 BUFDREQ	PMIS0 BUFDREQ
ACCESS_TYPE	RX,W0	RX,W0	RX,W0	RX,W	RX,W	RX,W	RX,W	RX,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] AHBERR: AHB master interface bus error interrupt clear

Bit	Description
0	Does not have any effect on operation.
1	Clear interrupts.

Note:

- *If interrupt generation and interrupt clear occur simultaneously, interrupt generation is given priority.*

[bit30:29] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit28] PMIS4DMAERR: PMIS4 DMA transfer error interrupt clear

[bit27] PMIS3DMAERR: PMIS3 DMA transfer error interrupt clear

[bit26] PMIS2DMAERR: PMIS2DMA transfer error interrupt clear

[bit25] PMIS1DMAERR: PMIS1DMA transfer error interrupt clear

[bit24] PMIS0DMAERR: PMIS0DMA transfer error interrupt clear

PMISnDMAERR (n=0 to 4)	Description
0	Does not have any effect on operation.
1	Clear interrupts.

Note:

- *If interrupt generation and interrupt clear occur simultaneously, interrupt generation is given priority.*

[bit23:13] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit12] PMIS4BUFOVFL: PMIS4 input buffer overflow interrupt clear

[bit11] PMIS3BUFOVFL: PMIS3 input buffer overflow interrupt clear

[bit10] PMIS2BUFOVFL: PMIS2 input buffer overflow interrupt clear

[bit9] PMIS1BUFOVFL: PMIS1 input buffer overflow interrupt clear

[bit8] PMIS0BUFOVFL: PMIS0 input buffer overflow interrupt clear

PMISnBUFOVFL (n=0 to 4)	Description
0	Does not have any effect on operation.
1	Clear interrupts.

Note:

- *If interrupt generation and interrupt clear occur simultaneously, interrupt generation is given priority.*

[bit7:5] Reserved

This bit is reserved.
Always write "0" to this bit. The read value is "0".

[bit4] PMIS4BUFDREQ: PMIS4 data transfer request interrupt clear

[bit3] PMIS3BUFDREQ: PMIS3 data transfer request interrupt clear

[bit2] PMIS2BUFDREQ: PMIS2 data transfer request interrupt clear

[bit1] PMIS1BUFDREQ: PMIS1 data transfer request interrupt clear

[bit0] PMIS0BUFDREQ: PMIS0 data transfer request interrupt clear

PMISnBUFDREQ (n=0 to 4)	Description
0	Does not have any effect on operation.
1	Clear interrupts.

Note:

- *If interrupt generation and interrupt clear occur simultaneously, interrupt generation is given priority.*

4.23. Mixer Input Buffer Count1 Register (MXINBUFFCNT1)

REGISTER_NAME	Mixer Input Buffer Count1 Register (MXINBUFFCNT1)
OFFSET	0x070
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	PMIS2CNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	PMIS1CNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	PMIS0CNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	WFG4C	WFG3C	WFG2C	WFG1C	WFG0C
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:30] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit29:24] PMIS2CNT[5:0]: PMIS2 input buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Used input buffer volume

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

[bit23:20] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit21:16] PMIS1CNT[5:0]: PMIS1 input buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Used input buffer volume

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

[bit15:14] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit13:8] PMIS0CNT[5:0]: PMIS0 input buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Used input buffer volume

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

[bit7:5] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit4] WFG4C: WFG4 input data present/not present indication

[bit3] WFG3C: WFG3 input data present/not present indication

[bit2] WFG2C: WFG2 input data present/not present indication

[bit1] WFG1C: WFG1 input data present/not present indication

[bit0] WFG0C: WFG0 input data present/not present indication

WFGnC (n=0 to 4)	Description
0	Without data
1	With data

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

4.24. Mixer Input Buffer Count2 Register (MXINBUFFCNT2)

REGISTER_NAME	Mixer Input Buffer Count2 Register (MXINBUFFCNT2)
OFFSET	0x074
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	PMIS4CNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	PMIS3CNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:14] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit13:8] PMIS4CNT[5:0]: Input buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Used input buffer volume

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

[bit7:6] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit5:0] PMIS3CNT[5:0]: Input buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Used input buffer volume

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

4.25. Mixer Channel Buffer Count Register (MXCHBUFFCNT)

REGISTER_NAME	Mixer Input Buffer Count Register (MXCHBUFFCNT)
OFFSET	0x078
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	PMIS4CNT[3:0]			
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PMIS3CNT[3:0]				PMIS2CNT[3:0]			
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMIS1CNT[3:0]				PMIS0CNT[3:0]			
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	WFG4C	WFG3C	WFG2C	WFG1C	WFG0C
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:28] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit27:24] PMIS4CNT[3:0]: PMIS4 internal buffer used volume indication

[bit23:20] PMIS3CNT[3:0]: PMIS3 internal buffer used volume indication

[bit19:16] PMIS2CNT[3:0]: PMIS2 internal buffer used volume indication

[bit15:12] PMIS1CNT[3:0]: PMIS1 internal buffer used volume indication

[bit11:8] PMIS0CNT[3:0]: PMIS0 internal buffer used volume indication

PMISnCNT (n=0 to 4)	Description
0000	Buffer is empty.
0001 to 1111	Internal buffer used volume display

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

[bit7:5] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit4] WFG4C: WFG4 internal buffer used volume indication

[bit3] WFG3C: WFG3 internal buffer used volume indication

[bit2] WFG2C: WFG2 internal buffer used volume indication

[bit1] WFG1C: WFG1 internal buffer used volume indication

[bit0] WFG0C: WFG0 internal buffer used volume indication

WFGnC (n=0 to 4)	Description
0	Buffer is empty.
1	Internal buffer used volume display

Note:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.

4.26. Mixer Output Buffer Count Register (MXOUTBUFFCNT)

REGISTER_NAME	Mixer Output Buffer Count Register (MXOUTBUFFCNT)
OFFSET	0x07C
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	OUTCNT[5:0]					
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:6] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit5:0] OUTCNT[5:0]: Output buffer used volume indication

Bits[5:0]	Description
000000	Buffer is empty.
000001 to 111111	Output buffer used volume

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

4.27. Mixer AHB Bus Error Register (MXAHBERR)

REGISTER_NAME	Mixer AHB Bus Error Register (MXAHBERR)
OFFSET	0x080
ACCESS_SIZE	B H W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	CNTREGERR[1:0]		PMIS4ERR[1:0]		PMIS3ERR[1:0]	
ACCESS_TYPE	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMIS2ERR[1:0]		PMIS1ERR[1:0]		PMIS0ERR[1:0]		WFG4ERR[1:0]	
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WFG3ERR[1:0]		WFG2ERR[1:0]		WFG1ERR[1:0]		WFG0ERR[1:0]	
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:22] Reserved

This bit is reserved.
The read value is "0".
This bit is read-only, and writing is prohibited.
Writing to this bit generates a sound mixer AHB Slave interface access error.

[bit21:20] CNTREGERR[1:0]: AHB Slave interface register access error information indication

Bits[1:0]	Description
00	There is no error.
01	Address error
10	Write access to Read Only register
11	Access size error

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

[bit19:18] PMIS4ERR[1:0]: AHB Slave interface PMIS4 transfer access error information indication

[bit17:16] PMIS3ERR[1:0]: AHB Slave interface PMIS3 transfer access error information indication

[bit15:14] PMIS2ERR[1:0]: AHB Slave interface PMIS2 transfer access error information indication

[bit13:12] PMIS1ERR[1:0]: AHB Slave interface PMIS1 transfer access error information indication

[bit11:10] PMIS0ERR[1:0]: AHB Slave interface PMIS0 transfer access error information indication

PMISnERR[1:0] (n=0 to 4)	Description
00	There is no error.
01	Address error
10	Write access to Read Only register
11	Access size error

Notes:

- *This bit is read-only, and writing is prohibited.*
- *Writing to this bit generates a sound mixer AHB Slave Interface access error.*

[bit9:8] WFG4ERR[1:0]: AHB Slave interface WFG4 transfer access error information indication

[bit7:6] WFG3ERR[1:0]: AHB Slave interface WFG3 transfer access error information indication

[bit5:4] WFG2ERR[1:0]: AHB Slave interface WFG2 transfer access error information indication

[bit3:2] WFG1ERR[1:0]: AHB Slave interface WFG1 transfer access error information indication

[bit1:0] WFG0ERR[1:0]: AHB Slave interface WFG0 transfer access error information indication

WFGnERR[1:0] (n=0 to 4)	Description
00	There is no error.
01	Address error
10	Write access to Read Only register
11	Access size error

Notes:

- *This bit is read-only, and writing is prohibited.*
  Writing to this bit generates a sound mixer AHB Slave Interface access error.
- *Every address error during transmission to WFG0 through 4 is informed to WFG0ERR[1:0].*
  It is also informed to CNTREGERR[1:0] during transmission to PMIS0 through 4.
- *Every write access error of Read Only register is informed to CNTREERR[1:0].*

4.28. Mixer WFGn Data Address Register (MXWFGnDADR, n=0 to 4)

REGISTER_NAME	Mixer WFGn Data Address Register (MXWFGnDADR, n=0 to 4)
OFFSET	0x100(n=0), 0x104(n=1), 0x108(n=2), 0x10C(n=3), 0x110(n=4)
ACCESS_SIZE	W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	WFGnDADR[31:24]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	WFGnDADR[23:16]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	WFGnDADR[15:8]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	WFGnDADR[7:0]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] WFGnDADR[31:0]: WFGn input data register

Note:

- *Because Sound waveform generator output has a fixed connection to this register, this register cannot be written from the CPU. Writing generates a sound mixer AHB Slave Interface access error.*

4.29. Mixer PMIS n Data Address Register0-15 (MXPMISnDADR0-15, n=0 to 4)

REGISTER_NAME	Mixer PMIS n Data Address Register (MXPMISnDADR, n=0 to 4)
OFFSET	0x200 to 0x23C(n=0), 0x240 to 0x27C(n=1), 0x280 to 0x2BC(n=2) 0x2C0 to 0x2FC(n=3), 0x300 to 0x33C(n=4)
ACCESS_SIZE	W
MULTIPLE	31:0
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	PMISnDADR0 to 15[31:24]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	PMISnDADR0 to 15[23:16]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PMISnDADR0 to 15[15:8]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PMISnDADR0 to 15[7:0]							
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE	Wp							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] PMISnDADR0 to 15[31:0]: PMIS input data register

5. Appendix

5.1. Gain for Volume Control

Table 5-1

Bit[7:0]	Description	
	Gain (dB)	Magnification
00000000	-96.0	0.000016
00000001	-96.0	0.000016
00000010	-96.0	0.000016
00000011	-96.0	0.000016
00000100	-96.0	0.000016
00000101	-96.0	0.000016
00000110	-96.0	0.000016
00000111	-96.0	0.000016
00001000	-96.0	0.000016
00001001	-96.0	0.000016
00001010	-96.0	0.000016
00001011	-96.0	0.000016
00001100	-96.0	0.000016
00001101	-96.0	0.000016
00001110	-96.0	0.000016
00001111	-96.0	0.000016
00010000	-96.0	0.000016
00010001	-96.0	0.000016
00010010	-96.0	0.000016
00010011	-96.0	0.000016
00010100	-96.0	0.000016
00010101	-96.0	0.000016
00010110	-96.0	0.000016
00010111	-96.0	0.000016
00011000	-96.0	0.000016
00011001	-96.0	0.000016
00011010	-96.0	0.000016
00011011	-96.0	0.000016
00011100	-96.0	0.000016
00011101	-96.0	0.000016
00011110	-96.0	0.000016
00011111	-96.0	0.000016
00100000	-96.0	0.000016
00100001	-96.0	0.000016
00100010	-96.0	0.000016

Bit[7:0]	Description	
	Gain (dB)	Magnification
00100011	-96.0	0.000016
00100100	-96.0	0.000016
00100101	-96.0	0.000016
00100110	-96.0	0.000016
00100111	-96.0	0.000016
00101000	-95.5	0.000017
00101001	-95.0	0.000018
00101010	-94.5	0.000019
00101011	-94.0	0.000020
00101100	-93.5	0.000021
00101101	-93.0	0.000022
00101110	-92.5	0.000024
00101111	-92.0	0.000025
00110000	-91.5	0.000027
00110001	-91.0	0.000028
00110010	-90.5	0.000030
00110011	-90.0	0.000032
00110100	-89.5	0.000033
00110101	-89.0	0.000035
00110110	-88.5	0.000038
00110111	-88.0	0.000040
00111000	-87.5	0.000042
00111001	-87.0	0.000045
00111010	-86.5	0.000047
00111011	-86.0	0.000050
00111100	-85.5	0.000053
00111101	-85.0	0.000056
00111110	-84.5	0.000060
00111111	-84.0	0.000063
01000000	-83.5	0.000067
01000001	-83.0	0.000071
01000010	-82.5	0.000075
01000011	-82.0	0.000079
01000100	-81.5	0.000084
01000101	-81.0	0.000089
01000110	-80.5	0.000094
01000111	-80.0	0.000100
01001000	-79.5	0.000106
01001001	-79.0	0.000112
01001010	-78.5	0.000119

Bit[7:0]	Description	
	Gain (dB)	Magnification
01001011	-78.0	0.000126
01001100	-77.5	0.000133
01001101	-77.0	0.000141
01001110	-76.5	0.000150
01001111	-76.0	0.000158
01010000	-75.5	0.000168
01010001	-75.0	0.000178
01010010	-74.5	0.000188
01010011	-74.0	0.000200
01010100	-73.5	0.000211
01010101	-73.0	0.000224
01010110	-72.5	0.000237
01010111	-72.0	0.000251
01011000	-71.5	0.000266
01011001	-71.0	0.000282
01011010	-70.5	0.000299
01011011	-70.0	0.000316
01011100	-69.5	0.000335
01011101	-69.0	0.000355
01011110	-68.5	0.000376
01011111	-68.0	0.000398
01100000	-67.5	0.000422
01100001	-67.0	0.000447
01100010	-66.5	0.000473
01100011	-66.0	0.000501
01100100	-65.5	0.000531
01100101	-65.0	0.000562
01100110	-64.5	0.000596
01100111	-64.0	0.000631
01101000	-63.5	0.000668
01101001	-63.0	0.000708
01101010	-62.5	0.000750
01101011	-62.0	0.000794
01101100	-61.5	0.000841
01101101	-61.0	0.000891
01101110	-60.5	0.000944
01101111	-60.0	0.001000
01110000	-59.5	0.001059
01110001	-59.0	0.001122
01110010	-58.5	0.001189

Bit[7:0]	Description	
	Gain (dB)	Magnification
01110011	-58.0	0.001259
01110100	-57.5	0.001334
01110101	-57.0	0.001413
01110110	-56.5	0.001496
01110111	-56.0	0.001585
01111000	-55.5	0.001679
01111001	-55.0	0.001778
01111010	-54.5	0.001884
01111011	-54.0	0.001995
01111100	-53.5	0.002113
01111101	-53.0	0.002239
01111110	-52.5	0.002371
01111111	-52.0	0.002512
10000000	-51.5	0.002661
10000001	-51.0	0.002818
10000010	-50.5	0.002985
10000011	-50.0	0.003162
10000100	-49.5	0.003350
10000101	-49.0	0.003548
10000110	-48.5	0.003758
10000111	-48.0	0.003981
10001000	-47.5	0.004217
10001001	-47.0	0.004467
10001010	-46.5	0.004732
10001011	-46.0	0.005012
10001100	-45.5	0.005309
10001101	-45.0	0.005623
10001110	-44.5	0.005957
10001111	-44.0	0.006310
10010000	-43.5	0.006683
10010001	-43.0	0.007079
10010010	-42.5	0.007499
10010011	-42.0	0.007943
10010100	-41.5	0.008414
10010101	-41.0	0.008913
10010110	-40.5	0.009441
10010111	-40.0	0.010000
10011000	-39.5	0.010593
10011001	-39.0	0.011220
10011010	-38.5	0.011885

Bit[7:0]	Description	
	Gain (dB)	Magnification
10011011	-38.0	0.012589
10011100	-37.5	0.013335
10011101	-37.0	0.014125
10011110	-36.5	0.014962
10011111	-36.0	0.015849
10100000	-35.5	0.016788
10100001	-35.0	0.017783
10100010	-34.5	0.018836
10100011	-34.0	0.019953
10100100	-33.5	0.021135
10100101	-33.0	0.022387
10100110	-32.5	0.023714
10100111	-32.0	0.025119
10101000	-31.5	0.026607
10101001	-31.0	0.028184
10101010	-30.5	0.029854
10101011	-30.0	0.031623
10101100	-29.5	0.033497
10101101	-29.0	0.035481
10101110	-28.5	0.037584
10101111	-28.0	0.039811
10110000	-27.5	0.042170
10110001	-27.0	0.044668
10110010	-26.5	0.047315
10110011	-26.0	0.050119
10110100	-25.5	0.053088
10110101	-25.0	0.056234
10110110	-24.5	0.059566
10110111	-24.0	0.063096
10111000	-23.5	0.066834
10111001	-23.0	0.070795
10111010	-22.5	0.074989
10111011	-22.0	0.079433
10111100	-21.5	0.084140
10111101	-21.0	0.089125
10111110	-20.5	0.094406
10111111	-20.0	0.100000
11000000	-19.5	0.105925
11000001	-19.0	0.112202
11000010	-18.5	0.118850

Bit[7:0]	Description	
	Gain (dB)	Magnification
11000011	-18.0	0.125893
11000100	-17.5	0.133352
11000101	-17.0	0.141254
11000110	-16.5	0.149624
11000111	-16.0	0.158489
11001000	-15.5	0.167880
11001001	-15.0	0.177828
11001010	-14.5	0.188365
11001011	-14.0	0.199526
11001100	-13.5	0.211349
11001101	-13.0	0.223872
11001110	-12.5	0.237137
11001111	-12.0	0.251189
11010000	-11.5	0.266073
11010001	-11.0	0.281838
11010010	-10.5	0.298538
11010011	-10.0	0.316228
11010100	-9.5	0.334965
11010101	-9.0	0.354813
11010110	-8.5	0.375837
11010111	-8.0	0.398107
11011000	-7.5	0.421697
11011001	-7.0	0.446684
11011010	-6.5	0.473151
11011011	-6.0	0.501187
11011100	-5.5	0.530884
11011101	-5.0	0.562341
11011110	-4.5	0.595662
11011111	-4.0	0.630957
11100000	-3.5	0.668344
11100001	-3.0	0.707946
11100010	-2.5	0.749894
11100011	-2.0	0.794328
11100100	-1.5	0.841395
11100101	-1.0	0.891251
11100110	-0.5	0.944061
11100111	0.0	1.000000
11101000	0.5	1.059254
11101001	1.0	1.122018
11101010	1.5	1.188502

Bit[7:0]	Description	
	Gain (dB)	Magnification
11101011	2.0	1.258925
11101100	2.5	1.333521
11101101	3.0	1.412538
11101110	3.5	1.496236
11101111	4.0	1.584893
11110000	4.5	1.678804
11110001	5.0	1.778279
11110010	5.5	1.883649
11110011	6.0	1.995262
11110100	6.5	2.113489
11110101	7.0	2.238721
11110110	7.5	2.371374
11110111	8.0	2.511886
11111000	8.5	2.660725
11111001	9.0	2.818383
11111010	9.5	2.985383
11111011	10.0	3.162278
11111100	10.5	3.349654
11111101	11.0	3.548134
11111110	11.5	3.758374
11111111	12.0	3.981072

5.2. Fade In/Fade Out Time

Table 5-2

Bit[4:0]	Description
00000	Fade in/fade out effect not used.
00001	20 ms (960 sample)
00010	40 ms (1920 sample)
00011	60 ms (2880 sample)
00100	80 ms (3840 sample)
00101	100 ms (4800 sample)
00110	200 ms (9600 sample)
00111	300 ms (14400 sample)
01000	400 ms (19200 sample)
01001	500 ms (24000 sample)
01010	600 ms (28800 sample)
01011	700 ms (33600 sample)
01100	800 ms (38400 sample)
01101	900 ms (43200 sample)
01110	1000 ms (48000 sample)
01111	1100 ms (52800 sample)
10000	1200 ms (57600 sample)
10001	1300 ms (62400 sample)
10010	1400 ms (67200 sample)
10011	1500 ms (72000 sample)
10100	1600 ms (76800 sample)
10101	1700 ms (81600 sample)
10110	1800 ms (86400 sample)
10111	1900 ms (91200 sample)
11000	2000 ms (96000 sample)
11001	2000 ms (96000 sample)
11010	2000 ms (96000 sample)
11011	2000 ms (96000 sample)
11100	2000 ms (96000 sample)
11101	2000 ms (96000 sample)
11110	2000 ms (96000 sample)
11111	2000 ms (96000 sample)

5.3. Digital Filter

5type of digital filters are prepared and automatically selected and applied to change sampling rate. See 3.10.2 of the description of FIR digital filter.

No	Sampling frequency	Cutoff frequency	remark
1	44 kHz	17 kHz	Figure 31-2
2	24 kHz	11 kHz	Figure 5-3
3	12 kHz	5 kHz	Figure 5-4
4	8 kHz	3 kHz	Figure 5-5
5	4 kHz	1.2 kHz	Figure 5-6

Figure 31-2 Filter No 1

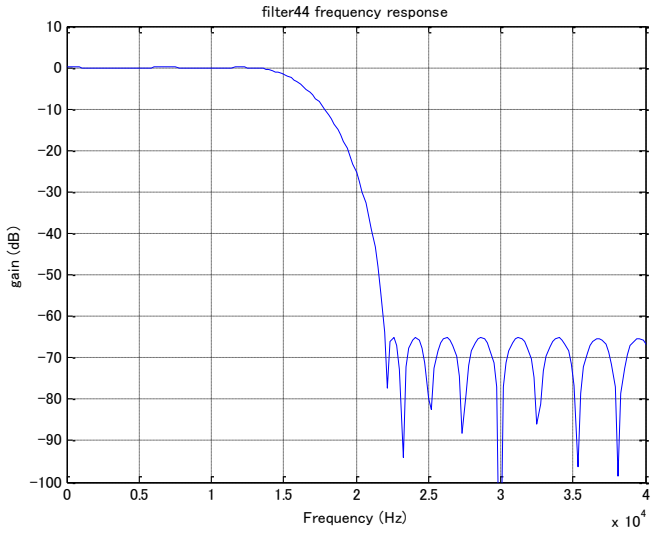


Figure 5-3 Filter No 2

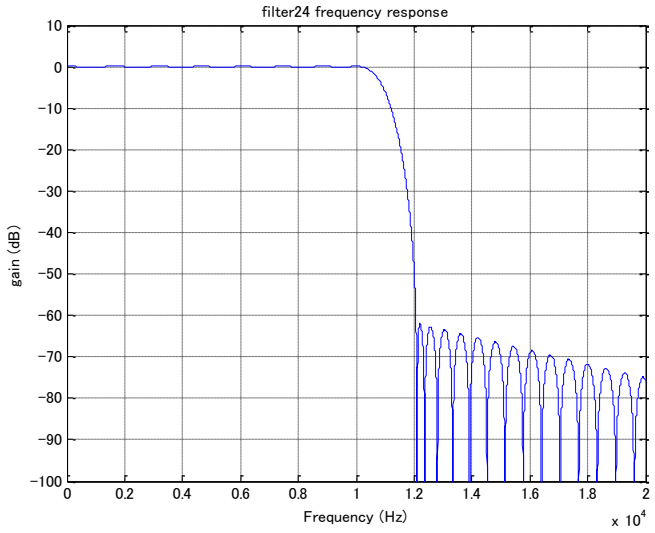


Figure 5-4 Filter No 3

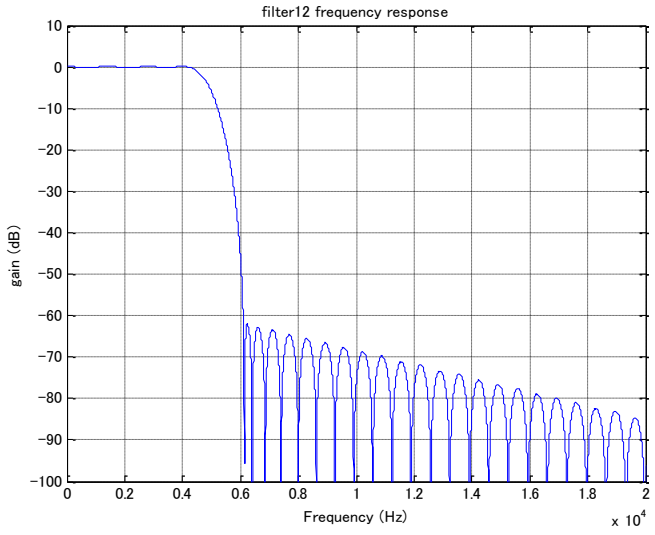


Figure 5-5 Filter No 4

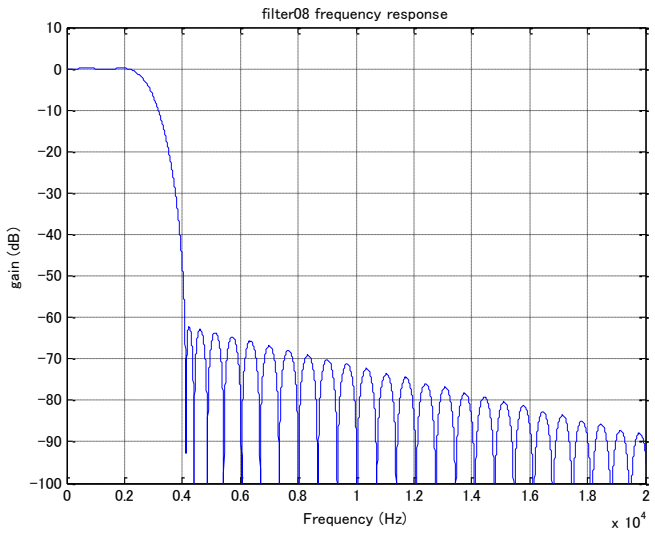
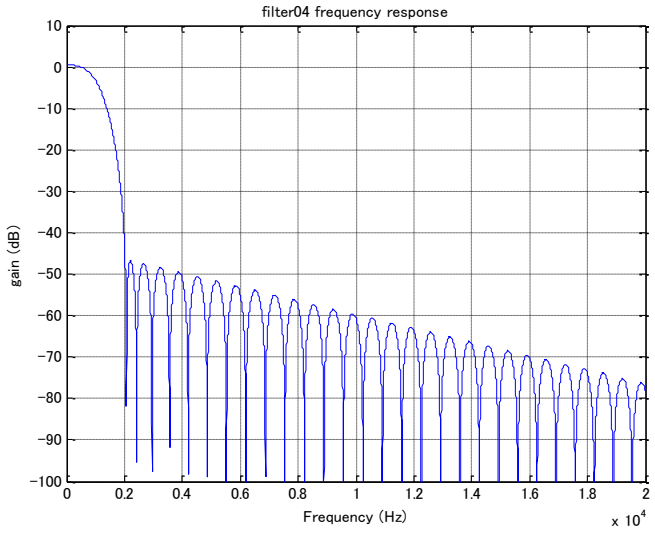


Figure 5-6 Filter No 5



CHAPTER 32: Inter-IC Sound (I2S)

This chapter explains the functions and operations of the serial audio interface that is the Inter IC Sound (I2S).

1. Overview
2. Configuration and Block Diagram
3. Operations of the I2S
4. Registers

CODE: I2S-S6J3200-E1

1. Overview

This section describes the features and the block diagram of the I2S module.

Also refer to the chapter of "Sound System Configuration" how to configure the I2S which is connected to the Sound Mixer.

I2S module is a full duplex, synchronous serial audio interface for multichannel specification. It can be configured to various frame formats by register setting.

This module can be set to operate as master and slave. In the master mode, clock (SCK) and frame synchronous signal (WS) are output to the external slave. In the slave mode, they are input from the external master.

During the master mode, SCK clock can be output by dividing external clock or internal bus clock (it is selectable by register). Frame synchronous signal can be generated by free-running or burst mode (generated only when there is transmission data).

This module has transmission/reception FIFOs, and their depths depend on the mode:

In transmission only mode, there is a 132-word 32-bit transmission FIFO. In reception only mode, there is a 132-word 32-bit reception FIFO. This module can also be configured in simultaneous mode. Simultaneous mode operates with a 66-word 32-bit transmission FIFO and a 66-word 32-bit reception FIFO.

Internal transfer between transmission and reception FIFO and internal system memory can be performed by DMA, interrupt, and polling.

Features of I2S

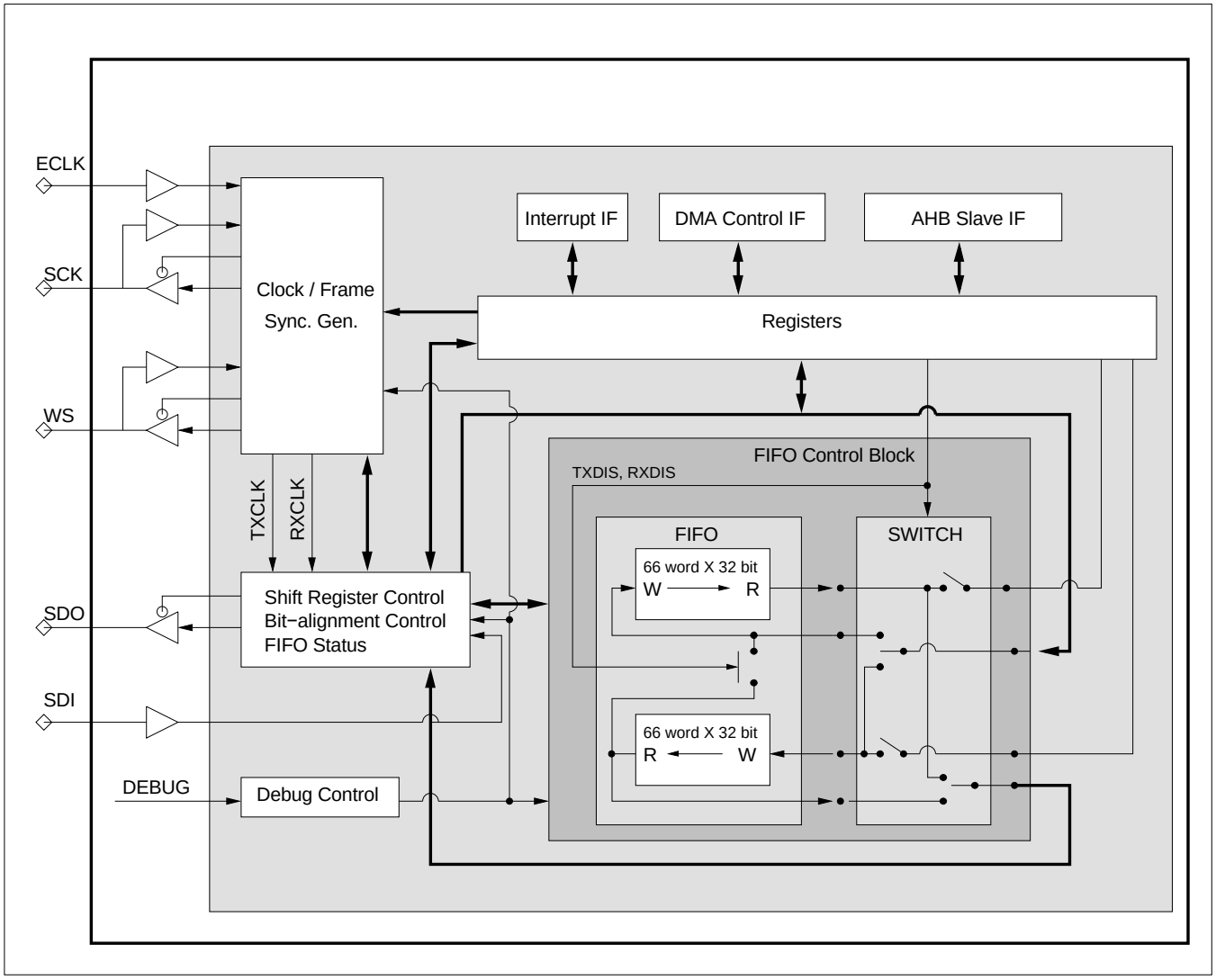
I2S interface has the following features:

- Programmable master/slave operations
- Support of transmission only, reception only and simultaneous transmission/reception modes
- Selecting 1 sub frame and 2 sub frame constructions
- Setting up to 32 channels to each sub frame
- Individually setting number of channel in each sub frame
- Individually setting channel length of each sub frame (number channel bit)
- Individually setting word length in channel of each sub frame
- Setting valid/invalid of each channel in each sub frame. Data is not sent or received to invalid channel
- Setting word length from 7 to 32 bits
- Programming frequency of frame synchronous signal
- Setting up to 3072 bits in 1 frame
- Programming width of frame synchronous signal (1 bit or 1 channel length)
- Programming phase of frame synchronous signal (0-bit or 1-bit delay)
- Setting polarity of frame synchronous signal
- Setting polarity of serial bit clock
- Programming sampling point of received data (center or at the end of received data)
- Selecting clock frequency source of serial bit clock in the master mode (internal bus clock and external clock)
- Setting clock frequency dividing ratio in the master mode
- Frequency of SCK = (frequency of internal bus clock or external clock) / (2 x I2Sn_CNTREG:CKRT[5:0])
- Frequency dividing ratio is settable within 0-126 in multiple of 2 (when the ratio is "0", frequency dividing source is bypassed)
- Data transfer to system memory by DMA, interrupt, and polling
- Debug support

2. Configuration and Block Diagram

This section describes the block diagram of I2S.

Figure 2-1 Block Diagram of I2S



- Four wire interface is used for full duplex data transfer (separate line for serial data input and serial data output)
- Frame and clock lines are bi-directional. These lines are output when module is configured in master mode and act as input line in slave mode
- DMA controller is used for DMA access

Clocking of I2S

- The supply clock of I2S can be internal bus clock or external (ECLK) clock source. For details refer to device specific datasheet. This clock is then pre-scaled to required frequency through I2S control register bits I2Sn_CNTREG:CKRT[5:0]
- Frame frequency can be adjusted using I2Sn_CNTREG:OVHD[9:0] bits

3. Operations of the I2S

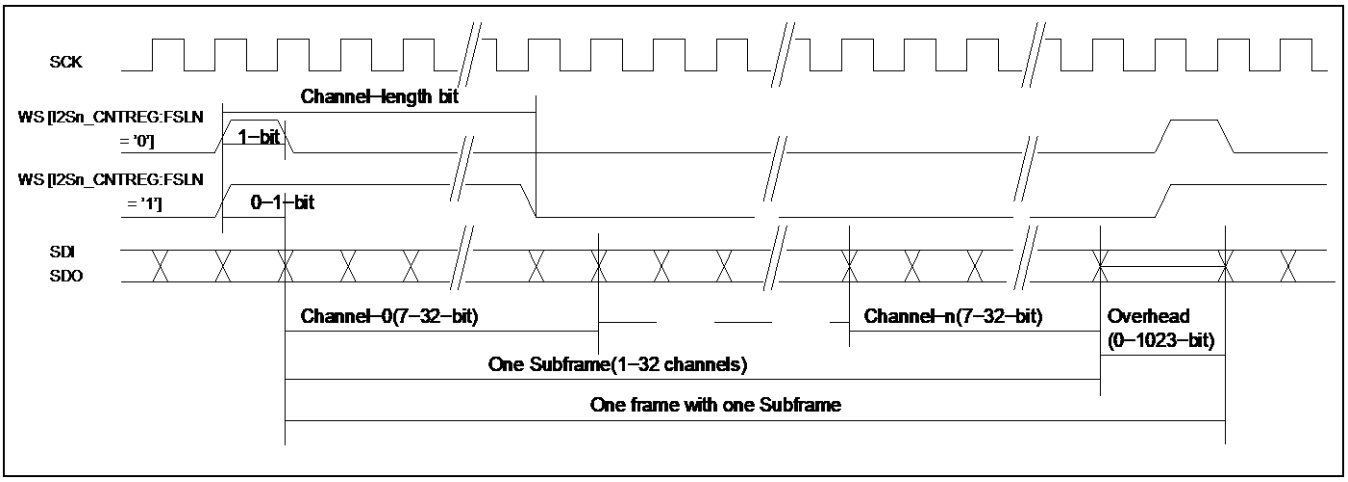
This section describes the operation of I2S.

3.1. I2S Frame Construction

I2S supports frame format of multiple channel construction. Frame can be configured to 1 or 2 sub frames. Number of each frame's channel and word length can be set individually.

1 Sub Frame Construction

Figure 3-1 1 Sub Frame Construction

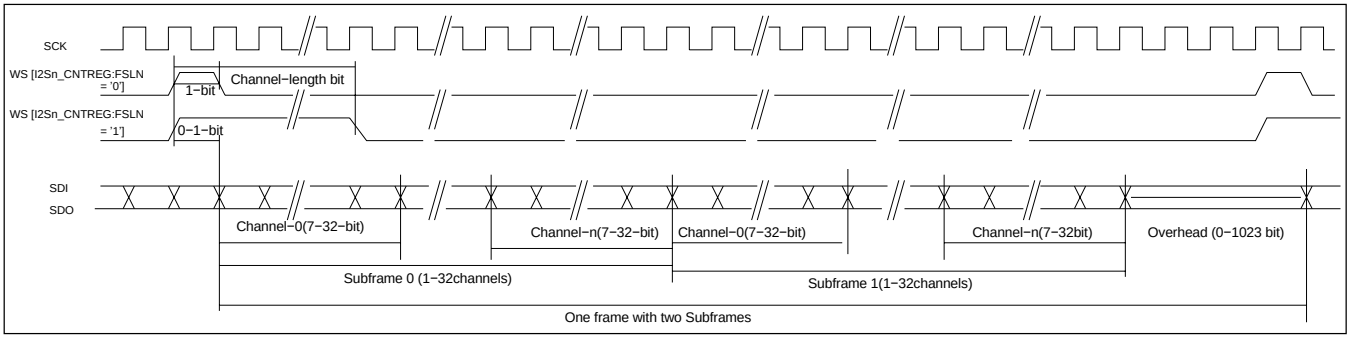


Description

- When I2Sn_CNTREG:SBFN bit is "0", frame becomes 1 sub frame composite
- Number of channels of 1-sub frame is determined by I2Sn_MCR0REG:S0CHN
- Up to 32 channels can be set
- Each channel bit length (word length) is determined by I2Sn_MCR0REG:S0WDL
- Sub frame channel starts from 0th. Each channel is set to valid/invalid with the corresponding bit of I2Sn_MCR1REG register. Transmission/reception of data is not performed to invalid channel
- Dummy bit can be inserted behind sub frame by setting I2Sn_CNTREG:OVHD. 0 to 1023 bits are insertable
- Polarity of WS is set with I2Sn_CNTREG:FSPL
- Pulse width of WS can be set to 1 bit or 1 channel length by setting I2Sn_CNTREG:FSLN
- Frame sync phase of WS can be set to "0" or "1" clock through I2Sn_CNTREG:FSPH
- In this construction, settings of I2Sn_MCR0REG:S1CHN and I2Sn_MCR2REG:S1WDL are ignored

2 Sub Frame Construction

Figure 3-2 2 Sub Frame Construction



Description

- When I2Sn_CNTREG:SBFN bit is "1", frame becomes 2 sub frame composition
- Set number of channel of sub frame 0 to I2Sn_MCR0REG:S0CHN, and set number of sub frame 1 channel to I2Sn_MCR0REG:S1CHN. Up to 32 channels can be set
- Channel bit length (word length) of sub frame 0 is determined by I2Sn_MCR0REG:S0WDL. For sub frame 1, they are determined by I2Sn_MCR0REG:S1WDL
- Sub frame channel starts from 0th. Each channel of sub frame 0 is set to valid/invalid with the corresponding bit of I2Sn_MCR1REG register, and corresponding bit of I2Sn_MCR2REG register for sub frame 1
- Transmission/reception of data is not performed to invalid channel
- Dummy bit can be inserted behind sub frame 1 by setting I2Sn_CNTREG:OVHD. 0 to 1023 bits are insertable
- Polarity of WS is set to I2Sn_CNTREG:FSPL bit
- Pulse width of WS can be set to 1 bit or 1 channel length by setting I2Sn_CNTREG:FSLN bit, channel length setting of 1 channel is determined by the channel length of sub frame "0"
- Frame sync phase of WS can be set to "0" or "1" clock through I2Sn_CNTREG:FSPH bit

3.2. I2S Configuration and Operation Modes

Transmission Only Mode

Table 3-1 Transmission Only Mode

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
Transmission only I2Sn_CNTREG: TXDIS = "0" I2Sn_CNTREG: RXDIS = "1"	Start	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** After I2Sn_OPRREG:START bit becomes "1" and I2Sn_OPRREG:TXENB bit is "1", frame synchronous signal starts to output when transmission FIFO is not empty. From the second time, frame synchronous signal is output with the frame rate determined by the register setting. If transmission FIFO is empty, empty frame is output at the same time of frame synchronous signal output. Serial data of the empty frame can be set to "0" or "1" by the register setting. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** If transmission FIFO is not empty, I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:TXENB bit is "1", frame synchronous signal is output. After completion of one frame output, transmission FIFO status is always confirmed. If transmission FIFO is not empty, frame synchronous signal is output to perform frame transmission.	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** The frame synchronous signal is input at the frame rate determined by the register setting. If transmission FIFO is empty when the frame synchronous signal is input when I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:TXENB bit is "1", empty frame is output. Serial data of the empty frame can be set to "0" or "1" by the register setting. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** When I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:TXENB bit is "1", one frame is output every time the frame synchronous signal is input. When transmission FIFO is empty at the time of frame synchronous signal input, empty frame is output.

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
Transmission only I2Sn_CNTREG: TXDIS = "0" I2Sn_CNTREG: RXDIS = "1"	Stop	At the time of stop, transmission FIFO becomes empty when there's no data transfer from internal memory to I2S transmission FIFO. **To maintain I2Sn_OPRREG:START bit to "1":** I2Sn_OPRREG:TXENB = "1": When "1" is written to I2Sn_OPRREG:TXENB, synchronous signal is output in the free-running mode. When transmission FIFO becomes empty, empty frame is output. In burst mode, frame synchronous signal is not output, and empty frame bits are output to serial data bus. I2Sn_OPRREG:TXENB = "0": When "0" is written to I2Sn_OPRREG:TXENB, transmission FIFO becomes empty. In the free-running mode, frame synchronous signal continues outputting and serial bus becomes high impedance state. In the burst mode, frame synchronous signal is not output and serial data bus becomes high impedance state. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to I2Sn_OPRREG:START bit, then transmission FIFO becomes empty. Clock supply to the serial control part is stopped regardless of I2Sn_OPRREG:TXENB setting. Serial Output Clock and Frame synchronous signal output is stopped. Serial data bus becomes high impedance state.	**To maintain I2Sn_OPRREG:START bit to "1":** I2Sn_OPRREG:TXENB = "1": Empty frame data is output to serial bus. I2Sn_OPRREG:TXENB = "0": When "0" is written to I2Sn_OPRREG:TXENB, transmission FIFO becomes empty, and data present in the transmission FIFO at the time "0" was written to I2Sn_OPRREG:TXENB is not transmitted. Writing to transmission FIFO and detection of the frame synchronous signal are stopped. Serial data bus becomes high impedance state. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to I2Sn_OPRREG:START bit, transmission FIFO becomes empty. Writing to transmission FIFO and detection of frame synchronous signal are stopped regardless of I2Sn_OPRREG:TXENB setting and serial bus becomes high impedance state.

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
	Abnormality	When reading from transmission FIFO occurs while it is empty, empty frame is output. For the setting conditions of I2Sn_STATUS:TXUDR0 and I2Sn_STATUS:TXUDR1, refer to their bit descriptions. When writing to transmission FIFO occurs while it is full, set I2Sn_STATUS:TXOVR to "1".	When reading from transmission FIFO occurs while it is empty, empty frame is output. For the setting conditions of I2Sn_STATUS:TXUDR0 and I2Sn_STATUS:TXUDR1, refer to their bit descriptions. However I2Sn_STATUS:TXUDR0/1 are not set to "1" for the 1st output frame after the bits become I2Sn_OPRREG:START = "1" and I2Sn_OPRREG:TXENB = "1". When writing to transmission FIFO occurs while it is full, I2Sn_STATUS:TXOVR is set to "1". If the frame synchronous signal is not input with the defined frame rate in the free-running mode, I2Sn_STATUS:FERR is set to "1". If the next frame synchronous signal is input before completing 1 frame transmission in the burst mode, I2Sn_STATUS:FERR is set to "1".

Reception Only Mode

Table 3-2 Reception Only Mode

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
Reception only I2Sn_CNTREG: TXDIS = "1" I2Sn_CNTREG: RXDIS = "0"	Start	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** Frame synchronous signal starts to output after I2Sn_OPRREG:START bit becomes "1" and I2Sn_OPRREG:RXENB bit is "1" when reception FIFO is not full. From the second time, frame synchronous signal with the frame rate determined by the register setting is output. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** When I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:RXENB bit is "1", frame synchronous signal is output to receive frame if reception FIFO is not full. If the FIFO is full, the signal is not output.	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** When I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:RXENB bit is "1", input frame synchronous signal with the frame rate determined by the register setting. Frame should be received every time the signal is input. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** When I2Sn_OPRREG:START bit is "1" and I2Sn_OPRREG:RXENB bit is "1", frame reception is performed every time frame synchronous signal is input. The signal is input with less speed than the frame rate in the free-running mode.
	Stop	At the time of stop, frame is not imported from serial bus even though reception FIFO is empty. **To maintain I2Sn_OPRREG:START bit to "1":** When "0" is written to I2Sn_OPRREG:RXENB, reception FIFO becomes empty. Although frame synchronous signal is kept outputting in the free-running mode, frame is not received. In the burst mode, frame is not received and the signal is not output. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to I2Sn_OPRREG:START, reception FIFO becomes empty. Clock supply to the serial control part stops regardless of I2Sn_OPRREG:RXENB setting, and SCK supply to the external part is stopped as well.	**To maintain I2Sn_OPRREG:START bit to "1":** Reception FIFO becomes empty by writing "0" to I2Sn_OPRREG:RXENB. The input frame synchronous signal is ignored, and frames are not received. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to the I2Sn_OPRREG:START bit, the reception FIFO becomes empty. The input frame synchronous signal is ignored regardless of I2Sn_OPRREG:RXENB setting, and frames are not received.
	Abnormality	When writing to reception FIFO occurs while it is full, I2Sn_STATUS:RXOVR is set to "1". I2Sn_STATUS:RXUDR bit is set to "1" when read access to reception FIFO occurs while it is empty.	When writing to reception FIFO occurs while it is full, I2Sn_STATUS:RXOVR is set to "1". When read access to reception FIFO occurs while it is empty, I2Sn_STATUS:RXUDR is set to "1". **Free-running mode:** If frame synchronous signal is not input with the frame rate defined by the register setting, I2Sn_STATUS:FERR bit is set to "1".

Simultaneous Transfer Mode

Table 3-3 Simultaneous Transfer Mode

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
Simultaneous Transfer I2Sn_CNTREG: TXDIS = "0" I2Sn_CNTREG: RXDIS = "0"	Start	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "0": The same operation as transmission only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "0", and I2Sn_OPRREG:RXENB = "1": The same operation as reception only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "1": Frame synchronous signal is output from the state that transmission FIFO is not empty and reception FIFO is not full. Then frame synchronous signal is output with the frame rate defined by the register setting. At the same time, empty frame is output if transmission FIFO is empty. Empty frame's serial data can be set to "0" or "1" by register setting. Every time frame synchronous signal is output, frame is received. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "0": The same operation as transmission only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "0", and I2Sn_OPRREG:RXENB = "1": The same operation as reception only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "1": Frame synchronous signal is output from the state that transmission FIFO is not empty and reception FIFO is not full. After completion of one frame output or at idle state, transmission/reception FIFO status is always confirmed. If transmission FIFO is not empty and reception FIFO is not full, frame synchronous signal is output to perform frame transmission/reception.	**Free-running mode (I2Sn_CNTREG:FRUN = "1"):** I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "0": The same operation as transmission only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "0", and I2Sn_OPRREG:RXENB = "1": The same operation as reception only mode. I2Sn_OPRREG:START = "1", I2Sn_OPRREG:TXENB = "1", and I2Sn_OPRREG:RXENB = "1": Frame synchronous signal is input with the frame rate defined by the register setting. At the same time, empty frame is output if transmission FIFO is empty. The serial data can be set to "0" or "1" by the register setting I2Sn_CNTREG:MSKB. Every time frame synchronous signal is input, frame is received. **Burst mode (I2Sn_CNTREG:FRUN = "0"):** Every time frame synchronous signal is input when I2Sn_OPRREG:START bit is "1", transmission and reception for one frame is performed. When the frame synchronous signal is input, empty frame is output if transmission FIFO is empty.

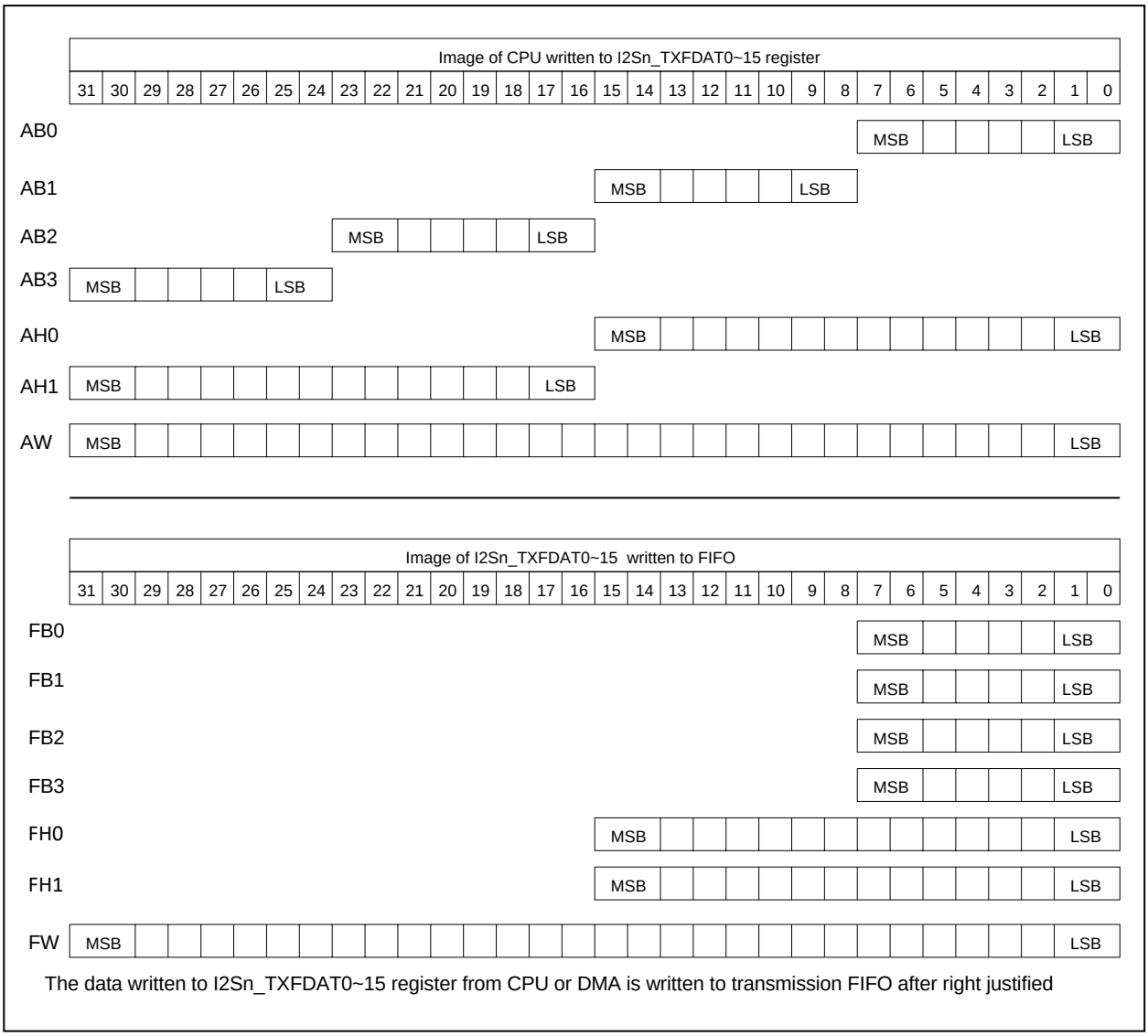
Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
Simultaneous Transfer I2Sn_CNTREG: TXDIS = "0" I2Sn_CNTREG: RXDIS = "0"	Stop	Stop operation has following states: Transmission stop: Transmission FIFO becomes empty when data is not transferred from internal memory to I2S transmission FIFO. Reception stop: Data does not need to be transferred from I2S reception FIFO to internal memory. **To maintain I2Sn_OPRREG:START bit to "1":** In free-running mode frame synchronous signal is output. In the burst mode, when transmission FIFO becomes empty, frame synchronous output is stopped. Transmission stop: I2Sn_OPRREG:TXENB = "1": Empty frame bit is output when transmission FIFO becomes empty. I2Sn_OPRREG:TXENB = "0": Transmission FIFO becomes empty and transmission serial data bus becomes high impedance. Writing to transmission FIFO stops. Reception stop: When "0" is written to I2Sn_OPRREG:RXENB, reception FIFO becomes empty and frame reception operation stops. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to I2Sn_OPRREG:START, transmission/reception FIFO becomes empty. The clock supply to the internal serial control part stops regardless of I2Sn_OPRREG:TXENB and I2Sn_OPRREG:RXENB status. SCK output to the external part and frame synchronous signal output is also stopped.	**To maintain I2Sn_OPRREG:START bit to "1":** Transmission stop: When I2Sn_OPRREG:TXENB = "1", empty frame bits are output after transmission FIFO becomes empty. When "0" is written to I2Sn_OPRREG:TXENB, transmission FIFO becomes empty and transmission serial data bus becomes high impedance. Data present in the transmission FIFO at the time "0" was written to I2Sn_OPRREG:TXENB is not transmitted. While I2Sn_OPRREG:TXENB = "0", data is not written to transmission FIFO. Reception stop: When "0" is written to I2Sn_OPRREG:RXENB, reception FIFO becomes empty and frame reception operation stops. **To make I2Sn_OPRREG:START bit "0":** When "0" is written to I2Sn_OPRREG:START, transmission/reception FIFO becomes empty. Transmission/reception is stopped regardless of I2Sn_OPRREG:TXENB and I2Sn_OPRREG:RXENB status.

Transfer Setting	Operation	Master Mode (I2Sn_CNTREG:MSMD = "1")	Slave Mode (I2Sn_CNTREG:MSMD = "0")
	Abnormality	When reading from transmission FIFO occurs while it is empty, empty frame bit is output. For the setting conditions of I2Sn_STATUS:TXUDR0 and I2Sn_STATUS:TXUDR1, refer to their bit descriptions. When writing to transmission FIFO occurs while it is full, I2Sn_STATUS:TXOVR is set to "1". When read access occurs to reception FIFO while it is empty, I2Sn_STATUS:RXUDR is set to "1". When writing to reception FIFO occurs while it is full, I2Sn_STATUS:RXOVR is set to "1".	When reading from transmission FIFO occurs while it is empty, empty frame bit is output. For the setting conditions of I2Sn_STATUS:TXUDR0 and I2Sn_STATUS:TXUDR1, refer to their bit descriptions. When writing to transmission FIFO occurs while it is full, I2Sn_STATUS:TXOVR is set to "1". When read access occurs to reception FIFO while it is empty, I2Sn_STATUS:RXUDR is set to "1". When writing to reception FIFO occurs while it is full, I2Sn_STATUS:RXOVR is set to "1". If the frame synchronous signal is not input with the defined frame rate in the free-running mode, I2Sn_STATUS:FERR is set to "1". If the following frame synchronous signal is input before completing one frame transmission in the burst mode, I2Sn_STATUS:FERR is set to "1".

3.3. Bit Alignment

Transmission Word Alignment

Figure 3-3 Transmission Word Line Chart



When transmission is performed with serial bus, word is sent MSB first when I2Sn_CNTREG:MSLB is "0" and LSB first when the value is "1". When channel length (set to I2Sn_MCR0REG:S0CHL and I2Sn_MCR0REG:S1CHL) is longer than the word length (set to I2Sn_MCR0REG:S0WDL and I2Sn_MCR0REG:S1WDL), remaining bits in the channel become I2Sn_CNTREG:MSKB. Setting the channel length to shorter than the word length is prohibited.

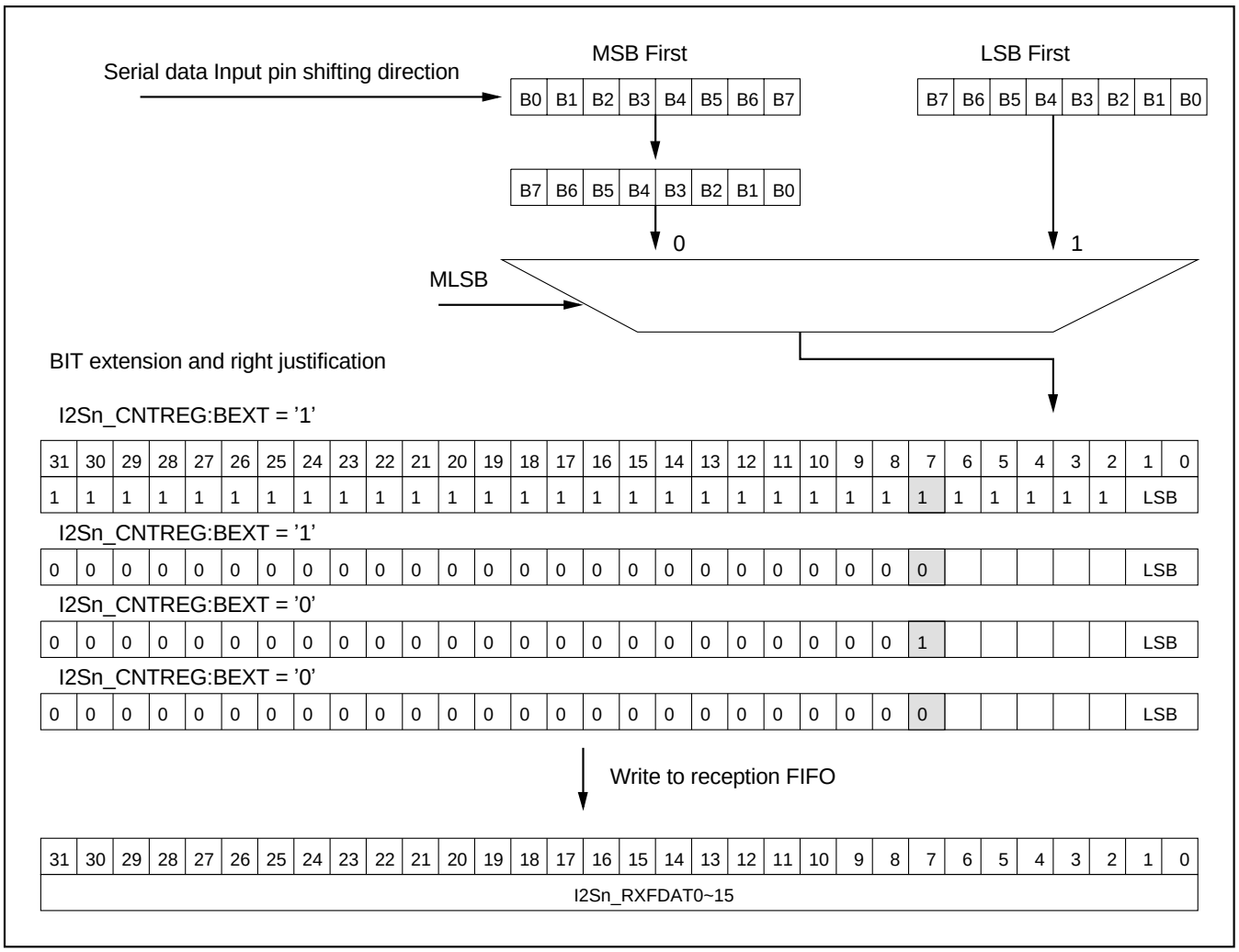
Notes:

- *AB0, AB1, AB2, AB3, AH0, AH1, and AW on the above chart indicate byte 0, byte 1, byte 2, byte 3, half word 0, half word 1, and word at write accessing to I2Sn_TXFDAT0 to 15 on AHB bus*

- *Each FB0, FB1, FB2, FB3, FH0, FH1, and FW indicate AB0, AB1, AB2, AB3, AH0, AH1, and AW at writing to transmission FIFO after they are right justified*

Reception Word Alignment

Figure 3-4 Reception Word Line Chart



This chart shows word line example of when word length is 8.

The word received from serial bus is always written to reception FIFO after being right justified.

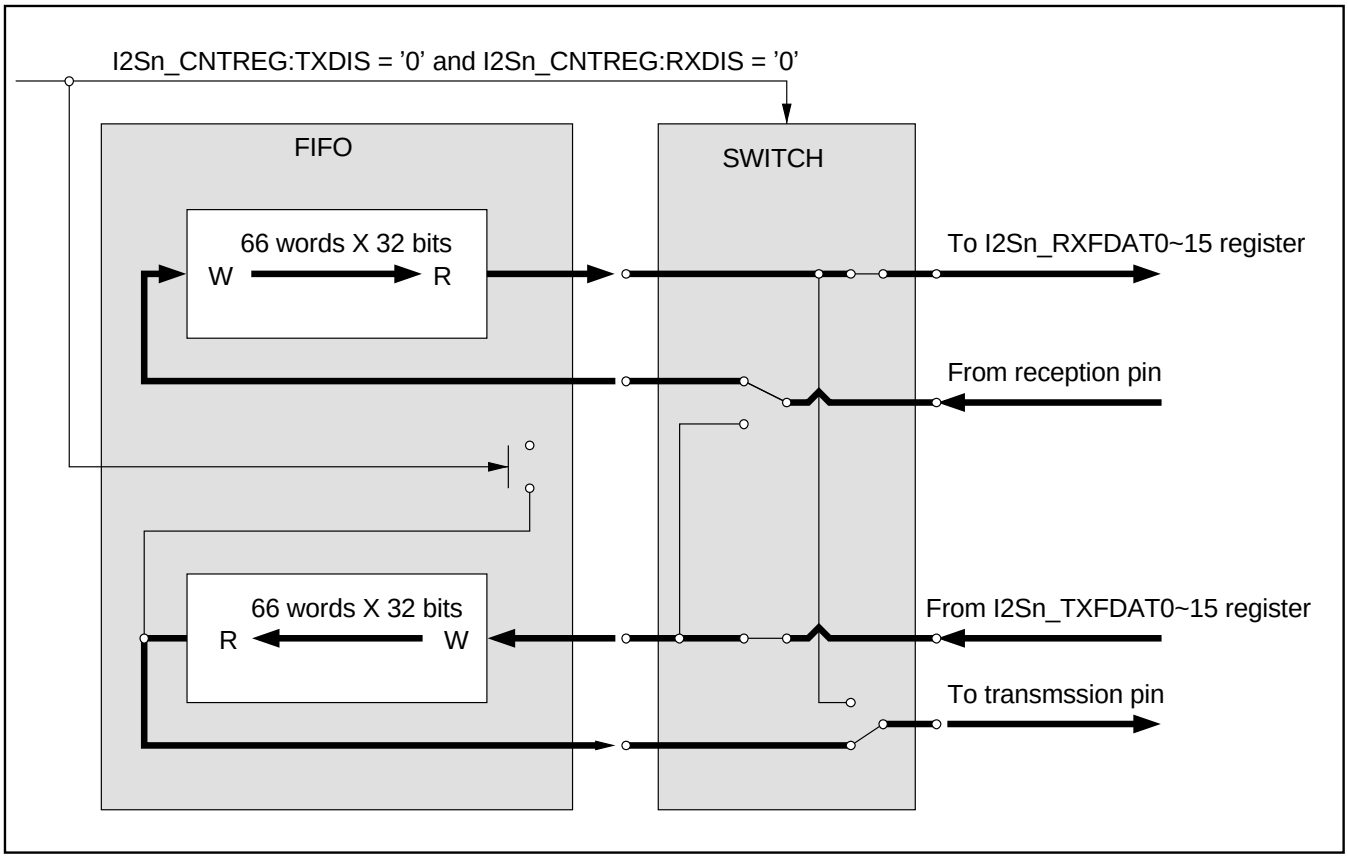
Therefore, read access should be performed from AHB bus to I2Sn_RXFDAT0 to 15 in order to read as follows:

Word length:

- 8 or less: byte 0
- 9- 16: half word 0
- 17-32: all words

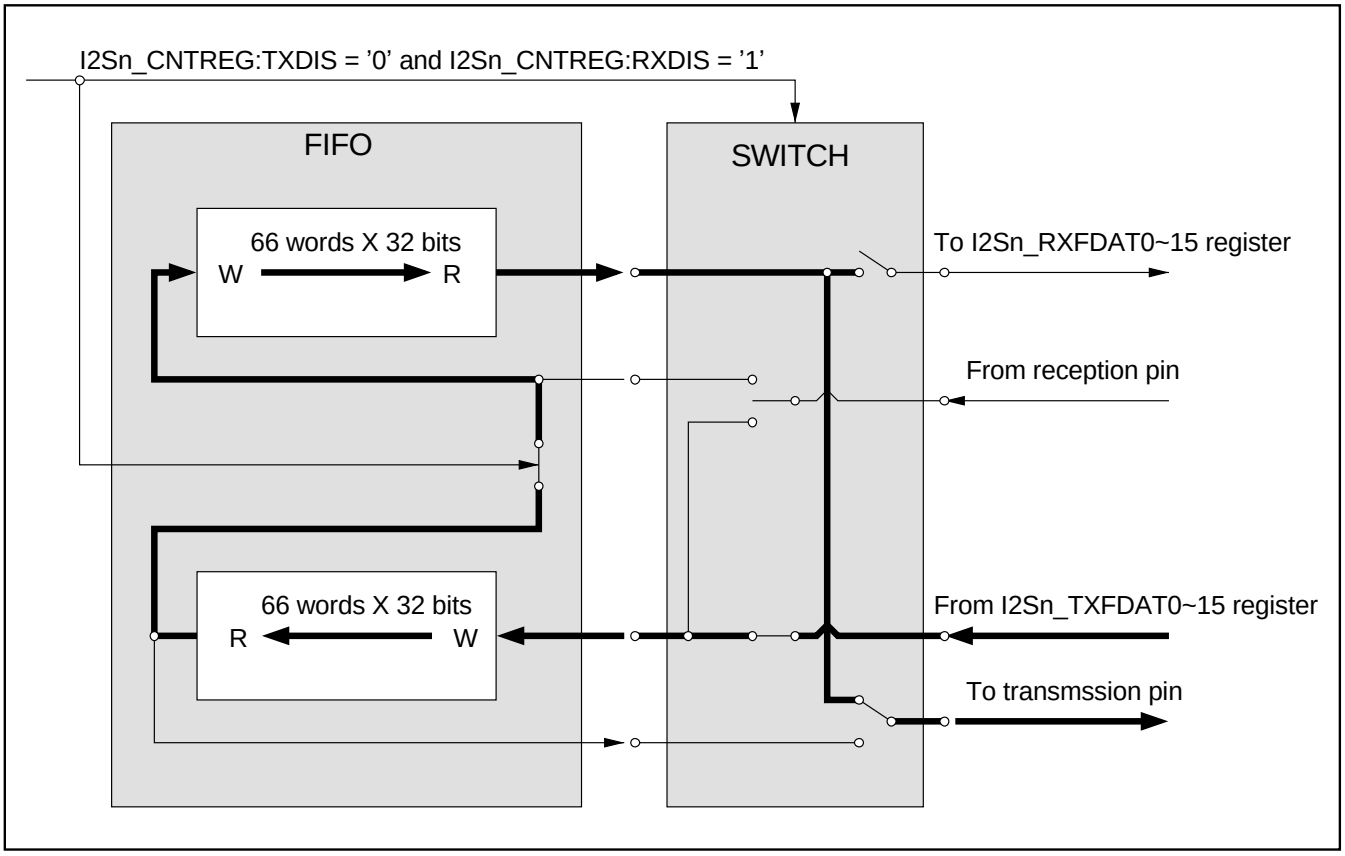
3.4. FIFO Construction

Figure 3-5 Simultaneous Transfer Mode Data Flow



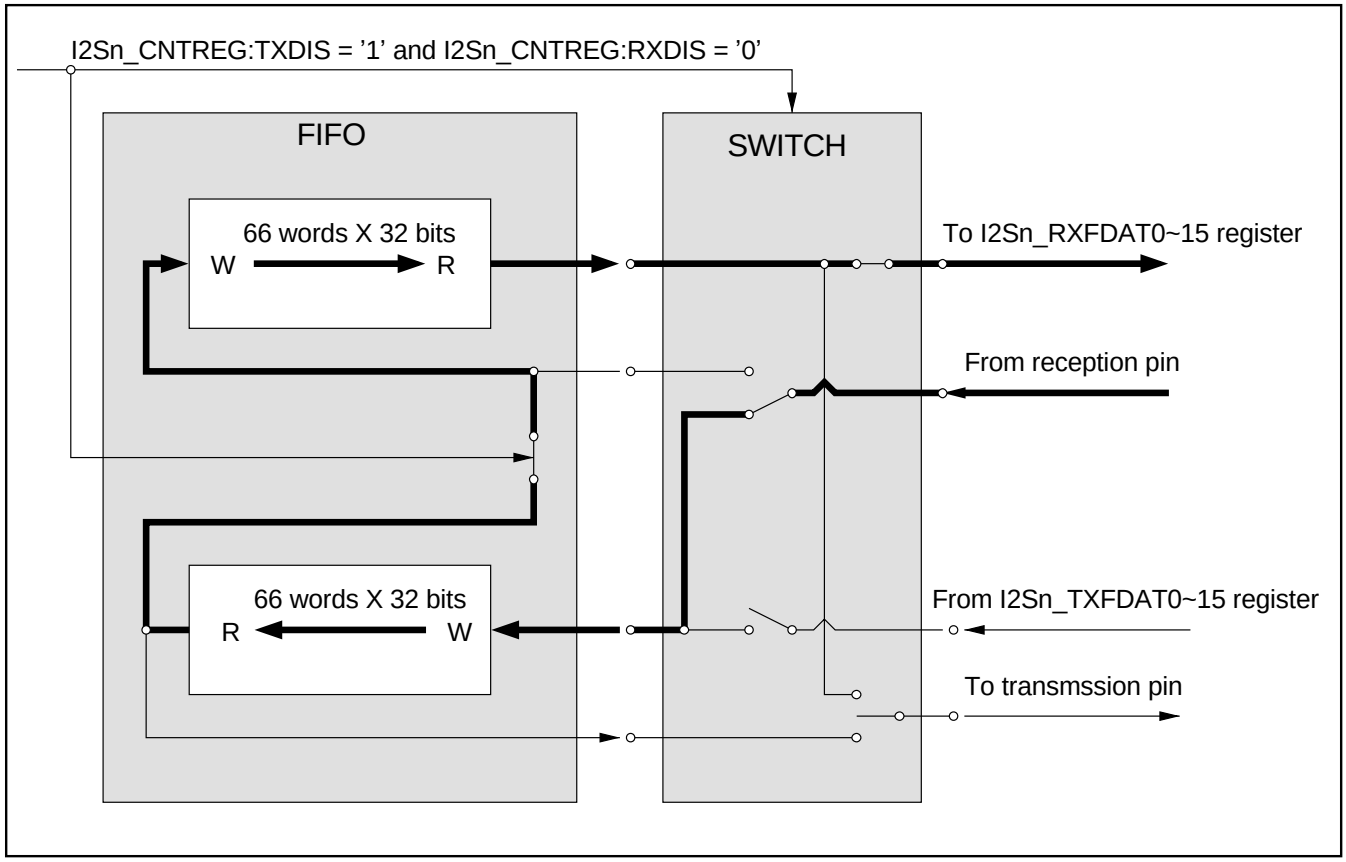
When I2Sn_CNTREG:TXDIS = "0" and I2Sn_CNTREG:RXDIS = "0", the mode is set to simultaneous transfer mode which operates with 66-word x 32-bit transmission and reception FIFOs.

Figure 3-6 Transmission Only Mode Data Flow



When I2Sn_CNTREG:TXDIS = "0" and I2Sn_CNTREG:RXDIS = "1", the mode is set to transmission only mode which operates with a 132-word x 32-bit transmission FIFO, and reception is not performed.

Figure 3-7 Reception Only Mode Data Flow



When I2Sn_CNTREG:TXDIS = "1" and I2Sn_CNTREG:RXDIS = "0", the mode is set to reception only mode which operates with a 132-word x 32-bit reception FIFO, and transmission is not performed.

3.5. Caution Summary

- I2Sn_MCR0REG:S0WDL, I2Sn_MCR0REG:S0CHL 1 to 6 bits are prohibited
- I2Sn_MCR0REG:S1WDL, I2Sn_MCR0REG:S1CHL 1 to 6 bits are prohibited
- When channel length (set to I2Sn_MCR0REG:S0CHL and I2Sn_MCR0REG:S1CHL) is longer than the word length (set to I2Sn_MCR0REG:S0WDL and I2Sn_MCR0REG:S1WDL), remaining bits in the channel become I2Sn_CNTREG:MSKB. Setting the channel length to shorter than the word length is prohibited
- Pulse width of one channel (I2Sn_CNTREG:FSLN = "1") is prohibited when frame length is set to one channel by I2Sn_MCR0REG:S0CHN ="0" and I2Sn_CNTREG:SBFN = "0"
- Rewrite to I2Sn_CNTREG, I2Sn_MCR0REG, I2Sn_MCR1REG, and I2Sn_MCR2REG is prohibited after I2Sn_OPRREG:START is set
- Rewrite to I2Sn_CNTREG, I2Sn_MCR0REG, I2Sn_MCR1REG, and I2Sn_MCR2REG is prohibited while DBGE is set to "1" and the processor is in debug state

4. Registers

This section describes the registers of the I2S in detail.

The suffix 'n' in the register name indicates that the register is in instance 'n' of the module.

Registers of I2S

The following registers are available for each instance of I2S:

- Reception FIFO Data Register (I2Sn_RXFDAT0 to 15)
- Transmission FIFO Data Register (I2Sn_TXFDAT0 to 15)
- Control Register (I2Sn_CNTREG)
- Channel Control Register 0 (I2Sn_MCR0REG)
- Channel Control Register 1 (I2Sn_MCR1REG)
- Channel Control Register 2 (I2Sn_MCR2REG)
- Operation Control Register (I2Sn_OPRREG)
- Software Reset Register (I2Sn_SRST)
- Interrupt Control Register (I2Sn_INTCNT)
- Status Register (I2Sn_STATUS)
- DMA Activate Register (I2Sn_DMAACT)
- Debug Register (I2Sn_DEBUG)
- Module ID Register (I2Sn_MIDREG)

Memory Layout of I2S Registers

Offset	+3	+2	+1	+0
0x00000000	I2Sn_RXFDAT0 00000000 00000000 00000000 00000000			
0x00000004	I2Sn_RXFDAT1 00000000 00000000 00000000 00000000			
0x00000008	I2Sn_RXFDAT2 00000000 00000000 00000000 00000000			
0x0000000C	I2Sn_RXFDAT3 00000000 00000000 00000000 00000000			
0x00000010	I2Sn_RXFDAT4 00000000 00000000 00000000 00000000			
0x00000014	I2Sn_RXFDAT5 00000000 00000000 00000000 00000000			
0x00000018	I2Sn_RXFDAT6 00000000 00000000 00000000 00000000			
0x0000001C	I2Sn_RXFDAT7 00000000 00000000 00000000 00000000			
0x00000020	I2Sn_RXFDAT8 00000000 00000000 00000000 00000000			

Offset	+3	+2	+1	+0
0x00000024	I2Sn_RXFDAT9 00000000 00000000 00000000 00000000			
0x00000028	I2Sn_RXFDAT10 00000000 00000000 00000000 00000000			
0x0000002C	I2Sn_RXFDAT11 00000000 00000000 00000000 00000000			
0x00000030	I2Sn_RXFDAT12 00000000 00000000 00000000 00000000			
0x00000034	I2Sn_RXFDAT13 00000000 00000000 00000000 00000000			
0x00000038	I2Sn_RXFDAT14 00000000 00000000 00000000 00000000			
0x0000003C	I2Sn_RXFDAT15 00000000 00000000 00000000 00000000			
0x00000040	I2Sn_TXFDAT0 00000000 00000000 00000000 00000000			
0x00000044	I2Sn_TXFDAT1 00000000 00000000 00000000 00000000			
0x00000048	I2Sn_TXFDAT2 00000000 00000000 00000000 00000000			
0x0000004C	I2Sn_TXFDAT3 00000000 00000000 00000000 00000000			
0x00000050	I2Sn_TXFDAT4 00000000 00000000 00000000 00000000			
0x00000054	I2Sn_TXFDAT5 00000000 00000000 00000000 00000000			
0x00000058	I2Sn_TXFDAT6 00000000 00000000 00000000 00000000			
0x0000005C	I2Sn_TXFDAT7 00000000 00000000 00000000 00000000			
0x00000060	I2Sn_TXFDAT8 00000000 00000000 00000000 00000000			
0x00000064	I2Sn_TXFDAT9 00000000 00000000 00000000 00000000			
0x00000068	I2Sn_TXFDAT10 00000000 00000000 00000000 00000000			
0x0000006C	I2Sn_TXFDAT11 00000000 00000000 00000000 00000000			
0x00000070	I2Sn_TXFDAT12 00000000 00000000 00000000 00000000			
0x00000074	I2Sn_TXFDAT13 00000000 00000000 00000000 00000000			
0x00000078	I2Sn_TXFDAT14 00000000 00000000 00000000 00000000			

Offset	+3	+2	+1	+0
0x0000007C	I2Sn_TXFDAT15 00000000 00000000 00000000 00000000			
0x00000080	I2Sn_CNTREG 00000000 00000000 00000000 01100000			
0x00000084	I2Sn_MCR0REG 00000000 00000000 00000000 00000000			
0x00000088	I2Sn_MCR1REG 00000000 00000000 00000000 00000000			
0x0000008C	I2Sn_MCR2REG 00000000 00000000 00000000 00000000			
0x00000090	I2Sn_OPRREG 00000000 00000000 00000000 00000000			
0x00000094	I2Sn_SRST 00000000 00000000 00000000 00000000			
0x00000098	I2Sn_INTCNT 01111111 00111111 00000000 00000000			
0x0000009C	I2Sn_STATUS 00000000 00000000 00000000 00000000			
0x000000A0	I2Sn_DMAACT 00000000 00000000 00000000 00000000			
0x000000A4	I2Sn_DEBUG 00000000 00000000 00000000 00000000			
0x000000A8	I2Sn_MIDREG 00000000 00000000 00000000 00000000			
0x000000AC	reserved XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

4.1. Reception FIFO Data Register (I2Sn_RXFDAT0 to 15)

These registers are reception FIFO registers that can maintain up to 66 words (simultaneous transfer mode) or 132 words (reception only mode). There are 16 such registers, all consecutively placed in the register map. This is to support AHB burst transfers. Read access to any of the I2Sn_RXFDAT0 to 15 registers returns the word from reception FIFO. Each of these 16 registers are identical in function. Only one register (i.e. I2Sn_RXFDAT0) is explained here.

Reception FIFO Data Register 0 (I2Sn_RXFDAT0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	RXDATA[31]	RXDATA[30]	RXDATA[29]	RXDATA[28]	RXDATA[27]	RXDATA[26]	RXDATA[25]	RXDATA[24]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	RXDATA[23]	RXDATA[22]	RXDATA[21]	RXDATA[20]	RXDATA[19]	RXDATA[18]	RXDATA[17]	RXDATA[16]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	RXDATA[15]	RXDATA[14]	RXDATA[13]	RXDATA[12]	RXDATA[11]	RXDATA[10]	RXDATA[9]	RXDATA[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RXDATA[7]	RXDATA[6]	RXDATA[5]	RXDATA[4]	RXDATA[3]	RXDATA[2]	RXDATA[1]	RXDATA[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] RXDATA : Receive Data

The word received from serial bus is written to reception FIFO.

When frame is 1 sub frame construction and word length set to I2Sn_MCR0REG:S0WDL is 32 bits or less (16 bits when I2Sn_CNTREG:RHLL register is "1"), it is written to reception FIFO after higher order bit is extended.

When frame is 2 sub frame construction and word length set to I2Sn_MCR0REG:S0WDL is 32 bits or less (16 bits when I2Sn_CNTREG:RHLL register is "1"), reception data of sub frame "0" is written to reception FIFO after higher order bit is extended.

For the case that word length set to I2Sn_MCR0REG:S1WDL is 32 bits or less, reception data of sub frame 1 is written to reception FIFO after higher order bit is extended.

When I2Sn_CNTREG:BEXT is "1", it is extended with MSB of reception word (sign extension). For the case that the value is "0", it is enhanced by "0".

Reading this register returns a word of data from the Rx FIFO location pointed by the Rx FIFO read pointer. After a read access to this register, the Rx FIFO read pointer is incremented, provided that the read cycle was initiated by the AHB master other than the Debug Access Port (DAP) controller. If the DAP controller reads this register, the Rx FIFO read pointer is not incremented.

When I2Sn_STATUS:RXNUM is 0x00000000, invalid data is read. An Rx FIFO underflow error (I2Sn_STATUS:RXUDR) flag is set, if the read cycle was initiated by the AHB master other than the DAP controller. If the DAP controller reads this register while the Rx FIFO is empty, the I2Sn_STATUS:RXUDR flag is not set.

4.2. Transmission FIFO Data Register (I2Sn_TXFDAT0 to 15)

These registers are transmission FIFO registers that can maintain up to 66 words (simultaneous transfer mode) or 132 words (transmission mode only). There are 16 such registers, all consecutively placed in the register map. This is to support AHB burst transfers. A write access to any of I2Sn_TXFDAT0 to 15 register writes a word of data in the transmission FIFO. Each of these 16 registers are identical in function. Only one register (i.e. I2Sn_TXFDAT0) is explained here.

Transmission FIFO Data Register 0 (I2Sn_TXFDAT0)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TXDATA[31]	TXDATA[30]	TXDATA[29]	TXDATA[28]	TXDATA[27]	TXDATA[26]	TXDATA[25]	TXDATA[24]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	TXDATA[23]	TXDATA[22]	TXDATA[21]	TXDATA[20]	TXDATA[19]	TXDATA[18]	TXDATA[17]	TXDATA[16]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TXDATA[15]	TXDATA[14]	TXDATA[13]	TXDATA[12]	TXDATA[11]	TXDATA[10]	TXDATA[9]	TXDATA[8]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	TXDATA[7]	TXDATA[6]	TXDATA[5]	TXDATA[4]	TXDATA[3]	TXDATA[2]	TXDATA[1]	TXDATA[0]
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] TXDATA : Transmit Data

Word to be transmitted can be written as long as transmission FIFO is not full.

Write access can be performed regardless of shift register's operation status. Write access to full transmission FIFO is ignored and I2Sn_STATUS:TXOVR flag is set. Although writing data is accessed in word, half-word, and byte access, actual number of bits to be transmitted is determined by I2Sn_MCR0REG:S0WDL and I2Sn_MCR0REG:S1WDL (when frame is 2 sub frame).

The data read from I2Sn_TXFDAT0 returns "0".

4.3. Control Register (I2Sn_CNTREG)

The Control Register (I2Sn_CNTREG) sets the module configuration.

Control Register (I2Sn_CNTREG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	CKRT[5]	CKRT[4]	CKRT[3]	CKRT[2]	CKRT[1]	CKRT[0]	OVHD[9]	OVHD[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	OVHD[7]	OVHD[6]	OVHD[5]	OVHD[4]	OVHD[3]	OVHD[2]	OVHD[1]	OVHD[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	MSKB	MSMD	SBFN	RHLL	ECKM	BEXT	FRUN
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	MSLB	TXDIS	RXDIS	SMPL	CPOL	FSPH	FSLN	FSPL
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	1	1	0	0	0	0	0

[bit31:26] CKRT : Clock Divider

This sets output clock frequency dividing ratio at master operation.

Internal bus clock is divided at I2Sn_CNTREG:ECKM = "0", and external clock (ECLK) is divided at I2Sn_CNTREG:ECKM = "1". Only even number of the ratio is supported and output clock's duty becomes 50%.

Bit	Description
000000	Output clock is not divided
000001	Output clock is divided by 2
000010	Output clock is divided by 4
000011	Output clock is divided by 6
	...
	...
111110	Output clock is divided by 124
111111	Output clock is divided by 126

[bit25:16] OVHD : Frame Rate Control

Frame rate can be adjusted by inserting overhead (OVHD) bits following the valid data of the frame. The overhead section of the transmission frame enters the state of high impedance. Up to 1023 overhead bits can be inserted at the end of the frame.

The value set to OVHD becomes the number of insertion bits.

The following expressions are formed for OVHD and frame synchronous signal cycle.

1 sub frame construction:

- OVHD = Frame synchronous signal cycle/SCK cycle - (I2Sn_MCR0REG:S0CHL + 1) * (I2Sn_MCR0REG:S0CHN + 1)

2 sub frame construction:

- OVHD = Frame synchronous signal cycle/SCK cycle - (I2Sn_MCR0REG:S0CHL + 1) * (I2Sn_MCR0REG:S0CHN + 1) - (I2Sn_MCR0REG:S1CHL + 1) * (I2Sn_MCR0REG:S1CHN + 1)

[bit15] read0 : -

[bit14] MSKB : Serial Output Data in case of Invalid/Empty Frame Transmission

For master operation (I2Sn_CNTREG:MSMD = "1"), free-running mode (I2Sn_CNTREG:FRUN = "0"), and I2Sn_OPRREG:TXENB = "1":

- When transmission FIFO is empty at frame synchronous signal output, MSKB is output to all valid channels of its transmission frame.

For slave operation (I2Sn_CNTREG:MSMD = "0") and I2Sn_OPRREG:TXENB = "1":

- When transmission FIFO is empty at frame synchronous signal reception, MSKB is output to all valid channels of its transmission frame.

For the case that transmission word length is shorter than the channel length, MSKB is driven to the rest of bit in transmission channel (channel length - word length).

[bit13] MSMD : Master and Slave Mode Select

Master and Slave modes are set.

Bit	Description
0	Slave operation
1	Master operation

[bit12] SBFN : Sub Frame Construction

Sub frame construction (number of sub frame) of the frame is specified.

Bit	Description
0	1 Sub frame construction (only sub frame 0)
1	2 Sub frame construction (sub frame 0 and sub frame 1). Frame starts from 0th sub frame

[bit11] RHLL : Word Construction

Word construction of FIFO – 1 word (32 bits) or 2 half words (16 bits) – is specified.

It is considered to be used at protocol, such as MSB-justified.

Bit	Description
0	32-bit FIFO word is handled as 1 word
1	32-bit FIFO word is handled as 2 half words at serial bus with dividing 16 bits each to low order and high order. They are transferred by serial bus in order of low order, high order, low order, and high order. At reception, 2 consecutive half words from serial bus are handled as low order and high order, and they are put in 1 word (32 bits) to write to reception FIFO

[bit10] ECKM : Clock Selector

Clock frequency dividing is selected in master mode.

Bit	Description
0	Internal bus clock is divided and output
1	External clock (ECLK)is divided and output

[bit9] BEXT : Bit Extension

When reception word length is shorter than the word length of FIFO (32 bits when RHLL is "0" and 16 bits when RHLL is "1"), extension mode of upper bit (word length of FIFO-reception word length) should be set.

Bit	Description
0	Extended by "0"
1	Extended by sign bit (if MSB of word/half word is "1", then it is extended by "1", if MSB is "0" then it is extended by "0")

[bit8] FRUN : Output Mode of Frame Synchronous Signal

Bit	Description
0	Burst mode When I2Sn_OPRREG:START bit is "1", frame synchronous signal is output according to I2Sn_OPRREG:TXENB, I2Sn_OPRREG:RXENB, and transmission/reception FIFO conditions. When I2Sn_OPRREG:START bit is "0", frame synchronous signal is not output.
1	Free-running mode When I2Sn_OPRREG:START register is "1", frame synchronous signal proceeds free-running with the set frame rate. When I2Sn_OPRREG:START bit is "0", frame synchronous signal is not output.

[bit7] MSLB : Shifting Order

Word bit's shift order is set.

Bit	Description
0	Shift starts from MSB of the word
1	Shift starts from LSB of the word

[bit6] TXDIS : Transmitter Disable

Transmitting function is enabled or disabled.

Bit	Description
0	Transmitting function is enabled
1	Transmitting function is disabled

[bit5] RXDIS : Receiver Disable

Receiving function is enabled or disabled.

Bit	Description
0	Receiving function is enabled
1	Receiving function is disabled

[bit4] SMPL : Sampling Point

Sampling point of the data is specified.

Bit	Description
0	Sampling at the center of reception data
1	Sampling at the end of reception data

[bit3] CPOL : Clock Polarity

SCK polarity which drives/samples serial data is specified.

Bit	Description
0	Data is driven at rising edge of SCK, and sampled at falling edge
1	Data is driven at falling edge of SCK, and sampled at rising edge

[bit2] FSPH : Frame Sync Phase

Phase is specified to WS frame data.

Bit	Description
0	WS becomes valid "1" clock before the first bit of frame data
1	WS becomes valid at the same time as the first bit of frame data

[bit1] FSLN : Frame Sync Pulse Width

Pulse width of WS is specified.

Bit	Description
0	Pulse width is 1 cycle/SCK long (1-bit)
1	Pulse width is 1 channel long (1 channel)

Pulse width of one channel FSLN = "1" is prohibited when frame length is set to one channel by I2Sn_MCR0REG:S0CHN = 0x0 and I2Sn_CNTREG:SBFN = "0".

[bit0] FSPL : Frame Sync Polarity

Polarity of WS is set.

Bit	Description
0	Frame synchronous signal becomes valid when WS is "1"
1	Frame synchronous signal becomes valid when WS is "0"

4.4. Channel Control Register 0 (I2Sn_MCR0REG)

The Channel Control Register 0 (I2Sn_MCR0REG) controls the frame construction.

Channel Control Register 0 (I2Sn_MCR0REG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	S1CHN[4]	S1CHN[3]	S1CHN[2]	S1CHN[1]	S1CHN[0]	S1CHL[4]	S1CHL[3]
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	S1CHL[2]	S1CHL[1]	S1CHL[0]	S1WDL[4]	S1WDL[3]	S1WDL[2]	S1WDL[1]	S1WDL[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	S0CHN[4]	S0CHN[3]	S0CHN[2]	S0CHN[1]	S0CHN[0]	S0CHL[4]	S0CHL[3]
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	S0CHL[2]	S0CHL[1]	S0CHL[0]	S0WDL[4]	S0WDL[3]	S0WDL[2]	S0WDL[1]	S0WDL[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] read0 : -

[bit30:26] S1CHN : Sub Frame 1 Channel Numbers

Number of channels of sub frame 1 is set.

This is valid only when the frame is 2 sub frame construction (I2Sn_CNTREG:SBFN is "1") and is invalid when the frame is 1 sub frame construction (I2Sn_CNTREG:SBFN is "0").

Up to 32 channels can be specified, and S1CHN needs to be set to "number of channel - 1".

Setting examples are shown below.

Bits	Description
00000	Sub frame 1 becomes 1 channel construction
00001	Sub frame 1 becomes 2 channel construction
00010	Sub frame 1 becomes 3 channel construction
	...
	...
11110	Sub frame 1 becomes 31 channel construction
11111	Sub frame 1 becomes 32 channel construction

[bit25:21] S1CHL : Sub Frame 1 Channel Length

Channel length of the channel constructing sub frame 1 (bit length of channel) is set.

7 to 32 bits of channel length are available but 1 to 6 bits are prohibited. S1CHL needs to be set to "channel length - 1"

Setting examples are shown below.

Bits	Description
00000-00101	Setting is prohibited
00110	Sub frame 1 channel length is 7 bits
00111	Sub frame 1 channel length is 8 bits
	...
	...
11110	Sub frame 1 channel length is 31 bits
11111	Sub frame 1 channel length is 32 bits

Channel length can be set to 32 bits or less regardless of I2Sn_CNTREG:RHLL.

[bit20:16] S1WDL : Sub Frame 1 Word Length

Word length of the channel constructing sub frame 1 (bit length of word) is set.

7 to 32 bits of word length are available but 1 to 6 bits are prohibited. S1WDL needs to be set to "word length - 1"

S1WDL needs to be set less than or equal to the value set in I2Sn_MCR0REG:S1CHL.

Setting examples are shown below.

Bits	Description
00000-00101	Setting is prohibited
00110	Sub frame 1 word length is 7 bits

Bits	Description
00111	Sub frame 1 word length is 8 bits
	...
	...
11110	Sub frame 1 word length is 31 bits
11111	Sub frame 1 word length is 32 bits

Notes:

- *1. If I2Sn_CNTREG:RHLL is "1", set word length to 16 bits or less.*
- *2. If I2Sn_CNTREG:RHLL is "0", set word length to 32 bits or less.*

S1WDL is valid only in 2 sub frame construction (I2Sn_CNTREG:SBFN is "1") and is invalid in 1 sub frame construction (I2Sn_CNTREG:SBFN is "0").

[bit15] read0 : -

[bit14:10] S0CHN : Sub Frame 0 Channel Numbers

Number of channels of sub frame 0 is set up to 32 channels.

S0CHN needs to be set to "number of channel - 1".

Setting examples are shown below.

Bits	Description
00000	Sub frame 0 becomes 1 channel construction
00001	Sub frame 0 becomes 2 channel construction
00010	Sub frame 0 becomes 3 channel construction
	...
	...
11110	Sub frame 0 becomes 31 channel construction
11111	Sub frame 0 becomes 32 channel construction

[bit9:5] S0CHL : Sub Frame 0 Channel Length

Channel length of the channel constructing sub frame 0 (bit length of channel) is set.

7 to 32 bits of channel length are available but 1 to 6 bits are prohibited. S0CHL needs to be set to "channel length - 1".

Setting examples are shown below.

Bits	Description
00000-00101	Setting is prohibited
00110	Sub frame 0 channel length is 7 bits
00111	Sub frame 0 channel length is 8 bits
	...
	...
11110	Sub frame 0 channel length is 31 bits
11111	Sub frame 0 channel length is 32 bits

Channel length can be set to 32 bits or less regardless of I2Sn_CNTREG:RHLL

[bit4:0] S0WDL: Sub Frame 1 Word Length

Word length of the channel constructing sub frame 0 (bit length of channel) is set.

7 to 32 bits of word length are available but 1 to 6 bits are prohibited. S0WDL needs to be set to "word length - 1".

S0WDL needs to be set less than or equal to the value set in I2Sn_MCR0REG:S0CHL.

Setting examples are shown below.

Bits	Description
00000-00101	Setting is prohibited
00110	Sub frame 0 word length is 7 bits
00111	Sub frame 0 word length is 8 bits
	...
	...
11110	Sub frame 0 word length is 31 bits
11111	Sub frame 0 word length is 32 bits

Notes:

- *1. If I2Sn_CNTREG:RHLL is "1", set word length to 16 bits or less.*
- *2. If I2Sn_CNTREG:RHLL is "0", set word length to 32 bits or less.*

4.5. Channel Control Register 1 (I2Sn_MCR1REG)

The Channel Control Register 1 (I2Sn_MCR1REG) controls enable and disable functions to each channel of sub frame 0.

Channel Control Register 1 (I2Sn_MCR1REG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	S0CH[31]	S0CH[30]	S0CH[29]	S0CH[28]	S0CH[27]	S0CH[26]	S0CH[25]	S0CH[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	S0CH[23]	S0CH[22]	S0CH[21]	S0CH[20]	S0CH[19]	S0CH[18]	S0CH[17]	S0CH[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	S0CH[15]	S0CH[14]	S0CH[13]	S0CH[12]	S0CH[11]	S0CH[10]	S0CH[9]	S0CH[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	S0CH[7]	S0CH[6]	S0CH[5]	S0CH[4]	S0CH[3]	S0CH[2]	S0CH[1]	S0CH[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] S0CH : Sub Frame 0 Channel Enable

Each bit enables/disables corresponding channel of sub frame 0 (e.g. S0CH[0] bit controls 0th channel of sub frame 0, S0CH[31] bit controls 31st channel of sub frame 0).

Bit	Description
0	The corresponding channel is disabled Transmission/reception is not performed to the disabled channel.
1	The corresponding channel is enabled Transmission/reception is performed to the enabled channel.

4.6. Channel Control Register 2 (I2Sn_MCR2REG)

The Channel Control Register 2 (I2Sn_MCR2REG) controls enable and disable functions to each channel of sub frame 1.

Channel Control Register 2 (I2Sn_MCR2REG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	S1CH[31]	S1CH[30]	S1CH[29]	S1CH[28]	S1CH[27]	S1CH[26]	S1CH[25]	S1CH[24]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	S1CH[23]	S1CH[22]	S1CH[21]	S1CH[20]	S1CH[19]	S1CH[18]	S1CH[17]	S1CH[16]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	S1CH[15]	S1CH[14]	S1CH[13]	S1CH[12]	S1CH[11]	S1CH[10]	S1CH[9]	S1CH[8]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	S1CH[7]	S1CH[6]	S1CH[5]	S1CH[4]	S1CH[3]	S1CH[2]	S1CH[1]	S1CH[0]
ACCESS_TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:0] S1CH : Sub Frame 1 Channel Enable

Each bit enables/disables corresponding channel of sub frame 1 (e.g. S1CH[0] bit controls 0th channel of sub frame 1), S1CH[31] bit controls 31st channel of sub frame 1. When frame is 1 sub frame construction (I2Sn_CNTREG:SBFN is "0"), this is invalid.

Bit	Description
0	The corresponding channel is disabled Transmission/reception is not performed to the disabled channel.
1	The corresponding channel is enabled Transmission/reception is performed to the enabled channel.

4.7. Operation Control Register (I2Sn_OPRREG)

The Operation Control Register (I2Sn_OPRREG) controls receive and transmit operation.

Operation Control Register (I2Sn_OPRREG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	RXENB
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	TXENB
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	START
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:25] read0 : -

[bit24] RXENB : Receive Enable

Enable/disable functions of receiving operation is set.

Bit	Description
0	Receiving operation is disabled Reception FIFO becomes empty with writing "0" to this bit. When RXENB is "0", the data received from serial reception bus is not written to reception FIFO. DMA reception channel stops during DMA transfer.
1	Receiving operation is enabled

[bit23:17] read0 : -

[bit16] TXENB : Transmit Enable

Enable/disable functions of transmitting operation is set.

Bit	Description
0	Transmit operation is disabled Transmit FIFO becomes empty with writing "0" to this bit. When TXENB is "0", the data written to Transmission FIFO Data Registers from CPU or DMA is not written to transmission FIFO. DMA transmit channel stops during DMA transfer.
1	Transmit operation is enabled

[bit15:8] read0 : -

[bit7:1] read0 : -

[bit0] START : I2S Enable

I2S is enabled/disabled.

Bit	Description
0	I2S is stopped and internal transmission/reception FIFO becomes empty by writing "0" to this bit
1	I2S is operable.

When START is "1", it is prohibited to rewrite I2Sn_CNTREG, I2Sn_MCR0REG, I2Sn_MCR1REG, and I2Sn_MCR2REG registers.

4.8. Software Reset Register (I2Sn_SRST)

This register is to control software reset.

Software Reset Register (I2Sn_SRST)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	SRST
ACCESS_TYPE	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R,W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] read0 : -

[bit7:1] read0 : -

[bit0] SRST : Software Reset

Software reset is performed by writing "1".

I2Sn_STATUS register and each internal state machine enter initial state by software reset and transmission/reception FIFO becomes empty.

There is no influence to registers other than I2Sn_STATUS, I2Sn_INTCNT, and I2Sn_DMAACT registers.

When read value is "0" after writing "1", it indicates software reset is completed. "1" indicates software reset is in process.

4.9. Interrupt Control Register (I2Sn_INTCNT)

This register is to enable or disable interrupt function.

Interrupt Control Register (I2Sn_INTCNT)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	TXUD1M	TBERM	FERRM	TXUD0M	TXOVM	TXFDM	TXFIM
ACCESS_TYPE	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	1	1	1	1	1	1	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	RBERM	RXUDM	RXOVM	EOPM	RXFDM	RXFIM
ACCESS_TYPE	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	1	1	1	1	1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	TFTH[3]	TFTH[2]	TFTH[1]	TFTH[0]
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	RPTMR[1]	RPTMR[0]	RFTH[3]	RFTH[2]	RFTH[1]	RFTH[0]
ACCESS_TYPE	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] read0 : -

[bit30] TXUD1M : Tx FIFO Underflow Interrupt Mask

This is transmission FIFO underflow interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:TXUDR1 is not masked
1	Interrupt to CPU by I2Sn_STATUS:TXUDR1 is masked

[bit29] TBERM : Tx Block Size Error Interrupt Mask

This is interrupt mask bit of block size error of transmission channel.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:TBERR is not masked
1	Interrupt to CPU by I2Sn_STATUS:TBERR is masked

[bit28] FERRM : Frame Error Interrupt Mask

This is frame error interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:FERR is not masked
1	Interrupt to CPU by I2Sn_STATUS:FERR is masked

[bit27] TXUD0M : Tx FIFO Underflow Interrupt Mask

This is transmission FIFO underflow interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:TXUDR0 is not masked
1	Interrupt to CPU by I2Sn_STATUS:TXUDR0 is masked

[bit26] TXOVM : Tx FIFO Overflow Interrupt Mask

This is transmission FIFO overflow interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:TXOVR is not masked
1	Interrupt to CPU by I2Sn_STATUS:TXOVR is masked

[bit25] TXFDM : Tx DMA Mask

This is transmission DMA request mask register bit.

It becomes "1" by software reset.

Bit	Description
0	DMA transfer is requested when empty space in transmission FIFO is more than threshold value
1	DMA transfer is not requested even when empty space in transmission FIFO is more than threshold value

[bit24] TXFIM : Tx FIFO Interrupt Mask

This is transmission FIFO interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:TXFI is not masked
1	Interrupt to CPU by I2Sn_STATUS:TXFI is masked

[bit23:22] read0 : -

[bit21] RBERM : Rx Block Size Error Interrupt Mask

This is interrupt mask bit of reception channel block size error.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:RBERR is not masked
1	Interrupt to CPU by I2Sn_STATUS:RBERR is masked

[bit20] RXUDM : Rx FIFO Underflow Interrupt Mask

This is reception FIFO underflow interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:RXUDR is not masked
1	Interrupt to CPU by I2Sn_STATUS:RXUDR is masked

[bit19] RXOVM : Rx FIFO Overflow Interrupt Mask

This is interrupt mask bit of reception FIFO overflow.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:RXOVR is not masked
1	Interrupt to CPU by I2Sn_STATUS:RXOVR is masked

[bit18] EOPM : EOPI Interrupt Mask

This is interrupt mask bit by EOPI of status register.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:EOPI is not masked
1	Interrupt to CPU by I2Sn_STATUS:EOPI is masked

[bit17] RXFDM : Rx FIFO DMA Mask

This is reception DMA request mask bit.

It becomes "1" by software reset.

Bit	Description
0	DMA transfer is requested when reception data written to reception FIFO is more than threshold value, previous Rx DMA block transfer has completed and there is no Rx block size error.
1	DMA transfer is not requested

[bit16] RXFIM : Rx FIFO Interrupt Mask

This is reception FIFO interrupt mask bit.

It becomes "1" by software reset.

Bit	Description
0	Interrupt to CPU by I2Sn_STATUS:RXFI is not masked
1	Interrupt to CPU by I2Sn_STATUS:RXFI is masked

[bit15:12] read0 : -

[bit11:8] TFTH : Tx FIFO Threshold

Threshold value of transmission FIFO is set.

Empty space of transmission FIFO is more than threshold value and I2Sn_INTCNT: Interrupt to CPU occurs.

Empty space of transmission FIFO is more than threshold value, previous Tx DMA block transfer has completed, there is no Tx block size error I2Sn_INTCNT:TXFDM is "0": DMA is requested to DMAC.

Note:

- *These bits must not be changed during DMA transfers*

[bit7:6] read0 : -

[bit5:4] RPTMR : Rx Completion Timer

This is packet reception completion timer setting bit which sets time-out value of the internal reception completion timer.

Reception FIFO is not empty and number of its data is smaller than or equal to threshold value: the timer always counts up.

Reception FIFO is empty or the data value is more than threshold value: the timer is cleared.

When the timer times out, I2Sn_STATUS:EOPI bit is set to "1".

The timer becomes "00" by software reset.

Bit	Description
00	0 (the timer is not in operation)
01	54000 internal bus clock cycles
10	108000 internal bus clock cycles
11	216000 internal bus clock cycles

[bit3:0] RFTH : Rx FIFO Threshold

Threshold value of reception FIFO is set.

Number of receive words in reception FIFO is more than threshold value and I2Sn_INTCNT: Interrupt to CPU occurs.

Number of receive words in reception FIFO is more than threshold value, previous Rx DMA block transfer has completed, there is no Rx block size error and I2Sn_INTCNT:RXFDM is "0": DMA is requested to DMAC.

Note:

- *These bits must not be changed during DMA transfers*

4.10. Status Register (I2Sn_STATUS)

This register gives status information.

Status Register (I2Sn_STATUS)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	TBERR	RBERR	FERR	TXUDR1	TXUDR0	TXOVR	RXUDR	RXOVR
ACCESS_TYPE	R,WX	R,WX	R,W1	R,W1	R,W1	R,W1	R,W1	R,W1
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	EOPI	BSY	TXFI	RXFI
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R,W1	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	TXNUM[7]	TXNUM[6]	TXNUM[5]	TXNUM[4]	TXNUM[3]	TXNUM[2]	TXNUM[1]	TXNUM[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	RXNUM[7]	RXNUM[6]	RXNUM[5]	RXNUM[4]	RXNUM[3]	RXNUM[2]	RXNUM[1]	RXNUM[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31] TBERR : Tx Block Size Error

When I2Sn_DMAACT:TDMACT is "1" and block transfer through DMA transmission channel is more than I2Sn_INTCNT:TFTH + 1, this bit is set to 1 and the DMA transmission channel is stopped.

When TBERR is "1" and I2Sn_INTCNT:TBERM is "0", interrupt to CPU occurs.

This bit becomes "0" by software reset.

[30] RBERR : Rx Block Size Error

When I2Sn_DMAACT:RDMACT is "1" and block transfer through DMA reception channel is more than I2Sn_INTCNT:RFTH + 1, this bit is set to 1 and the DMA reception channel is stopped.

When RBERR is "1" and I2Sn_INTCNT:RBERM is "0", interrupt to CPU occurs.

This bit becomes "0" by software reset.

[bit29] FERR : Frame Error

Occurrence of frame error is indicated. This bit is set to "1" in the following cases:

- Frame synchronous signal cannot be received with the set frame rate in the free-running mode (I2Sn_CNTREG:FRUN = "0") and the slave mode (I2Sn_CNTREG:MSMD = "0").
- The next frame synchronous signal is received during frame transmission/reception in the slave mode (I2Sn_CNTREG:MSMD), not free-running mode (I2Sn_CNTREG:FRUN = "1").

When FERR is "1" and I2Sn_INTCNT:FERRM is "0", interrupt to CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

[bit28] TXUDR1 : Tx FIFO Underflow Error

When transmission FIFO underflows at the start of frame, the value is set to "1".

When TXUDR1 is "1" and I2Sn_INTCNT:TXUD1M is "0", interrupt to the CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

Note:

- *When the transmission FIFO underflows at the start of frame, there will be no more attempts to read the transmission FIFO during the rest of the frame, so if the transmission FIFO is not written after it has underflown, I2Sn_STATUS:TXUDR0 is not set to "1".*

[bit27] TXUDR0 : Tx FIFO Underflow Error

When transmission FIFO underflows during frame transmission (from 2nd bit word to the last frame of the word), the value is set to "1".

When TXUDR0 is "1" and I2Sn_INTCNT:TXUD0M is "0", interrupt to the CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

Note:

- *When the transmission FIFO underflows during frame transmission and is still empty at the start of the next frame, I2Sn_STATUS:TXUDR1 is set to "1" too.*

[bit26] TXOVR : Tx FIFO Overflow Error

When transmission FIFO overflows, the value is set to "1" indicating transmission data is written in the condition that transmission FIFO is full. The value "1" indicates 1 word or more of transmission data is ignored.

When TXOVR is "1" and I2Sn_INTCNT:TXOVM is "0", interrupt to CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

[bit25] RXUDR : Rx FIFO Underflow Error

When reception FIFO underflows, the value is set to "1" indicating read access is carried out to reception FIFO in the condition that reception FIFO is empty.

When RXUDR is "1" and I2Sn_INTCNT:RXUDM is "0", interrupt to the CPU occurs. Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

Note:

- *This flag is not set when the DAP controller reads the Reception FIFO Data Registers while the reception FIFO is empty.*

[bit24] RXOVR : Rx FIFO Overflow Error

When reception FIFO overflows, the value is set to "1" indicating reception is carried out in the condition that reception FIFO is full. The value "1" indicates 1 word or more of reception data is ignored.

When RXOVR is "1" and I2Sn_INTCNT:RXOVM is "0", interrupt to CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

[bit23:20] read0 : -

[bit19] EOPI : Interrupt Flag for Rx Timer

This is an interrupt flag that is triggered when an internal reception timer times out. The reception timer is enabled when following conditions are met at the same time:

I2Sn_CNTREG:RXDIS is set to "0".

I2Sn_OPRREG:START bit is set to "1" and I2Sn_OPRREG:RXENB = "1".

After the reset, operation starts with the 1st word reception.

The count value is automatically cleared if reception FIFO data is more than threshold or it becomes empty. When the reception FIFO is not empty and the number of data is less than or equal to the threshold, the reception timer is incremented with each internal bus clock cycle.

EOPI is set to "1" when reception FIFO is non-empty and reception timer count value reaches timeout set by I2Sn_INTCNT:RPTMR.

When EOPI is "1" and I2Sn_INTCNT:EOPM is "0", interrupt to CPU occurs.

Writing "1" from CPU clears the value to "0".

This bit becomes "0" by software reset.

[bit18] BSY : Serial Tx Busy

Serial transmission control part is in busy state. This bit is not affected by software reset.

Bit	Description
0	Serial transmission control part is in idle state
1	Serial transmission control part is in busy state

[bit17] TXFI : Tx FIFO Empty

When number of empty space of transmission FIFO is greater than the threshold set in I2Sn_INTCNT:TFTH, this bit is set to "1".

This bit is "1" and I2Sn_INTCNT: interrupt to CPU occurs.

When number of empty slot of transmission FIFO becomes smaller or equal to the threshold by writing to Transmission FIFO Data Registers from CPU or DMAC, this bit is cleared automatically to "0".

The value also become "0" when I2Sn_OPRREG:START bit is "0" or I2Sn_OPRREG:TXENB bit is "0".

If software reset is performed at I2Sn_OPRREG:START bit = "1" and I2Sn_OPRREG:TXENB bit = "1", the value becomes "0" during software reset then changes to "1" after the process.

[bit16] RXFI : Rx FIFO Full

When number of reception FIFO data becomes more than the threshold set in I2Sn_INTCNT:RFTH, this bit is set to "1".

This bit is "1" and I2Sn_INTCNT: interrupt to CPU occurs.

When number of data in reception FIFO becomes smaller or equal to the threshold by reading Reception FIFO DATA Registers from CPU or DMAC, this bit is automatically cleared to "0".

This bit becomes "0" by software reset.

[bit15:8] TXNUM : Number of Tx FIFO Data

The number of data in transmission FIFO is indicated.

This field is incremented by write access to Transmission FIFO Data Registers and decremented by serial word transmission.

Max. value of 66 can be displayed in the simultaneous transfer mode and value of 132 in the transmission only mode.

This field becomes "00000000" by software reset.

[bit7:0] RXNUM : Number of Data in Rx FIFO

The number of data in reception FIFO is indicated.

This field is incremented by word reception from serial bus and decremented by read access to Reception FIFO Data Registers.

Maximum value of 66 can be displayed in the simultaneous transfer mode and value of 132 in the reception mode.

This field becomes "00000000" by software reset.

4.11. DMA Activate Register (I2Sn_DMAACT)

This register is to enable or disable DMA control function.

DMA Activate Register (I2Sn_DMAACT)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	TDMACT
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	RDMACT
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:24] read0 : -

[bit23:17] read0 : -

[bit16] TDMACT : Tx DMA Control

The DMA transmission channel is activated.

After transmission channel starts, software should write "1" to TDMACT for transmit DMA channel to be active. When TDMACT is "0", transfer request of transmission channel block is not sent to DMAC.

Writing "0" from CPU clears the value to "0".

This bit becomes "0" by software reset.

Bit	Description
0	DMA transmission channel is disabled
1	DMA transmission channel is enabled

Clearing TDMACT also clears write transmission request.

[bit15:8] read0 : -

[bit7:1] read0 : -

[bit0] RDMACT : Rx DMA Control

The DMA reception channel is activated.

After reception channel starts, software should write "1" to RDMACT for receive DMA channel to be active. When RDMACT is "0", transfer request of reception channel block is not sent to DMAC.

Writing "0" from CPU clears the value to "0".

This bit becomes "0" by software reset.

Bit	Description
0	DMA reception channel is disabled
1	DMA reception channel is enabled

Clearing RDMACT also clears reception transfer request.

4.12. Debug Register (I2Sn_DEBUG)

This register is to enable or disable debug function.

Debug Register (I2Sn_DEBUG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	read0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	read0	read0	read0	read0	read0	read0	read0	DBGE
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] read0 : -

[bit7:1] read0 : -

[bit0] DBGE : Debug Enable (DBGE)

This bit is used to enable/disable debug mode for I2S.

Bit	Description
0	Debug mode disabled
1	Debug mode enabled

This bit takes effect only in master mode (i.e. I2Sn_CNTREG:MSMD = "1").

When DBGE is set to "1" and the processor is in debug state and I2S is working as a master, then the serial interface is halted by stopping the activity on the SCK output. The activity on the serial clock resumes either when the processor leaves debug state or DBGE is set to "0".

4.13. Module ID Register (I2Sn_MIDREG)

This register implements unique module identification number. Refer to the device datasheet for the module identification number of I2S module in the device.

Module ID Register (I2Sn_MIDREG)

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
ACCESS_TYPE	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX
PROT_TYPE								
INITIAL_VALUE	0	0	0	0	0	0	0	1

[bit31:0] MID : Module ID

This read-only register gives the unique module identification number of I2S module. The unique Module ID number identifies the version of the I2S module used in the MCU. Refer to the device specific datasheet for the module identification number of its I2S.

CHAPTER 33: PCMPWM

This chapter explains the function and operation of the PCMPWM module.

1. Overview
2. Configuration and Block Diagram
3. Operation of the PCMPWM
4. Registers

CODE: PCMPWM-S6J3200-E1.1

1. Overview

The PCMPWM module converts Pulse Code Modulated (PCM) data samples to Pulse Width Modulated (PWM) signals. The intended purpose of the module is to provide simple audio output capabilities.

Also refer to the chapter of "Sound System Configuration" how to configure the PCMPWM.

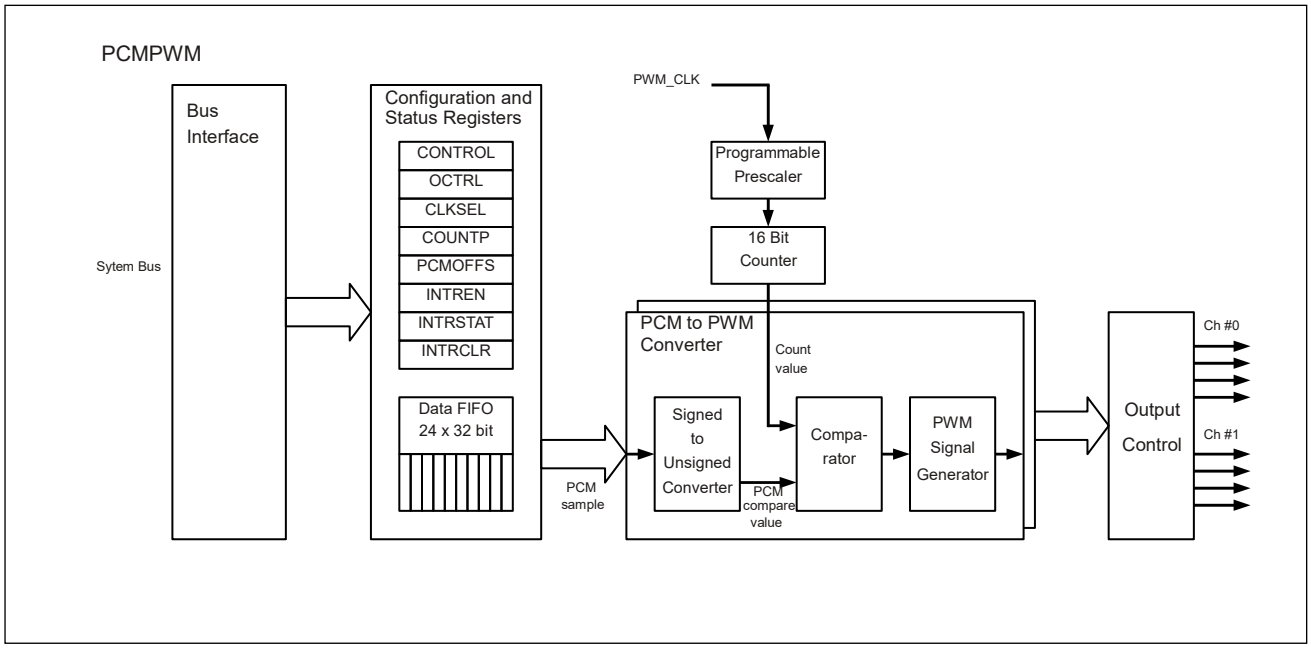
Features of the PCMPWM

- Two separate channels for stereo audio output
- Optional mono audio output mode
- A maximum output resolution of 16 bits
- An output resolution of 12 bits at a sampling frequency of 24.4 kHz (at 100 MHz PWM clock)
- An output resolution of 11 bits at s sampling frequency of 48.8 kHz (at 100 MHz PWM clock)
- Three modes of operation for different speaker interfacing:
  - Low-pass filter output mode
  - Simplified H-bridge output mode
  - Full H-bridge output mode
- Support for the DMA transfer of PCM data samples using DMA block transfer mode
- FIFO input buffer for PCM samples, with a depth 24
- Programmable clock divider for PWM cycle time
- Optional output of silence signals in debug mode and normal mode

2. Configuration and Block Diagram

2.1. Block Diagram

Figure 2-1 Block Diagram of the PCMPWM Module



Bus Interface

The Bus Interface connects the PCMPWM module's register file to the system bus.

Configuration and Status Registers

This block holds all the module's user accessible registers.

Data FIFO

This block represents a buffer for PCM samples and a boundary between the system bus clock and the PWM clock.

Clock Prescaler and 16-Bit Counter

These two blocks represent the free running counter and its prescaler used for the PCM to PWM conversion.

PCM to PWM Converters

These blocks represent the actual PCM to PWM conversion units. Each channel uses its own converter block.

Output Control

This block controls the PWM signal outputs. It e.g. defines the signals' output polarity and keeps unused outputs at their inactive levels.

2.2. Configuration of the PCMPWM Module

PCMPWM Enable

After reset the PCMPWM module is disabled. All PWM outputs are held at their inactive values. To start the PCM to PWM conversion and generate the PWM output signals, all configuration registers should be initialized and the global enable bit PCMPWMi_CONTROL:EN set. Once the module has been enabled, it must be ensured that the PCM data samples are provided to the FIFO buffer at the required rate. This may be done by either the CPU or the DMA unit.

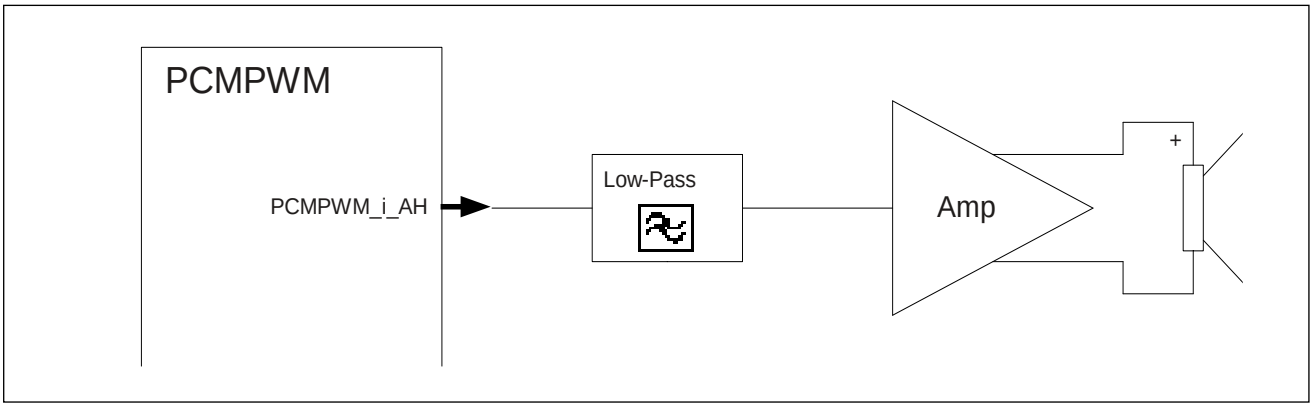
Operation Modes

The PCMPWM module supports three general modes of operation. Each mode is dedicated to a specific type of connection to the speaker. The operation mode is configured in the MODE bits of the CONTROL register (PCMPWMi_CONTROL:MODE). The following paragraphs describe these modes in detail.

1. Low-Pass Filter Mode

In the low-pass filter mode only a single output is used. The output drives an audio amplifier with a low-pass filter connected in-between. Figure 2-2 depicts the corresponding circuit.

Figure 2-2 Simplified Schematic Illustration of Low Pass Filter Mode Output Circuit

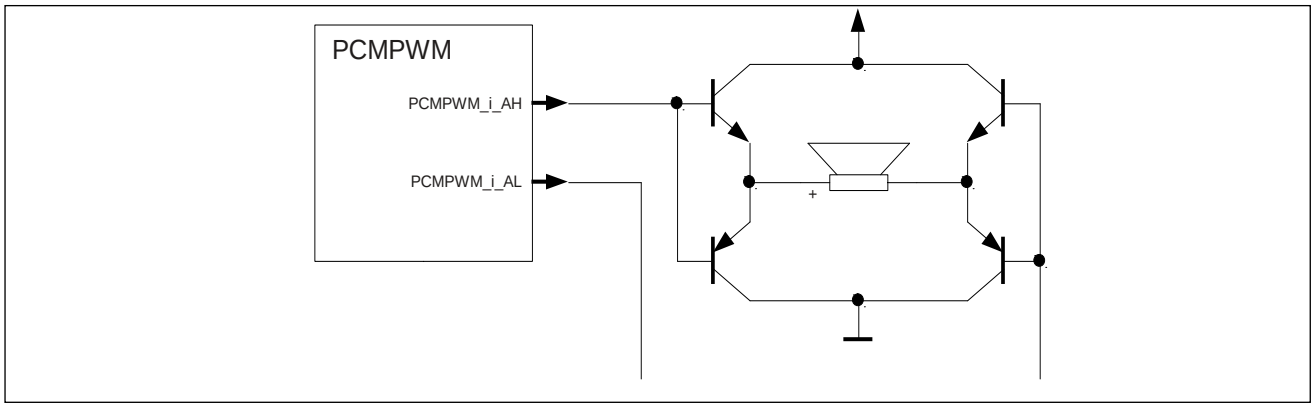


For positive PCM data samples an output signal with a duty cycle of more than 50% is generated. For negative PCM data samples, the signal has a duty cycle of less than 50%. A PCM data sample of zero generates an exact 50% duty cycle output signal. A 0% and a 100% duty cycle may be reached as well if the PWM count period and the PCM conversion offset are configured appropriately (registers PCMPWMi_COUNTP and PCMPWMi_PCMOFFS).

2. Simplified H-Bridge Mode

In the simplified H-bridge mode two outputs are used to drive a speaker. A pair of complementary emitter followers is connected to each output. Figure 2-3 depicts the circuit for simplified H-bridge mode.

Figure 2-3 Simplified Schematic Illustration of Simplified H-Bridge Mode Output Circuit

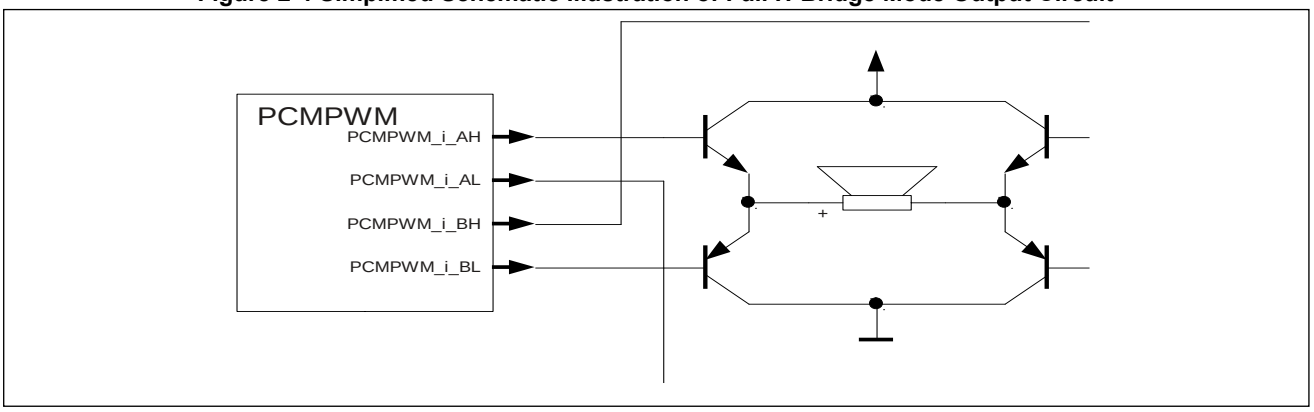


In simplified H-bridge mode, the signal at the PCMPWM_i_AL output is the inverted PCMPWM_i_AH signal.

3. Full H-Bridge Mode

In the full H-bridge mode four outputs per speaker are used. All transistors of the H-bridge have individual driver outputs. Figure 2-4 depicts the circuit for full H-bridge mode.

Figure 2-4 Simplified Schematic Illustration of Full H-Bridge Mode Output Circuit

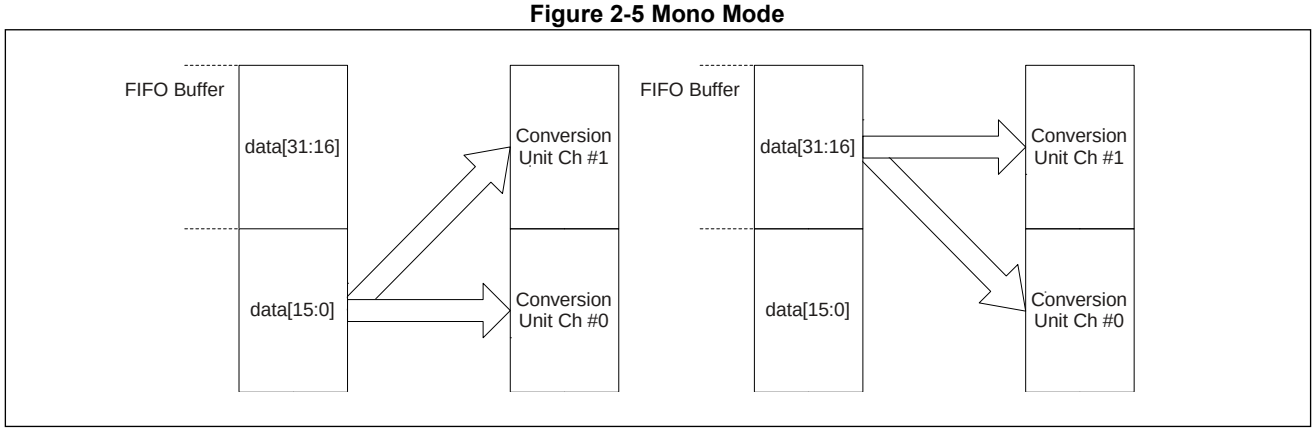


In full H-bridge mode the PCMPWM_i_AL signal is again the inverted PCMPWM_i_AH signal whereas the signal at the PCMPWM_i_BL output is the inverted PCMPWM_i_BH signal. The PCMPWM_i_Ax output pair is activated for positive PCM data samples. The PCMPWM_i_Bx output pair is activated for negative PCM data samples. If the PCM data sample is zero, no branch of the H-bridge is activated. The PCMPWM_i_Ax and PCMPWM_i_Bx branches are never active at the same time.

Mono or Stereo Mode

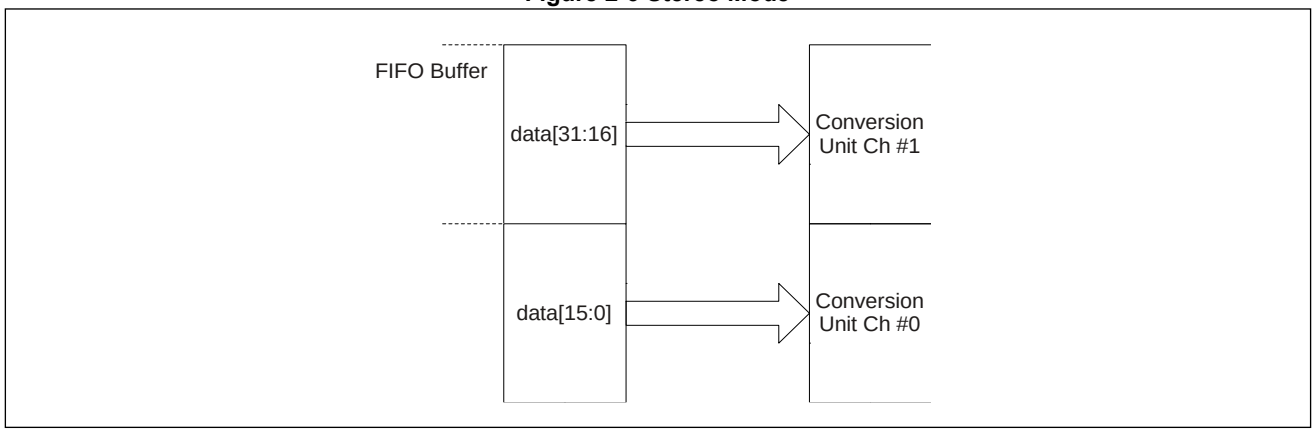
The PCMPWM module supports two modes of driving the two output channels – mono and stereo. In mono mode a single stream of PCM data samples is used to drive both output channels. I.e. both conversion units use the same PCM sample data values. Mono mode alternates between the lower (used first) and the upper 16 bits of the PCM data sample word. Thus the effective depth of the FIFO buffer increases to 48. Figure 2-5 depicts the data transfer from the FIFO buffer to the PCM to PWM conversion units in mono mode.

Figure 2-5 Mono Mode



In stereo mode each PCM sample consists of a pair of values for the two channels. They are written to the module at the same time but the values for both channels are independent of each other. The lower 16 bits of the sample word are used for channel #0 and the upper 16 bits are used for channel #1. Mono or stereo mode is configured with the PCMPWMi_CONTROL:STEREO bit. Figure 2-6 depicts the data transfer from the FIFO buffer to the PCM to PWM conversion units in stereo mode.

Figure 2-6 Stereo Mode



Single or Double Output Mode

The PCMPWM module supports two ways of processing the PCM data samples. In single output mode each PCM data sample is read from the FIFO buffer and then processed by the conversion unit. In double output mode each PCM data sample is processed twice. I.e. the same PCM data sample is used for two PWM cycles. The double output mode enables doubling of the PWM output frequency at the same PCM sample data rate.

The single/double output mode is independent of the mono/stereo mode setting.

Behavior in Debug Mode

The PCMPWM module supports debug mode. If enabled, the module stops converting PCM data samples after the one in progress and instead simply outputs a silent signal on both channels. The silent signal corresponds to a continuous stream of PCM data samples with the value of zero. In this mode no PCM samples are read from the FIFO buffer. However, new sample data may still be written to the FIFO buffer.

If the module's support for debug mode is not enabled, the PCMPWM module continues with regular PCM data sample conversion and PWM signal output.

DMA Mode

The PCM data samples may be transferred to the PCMPWM module via DMA. For this the DMA mode must be enabled by setting the PCMPWMi_CONTROL:DMAEN bit. The transfer works fully autonomously without any CPU intervention. The transfer takes place in chunks. The respective maximum number of 32-bit words per transfer is defined by PCMPWMi_CONTROL:FEST + 1. If the number of free entries in the FIFO buffer exceeds the number specified in PCMPWMi_CONTROL:FEST, a DMA transfer is requested by the PCMPWM module. The DMA then may fill up the full number of entries given by PCMPWMi_CONTROL:FEST + 1.

To set up the PCMPWM module for a DMA data transfer, the DMA unit must be configured for block transfer mode. The block size configured in the DMA must match PCMPWMi_CONTROL:FEST + 1. The PCMPWM module will then assert the PCMPWM_i_DMA_REQ output when the number of free entries in the FIFO buffer exceeds the value given by FEST. PCMPWM_i_DMA_REQ stays active until the DMA acknowledges the request by asserting PCMPWM_i_DMA_REQ_ACK. After the DMA has transferred FEST + 1 PCM data samples to the FIFO buffer, the PCMPWM will again assert the PCMPWM_i_DMA_REQ line again when the required number of free FIFO buffer entries is available. The PCMPWM module keeps track of the number of transferred data words by itself. There is an internal down counter for this purpose which is reloaded with the value of FEST + 1 when PCMPWM_i_DMA_REQ is asserted. It is decremented with each word written to the FIFO buffer. When the counter reaches zero the transfer is complete and another DMA transfer may be requested. It is an error if the DMA writes data to the FIFO while the down counter is zero. This is condition is flagged as a DMA block error in the interrupt status register.

Silence Mode

By setting the PCMPWMi_CONTROL:SILENCE trigger, the module is set to silence mode. When this mode is triggered, the register holding the PCM sample to be converted is cleared to 0 and the reading from the FIFO is stopped. Also the FIFO is flushed. The FIFO can then be filled again via DMA or the CPU. As soon as the number of empty entries in the FIFO is equal or less than PCMPWMi_CONTROL:FEST, reading from the FIFO is resumed.

At begin of operation when the PCMPWM is enabled, the module is in silence mode until the required number of FIFO entries are written. This prevents possible FIFO under-runs directly after the start of operation.

2.3. Output Configuration

The two output channels of the PCMPWM module may be individually enabled and disabled. In addition, the output polarity of each channel may be selected. The enable bits are PCMPWMi_OCTRL:EN0 and PCMPWMi_OCTRL:EN1. The output polarity is programmed with PCMPWMi_OCTRL:LEVL0 and PCMPWMi_OCTRL:LEVL1.

If a channel's output is disabled, all the corresponding signals stay at their inactive values. The inactive value depends on the programmed output polarity. If a channel is programmed to an operation mode which does not use all outputs, the unused outputs also stay at their inactive values. Table 2-1 shows the inactive values of all outputs depending on the selected polarity.

Table 2-1 Inactive Output Values

PCMPWMi_OCTRL:LEVL0/LEVL1	PCMPWM_i_AH/BH	PCMPWM_i_AL/BL
0	1	0
1	0	1

3. Operation of the PCMPWM

This section describes the operation of the PCMPWM module.

3.1. Description of the PCM to PWM Conversion Process

The actual conversion of the PCM data samples to a PWM signal is done in the PCM to PWM converter as shown in Figure 2-1. The first step is a conversion of the signed PCM sample to an unsigned PCM compare value. How this is done depends on the operation mode. In low-pass filter mode and simplified H-bridge mode the step is a simple addition. The value to be added is configured in the PCMPWMi_PCMOFFS register. In full H-bridge mode the PCM compare value is the absolute value of the PCM data sample. Table 3-1 summarizes how the PCM compare value is computed in each operation mode.

Table 3-1 Inactive Output Values

Operation Mode	Computation
Low Pass Filter	PCM sample + PCMOFFS
Simplified H-Bridge	PCM sample + PCMOFFS
Full H-Bridge	|PCM sample|

The second step of the PCM to PWM conversion process comprises a free running counter and two comparators. The counter counts up from zero to a programmable maximum value. The maximum value is given by the PCMPWMi_COUNTP register. Whenever the counter has reached its maximum value, it wraps around to zero. The counter value is continuously compared against two values. The first value is zero and the second is the PCM compare value. The outputs of the two comparators are fed into the actual PWM signal generation. Depending on the operation mode this final block generates the actual PWM signals. Table 3-2 summarizes the logic for generating the PWM signals from the two comparator outputs. Please note the signal levels in the table are for the output polarity with PCMPWMi_OCTRL:LEVLx set to logic 1.

Table 3-2 Generation of PWM Output Signals

Event		Low-Pass Filter Mode	Simplified H-Bridge Mode		Full H-Bridge Mode			
		AH	AH	AL	AH	AL	BH	BL
Counter = 0	PCM sample positive	↑	↑	↓	↑	↓	0	1
	PCM sample = 0				0	1	0	1
	PCM sample negative				0	1	↑	↓
Counter = PCM compare value	PCM sample positive	↓	↓	↑	↓	↑	0	1
	PCM sample = 0				0	1	0	1
	PCM sample negative				0	1	↓	↑

Note:

- *In case the PCM compare value is zero, the event given for equality with the PCM compare value takes precedence.*

Figure 3-1 to Figure 3-4 depict some exemplary conversion cycles for all three operation modes. For low-pass filter mode and simplified H-bridge mode, PCMPWMi_COUNTP:COUNTP is set to 0xFFFF and PCMPWMi_PCMOFFS:PCMOFFS is set to 0x8000. For full H-bridge mode PCMPWMi_COUNTP:COUNTP is set to 0x7FFF. The diagrams show the value of the

free running counter periodically ascending from 0 to its maximum value as well as the PCM samples and the corresponding PCM compare value.

Figure 3-1 PCM to PWM Conversion in Low-Pass Filter Mode

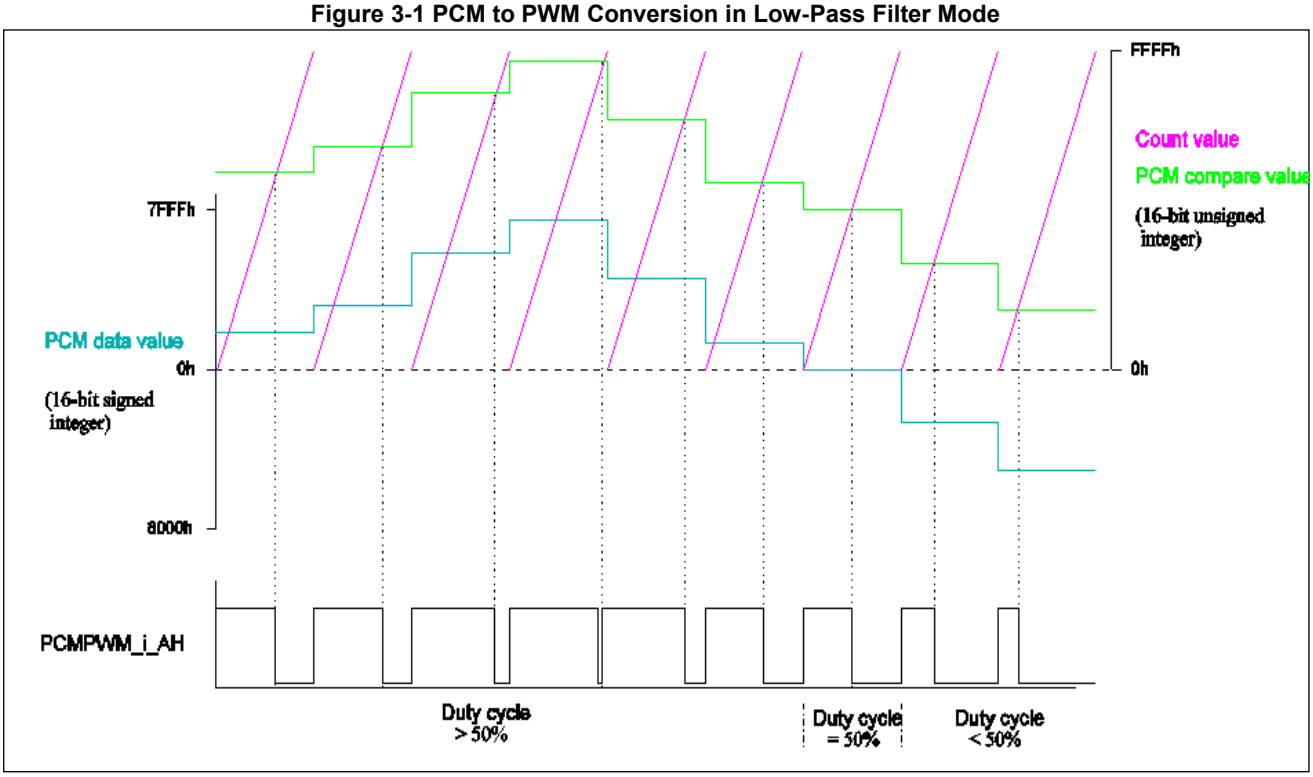


Figure 3-2 PCM to PWM Conversion in Simplified H-Bridge Mode

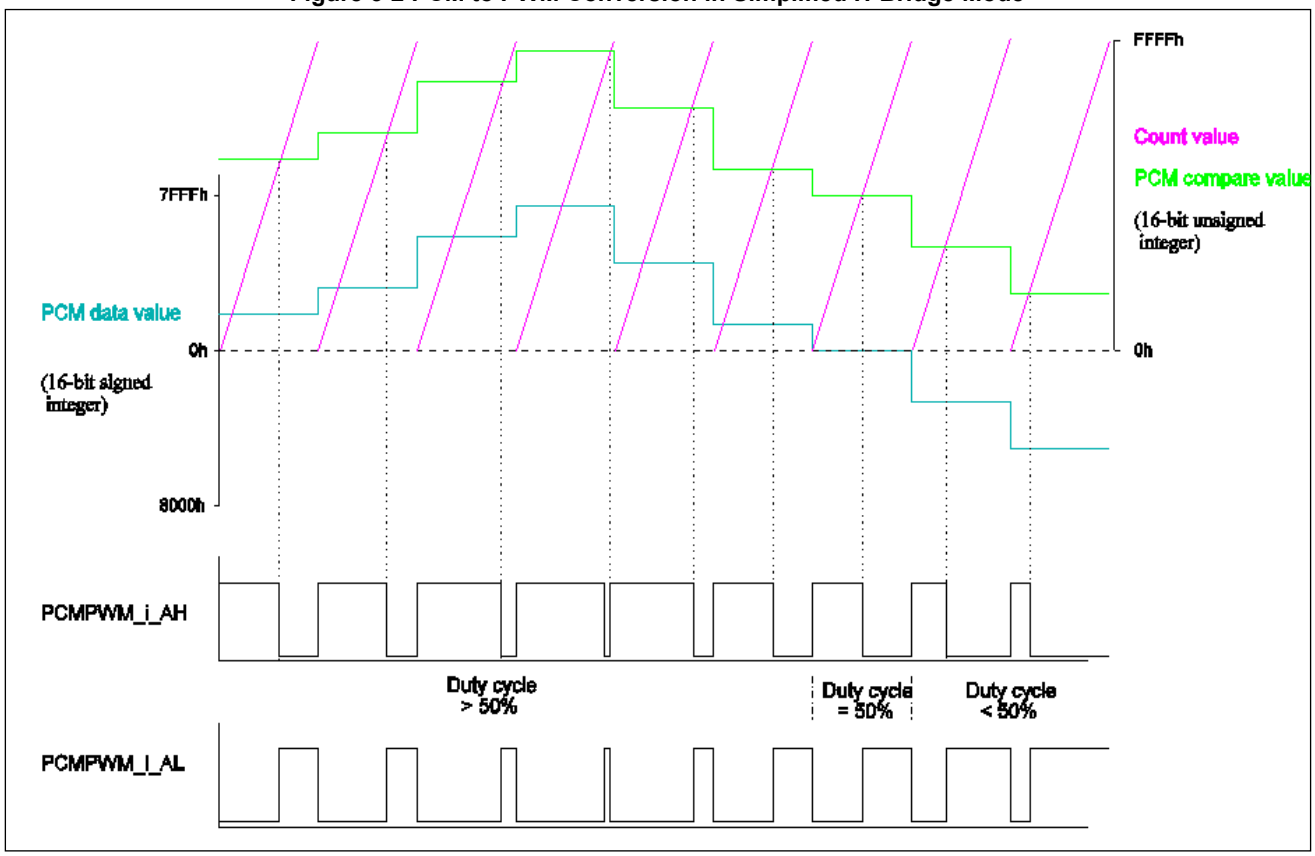


Figure 3-3 PCM to PWM Conversion in Full H-Bridge Mode

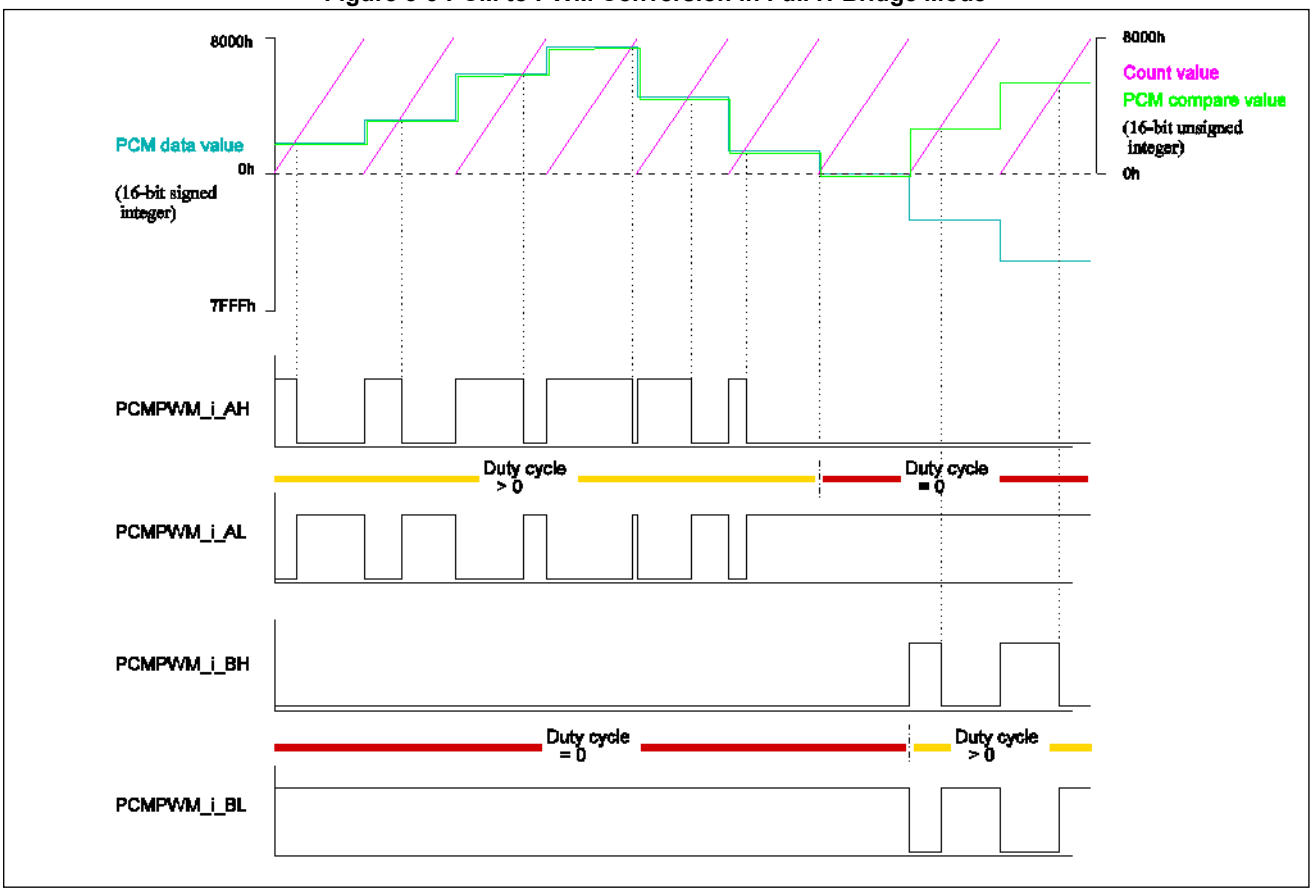
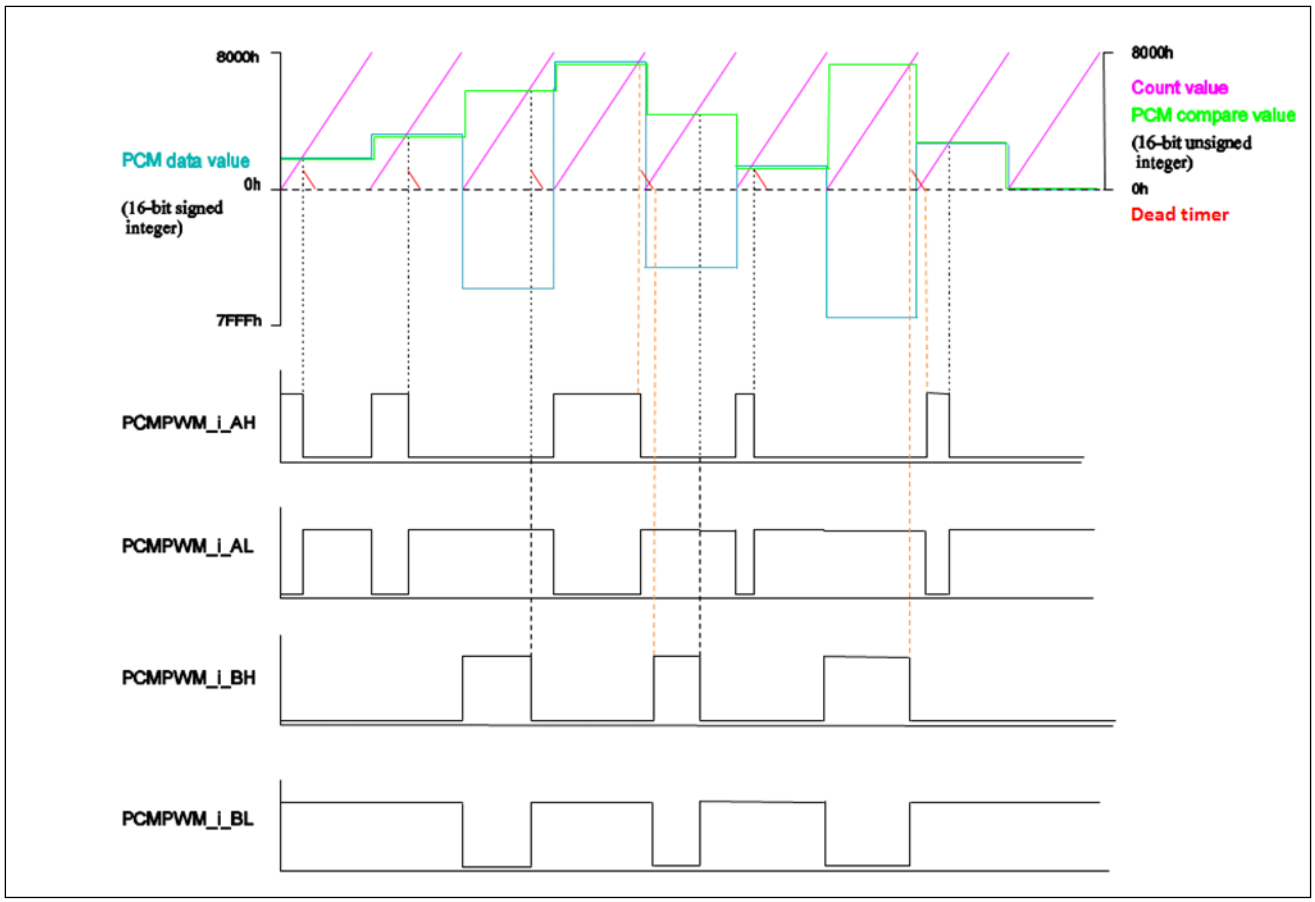


Figure 3-4 PCM to PWM Conversion in Full H-Bridge Mode (with Dead Timer)



3.2. PWM Cycle Time Configuration

The free running counter used in the PCM to PWM conversion process determines the cycle time of the generated PWM signal. The PWM cycle time is equal to the counter's full count period. So, the cycle time computes as follows:

$$T_{PWM} = \frac{COUNTP + 1}{f_{counter}}$$

The countp parameter is configured in the PCMPWMi_COUNTP register. The count frequency of the counter may be configured in the PCMPWMi_CLKSEL register. It determines the counter clock by dividing the PWM clock PWM_CLK by 1, 2, 4 or 8. So, the final formula for the PWM cycle time results to:

$$T_{PWM} = \frac{(COUNTP + 1) \times 2^{CLKSEL}}{f_{PWM_CLK}}$$

3.3. PCM Data Sample Input

The PCM to PWM conversion block reads the PCM data samples from a FIFO buffer. To ensure continuous operation of the PCM to PWM conversion, it is essential to provide the FIFO buffer with the same sustained rate of PCM data samples. The required frequency for the data samples corresponds to the reciprocal of the PWM cycle time:

$$f_{sample} = \frac{f_{PWM_CLK}}{(COUNTP + 1) \times 2^{CLKSEL}}$$

The data samples may be provided by the CPU or may be transferred via DMA. For a CPU based transfer there is support for requesting data by interrupt. Alternatively the CPU may poll the interrupt status register to determine, if there is free space in the FIFO buffer. Regardless of how the buffer FIFO is supplied with data samples, the PCMPWMi_CONTROL:FEST field specifies, at which number of free FIFO buffer entries the data transfer is requested. If the number of free entries exceeds that specified in the FEST field, the PCMPWMi_INTRSTAT:DREQ status bit is set and an interrupt request may be asserted or another DMA data transfer may be initiated.

If the PCM to PWM conversion tries to read a PCM data sample from the FIFO buffer and finds it empty, it uses the PCM value instead. I.e. the module outputs the last PWM pattern again, if no PCM data is supplied. To detect this error case of FIFO buffer under-run, an interrupt request is asserted.

Please note the above formula limits the resolution of the PCM to PWM conversion at a given f_{PWM_CLK} and f_{sample}. E.g. at a clock frequency of 80 MHz, a sample frequency of 19.5 kHz, and CLKSEL set to zero, the counter may count to 4103 only. This corresponds to a resolution of about 12 bits. Table 3-3 shows some examples for the configuration of the PCMPWM module and the resulting sample frequencies and resolutions.

Table 3-3 Configuration Examples

f_{PWM_CLK}	CLKSEL	COUNTP	f_{sample}	Resolution
80 MHz	0	4,095	19.53 kHz	12 bit
80 MHz	0	65,535	1.22 kHz	16 bit
80 MHz	2	255	78.13 kHz	8 bit
100 MHz	1	1,023	48.83 kHz	10 bit

3.4. Interrupts

The PCMPWM module interrupts are controlled by three registers: the Interrupt Enable Register (PCMPWMi_INTREN), the Interrupt Status Register (PCMPWMi_INTRSTAT) and the Interrupt Clear Register (PCMPWMi_INTRCLR). After reset all interrupts are disabled. If an interrupt is to be used, it must be first enabled. The current status of an interrupt may be checked at any time in the Interrupt Status Register. The Interrupt Status Register contents are independent of the enable status of the interrupts. I.e. the Interrupt Status Bits are not masked by the Interrupt Enable Register.

If an interrupt has occurred, it can be reset by the Interrupt Clear Register. Writing a logic 1 to a bit in the Interrupt Clear Register clears the corresponding interrupt line as well as the interrupt status bit.

1. DMA Block Error Interrupt

This interrupt indicates an error case when the DMA tries to transfer more data to the FIFO buffer than configured by PCMPWMi_CONTROL:FEST + 1.

2. FIFO Buffer Under-Run Error Interrupt

This interrupt indicates an under-run of the FIFO buffer for PCM data samples. I.e. the PCM to PWM conversion has tried to read a PCM data sample from the FIFO buffer when it was empty. Instead of a new PCM data sample the PCM to PWM conversion uses the last PCM value instead.

3. FIFO Buffer Overflow Error Interrupt

This interrupt indicates the CPU has tried to write another PCM data sample to the FIFO buffer when it was already completely filled.

4. Data Request Interrupt

This interrupt indicates there is at least space for another FEST + 1 PCM data samples in the FIFO buffer. I.e. the PCMPWM module asserts the interrupt request when a read from the FIFO buffer frees up another PCM samples slot, so that there are FEST + 1free entries in total.

3.5. Dead Timer Operation

In full H-Bridge mode, the 16-bit counter is compared in each count period CP with the value from the function minimum(duty, cycle) (this is the falling edge or the end of the PWM cycle). If the two values match, a down counter loaded with PCMPWMi_CONTROL:DTVAL starts to decrement with every clock cycle of the 16-bit counter. As long as the down counter is not zero, the opposite phase of the PWM outputs is masked with inactive values. Opposite phase in this case means:

- If the PCM data sample of count period CP is positive, then output signals PCMPWM_i_BL/BH are masked, although this doesn't happen in usual setup.
- If the PCM data sample of count period CP is negative, then output signals PCMPWM_i_AL/AH are masked, although this doesn't happen in usual setup.

4. Registers

This section describes the registers of the PCMPWM module.

Note:

- *The suffix 'i' in the register name indicates that the register is in instance 'i' of the module.*

The following registers are available for each instance of the PCMPWM:

- PCMPWM Control Register (PCMPWMi_CONTROL)
- PCMPWM Output Control Register (PCMPWMi_OCTRL)
- PCMPWM Clock Select Register (PCMPWMi_CLKSEL)
- PCMPWM Count Period Register (PCMPWMi_COUNTP)
- PCM Offset Register (PCMPWMi_PCMOFFS)
- PCM Interrupt Enable Register (PCMPWMi_INTREN)
- PCM Interrupt Status Register (PCMPWMi_INTRSTAT)
- PCM Interrupt Clear Register (PCMPWMi_INTRCLR)
- PCM Data Register 0..15 (PCMPWMi_DATA)

4.1. PCMPWM Control Register (PCMPWMi_CONTROL)

This Global Control Register is used to configure the PCMPWM module's basic mode of operation.

REGISTER_NAME	PCMPWMi_CONTROL
OFFSET	0x00000000
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	DTVAL[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved			FEST[4:0]				
ACCESS_TYPE	R0,WX			R/W				
PROT_TYPE	Wp							
INITIAL_VALUE	0			0				

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved		DOUBLE	MODE[1:0]		STEREO	DBGEN	DMAEN
ACCESS_TYPE	R0,WX		R/W	R/W		R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0		0	0		0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved						SILENCE	EN
ACCESS_TYPE	R0,WX						R0,W1	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0						0	0

[bit31:24] DTVAL[7:0] : Dead Timer Value

This field defines the number of PWM_CLK cycles after the trailing edge for which the opposite phase of the PWM outputs in full H-Bridge Mode are masked with inactive value.

Setting of this field to a value greater than or equal to PCMPWMi_COUNTP:COUNTP is not allowed.

[bit20:16] FEST[4:0] : FIFO Empty Space Threshold

This field defines the number of empty entries in the FIFO buffer, which triggers a DMA request and/or a data request interrupt when it is exceeded. If this field is set to a value from 0 to 23 a data request is generated when the number of empty FIFO buffer entries exceeds this value. If this field is set to a value greater than 23, a data request is never generated.

Note:

- *These bits must not be changed during DMA transfers.*

[bit13] DOUBLE : Double Mode Enable

This bit enables operation in single or double mode.

Bit	Description
0	The module operates in single output mode, i.e. each PCM data sample is processed once by the conversion unit
1	The module operates in double output mode, i.e. each PCM data sample is processed twice by the conversion unit

[bit12:11] MODE[1:0] : Operation Mode

This field defines the module's mode of operation.

Bits	Description
00	The module is configured for low-pass filter mode
01	The module is configured for simplified H-bridge mode
10	The module is configured for full H-bridge mode
11	Not allowed

[bit10] STEREO : Stereo Mode Enable

This bit enables operation in mono or stereo mode.

Bit	Description
0	The module operates in mono mode, i.e. both channels output the same data
1	The module operates in stereo mode, i.e. each channel outputs its individual data

[bit9] DBGEN : Debug Mode Enable

This bit controls the module's operation when the CPU is in debug state.

Bit	Description
0	Disabled - The module continues regular operation when the CPU enters debug state
1	The module generates silence when the CPU is in debug state; no data is fetched from the FIFO buffer

[bit8] DMAEN : DMA Mode Enable

This bit controls the module's DMA interface.

Bit	Description
0	The DMA interface is disabled

Bit	Description
1	The DMA interface is enabled

[bit1] SILENCE : Silence Module Trigger

This bit triggers the Silence Mode of the PCMPWM module. Reading of this bit returns always 0. Writing 1 enables the Silence Mode, writing 0 has no effect. Since this bit is in the same byte as the EN bit and writing to SILENCE has only sense if the module is enabled, EN = 1 needs to be written at the same time as SILENCE is written.

Bit	Description
0	No change
1	Enable Silence Mode

[bit0] EN : PCMPWM Module Enable

This bit globally enables or disables the PCMPWM module. This bit must not be set before the module's configuration has been completed. After the module has been enabled via this bit and the PCM sample data FIFO buffer has been prefilled, the PCM to PWM conversion starts immediately. If the FIFO buffer is empty when the module gets enabled, silence is driven on the PWM outputs and the PCM to PWM conversion starts as soon as there are equal or less than PCMPWMi_CONTROL:FEST empty spaces in the FIFO. The conversion counter is always restarted from zero when the module gets enabled via this bit.

If the module's configuration is changed while this bit is set, i.e. the PCM to PWM conversion is ongoing; the changes take effect immediately and may lead to audible noise. Depending on the type of changes the reprogramming may also cause misbehavior on the module's other hardware interfaces. E.g. the DMA transfer protocol may be violated and thus a DMA error condition might be caused.

Bit	Description
0	PCMPWM is disabled; if the module is switched off via this bit, the PWM output stops immediately and the outputs are set to their inactive values; no interrupts and DMA transfers can be requested
1	PCMPWM is enabled

4.2. PCMPWM Output Control Register (PCMPWMi_OCTRL)

This Output Control Register is used to configure the individual output channels of the PCMPWM module.

REGISTER_NAME	PCMPWMi_OCTRL
OFFSET	0x00000004
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved						LEVL1	LEVL0
ACCESS_TYPE	R0,WX						R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0						1	1

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved						EN1	EN0
ACCESS_TYPE	R0,WX						R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0						0	0

[bit17] LEVL1 : Output Level Select Channel #1

This bit selects the output polarity of channel #1.

Bit	Description
0	PCMPWM_1_AH/BH are low-active and PCMPWM_1_AL/BL are high-active
1	PCMPWM_1_AH/BH are high-active and PCMPWM_1_AL/BL are low-active

[bit16] LEVL0 : Output Level Select Channel #0

This bit selects the output polarity of channel #0.

Bit	Description
0	PCMPWM_0_AH/BH are low-active and PCMPWM_0_AL/BL are high-active
1	PCMPWM_0_AH/BH are high-active and PCMPWM_0_AL/BL are low-active

[bit1] EN1 : Output Enable Channel #1

This bit enables or disables PWM signal generation on channel #1.

Bit	Description
0	PCMPWM_1_AH/BH/AL/BL are set to the inactive value
1	The PWM signal is enabled on channel #1

[bit0] EN0 : Output Enable Channel #0

This bit enables or disables PWM signal generation on channel #0.

Bit	Description
0	PCMPWM_0_AH/BH/AL/BL are set to the inactive value
1	The PWM signal is enabled on channel #0

4.3. PCMPWM Clock Select Register (PCMPWMi_CLKSEL)

This Clock Select Register is used to configure the PCMPWM conversion counter clock frequency.

REGISTER_NAME	PCMPWMi_CLKSEL
OFFSET	0x00000008
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved						CLK_SEL[1:0]	
ACCESS_TYPE	R0,WX						R/W	
PROT_TYPE	Wp							
INITIAL_VALUE	0						0	

[bit1:0] CLK_SEL : Clock Select

This field selects the PCMPWM conversion counter clock.

Bits	Description
00	Divide by 1: Conversion counter clock = PWM_CLK
01	Divide by 2: Conversion counter clock = PWM_CLK/2
10	Divide by 4: Conversion counter clock = PWM_CLK/4
11	Divide by 8: Conversion counter clock = PWM_CLK/8

4.4. PCMPWM Count Period Register (PCMPWMi_COUNTP)

This Count Period Register is used to configure the PCMPWM conversion counter period.

REGISTER_NAME	PCMPWMi_COUNTP
OFFSET	0x0000000C
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	COUNTP[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	Wp							
INITIAL_VALUE	1							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	COUNTP[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Wp							
INITIAL_VALUE	1							

[bit15:0] COUNTP : Count Period

This field defines the value of the PCMPWM conversion counter after which it continues to count from 0.

Setting of this field to 0 is not allowed.

4.5. PCM Offset Register (PCMPWMi_PCMOFFS)

This PCM Offset Register is used to configure the offset value used in the PCM signed to unsigned conversion.

REGISTER_NAME	PCMPWMi_PCMOFFS
OFFSET	0x00000010
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	PCM_OFFS[15:8]							
ACCESS_TYPE	R/W							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	PCM_OFFS[7:0]							
ACCESS_TYPE	R/W							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

[bit15:0] PCM_OFFS : PCM Offset

This field defines the offset value used in the PCM signed to unsigned conversion in low-pass filter and simplified H-bridge mode.

4.6. PCM Interrupt Enable Register (PCMPWMi_INTREN)

This PCM Interrupt Enable Register is used to enable or disable individual interrupts of the PCMPWM module.

REGISTER_NAME	PCMPWMi_INTREN
OFFSET	0x00000014
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved				DMA_ERR	UDRN	OVFL	DREQ
ACCESS_TYPE	R0,WX				R/W	R/W	R/W	R/W
PROT_TYPE	Wp							
INITIAL_VALUE	0				0	0	0	0

[bit3] DMA_ERR : DMA Block Error

This bit enables or disables the DMA block error interrupt.

Bit	Description
0	The DMA block error interrupt is disabled
1	The DMA block error interrupt is enabled

[bit2] UDRN : FIFO Under-Run Error

This bit enables or disables the FIFO under-run error interrupt.

Bit	Description
0	The FIFO under-run error interrupt is disabled
1	The FIFO under-run error interrupt is enabled

[bit1] OVFL : FIFO Overflow Error

This bit enables or disables the FIFO overflow error interrupt.

Bit	Description
0	The FIFO overflow error interrupt is disabled
1	The FIFO overflow error interrupt is enabled

[bit0] DREQ : Data Request

This bit enables or disables the data request interrupt.

Bit	Description
0	The FIFO data request interrupt is disabled
1	The FIFO data request interrupt is enabled

4.7. PCM Interrupt Status Register (PCMPWMi_INTRSTAT)

This PCM Interrupt Status Register reflects the status of the individual interrupt sources of the PCMPWM module. The bits in this registers are "sticky", i.e. once they have been set from the hardware, they remain set until they are reset via a write access to the PCMPWMi_INTRCLR register.

REGISTER_NAME	PCMPWMi_INTRSTAT
OFFSET	0x00000018
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WN							
PROT_TYPE	WN							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WN							
PROT_TYPE	WN							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WN							
PROT_TYPE	WN							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved				DMA_ERR	UDRN	OVFL	DREQ
ACCESS_TYPE	R0,WN				R,WN	R,WN	R,WN	R,WN
PROT_TYPE	WN							
INITIAL_VALUE	0				0	0	0	0

[bit3] DMA_ERR : DMA Block Error

This status bit reflects if a DMA block error has occurred.

Bit	Description
0	No error
1	A DMA block error has occurred, i.e. there were more data transfers to the FIFO buffer than specified by PCMPWMi_CONTROL:FEST + 1

[bit2] UDRN : FIFO Under-Run Error

This status bit reflects if a FIFO buffer under-run has occurred.

Bit	Description
0	No error
1	A FIFO buffer under-run has occurred, i.e. the PCM to PWM conversion block tried to read another PCM data sample, but the FIFO buffer is empty

[bit1] OVFL : FIFO Overflow Error

This status bit reflects if a FIFO buffer overflow has occurred.

Bit	Description
0	No error
1	A FIFO buffer overflow has occurred, i.e. there was a write access to the FIFO buffer when there was no more space available

[bit0] DREQ : Data Request

This status bit reflects if there is space for another n PCM data samples in the FIFO buffer. The number n is defined by PCMPWMi_CONTROL:FEST + 1.

Bit	Description
0	No data request: There are less than PCMPWMi_CONTROL:FEST + 1empty entries available in the FIFO buffer
1	Data request: There are PCMPWMi_CONTROL:FEST + 1 or more empty entries available in the FIFO buffer

4.8. PCM Interrupt Clear Register (PCMPWMi_INTRCLR)

This PCM Interrupt Clear Register is used to clear individual interrupts of the PCMPWM module.

REGISTER_NAME	PCMPWMi_INTRCLR
OFFSET	0x0000001C
ACCESS_SIZE	B, H, W
MULTIPLE	0
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved							
ACCESS_TYPE	R0,WX							
PROT_TYPE	Wp							
INITIAL_VALUE	0							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved				DMA_ERR	UDRN	OVFL	DREQ
ACCESS_TYPE	R0,WX				R0/W	R0/W	R0/W	R0/W
PROT_TYPE	Wp							
INITIAL_VALUE	0				0	0	0	0

[bit3] DMA_ERR : DMA Block Error

When this bit is written as "1", the DMA block error interrupt is cleared.

Bit	Description
0	Leave the DMA block error interrupt unchanged
1	Writing a "1" to this bit clears the DMA block error interrupt. This bit is always read as "0".

[bit2] UDRN : FIFO Under-Run Error

When this bit is written as "1", the FIFO under-run error interrupt is cleared.

Bit	Description
0	Leave the FIFO under-run error interrupt unchanged
1	Writing a "1" to this bit clears the FIFO under-run error interrupt. This bit is always read as "0".

[bit1] OVFL : FIFO Overflow Error

When this bit is written as "1", the FIFO overflow error interrupt is cleared.

Bit	Description
0	Leave the FIFO overflow error interrupt unchanged
1	Writing a "1" to this bit clears the FIFO overflow error interrupt. This bit is always read as "0".

[bit0] DREQ : Data Request

When this bit is written as "1", the data request interrupt is cleared.

Bit	Description
0	Leave the data request interrupt unchanged
1	Writing a "1" to this bit clears the data request interrupt. This bit is always read as "0".

4.9. PCM Data Register 0..15 (PCMPWMi_DATA)

This PCM Data Register used to write PCM data samples to the PCMPWM module.

REGISTER_NAME	PCMPWMi_DATA
OFFSET	0x00000040
ACCESS_SIZE	W
MULTIPLE	0:15
NUMERIC_TYPE	-
OTHER	-

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	DATA1[15:8]							
ACCESS_TYPE	RN,W							
PROT_TYPE	RN							
INITIAL_VALUE	-							

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	DATA1[7:0]							
ACCESS_TYPE	RN,W							
PROT_TYPE	RN							
INITIAL_VALUE	-							

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	DATA0[15:8]							
ACCESS_TYPE	RN,W							
PROT_TYPE	RN							
INITIAL_VALUE	-							

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DATA0[7:0]							
ACCESS_TYPE	RN,W							
PROT_TYPE	RN							
INITIAL_VALUE	-							

[bit31:16] DATA1 : Data Input Channel #1

This field holds the PCM data for channel #1. In mono mode the data is used for channel #0 as well.

[bit15:0] DATA0 : Data Input Channel #0

This field holds the PCM data for channel #0. In mono mode the data is used for channel #1 as well.

CHAPTER 34: LCD Bus Interface (LCDBusIF)

1. Preface

1.1 Purpose and Scope

This manual explains the functions and configurations of Cypress' LCDBusIF. It is intended for engineers engaged in the actual development of related products.

The LCDBusIF is an interface IP to communicate to external LCD controllers with an embedded graphic ram (GRAM). Its feature set is targeted to simplify the process of setting the LCD controllers configuration and updating the display content. It is applicable for Consumer, Industrial and Automotive market.

Note that there might be use cases documented in this manual that cannot be used in the context of a certain MCU, for example, because not all HW ports are connected to external pins or to the main system. So it is essential to check the related MCU manual accordingly.

1.2 Document Structure

The manual is divided into two main sections:

- Chapter **Function** contains pure functional descriptions only. It is a detailed documentation from feature point of view, without any details regarding setup and configuration.
- Chapter **Application** is a guideline for SW programmers. It describes which sub modules to setup and how to configure them for certain use cases. Also it contains a complete register list for all components (SW interface), interrupt descriptions and the address map.

Both chapters only cover aspects specific to this LCDBusIF IP. The related use cases, however, also require certain configurations in the embodying MCU system, for example, clock generation and pin multiplexing. For these refer to the relevant MCU manual.

2. Function

2.1 Feature Summary

The LCDBusIF implements the following features:

- Interfaces to LCD controller ICs with an Intel-8080 or Motorola-6800 compatible LCDBus[1] with an 8-, 9-, 16- or 18-bit wide data bus
- Direct link from Iris graphics core (if available in the system)
  - ☐ Wide range of input color formats (RGBA, Grayscale, Color Indexed or Compressed)
  - ☐ Memory efficient (sparse) scene composition
- Data master interface to read and write pixel or read command information from system memories
- Color lookup table to support memory efficient indexed color formats
- LCD display and configuration access is controlled via a programmable command sequencer to load off the CPU

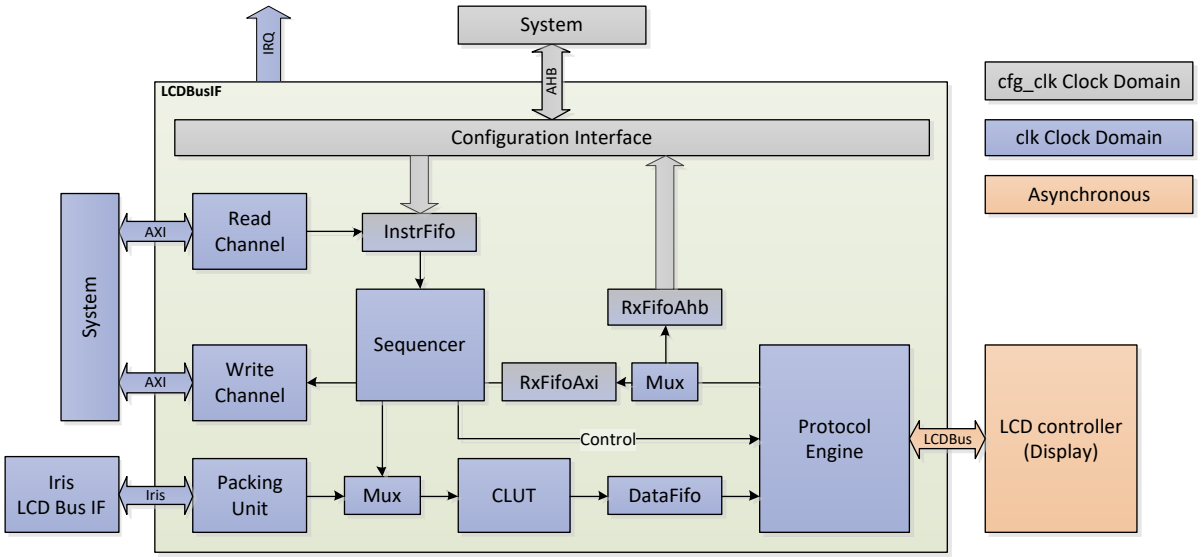
[1] In this document the parallel interface towards the LCD controller is refered to as LCDBus.

- AHB slave interface for configuration, command lists and pixel data
- Programmable interrupts (embedded in command lists) for synchronization with the CPU
- Tearing effect (VSYNC) interrupt (if supported by display)
- Support for different display pixel formats (like RGB666, RGB565, RGB444, GREY, B/W)

2.2 Block Diagram

In Figure 1 the LCDBusIF together with its external interfaces is depicted. Inside the conceptual structure and data flow is shown. The colors illustrate the different clock domains.

Figure 1: LCDBusIF Block Diagram



2.3 Functional Limitations

In Table 1 the Functional Limitations of the LCDBusIF are listed.

Table 1: Functional Limitations

Maximum resolution	WQVGA (480x272)
Maximum write cycle[2] frequency	clk/3 (clock frequency of the module divided by 3)

Additionally to the limitations from Table 1 certain display features can't be mapped with the current architecture:

- Any color format which requires color components from more than one pixel being transferred in one external interface word (*two pixel every three cycles mode*, used for example for RGB666 over 16-bit or RGB444 over 8-bit by some displays) won't be supported when pixel data is received from Iris graphics core.
- Some displays require manual increment of page address during content update (they only increment the column address). These displays won't be supported when pixel data is received from Iris graphics core.
- Additional address lines (to access GPIO or status information) offered by some displays are not supported.

[2] A write cycle refers to the minimum time between two write accesses on the LCD Bus and is usually defined in the AC characteristic of the display specification.

2.4 Nomenclature

The market offers a wide variety of LCD controllers with an equally high variation of naming conventions. This section should define the nomenclature used throughout this document and maps it to names found on the market.

- **LCDBus:** describes the parallel data and control interface towards the LCD controller. The LCD controllers usually refers to it as the *Host Interface*. The LCDBus can adhere to different protocols. The LCDBusIF supports the Intel-8080 (I80) and Motorola-6800 (M68) protocols.
- **GRAM:** the Graphic RAM describes the memory in the LCD controller which stores the pixel data shown on the display.
- **Iris:** is the nickname of the 2-D graphics core integrated into many MCUs provided by Spansion.
- **Index Register:** the LCDBus protocols implement an indirect addressing where the address is first set with a command write access and the actual register contents are accessed with one or more data accesses subsequently. The Index Register is the name of the register storing the address.
- **Status Register:** some LCD controllers provide a status register which can be read with a command read access.
- **Data Register:** while the Index Register refers to the address of the current access, the Data Register refers to the contents referenced by the Index Register.
- **Tearing Effect:** synchronization signal some LCD controller provide to synchronize the GRAM update with the start of the vertical blanking period of their internal timing controller. This should allow to avoid tearing artefacts on the display due to changing GRAM contents.
- **Signal Names:** while the LCDBus protocols define clear roles for the signals involved, the naming of them is not standardized and variations exist. Especially the active low property is encoded by different means (SIG#, SIGx, xSIG, SIGn, nSIG, $\overline{\text{SIG}}$).In this specification the number sign (#) is used to encode an active-low property. Table 2 lists the signal names used in this document together with some aliases seen in LCD controller specifications.

Table 2: Signal Names and Typical Aliases Found in LCD Controller Specifications

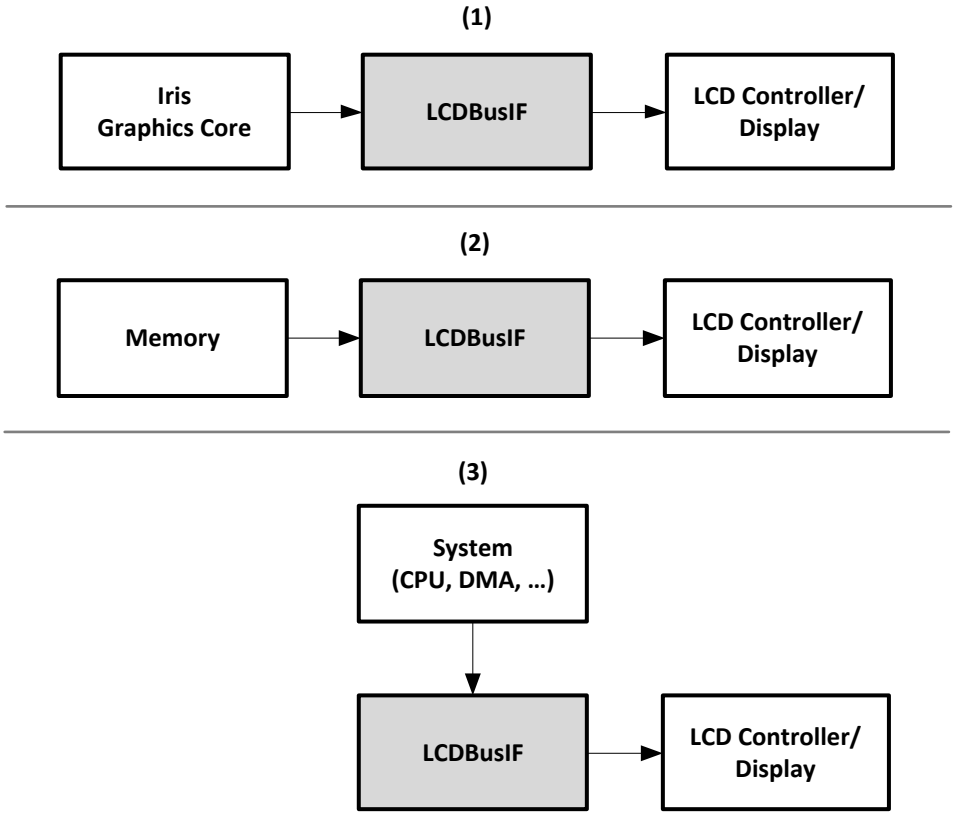
Signal	Description	LCD Controller Pin Names	Protocols
CS#	Chip select (active low)	nCS, CSn, CSx, XCS, CS#	I80, M68
WR#	Write select (active low)	nWR, WRn, WRx, XWR, WR#	I80
R/W#	Read/Write select (read = high, write = low)	R/Wn, R/Wx, R/W#	M68
RD#	Read select (active low)	nRD, RDn, RDx, XRD, RD#	I80
E	Enable	E	M68
D/C#	Data/Command select (data = high, command = low)	A0, RS, P/C#, D/C#	I80, M68
DB	Data Bus	DB, D, DAT	I80, M68
TE	Tearing Effect	TE, FMARK	I80, M68
RES#	Reset (active low)	RESET, RESX, XRES, RES#	I80, M68

2.5 Use Cases

The LCDBusIF acts as communication interface towards external LCD controllers. As such it allows updating the display content, but it is also required to set configuration register or read status registers of the LCD controller. The main use cases differentiate by the source of the display content, see Figure 2.

1. Content from Iris Graphics Core: the display scene is composed by the Iris graphics core, converted to the image format of the LCD controller and written to it. This is the preferred use case for systems having an Iris graphics core.
2. Content from Memory: the display scene is already available in memory in the data format required by the LCD controller (e.g. rendered by software). A read channel transfer is setup to fetch the display scene and forward it to the LCD controller. This is the preferred use case for systems not having an Iris graphics core.
3. Content from system: the display scene is written to the LCDBusIF from any system master. This could for example be the CPU rendering content on the fly or a DMA copying an available display scene from memory.

Figure 2: Use Cases from a Data Flow Perspective



2.6 Protocol Engine

The protocol engine of the LCDBusIF is designed to support the different protocol flavors and characteristics as they are used by LCD controllers on the market. Their variance is normally determined by the following three factors:

1. Protocol Types: the LCDBusIF supports Intel-8080 (sometimes also called System-80) and Motorola-6800 kind of interfaces with different data widths and flavors regarding the read/write strobe signaling.
2. Access Timing: above mentioned protocols are all asynchronous, so to guarantee a stable communication timing characteristics need to be considered. The LCDBusIF implements these timing requirements by generating the output signals with a raster defined by module clock cycles.
3. Data Mapping: depending on the available LCDBus data width and the pixel format used internally the LCD controller might require several access cycles to read/write one pixel (e.g. an RGB565 pixel is transferred using 2 access cycles on an 8-bit interface)

The subsequent sections will detail how the LCDBusIF handles these variations.

2.6.1 Protocol Types

The LCDBusIF can support different LCDBus interface configurations. It supports Intel-8080 and Motorola-6800 type interfaces of different data bus widths. Additionally can either the RD#/WR#/E (Type 1) or the CS# (Type 2) signals be used as strobe signal while the other functions as select signal. Figure 3 and Figure 4 illustrate the different LCDBus protocol variants supported. See Table 3 for the different configuration options available. Note the different interpretation of the read/write interface signals depending on the interface dialect. Additionally the polarity of each interface signal can be configured to support LCD controllers having a special interpretation of the protocols.

Table 3: Interface Configuration Options

Interface	Variant	Strobe signal(s)	Select signal(s)
Intel-8080	Type 1	WR#, RD#	CS#
	Type 2	CS#	WR#, RD#
Motorola-6800	Type 1	E	CS#
	Type 2	CS#	E

Figure 3: Intel-8080 LCDBus Protocol Variants

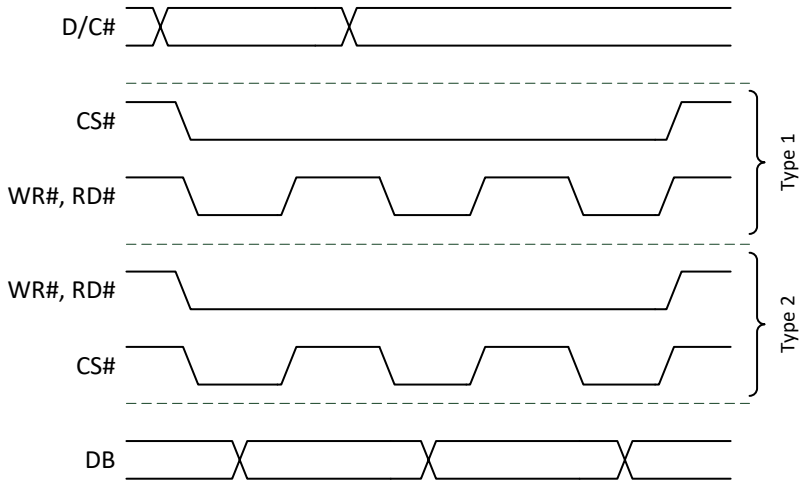
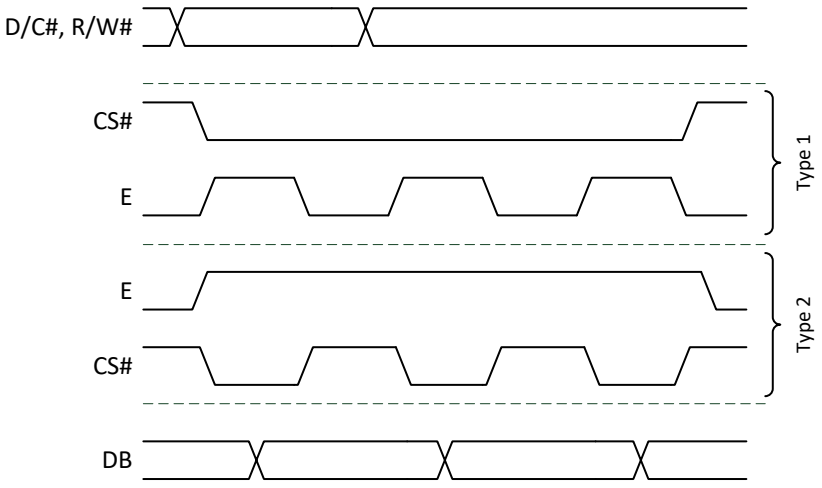


Figure 4: Motorola-6800 LCDBus Protocol Variants



2.6.2 Access Timing

The LCD controllers have different requirements for read and write timing usually described in the AC characteristics section of their specification. The LCDBusIF allows to adhere to these timing requirements by implementing a read and write cycle timing scheme. Within each cycle three windows are defined, the setup window, the active window and the hold window. The width of the window is defined in number of module clock cycles. The windows have the following functional meanings:

- Setup window: activate select, data/command and read/write select signals; control output enable according to the access direction
- Active window: update data bus for write access; active period for strobe signal
- Hold window: register data bus for read access; hold data bus and output enable for write access

Figure 5 below illustrates the timing for a command and data write sequence and Figure 6 the timing for a command write and data read sequence.

Note that the maximum pixel clock the LCDBusIF can support without causing tearing artefacts depend on the module clock and how well it can be matched to the required interface timing.

Figure 5: Illustration of Command/register Write Sequence

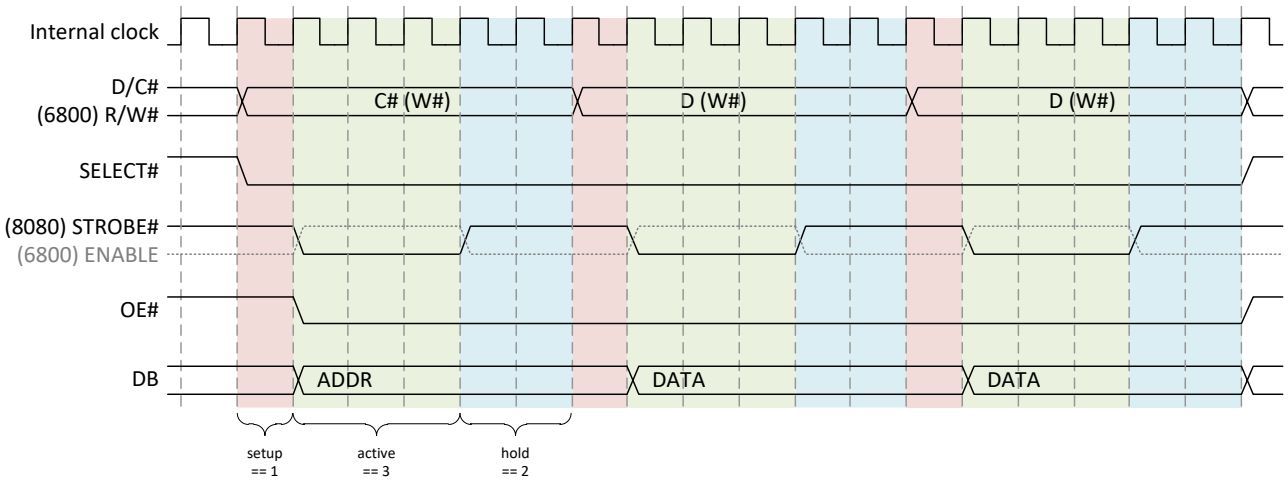
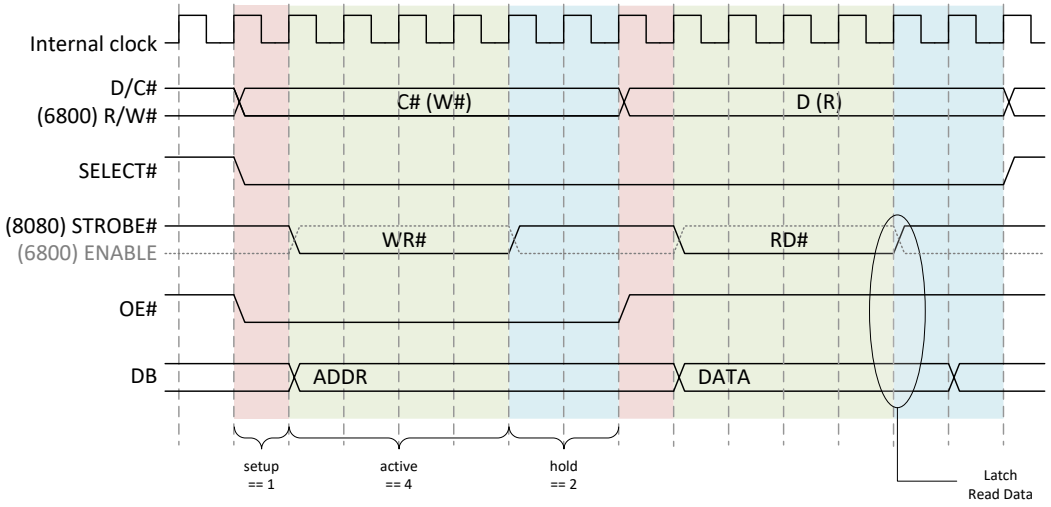


Figure 6: Illustration Command Write/register Read Sequence



2.6.3 Data Mapping

The mapping configuration instructs the LCDBusIF how to transfer a given command or data word with the given interface width. Distinct transfer mappings can be programmed for a data or a command transfer. The LCD controller might have a native command/data width which is not equal to the interface width. Hence several transfers might be required (if the interface width is smaller than the command/data width) or the data might need to be properly aligned (if the interface width is bigger than the command/data width).

In Table 4 the mapping of different color formats depending on the interface width is illustrated. The number signifies the number of transfers required to transport one pixel via the corresponding interface width (the numbers in parenthesis should indicate that such a combination of interface width and color format is relatively unlikely).

Table 4: Correlation between Interface Width and Color Format

Interface Width / Color Format	18-bit	16-bit	9-bit	8-bit
Black/White	(1)	(1)	(1)	1
Gray 4-bit	(1)	(1)	(1)	1
Gray 8-bit	(1)	(1)	(1)	1
RGB565	1	1	2	2
RGB666	1	2	2	3

Most LCD controllers have different requirements for access of their internal registers and access of the GRAM. Therefore the LCDBusIF provides one set of mapping registers for each access type (named command access and data access respectively).

The mapping is implemented as such that one command/data word (source) can be spread over up to three LCDBus (destination) transfers. For every transfer one register defines a vector to be extracted from the source and put into the destination transfer. The source vector is determined by a bit offset (slsb) and a number of bits (sbits). The position within the destination transfer is determined by another bit offset (dlsb). An additional bit (next) indicates if a subsequent destination transfer is required for this command/data word (next = 1) or if the protocol engine is done with it (next = 0). Figure 7 illustrates different mapping configurations to transport an RGB666 word via different interface widths.

For read transfer mapping the interpretation of source and destination needs to be swapped, i.e. dlsb will select the start bit of the read transfer received, dbits will define how many bits to extract and slsb will define where the data should be placed into the reception fifo word.

Figure 7: Different Mapping Scenarios for an RGB666 Color Format via Different Interface Widths



2.7 Iris Graphics Core Interface

The Iris graphics core, if available in the system, already provides a strong feature set to compose a display scene in a high quality and resource efficient way. In order to benefit from this the LCDBusIF offers an interface to the Iris graphics core that allows to request frames and convert the pixel information received into the color format required by the LCD controller.

2.7.1 Handshake

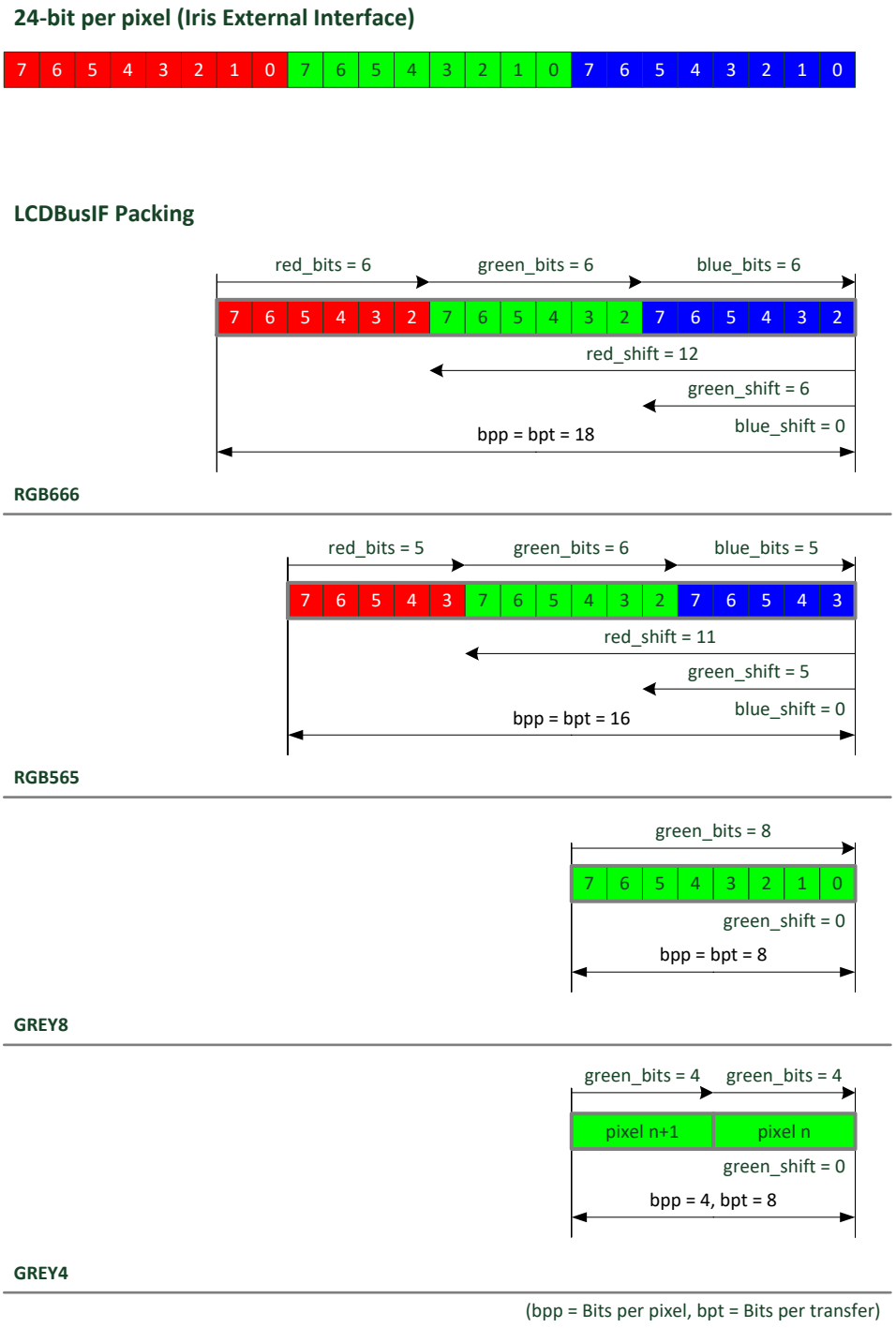
The image source from the Iris graphics core can be selected among the available content or safety streams in the system. The selected stream must be setup to compose the desired display scene with the correct frame dimension before the LCDBusIF should update the display content. The frame transaction is initiated from the LCDBusIF by sending a frame request (see IRIS_KICK instruction) to the Iris. Subsequently the LCDBusIF might stall the Iris until the display update is initiated (see IRIS_FRAME instruction). The Iris graphics core must not send any data without a frame request.

2.7.2 Data Packing

The packing configuration instructs the LCDBusIF how to transform the RGBA data received from the Iris interface to the color format required by the LCD controller. Input color is interpreted as 24-bit RGB. The configuration will define how many bits of each color component will be used (starting from most significant bit) and at which position it should be mapped into the transfer word.

For color formats requiring less bits per pixel as available by the interface width multiple pixels can be packed into one transfer. This is often required for B/W or grayscale displays. A transfer in this context specifies an entry in the DataFifo. The LCDBusIF will pack as many complete pixel as possible into one transfer. Figure 8 illustrates some common packing configurations.

Figure 8: Packing Configurations for Different LCD Color Formats



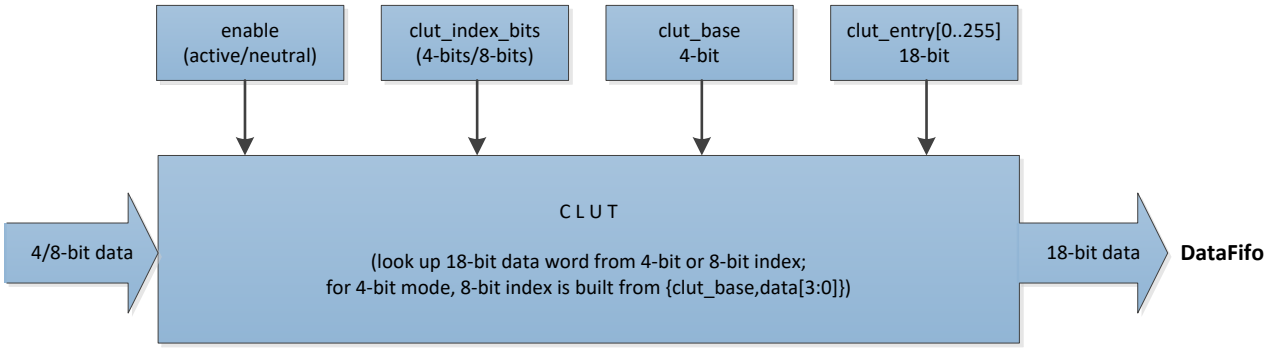
2.8 Color Lookup Table

The LCDBusIF provides a color lookup table (CLUT) to support indexed color formats allowing a reduced memory footprint of the display framebuffer. The index address size can either be 4-bit or 8-bit, so that 16 or 256 different colors can be

addressed. For an index address size of 4-bit, 16 separate color tables can be stored in the CLUT. The active color table can be selected by a sequencer instruction.

The color lookup table if enabled will be active on data from a WR_PIXEL or IRIS_FRAME instruction only.

Figure 9: Conceptual Overview of the CLUT Function



2.9 Command Sequencer

The operation of the LCDBusIF is controlled by a programmable command sequencer. A set of instructions are provided to allow the sequencer to synchronize and control the data flow between CPU, Iris graphics core and the LCD controller.

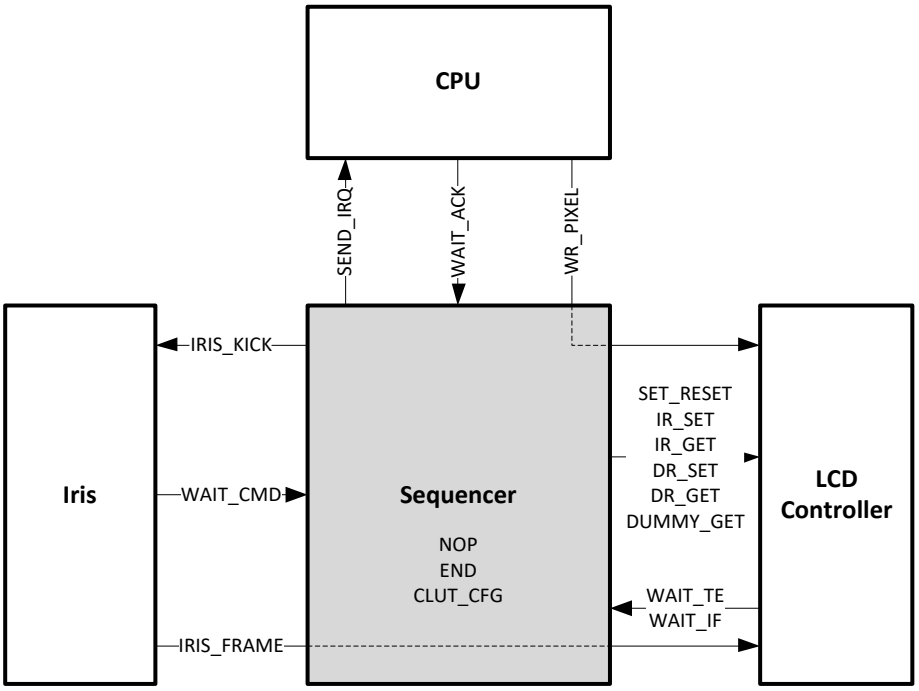
Table 5 illustrates the instruction format of the command sequencer. Each instruction is a 32-bit word whose most significant byte defines the type (opcode) of the instruction. Additional parameters can be encoded in the lower 3 bytes. Most instructions are self contained in a single 32-bit word, only the WR_PIXEL instruction is followed by a list of 32-bit words which contain the pixel data to be written to the LCD controller.

Table 5: Command Sequencer Instruction Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								Parameters																							
Optional: pixel data																															

In Figure 10 the different instructions with their relation to the CPU, the Iris graphics core and the LCD controller are shown. Arrows to the sequencer indicate that the sequencer is synchronizing to an event from this source, arrows from the sequencer indicate that the sequencer is sending a request to this destination and arrows through the sequencer indicate the routing of the pixel stream (either from CPU or Iris graphics core) to the LCD controller.

Figure 10: Conceptual Overview of Sequencer Interaction with CPU, Iris and LCD Controller



In Table 6 the supported instructions together with their opcode, parameters and description are summarized. See Instruction Set for a more elaborate description.

Table 6: Summary of Instructions Supported by Sequencer

Instruction	Opcode	Description	Parameter(s)
NOP/WAIT	0x00	Wait cycle(s) (module clock)	count[23:0]
IR_SET	0x10	Write index register	addr[17:0]
IR_GET	0x11	Read status register	count[17:0]
DR_SET	0x12	Write data register	data[17:0], type[cmd,data]
DR_GET	0x13	Read data register	count[17:0], type[cmd,data]
DUMMY_GET	0x14	Dummy read	
WAIT_TE	0x15	Wait for sync input from display	
WAIT_IF	0x16	Wait until protocol engine is idle	
SET_RESET	0x1F	Set LCDBus RES# signal value	level[logic0, logic1]
SEND_IRQ	0x20	Send synchronization interrupt to CPU	
WAIT_ACK	0x21	Wait for acknowledge from CPU	
IRIS_KICK	0x40	Send kick (frame request) to Iris	
WAIT_CMD	0x41	Wait for command word at Iris interface	
IRIS_FRAME	0x42	Forward pixel data from Iris interface	

Instruction	Opcode	Description	Parameter(s)
WR_PIXEL	0x80	Initiate writing of pixel data via instruction fifo	bpp[2:0], num_pixel[17:0], list of pixel[31:0]
CLUT_CFG	0xC0	Set CLUTconfiguration	enable, index_bits[3:0], clut_base[3:0]
END	0xFF	End of command list	

2.9.1 Instruction Set

The following sections will discuss each instruction individually.

2.9.1.1 NOP/WAIT

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00								Count																							

The NOP instruction doesn't have any side-effects but instructs the sequencer to wait a given number of module clock cycles before processing the next instruction. This allows implementing wait time requirements between the executions of other instructions.

Parameters:

- *Count*: (count + 1) number of module clock cycles to wait before loading next instruction.

The following formula can be used to calculate a required wait time:

Count = <WaitTime[s]> * <ModuleClockFreq[Hz]> – 1

For example with a given module clock frequency of 100 MHz and a required wait period of 50 ms the following NOP count should be programmed:

Count = 50×10^{-3} * 100×10^{6} – 1 = 5×10^{6} – 1 = 0x4c4b3f

2.9.1.2 IR_SET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10														Addr																	

The index register set instruction will generate an LCDBus write transfer with the D/C# signal set to command. The addr parameter will be driven on the data pins.

Note:

The LCDBusIF will drive the Addr on the data pins as is, so if the LCD controller requires a special mapping for the address (e.g. {db[17:10],db[8:1]} gap assignment) the SW needs to properly convert the Addr. If the Addr to be transferred is wider than the interface width, two IR_SET commands need to be chained to transport the upper and lower part.

Parameters:

- *Addr:* address of the index register of the LCD controller. Will be mapped as command transfer.

2.9.1.3 IR_GET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x11														Count																	

The status register get instruction will generate (count+1) number of LCDBus read transfers with the D/C# signal set to command.

Parameters:

- *Count*: set to number of read transfers to generate minus one (e.g. count == 0 will result in 1 read transfer, count == 3 will result in 4 read transfers).

Note:

The read data will be mapped as command transfer.

2.9.1.4 DR_SET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x12								Type						Data																	

The data register set instruction will generate an LCDBus write transfer with the D/C# signal set to data.

Parameters:

- *Data*: the data to be sent via the LCDBus.
- *Type*: select if data should be mapped as command (Type = 1) or data (Type = 0) transfer.

Note:

Data written with Type = 0 will not be considered by the CLUT function but written as is.

2.9.1.5 DR_GET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x13								Type						Count																	

The data register get instruction will generate (count+1) number of LCDBus read transfers with the D/C# signal set to data.

Parameters:

- *Count*: set to number of read transfers to generate minus one (e.g. count == 0 will result in 1 read transfer, count == 3 will result in 4 read transfers).
- *Type*: select if read data should be mapped as command (Type = 1) or data (Type = 0) transfer.

2.9.1.6 DUMMY_GET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14																															

The dummy get instruction will generate a single read cycle on the LCDBus, but discard the read data. This allows discarding the data a lot of LCD controllers return upon the first read cycle of a read burst.

2.9.1.7 WAIT_TE

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x15																															

Wait for a rising edge of the LCD controllers tearing effect output.

2.9.1.8 WAIT_IF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x16																															

Wait for protocol engine to finish all pending transactions.

2.9.1.9 SET_RESET

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1F																															Level

Control the level of the external reset line. After reset the external reset line will have a low signal level (assuming most displays will have reset input that is active low).

Parameters:

- *Level*: 0 = drive logic 0; 1 = drive logic 1

2.9.1.10 SEND_IRQ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x20																															

Send an interrupt pulse (SequencerSyncInterrupt, see Interrupt Map) to the CPU; can be used for synchronization with software.

2.9.1.11 WAIT_ACK

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x21																															

Wait for an acknowledge trigger from the CPU before continuing to the next instruction; can be used for synchronization with software in conjunction with the SEND_IRQ instruction.

2.9.1.12 IRIS_KICK

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40																															

Send a kick pulse (frame request) on the Iris Interface.

Note:

Every kick sent must result in a frame accepted. So as a requirement, every IRIS_KICK must be followed by an IRIS_FRAME command eventually. Only one kick can be pending. If an IRIS_KICK opcode is read while a kick is already pending, the command will be ignored and the sequencer will go into an error state and send a SequencerErrorInterrupt (see also Error Detection).

2.9.1.13 WAIT_CMD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x41																															

Wait until a command word has been received from the Iris Interface.

Note:

(1) requires that an IRIS_KICK is pending otherwise the sequencer will run into timeout or wait forever. (2) the command received status will be true as soon as the command word has been received until either the WAIT_CMD or the IRIS_FRAME instruction is read. This way it is not required that this instruction is read before the Iris interface command word is actually received.

2.9.1.14 IRIS_FRAME

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x42																															

This command will enable the transfer of pixel information towards the protocol engine. If the CLUT is enabled iris pixels (after packing) will be used as indices to the lookup table.

Note:

Requires that an IRIS_KICK is pending otherwise the sequencer will run into timeout or wait forever.

2.9.1.15 WR_PIXEL

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x80									Bpp					NumPixel																	
PixelData																															

Initiate update of display content directly via the configuration interface. After this instruction the sequencer will interpret all subsequent entries as pixel data until NumPixel have been written to the data fifo. Using the Bpp parameter each PixelData word can contain multiple pixel. Every output pixel will be written to the data fifo as if a DR_SET instruction with Type = 0 would have been initiated. If the CLUT is enabled output pixels will be used as indices to the lookup table.

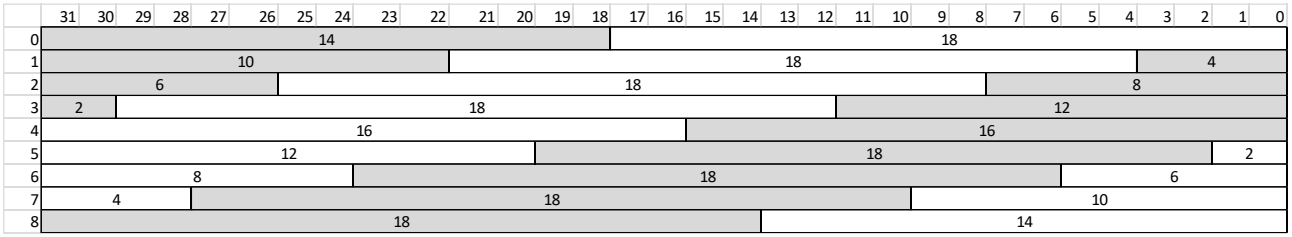
Parameters:

- *NumPixel*: number of output pixels to write to the data fifo minus one. SW is responsible to provide the required amount of input data depending of Bpp setting configured (see below).
- *Bpp*: data width of output pixels. The sequencer will extract 1..4 pixels from one PixelData word using a fixed unpacking scheme. Little endian organization is expected for pixel data. The following settings are supported:
  - ☐ 0: Extract 1 × 18-bit data from each instruction fifo entry ([17:0]). Number of required PixelData words is equal to NumPixel+1.
  - ☐ 1: Extract 2 × 16-bit data from each instruction fifo entry ([15:0], [31:16]). Number of required PixelData words is ceil((NumPixel+1)/2).
  - ☐ 2: Extract 4 × 8-bit data from each instruction fifo entry ([7:0], [15:8], [23:16], [31:24]). Number of required PixelData words is ceil((NumPixel+1)/4).
  - ☐ 3: Extract 8 × 4-bit data from each instruction fifo entry ([3:0], [7:4], [11:8], [15:12], [19:16], [23:20], [27:24], [31:28]). Number of required PixelData words is ceil((NumPixel+1)/8).

- ☐ 4: Extract 16 × 18-bit data from every 9 instruction fifo entries (see Figure 11). Number of required PixelData words is ceil((NumPixel+1)*9/16). NOTE: the number of pixels doesn't have to be dividable by 16.

- *PixelData*: packed according to Bpp setting. Number of words depending on Bpp setting.

Figure 11: Mapping for Tight 18-bit Unpacking



2.9.1.16 CLUT_CFG

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC0																							Enable				IndexBits		ClutBase		

Configure color lookup table operation.

Parameters:

- *Enable*: set to 1 to enable the color lookup table
- *IndexBits*: configure clut index mode
  - ☐ *0*: 8-bit data is used as index address
  - ☐ *1*: 4-bit data is used as index address together with ClutBase
- *ClutBase*: base address for 4-bit index mode. Actual clut address will be build from {ClutBase,pixel[3:0]}

2.9.1.17 END

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF																															

End of current command list. If no more instructions are left in the instruction fifo the sequencer will go into idle state.

2.9.2 Error Detection

The command sequencer can detect three kinds of error conditions:

- Illegal opcode error: an instruction with an unsupported opcode is read
- Kick error: a frame request (IRIS_KICK) instruction is read although there is already a frame request pending
- Timeout error: a timeout mechanism can be enabled to notify the software about unexpected long execution times. With this mechanism enabled, a timeout counter will be preloaded with the configured timeout value every time a new instruction is read. With each cycle an instruction remains active in the command sequencer the timeout counter is decremented. The error materializes if the counter reaches 0 before the next instruction is read.

In case any of the above mentioned error conditions occurs the command sequencer will stop processing, go into an error state and trigger the SequencerErrorInterrupt. The error condition and the context upon the error detection can be read from the register interface for analysis.

To summarize a probably incomplete list of situations that will result in an error:

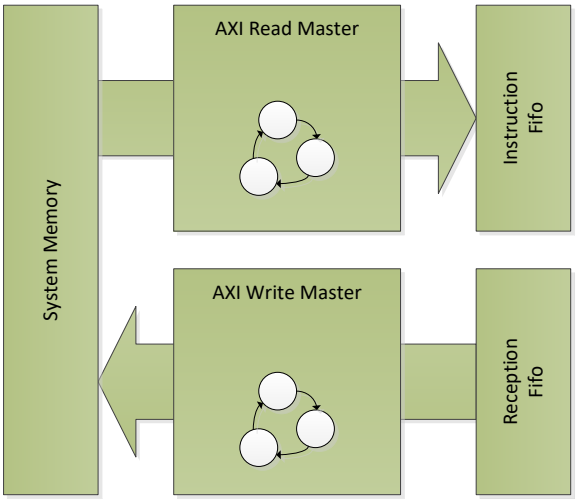
- Illegal opcode read by sequencer (will trigger an SequencerErrorInterrupt, ErrorCode = OPCODE)
- Sequencer timeout reached during execution (will trigger an SequencerErrorInterrupt, ErrorCode = TIMEOUT)
- An IRIS_KICK instruction read while a previous IRIS_KICK is still pending (will trigger an SequencerErrorInterrupt, ErrorCode = KICK)

- A WAIT_CMD or IRIS_FRAME instruction is pending without corresponding IRIS_KICK (will trigger an SequencerErrorInterrupt eventually if timeout mechanism is enabled, ErrorCode = TIMEOUT)
- A WAIT_CMD or IRIS_FRAME instruction is pending but no data is received due to misconfiguration of Iris (will trigger an SequencerErrorInterrupt eventually if timeout mechanism is enabled, ErrorCode = TIMEOUT)
- A WAIT_TE instruction is pending without an activated tearing effect signal (will trigger an SequencerErrorInterrupt eventually if timeout mechanism is enabled, ErrorCode = TIMEOUT)
- A WR_PIXEL instruction with insufficient pixel data (will trigger an SequencerErrorInterrupt eventually if timeout mechanism is enabled, ErrorCode = TIMEOUT)

2.10 Data Master Interface

The data master interface function will handle the filling and emptying of the instruction and reception fifo respectively, reducing the load on system masters. Compared to a DMA controller solution the data master interface reduces the access latency and the load on the system bus as the memory resource can be accessed directly (contrary to the DMA controller accessing first the memory resource and subsequently the LCDBusIF).

Figure 12: Conceptual Overview of the Data Master Interface Function



2.10.1 Read Channel Master

The read channel master is responsible to fill the instruction fifo. When enabled, the instruction fifo is no longer accessible from the configuration interface and any access from there will be terminated with an error response. The data read master takes a base address and the number of 32-bit words to read as parameters. The read process will be initiated by a software trigger. Burst requests will be emitted as long as there is enough space in the instruction fifo to accept the whole burst and there are still pending words to read. The default burst length can be configured. When the number of words to read is not an even multiple of the burst length, the length of the last burst will be reduced to number of remaining words. Once all configured words have been read an interrupt will be issued.

2.10.2 Write Channel Master

The write channel master is responsible to read pixel data from the reception fifo and store it to a system memory. The write channel master takes a base address and the pixel format (8-, 16- and 18-bits per pixel is supported) as parameters. If the write master is enabled the write process will be initiated by a DR_GET if its *Type* parameter is set to Data. At the beginning of the write process the base address of the destination buffer will be loaded from the configuration. Burst requests will be emitted as soon as there is enough data in the reception fifo to serve the whole burst and there are still pending pixels to write (*Count* parameter of DR_GET). The default burst length can be configured. When the number of pixels can't be packed into an even multiple of the burst length, the length of the last burst will be reduced to only contain the remaining pixels. Optionally the write master can pack multiple pixel into every 32-bit data word (supported for 8-, 16- and 18-bits per pixel). Once all pixels have been written to memory an interrupt will be issued.

Note that the write master can only be used to store pixel data in memory as it is not sensitive to any read instruction which is based on command data mapping. All other read data will still be accessible via the the reception fifo config interface.

2.11 Interrupt Controller

The IP provides a built-in interrupt controller with the following features for all relevant HW events:

- Enable bit (mask)
- Status bit (set by an HW event)
- Preset bit (can be used by SW to set status)
- Clear bit (used by SW to reset the status)

Each interrupt can be connected as IRQ (maskable) and/or NMI (non-maskable) in the embodying system. Alternatively the un-masked trigger signals for all HW events are provided, allowing it to use a global interrupt controller instead.

3. Application

3.1 Map Tables

3.1.1 Interrupt Map

The LCDBusIF provides the interrupt signals as described in Table 7.

Table 7: Interrupt Map

Name	Description
SequencerSyncInterrupt	Programmable sequencer interrupt for synchronization
SequencerErrorInterrupt	Sequencer Error Interrupt
InstrFifoInterrupt	Instruction Fifo Interrupt
RxFifoInterrupt	Reception Fifo Interrupt
TearingEffectInterrupt	Interrupt upon rising edge on tearing effect (TE) input
ReadChannelDone	Read Channel transfer complete
WriteChannelDone	Write Channel transfer complete

3.1.2 Key Map

Inside this IP all address blocks, that can be protected, use the same key values:

Table 8: Key Map

Name	Value	Function
Lock key	0x5651F763	Enable all access protection.
Unlock key	0x691DB936	Disable all access protection.
Privilege key	0xAEE95CDC	Enable non-privileged access protection.
Unprivileged key	0xB5E2466E	Disable non-privileged access protection.
Freeze key	0xFBE8B1E6	Freeze current protection status (cannot be changed any longer).

3.2 Getting Started

Assuming all configuration fields being in reset state, a minimal setup to initialize an exemplary LCD display and update its GRAM is:

- WriteTimingConfig:
  - WriteSetupCycles = 1
  - WriteActiveCycles = 2
  - WriteHoldCycles = 2
- ReadTimingConfig:
  - ReadSetupCycles = 1
  - ReadActiveCycles = 11 + 2
  - ReadHoldCycles = 6
- Command List (InstructionFifo), see Table 9
- Wait for SequencerSyncInterrupt or poll for OperationState to become IDLE

The example assumes a module clock period (tCLK) of 12.5 ns and an exemplary QVGA (320x240) LCD controller that has an 8-bit LCDBus implementing the Intel-8080 protocol (strobing via the read and write signals). Index register, configuration register and pixel width is 8-bit. The timing parameters should be assumed as in Table 10 and the command set as in Table 11.

Table 9: Example command list

Instruction	Instruction (hex)	Description
SET_RESET(0) NOP(3999) SET_RESET(1) NOP(799999)	0x1f000000 0x00000f9f 0x1f000001 0x000c34ff	Toggle LCD controller reset according to reset requirements.
IR_SET(0x11)	0x10000011	Wakeup LCD controller.
IR_SET(0x2a) DR_SET(0x00) DR_SET(0x10)	0x1000002a 0x12000000 0x12000010	Set column address.
IR_SET(0x2b) DR_SET(0x00) DR_SET(0x20)	0x1000002b 0x12000000 0x12000020	Set page address.
IR_SET(0x3a) DR_SET(0x01)	0x1000003a 0x12000001	Set color mode.
IR_SET(0x2c)	0x1000002c	Initiate GRAM write procedure.
WAIT_TE() WR_PIXEL(2, 76799) *Pixel Data* ...	0x15000000 0x80212bff *0xff00ff00* ...	Synchronize to tearing effect line and write pixel data to GRAM.
IR_SET(0x29)	0x10000029	Switch display on.
SEND_IRQ()	0x21000000	Notify software about command list completion.
END()	0xff000000	End of command list.

Table 10: AC Characteristics of an Fictional LCD Controller

Symbol	Description	Min	Max	Unit
tCW	Chip select setup time	20	–	ns
tAH tAS	Address hold time Address setup time	20 20	–	ns
tCYCW tCCHW tCCLW	Write cycle time Write pulse H width Write pulse L width	100 35 35	–	ns
tCYCR tCCHR tCCLR	Read cycle time Read pulse H width Read pulse L width	255 90 150	–	ns
tDS tDH	Write data set time Write data hold time	20 20	–	ns
tACC tOH	Read data access time Read data disable time	– 15	145 80	ns
tRW tRT	Reset pulse width Reset clear time	50 –	– 10	us ms

Table 11: Register Set of a Fictional LCD Controller

Register Name	Index Address	Parameters	Description
SLPOUT	0x11	–	Wake up controller.
CASET	0x2a	CA[7:0], CA[15:8]	Column address set.
PASET	0x2b	PA[7:0], PA[15:8]	Page address set.
COLMOD	0x3a	Mode	Color Mode select.
RAMWR	0x2c	–	Initiate GRAM update.
DISPON	0x29	–	Switch display on.

3.3 Display Setup

The following sections should describe how the LCDBusIF can be configured to match the LCD controller it is attached to. A display setup involves the interface configuration and optionally the packing configuration (color format selection) in case the Iris graphics core interface should be used.

3.3.1 Interface Configuration

From the perspective of the LCDBusIF the LCDBus towards an LCD controller is defined by the interface type, the interface timing and interface mapping. Each of these properties should be discussed in the following sections.

3.3.1.1 Interface Type

To select the LCDBus protocol the LCDBusIF should use to communicate with the LCD controller, the following setup should be done:

1. Select LCDBus protocol using InterfaceType
2. Configure the LCDBus width using InterfaceWidth
3. Optional: adapt the interface signal polarities if they are in conflict with the default polarities InterfaceConfig.<signal>polarity

3.3.1.2 Interface Timing

Figure 14 shows typical AC characteristics diagrams for both Intel-8080 and Motorola-6800 type of LCDBus. Overlaid the Setup, Active and Hold scheme as used by the LCDBusIF is depicted. To calculate the required timing configuration, the following formulas must hold both for read and for write (time based results must be rounded up to the next integer value):

$$\text{SETUP} = \left(\frac{t_{AW} + t_{UNCERT}}{t_{CLK}}\right)$$

$$\text{(read) ACTIVE} = \max\left(\frac{t_{ACC} + t_{OUT2IN}}{t_{CLK}}, \frac{t_{CCLR}}{t_{CLK}}\right)$$

$$\text{(write) ACTIVE} = \max\left(\frac{t_{DS} + t_{UNCERT}}{t_{CLK}}, \frac{t_{CCLW}}{t_{CLK}}\right)$$

$$\text{(write) HOLD} = \max\left(\frac{t_{AH}}{t_{CLK}}, \frac{t_{CCHW}}{t_{CLK}} - \text{SETUP}, \frac{t_{DH}}{t_{CLK}} - \text{SETUP}\right)$$

$$\text{(read) HOLD} = \max\left(\frac{t_{AH}}{t_{CLK}}, \frac{t_{CCHR}}{t_{CLK}} - \text{SETUP}\right)$$

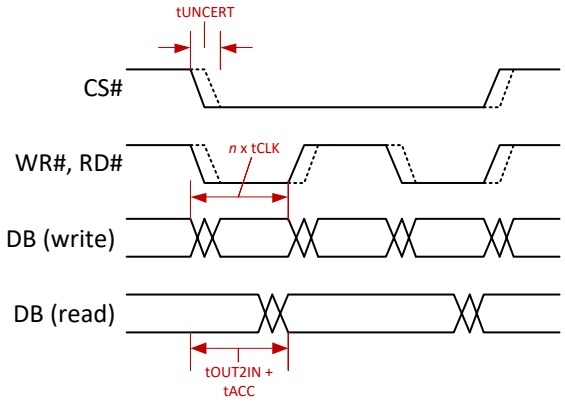
$$\text{SETUP} + \text{ACTIVE (read)} + \text{HOLD (read)} \geq t_{CYCR}$$
$$\text{SETUP} + \text{ACTIVE (write)} + \text{HOLD (write)} \geq t_{CYCW}$$

Where tCLK is the period of one module clock cycle, tUNCERT describes the error which results due to different routing delays of the depending signals and tOUT2IN describes the sum of maximum output and input delay. Both tUNCERT and

tOUT2IN are properties of the device embedding the LCDBusIF and need to be looked up in the AC characteristics of the device specification (see Figure 13 for an illustration of these error terms).

Note how in Figure 14 the SETUP configuration contributes to the HOLD timing window for the data signals. Therefore the HOLD configuration can be reduced by the SETUP time (but must be at least 1). Table 12 summarizes the LCDBus timing symbols as used in Figure 14.

Figure 13: AC Characteristics of the LCDBusIF from Device Point of View



The calculated Setup, Active and Hold values must be programmed to the WriteTimingConfig and ReadTimingConfig registers. Note that the intended value must be programmed with minus one.

Figure 14: Timing Specification for Intel-8080 and Motorola-6800 Interface Protocols

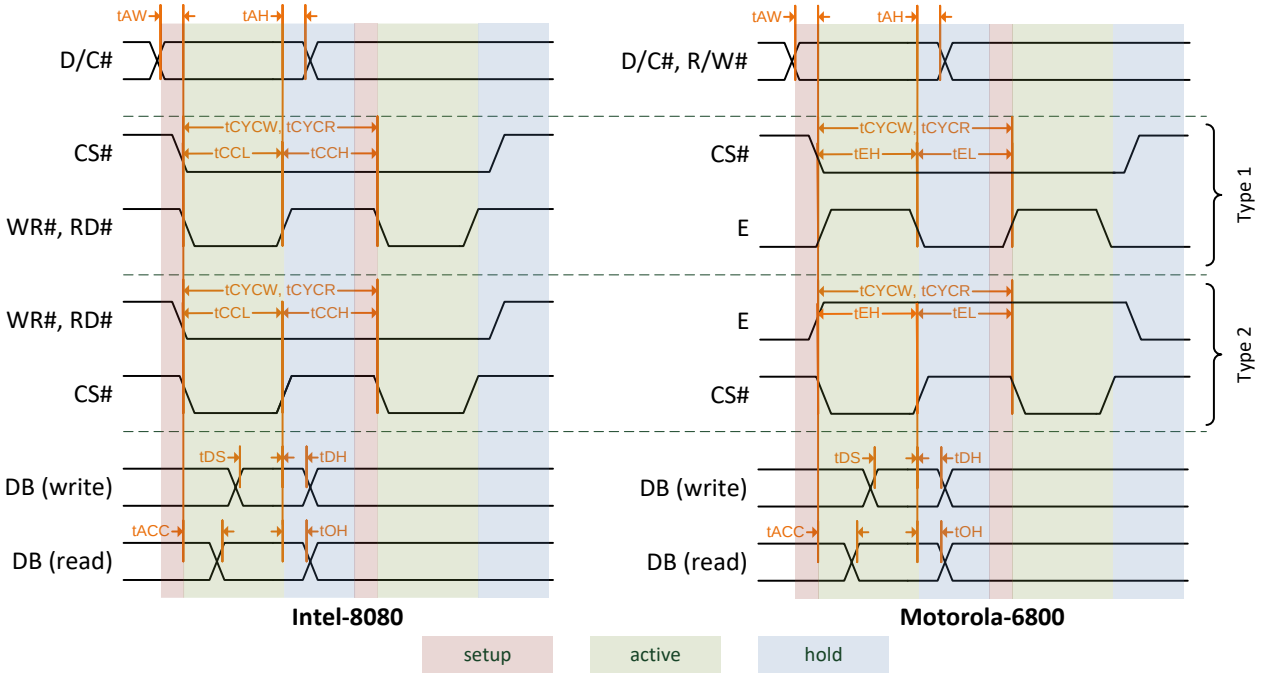


Table 12: Timing Symbol Descriptions

Symbol	*Description*
tAH	Address hold time
tAW	Address setup time
tCYCR/W	Read/Write cycle time
tCCHR/W, tEH	Read/Write, Enable pulse H width
tCCLR/W, tEL	Read/Write, Enable pulse L width
tDS	Write data set time
tDH	Write data hold time
tACC	Read data access time
tOH	Read data disable time

3.3.1.3 Interface Mapping

Depending on the InterfaceWidth set above and the native LCDBus register and pixel data width a transfer mapping should be configured for both command and data accesses as described in Data Mapping using the DataTransferMapping0..2 and CommandTransferMapping0..2 register respectively.

3.3.2 Packing Configuration

To program the color format conversion from Iris 24-bit to the color format required by the LCD controller the following configuration should be done:

1. Program BitsPerPixel as defined by the color format.
2. In most cases BitsPerTransfer will be equal to BitsPerPixel. Only if the InterfaceWidth is greater than BitsPerPixel and the LCD controller can handle multiple pixel per LCDBus transfer, BitsPerTransfer should be set to the InterfaceWidth to enable the multiple pixel per transfer feature of the LCDBusIF.
3. Program the RGB color bits selection using the ColorComponentBits and ColorComponentShift registers.

Table 13 lists a selection of packing configurations for color formats often used by LCD controllers.

Table 13: Selection of Standard LCD Controller Color Formats

Color Format	ColorComponentBits			ColorComponentShift			DestinationAttributes	
	Red	Green	Blue	Red	Green	Blue	BPT	BPP
RGB666	6	6	6	12	6	0	18	18
RGB565	5	6	5	11	5	0	16	16
RGB444	4	4	4	8	4	0	12	12
GREY8	0	8	0	0	0	0	8	8
GREY4	0	4	0	0	0	0	8	4
B/W	0	1	0	0	0	0	8	1

3.4 Color Table Setup

For data pixel write transfers, i.e. initiated through WR_PIXEL, a color lookup table can be enabled to support indexed color formats. This is especially efficient if the WR_PIXEL instruction is used with a bpp setting of 4-bit or 8-bit.

The color lookup table can act in two modes, either with one table of 256 entries and an 8-bit address or with 16 tables of 16 entries each with a 4-bit address per table. All color lookup table setup is done via the CLUT_CFG instruction.

The following steps are required to configure color lookup table behavior:

1. Make sure the color lookup table is disabled or at least not active due to an on-going transfer
2. Setup the ColorLookupTable with the desired data
3. Start the command sequencer with command list containing a CLUT_CFG command to control enable, index mode or a base address of the CLUT and the required WR_PIXEL command to initate the pixel transfer

3.5 Control Flow

To operate the LCDBusIF sequencer the InstructionFifo must be populated with instructions. This can either be done from a system master (e.g. CPU or DMA) or by setting up a read channel transfer.

In case a command list is written by the CPU, it should use the InstrFifoStatus register to check how much space is left to avoid that the fifo runs full.

The sequencer will start execution as soon as instructions are seen in the InstructionFifo. Each command list should be finished with an END instruction to allow the sequencer to return into idle state. When in idle state the instruction counter will reset only once the next instruction is read for better debugging. Once or if more instructions are available in the InstructionFifo the sequencer will restart execution.

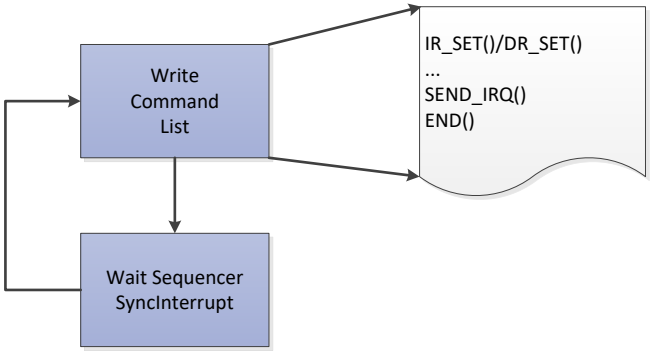
3.5.1 Generic Sequencer Control Flow

For command lists which don't depend on pixel content a simple control flow can be implemented.

1. The CPU writes the commands to be sent to the display to the InstructionFifo. In case it should be notified about the completion a SEND_IRQ command must be embedded in the command list.
2. In case an interrupt was setup, the CPU can wait for it to finish preparing the next command list. Alternatively the CPU can poll the OperationState field to become IDLE.

This flow is recommended for one shot operations like initialization, power-saving modes or shutdown.

Figure 15: Generic Sequencer Control Flow



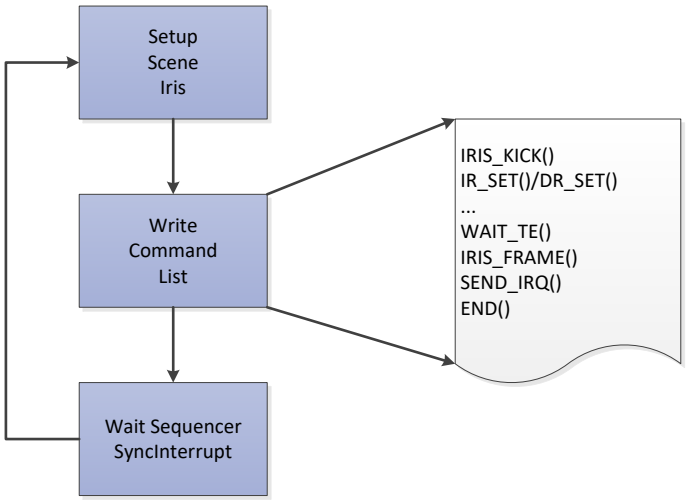
3.5.2 Iris Content Control Flow

In case the display content should be updated by the Iris a slightly modified control flow needs to be implemented.

1. The CPU configures the Iris graphics core to compose the desired scene. The LCD Bus Interface bypass needs to be active.
2. A command list is written which consists of the following sections:
   a. IRIS_KICK to trigger frame creation in the Iris
   b. Optional: a sequence of IR_SET/DR_SET commands to prepare the display GRAM access
   c. WAIT_TE to synchronize to tearing effect signal of the timing controller inside the display
   d. IRIS_FRAME to start sending pixels from the Iris to the display
   e. SEND_IRQ to inform the CPU about command list completion (sequence complete interrupt)

3. Wait for SequencerSyncInterrupt

Figure 16: Control Flow for Updating Display Content via Iris Interface



3.5.3 Data Master Control Flows

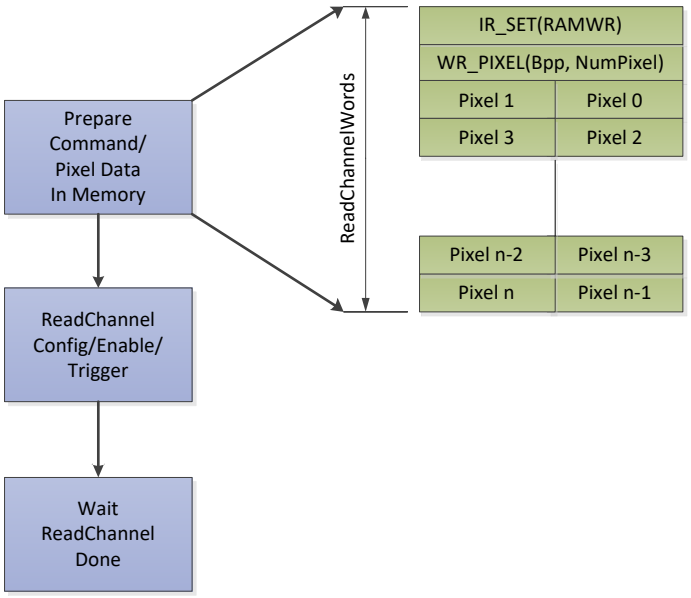
For systems that don't have an Iris graphics core the preferred way to fed the LCDBusIF with pixel data is to use the read channel master. Additionally the Write channel master can be used to read out the LCD controller framebuffer and store it into system memory.

For the read channel master the software can either prepare the complete command and pixel data in memory and trigger the read channel to fetch it or it can still provide normal control instructions via the InstructionFifo and use the read channel only for transferring the pixel data. Both flows are described hereafter.

3.5.3.1 Read Channel for Instruction and Pixel Data

1. Prepare command list in memory
2. Enable the read channel master using ReadChannelEnable (as a side-effect InstructionFifo will no longer be accessible to software)
3. Configure ReadChannelMaxBurstLength to define the default burst length
4. Set ReadChannelBaseAddress to point to the command list
5. Set ReadChannelWords according to the length of the command list
6. Initiate the read process by writing a '1' to ReadChannelTrigger
7. If required wait for ReadChannelDone interrupt (alternatively poll the ReadChannelStatus register)

Figure 17: Control Flow for Read Channel on Combineded Command and Pixel Data

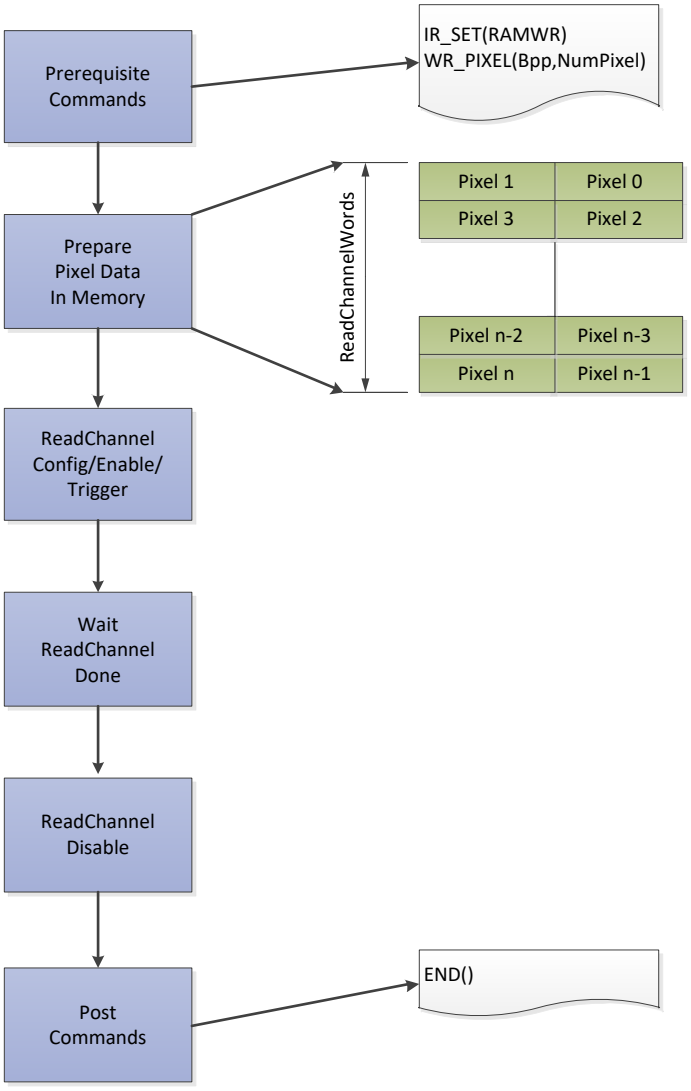


3.5.3.2 Read Channel for Pixel Data only

In case only pixel data should be read by the read channel the procedure must be slightly modified:

1. Make sure no previous read channel transfer is pending and the read channel is not enabled
2. Write any prerequisite commands directly to the InstructionFifo. The last instruction must be the WR_PIXEL instruction without the pixel data.
3. Enable the read channel master using ReadChannelEnable (as a side-effect InstructionFifo will no longer be accessible to software)
4. Configure ReadChannelMaxBurstLength to define the default burst length
5. Set ReadChannelBaseAddress to point to the frame buffer
6. Set ReadChannelWords according to the length of the frame buffer
7. Initiate the read process by writing a ‘1’ to ReadChannelTrigger
8. If required wait for ReadChannelDone interrupt (alternatively poll the ReadChannelStatus register)
9. Disable the read channel again
10. And write an END instruction to the InstructionFifo to let the sequencer go back into idle state

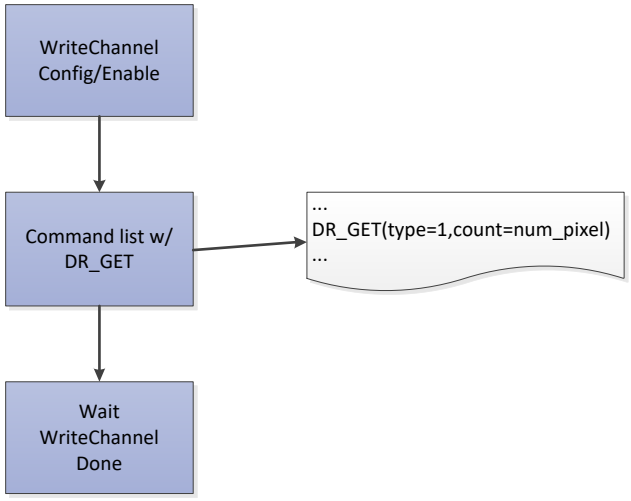
Figure 18: Control Flow for Read Channel on Pixel Data Only



3.5.3.3 Write Channel Control Flow

1. Enable the data write channel using WriteChannelEnable
2. Configure WriteChannelMaxBurstLength to define the default burst length
3. Set WriteChannelBaseAddress to point to the destination buffer for the read data
4. Configure WriteChannelPacking
5. Start the sequencer with a command list containing the required DR_GET instruction(s). Every DR_GET instruction where *Type* is set to Data will trigger a write process transferring as many pixels as specified by the *Count* parameter. Upon start of the write process the base address will be loaded from WriteChannelBaseAddress.
6. If required wait for WriteChannelDone interrupt (alternatively poll the WriteChannelStatus register)

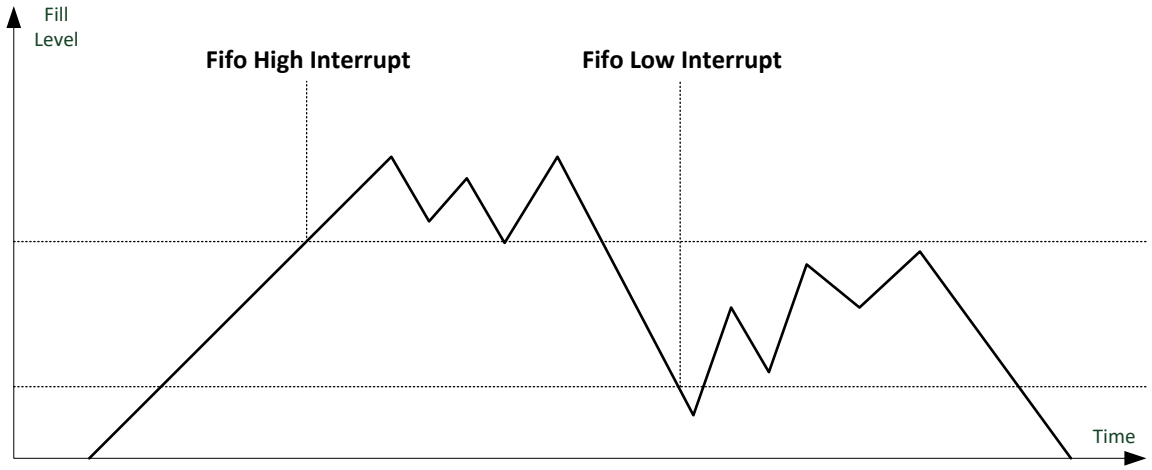
Figure 19: Control Flow for Reading Pixel Data from the Display



3.6 Fifo Operation

The LCDBusIF has two fifos which are accessible via the configuration interface. Both fifos implement a similar status and control interface for SW interaction. To trigger reception fifo read or instruction fifo write requests a programmable hysteresis on the fifo fill level is implemented. The software can configure to receive an interrupt either upon crossing a high or low level threshold. A high threshold interrupt will only be given if the low threshold boundary has been crossed before, similarly a low threshold interrupt will only be given if the high threshold boundary has been crossed before. This behavior is depicted in Figure 20. The following sections describe the behavior of the instruction and the reception fifo.

Figure 20: Fifo Hysteresis Implementation with Low and High Threshold Interrupt.



3.6.1 Instruction Fifo

Data can be sent to instruction fifo using either a fixed or incremented address within its address space (InstructionFifo). The number of available entries left can be read from the fifo status register (FifoStatus). Writing to a full InstructionFifo is considered an error condition and should be avoided from SW. If the software does write to a full InstructionFifo the bus will be stalled for some time to give the InstructionFifo some time to empty. If the fifo remains full an error response will be given on the configuration bus interface and the data will be discarded.

To clear instruction fifo the error recovery procedure should be executed, see Error Recovery.

3.6.2 Reception Fifo

Data can be read from reception fifo using either a fixed or incremented addresses from its address space (RxFifo). The number of available entries can be read from the fifos status registers (FifoStatus). Reading from an empty RxFifo is considered an error condition and should be avoided from SW. If the software does read from an empty RxFifo the LCDBusIF will respond with unknown data.

The reception fifo can be cleared individually (RxClear) if data in the fifo is obsolete. Otherwise the reception fifo will also be cleared with the error recovery procedure, see Error Recovery.

3.7 Error Recovery

In case the LCDBusIF is entering an erroneous state the following procedure should be applied to recover to normal operation:

1. Optional: read the SequencerStatus register to analyze the error condition.
2. Optional: read the RxFifo if it has any contents of value.
3. Trigger a SoftwareReset. This will clear both the InstructionFifo and the RxFifo and the sequencer will return into IDLE state. Any pending read or write channel process will be cancelled while making sure the AXI protocol is not violated.
4. Optional: wait until the Iris is idle if a frame transmission is still pending.
5. If applicable: wait until read channel is done by polling ReadChannelBusy
6. If applicable: wait until read channel is done by polling WriteChannelBusy
7. Clear interrupt status (interrupt status is not affected by software reset)

Afterwards the sequencer is ready to process a new command list.

3.8 Interrupt Setup

The current status of all interrupt lines (see Interrupt Map) can be read from the InterruptStatus register. Corresponding interrupt control:

- IRQ enable: InterruptEnable. Note, that this has no effect for interrupts connected as NMI.
- Clear status: InterruptClear.
- Set status: InterruptPreset. This has the same effect like the corresponding HW event and can be used for test purposes.

3.9 SW Interface

3.9.1 General

3.9.1.1 Register Addresses

The address of a register in a system environment is the sum of

- Register offset as specified in the following register tables.
- Address offset of the Iris core configuration in the address space of the embodying system.

3.9.1.2 Field Types

The field tables specify a *Type* for each field, which means the following:

Table 14: Field Types

R	Read only. Writing has no effect.
W	Write only. Reading always returns ‘0‘.
RW	Read and write to active configuration.
RWS	Read and write to shadowed configuration (must be activated by shadow load).
RW1C	Status bit. Set by HW. Writing ‘0‘ has no effect, writing ‘1‘ clears status. Reading returns current status.
W1P	Trigger bit. Writing ‘0‘ has no effect, writing ‘1‘ triggers some HW activity. Reading returns ‘0‘.

3.9.1.3 Field Formats

The field tables specify a *Format* for each field, which means the following:

Table 15: Field Formats

bit	Single bit.
Uint<*w*>	Unsigned integer with *w* bits.
Int<*w*>	Signed integer in two’s complement notation with *w* bits in total (including sign bit).
Ufix<*i*>.<*f*>	Unsigned fix-point with *I* integer and *f* fraction bits.
Fix<*i*>.<*f*>	Signed fix-point in two’s complement notation with *I* integer bits (including sign bit) and *f* fraction bits.
Float1e<*e*>m<*m*>	Floating point value with 1 sign bit, *e* exponent bits and *m* mantissa bits. Notation is according to IEEE 754.
Enum	Values have symbolic names (enumeration type).

3.9.1.4 Register Protection

If registers can be access protected can be read from *Prot* column in register tables:

Table 16: Register Lock Properties

key	Special register to change protection status of all registers in the same address block (= register table) by writing certain key values to it (see Key Map).
status	Special register to read current protection status.
no	Register that cannot be protected.
yes	Register that can be protected against non-privileged access (privilege key) and/or against any kind of write access (lock key).

3.9.1.5 Error Responses

The following conditions result in an error response on the AHB bus:

- Read or write access to an address with no register / no fields.
- Read or write access with transfer size other than 32-bit.
- Write access to a register that has read-only fields only.
- Read access to a register that has write-only fields only.
- Write access to a register with lock property ‘yes’ when lock status is active.

- Non-privileged read or write access to a register with lock property ‘yes’ or the lock/unlock register when privilege status is active (except read access to the lock status register, which is always allowed).
- Read access to lock/unlock register.
- Write access to the lock/unlock register with...
  - ☐ ...an invalid key value.
  - ☐ ...a valid key value when the freeze status is active.
  - ☐ ...the lock key value when internal unlock counter is 0.
  - ☐ ...the unlock key value when internal unlock counter is 15.
  - ☐ ...the privilege key value when the privilege status is active.
  - ☐ ...the un-privilege key value when the privilege status is not active.

3.9.2 Register Overview

Addr. Offset	Register Name	Prot.	Register Description (Address Block 0)
0x0000	LockUnlockLcd	key	Register to change the protection status of this address block.
0x0004	LockStatusLcd	status	Protection status of this address block.
0x0008	IPIdentifier	yes	IP Identifier for this LCDBusIF derivate, needs to be unlocked.
0x000C	InterfaceConfig	yes	LCDBus interface configuration register.
0x0010	WriteTimingConfig	yes	Write timing configuration register. Sum of all fields is write cycle time in module clock cycles.
0x0014	ReadTimingConfig	yes	Read timing configuration register. Sum of all fields is read cycle time in module clock cycles.
0x0018	CommandTransferMapping0	yes	Transfer mapping for command data on LCDBus.
0x001C	CommandTransferMapping1	yes	Transfer mapping for command data on LCDBus.
0x0020	CommandTransferMapping2	yes	Transfer mapping for command data on LCDBus.
0x0024	DataTransferMapping0	yes	Transfer mapping for data/parameter data on LCDBus.
0x0028	DataTransferMapping1	yes	Transfer mapping for data/parameter data on LCDBus.
0x002C	DataTransferMapping2	yes	Transfer mapping for data/parameter data on LCDBus.
0x0030	ColorComponentBits	yes	Color component size for color format of LCD controller.
0x0034	ColorComponentShift	yes	Color component offset for color format of LCD controller.
0x0038	DestinationAttributes	yes	Configuration for packing unit.
Addr. Offset	**Register Name**	**Prot.**	**Register Description (Address Block 1)**
0x0080	LockUnlockControl	key	Register to change the protection status of this address block.
0x0084	LockStatusControl	status	Protection status of this address block.
0x0088	InterruptEnable	yes	LCD Bus Interface interrupt enable register
0x008C	InterruptPreset	yes	LCD Bus Interface interrupt preset register
0x0090	InterruptClear	yes	LCD Bus Interface interrupt clear register
0x0094	InterruptStatus	yes	LCD Bus Interface interrupt status register
0x0098	SequencerConfig	yes	Sequencer configuration register.
0x009C	InstructionFifoConfig	yes	Instruction fifo configuration register.

Addr. Offset	Register Name	Prot.	Register Description (Address Block 1)
0x00A0	ReadChannelConfig	yes	Read Channel Configuration Register.
0x00A4	ReadChannelBuffer	yes	Read Channel Buffer Configuration Register.
0x00A8	ReadChannelControl	yes	Read Channel Control Register.
0x00AC	WriteChannelConfig	yes	Write Channel Configuration Register.
0x00B0	WriteChannelBuffer	yes	Write Channel Buffer Configuration Register.
0x00B4	SoftwareReset	yes	Software Reset Register.
0x00B8	SequencerSync	yes	Sequencer synchronization register.
0x00BC	SequencerStatus	yes	Status of the internal sequencer.
0x00C0	SequencerTransferStatus	yes	Transfer remaining of current executed command.
0x00C4	InstrFifoStatus	yes	Instruction fifo status register.
0x00C8	ReadChannelStatus	yes	Read Channel Status Register.
0x00CC	WriteChannelStatus	yes	Write Channel Status Register.
Addr. Offset	**Register Name**	**Prot.**	**Register Description (Address Block 2)**
0x0100	LockUnlockFifo	key	Register to change the protection status of the fifo address blocks.
0x0104	LockStatusFifo	status	Protection status of this address block.
0x0108	ReceptionFifoConfig	yes	Reception fifo configuration register.
0x010C	RxFifoControl	yes	Reception fifo control register.
0x0110	RxFifoStatus	yes	Reception fifo status register.
Addr. Offset	**Register Name**	**Prot.**	**Register Description (Address Block 3)**
		key	Same as for address block 2 of this unit.
		status	Same as for address block 2 of this unit.
0x0200	InstructionFifo[0..31]	yes	Instruction Fifo.
Addr. Offset	**Register Name**	**Prot.**	**Register Description (Address Block 4)**
		key	Same as for address block 2 of this unit.
		status	Same as for address block 2 of this unit.
0x0300	RxFifo[0..15]	yes	Reception Fifo.
Addr. Offset	**Register Name**	**Prot.**	**Register Description (Address Block 5)**
		key	Same as for address block 2 of this unit.
		status	Same as for address block 2 of this unit.
0x0400	ColorLookupTable[0..255]	yes	Color lookup table memory.

3.9.3 LockUnlockLcd

Register to change the protection status of this address block.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	W	-	enum	LockUnlockLcd	The protection status is changed by writing one of the following key values to this field: ■ 0x5651F763 = lock_key: Decrements the unlock counter. When the counter value is null, lock protection is active. Reset counter value is 1. ■ 0x691DB936 = unlock_key: Increments the unlock counter. Max allowed value is 15. ■ 0xAEE95CDC = privilege_key: Enables privilege protection. Disabled after reset. ■ 0xB5E2466E = unprivilege_key: Disables privilege protection. ■ 0xFBE8B1E6 = freeze_key: Freezes current protection status. Writing keys to this register has no more effect until reset.
When lock protection is active, no write but only read access is possible to all registers of this address block. When privilege protection is active, only privileged read and write access is possible. Both protections can be active at the same time. Reading this register returns the current unlock counter value, but results in an error response (for HW test purposes only).					

3.9.4 LockStatusLcd

Protection status of this address block.

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	bit	LockStatusLcd	Current status of lock protection: 0 = inactive (unlock counter > 0), 1 = active (unlock counter == 0).
[4]	R	0x0	bit	PrivilegeStatusLcd	Current status of privilege protection: 0 = inactive , 1 = active.
[8]	R	0x0	bit	FreezeStatusLcd	Current freeze status: 0 = protection status can be changed, 1 = cannot be changed.

3.9.5 IPIdentifier

IP Identifier for this LCDBusIF derivate, needs to be unlocked.

Bits	Type	Reset	Format	Field Name	Description
[3,0]	RW	0x0	uint4	Reserved	Reserved bits
[7,4]	RW	0x0	uint4	DesignDeliveryID	Design delivery ID (increased with each official delivery when maturity keeps the same)

Bits	Type	Reset	Format	Field Name	Description
[11,8]	RW	0x4	enum	DesignMaturityLevel	Design maturity level (corresponds to status at time of IP delivery, Fujitsu internal development stages) ■ 1 = PreFS: Pre feasibility study ■ 2 = FS: Feasibility study ■ 3 = R0: Functionality complete ■ 4 = R1: Verification complete
[15,12]	RW	0x0	uint4	IPEvolution	IP evolution (increased for functional spec changes only when feature set keeps the same)
[19,16]	RW	0x1	enum	IPFeatureSet	IP feature set (complexity of implemented features, e.g. availability of re-sampling filter etc) ■ 0 = LIGHT: LCDBusIF without data master functionality. ■ 1 = STANDARD: LCDBusIF with data master functionality.

3.9.6 InterfaceConfig

LCDBus interface configuration register.

Bits	Type	Reset	Format	Field Name	Description
[1,0]	RW	0x0	enum	InterfaceType	Describes the type of the external interface. ■ 0x0 = INTEL80_TYPE1: Intel 8080 compatible interface (strobe with RD#/WR#). ■ 0x1 = INTEL80_TYPE2: Intel 8080 compatible interface (strobe with CS#). ■ 0x2 = MOTOROLA68_TYPE1: Motorola 6800 compatible interface (strobe with E). ■ 0x3 = MOTOROLA68_TYPE2: Motorola 6800 compatible interface (strobe with CS#).
[9,8]	RW	0x0	enum	InterfaceWidth	Describes the data width of the external interface. ■ 0x0 = LCD8: 8-bit LCDBus ■ 0x1 = LCD9: 9-bit LCDBus ■ 0x2 = LCD16: 16-bit LCDBus ■ 0x3 = LCD18: 18-bit LCDBus
[16]	RW	0x0	enum	CsPolarity	Polarity for chip select output. ■ 0x0 = ACTIVE_LOW: CS is a low active signal. ■ 0x1 = ACTIVE_HIGH: CS is a high active signal.

Bits	Type	Reset	Format	Field Name	Description
[17]	RW	0x0	enum	WrPolarity	Polarity for WR# (8080) or R/W# (6800) otuput. ■ 0x0 = DEFAULT: WR# (R/W#) has default protocol polarity. ■ 0x1 = INVERTED: WR# (R/W#) has inverted protocol polarity.
[18]	RW	0x0	enum	RdPolarity	Polarity for RD# (8080) or E (6800) output. ■ 0x0 = DEFAULT: RD# (E) has default protocol polarity. ■ 0x1 = INVERTED: RD# (E) has inverted protocol polarity.
[19]	RW	0x0	enum	RsPolarity	Polarity of register select output. ■ 0x0 = COMMAND_LOW: RS is command with low signal level. ■ 0x1 = COMMAND_HIGH: RS is command with high signal level.
[20]	RW	0x0	enum	DataPolarity	Inversion for data signals upon write. ■ 0x0 = DEFAULT: Data bus is used as is. ■ 0x1 = INVERTED: Data bus is inverted.
[21]	RW	0x0	enum	TePolarity	Polarity for tearing effect input. ■ 0x0 = ACTIVE_HIGH: TE is a high active signal. ■ 0x1 = ACTIVE_LOW: TE is a low active signal.

3.9.7 WriteTimingConfig

Write timing configuration register. Sum of all fields is write cycle time in module clock cycles.

Bits	Type	Reset	Format	Field Name	Description
[7,0]	RW	0x4	uint8	WriteSetupCycles	Number of module clock cycles minus one to wait before wr#/cs#/en terminal can become active after address set (tAW).
[15,8]	RW	0x8	uint8	WriteActiveCycles	Number of module clock cycles minus one wr#/cs#/en terminal should remain active (tCCLW).
[23,16]	RW	0x4	uint8	WriteHoldCycles	Number of module clock cycles minus one to wait after wr#/cs#/en terminal became inactive before address can change (tAH).

3.9.8 ReadTimingConfig

Read timing configuration register. Sum of all fields is read cycle time in module clock cycles.

Bits	Type	Reset	Format	Field Name	Description
[7,0]	RW	0x4	uint8	ReadSetupCycles	Number of module clock cycles minus one to wait before rd#/cs#/en terminal can become active after address set (tAW).
[15,8]	RW	0x8	uint8	ReadActiveCycles	Number of module clock cycles minus one rd#/cs#/en terminal should remain active (tCCLR).
[23,16]	RW	0x4	uint8	ReadHoldCycles	Number of module clock cycles minus one to wait after rd#/cs#/en terminal became inactive before address can change (tAH).

3.9.9 CommandTransferMapping0

Transfer mapping for command data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x8	uint5	CommandTransfer0Bits	Number of bits from InstructionFifo to be transfered with first transfer. Must be greater than 0.
[9,5]	RW	0x0	uint5	CommandTransfer0SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (first transfer).
[14,10]	RW	0x0	uint5	CommandTransfer0DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (first transfer).
[15]	RW	0x0	bit	CommandTransfer0Next	If set, a second transfer will be required to transmit the command data, see CommandTransferMapping1.
Command mapping is applicable for IR_SET, IR_GET, DR_SET(type==1), DR_GET(type==1) instructions.					

3.9.10 CommandTransferMapping1

Transfer mapping for command data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x0	uint5	CommandTransfer1Bits	Number of bits from InstructionFifo to be transfered with second transfer. Must be greater than 0 if CommandTransfer0Next is set.
[9,5]	RW	0x0	uint5	CommandTransfer1SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (second transfer).
[14,10]	RW	0x0	uint5	CommandTransfer1DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (second transfer).
[15]	RW	0x0	bit	CommandTransfer1Next	If set, a third transfer will be required to transmit the command data, see CommandTransferMapping2.
Command mapping is applicable for IR_SET, IR_GET, DR_SET(type==1), DR_GET(type==1) instructions.					

3.9.11 CommandTransferMapping2

Transfer mapping for command data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x0	uint5	CommandTransfer2Bits	Number of bits from InstructionFifo to be transfered with third transfer.
[9,5]	RW	0x0	uint5	CommandTransfer2SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (third transfer).
[14,10]	RW	0x0	uint5	CommandTransfer2DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (third transfer).
Command mapping is applicable for IR_SET, IR_GET, DR_SET(type==1), DR_GET(type==1) instructions.					

3.9.12 DataTransferMapping0

Transfer mapping for data/parameter data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x8	uint5	DataTransfer0Bits	Number of bits from DataFifo to be transfered with first transfer.
[9,5]	RW	0x0	uint5	DataTransfer0SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (first transfer).
[14,10]	RW	0x0	uint5	DataTransfer0DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (first transfer).
[15]	RW	0x0	bit	DataTransfer0Next	If set, second transfer will be required to transmit a data word.
Data mapping is applicable for DR_SET(type==0), DR_GET(type==0), IRIS_FRAME and WR_PIXEL instructions.					

3.9.13 DataTransferMapping1

Transfer mapping for data/parameter data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x0	uint5	DataTransfer1Bits	Number of bits from DataFifo to be transfered with second transfer.
[9,5]	RW	0x0	uint5	DataTransfer1SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (second transfer).
[14,10]	RW	0x0	uint5	DataTransfer1DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (second transfer).
[15]	RW	0x0	bit	DataTransfer1Next	If set, third transfer will be required to transmit a data word.
Data mapping is applicable for DR_SET(type==0), DR_GET(type==0), IRIS_FRAME and WR_PIXEL instructions.					

3.9.14 DataTransferMapping2

Transfer mapping for data/parameter data on LCDBus.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x0	uint5	DataTransfer2Bits	Number of bits from DataFifo to be transfered with third transfer.
[9,5]	RW	0x0	uint5	DataTransfer2SrcLsb	Least significant bit of vector to extract from source (Wr=InstructionFifo, Rd=Interface data) (third transfer).
[14,10]	RW	0x0	uint5	DataTransfer2DstLsb	Least significant bit in destination word (Wr=Interface data, Rd=RxFifo) to insert vector into (third transfer).
Data mapping is applicable for DR_SET(type==0), DR_GET(type==0), IRIS_FRAME and WR_PIXEL instructions.					

3.9.15 ColorComponentBits

Color component size for color format of LCD controller.

Bits	Type	Reset	Format	Field Name	Description
[11,8]	RW	0x5	uint4	ComponentBitsBlue	Blue component bits.
[19,16]	RW	0x6	uint4	ComponentBitsGreen	Green component bits.
[27,24]	RW	0x5	uint4	ComponentBitsRed	Red component bits.
Size of RGB is from 0 to 8 bits. The sum of RGB bits must be greater than 0.					

3.9.16 ColorComponentShift

Color component offset for color format of LCD controller.

Bits	Type	Reset	Format	Field Name	Description
[12,8]	RW	0x0	uint5	ComponentShiftBlue	Blue component shift.
[20,16]	RW	0x5	uint5	ComponentShiftGreen	Green component shift.
[28,24]	RW	0xB	uint5	ComponentShiftRed	Red component shift.

3.9.17 DestinationAttributes

Configuration for packing unit.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x10	uint5	BitsPerTransfer	How many bits can be packed into one transfer (allows packing multiple bits into one LCDBus transfer).
[12,8]	RW	0x10	uint5	BitsPerPixel	How many bits are required for one pixel.
The packing factor per transfer is the integer division of BitsPerTransfer/BitsPerPixel. BitsPerTransfer must be greater than or equal to BitsPerPixel.					

3.9.18 LockUnlockControl

Register to change the protection status of this address block.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	W	-	enum	LockUnlockControl	The protection status is changed by writing one of the following key values to this field: • 0x5651F763 = lock_key: Decrements the unlock counter. When the counter value is null, lock protection is active. Reset counter value is 1. • 0x691DB936 = unlock_key: Increments the unlock counter. Max allowed value is 15. • 0xAEE95CDC = privilege_key: Enables privilege protection. Disabled after reset. • 0xB5E2466E = unprivilege_key: Disables privilege protection. • 0xFBE8B1E6 = freeze_key: Freezes current protection status. Writing keys to this register has no more effect until reset.
When lock protection is active, no write but only read access is possible to all registers of this address block. When privilege protection is active, only privileged read and write access is possible. Both protections can be active at the same time. Reading this register returns the current unlock counter value, but results in an error response (for HW test purposes only).					

3.9.19 LockStatusControl

Protection status of this address block.

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	bit	LockStatusControl	Current status of lock protection: 0 = inactive (unlock counter > 0), 1 = active (unlock counter == 0).
[4]	R	0x0	bit	PrivilegeStatusControl	Current status of privilege protection: 0 = inactive , 1 = active.
[8]	R	0x0	bit	FreezeStatusControl	Current freeze status: 0 = protection status can be changed, 1 = cannot be changed.

3.9.20 InterruptEnable

LCD Bus Interface interrupt enable register

Bits	Type	Reset	Format	Field Name	Description
[0]	RW	0x0	bit	SequencerErrorInterruptEnable	If set to '1' the SequencerError interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[1]	RW	0x0	bit	SequencerSyncInterruptEnable	If set to '1' the SequencerSync interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[2]	RW	0x0	bit	InstrFifoInterruptEnable	If set to '1' the TxFifo interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[3]	RW	0x0	bit	RxFifoInterruptEnable	If set to '1' the RxFifo interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[4]	RW	0x0	bit	TearingEffectInterruptEnable	If set to '1' the TearingEffect interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[5]	RW	0x0	bit	ReadChannelDoneInterruptEnable	If set to '1' the ReadChannelDone interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.
[6]	RW	0x0	bit	WriteChannelDoneInterruptEnable	If set to '1' the WriteChannelDone interupt is enabled, if set to '0' the interrupt is disabled and only the InterruptStatus register can be used to see the status.

3.9.21 InterruptPreset

LCD Bus Interface interrupt preset register

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	-	bit	SequencerErrorInterruptPreset	Write '1' to this field to set the SequencerError interrupt.
[1]	W1P	-	bit	SequencerSyncInterruptPreset	Write '1' to this field to set the SequencerSync interrupt.
[2]	W1P	-	bit	InstrFifoInterruptPreset	Write '1' to this field to set the TxFifo interrupt.
[3]	W1P	-	bit	RxFifoInterruptPreset	Write '1' to this field to set the RxFifo interrupt.
[4]	W1P	-	bit	TearingEffectInterruptPreset	Write '1' to this field to set the TearingEffect interrupt.
[5]	W1P	-	bit	ReadChannelDoneInterruptPreset	Write '1' to this field to set the ReadChannelDone interrupt.
[6]	W1P	-	bit	WriteChannelDoneInterruptPreset	Write '1' to this field to set the WriteChannelDone interrupt.

3.9.22 InterruptClear

LCD Bus Interface interrupt clear register

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	-	bit	SequencerErrorInterruptClear	Write '1' to this field to clear the SequencerError interrupt.
[1]	W1P	-	bit	SequencerSyncInterruptClear	Write '1' to this field to clear the SequencerSync interrupt.
[2]	W1P	-	bit	InstrFifoInterruptClear	Write '1' to this field to clear the TxFifo interrupt.
[3]	W1P	-	bit	RxFifoInterruptClear	Write '1' to this field to clear the RxFifo interrupt.
[4]	W1P	-	bit	TearingEffectInterruptClear	Write '1' to this field to clear the TearingEffect interrupt.
[5]	W1P	-	bit	ReadChannelDoneInterruptClear	Write '1' to this field to clear the ReadChannelDone interrupt.
[6]	W1P	-	bit	WriteChannelDoneInterruptClear	Write '1' to this field to clear the WriteChannelDone interrupt.

3.9.23 InterruptStatus

LCD Bus Interface interrupt status register

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	bit	SequencerErrorInterruptStatus	Status of the SequencerError interrupt.
[1]	R	0x0	bit	SequencerSyncInterruptStatus	Status of the SequencerSync interrupt.
[2]	R	0x0	bit	InstrFifoInterruptStatus	Status of the TxFifo interrupt.
[3]	R	0x0	bit	RxFifoInterruptStatus	Status of the RxFifo interrupt.
[4]	R	0x0	bit	TearingEffectInterruptStatus	Status of the TearingEffect interrupt.
[5]	R	0x0	bit	ReadChannelDoneInterruptStatus	Status of the ReadChannelDone interrupt.
[6]	R	0x0	bit	WriteChannelDoneInterruptStatus	Status of the WriteChannelDone interrupt.

3.9.24 SequencerConfig

Sequencer configuration register.

Bits	Type	Reset	Format	Field Name	Description
[30,0]	RW	0x7FFFFFF	uint31	InstructionTimeout	Value to preload the timeout counter with each instruction read.
[31]	RW	0x1	bit	InstructionTimeoutEnable	Enable or disable instruction timeout counter.
If the instruction timeout mechanism is enabled, a timeout counter will be preloaded with the InstructionTimeout value every time a new instruction is read. With each cycle an instruction remains active the timeout counter is decremented. In case the counter reaches 0 the sequencer will enter an error state and generate a SequencerErrorInterrupt.					

3.9.25 InstructionFifoConfig

Instruction fifo configuration register.

Bits	Type	Reset	Format	Field Name	Description
[5,0]	RW	0x0	uint6	InstrThresholdLow	The threshold below which the instruction fifo should generate a low interrupt.
[13,8]	RW	0x20	uint6	InstrThresholdHigh	The threshold above which the instruction fifo should generate a high interrupt.
[17,16]	RW	0x1	enum	InstrThresholdTrigger	Select trigger condition for InstrFifoInterrupt. ■ 0 = LOW: InstrFifoInterrupt triggers if level falls below low threshold after being above high level. ■ 1 = HIGH: InstrFifoInterrupt triggers if level raises above high threshold after being below low level. ■ 2 = BOTH: InstrFifoInterrupt triggers for both low and high threshold conditions.
Note: InstrThresholdHigh must be greater than InstrThresholdLow.					

3.9.26 ReadChannelConfig

Read Channel Configuration Register.

Bits	Type	Reset	Format	Field Name	Description
[23,0]	RW	0x0	uint24	ReadChannelWords	Set to number of 32-bit words to read minus 1.
[28,24]	RW	0x8	uint5	ReadChannelMaxBurstLength	Default burstlength for read channel requests. Will always be used except for the last access if remaining words is less than ReadChannelMaxBurstLength.
[31]	RW	0x0	bit	ReadChannelEnable	Enable read channel interface. Locks the InstructionFifo from configuration interface.

3.9.27 ReadChannelBuffer

Read Channel Buffer Configuration Register.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	RW	0x0	uint32	ReadChannelBaseAddress	Base address of read buffer. Must be 32-bit aligned.

3.9.28 ReadChannelControl

Read Channel Control Register.

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	0x0	bit	ReadChannelTrigger	Initiate read of data via read channel interface.

3.9.29 WriteChannelConfig

Write Channel Configuration Register.

Bits	Type	Reset	Format	Field Name	Description
[2,0]	RW	0x0	enum	WriteChannelPacking	Set write channel packing configuration. ■ 0x0 = DISABLED: Write LCDBus read data as mapped into reception fifo. ■ 0x1 = BPP16: Pack 16-bit pixel data into 32-bit memory transfers. ■ 0x2 = BPP8: Pack 8-bit pixel data into 32-bit memory transfers. ■ 0x3 = RSVD: Reserved do not use. ■ 0x4 = BPP18TIGHT: Pack 18-bit pixel data into 32-bit memory transfers.
[28,24]	RW	0x8	uint5	WriteChannelMaxBurstLength	Default burstlength for write channel requests. Will always be used except for the last access if remaining words is less than WriteChannelMaxBurstLength.
[31]	RW	0x0	bit	WriteChannelEnable	Enable write channel interface. Locks the RxFifo from configuration interface.

3.9.30 WriteChannelBuffer

Write Channel Buffer Configuration Register.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	RW	0x0	uint32	WriteChannelBaseAddress	Base address of write buffer. Must be 32-bit aligned.

3.9.31 SoftwareReset

Software Reset Register.

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	-	bit	SoftwareReset	Writting a 1 to this field will reset all fifos and control logic of LCDBusIF.

3.9.32 SequencerSync

Sequencer synchronization register.

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	-	bit	SequencerAck	A write to this bit will allow the sequencer to continue from the WAIT_ACK instruction.

3.9.33 SequencerStatus

Status of the internal sequencer.

Bits	Type	Reset	Format	Field Name	Description
[1,0]	R	0x0	enum	OperationState	Indicate operation status. ■ 0x0 = IDLE: Processing finished, nothing to do. ■ 0x1 = BUSY: Currently processing instruction list. ■ 0x2 = WAIT: Waiting for next instruction or data. ■ 0x3 = ERROR: An error has occured
[2]	R	0x0	enum	WaitAck	Indicate if a WAIT_ACK instruction is currently pending. ■ 0x0 = DONE: No waiting for acknowledge. ■ 0x1 = PENDING: Acknowledge from CPU is pending.
[7,4]	R	0x0	enum	ErrorCode	Type of error that has occured ■ 0x0 = NONE: No error. ■ 0x1 = OPCODE: Illegal opcode error. ■ 0x2 = KICK: Too many outstanding kicks error. ■ 0x3 = TIMEOUT: Sequencer timout reached.
[13,8]	R	-	uint6	InstructionCounter	Number of instructions processed in current command list.
[23,16]	R	0x0	enum	InstructionRegister	Current instruction executed by sequencer. ■ 0x00 = NOP: Currently executing NOP instruction. ■ 0x10 = IR_SET: Currently executing IR_SET instruction. ■ 0x11 = IR_GET: Currently executing IR_GET instruction. ■ 0x12 = DR_SET: Currently executing DR_SET instruction. ■ 0x13 = DR_GET: Currently executing DR_GET instruction. ■ 0x14 = DUMMY_GET: Currently executing DUMMY_GET instruction. ■ 0x15 = WAIT_TE: Currently executing WAIT_TE instruction. ■ 0x16 = WAIT_IF: Currently executing WAIT_IF instruction. ■ 0x1F = SET_RESET: Currently executing SET_RESET instruction. ■ 0x20 = SEND_IRQ: Currently executing SEND_IRQ instruction. ■ 0x21 = WAIT_ACK: Currently executing WAIT_ACK instruction. ■ 0x40 = IRIS_KICK: Currently executing IRIS_KICK instruction.

Bits	Type	Reset	Format	Field Name	Description
					■ 0x41 = WAIT_CMD: Currently executing WAIT_CMD instruction. ■ 0x42 = IRIS_FRAME: Currently executing IRIS_FRAME instruction. ■ 0x80 = WR_PIXEL: Currently executing WR_PIXEL instruction. ■ 0xC0 = CLUT_CFG: Currently executing CLUT_CFG instruction. ■ 0xFF = END: Currently executing END instruction.
[25,24]	R	0x0	enum	IrisInterfaceStatus	Status of Iris interface. ■ 0x0 = IDLE: Nothing pending. ■ 0x1 = KICK: Kick sent, frame pending. ■ 0x2 = CMD: Command received, pixel transmission pending. ■ 0x3 = FRAME: Frame transmission ongoing.
[28]	R	0x0	enum	LcdInterfaceState	Indicate LCDBus status. ■ 0x0 = IDLE: No transactions pending. ■ 0x1 = BUSY: Currently sending transactions.

3.9.34 SequencerTransferStatus

Transfer remaining of current executed command.

Bits	Type	Reset	Format	Field Name	Description
[27,0]	R	0x0	uint28	TransferRemaining	Internal counter of the sequencer for current executed instruction NOP, IR_GET, DR_GET, WR_PIXEL or IRIS_FRAME.

3.9.35 InstrFifoStatus

Instruction fifo status register.

Bits	Type	Reset	Format	Field Name	Description
[5,0]	R	0x20	uint6	InstrSpace	Available space in InstructionregistFifo in number of entries.
[8]	R	0x1	bit	InstrEmpty	Indicates if InstructionFifo is empty.
[12]	R	0x0	bit	InstrFull	Indicates if InstructionFifo is full.

3.9.36 ReadChannelStatus

Read Channel Status Register.

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	enum	ReadChannelBusy	Status indication for read channel interface. ■ 0x0 = IDLE: No operation. ■ 0x1 = BUSY: Reading data.
[31,8]	R	0x0	uint24	ReadChannelWordsRemaining	Number of 32-bit words still to be read.

3.9.37 WriteChannelStatus

Write Channel Status Register.

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	enum	WriteChannelBusy	Status indication for write channel interface. ■ 0x0 = IDLE: No operation. ■ 0x1 = BUSY: Writeing data.

3.9.38 LockUnlockFifo

Register to change the protection status of the fifo address blocks.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	W	-	enum	LockUnlockFifo	The protection status is changed by writing one of the following key values to this field: ■ 0x5651F763 = lock_key: Decrements the unlock counter. When the counter value is null, lock protection is active. Reset counter value is 1. ■ 0x691DB936 = unlock_key: Increments the unlock counter. Max allowed value is 15. ■ 0xAEE95CDC = privilege_key: Enables privilege protection. Disabled after reset. ■ 0xB5E2466E = unprivilege_key: Disables privilege protection. ■ 0xFBE8B1E6 = freeze_key: Freezes current protection status. Writing keys to this register has no more effect until reset.
When lock protection is active, no write but only read access is possible to all registers of this address block. When privilege protection is active, only privileged read and write access is possible. Both protections can be active at the same time. Reading this register returns the current unlock counter value, but results in an error response (for HW test purposes only).					

3.9.39 LockStatusFifo

Protection status of this address block.

Bits	Type	Reset	Format	Field Name	Description
[0]	R	0x0	bit	LockStatusFifo	Current status of lock protection: 0 = inactive (unlock counter > 0), 1 = active (unlock counter == 0).
[4]	R	0x0	bit	PrivilegeStatusFifo	Current status of privilege protection: 0 = inactive , 1 = active.
[8]	R	0x0	bit	FreezeStatusFifo	Current freeze status: 0 = protection status can be changed, 1 = cannot be changed.

3.9.40 ReceptionFifoConfig

Reception fifo configuration register.

Bits	Type	Reset	Format	Field Name	Description
[4,0]	RW	0x0	uint5	RxThresholdLow	The threshold below which the reception fifo should generate a low interrupt.
[12,8]	RW	0x10	uint5	RxThresholdHigh	The threshold above which the reception fifo should generate a high interrupt.
[17,16]	RW	0x1	enum	RxThresholdTrigger	Select trigger condition for RxFifoInterrupt. ■ 0 = LOW: RxFifoInterrupt triggers if level falls below low threshold after being above high level. ■ 1 = HIGH: RxFifoInterrupt triggers if level raises above high threshold after being below low level. ■ 2 = BOTH: RxFifoInterrupt triggers for both low and high threshold conditions.
Note: RxThresholdHigh must be greater than RxThresholdLow.					

3.9.41 RxFifoControl

Reception fifo control register.

Bits	Type	Reset	Format	Field Name	Description
[0]	W1P	-	bit	RxClear	Writting a 1 to this field will clear the contents of the RxFifo only.

3.9.42 RxFifoStatus

Reception fifo status register.

Bits	Type	Reset	Format	Field Name	Description
[20,16]	R	0x0	uint5	RxLevel	Number of entries in RxFifo.
[24]	R	0x1	bit	RxEmpty	Indicates if RxFifo is empty.
[28]	R	0x0	bit	RxFull	Indicates if RxFifo is full.

3.9.43 InstructionFifo[0..31]

Instruction Fifo.

Bits	Type	Reset	Format	Field Name	Description
[31,0]	W	-	uint32	Instruction	Command instructions to be executed by the sequencer.
The sequencer (if IDLE) will start operation as soon as instructions are added to the fifo. Instruction format is upper byte for opcode and lower three bytes for parameter(s) if any. If the InstructionFifo is written while being full the configuration interface will wait for defined period for a free entry before discarding the Instruction and giving an ERROR response.					

3.9.44 RxFifo[0..15]

Reception Fifo.

Bits	Type	Reset	Format	Field Name	Description
[17,0]	R	-	uint18	RxEntry	Data received from the LCDBus.
Read data from the LCDBus packed according CommandTransferMapping or DataTransferMapping depending on the triggering instruction. If the RxFifo is read while being empty, the data will be unknown. No ERROR response is generated.					

3.9.45 ColorLookupTable[0..255]

Color lookup table memory.

Bits	Type	Reset	Format	Field Name	Description
[17,0]	RW	-	uint18	ClutEntry	ClutEntry in LCD controller target color format (up to 18-bit).

CHAPTER 35: LCD Controller

This chapter explains the LCD controller.

1. Overview
2. Features
3. Configuration
4. Operation
5. Setting
6. Registers
7. Q&A
8. Sample Program
9. Notes

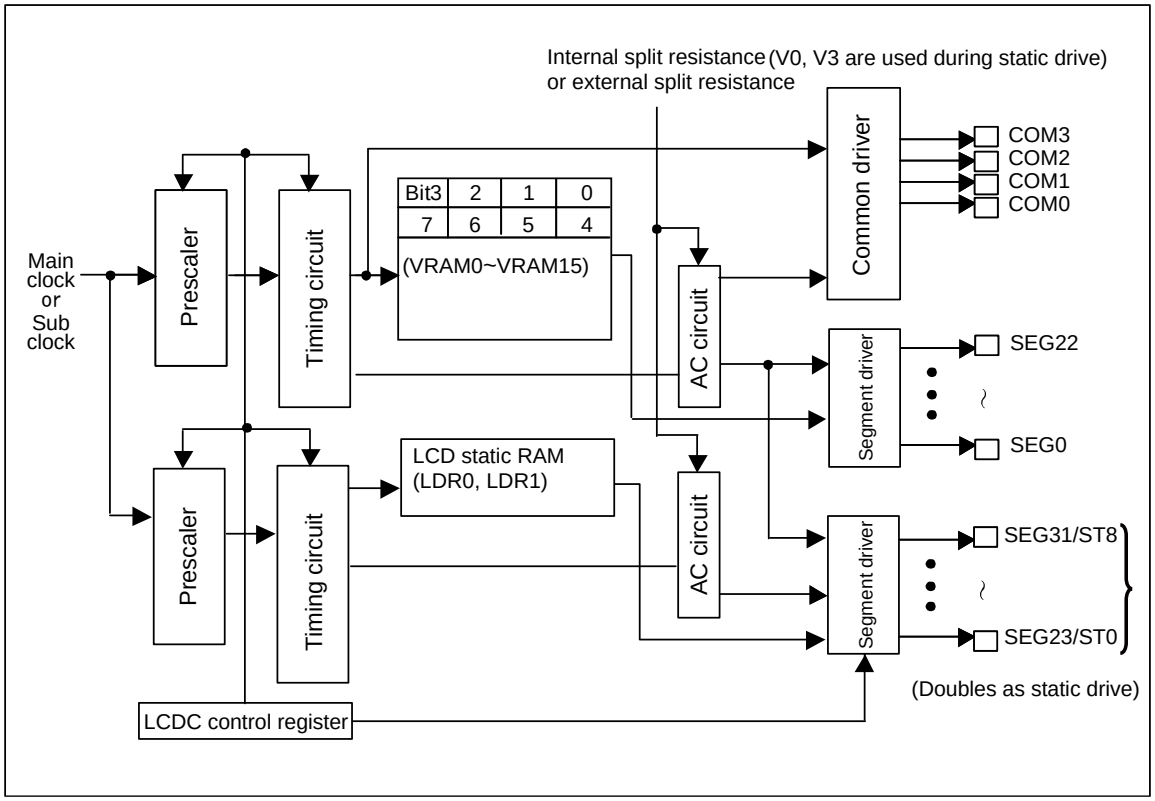
LCDC-S6J3200-E1

1. Overview

This section explains the overview of the LCD controller.

The LCD enables the selection of duty from 1/2, 1/3, and 1/4, and up to 128-element display.

Furthermore, it enables up to 8-element display as static LCD output.



2. Features

This section explains the features of the LCD controller.

<Duty drive>

- Number: 1 (4-common × 32-segment)
- Display: Up to 128-element (on 1/4 duty)
- Duty: Selection from three types (1/2, 1/3, 1/4)
- Bias: Selection from 1/2 and 1/3

Combination of Bias, Duty, and Common Output

Bias	1/2 Duty Output Mode	1/3 Duty Output Mode	1/4 Duty Output Mode
1/2 bias	○	×	×
1/3 bias	×	○	○

○ : Recommended mode

× : Prohibited

- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Driver: Built-in (Internal division resistor), or an external division resistor can be connected to V0 to V3 pins.
- Data memory: Built-in 16-byte data memory for display
- No display selection: Enabled
  - Pin: COM0 to COM3, SEG0 to SEG31, V0, V1, V2, V3 pins are used for general-purpose ports also and switching is enabled.
    SEG23 to SEG31 pins are used as static drive pins (ST0 to ST8) also and switching is enabled.
    VCC can be selected instead of V3.
- During duty driving, up to 4 common outputs (COM0 to COM3) and 32 segment outputs (SEG0 to SEG31) can be used.
  - Others: The external division resistor also can block a current when LCD is stopped.

< Static drive >

- Number: 1 (1-common × 8-segment)
- Display: Up to 8 element
- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Pin: ST0 to ST8 pins are used also as general-purpose ports and duty drive pins (SEG23 to SEG31) and switching is enabled.
- During static driving, up to 1 static common output and 8 static segment outputs (ST0 to ST8) can be used.

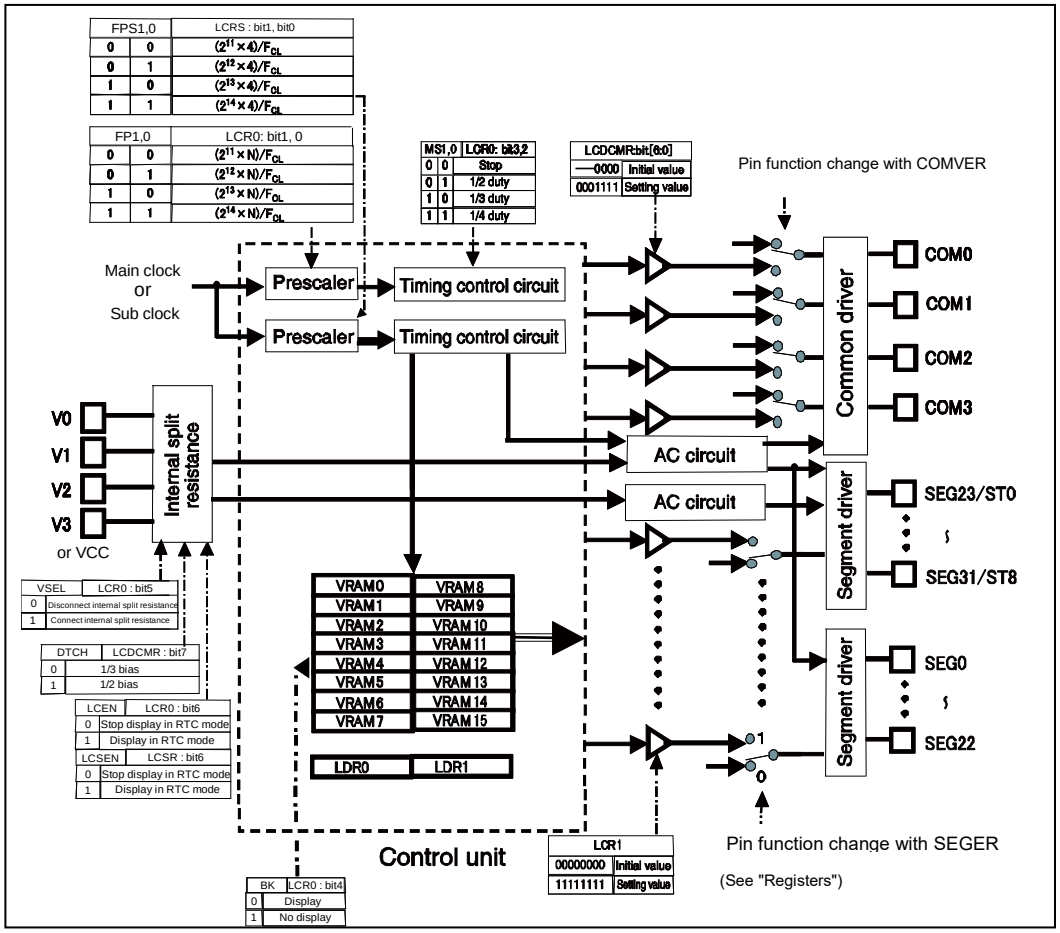
3. Configuration

This section shows the configuration of the LCD controller.

■ Configuration Diagram of LCD Controller

Figure 3-1 shows the configuration diagram of the LCD controller.

Figure 3-1 Configuration Diagram



4. Operation

The following sections explain the operation of the LCD controller.

4.1. Operation of LCD Controller/Driver (LCDC)

4.2. 1/2 Duty Output Waveform

4.3. 1/3 Duty Output Waveform

4.4. 1/4 Duty Output Waveform

4.5. Static Drive Output Waveform

4.1. Operation of LCD Controller/Driver (LCDC)

The operation of LCD controller/driver (LCDC) is shown below.

1. Data Memory for Display

<Duty drive>

Set a value to the data memory for display (VRAM) in advance.

<Static drive>

Set a value to the data memory for display (LDR0, LDR1) in advance.

2. Write Necessary Settings to Each Register.

3. Output Pin

<Duty drive>

When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (COM0 to COM3, SEG0 to SEG31).

The content of the VRAM is read automatically in synchronization with the timing of the common signal and output from the segment output pin.

(When the bit of VRAM is set to "1", a selection waveform is output from the segment output pin.

When the bit of VRAM is set to "0", a non-selection waveform is output from the segment output pin.)

Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.

<Static drive>

When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (ST0 to ST8).

4. Output Waveform

<Duty drive>

Output waveforms drive in 2 frames alternating waveform according to the duty setting.

<Static drive>

Output waveforms drive in 1 frame alternating waveform.

5. Operation in the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)

<Duty drive>

For operation enable (LCEN = "1"), the LCD is displayed.

Please do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

<Static drive>

For operation enable (LCSEN = "1"), the LCD is displayed.

Please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode.

LCD display stop when power shutdown

6. Blanking Function

<Duty drive>

Light of the LCD can be turned off with selection of non-display (BK = "1") of blanking.

However, non-selection waveforms are output.

7. LCD Stop State

<Duty drive>

When display operation of the LCD is stopped (MS [1:0] = "00"), both the common/segment output pins become "L" level.

<Static drive>

When display operation of the LCD is stopped (LCS [3:0] = "0000"), both the common/segment output pins become "L" level.

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

4.2. 1/2 Duty Output Waveform

1/2 duty output waveform is shown below.

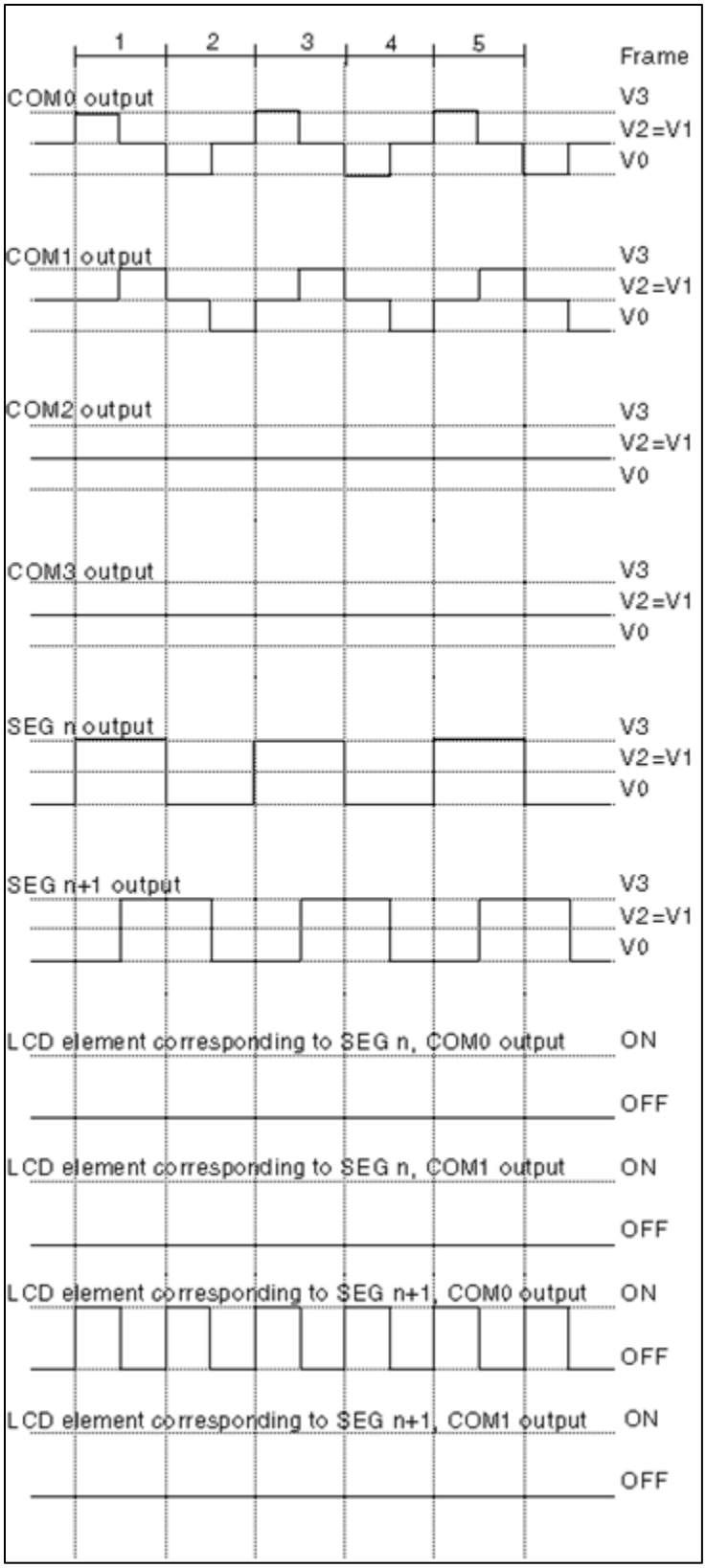
Only COM0 output and COM1 output are used for LCD display. COM2 output and COM3 output are not used.

1/2 Bias Output Waveform Case

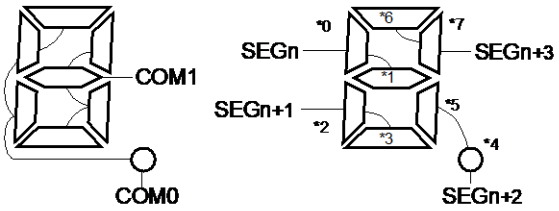
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

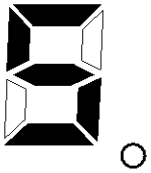
Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	-	-	0	0
SEG n+1 output	-	-	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2=V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
ON
OFF

■ LCD panel connection case and display data case (1/2 duty drive method)



Example for displaying 5:

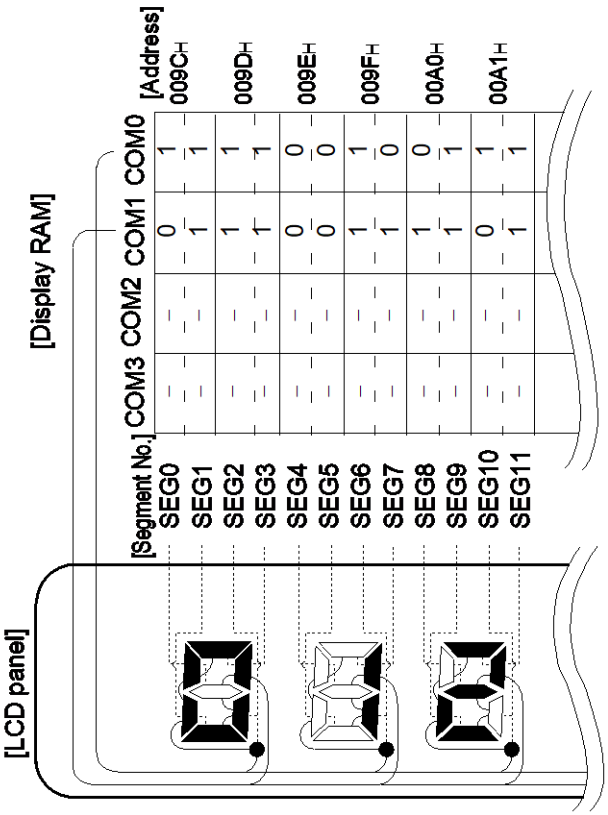


Address	COM3	COM2	COM1	COM0	
nH	bit3	bit2	bit1*1	bit0*0	SEGn
	bit7	bit6	bit5*3	bit4*2	SEGn+1
n+1H	bit3	bit2	bit1*5	bit0*4	SEGn+2
	bit7	bit6	bit5*7	bit4*6	SEGn+3

Address	COM3	COM2	COM1	COM0	
3950H	–	–	1	1	SEG0
	–	–	1	0	SEG1
3951H	–	–	1	0	SEG2
	–	–	0	1	SEG3

*0 to *7 : Indicates correspondence with display RAM bit2, 3, 6, 7 are unused

0 : OFF
1 : ON



LCD display	Example of data correspondence from 0 to 9							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	–	–	1	1	–	–	0	1
	–	–	1	1	–	–	1	1
1	–	–	0	0	–	–	0	0
	–	–	1	0	–	–	1	1
2	–	–	1	1	–	–	1	0
	–		1	1			0	1
3	–	–	1	0	–	–	1	0
	–	–	1	1	–	–	1	1
4	–	–	0	0	–	–	1	1
	–	–	1	0	–	–	1	1
5	–	–	1	0	–	–	1	1
	–	–	0	1	–	–	1	1
6	–	–	1	1	–	–	1	1
	–	–	0	1	–	–	1	1
7	–	–	0	0	–	–	0	1
	–	–	1	1	–	–	1	1
8	–	–	1	1	–	–	1	1
	–	–	1	1	–	–	1	1
9	–	–	1	0	–	–	1	1
	–	–	1	1	–	–	1	1

4.3. 1/3 Duty Output Waveform

1/3 duty output waveform is shown below.

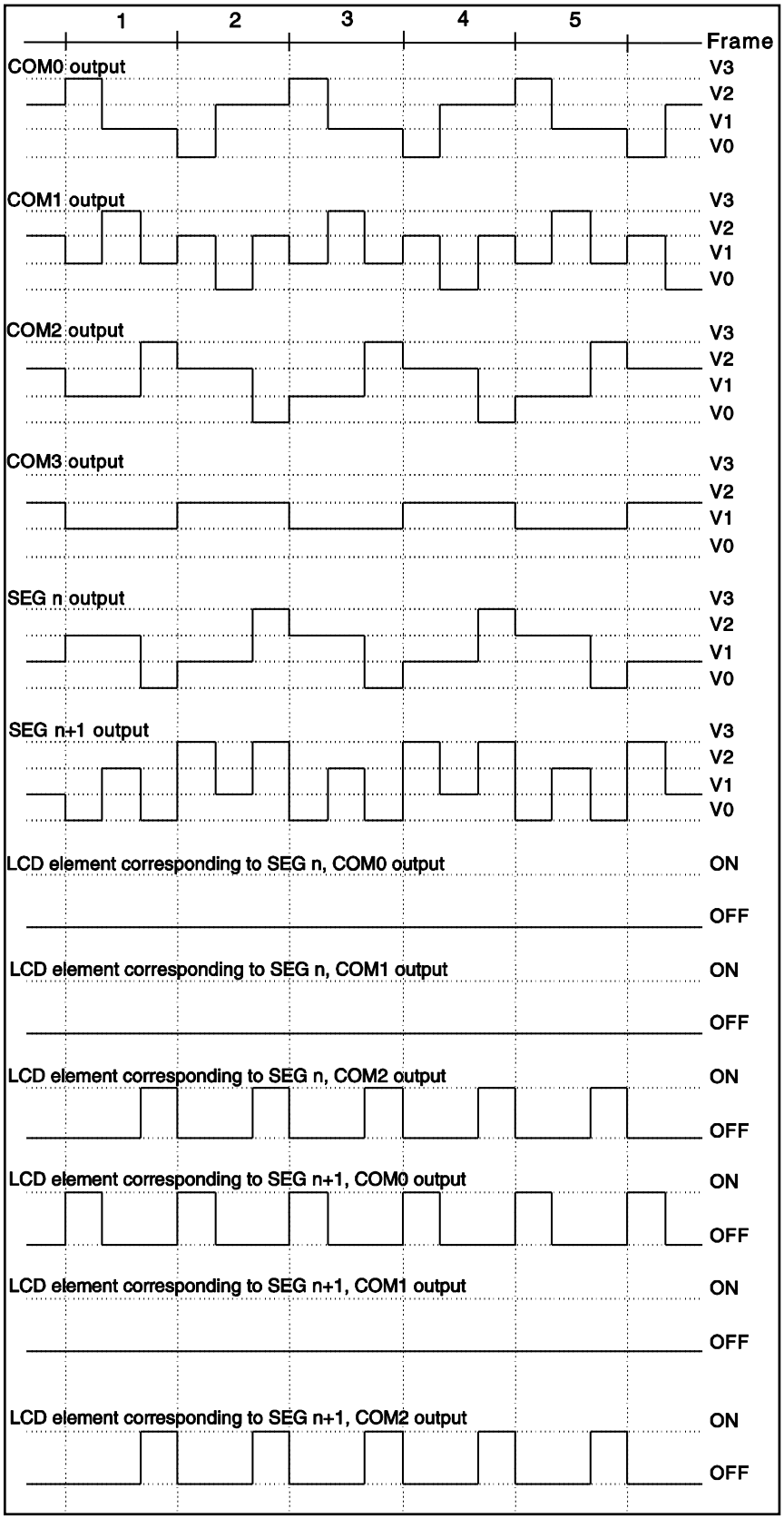
For 1/3 duty output mode, COM0 output, COM1 output, and COM2 output are used for LCD display. COM3 output is not used.

1/3 Bias Output Waveform Case

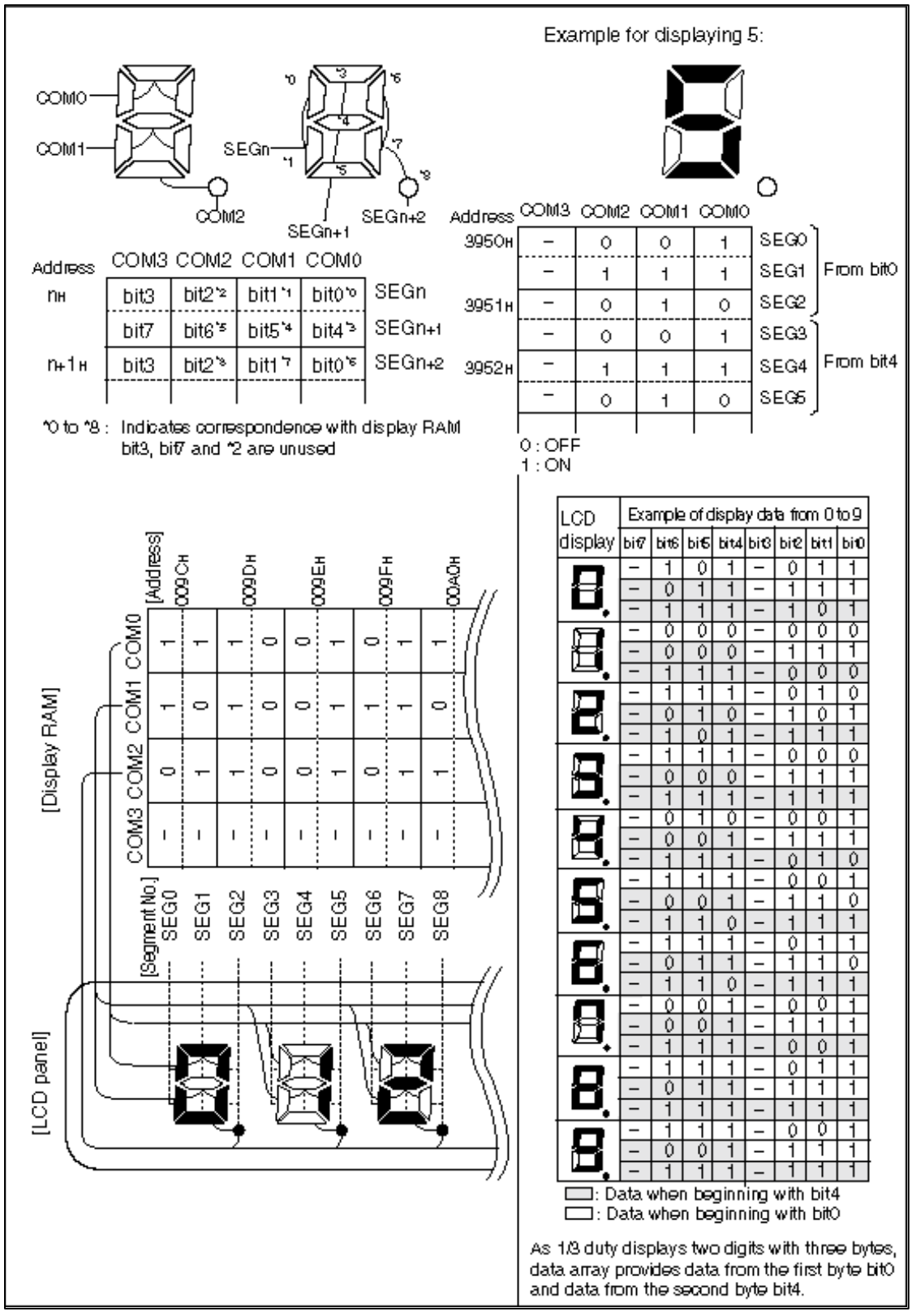
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	-	1	0	0
SEG n+1 output	-	1	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n, COM2 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
LCD element corresponding to SEG n+1, COM2 output
ON
OFF

■ LCD panel connection case and display data case (1/3 duty drive method)



4.4. 1/4 Duty Output Waveform

1/4 duty output waveform is shown below.

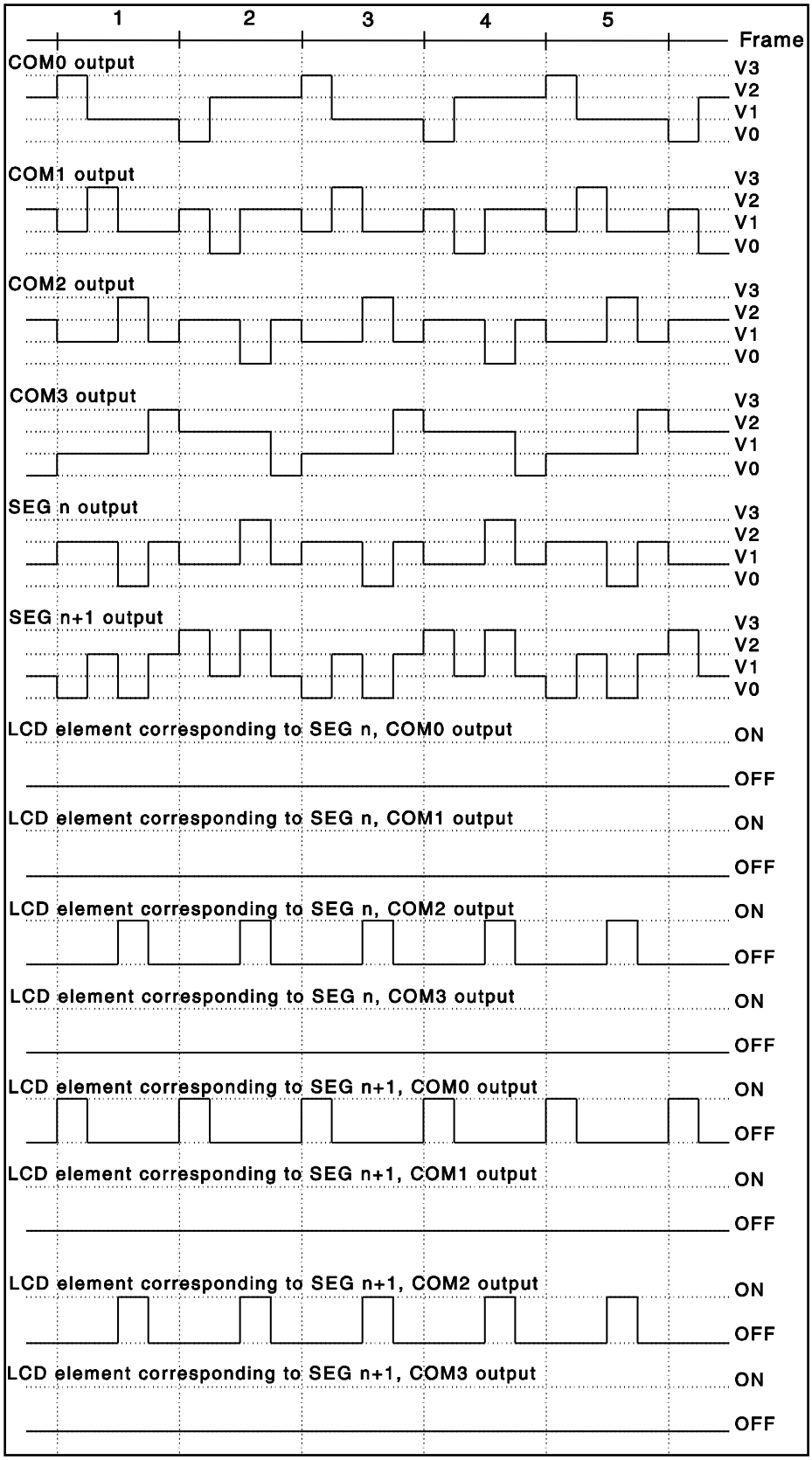
For 1/4 duty output mode, all of COM0 output, COM1 output, COM2 output, and COM3 output are used for LCD display.

1/4 Bias Output Waveform Case

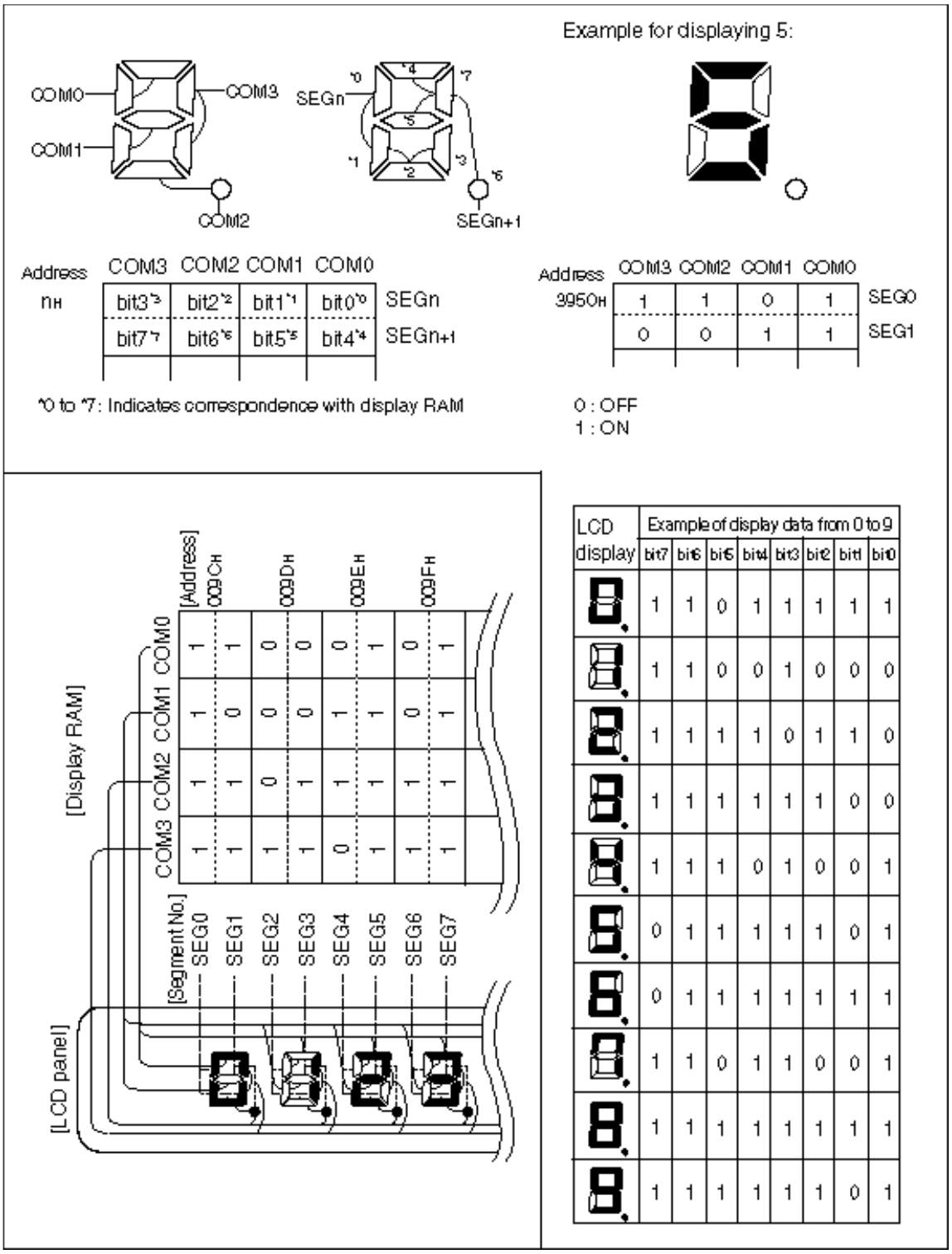
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

■ Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n output	0	1	0	0
SEG n+1 output	0	1	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n, COM2 output
LCD element corresponding to SEG n, COM3 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
LCD element corresponding to SEG n+1, COM2 output
LCD element corresponding to SEG n+1, COM3 output
ON
OFF

■ LCD panel connection case and display data case (1/4 duty drive method)



4.5. Static Drive Output Waveform

Static drive output waveform is shown below.

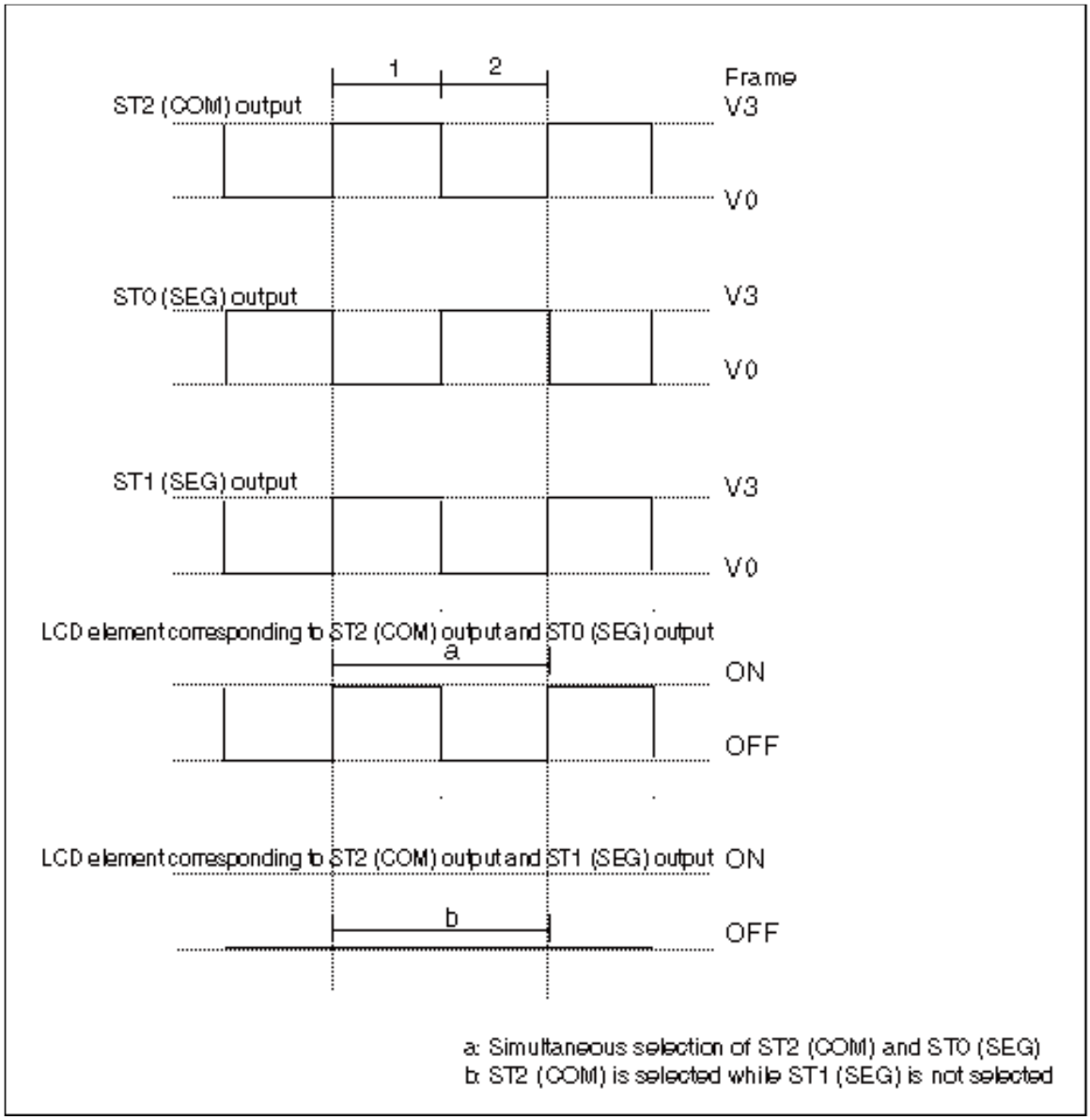
For the static drive output mode, the common/segment output pins (ST0 to ST8) are used for LCD display.

Static Drive Output Waveform Case

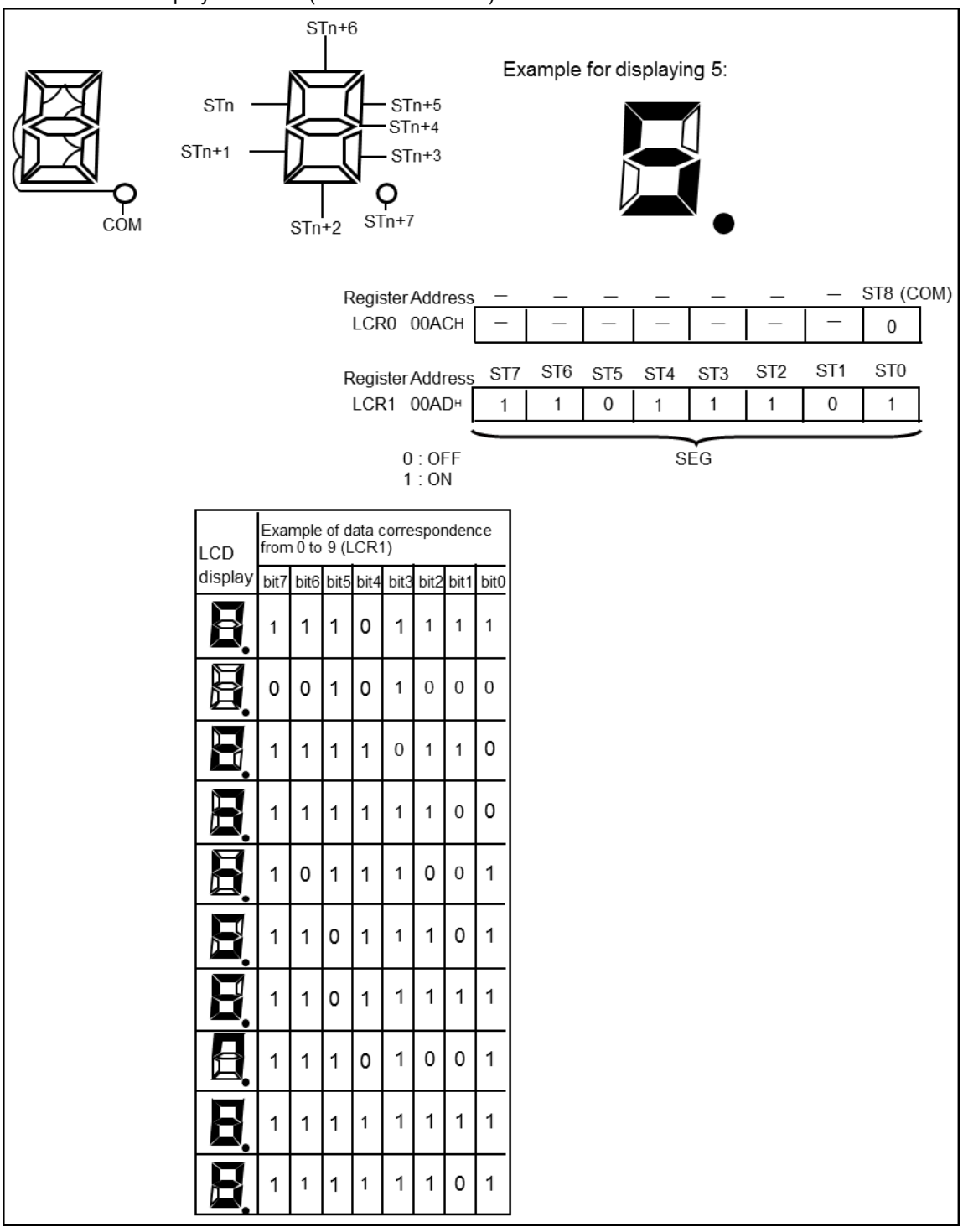
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

- Example of Content of Data Memory for Display (Example of static SEG output to the output pins ST0 to ST1 and static COM output to ST2)

Content of Data Memory for Display (LDR0, LDR1)		
ST2 Output LDR0[2]	**ST1 Output LDR0[1]**	**ST0 Output LDR0[0]**
0	0	1


1
2
Frame
ST2 (COM) output
V3
V0
ST0 (SEG) output
V3
V0
ST1 (SEG) output
V3
V0
LCD element corresponding to ST2 (COM) output and ST0 (SEG) output
a
ON
OFF
LCD element corresponding to ST2 (COM) output and ST1 (SEG) output
ON
b
OFF
a: Simultaneous selection of ST2 (COM) and ST0 (SEG)
b: ST2 (COM) is selected while ST1 (SEG) is not selected

■ LCD panel connection case and display data case (Static drive method)



LCD display	Example of data correspondence from 0 to 9 (LCR1)							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	1	1	1	0	1	1	1	1
	0	0	1	0	1	0	0	0
	1	1	1	1	0	1	1	0
	1	1	1	1	1	1	0	0
	1	0	1	1	1	0	0	1
	1	1	0	1	1	1	0	1
	1	1	0	1	1	1	1	1
	1	1	1	0	1	0	0	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	0	1

5. Setting

This section shows the setting of the LCD controller.

<Duty drive>

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 6.3, 6.4
Division resistor setting	LCD control register 0 (LCR0)	See 7.9, 7.11
Port setting	Segment output register (SEGER) Common output V pin control register(COMVER)	See 6.8, 6.9
Display data setting	Data memory for display (VRAM)	See 7.2
Frame cycle setting	LCD control register 0 (LCR0)	See 7.3
Duty selection (activation)		See 7.5
Display selection		See 7.7

Setting Necessary for Cancellation of LCD Display

Setting	Set Register	Set Method
Non-display selection	LCD control register 0 (LCR0)	See 7.7

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register 0 (LCR0)	See 7.6

Setting Necessary for LCD Display during PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)

Setting	Set Register	Set Method
Display selection during PSS timer mode(main oscillation operation/ sub oscillation operation)	LCD control register 0 (LCR0)	See 7.8
Transition to PSS timer mode(main oscillation operation/ sub oscillation operation)	See chapter of "LOW-POWER CONSUMPTION".	-

< Static drive >

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 6.3, 6.4
Port setting	Set the pin to LCDC output. See chapter of "I/O PORTS".	See 0, 6.9
	LCD control register (LCRS)	See 6.5
Display data setting	Data memory for display (LDR0,LDR1)	See 6.6
Frame cycle setting	LCD control register (LCRS)	See 7.3

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register (LCRS)	See 7.6

Setting Necessary for LCD Display during PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)

Setting	Set Register	Set Method
Display selection during PSS timer mode(main oscillation operation/ sub oscillation operation)	LCD control register (LCRS)	See 7.8
Transition to PSS timer mode(main oscillation operation/ sub oscillation operation)	See chapter of "LOW-POWER CONSUMPTION".	-

6. Registers

This section explains the registers of the LCD controller.

■ Register map

Table 6-1 Registers of the LCD Controller

Register Abbreviation	Register Name	Reference
LCR0	LCD control register 0	6.1
VRAMn(n=0 to15)	Data memory for display	6.2
LCR1	LCDC control register 1	6.3
LCDCMR	Common pin switching register	6.4
LCRS	LCDC Static control register	6.5
LDR	Static LCD display data register	6.6

Table 6-2 Registers of LCD Port-related

Register Abbreviation	Register Name	Reference
LCD_KEYCDR	key code register	6.7
SEGER	Segment output Register	6.8
COMVER	Common output V pin control register	6.9

■ **Register Offset Address Map (LCDC)**

OFFSET_ADDRESS	REGISTER_NAME				ACCESS_SIZE
	+3	+2	+1	+0	
0x00000000	LCDCMR	LCRS	LCR0	LCR1	B, H, W
0x00000004	VRAM0	VRAM1	VRAM2	VRAM3	B, H, W
0x00000008	VRAM4	VRAM5	VRAM6	VRAM7	B, H, W
0x0000000C	VRAM8	VRAM9	VRAM10	VRAM11	B, H, W
0x00000010	VRAM12	VRAM13	VRAM14	VRAM15	B, H, W
0x00000014	LDR		Reserved		B, H, W

■ **Register Offset Address Map (LCDE)**

OFFSET_ADDRESS	REGISTER_NAME				ACCESS_SIZE
	+3	+2	+1	+0	
0x00000000	LCD_KEYCDR				W
0x00000004	SEGER				B, H, W
0x00000008	COMVER				B, H, W

6.1. LCD Control Register 0: LCR0

The bit configuration of the LCD control register 0 is shown below.

This register selects a frame cycle with its clock to be used, selects display mode, enables the LCD controller to operate in the condition for selecting display/non-display mode, and controls LCD drive power.

■ LCR0

bit	7	6	5	4	3	2	1	0
Field	CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	1	0	0	0	0

[bit7] CSS: Clock selection bit

Select a clock to be used for this module.

Bit	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCEN: PSS Timer mode (main oscillation operation / sub oscillation operation) operation enable

Bit	Operation
0	LCD display stop in the PSS timer mode(main oscillation operation/ sub oscillation operation)
1	LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)

When LCEN="1", do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

[bit5] VSEL: LCD drive power control

Bit	Operation
0	Internal division resistor disconnection
1	Internal division resistor connection

When an external division resistor is to be connected, the LCD drive power control bit (VSEL) must be set to "0".

[bit4] BK: Blanking selection

Bit	Operation
0	LCD display
1	LCD no display

[bit3, bit2] MS1, MS0: Display mode selection

MS1	MS0	Display mode
0	0	LCD operation stop
0	1	1/2 duty output mode (Time division number: N=2, COM0, COM1)
1	0	1/3 duty output mode (Time division number: N=3, COM0 to COM2)
1	1	1/4 duty output mode (Time division number: N=4, COM0 to COM3)

When the display mode selection bits (MS [1:0]) are set to "00", the LCD controller's operation is stopped, and the common pin/segment pin output "L" level.

[bit1, bit0] FP1, FP0: Frame cycle

FP1	FP0	Frame cycle
0	0	For CSS=0 : $(2^{11} \times N)/F_{CL}$ For CSS=1 : $(2^{3} \times N)/F_{CL}$
0	1	For CSS=0 : $(2^{12} \times N)/F_{CL}$ For CSS=1 : $(2^{4} \times N)/F_{CL}$
1	0	For CSS=0 : $(2^{13} \times N)/F_{CL}$ For CSS=1 : $(2^{5} \times N)/F_{CL}$
1	1	For CSS=0 : $(2^{14} \times N)/F_{CL}$ For CSS=1 : $(2^{6} \times N)/F_{CL}$

F_{CL} : Main clock (LCR0:CSS=0) or sub clock (LCR0:CSS=1)

N : Time division number (depends on the setting of the display mode selection bits MS1, MS0)

Select the setting for FP1 and FP0 that is the optimum condition for the frame frequency of the LCD panel to be used.

6.2. Data Memory for Display: VRAM

The data memory for display is shown below.

Memory (VRAM) area for setting of data for display

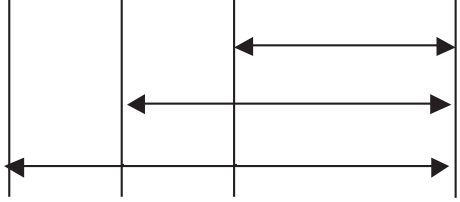
- VRAM0 (SEG0, SEG1)
- VRAM1 (SEG2, SEG3)
- VRAM2 (SEG4, SEG5)
- VRAM3 (SEG6, SEG7)
- VRAM4 (SEG8, SEG9)
- VRAM5 (SEG10, SEG11)
- VRAM6 (SEG12, SEG13)
- VRAM7 (SEG14, SEG15)
- VRAM8 (SEG16, SEG17)
- VRAM9 (SEG18, SEG19)
- VRAM10 (SEG20, SEG21)
- VRAM11 (SEG22, SEG23)
- VRAM12 (SEG24, SEG25)
- VRAM13 (SEG26, SEG27)
- VRAM14 (SEG28, SEG29)
- VRAM15 (SEG30, SEG31)

bit	7	6	5	4	3	2	1	0
Field	D07	D06	D05	D04	D03	D02	D01	D00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

The RAM for display can be read /written with any timing regardless of the LCD controller/driver operation.

Correspondence between VRAM and Common Pins/Segment Pins

VRAM0	bit3	bit2	bit1	bit0	SEG0
	bit7	bit6	bit5	bit4	SEG1
VRAM1	bit3	bit2	bit1	bit0	SEG2
	bit7	bit6	bit5	bit4	SEG3
VRAM2	bit3	bit2	bit1	bit0	SEG4
	bit7	bit6	bit5	bit4	SEG5
VRAM3	bit3	bit2	bit1	bit0	SEG6
	bit7	bit6	bit5	bit4	SEG7
VRAM4	bit3	bit2	bit1	bit0	SEG8
	bit7	bit6	bit5	bit4	SEG9
VRAM5	bit3	bit2	bit1	bit0	SEG10
	bit7	bit6	bit5	bit4	SEG11
VRAM6	bit3	bit2	bit1	bit0	SEG12
	bit7	bit6	bit5	bit4	SEG13
VRAM7	bit3	bit2	bit1	bit0	SEG14
	bit7	bit6	bit5	bit4	SEG15
VRAM8	bit3	bit2	bit1	bit0	SEG16
	bit7	bit6	bit5	bit4	SEG17
VRAM9	bit3	bit2	bit1	bit0	SEG18
	bit7	bit6	bit5	bit4	SEG19
VRAM10	bit3	bit2	bit1	bit0	SEG20
	bit7	bit6	bit5	bit4	SEG21
VRAM11	bit3	bit2	bit1	bit0	SEG22
	bit7	bit6	bit5	bit4	SEG23
VRAM12	bit3	bit2	bit1	bit0	SEG24
	bit7	bit6	bit5	bit4	SEG25
VRAM13	bit3	bit2	bit1	bit0	SEG26
	bit7	bit6	bit5	bit4	SEG27
VRAM14	bit3	bit2	bit1	bit0	SEG28
	bit7	bit6	bit5	bit4	SEG29
VRAM15	bit3	bit2	bit1	bit0	SEG30
	bit7	bit6	bit5	bit4	SEG31
	COM3	COM2	COM1	COM0	



RAM area and a common pin used for 1/2 duty output mode

RAM area and a common pin used for 1/3 duty output mode

RAM area and a common pin used for 1/4 duty output mode

6.3. LCDC Control Register 1: LCR1

The bit configuration of the LCDC control register 1 is shown below.

■ LCR1

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7 to bit0] Reserved

When the LCD is used, set "11111111" always. When it is not used, set "00000000".

6.4. Common Pin Switching Register: LCDCMR

The bit configuration of the common pin switching register is shown below.

■ LCDCMR

bit	7	6	5	4	3	2	1	0
Field	DTCH	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W0	R/W0	R/W0	R/W1	R/W1	R/W1	R/W1
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7] DTCH: Bias selection

Bit	Operation
0	1/3 bias
1	1/2 bias

[bit6 to bit4] Reserved

When the LCD is used, set "000" always.

[bit3 to bit0] Reserved

When the LCD is used, set "1111" always. When it is not used, set "0000".

6.5. LCDC Static Control Register: LCRS

The bit configuration of the LCDC static control register is shown below.

■ LCRS

bit	7	6	5	4	3	2	1	0
Field	SCSS	LCSEN	LCS3	LCS2	LCS1	LCS0	FPS1	FPS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7] SCSS: Frame cycle generation clock for static drive selection bit

Bit	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCSEN: PSS Timer mode (main oscillation operation / sub oscillation operation) operation enable

Bit	Operation
0	LCD display stop in the PSS timer mode(main oscillation operation/ sub oscillation operation)
1	LCD display in the PSS timer mode(main oscillation operation/ sub oscillation operation)

When LCSEN="1", please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode.

LCD display stop when power shutdown.

[bit5:2] LCS3 to LCS0: Static drive selection

LCS3	LCS2	LCS1	LCS0	Static Drive Selection Port
				Common / Segment Output
0	0	0	0	OFF
0	0	0	1	ST0, ST1

LCS3	LCS2	LCS1	LCS0	Static Drive Selection Port
				Common / Segment Output
0	0	1	0	ST0 to ST2
0	0	1	1	ST0 to ST3
0	1	0	0	ST0 to ST4
0	1	0	1	ST0 to ST5
0	1	1	0	ST0 to ST6
0	1	1	1	ST0 to ST7
1	X	X	X	ST0 to ST8

Note:

- *For the use method for the static drive selection, refer to "4.5 Static Drive Output Waveform".*

[bit1:0] FPS1, FPS0: Frame cycle

FPS1	FPS0	Frame Cycle
0	0	For SCSS=0 : $(2^{11} \times 4)/F_{CL}$ For SCSS=1 : $(2^{3} \times 4)/F_{CL}$
0	1	For SCSS=0 : $(2^{12} \times 4)/F_{CL}$ For SCSS=1 : $(2^{4} \times 4)/F_{CL}$
1	0	For SCSS=0 : $(2^{13} \times 4)/F_{CL}$ For SCSS=1 : $(2^{5} \times 4)/F_{CL}$
1	1	For SCSS=0 : $(2^{14} \times 4)/F_{CL}$ For SCSS=1 : $(2^{6} \times 4)/F_{CL}$

F_{CL} : Main clock (LCRS:SCSS=0) or sub clock (LCRS:SCSS=1)

Select the setting for FPS1 and FPS0 that is the optimum condition for the frame frequency of the LCD panel to be used.

The frame cycle during static drive is equal to the cycle of the 1/4 duty output mode.

The static drive is enabled when SEGER setting and any setting other than LCS [3:0] = "0000" are executed.

The duty drive is enabled when SEGER and LCS [3:0] = "0000" are set.

6.6. Static LCD Display Data Register: LDR

The bit configuration of the static LCD display data register is shown below.

■ LDR0

bit	15	14	13	12	11	10	9	8
Field	-		-	-	-	-	-	ST8
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7:1] Undefined bits

The read value is always "0". Writing has no effect on operation.

[bit0] ST8: Static output data

This bit is static output data for ST8.

■ LDR1 : Address 05BD$_H$ (Access : Byte, Half-word, Word)

bit	7	6	5	4	3	2	1	0
Field	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit7:0] ST7 to ST0: Static output data

These bits are static output data for ST0 to ST7.

Static Output Data Correspondence Table

LDR	Operation
ST8	ST8 static output data
ST7	ST7 static output data
ST6	ST6 static output data
ST5	ST5 static output data
ST4	ST4 static output data
ST3	ST3 static output data
ST2	ST2 static output data
ST1	ST1 static output data
ST0	ST0 static output data

This register is for setting of data output to the LCD static drive port (set by the LCRS register). The LCD static drive is executed by inverting output of the specified data periodically.

Common output pins are not specified especially. Assign common output to one of segments. For example, if ST0 to ST8 are assigned static drive pins, ST8 is a common pin and set to LDR0:ST8 = "0". Furthermore, by setting LDR1: ST [7:0] = "11111111", the LCD selected with ST0 to ST7 is displayed.

6.7. Key Code Register: LCD_KEYCDR

This section shows the bit configuration of the key code register.

This register makes the write setting of a register that has the function for protection against erroneous writing (SEGER, COMVER). If writing to this register is not done using the prescribed method, writing to the relevant register is invalid. This register is valid only for word access.

■ KEYCDR

bit	31	30	29	28	27	26	25	24
Field	KEY1	KEY0	SIZE1	SIZE0	Reserved			
Attribute	R0,W	R0,W	R0,W	R0,W	R0,WX	R0,WX	R0,WX	R0,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved				ADR11	ADR10	ADR09	ADR08
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W	R0,W	R0,W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02	ADR01	ADR00
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:30] KEY1 to KEY0: Key code setting bits

You need to write "00", "01", "10", and "11" continuously to these bits in this order.
The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, you need to set the key codes again from the beginning.

Bit31:30		Description
0	0	1st key code
0	1	2nd key code
1	0	3rd key code
1	1	4th key code

[bit29:28] SIZE [1:0]: Access size setting bits

These bits set the access size for writing to a key code target register.

Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY [1:0] in this order.

Bit29:28		Description
0	0	Set byte access.
0	1	Set half-word access.
1	0	Set word access.
1	1	Reserved

[bit27:12] Reserved

Always write "0" to these bits.

[bit11:0] ADR11 to ADR0: Address specification bits

These bits set the lower 12 bits of the address of a key code target register. Write the same data to these bits when writing the key codes of "00", "01", "10", and "11" to KEY [1:0] in this order.

Bit11:0	Description
Set the lower 12 bits of the address of a key code target register.	

Notes:

The following are the notes on key code setting.

- *You need to write "00", "01", "10", and "11" continuously to KEY [1:0] in this order in 4 write operations.*

The key code setting becomes invalid immediately upon any writing to these bits in a different order. In such cases, you need to set the key codes again from the beginning.

- *You need to write the same pair of values to SIZE [1:0] and ADR [11:0] 4 times when writing data to KEY [1:0] 4 times.*
  The key code setting becomes invalid upon any writing of a different pair of values. In such cases, you need to set the key codes again from the beginning.
- *When the KEYCDR register is read during writing to KEY [1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
- *When access to the address set in ADR [11:0] occurs during writing to KEY [1:0], the key code setting becomes invalid. In such cases, you need to set the key codes again from the beginning.*
  When access to any register other than the above occurs, the key code setting does not become invalid.
  Writing the remaining data to KEY [1:0] subsequently makes the key code setting valid.
- *Once the address set in KEYCDR: ADR [11:0] is accessed after KEY [1:0] is set normally, it is required to set the key codes again to access the address again.*

6.8. Segment Output Register: SEGER

This section shows the bit configuration of the Segment output Register.

Segment output register (SEGER) is a register which control the segment output enable of LCD controller duty driving driver.

■ SEGER

bit	31	30	29	28	27	26	25	24
Field	SEGE7	SEGE6	SEGE5	SEGE4	SEGE3	SEGE2	SEGE1	SEGE0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	SEGE15	SEGE14	SEGE13	SEGE12	SEGE11	SEGE10	SEGE9	SEGE8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	SEGE23	SEGE22	SEGE21	SEGE20	SEGE19	SEGE18	SEGE17	SEGE16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SEGE31	SEGE30	SEGE29	SEGE28	SEGE27	SEGE26	SEGE25	SEGE24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

[bit31:0] SEGEn (n=0 to 31)

SEGEn (n=0 to 31)	Operation
0	Disable LCDC segment output
1	Enable LCDC segment output

6.9. Common Output V Pin Control Register: COMVER

This section shows the bit configuration of the common output V pin control register.

Common output V pin control register (COMVER) is a register which control the common output enable of LCD controller static driving driver, input enable and selection of LCD controller reference voltage.

■ COMVER

bit	31	30	29	28	27	26	25	24
Field	VE3	VE2	VE1	VE0	COME3	COME2	COME1	COME0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Protection attribute	-	-	-	-	-	-	-	-
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
Protection attribute	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1

[bit31] VE3

VE3	Operation
1	Use V3 pin as the LCDC V3 reference voltage input
0	VCC is used for the LCDC V3 reference voltage

Please to switch on the main oscillation state when switching external V3 and VCC as LCDC reference supply V3 input.

[bit30:28] VEn (n=0 to 2)

VEn (n=0 to 2)	Operation
0	Disable LCDC reference voltage input
1	Enable LCDC reference voltage input

[bit27:24] COMEn (n=0 to 3)

COMEn (n=0 to 3)	Operation
0	Disable LCDC common output
1	Enable LCDC common output

[bit23:0] Reserved

The read value is always "1". Writing has no effect on operation.

7. Q&A

This section shows Q&A of the LCD controller.

7.1. How can I Set Pins to COM Output Pins or SEG Output Pins?

7.2. How to Set VRAM?

7.3 How can I Setting the Frame Cycle?7.4. How can I Set the Bias?

7.5. How can I Set the Duty?

7.6. How can I Control the LCD Operation Start/Stop?

7.7. How can I Execute/Cancel the Display?

7.8. How can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)?

7.9. How can I Select either Internal or External for the Division Resistor?

7.10. How can I Select Pin of V3 Voltage?

7.11. How can I Select either Internal or External for the Division Resistor?

7.12. How can I Adjust the Brightness When the Internal Division Resistor is Used?

7.13. How can I Block the Current with the External Division Resistor When the LCD Stops?

7.14. How can I Display/Non-display the LCD with Static Drive (ST0 to ST8)?

7.1. How can I Set Pins to COM Output Pins or SEG Output Pins?

This section shows how to set pins to COM output pins or SEG output pins.

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for COM output and SEG output to peripheral output. See following table for pin setting.

<Duty drive>

Pin	V0-V3 and COM and SEG Setting Method	Port Setting Method
V0	Set pins to LCDC V0/V1/V2/V3 (LCDC reference power input). See 6.9.	See "Port Configuration 5.6 Analog Function Input or Output"
V1		
V2		
V3		
COM0		
COM1		
COM2		
COM3		
SEG0	See 6.8.	
SEG1		
SEG2		
SEG3		
SEG4		
SEG5		
SEG6		
SEG7		
SEG8		
SEG9		
SEG10		
SEG11		
SEG12		
SEG13		
SEG14		
SEG15		
SEG16		
SEG17		
SEG18		

Pin	V0-V3 and COM and SEG Setting Method	Port Setting Method
SEG19	See 6.8.	See "Port Configuration 5.6 Analog Function Input or Output"
SEG20		
SEG21		
SEG22		
SEG23/ST0		
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

<Static drive >

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for static drive to peripheral output. See 6.8. for pin setting.

Pin	V0-V3 and SEG Setting Method	Port Setting Method
V0	Set pins to LCDC V0/V3 (LCDC reference power input). See 6.9.	See "Port Configuration 5.6 Analog Function Input or Output"
V3		
SEG23/ST0	See 6.8.	
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

7.2. How to Set VRAM?

This section shows how to set VRAM.

<Duty drive>

The matrix of pins and bit locations of VRAM (n) is shown below. (n = 0 to 15)

1/2 Duty

Pin	COM1	COM0
SEG 2n	bit1	bit0
SEG 2n+1	bit5	bit4

1/3 Duty

Pin	COM2	COM1	COM0
SEG 2n	bit2	bit1	bit0
SEG 2n+1	bit6	bit5	bit4

1/4 Duty

Pin	COM3	COM2	COM1	COM0
SEG 2n	bit3	bit2	bit1	bit0
SEG 2n+1	bit7	bit6	bit5	bit4

(Non-selection waveforms are output from irrelevant pins.)

Example: 1/4 duty

When the bit6 of VRAMn is set to "1", the selection waveform is output from the SEGn+1 of COM2.

For bits with "0"setting, non-selection waveforms are output to pins.

7.3. How can I Setting the Frame Cycle?

This section shows how to set pins to COM output pins or SEG output pins.

<Duty drive>

The frame cycle can be set with the frame cycle bits (LCR0: FP [1:0]).The following settings are available.

Frame Cycle (When a Main Clock Is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{11} \times N)$/ Main clock frequency	Set "00".
$(2^{12} \times N)$/ Main clock frequency	Set "01".
$(2^{13} \times N)$/ Main clock frequency	Set "10".
$(2^{14} \times N)$/ Main clock frequency	Set "11".

N (Time division number) = Value of MS [1:0] + "1"

Frame Cycle (When a Sub Clock Is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{3} \times N)$/ Sub clock frequency	Set "00".
$(2^{4} \times N)$/ Sub clock frequency	Set "01".
$(2^{5} \times N)$/ Sub clock frequency	Set "10".
$(2^{6} \times N)$/ Sub clock frequency	Set "11".

N (Time division number) = Value of MS [1:0] + "1"

< Static drive >

The frame cycle can be set with the frame cycle bits (LCRS: FPS [1:0]).The following settings are available.

Frame Cycle	Selection Value
	Frame Cycle Bits (FPS[1:0])
$(2^{11} \times 4)$/Main clock (F_{CL})	Set "00".
$(2^{12} \times 4)$/ Main clock (F_{CL})	Set "01".
$(2^{13} \times 4)$/ Main clock (F_{CL})	Set "10".
$(2^{14} \times 4)$/ Main clock (F_{CL})	Set "11".

7.4. How can I Set the Bias?

This section shows how to set the bias.

<Duty drive>

Set the bias selection bit (LCDCMR: DTCH).

Bias	Bias Selection Bit (DTCH)
To set 1/3 bias	Set "0".
To set 1/2 bias	Set "1".

7.5. How can I Set the Duty?

This section shows how to set the duty.

<Duty drive>

Set the display mode selection bits (LCR0: MS [1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])	N (Time Division Number)
LCD operation stop (Pin output "L")	Set "00".	-
To set 1/2 duty output mode	Set "01".	2
To set 1/3 duty output mode	Set "10".	3
To set 1/4 duty output mode	Set "11".	4

The display mode selection bits are also the control bit for operation start/stop.

7.6. How can I Control the LCD Operation Start/Stop?

This section shows how to control the LCD operation start/stop.

<Duty drive>

Operation start/stop can be controlled with the display mode selection bits (LCR0: MS [1:0]).

See "7.5 How can I set the duty?".

<Static drive>

Operation start/stop can be controlled with the display mode selection bits (LCRS: LCS [3:0]).

7.7. How can I Execute/Cancel the Display?

This section shows how to execute/cancel the LCD display.

<Duty drive>

The two methods below are available.

■ Setting of the blanking selection bit (LCR0:BK)

Control Detail	Blanking Selection Bit (BK)
To execute LCD display	Set "0".
To cancel LCD display (A non-selection waveform is output to a segment pin.)	Set "1".

■ Cancellation of display with operation stop by the display mode selection bits (LCR0: MS [1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])
LCD operation stop ("L" output from the common pin and segment pin.)	Set "00".

7.8. How can I Display during the PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation)?

This section shows how to display during the PSS timer mode (main oscillation operation/ sub oscillation operation).

<Duty drive>

Set the PSS timer mode (main oscillation operation/ sub oscillation operation) operation enable bit (LCR0: LCEN).

Control Detail	PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation) Operation Enable Bit (LCEN)
For no LCD display in the PSS timer mode (main oscillation operation/ sub oscillation operation)	Set "0".
For LCD display in the PSS timer mode (main oscillation operation/ sub oscillation operation)	Set "1".

When LCEN="1", please do not stop the clock oscillation which selected by the CSS bit during the transition to PSS mode. LCD display stop when power shutdown.

<Static drive>

Set the PSS timer mode (main oscillation operation/ sub oscillation operation) operation enable bit (LCRS: LCSEN).

Control Detail	PSS Timer Mode (Main Oscillation Operation/ Sub Oscillation Operation) Operation Enable Bit (LCSEN)
For no LCD display in the PSS timer mode (main oscillation operation/ sub oscillation operation)	Set "0".
For LCD display in the PSS timer mode (main oscillation operation/ sub oscillation operation)	Set "1".

When LCSEN="1", please do not stop the clock oscillation which selected by the SCSS bit during the transition to PSS mode. LCD display stop when power shutdown.

7.9. How can I Select either Internal or External for the Division Resistor?

This section shows how to select either internal or external for the division resistor.

<Duty drive>

Set the LCD drive power control bit (LCR0: VSEL).

Control Detail	LCD Drive Power Control Bit (VSEL)
For use of the external division resistor (The internal division resistor is disconnected.)	Set "0".
For use of the internal division resistor (The internal division resistor is connected.)	Set "1".

7.10. How can I Select Pin of V_3 Voltage?

This section shows how to select the pin of V_3 voltage.

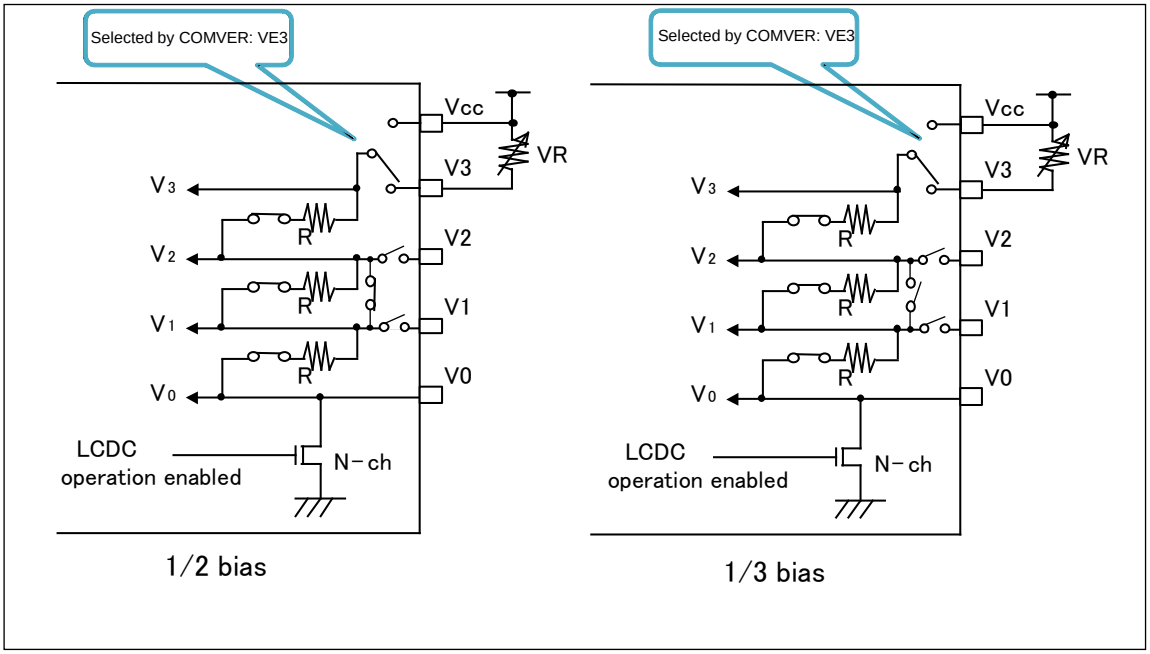
Set the COMVER: VE3 bit. See chapter of 6.9 for details.

7.11. How can I Select either Internal or External for the Division Resistor?

This section shows how to select either internal or external for the division resistor.

■ When the internal division resistor is selected

Even when the internal division resistor is used if the V3 pin is used as V_3 voltage, the external resistor must be connected between Vcc and V3.



■ When the external division resistor is selected

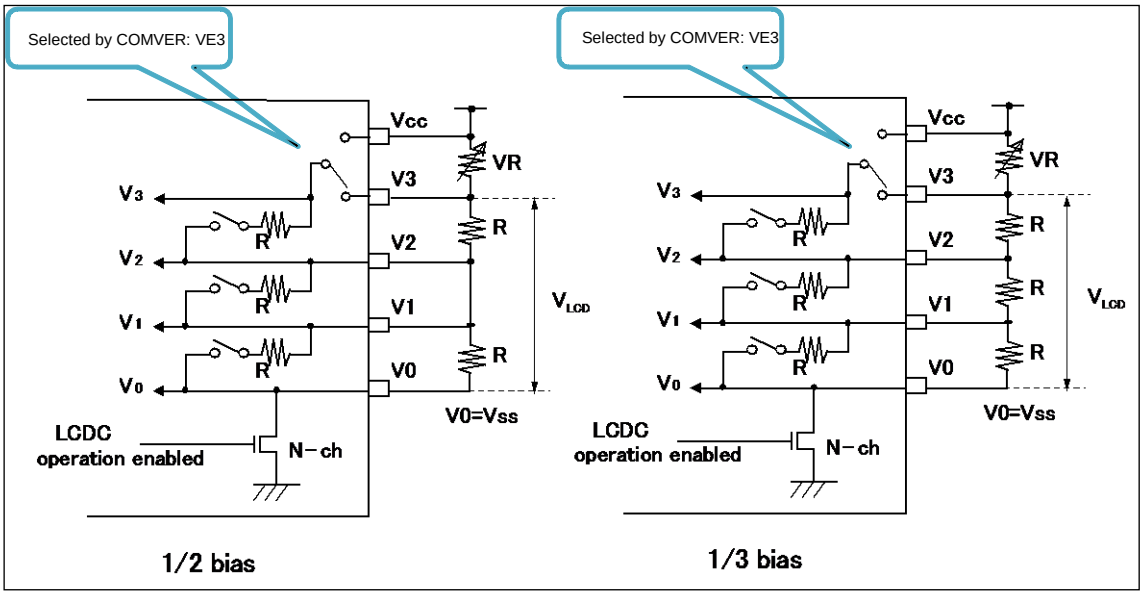
Voltage for the LCD drive is set with the external division resistor connected to the power pins for LCD drive (V0 to V3).

Setting of LCD drive voltage

	V3	V2	V1	V0
1/2 bias	VLCD	1/2VLCD	1/2VLCD	VSS
1/3 bias	VLCD	2/3VLCD	1/3VLCD	VSS

V_0 to V_3 : Voltage of V0 to V3 pins

V_{LCD} : Operation voltage of the LCD



To avoid the effect of the internal division resistor, the LCD drive power control bit (LCR0: VSEL) must be set to "0" to disconnect the internal division resistor.

7.12. How can I Adjust the Brightness When the Internal Division Resistor is Used?

This section shows how to adjust the brightness when the internal division resistor is used.

If desired brightness cannot be obtained with the use of the internal division resistor, adjust the voltage V3 by putting a variable resister (VR) between the outer terminals Vcc and V3.

7.13. How can I Block the Current with the External Division Resistor When the LCD Stops?

This section shows how to block the current with the external division resistor when the LCD stops.

The V0 pin is connected to the Vss (GND) via a transistor internally. Therefore, when the external division resistor is used, by connecting the Vss side of the external division resistor to the V0 pin, the current which flows on the LCD controller which is stopped can be blocked. Block the current with the display mode selection bits (MS [1:0] = "00").

7.14. How can I Display/Non-display the LCD with Static Drive (ST0 to ST8)?

This section shows how to display/non-display the LCD with static drive (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000", and set the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1: ST [7:0] without change of (LCS [3:0]).

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1: ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

8. Sample Program

This section shows sample program.

Setting procedure 1

With the 1/2 duty drive method, make the LCD display a four-digit number, "0 1 2 3".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM07

3. - Setting of control register

	Register name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

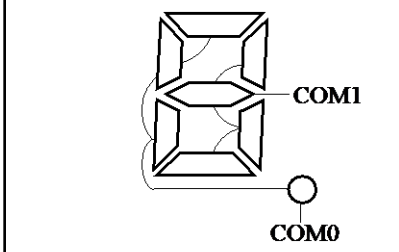
Program 1

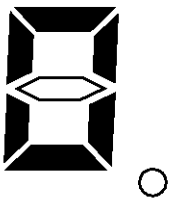
```
void LCD_sample_1(void)
{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();    /* Set the LCD controller pin to */
                                    /* peripheral output.   */

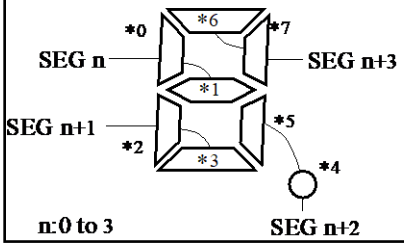
        IO_VRAM00 = 0x31;
        IO_VRAM01 = 0x32;
        IO_VRAM02 = 0x00;
        IO_VRAM03 = 0x22;
        IO_VRAM04 = 0x32;
        IO_VRAM05 = 0x30;
        IO_VRAM06 = 0x22;
        IO_VRAM07 = 0x32;

        IO_LCR1.byte = 0xFF;        /* Set to FF. */
        IO_LCDCMR. byte = 0x0F;     /* 1/2 bias */
        IO_LCR0.byte = 0x04;        /* Setting value =0000_0100 */
                                    /* bit7 = 0 Clock selection   Main clock */
                                    /* bit6 = 0 LCEN   Display stopped with PSS mode   */
                                    /* bit5 = 0 VSEL Internal division resistor disconnected
                                    */
                                    /* bit4 = 0 BK Blanking selection bit */
                                    /* bit3-2 = 01 MS[1:0]          1/2 duty mode */
                                    /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 0 and its value (in parentheses)







COM3	COM2	COM1	COM0	
–	–	*1(0)	*0(1)	SEG0
–	–	*3(1)	*2(1)	SEG1
–	–	*5(1)	*4(0)	SEG2
–	–	*7(1)	*6(1)	SEG3

Setting procedure 2

With the1/3 duty drive method, make the LCD display a four-digit number , "0 1 2 3 ".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM05

3. - Setting of control register

Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

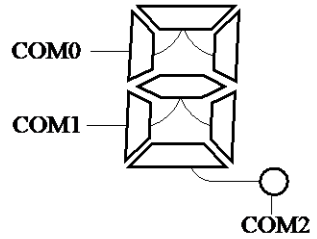
Program 2

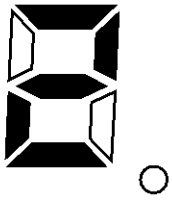
```
void LCD_sample_2(void)
{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.    */

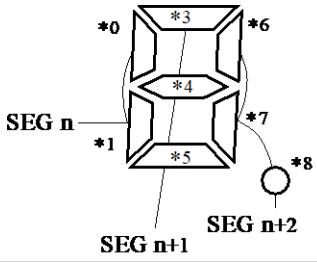
        IO_VRAM00 = 0x53;
        IO_VRAM01 = 0x03;
        IO_VRAM02 = 0x30;
        IO_VRAM03 = 0x72;
        IO_VRAM04 = 0x01;
        IO_VRAM05 = 0x37;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCR0.byte = 0x08;            /* Setting value =0000_1000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN Display stopped with PSS mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 10 MS[1:0] 1/3 duty   mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 2 and its value (in parentheses)







COM3	COM2	COM1	COM0		
–	(-)	*1(1)	*0(0)	SEG0	First digit
–	*5(1)	*4(1)	*3(1)	SEG1	
–	*8(0)	*7(0)	*6(1)	SEG2	
–	–	1	0	SEG3	Second digit
–	1	1	1	SEG4	
–	0	0	1	SEG5	

Setting procedure 3

With the 1/4 duty drive method, make the LCD display a four-digit number, "0 1 2 3 ".

Initial setting (LCD)

<Initial setting>

1. - PORT

	Register name
COM, SEG output setting for the port	See 6.8. and 6.9.

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM03

3. - Setting of control register

	registername . bit name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See chapter of "CLOCK SYSTEM" and chapter of "INTERRUPT CONTROLLER" for details.

Program 3

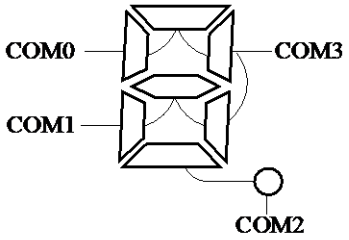
```
void LCD_sample_3(void)
{
        LCD_initial();
}

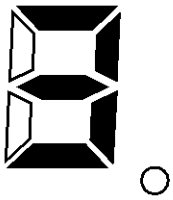
void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output. */

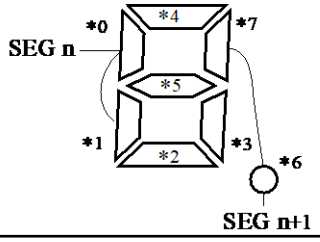
        IO_VRAM00 = 0x9F;
        IO_VRAM01 = 0x88;
        IO_VRAM02 = 0xB6;
        IO_VRAM03 = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCR0.byte = 0x0C;            /* Setting value =0000_1100 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN   Display stopped with PSS mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 11 MS[1:0]  1/4 duty mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 3 and its value (in parentheses)







COM3	COM2	COM1	COM0	
*3(1)	*2(1)	*1(0)	*0(1)	SEG0
*7(0)	*6(0)	*5(1)	*4(1)	SEG1

Setting procedure 4

With the static drive method, make the LCD display an one-digit number.

Initial setting (LCD)

<Initial setting>

1. - PORT

COM, SEG output setting for the port	Register name See 6.8. and 6.9.

2. - Data setting

Data setting	Register name LDR0 LDR1

3. - Setting of control register

Fixed value	Register name . bit name LCR1
Bias setting	LCDCMR
Setting of control register	LCRS
	. LCSEN
	. LCS[3:0]
	. FPS[1:0]

Program 4

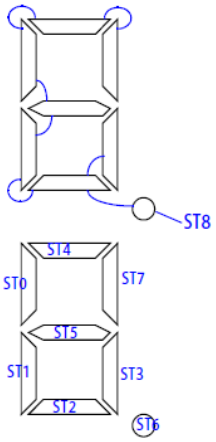
```
void LCD_sample_4(void)
{
        LCD_initial();
}

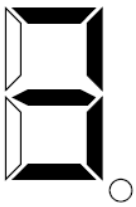
void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.   */

        IO_LDR0.byte = 0x00;
        IO_LDR1.byte = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x00;         /* 1/3 bias */
        IO_LCRS.byte = 0x20;            /* Setting value =0001_0000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCSEN   Non-display with PSS mode   */
                                        /* bit5-2 = 1000   LCS[3:0]   ST0 to ST8 */
                                        /* bit1-0 = 00 FPS[1:0]   Frame cycle setting */
}
```

Example) Values to display 3





LDR0/ LDR1 register

ST 8	ST 7	ST 6	ST 5
0	1	0	1

ST 4	ST 3	ST 2	ST 1	ST 0
1	1	1	0	0

9. Notes

This section shows notes.

- Since resistance value of the external division resisters depends on the LCD to be used, connect the resistors with a suitable value.
- Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.
- If the settings of the LCD drive power control (VSEL), duty selection (MS [1:0]), frame cycle selection (FP [1:0]), etc. are not proper, the LCD does not display correctly.
- When neither the LCD nor general-purpose ports are used, connect a pull-up or pull-down resistor to V3 to V0 pins.
- The static drive is enabled when any setting other than LCS [3:0] = "0000" is set. The duty drive is enabled when LCS [3:0] = "0000" is set.
- Setting of display/non-display of the LCD with static drive (ST0 to ST8)

When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1:ST [7:0] = "00000000", and set the static drive selection port (LCS [3:0]). The same potential pulses are output from the static drive pins (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1:ST [7:0] without change of the static drive selection port (LCS [3:0]).
When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1:ST [7:0] = "00000000" without change of the static drive selection port (LCS [3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

- When the LCD is used (static drive, duty drive), set the following.

LCR1 [7:0] = "11111111"
LCDCMR [3:0] = "1111"

In case of power shutdown, it works as following.

- When port setting is LCD output, output the "Low".
- LCD control is power shutdown, registers (LCR0, VRAMn (n=0 to 15), LCR1, LCDCMR, LCRS, LDR) of LCD control is initialized when returning from power shutdown.
- Registers (LCD_KEYCDR, SEGER, COMVER) of LCD port-related is not initialized.

CHAPTER 36: Indicator PWM

This chapter explains the indicator PWM.

1. Overview
2. Configuration and Block Diagram
3. Operation of Indicator PWM
4. Registers
5. Precautions for Using This Device

CODE: INDICATOR_PWM-S6J3200-E1

1. Overview

The indicator PWM (Pulse Width Modulation) is an output timer that consists of a 16-bit down counter, a 16-bit data register for cycle setting, a 16-bit compare register for duty setting, and a pin controller.

Compared to other output timers, it is capable of output of waveforms with relatively long cycles. Output operation is also supported while in standby mode.

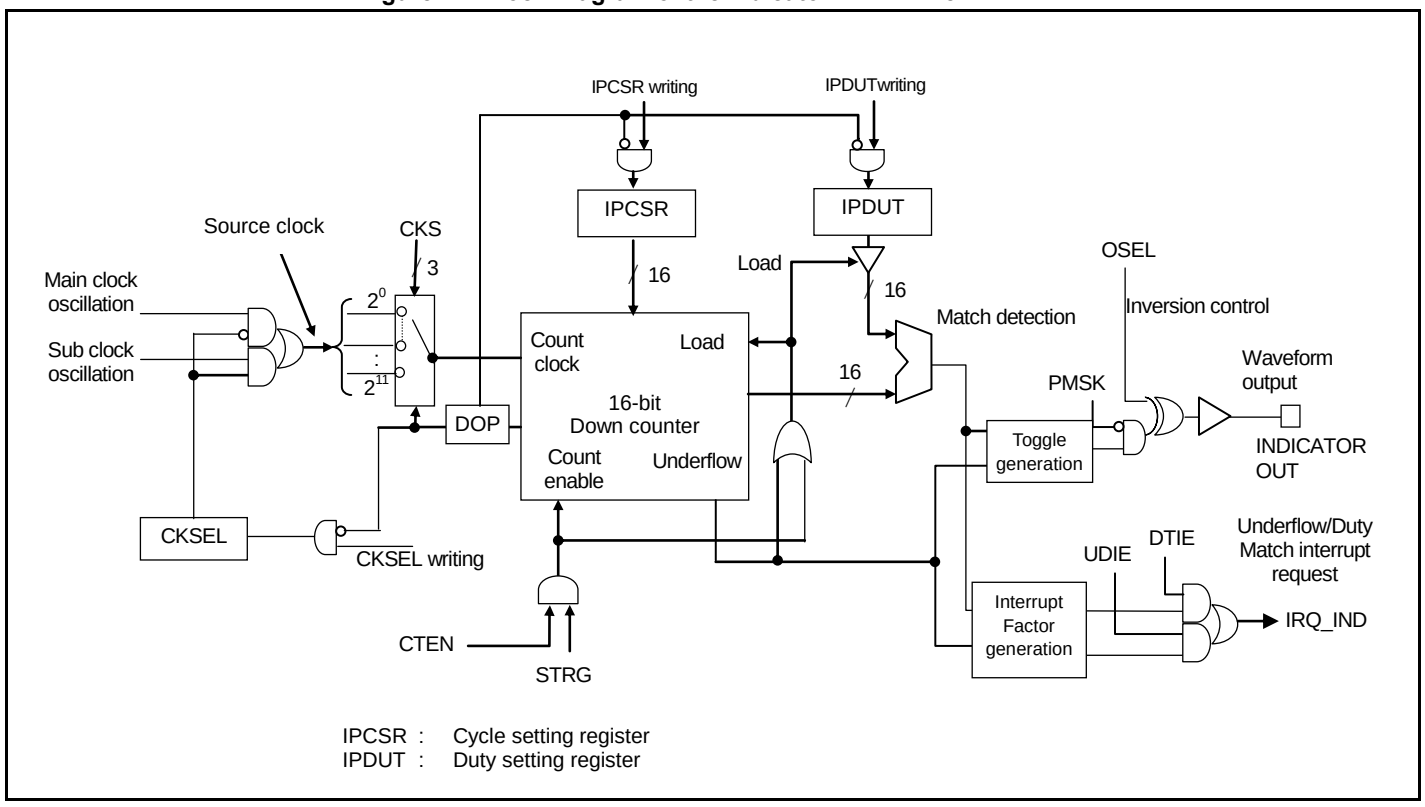
The 16-bit down counter clock can be selected from among six 6 types of clocks (divided by 1, 16, 64, 256, 1024, 2048). Source clock of these clocks can be selected either Main clock or Sub clock.

For the count operation, an underload reloads from the PWM cycle register and the count is repeated.

Indicator PWM timer activation is by software trigger only.

2. Configuration and Block Diagram

Figure 2-1 Block Diagram of the Indicator PWM Timer



Notes:

- *The clock selected by CKSEL is named as “source clock” in this chapter.*
- *The clock selected by CKS is named as “count clock” in this capter.*

3. Operation of Indicator PWM

This section explains the operations of the indicator PWM timer.

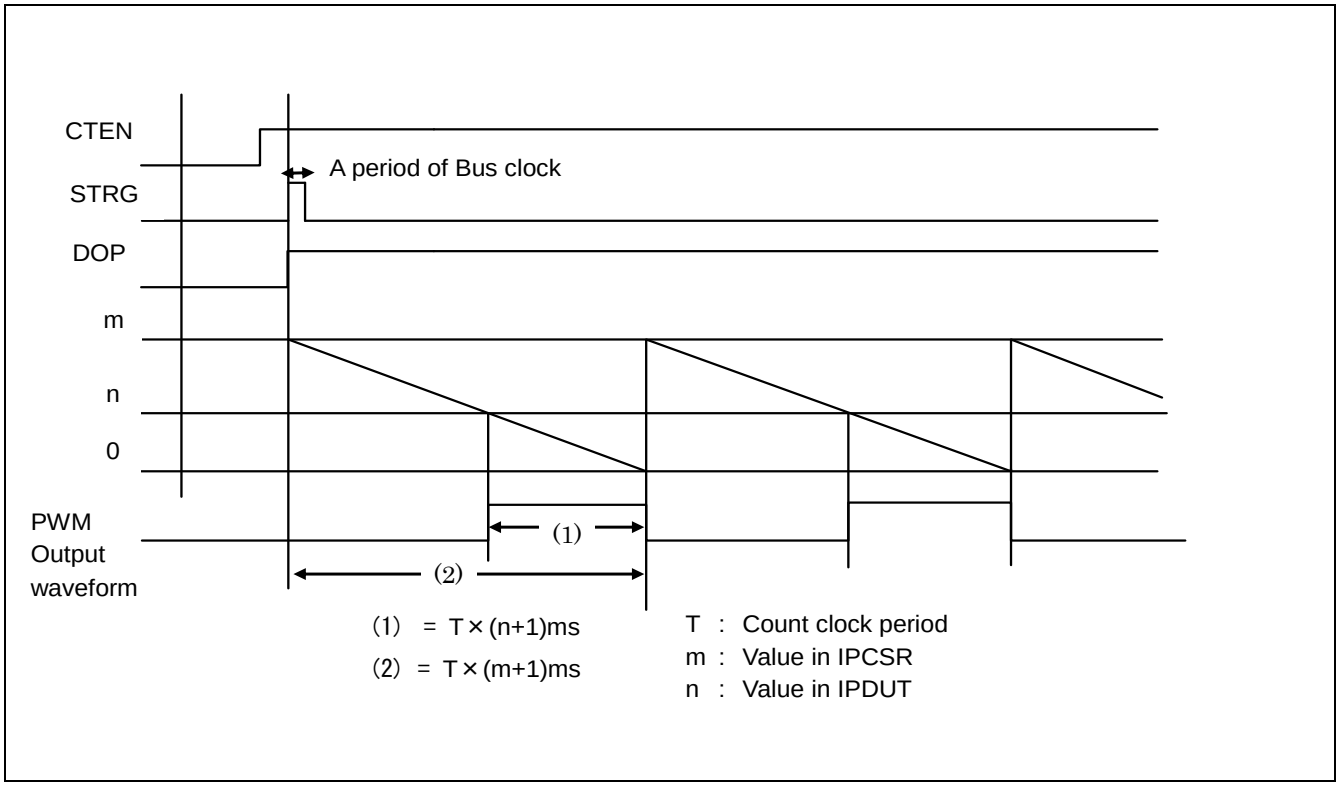
Notes:

- *The explanations in this section assume that PWM output is normal polarity. When inverse polarity is set, invert the PWM output polarity shown in this explanation.*
- *For details on how to set inverse polarity, see the OSEL bit in 4.1 Timer Control Register (ITMCR).*

3.1. Indicator PWM Timer

The indicator PWM timer starts counting down from the set cycle value upon software trigger activation. The first output at that time is the L level. If the 16-bit down counter matches the set value in the duty setting register, the output is inverted to the H level. Then, the output is inverted to the L level again when the counter underflows. Thus, this timer can generate a waveform with an arbitrary cycle and duty. The cycle of the output pulse can be controlled by configuring IPCSR register settings. The duty ratio can be controlled by configuring IPDUT register settings.

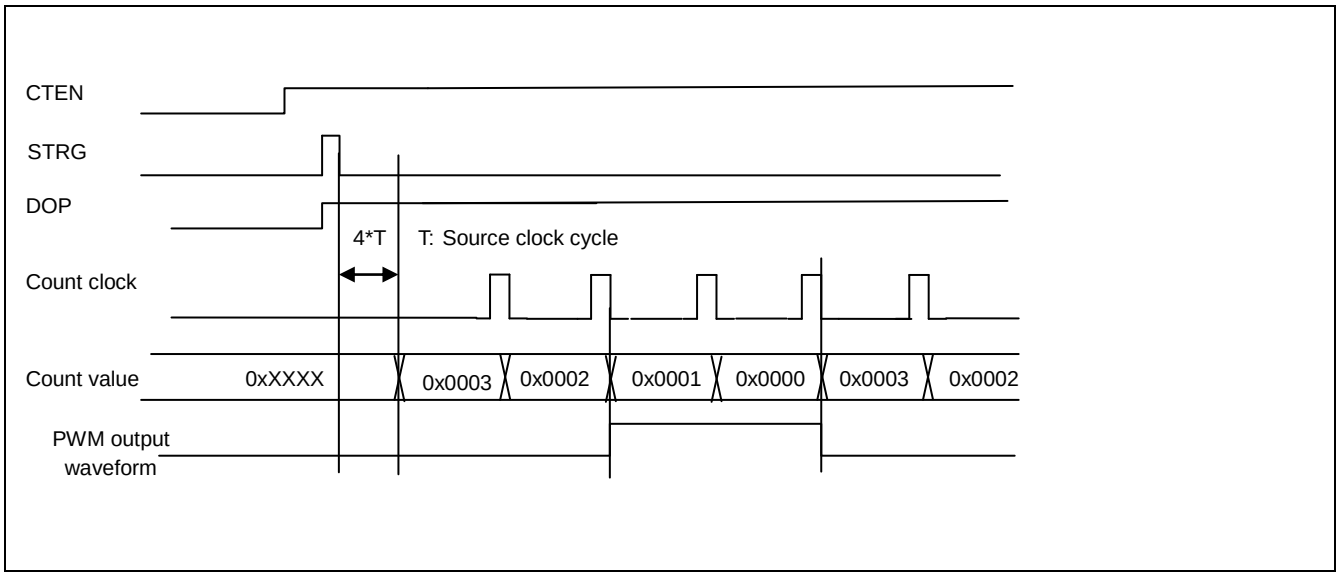
Figure 36-1 Indicator PWM Operation Timing Chart



3.2. PWM Timer Count Operation Start

Configure the PWM timer count lock select setting (ITMCR:CKS2-0), PWM cycle setting (IPCSR), and PWM duty setting (IPDUT) while the timer countdown is stopped (ICNTCR:DOP=0). After configuring settings, enable the count (ICNTCR:CTEN=1) and restart the counter operation by setting the software trigger (ICNTCR:STRG=1). As soon as the software setting is configured, the timer status becomes timer count operation in progress (ICNTCR:DOP=1).

Figure 36-2 Timing Chart When Indicator PWM Timer Count Operation Starts

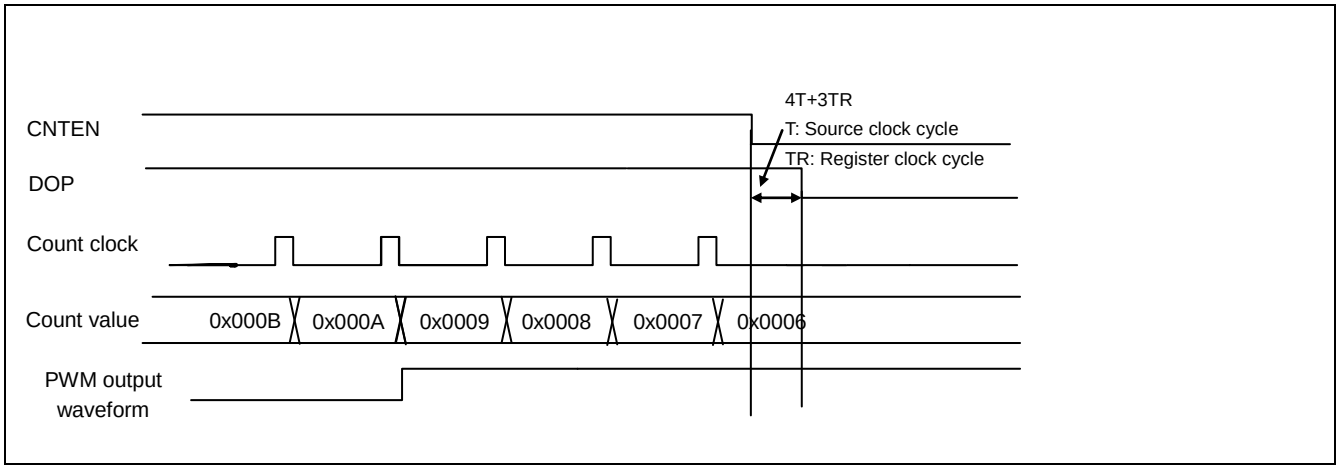


A time of 4T (T: source clock cycle) is required until the counter value is loaded after the STRG bit becomes 1. Figure 36-1 shows the timing chart for when the cycle setting value is 0x3 and the duty value is 0x1.

3.3. Timer Count Operation Stop

A timer count stop is requested by setting the timer count enable bit (ICNTCR:CTEN) to 0 while a timer count operation is in progress (ICNTCR:DOP=1). When the PWM timer stops the count, timer count operation stop (ICNTCR:DOP=0) is indicated as the timer status. PWM output maintains the output status as during timer count stop.

Figure 36-3 Timing Chart while Timer Count Operation is Stopped



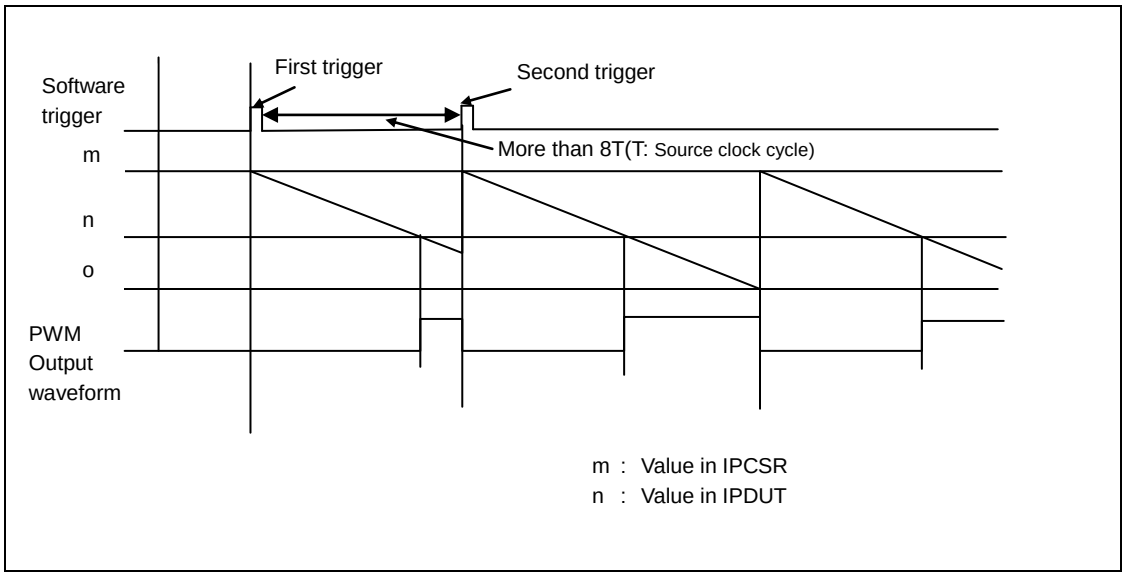
A time of 4T+3TR (T: source clock cycle, TR: register clock) is required until the counter is stopped (ICNTCR:DOP=0) after the CTEN bit becomes 0.

Figure 36-3 shows the interrupt factors and a timing chart for when the cycle setting value is 0xB and the duty value is 0x9.

3.4. Counter Value Updating while PWM Counter Operation is in Progress

Setting a software trigger (ICNTCR:STRG=1) while a PWM counter operation is in progress loads the values of the PWM cycle setting register (IPCSR) and PWM duty setting register (IPDUT) and updates the count value and duty setting. However, allow a time of at least 8T (T: source clock cycle) before setting a second software trigger. Failure to do so may result in the software trigger not being reflected.

Figure 36-4 Timing Chart for Counter Update while a Timer Count Operation is in Progress



3.5. Indicator PWM Interrupts

This section provides information about indicator PWM interrupt request flags, interrupt enable bits, and interrupt factors.

Notes:

- *The explanations in this section assume that PWM output is normal polarity. When inverse polarity is set, invert the PWM output polarity shown in this explanation.*
- *For details on how to set inverse polarity, see the OSEL bit in 4.1 Timer Control Register (ITMCR).*

3.5.1. Interrupt Control Bits and Interrupt Factors

Table 3-1 lists interrupt control bits and interrupt factors. Underflow detection and detection of duty match share the same interrupt factory output signals (IRQ_IND).

Table 3-1 Interrupt Control Bits and Interrupt Factors

	Status Control Register (ISTC)			
	Interrupt Request Flag Bit	**Interrupt Request Enable Bit**	**Interrupt Factor**	**Interrupt Factor Output Signal**
Indicator PWM	UDIR : bit0	UDIE : bit4	Detection of underflow	IRQ_IND
	DTIR : bit1	DTIE : bit5	Detection of duty match	

An underflow in the PWM counter value causes the underflow interrupt request bit (ISTC:UDIR) of the status control register (ISTC) to be set to "1". Enabling underflow interrupt request enable (ISTC:UDIE=1) while this status exists will cause interrupt request output signal IRQ_IND to become "H". In the same way, "1" is set to the duty match interrupt request bit (ISTC:DTIR) of the status control register (ISTC) when the PWM counter count value matches the duty setting register (IPDUT) value. Enabling duty interrupt request enable (ISTC:DTIE=1) while this status exists will cause interrupt request output signal IRQ_IND to become "H".

3.5.2. Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

Figure 36-5 PWM Timer Interrupt Factors and Timing Chart

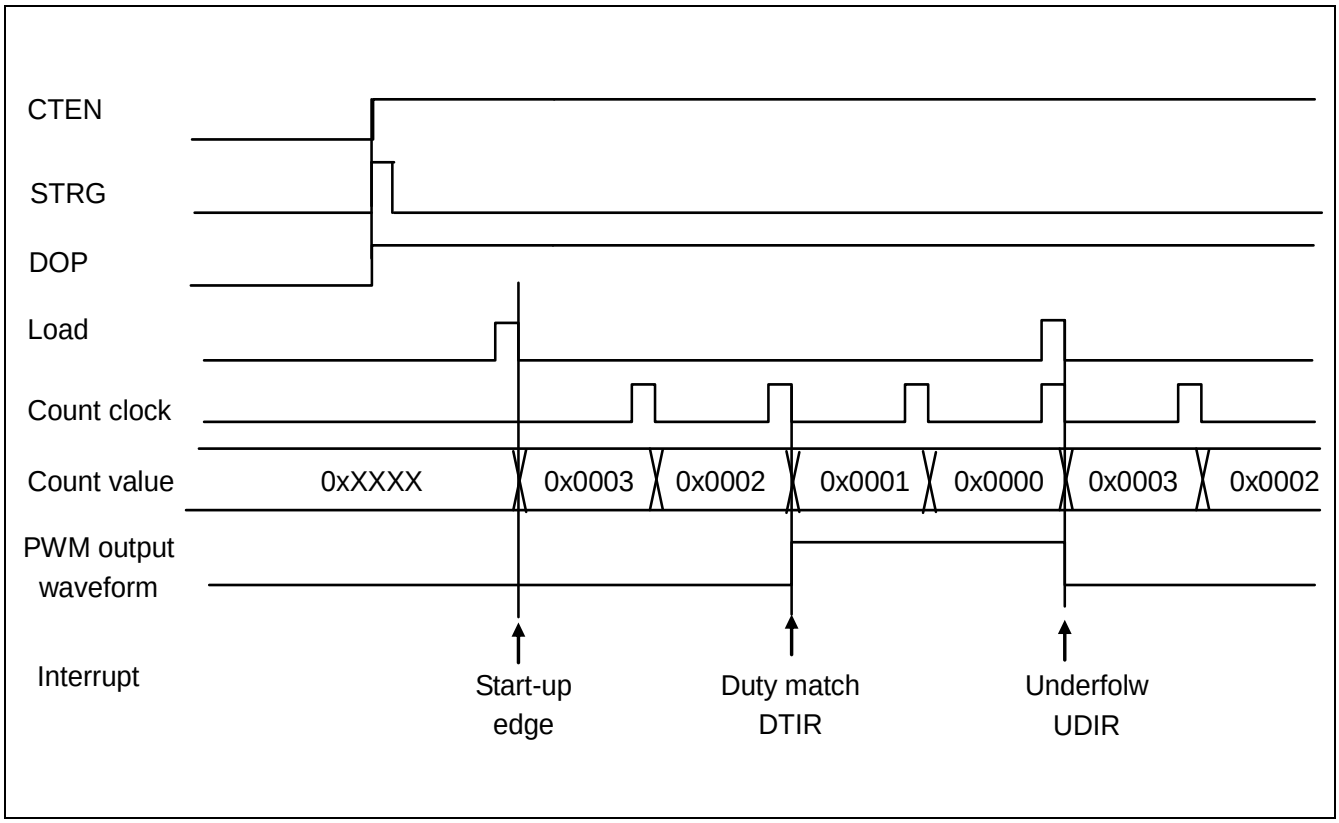


Figure 36-5 shows the interrupt factors and a timing chart for when the setting value is 0x3 and the duty value is 0x1.

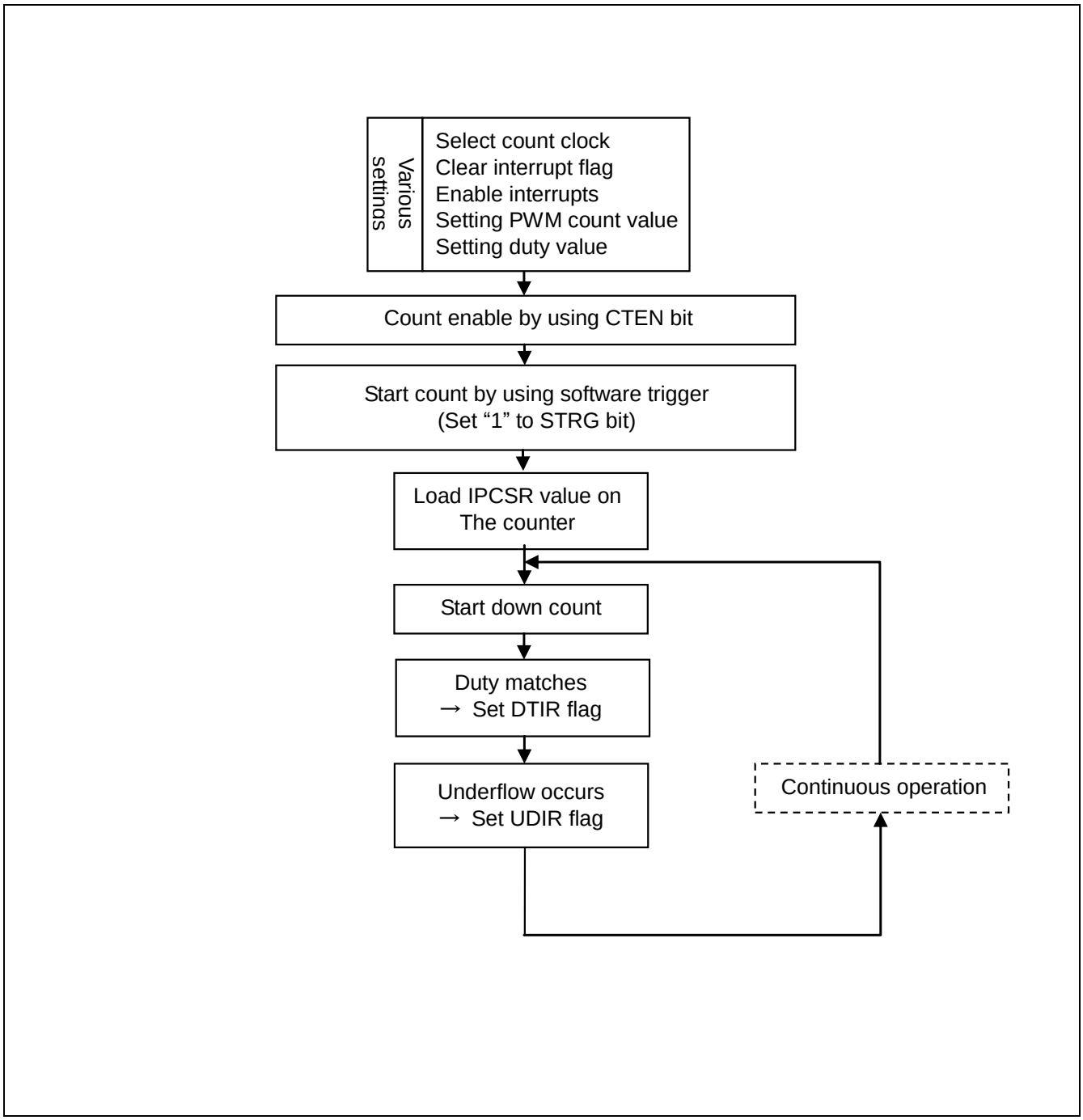
3.5.3. Return from PSS Mode

The interrupt of the indicator PWM module cannot be used as a return factor from PSS mode.

3.6. Operation Flowchart

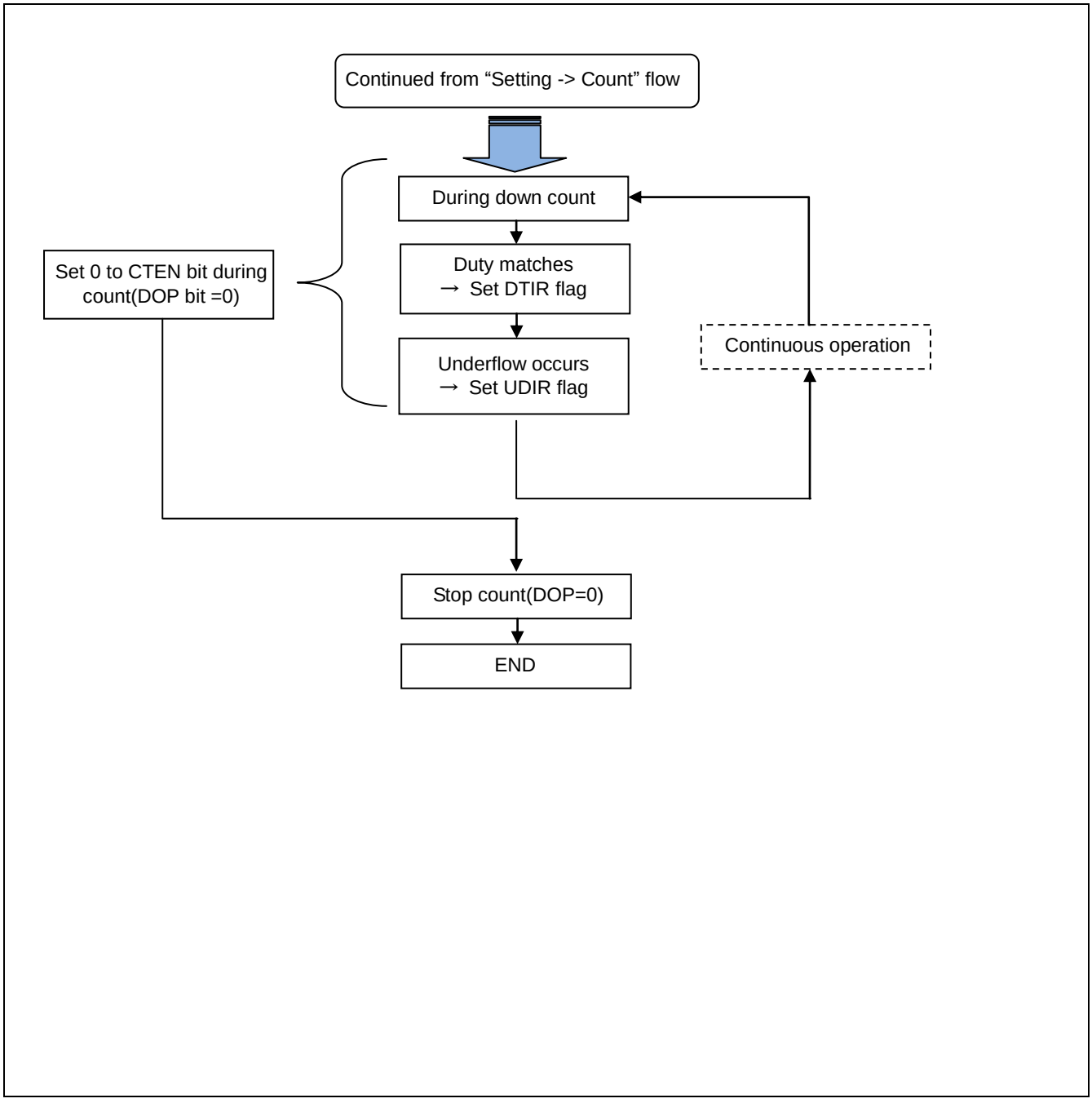
3.6.1. Operation Flowchart (Setting -> Count)

Figure 36-6



3.6.2. Operation Flowchart (Count -> Stop)

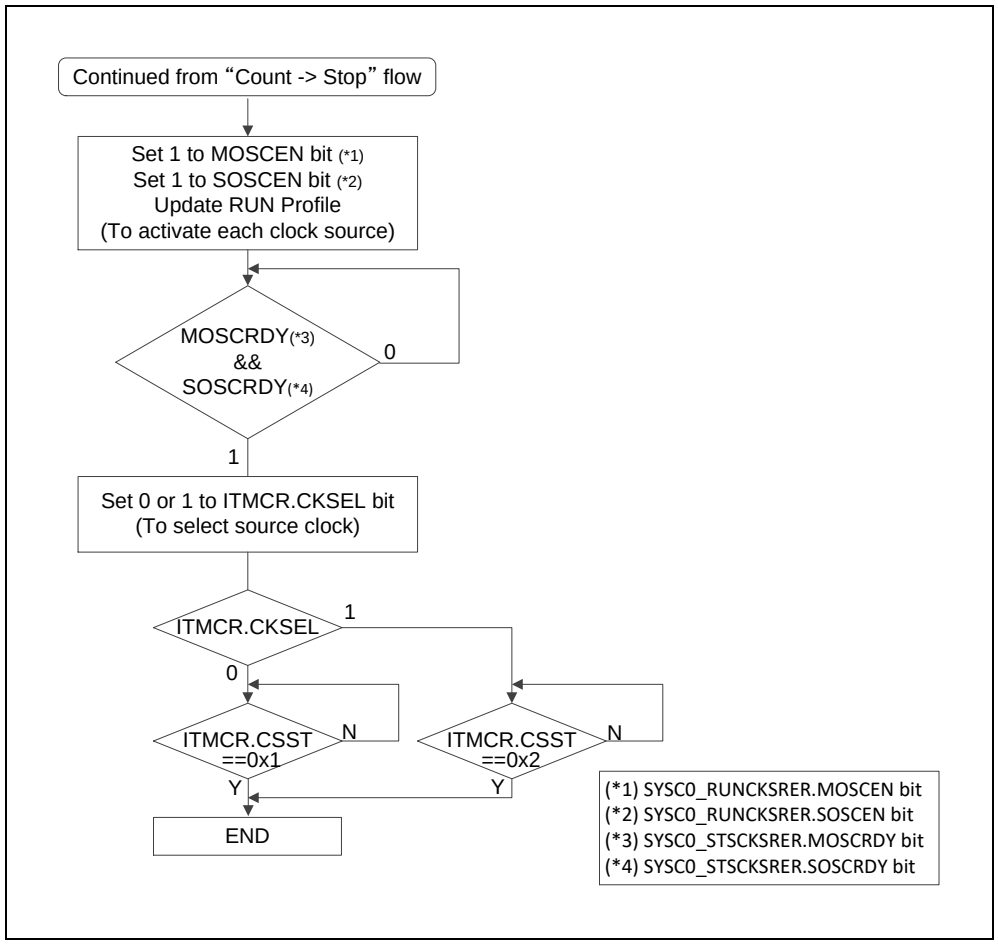
Figure 36-7



3.6.3. Switch source clock

Default source clock for indicator PWM is the main clock. Use the following configuration flow in switching the source clock for both cases from main to sub and from sub to main.

Figure 36-8



4. Registers

Table 4-1 Register List

Abbreviated Register Name	Register Name	Reference
ITMCR	Timer control register	4.1
ICNTCR	Count control register	4.2
ISTC	Status control register	4.3
ISTCC	Status control clear register	4.4
ISTCS	Status control set register	4.5
IPCSR	PWM cycle setting register	4.6
IPDUT	PWM duty setting register	4.7

4.1. Timer Control Register (ITMCR)

The timer control register (ITMCR) controls the indicator PWM timer. Note that there are bits that cannot be rewritten during timer operation (ICNTCR:DOP[bit]=1).

REGISTER_NAME	ITMCR
OFFSET	0x0C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CSST1	CSST0
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	1

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CKSEL
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	CKS2	CKS1	CKS0	Reserved	PMSK	Reserved	Reserved
ACCESS_TYPE	R0,WX	R,W	R,W	R,W	R0,WX	R/W	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	Reserved	Reserved	OSEL	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:26] Reserved

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit25:24] CSST1 to CSST0: Source clock Selection Status bits

These bits indicate status of source clock selection.

CSST1	CSST0	Description
0	0	Source clock Selection is in processing.
0	1	Main clock is selected as source clock.
1	0	Sub clock is selected as source clock.
1	1	N/A

[bit23:17] Reserved.

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit16] CKSEL: Source Clock selection bit

This bit selects either Main clock or Sub clock as source clock.

Stop the timer (ICNTCR:DOP[bit]=0) before changing the source clock

Bit	Description
0	Select Main clock
1	Select Sub clock

[bit15] Reserved.

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit14:12] CKS2 to CKS0: Count clock selection bits

These bits select a count clock for the 16-bit down counter.

Stop the timer (ICNTCR:DOP[bit]=0) before changing the count clock.

CKS2	CKS1	CKS0	Description
0	0	0	Source clock
0	0	1	Source clock divided by 16
0	1	0	Source clock divided by 64
0	1	1	Source clock divided by 256
1	0	0	Source clock divided by 1024
1	0	1	Source clock divided by 2048
1	1	0	Source clock
1	1	1	Source clock

[bit11] Reserved

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit10] PMSK: Pulse output mask bit

This bit controls the output waveform level of the PWM output waveform.

When the bit is "0", the PWM waveform is output as is.

When the bit is "1", PWM output is masked to L output regardless of the cycle or duty setting value.

Note:

- *If the output polarity specification bit (OSEL) in the timer control register (lower byte of ITMCR) is set for inverted output, setting the PMSK bit to "1" results in masking to H output.*

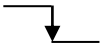
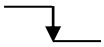
Bit	Description
0	Normal output
1	Fixed at L output

[bit9:4] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit3] OSEL: Output polarity specification bit

This bit sets the PWM output polarity.

Polarity	After Reset	Duty Match	Underflow
Normal	"L" output		
Inverse	"H" output		

Bit	Description
0	Normal polarity
1	Inverse polarity

[bit2:0] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

4.2. Count Control Register (ICNTCR)

The count control register(ICNTCR) is PWM timer count control register.

REGISTER_NAME	ICNTCR
OFFSET	0x10
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	DOP	Reserved	Reserved	Reserved	Reserved	Reserved	CTEN	STRG
ACCESS_TYPE	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:8] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit7] DOP: Count status bit

Setting the STRG bit to "1" while the CTEN bit is "1" will immediately make this bit "1".

The read value for this bit is "1" while a PWM timer count is in progress.

The read value for this bit is "0" while the PWM timer is stopped.

The read value for this bit is "0" while the PWM timer count is stopped after the CTEN bit is set to "0".

Set this bit to "0" before selecting a timer clock (ITMCR:CKS2-0), or configuring PWM cycle (IPCSR) or PWM duty (IPDUT) settings. These settings are ignored if they are configured while this bit is "1".

Bit	Description
0	Timer count stopped
1	Timer count operation in progress

[bit6:2] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit1] CTEN: Count operation enable bit

This bit enables operation of the down counter.

If "0" is written to this bit while a counter operation is in progress (DOP bit is "1"), a stop request will be output for the counter.

Counter stop can be checked by confirming that the counter status bit (DOP bit) is "0".

After CTEN bit is set to 0, if you re-set CTEN bit to "1", you must check DOP bit is "0".

Bit	Description
0	Request stop
1	Enable operation.

[bit0] STRG: Software trigger bit

If "1" is written to the STRG bit when the CTEN bit is "1", a software trigger is applied.

The read value of the STRG bit is always "0".

Notes:

- *Even if "1" is written to the CTEN and STRG bits at the same time, a software trigger is applied.*
- *When consecutively setting STRG bits, allow a time of at least 8T (T: source clock cycle) between settings. Failure to do so may result in a setting not being reflected.*

Bit	Description
0	Disabled
1	Startup by software

4.3. Status Control Register (ISTC)

For details on writing to the status control register (ISTC), see "5.1 Notes to observe when accessing a register".

REGISTER_NAME	ISTC
OFFSET	0x14
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	DTIE	UDIE	Reserved	Reserved	DTIR	UDIR
ACCESS_TYPE	R0,WX	R0,WX	R,W	R,W	R0,WX	R0,WX	R,WX	R,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:6] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit5] DTIE: Duty match interrupt request enable bit

This bit controls interrupt requests of the duty match interrupt request bit (bit1 DTIR).

If the DTIE bit is enabled and the DTIR bit is set to "1", an interrupt request is issued to the CPU.

Writing "1" to the ISTCC:DTIEC bit clears this bit.

Writing "1" to the ISTCS:DTIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4] UDIE: Underflow interrupt request enable bit

This bit controls interrupt requests of the underflow interrupt request bit (bit0 UDIR).

If the UDIE bit is enabled and the UDIR bit is set to "1", an interrupt request is issued to the CPU.

Writing "1" to the ISTCC:UDIEC bit clears this bit.

Writing "1" to the ISTCS:UDIES bit sets this bit.

Bit	Description
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit3:2] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit1] DTIR: Duty match interrupt request bit

The DTIR bit is set to "1" when the count value matches the duty setting value.

Writing "1" to the ISTCC:DTIRC bit clears this bit.

This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

[bit0] UDIR: Underflow interrupt request bit

The UDIR bit is set to "1" when the count value underflows.

Writing "1" to the ISTCC:UDIRC bit clears this bit.

This bit is read-only. Writing data to this bit has no effect on operation.

Bit	Description
0	The interrupt factor is cleared.
1	Detect the interrupt factor.

4.4. Status Control Clear Register (ISTCC)

The status control clear register (ISTCC) is a register that is used to clear bits of the status control register (ISTCC).

REGISTER_NAME	ISTCC
OFFSET	0x18
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	DTIEC	UDIEC	Reserved	Reserved	DTIRC	UDIRC
ACCESS_TYPE	R0,WX	R0,WX	R0,W	R0,W	R0,WX	R0,WX	R0,W	R0,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:6] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit5] DTIEC: Duty match interrupt request enable clear bit

If "1" is written to this bit, the ISTC:DTIE bit is cleared to "0".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Clear the DTIE bit.

[bit4] UDIEC: Underflow interrupt request enable clear bit

If "1" is written to this bit, the ISTC:UDIE bit is cleared to "0".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Clear the UDIE bit.

[bit3:2] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit1] DTIRC: Duty match interrupt request clear bit

If "1" is written to this bit, the ISTC:DTIR bit is cleared to "0".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Clear the DTIR bit.

[bit0] UDIRC: Underflow interrupt request clear bit

If "1" is written to this bit, the ISTC:UDIR bit is cleared to "0".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Clear the UDIR bit.

4.5. Status Control Set Register (ISTCS)

The status control set register (ISTCS) is a register that is used to set bits of the status control register (ISTC).

REGISTER_NAME	ISTCS
OFFSET	0x1C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	Reserved	Reserved	DTIES	UDIES	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,W	R0,W	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:6] Reserved

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit5] DTIES: Duty match interrupt request enable set bit

If "1" is written to this bit, the ISTC:DTIE bit is set to "1".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Set the DTIE bit.

[bit4] UDIES: Underflow interrupt request enable set bit

If "1" is written to this bit, the ISTC:UDIE bit is set to "1".

"0" is always read from this bit.

Bit	Description
0	Disabled
1	Set the UDIE bit.

[bit3:0] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

4.6. PWM Cycle Setting Register (IPCSR)

The PWM cycle setting register (IPCSR) is a register with a buffer for setting a cycle. Transfer to the timer is performed upon software trigger activation. Note that rewriting is not supported while a timer count operation is in progress (ICNTCR:DOP[bit]=1).

REGISTER_NAME	IPCSR
OFFSET	0x00
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	IPCSR[15]	IPCSR[14]	IPCSR[13]	IPCSR[12]	IPCSR[11]	IPCSR[10]	IPCSR[9]	IPCSR[8]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	IPCSR[7]	IPCSR[6]	IPCSR[5]	IPCSR[4]	IPCSR[3]	IPCSR[2]	IPCSR[1]	IPCSR[0]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] Reserved

These are reserved bits. Writing data to these bits has no effect on operation.

[bit15:0] IPCSR: PWM cycle setting register

This register is for setting the PWM timer cycle. Transfer to the timer is performed upon software trigger activation.

Stop the timer (ICNTCR:DOP[bit]=0) before configuring the PWM cycle setting register (IPCSR) setting. Settings are ignored if they are configured while timer operation is in progress (ICNTCR:DOP[bit]=1).

4.7. PWM Duty Setting Register (IPDUT)

The PWM duty setting register (IPDUT) is a register for setting a duty. Transfer is performed upon software trigger activation. Note that rewriting is not supported while a timer count operation is in progress (ICNTCR:DOP[bit]=1).

REGISTER_NAME	IPDUT
OFFSET	0x04
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

BIT_OFFSET	31	30	29	28	27	26	25	24
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	23	22	21	20	19	18	17	16
BIT_NAME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
ACCESS_TYPE	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	15	14	13	12	11	10	9	8
BIT_NAME	IPDUT[15]	IPDUT[14]	IPDUT[13]	IPDUT[12]	IPDUT[11]	IPDUT[10]	IPDUT[9]	IPDUT[8]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

BIT_OFFSET	7	6	5	4	3	2	1	0
BIT_NAME	IPDUT[7]	IPDUT[6]	IPDUT[5]	IPDUT[4]	IPDUT[3]	IPDUT[2]	IPDUT[1]	IPDUT[0]
ACCESS_TYPE	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W
PROT_TYPE	-							
INITIAL_VALUE	0	0	0	0	0	0	0	0

[bit31:16] Reserved

This is a reserved bit.

Writing data to these bits has no effect on operation.

[bit15:0] IPDUT: PWM Duty Setting Register

This is the register for setting the duty. Transfer is performed upon software trigger activation.

If the set values of the cycle setting register and duty setting register are the same value, the output for normal polarity is all "H", and the output for inverse polarity is all "L".

If IPCSR < IPDUT in the set values, the output for normal polarity is all "L", and the output for inverse polarity is all "H".

Stop the timer (ICNTCR:DOP[bit]=0) before configuring PWM duty setting register (IPDUT) settings. Settings are ignored if they are configured while timer operation is in progress (ICNTCR:DOP[bit]=1).

5. Precautions for Using This Device

This section explains precautions during use.

5.1. Notes to Observe When Accessing a Register

■ Status control register (ISTC) access

To clear a specific bit in this register, write "1" to the corresponding bit in the status control clear register (ISTCC).

To set a specific bit in this register, write "1" to the corresponding bit in the status control set register (ISTCS).

Data can be written directly to this register only when writing to all bits.

5.2. Indicator PWM Operation Precautions

■ Precautions when configuring settings using a program

Timer control register (ITMCR) CKS2-0 bits, CKSEL bit, PWM cycle setting register (IPCSR), and PWM duty setting register (IPDUT) settings are not reflected if they are configured while a count operation is in progress (ICNTCR:DOP=1). Be sure to stop the count (ICNTCR:DOP=0) before configuring these settings.

If the interrupt request flag set timing and clear timing overlap, the flag set has priority, and the clear operation is disabled.

If the load timing and count timing overlap, the load operation has priority for the down counter.

When consecutively setting STRG bits, allow a time of at least 8T (T: source clock cycle) between settings. Failure to do so may result in a setting not being reflected.

Initial value of CKSEL bit in the Timer control register (ITMCR) is “Select Main clock”. Software compatibility is guaranteed because only Main clock is supported as the source clock of the former products.

Clock switching flow which is described in section 3.6.3 shall be preserved. If the clock selection flow is not preserved, PWM output is not guaranteed with register configurations.

Recommendation:

The same clock source should be configured as PWM clock (Source Clock) and RTC clock.

Note that the PWM clock cannot be configured slow RC however.

- ITMCR:CKSEL: Source Clock selection bit for Indicator PWM
- RTC_WTCR:RCKSEL: Clock select for RTC

The above-mentioned purpose is to stop the oscillation of unused clock source, especially in PSS timer mode.

■ Note that re-setting STRG bit during timer operation.

During PWM timer operation, the re-setting of the software trigger (ICNTCR:STRG="1") allows to update the counter value. However, as a result of different between register clock speed and counter clock or source clock speed, the count value of the counter at last time would reach the register value of IPDUT or IPCSR before the re-setting of the software trigger is reflected in the counter. In this case, Note that there is a possibility that the Duty match interrupt and Underflow interrupt is generated.

CHAPTER 37: Memory Protection Unit for AXI

This chapter explains the functions and operations of the Memory Protection Unit for AMBA AXI protocol bus (MPU AXI).

1. Overview
2. Configuration and Block Diagram
3. Operation of the MPU AXI
4. Registers
5. Notes on Using MPU AXI

CODE: MPUAXI-S6J3200-E1

1. Overview

This section describes the features of the MPU AXI.

■ Features of the MPU AXI

The MPU AXI module monitors the accesses from AXI masters and checks each access against an authorized set of access permissions. Access permissions ("permission attributes" here onwards) are defined by "access permissions" bits. These bits are explained in Section "MPU access permissions". MPU AXI provides eight regions and one background region. Each region has corresponding access permission bits that defines the permission attributes for that particular region. Any unauthorized access to memory space is flagged using Non-Maskable Interrupt. MPU AXI also collects information about the unauthorized bus access and stores it in its internal registers.

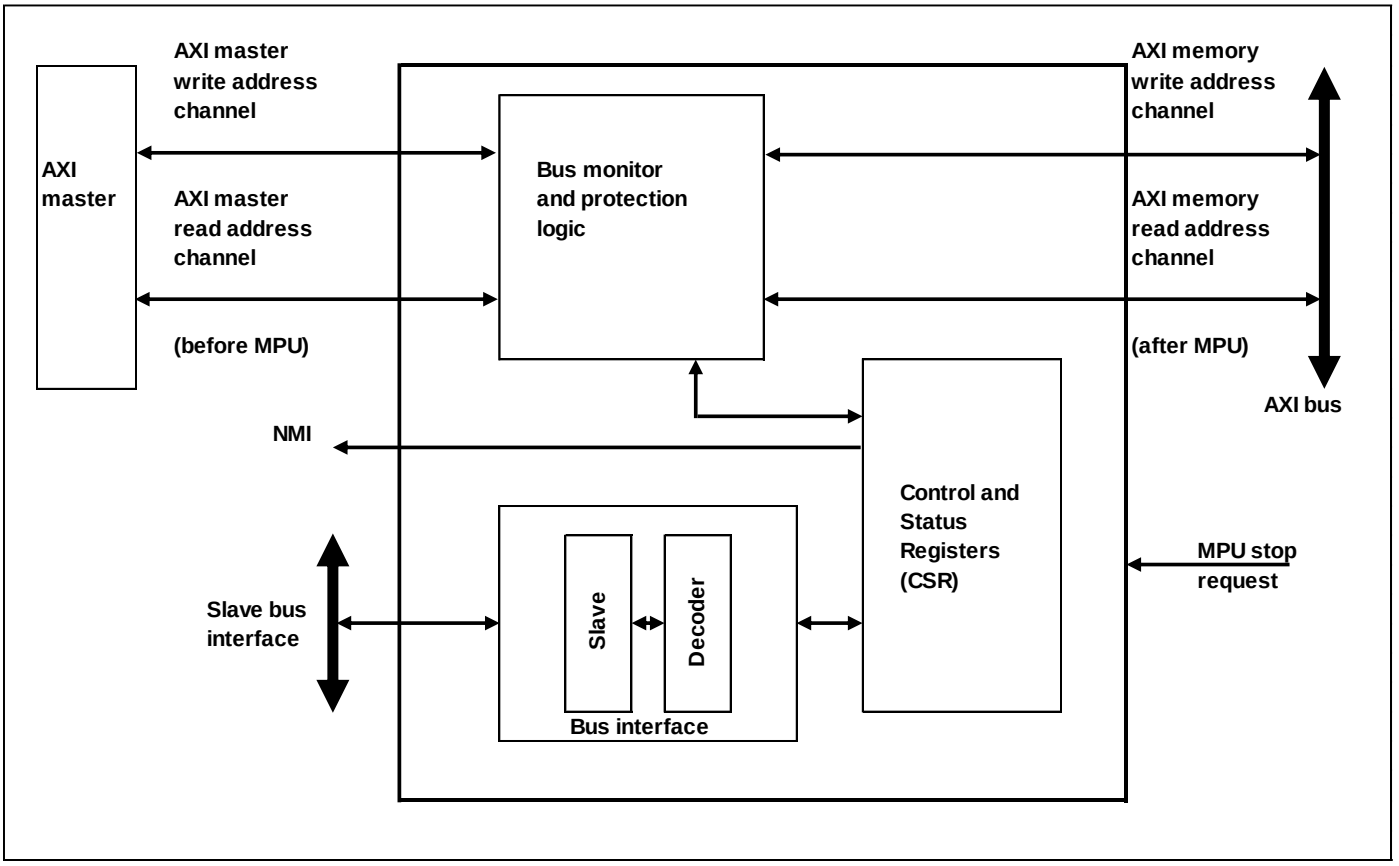
- Each of the eight regions in MPU AXI is specified using corresponding start address and end address
- Background region covers entire 4 GB address space
- On unauthorized access MPU AXI generates an NMI to the CPU
- MPU AXI collects information about the AXI master bus access that caused unauthorized access
- Supports 8-bit, 16-bit, and 32-bit bus accesses for configuration of registers in MPU AXI
- Supports lock, unlock feature for protection of registers from illegal write accesses
- Modification of registers in MPU AXI is only allowed in privileged mode
- Supports MPU stop feature that allows blocking of all the accesses to memory space
- Optionally supports privileged mode overwrite feature that allows overwrite of privilege attribute of memory side AXI interface

2. Configuration and Block Diagram

This section shows a block diagram of MPU AXI.

■ Block Diagram of MPU AXI

Figure 2-1 Block Diagram of MPU AXI



■ Bus Interface

The bus masters can access the MPU AXI module through its slave bus interface.

■ Bus Monitor and Protection Logic

This logic monitors the transaction on AXI master interfaces. It finds out the region/s where the current transfer belongs to and then applies permissions based on the region match. It signals any permission violation to CSR logic that in turn generates NMI interrupt. All transactions on AXI master interfaces (including the transfer that caused permission violation) are blocked until NMI is cleared.

■ Control and Status Registers

The operation of MPU AXI can be controlled and monitored through its Control and Status Registers (CSR)

3. Operation of the MPU AXI

This section describes the operation of MPU AXI.

MPU AXI provides start address and end address for each of the eight regions. MPU AXI regions are defined with granularity of 128 bytes.

Start address specifies the first address of the region and is specified by registers MPUXn_SADDR1 to MPUXn_SADDR8 for region 1 to region 8 respectively. Since the region granularity is 128 bytes least significant 7 bits of start addresses will read 0.

End addresses are specified by registers MPUXn_EADDR1 to MPUXn_EADDR8 for region 1 to region 8 respectively. Least significant 7 bits of End Address Registers are read-only bits and will always read "1". This ensures the granularity of 128 bytes.

■ AXI Burst Monitoring

Bus monitor and protection logic monitors AXI master "write address channel" signals and "read address channel" signals ("AXI master interfaces" here onwards). The transactions on AXI master interfaces are manipulated (if required) before they are passed on to AXI memory "write address channel" signals and "read address channel" signals ("AXI memory interfaces" here onwards).

AXI protocol supports following features

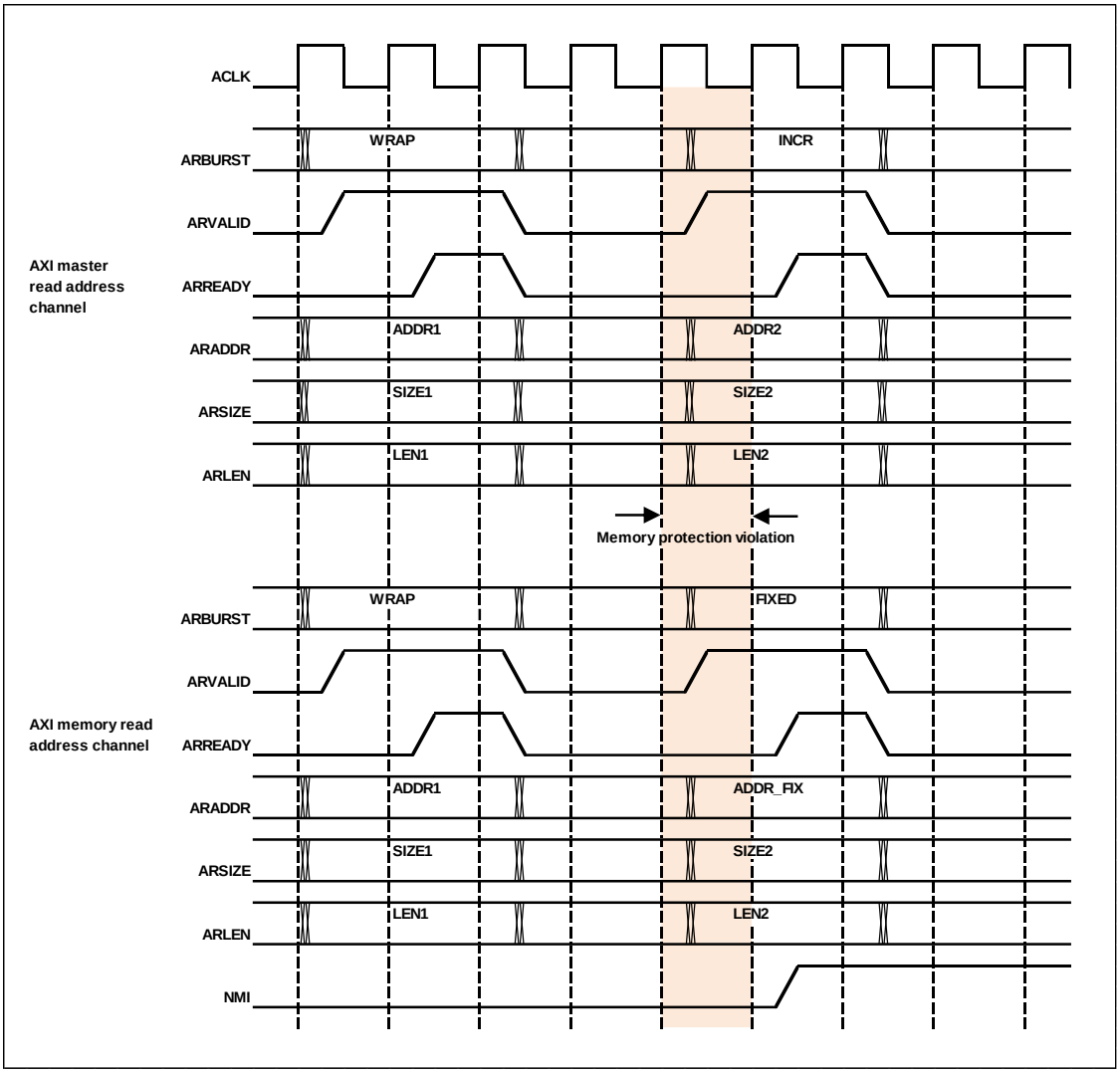
- Separate address or control and data phases
- Burst based transaction where only start address is issued
- Separate write and read address/control channels
- Separate write and read data channels

AXI master begins each burst by driving transaction control information and address of the first byte in the transaction. As the burst transaction progresses, AXI slave calculates the addresses of subsequent transfers in the burst. The AWLEN (write address channel signal), ARLEN (read address channel signal) specifies the number of data transfers for a burst transaction. Each burst can be of 1 to 16 transfers long. The AWSIZE, ARSIZE signals specifies the maximum number of data transfers in terms of bytes in each data transfer within a burst. The AXI protocol defines FIXED, incrementing and wrapping burst types.

As the addresses are defined in separate channels than data channels MPU AXI monitors and controls only address channels of AXI. If memory protection violation is detected entire burst transaction is manipulated to FIXED address burst with address of predefined value.

Figure 3-1 shows timings for read address channel signals. First burst is of WRAP type for which no memory protection violation was detected. Second burst is of INCR type for which memory protection violation was detected and hence MPU AXI manipulates the ARBURST to FIXED type and ARADDR to predefined value (shown as ADDR_FIX).

Figure 3-1 MPU AXI Example Timings



MPU AXI uses burst start address (AWADDR, ARADDR signals), burst type (AWBURST, ARBURST signals), burst length (AWLEN, ARLEN signals), and burst size (AWSIZE, ARSIZE signals) from AXI burst transaction to calculate the lowest address and the highest address of the burst and then both the addresses are used to find the region match.

For FIXED type of burst:

Lowest address = Highest address = Burst start address

For INCR type of burst:

Lowest address = Burst start address

Highest address = Burst start address + (num_bytes x num_transfers) – 1

where, num_bytes is number of bytes in each transfer of burst and

num_transfers is number of transfers in the burst

For WRAP type of burst:

Lowest address = Wrap boundary

Highest address = Wrap boundary + (num_bytes x num_transfers) – 1

■ MPU Access Permissions

Region control registers in MPU AXI, MPUXn_CTRL1 to MPUXn_CTRL8 are used to control the access permission for region 1 to region 8 respectively. Also MPUXn_CTRL0 is used to control the access permission for background region.

Table 3-1 Access Permissions

AP Bits	Access in Privileged Mode	Access in Non- Privileged Mode	Comment
000 (default)	No access	No access	All bus accesses are blocked and hence would generate memory protection violation.
001	read, write	No access	Reads and writes are permitted in privileged mode only. Any access in non- privileged mode would generate memory protection violation.
010	read, write	read only	Reads and writes are permitted in privileged mode. Only reads are permitted in non-privileged mode. Writes in non- privileged mode would generate memory protection violation.
011	read, write	read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.
100	No access	No access	All bus accesses are blocked and hence generate memory protection violation.
101	read only	No access	Reads are permitted in privileged mode only. Writes in privileged mode and any access in non-privileged mode would generate memory protection violation.
110	read only	read only	Reads are permitted in privileged as well as non-privileged mode. Any write access would generate memory protection violation.
111	read, write	read, write	All bus transfers are permitted. No memory protection violation is generated in this mode.

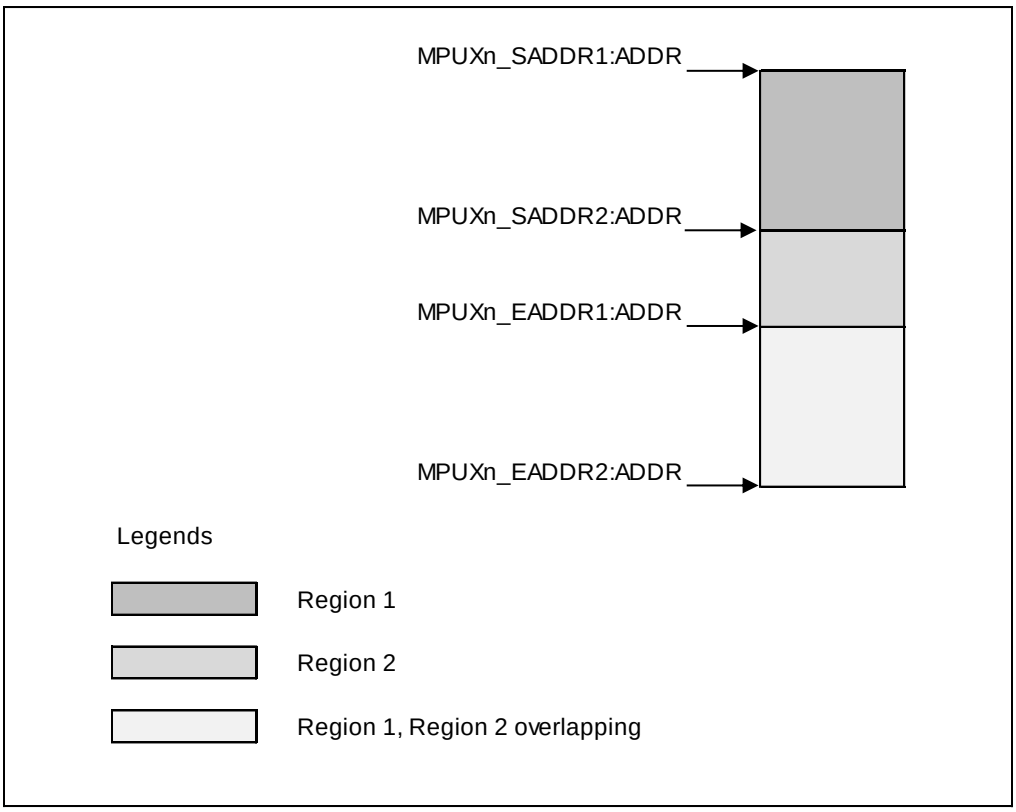
■ MPU Priority Decision

MPU AXI provides start address and end address for each region. Start address specifies the first address of the region and end address specifies the last address of the region. It is allowed in MPU AXI that the region may overlap each other. Hence it is also possible that an AXI transaction address may match with multiple regions.

For AXI protocol transaction following scenarios are possible.

1. Lowest address and highest address might fall in the same region.
2. Lowest address might match in one region, but highest address might match in some other region.
3. Scenario mentioned in point 1 here, happens for multiple regions due to region overlapping.
4. Scenario mentioned in point 2 here, happens for multiple regions due to region overlapping.

Figure 3-2 Example Region Overlapping



MPU AXI finds region match signals based on lowest address of AXI burst transaction and also separate region match signals based on highest address. Due to overlapping of regions it is possible lowest address and/or highest address may match with multiple regions.

MPU AXI supports upto eight regions. Each region is identified as region 1, region 2, region 3, and so on upto region 8. Region 8 is given highest priority, region 7 has second highest priority, and so on with region 1 as the second lowest priority region. Background region is the lowest priority region.

Whenever the AXI transaction lowest address matches with multiple regions the permissions corresponding to the highest priority (among all matching region) region is applied. Similarly, if AXI transaction highest address matches with multiple regions the permissions corresponding to highest priority (among all matching region) region is applied.

It is possible that lowest address and highest address may match with different highest priority regions. In this case the more restrictive permission is applied. Table 3-2 shows access permissions with corresponding restriction index for privileged mode access. Restriction index 1 has the most restrictive access and access permissions with restriction index 3 has the least restrictive access. This table is applied when privileged mode attribute of current transaction is of type privileged mode.

Similarly, Table 3-3 shows access permissions with corresponding restriction index for non- privileged mode access. This table is applied when privileged mode attribute of current transaction is of type non-privileged mode.

Table 3-2 Restrictive Access Permission Matrix for Privileged Mode Access

AP Bits	Access in Privileged Mode	Restriction Index
"000", "100"	No access	1
"101", "110"	read only	2
"001", "010", "011", "111"	read, write	3

Table 3-3 Restrictive Access Permission Matrix for Non-privileged Mode Access

AP Bits	Access in Non-privileged Mode	Restriction Index
"000", "001", "100", "101"	No access	1
"010", "110"	read only	2
"011", "111"	read, write	3

■ Bus Monitor and Protection Logic

All transactions on the AXI master interfaces are monitored and checked for permitted access.

- Bus monitor and protection logic within MPU AXI compares the lowest address and highest address of current transaction with start addresses and end addresses of each region to find region match where the current transaction matches to among the eight defined regions
- As explained in Section MPU priority decision the AXI transaction address may match with multiple regions. The permission attributes of highest priority region are checked against the attributes of currently applied transaction from AXI master
- If the attributes of currently applied transaction are within permitted attributes, current transaction is passed on to the AXI memory interfaces
- If the attributes are not within permitted attributes current transaction is blocked. The Non Maskable Interrupt (MPUXn_CTRL0:NMI) flag is set. If the memory protection violation is detected on write address channel, the address and control information is stored in MPUXn_WERRA and MPUXn_WERRC registers respectively. If the memory protection violation is detected on read address channel, the address and control information is stored in MPUXn_RERRA and MPUXn_RERRC registers respectively. It is possible that memory protection violation is detected simultaneously on both the channels
- All further transaction are blocked until the MPUXn_CTRL0:NMI flag is cleared by software. Further monitoring of AXI master interfaces is also stalled until the MPUXn_CTRL0:NMI flag is cleared

When a transfer is blocked, MPU AXI does the following actions:

- Current transaction on AXI master is manipulated to FIXED address burst

- The transaction address is changed to predefined address. Check device specific datasheet for details of predefined address

■ MPU Stop Feature

Optionally MPU AXI supports MPU stop feature.

When this mode is enabled all accesses to memory space are blocked and MPU AXI does the following actions:

- Burst type signal is driven to FIXED type burst
- Burst address is driven to predefined FIXED address

■ Privileged Mode Overwrite Feature

Optionally MPU AXI supports privileged mode overwrite feature.

When this mode is enabled privileged mode attribute on the AXI memory interfaces are set by register bit setting MPUXn_CTRL0:PROT.

Note:

- *Bus monitor and protection logic for detecting memory protection violation uses privilege mode attribute on AXI master interfaces and not MPUXn_CTRL0:PROT bit setting even when privileged mode overwrite feature is enabled.*

4. Registers

This section describes the registers of MPU AXI

The MPU AXI module contains various registers to configure its operation, to monitor its status and to read the information it has collected from the AXI master interfaces at the time of the memory protection violation.

The MPU AXI module is allocated 1 KB of MCU address space for mapping the Configuration and Status Registers (i.e. CSRs). The address area allocated to MPU AXI and the Control and Status Registers in MPU AXI are explained in this section.

The suffix 'n' in the register name indicates that the register is an instance 'n' of the module.

■ Registers of MPU AXI

The following registers are available for MPU AXI:

- MPU AXI Control Register (MPUXn_CTRL0)
- MPU AXI NMI Enable Register (MPUXn_NMIEN)
- MPU AXI Write Error Control Register (MPUXn_WERRC)
- MPU AXI Write Error Address Register (MPUXn_WERRA)
- MPU AXI Read Error Control Register (MPUXn_RERRC)
- MPU AXI Read Error Address Register (MPUXn_RERRA)
- MPU AXI Region Control Registers (MPUXn_CTRL1 to 8)
- MPU AXI Start Address Registers (MPUXn_SADDR1 to 8)
- MPU AXI End Address Registers (MPUXn_EADDR1 to 8)
- MPU AXI Unlock Register (MPUXn_UNLOCK)
- MPU AXI Module ID Register (MPUXn_MID)

■ Memory Layout of MPU AXI Registers

Offset	+3	+2	+1	+0
0x00000000	MPUXn_CTRL0 00000000 00000000 00000001 00000000			
0x00000004	MPUXn_NMIEN 00000000 00000000 00000000 00000001			
0x00000008	MPUXn_WERRC 00000000 00000000 00000XXX XXXXXXX0			
0x0000000C	MPUXn_WERRA XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x00000010	MPUXn_RERRC 00000000 00000000 00000XXX XXXXXXX0			
0x00000014	MPUXn_RERRA XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Offset	+3	+2	+1	+0
0x00000018	MPUXn_CTRL1 00000000 00000000 00000000 00000000			
0x0000001C	MPUXn_SADDR1 00000000 00000000 00000000 00000000			
0x00000020	MPUXn_EADDR1 00000000 00000000 00000000 01111111			
0x00000024	MPUXn_CTRL2 00000000 00000000 00000000 00000000			
0x00000028	MPUXn_SADDR2 00000000 00000000 00000000 00000000			
0x0000002C	MPUXn_EADDR2 00000000 00000000 00000000 01111111			
0x00000030	MPUXn_CTRL3 00000000 00000000 00000000 00000000			
0x00000034	MPUXn_SADDR3 00000000 00000000 00000000 00000000			
0x00000038	MPUXn_EADDR3 00000000 00000000 00000000 01111111			
0x0000003C	MPUXn_CTRL4 00000000 00000000 00000000 00000000			
0x00000040	MPUXn_SADDR4 00000000 00000000 00000000 00000000			
0x00000044	MPUXn_EADDR4 00000000 00000000 00000000 01111111			
0x00000048	MPUXn_CTRL5 00000000 00000000 00000000 00000000			
0x0000004C	MPUXn_SADDR5 00000000 00000000 00000000 00000000			
0x00000050	MPUXn_EADDR5 00000000 00000000 00000000 01111111			
0x00000054	MPUXn_CTRL6 00000000 00000000 00000000 00000000			
0x00000058	MPUXn_SADDR6 00000000 00000000 00000000 00000000			
0x0000005C	MPUXn_EADDR6 00000000 00000000 00000000 01111111			
0x00000060	MPUXn_CTRL7 00000000 00000000 00000000 00000000			
0x00000064	MPUXn_SADDR7 00000000 00000000 00000000 00000000			
0x00000068	MPUXn_EADDR7 00000000 00000000 00000000 01111111			

Offset	+3	+2	+1	+0
0x0000006C	MPUXn_CTRL8 00000000 00000000 00000000 00000000			
0x00000070	MPUXn_SADDR8 00000000 00000000 00000000 00000000			
0x00000074	MPUXn_EADDR8 00000000 00000000 00000000 01111111			
0x00000078	MPUXn_UNLOCK 00000000 00000000 00000000 00000000			
0x0000007C	MPUXn_MID 00000000 00000000 00000000 00000000			

4.1. MPU AXI Control Register (MPUXn_CTRL0)

MPU AXI Control Register can be used by the software to configure the MPU AXI. This register provides a bit to enable the MPU AXI monitoring and protection function. It also provides permission attributes for background region. It also provides controls for enabling or disabling of privileged mode overwrite feature and MPU stop feature. Lastly it provides MPU AXI lock status and controls the status for Non-Maskable Interrupt.

■ **MPUXn_CTRL0**

MPUXn_CTRL0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	AP[2]	AP[1]	AP[0]	read0	read0	read0	read0	read0	read0	MPUENC	MPUEN	read0	read0	read0	PROT	POEN	MPUSTOPEN	MPUSTOP	LST	read0	read0	read0	read0	read0	read0	NMICL	NMI
R0	R0	R0	R0	R0	RWP	RWP	RWP	R0	R0	R0	R0	R0	R0	RWP	R	R0	R0	R0	RWP	RWP	RWP	R	R	R0	R0	R0	R0	R0	R0	R0WP	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:27]	read0	-
[26:24]	AP	Access Permissions for Background Region These bits are used to control access permissions for background region. For detailed description of these bits refer to Table "Access Permissions".
[23:18]	read0	-
[17]	MPUENC	MPU AXI Enable Control "0": MPU AXI monitoring and protection function is disabled "1": MPU AXI monitoring and protection function is enabled Read returns status of MPUENC bit. The region enable status to be effective takes some delay with reference to write on MPUENC bit. The actual enable status can be read through
[16]	MPUEN	MPU AXI Enable Status "0": MPU AXI monitoring and protection function is disabled. All accesses on AXI master write or read address channel inter- faces are passed on to AXI memory write or read address chan- nel interfaces without any protection "1": MPU AXI monitoring and protection function is enabled
[15:13]	read0	-
[12]	PROT	Privilege Attribute When privileged mode overwrite feature is available and also MPUXn_CTRL0:POEN = "1", privilege mode attribute on AXI memory interfaces are controlled by this bit. "0": Non-privilege mode "1": Privilege mode

Bit Position	Bit Field Name	Bit Description
[11]	POEN	Privileged Mode Overwrite Feature Enable "0": Privileged mode overwrite feature is disabled "1": Privileged mode overwrite feature is enabled Note: For availability of privileged mode overwrite feature, refer to device specific datasheet.
[10]	MPUSTOPEN	Enable for MPU STOP Feature "0": MPU stop feature is disabled "1": MPU stop feature is enabled This bit along with MPU stop input controls the STOP status of MPU AXI.
[9]	MPUSTOP	MPU Stop Status "0": MPU AXI is not stopped "1": MPU AXI is stopped (i.e.MPUXn_CTRL0:MPUSTOPEN = "1" and MPU stop input is asserted). All accesses on AXI master memory write or read address channel are converted to FIXED type burst with predefined address Note: For availability of MPU stop feature, refer to device specific datasheet.
[8]	LST	MPU Lock Status "0": MPU AXI is unlocked, registers in MPU AXI can be written "1": MPU AXI is locked, no registers (other than MPUXn_UNLOCK register) in MPU AXI can be written
[7:2]	read0	-
[1]	NMICL	NMI Interrupt Clear "0": No effect "1": Clears the NMI interrupt flag Read returns "0".
[0]	NMI	NMI Interrupt Flag This bit indicates that the memory protection violation is detected for an AXI transaction.

Note:

- *The register bits MPUXn_CTRL0:AP, MPUXn_CTRL0:MPUSTOPEN, MPUXn_CTRL0:POEN and MPUXn_CTRL0:PROT in MPUXn_CTRL0 register can only be written when MPU is disabled (MPUXn_CTRL0:MPUEN = "0"). This also implies that when MPUXn_CTRL0:MPUEN = "1":*
  - *32-bit or 16-bit access to this register is not allowed*
  - *8-bit access is required to disable the MPU*
  - *8-bit access is required to clear the NMI interrupt flag*

4.2. MPU AXI NMI Enable Register (MPUXn_NMIEN)

MPU AXI NMI Enable Register can be used by software to reset NMI enable bit. Default value of NMI enable bit is "1". This bit can be reset by software only once after reset operation.

■ MPUXn_NMIEN

MPUXn_NMIEN																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	NMIEN
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit Position	Bit Field Name	Bit Description
[31:8]	read0	-
[7:1]	read0	-
[0]	NMIEN	NMI Interrupt Enable This bit decides whether the NMI interrupt flag is routed to NMI interrupt signal or not. "0": NMI interrupt flag does not trigger NMI interrupt signal "1": NMI interrupt flag triggers NMI interrupt signal The value of this bit can be changed only once after reset.

4.3. MPU AXI Write Error Control Register (MPUXn_WERRC)

This is a read-only register that provides the software the control information of AXI transaction on AXI master write address channel interface for which memory protection violation was detected. Software can read this register to get privileged mode, burst length, burst size, and burst type information of AXI transaction.

■ MPUXn_WERRC

MPUXn_WERRC																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	AWSIZE[2]	AWSIZE[1]	AWSIZE[0]	AWBURST[1]	AWBURST[0]	AWLEN[3]	AWLEN[2]	AWLEN[1]	AWLEN[0]	AWPROTPRIV	AWMPV
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[10:8]	AWSIZE	AXI Transaction Burst Size This bit provides the status of AWSIZE[2:0] signals of AXI transaction for which memory protection violation is detected.
[7:6]	AWBURST	AXI Transaction Burst Type This bit provides the status of AWBURST[1:0] signals of AXI transaction for which memory protection violation is detected.
[5:2]	AWLEN	AXI Transaction Burst Length This bit provides the status of AWLEN[3:0] signals of AXI transaction for which memory protection violation is detected.
[1]	AWPROTPRIV	AXI Transaction Privileged Mode This bit provides the status of AWPROT[0] signal of AXI transaction for which memory protection violation is detected.
[0]	AWMPV	AXI Write Memory Protection Violation This bit indicates that memory protection violation is detected on AXI write address channel. Writing "1" to MPUXn_CTRL0:NMICL bit clears AWMPV bit.

4.4. MPU AXI Write Error Address Register (MPUXn_WERRA)

This is a read-only register that provides the software the write address of AXI transaction on AXI master write address channel interface for which memory protection violation was detected.

■ MPUXn_WERRA

MPUXn_WERRA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
AWADDR[31]	AWADDR[30]	AWADDR[29]	AWADDR[28]	AWADDR[27]	AWADDR[26]	AWADDR[25]	AWADDR[24]	AWADDR[23]	AWADDR[22]	AWADDR[21]	AWADDR[20]	AWADDR[19]	AWADDR[18]	AWADDR[17]	AWADDR[16]	AWADDR[15]	AWADDR[14]	AWADDR[13]	AWADDR[12]	AWADDR[11]	AWADDR[10]	AWADDR[9]	AWADDR[8]	AWADDR[7]	AWADDR[6]	AWADDR[5]	AWADDR[4]	AWADDR[3]	AWADDR[2]	AWADDR[1]	AWADDR[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit Position	Bit Field Name	Bit Description
[31:0]	AWADDR	AXI Write Address Write address of AXI transaction for which memory protection violation is detected.

4.5. MPU AXI Read Error Control Register (MPUXn_RERRC)

This is a read-only register that provides the software the control information of AXI transaction on AXI master read address channel interface for which memory protection violation was detected. Software can read this register to get privileged mode, burst length, burst size, and burst type information of AXI transaction.

■ MPUXn_RERRC

MPUXn_RERRC																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	ARSIZE[2]	ARSIZE[1]	ARSIZE[0]	ARBURST[1]	ARBURST[0]	ARLEN[3]	ARLEN[2]	ARLEN[1]	ARLEN[0]	ARPROTPRIV	ARMPV
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[10:8]	ARSIZE	AXI Transaction Burst Size This bit provides the status of ARSIZE[2:0] signals of AXI transaction for which memory protection violation is detected.
[7:6]	ARBURST	AXI Transaction Burst Type This bit provides the status of ARBURST[1:0] signals of AXI transaction for which memory protection violation is detected.
[5:2]	ARLEN	AXI Transaction Burst Length This bit provides the status of ARLEN[3:0] signals of AXI transaction for which memory protection violation is detected.
[1]	ARPROTPRIV	AXI Transaction Privileged Mode This bit provides the status of ARPROT[0] signal of AXI transaction for which memory protection violation is detected.
[0]	ARMPV	AXI Read Memory Protection Violation This bit indicates that memory protection violation is detected on AXI read address channel. Writing "1" to MPUXn_CTRL0:NMICL bit clears ARMPV bit.

4.6. MPU AXI Read Error Address Register (MPUXn_RERRA)

This is a read-only register that provides the software the read address of AXI transaction on AXI master read address channel interface for which memory protection violation was detected.

■ MPUXn_RERRA

MPUXn_RERRA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ARADDR[31]	ARADDR[30]	ARADDR[29]	ARADDR[28]	ARADDR[27]	ARADDR[26]	ARADDR[25]	ARADDR[24]	ARADDR[23]	ARADDR[22]	ARADDR[21]	ARADDR[20]	ARADDR[19]	ARADDR[18]	ARADDR[17]	ARADDR[16]	ARADDR[15]	ARADDR[14]	ARADDR[13]	ARADDR[12]	ARADDR[11]	ARADDR[10]	ARADDR[9]	ARADDR[8]	ARADDR[7]	ARADDR[6]	ARADDR[5]	ARADDR[4]	ARADDR[3]	ARADDR[2]	ARADDR[1]	ARADDR[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit Position	Bit Field Name	Bit Description
[31:0]	ARADDR	AXI Read Address Read address of AXI transaction for which memory protection violation is detected.

4.7. MPU AXI Region Control Registers (MPUXn_CTRL1 to 8)

MPU AXI Region Control Register is used to specify access permission for a particular region. Software can also use this register for enabling or disabling the particular region. MPUXn_CTRL1 Control Register for Region 1 is explained below.

■ MPUXn_CTRL1

MPUXn_CTRL1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	read0	AP[2]	AP[1]	AP[0]	read0	read0	read0	read0	read0	read0	MPUENC	MPUEN
R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RWP	RWP	RWP	R0	R0	R0	R0	R0	R0	RWP	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:16]	read0	-
[15:11]	read0	-
[10:8]	AP	Access Permissions These bits are used to control access permissions for region 1.
[7:2]	read0	-
[1]	MPUENC	Enable Control bit "0": Memory protection for region 1 is disabled "1": Memory protection for region 1 is enabled Read returns status of MPUENC bit. The region enable status to be effective takes some delay with reference to write on MPUENC bit. The actual enable status can be read through MPUXn_CTRL1:MPUEN bit.
[0]	MPUEN	Enable Status "0": Memory protection for region 1 is disabled "1": Memory protection for region 1 is enabled This is a read-only bit, writing to this bit does not have any effect.

Notes:

- *Access permission bits (i.e. MPUXn_CTRL1 to 8:AP) can only be written when corresponding region is disabled (MPUXn_CTRL1 to 8:MPUEN = "0") or MPU is disabled (MPUXn_CTRL0:MPUEN = "0").*
- *This also implies that when MPUXn_CTRL0:MPUEN = "1" and MPUXn_CTRL1 to 8:MPUEN = "1":*
  - *32-bit or 16-bit access to this register is not allowed*
  - *8-bit access is required to disable the MPU region*

4.8. MPU AXI Start Address Registers (MPUXn_SADDR1 to 8)

Each region in MPU AXI can be set by specifying start address and end address for that particular region. MPUXn_SADDR1 to 8 registers are used to specify start address for eight regions. Start address indicates the first address for that region. MPUXn_SADDR1 Start Address Register for Region 1 is explained below.

■ MPUXn_SADDR1

MPUXn_SADDR1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SADDR[31]	SADDR[30]	SADDR[29]	SADDR[28]	SADDR[27]	SADDR[26]	SADDR[25]	SADDR[24]	SADDR[23]	SADDR[22]	SADDR[21]	SADDR[20]	SADDR[19]	SADDR[18]	SADDR[17]	SADDR[16]	SADDR[15]	SADDR[14]	SADDR[13]	SADDR[12]	SADDR[11]	SADDR[10]	SADDR[9]	SADDR[8]	SADDR[7]	SADDR[6]	SADDR[5]	SADDR[4]	SADDR[3]	SADDR[2]	SADDR[1]	SADDR[0]
RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	SADDR	Start Address Start address for region 1

Note:

- *MPUXn_SADDR1 to 8 registers can only be written when corresponding region is disabled (MPUXn_CTRL1 to 8:MPUEN = "0") or MPU is disabled (MPUXn_CTRL0:MPUEN = "0").*

4.9. MPU AXI End Address Registers (MPUXn_EADDR1 to 8)

Each region in MPU AXI can be set by specifying start address and end address for that particular region. MPUXn_EADDR1 to 8 registers are used to specify end addresses for eight regions. End Address indicates the last address for that region. MPUXn_EADDR1 End Address Register for Region 1 is explained below.

■ MPUXn_EADDR1

MPUXn_EADDR1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EADDR[31]	EADDR[30]	EADDR[29]	EADDR[28]	EADDR[27]	EADDR[26]	EADDR[25]	EADDR[24]	EADDR[23]	EADDR[22]	EADDR[21]	EADDR[20]	EADDR[19]	EADDR[18]	EADDR[17]	EADDR[16]	EADDR[15]	EADDR[14]	EADDR[13]	EADDR[12]	EADDR[11]	EADDR[10]	EADDR[9]	EADDR[8]	EADDR[7]	EADDR[6]	EADDR[5]	EADDR[4]	EADDR[3]	EADDR[2]	EADDR[1]	EADDR[0]
RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP	RWP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit Position	Bit Field Name	Bit Description
[31:0]	EADDR	End Address End address for region 1

Note:

- *MPUXn_EADDR1 to 8 registers can only be written when corresponding region is disabled (MPUXn_CTRL1 to 8:MPUEN = "0") or MPU is disabled (MPUXn_CTRL0:MPUEN = "0").*

4.10. MPU AXI Unlock Register (MPUXn_UNLOCK)

The software can use this register to lock or unlock the MPU AXI registers for write access.

■ MPUXn_UNLOCK

MPUXn_UNLOCK																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
UNLOCK[31]	UNLOCK[30]	UNLOCK[29]	UNLOCK[28]	UNLOCK[27]	UNLOCK[26]	UNLOCK[25]	UNLOCK[24]	UNLOCK[23]	UNLOCK[22]	UNLOCK[21]	UNLOCK[20]	UNLOCK[19]	UNLOCK[18]	UNLOCK[17]	UNLOCK[16]	UNLOCK[15]	UNLOCK[14]	UNLOCK[13]	UNLOCK[12]	UNLOCK[11]	UNLOCK[10]	UNLOCK[9]	UNLOCK[8]	UNLOCK[7]	UNLOCK[6]	UNLOCK[5]	UNLOCK[4]	UNLOCK[3]	UNLOCK[2]	UNLOCK[1]	UNLOCK[0]
R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP	R0WP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	UNLOCK	MPU AXI Unlock The MPU AXI Unlock Register protects the MPU AXI module from being modified accidently by software. The MPU AXI registers cannot be written until this register has been written with a specific unlock value. The correct value for unlocking can be written only in privileged mode. Reading this register always returns a zero. To lock the MPU AXI again software must write another value specific to lock. Write access to MPU AXI registers without unlocking or writing a value other than the lock or unlock values to this register causes a protection error. Notes: This register can not be written by 8-bit or 16-bit write access; any such access causes protection error. For more details on lock and unlock values, refer to the device specific datasheet.

4.11. MPU AXI Module ID Register (MPUXn_MID)

This is a read-only register with a unique module identification number which identifies the version of the MPU AXI module used in the MCU.

■ MPUXn_MID

MPUXn_MID																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MID[31]	MID[30]	MID[29]	MID[28]	MID[27]	MID[26]	MID[25]	MID[24]	MID[23]	MID[22]	MID[21]	MID[20]	MID[19]	MID[18]	MID[17]	MID[16]	MID[15]	MID[14]	MID[13]	MID[12]	MID[11]	MID[10]	MID[9]	MID[8]	MID[7]	MID[6]	MID[5]	MID[4]	MID[3]	MID[2]	MID[1]	MID[0]
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Position	Bit Field Name	Bit Description
[31:0]	MID	Module ID The MPU AXI module implemented in the device may vary from device to device. This register identifies the particular version of the hardware used in the device. This register helps in developing software to hardware version implemented in the device

5. Notes on Using MPU AXI

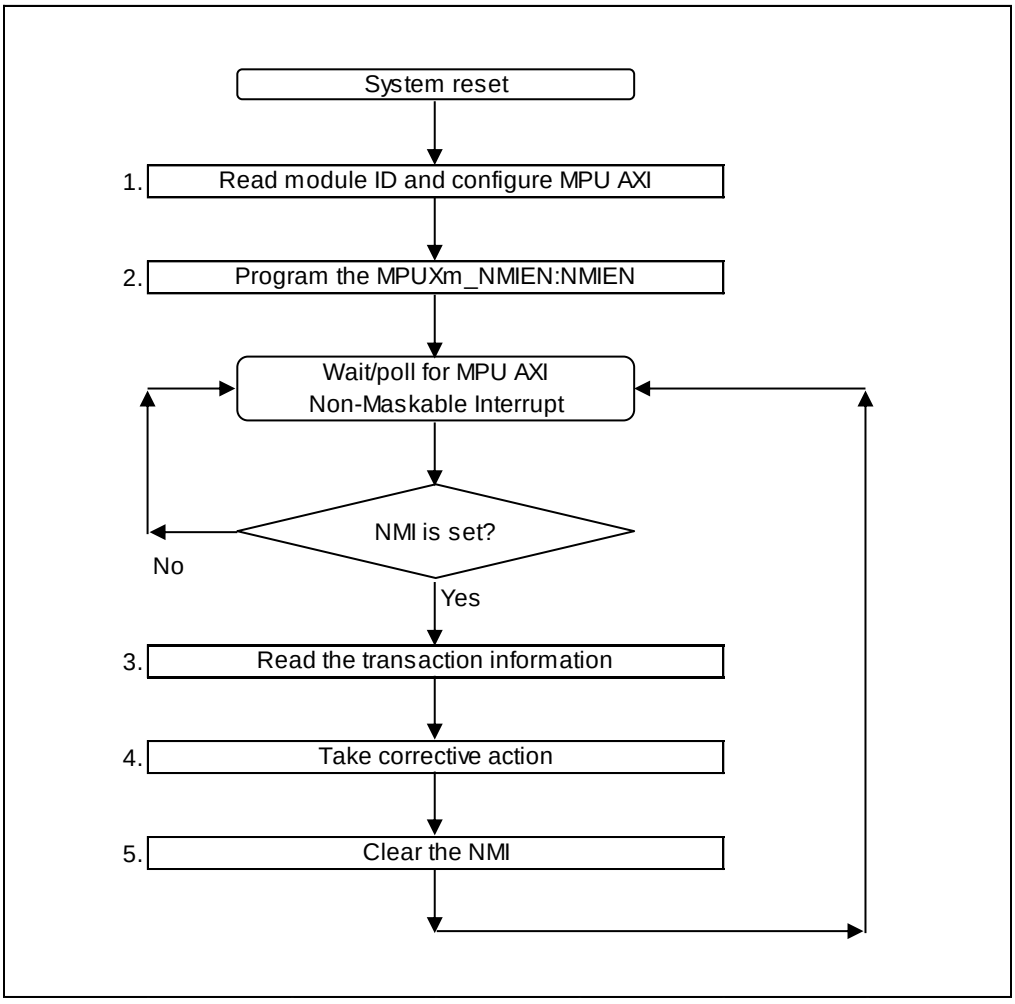
This section is the "programmer's guide", which lists the usage notes for programming the MPU AXI module. It is recommended to read these guidelines before programming the MPU AXI module.

■ General Usage Notes

- Reserved bits return undefined values. The software programs shall be independent of the values read from the reserved register bits
- The MPU AXI supports storage of information of first memory protection violation on the bus. Therefore, once an NMI interrupt flag is set, the further monitoring of AXI master interface is stalled until MPUXn_CTRL0:NMI bit is cleared. This implies that any protection violation occurring on the bus while the MPUXn_CTRL0:NMI is set are simply ignored by MPU AXI. Software developers must therefore try to keep the ISR size small, so that the NMI interrupt of MPU AXI does not remain unattended for long

■ Steps in Programming the MPU AXI Module

Figure 5-1 Programmer's Flowchart



1. After the system reset, the software detects the module ID number of MPU AXI, by reading the MPUXn_MID register. This helps it in identifying the attributes and capabilities supported by the MPU AXI module. Then the software configures MPU AXI by setting appropriate registers.

2. By default, MPU AXI propagates the MPUXn_CTRL0:NMI flag to the CPU through the Interrupt Controller. If polling mode is desired, the software can reset the MPUXn_NMIEN:NMIEN bit to "0".

Note:

- *The MPUXn_NMIEN:NMIEN can be written only once after reset. Subsequent write accesses to this bit have no visible impact on the state of this bit.*

3. When the NMI is triggered or is in polling mode, if the software detects during its polling cycle that the MPUXn_CTRL0:NMI status flag is set, the CPU is invoked. This would read the status information collected and stored by MPU AXI in its CSR.

4. The software diagnoses the information about the transaction for which memory protection violation was detected and initiates a corrective action (if any).

5. Once the software has processed the information from the status registers, it shall clear the MPUXn_CTRL0:NMI flag by writing a "1" to the MPUXn_CTRL0:NMICL bit. Clearing MPUXn_CTRL0:NMI flag ensures that the MPU AXI starts monitoring the AXI master interfaces again for checking memory protection violation.

Note:

- *Software may clear the NMI flag before taking corrective action, hence steps 4 and 5 can be interchanged.*

CHAPTER 38: Sound System Configuration

This chapter explains the definition of sound system and configuration.

1. Overview
2. Configuration and Block Diagram
3. Operation
4. Configuration and limitation of each module
5. Note

CODE: SOUNDSYSTEM_E1.03-0

1. Overview

The sound system is defined as the function group of following modules.

Function Group	Function Module
Input module	DMAC
	Sound waveform generator (SWFG)
Sound mixer	Sound mixer (SMIX)
Output module	Audio DAC (DAC)
	PCM-PWM
	I2S

Here the configuration procedure and notes are described in this chapter for each applied scope when the modules are used as the sound system.

See the dedicated chapter that describes in detailed each module composing the sound system before referring this chapter.

1.1. Input Module

There are 2 types of sound sources.

One is a sound source that is generated by the sound waveform generator (SWFG). SWFG with easy configurations by software can generate 16bit sound sources of PCM format, and output it as 32bit monaural source.

The other is a sound source that is in internal or external storage. DMAC or CPU can transfer sound source in the embedded Flash memory or data RAM, or comes from external memory via communication interface such as I2S.

1.2. Sound Mixer

The sound mixer (SMIX) synthesizes inputted sound sources by saturation calculation, and performs various effects of sound volume (configured volume, mute effect, fade-in, or fade out) automatically by hardware operation.

1.3. Output Module

The mixed sound source is outputted out of microcontroller.

Audio DAC converts PCM sound source to level of analog voltage. It enables to mount analog amplifier directly at analog output port of MCU.

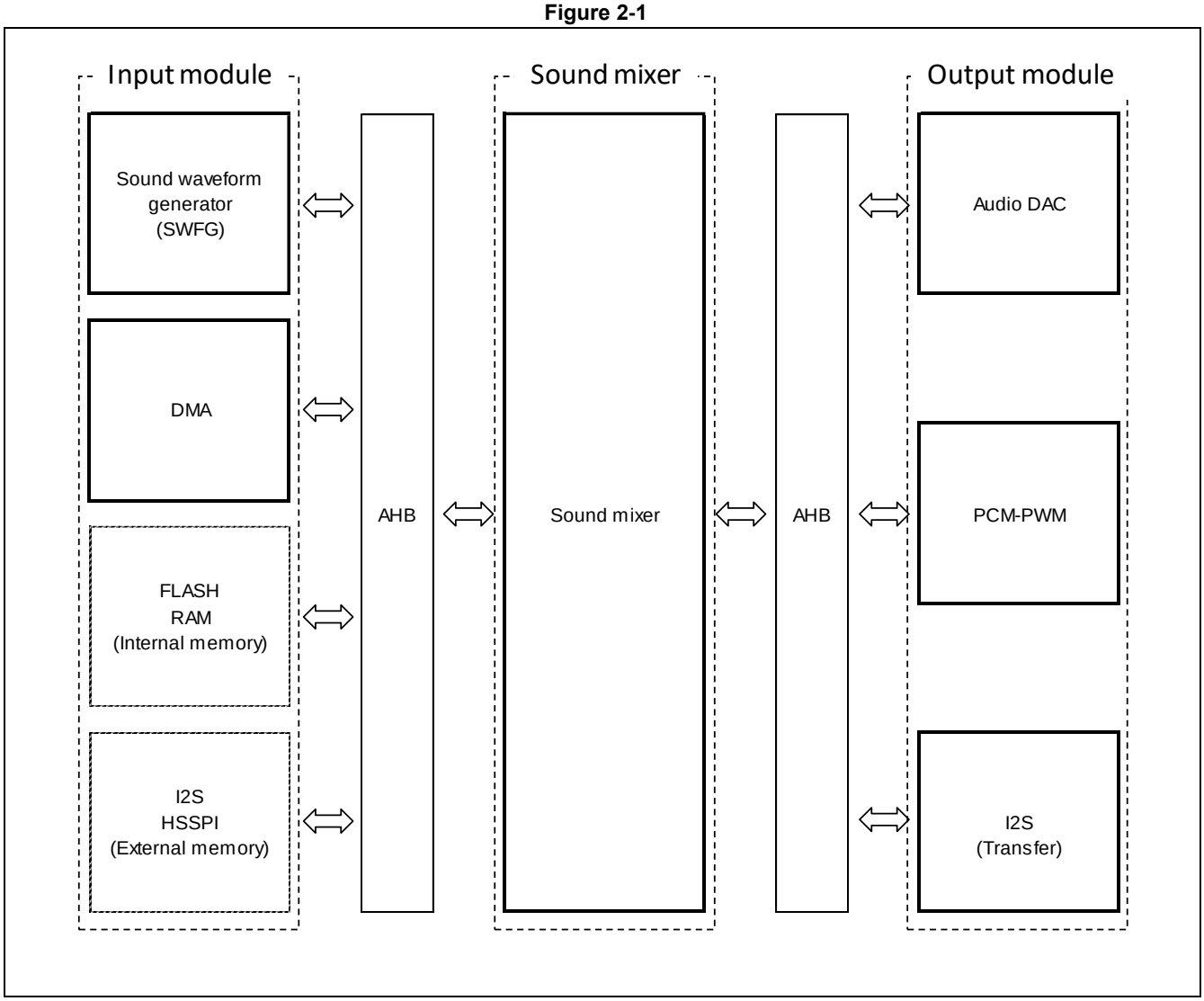
PCM-PWM converts PCM sound source to PWM (Pulse Width Modulation) waveform and outputs it. Through a kind of Integral circuit it can be converted to level of analog voltage.

I2S (Inter IC Sound) is a serial interface protocol. The sound system has its transfer facility as sound system function.

These 3 functions are exclusively used for the output module of the sound system.

2. Configuration and Block Diagram

Figure 2-1



Note:

- *I2S connected to the Sound Mixer supports only Transfer mode.*

3. Operation

The operation of the sound system is a co-operation of the modules which are defined as 1. The followings are descriptions of the procedures for each case.

The detailed explanation of the module operation and its note is described in each chapter. Before referring this chapter, see the chapter of each module in advance.

3.1. Start Operations

Take the following procedure of 3.1.1 to 3.1.3 to start up the operation of the sound system.

3.1.1. Configuration of Input Modules

SWFG Configuration

Start sound source generations configuring SWFG.

The generations will practically begin by means of each transfer request from the sound mixer.

DMAC Configuration

Start sound source transmissions configuring DMAC.

The transmissions will practically begin by means of each transfer request from SMIX.

3.1.2. Configuration of Sound Mixer Operations

Start a mixing operation configuring SMIX.

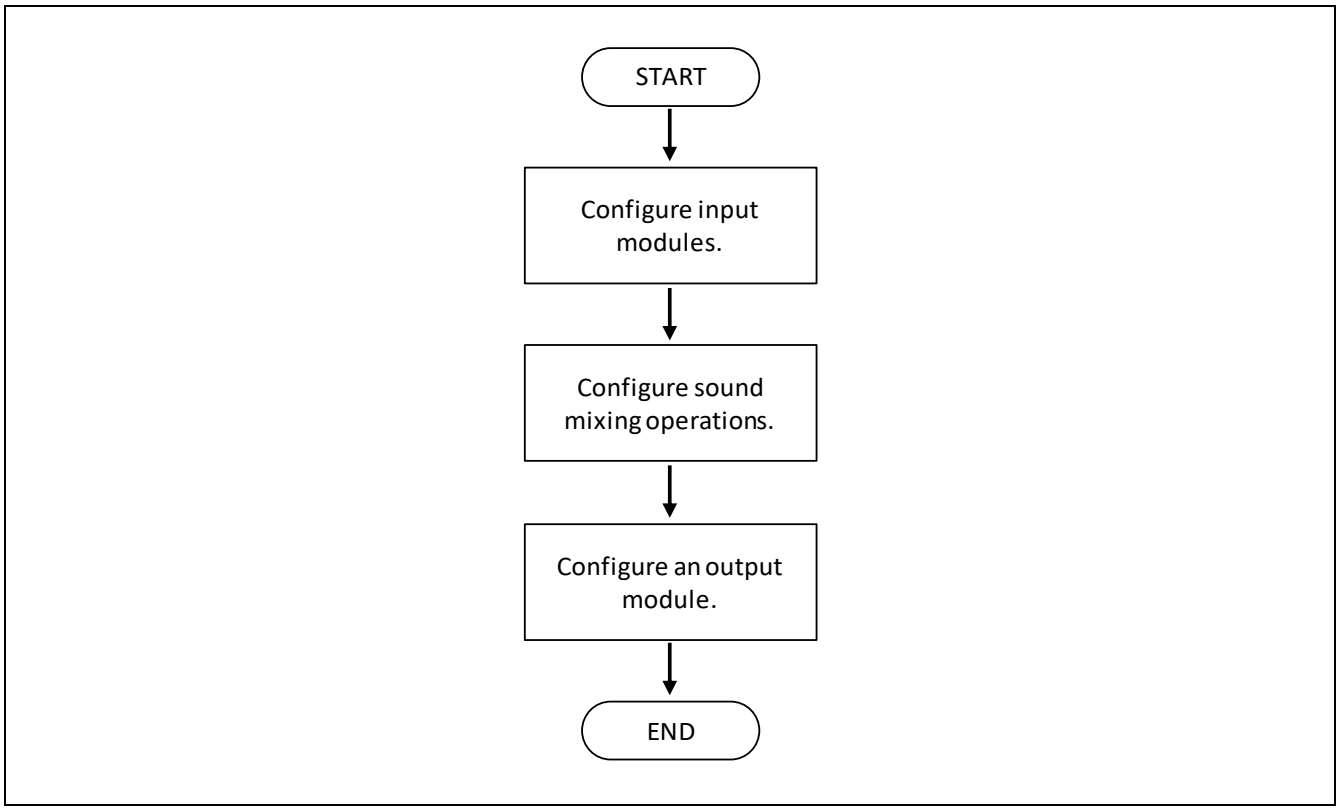
The mixing will practically begin by means of a transfer request from the output module.

3.1.3. Configuration of an Output Module

Start an output operation configuring one of the modules out of the audio DAC, PCM-PWM, or I2S.

The co-operation of the modules as a sound system will begins by means of a start operation of the output module.

Figure 3-1 Start Operations



Notes:

- *CPU can transfer sound sources to the input channel PMIS0 to 4 of SMIX with data transfer requests of SMIX.*
- *Note some sound source which is transferred from SWFG is deleted by buffer initialization (clear control) during SMIX configuration. In order to avoid the phenomenon, WGSTART control to start sound source generation of SWFG can only be done after SMIX configuration.*

3.2. Addition of Sound Source Inputs

Take the following procedure to add additional sound sources to the operating sound system.

The existing mixed sound source can be continuously reproduced after the addition of new sound sources.

3.2.1. Configuration of Additional Input Sound Source

SWFG Configuration

Start additional sound source generations configuring channels of SWFG.

The generations will practically begin by means of each transfer request from SMIX.

DMA Configuration

Start additional sound source transmissions configuring a channel of DMA.

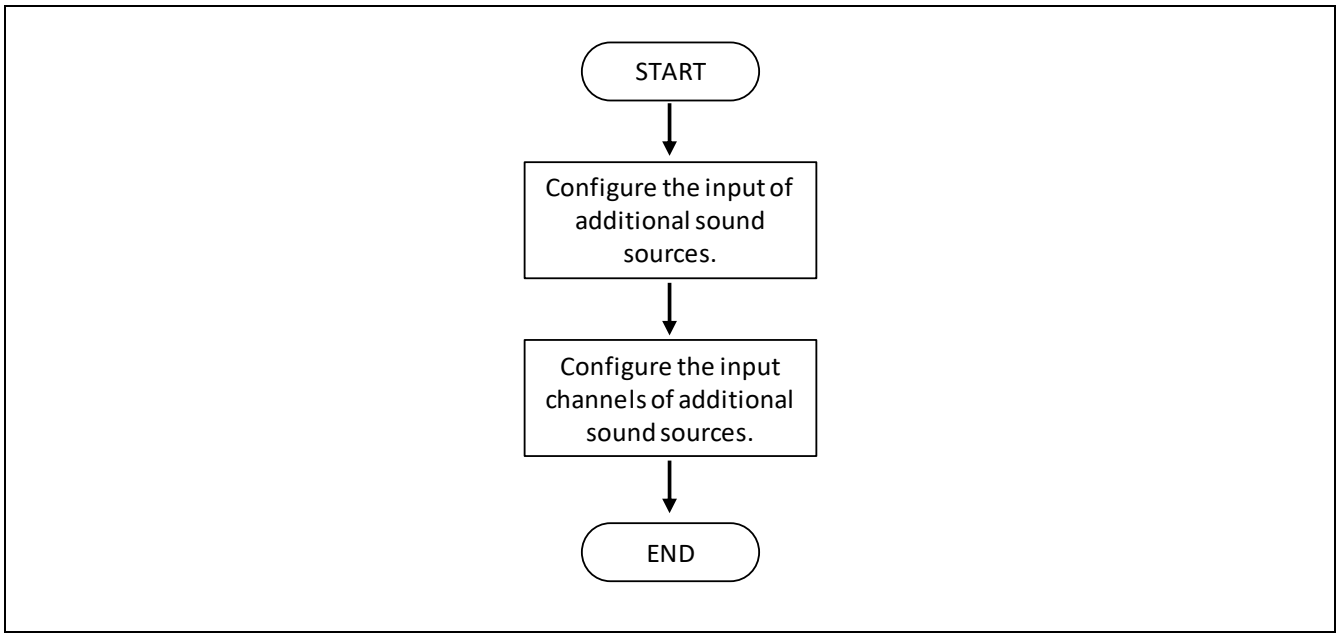
The transmissions will practically begin by means of each transfer request from SMIX.

3.2.2. Configuration of Additional Input Channel

Start an additional mixing operation configuring an input channel of SMIX.

The mixing will practically begin by means of a transfer request from the output module.

Figure 3-2 Addition of Sound Source Inputs



3.3. Stop Operations

Take the following procedure of 3.3.1 to 3.3.3 to stop the operation of the sound system.

3.3.1. Disabling the Output Module

Disable the output module. The procedure is described in the chapter of the audio-DAC, PCM-PWM and I2S in detailed.

Notes:

- *When the PCM-PWM or I2S is used as output module and the output module is changed from "enabled" to "disabled" (*1), the sound mixer should not be disabled and the output module should not be enabled during 16 sampling cycles. If this sequence is not followed, PCM data requested before disabling the output module may be transferred to the output module after enabling the output module. This may cause the DMA block error, Tx Block Size Error or FIFO overflow error.*
- *(*1) In PCM-PWM case, this means that the global enable bit (PCMPWMi_CONTROL,EN) is changed from 1 to 0.*
- *(*1) In I2S case, this means that the transmission enable bit (I2Sn_OPRREG:TXENB) or the I2S enable bit (I2Sn_OPRREG.START) is changed from 1 to 0.*

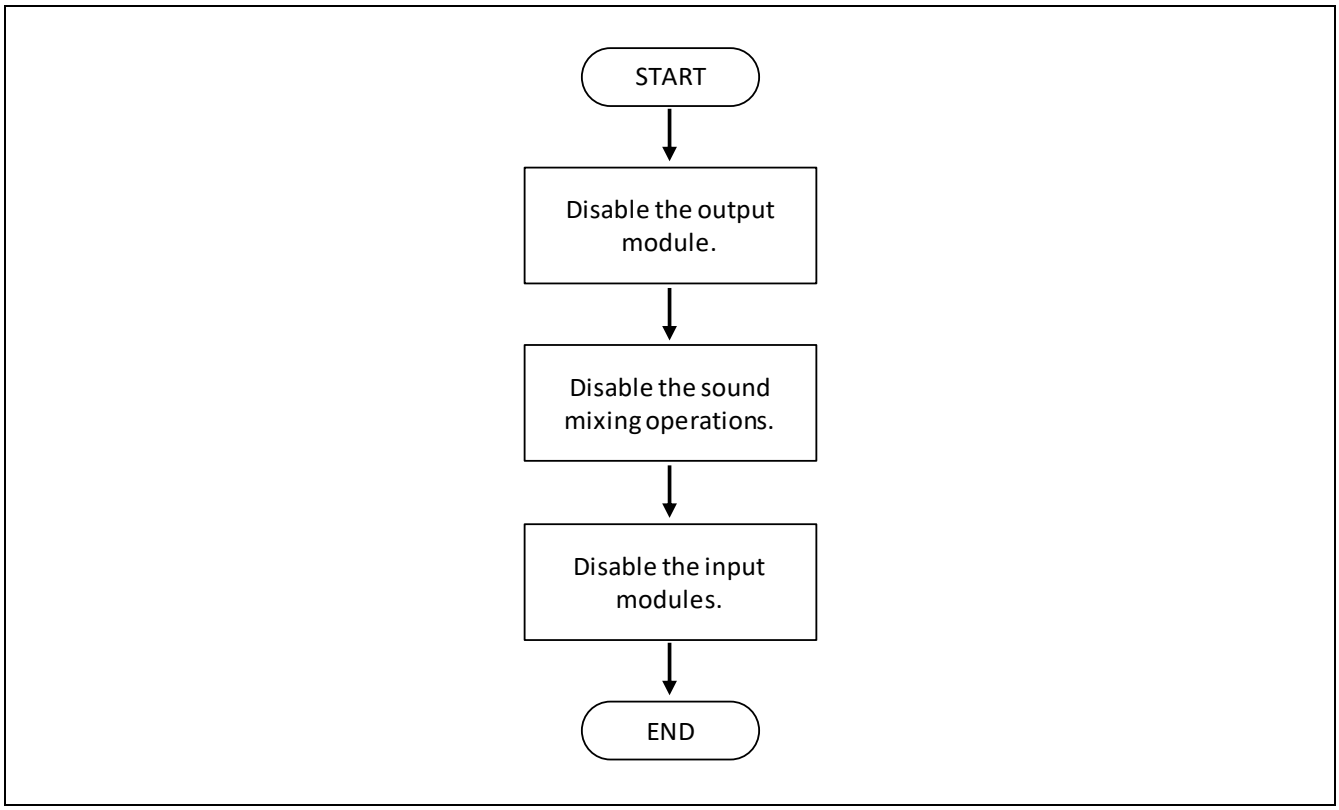
3.3.2. Disabling the Mixing

Disable the mixing operations of SMIX. The procedure is described in the chapter of SMIX in detailed.

3.3.3. Disabling the Input Modules

Disable the sound source generations and transmissions. The procedure is described in the chapter of DMA, interrupt and I2S in detailed.

Figure 3-3 Stop Operations



Notes:

- *While the sound system operates continuously, a stop of a sound source generation of SWFG keeps all mixing operations of SMIX being hung up. Suspend a particular channel of SWFG only after disabling a corresponding input channel SMIX.*
- *DMA transfer needs a same operation as the case of SWFG. Before suspending DMA transfer, disable a corresponding input channel of SMIX.*

3.4. Operation of a Sound Source

An application doesn't need the mixing operation but only needs an operation of an output module.

In the case the sound source should be transferred to SMIX as same as the case of a mixing of multi sound sources.

The sound system cannot directly transfer the sound source from an input module to an output module. Because every output module doesn't support any transfer requests for CPU and DMA.

On the other hand, every output module supports a transfer request for SMIX, that is, it is only SMIX that can transfer the sound source to an output module.

Notes:

- *SMIX only outputs the mixed sound source of 48 kHz sampling rate. Therefore the sound system cannot obtain the sound source with an original sampling rate even if the application doesn't need mixing function.*

- *SMIX has the volume effect control of the mute, fade-in, and fade-out. They are quite effective for a sound source even if the mixing operation is not necessary.*

3.5. Sound system reset Operations

The sound system has a dedicated reset.

It is controlled by Resource Input Setting register (RIC register).

This reset is used when the sound system cannot return from error status like DMA transfer error status or underflow error status.

Take the following procedure from 3.5.1 to 3.5.3 to use of the sound system reset.

The confirmation way of the sound system reset status is described on 3.5.4.

After releasing the sound system reset, confirm that the reset negation is completed, then the master can access to the sound system.

3.5.1. Disabling the Output Module

The sound system reset is asserted asynchronously.

Therefore, audio-DAC, PCM-PWM and I2S are stopped immediately.

To avoid this, audio-DAC, PCM-PWM and I2S should be disabled before sound system reset asserts.

Note:

- *Regarding PCMPWM, the output level on disable status is configurable by the register in PCMPWM. To prevent from changing the PCMPWM output by the sound system reset, output level selection register in PCMPWM should be initialized.*

3.5.2. Disabling the DMA controller

The sound system reset is asserted asynchronously.

Therefore, if the sound system reset occurs during DMA transfer, the transfer data from DMA controller cannot be written into the sound system.

However, the transfer data appear to be written for DMA, because any error responses are not returned from the sound system during the reset.

To avoid this, DMA controller should be disabled before sound system reset asserts.

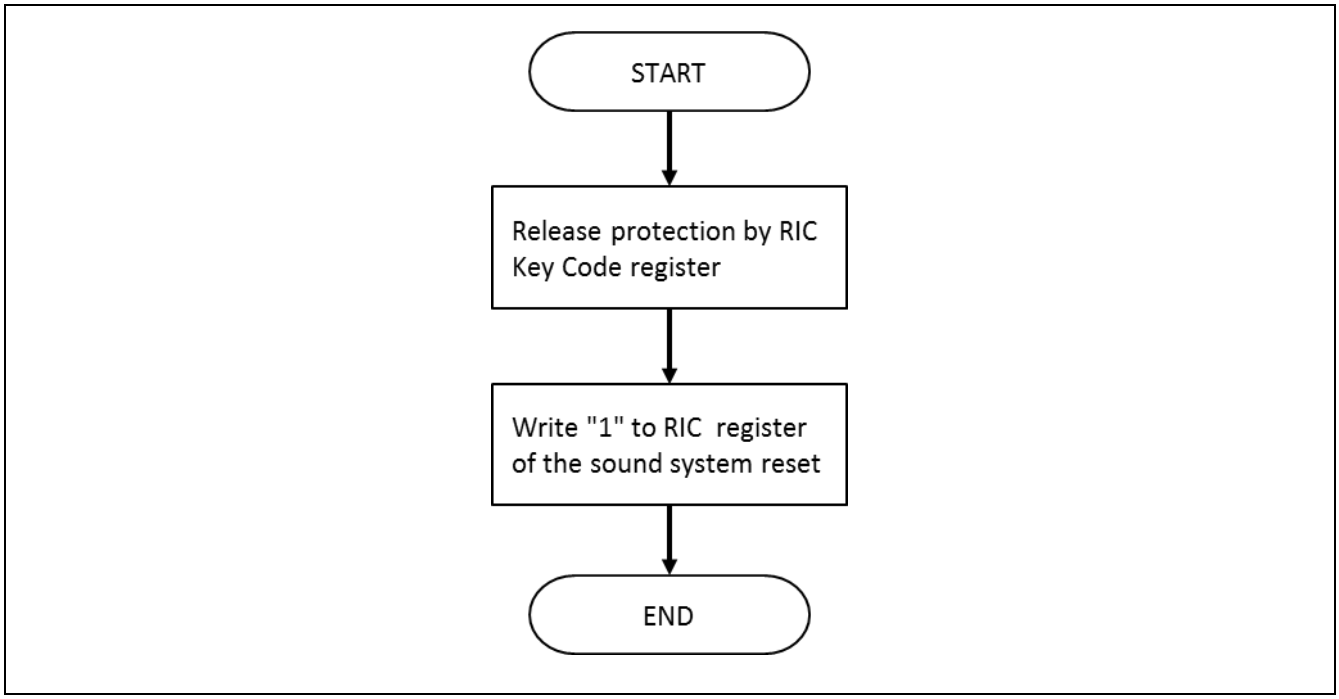
3.5.3. Configuration of the RIC register

Sound system reset is controlled by Resource Input Setting register (RIC register).

Therefore, it is protected by RIC Key Code register.

To assert/negate the sound system reset, the protection by RIC Key Code register must be released.

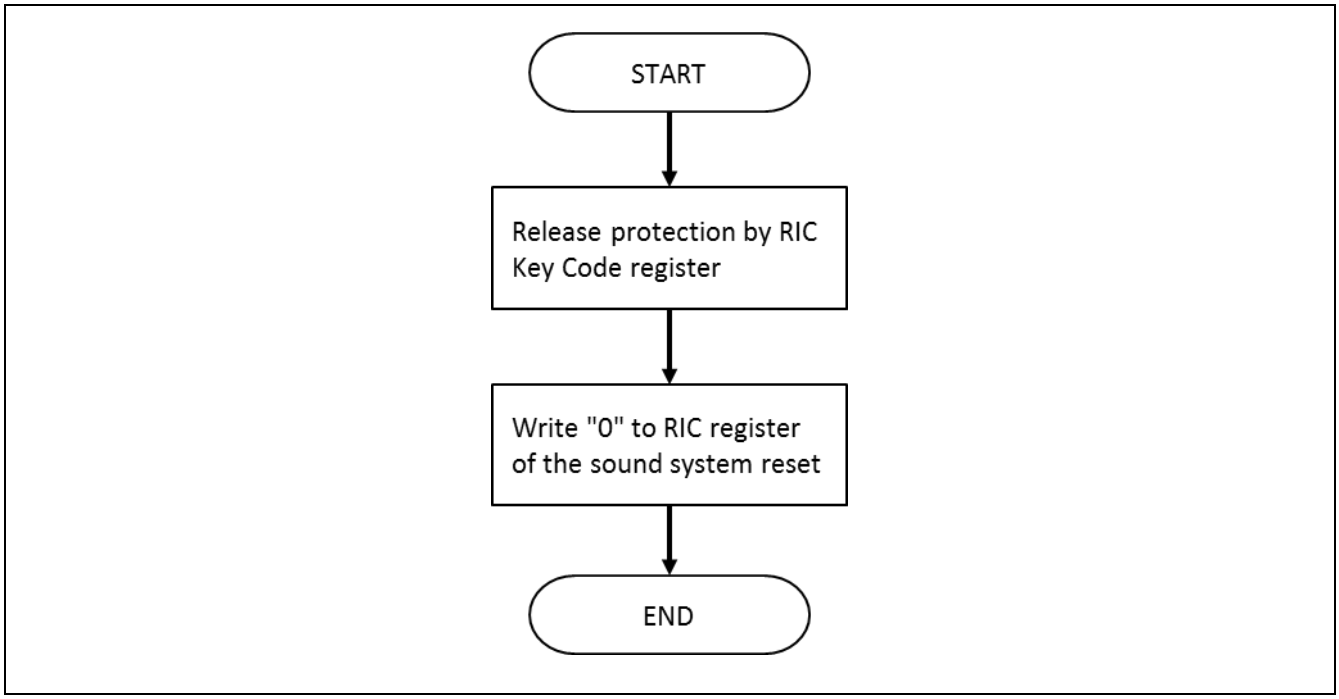
Figure 3-4 Sound System Assert Flow



Note:

- *Does not write other than "0x1" to RIC_RESIN1200 register in case of asserting the sound system reset.*

Figure 3-5 Sound System Negate Flow



Note:

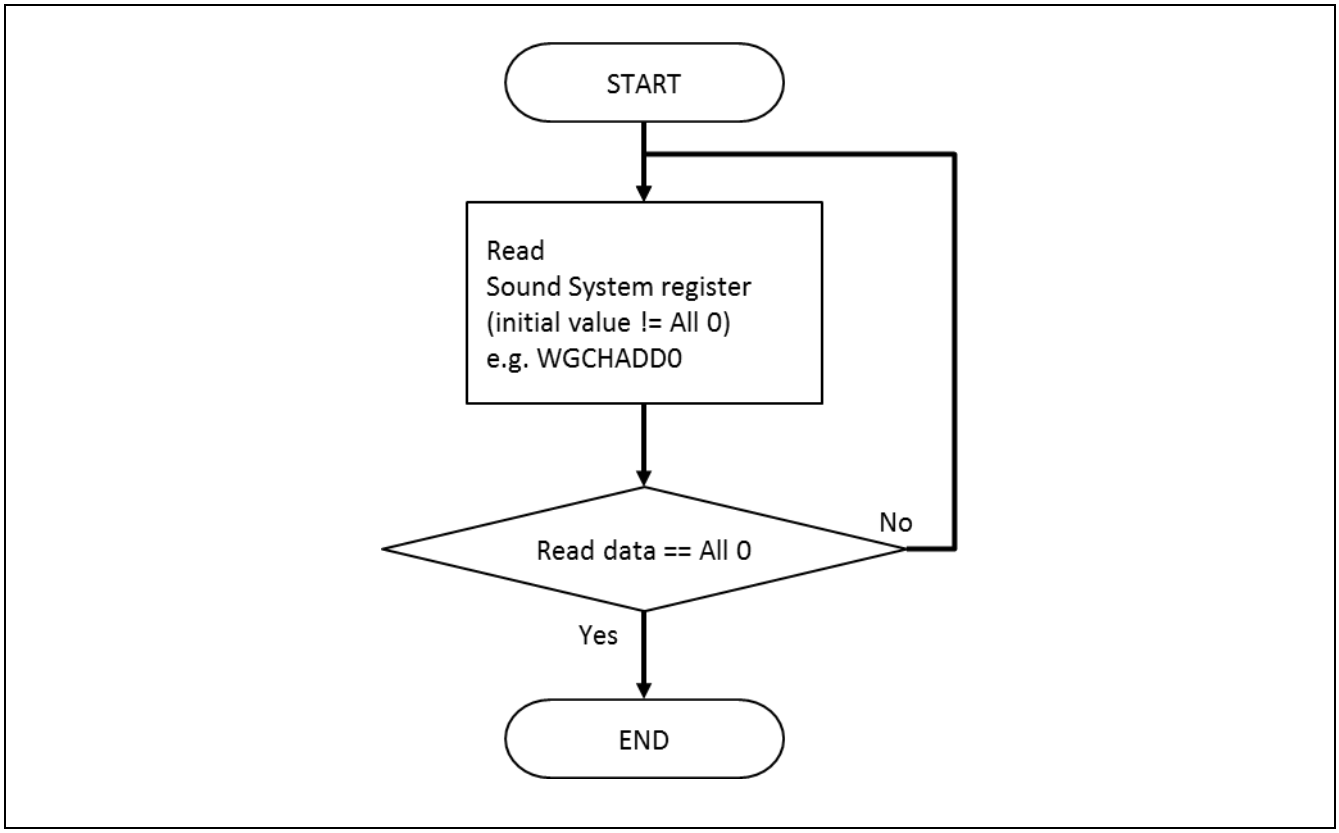
- *Does not write other than “0x0” to RIC_RESIN1200 register in case of negating the sound system reset.*

3.5.4. Confirmation of the reset status

To confirm the assert status of sound system reset, read the register in sound system whose initial value is other than all 0 after asserting sound system reset by RIC register.

If the read value is all 0, sound system reset is in asserting.

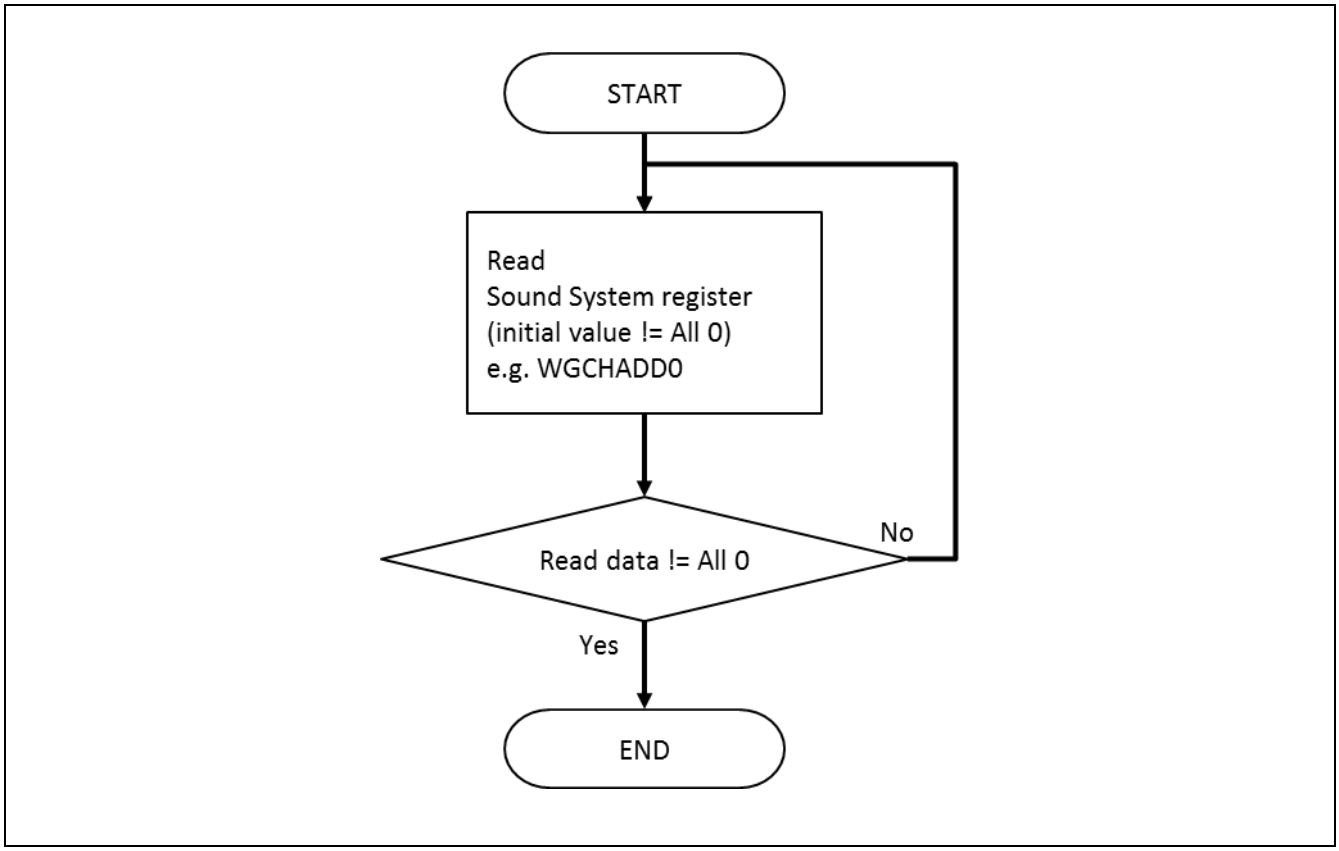
Figure 3-6 Assert Status Confirmation



To confirm the negate status of sound system reset, read the register in sound system whose initial value is other than all 0 after negating sound system reset by RIC register.

If the read value is not all 0, sound system reset is in negating.

Figure 3-7 Negate Status Confirmation



4. Configuration and limitation of each module

This section shows configurations and limitations about each module interface in the sound system.

4.1. Interface between SMIX and DAC

Output modules including DAC are connected to SMIX by a dedicated bus interface. On this interface, the sound source transfer between SMIX and DAC is triggered by a data request from DMA interface of DAC.

DAC has two initialization procedures (refer to the chapter of the DAC for the details). One is "DMA Interface Unused", and the other is "DMA Interface Used". DAC must be initialized by the procedure of "DMA Interface used", because the sound source is transferred from SMIX to DAC by using DMA interface.

Because it is impossible to transfer the sound source from DMAC to DAC directly, the sound source needs to be transferred to DAC via SMIX. Therefore, the initialization of the SMIX is necessary even if it is not necessary to do the mixing of the sound source. If the sound source is not mixing, only 1 PMIS input channel (any number of channels can be used) of SMIX should be enabled.

4.1.1. Configuration

The address of DAC must be set to Output destination address of SMIX (MACRO of MXOCTRL register). The actual setting value is the following.

- MACRO of MXOCTRL register : 0x1

Number of transfers to output destination of SMIX (DATAN of MXOCTRL register) must be same as FIFO empty threshold of DAC (FEST of DACTRL register).

4.1.2. Limitation

Because the sound sources are transferred surely from SMIX to DAC, the following registers must not be changed during the sound source translation.

Even if SMIX is disabled by MXCH register, the data transfer from SMIX to DAC is not stopped immediately. Therefore, the following register must not be changed after setting it at the initialization.

- DACTRL
- MXOCTRL

4.2. Interface between SMIX and PCM-PWM

Output modules including PCM-PWM are connected to SMIX by a dedicated bus interface. On this interface, the sound source transfer between SMIX and PCM-PWM is triggered by a data request from DMA interface of PCM-PWM.

PCM-PWM supports two type data transfers. One is "DMA mode", and the other is "CPU mode". PCM-PWM must be set to "DMA mode", because the sound source is transferred from SMIX to PCM-PWM by using DMA interface.

Because it is impossible to transfer the sound source from DMAC to PCM-PWM directly, the sound source needs to be transferred to PCM-PWM via SMIX. Therefore, the initialization of the SMIX is necessary even if it is not necessary to do the mixing of the sound source. If the sound source is not mixing, only 1 PMIS input channel (any number of channels can be used) of SMIX should be enabled.

4.2.1. Configuration

The address of PCM-PWM must be set to Output destination address of SMIX (MACRO of MXOCTRL register). The actual setting value is the following.

- MACRO of MXOCTRL register : 0x2

Number of transfers to output destination of SMIX (DATAN of MXOCTRL register) must be same as FIFO empty threshold of PCM-PWM (FEST of PCMPWMi_CONTROL register).

4.2.2. Limitation

Because the sound sources are transferred surely from SMIX to PCM-PWM, the following registers must not be changed during the sound source translation.

Even if SMIX is disabled by MXCH register, the data transfer from SMIX to PCM-PWM is not stopped immediately. Therefore, the following register must not be changed after setting it at the initialization.

- MXOCTRL

And, FEST of PCMPWMi_CONTROL must not be changed under the following condition.

- EN of PCMPWMi_CONTROL is set to 1, and
- DMAEN of PCMPWMi_CONTROL is set to 1

4.3. Interface between SMIX and I2S

Output modules including I2S are connected to SMIX by a dedicated bus interface. On this interface, the sound source transfer between SMIX and I2S is triggered by a data request from DMA interface of I2S.

I2S supports three type data transfers. These are "DMA", "Interrupt" and "Poling". I2S must be set to "DMA", because the sound source is transferred from SMIX to I2S by using DMA interface.

Because it is impossible to transfer the sound source from DMAC to I2S directly, the sound source needs to be transferred to I2S via SMIX. Therefore, the initialization of the SMIX is necessary even if it is not necessary to do the mixing of the sound source. If the sound source is not mixing, only 1 PMIS input channel (any number of channels can be used) of SMIX should be enabled.

4.3.1. Configuration

The address of I2S must be set to Output destination address of SMIX (MACRO of MXOCTRL register). The actual setting value is the following.

- MACRO of MXOCTRL register : 0x3

Number of transfers to output destination of SMIX (DATAN of MXOCTRL register) must be same as Tx FIFO Threshold of I2S (TFTH of I2Sn_INTCNT register).

4.3.2. Limitation

Because the sound sources are transferred surely from SMIX to I2S, the following registers must not be changed during the sound source translation.

Even if SMIX is disabled by MXCH register, the data transfer from SMIX to I2S is not stopped immediately. Therefore, the following register must not be changed after setting it at the initialization.

- MXOCTRL

And, TFTH of I2Sn_INTCNT must not be changed under the following condition.

- START of I2Sn_OPRREG is set to 1, and
- TXENB of I2Sn_OPRREG is set to 1, and
- TXDMACT of I2Sn_DMAACT is set to 1

4.4. Interface between SWFG and SMIX

SWFG is connected to SMIX by a dedicated bus interface. On this interface, the sound source transfer between SWFG and SMIX is triggered by a data request from SMIX.

4.4.1. Configuration

Sound sources from SWFG need to be input to MXWFGnDADR register of SMIX. Therefore, the address of MXWFGnDADR register should be set to WGCHADD1, WGCHADD2 and WGCHADD3. The actual setting values are the followings.

- WGCHADD1 : 0x01040100
- WGCHADD2 : 0x010C0108
- WGCHADD3 : 0x00000110

4.4.2. Limitation

Because the sound sources are transferred surely from SWFG to SMIX, the following registers must not be changed during the sound source translation.

Even if SWFG is disabled by WGCHEN register, the data transfer from SWFG to SMIX is not stopped immediately. Therefore, the following register must not be changed after setting it at the initialization.

- WGCHADD1
- WGCHADD2
- WGCHADD3

4.5. Interface between DMAC and SMIX

DMAC can be used to transfer the sound source to PMIS interface of SMIX.

4.5.1. Configuration

PMISn Request assert threshold setting of SMIX (FESTCHn of MXDRQCTRL register) must be same as the number of block count of DMAC.

4.5.2. Limitation

Because the sound sources are transferred surely from DMAC to SMIX, the following registers must not be changed during the sound source translation (from data request by SMIX to data transfer completion of DMAC).

- PMISnMONO of MXCHMONO

And, DMAENCHn of MXDREQCTRL must be 0 during the following procedures.

- Clear PMIS Input Buffer of the channel (Write 1 to PMOSnBCLR of MXBUFFCLR)
- Enable the channel by MXCH register (change from 0 to 1)
- Change FESTCHn of MXDREQCTRL of the channel

4.6. Interface between CPU and SMIX

CPU can be used to transfer the sound source to PMIS interface of SMIX. Data transfer request interrupt of SMIX can be used as trigger for the data transfer.

No special configuration or limitation to be considered.

5. Note

- Every output module doesn’t support any transfer requests for CPU and DMA. The transfer request only supports for the sound source transmission of SMIX. That is, CPU and DMA cannot directly transfer sound sources to the output modules without going through SMIX.

CHAPTER 39: Expand PLL

This chapter explains the functions and operations of the Expand PLL.

---

1. Overview
2. Configuration and Block Diagram
3. Operation of the Expand PLL
4. Setting Procedure
5. Registers

1. Overview

This section explains the overview of the Expand PLL.

Total 3 Expand PLLs are embedded and generates PLL clocks for Sound System, Inter-IC Sound, DDR High Speed SPI.

Each Expand PLLx has 3 divider circuits to output CLK_CDx and CLK_CDAx and CLK_CDBx. (x=0 to 2)

Source clock for them is only Main clock.

Table 1-1 Overview of Expand PLL Clock Instances

Expand PLL	PLL Divider	The destination system
Expand PLL0	Expand_CLK_CD0	DDR High Speed SPI
	Expand_CLK_CDA0	Unused
	Expand_CLK_CDB0	Unused
Expand PLL1	Expand_CLK_CD1	Sound System
	Expand_CLK_CDA1	Sound System
	Expand_CLK_CDB1	Sound System
Expand PLL2	Expand_CLK_CD2	Inter-IC Sound
	Expand_CLK_CDA2	Unused
	Expand_CLK_CDB2	Unused

Expand clock supervisors (Expand CSVx) are equipped to monitor and supervise the Expand PLLx. (x=0 to 2)

Table 1-2 Overview of Clock Supervisor Instances

Clock supervisor	Monitor Clock (before divider)	Reference Clock	Clock Supervisor Type
Expand CSV0	Expand PLL0	Main clock	Detector for stopped clock or abnormal frequency of the monitored clock
Expand CSV1	Expand PLL1	Main clock	
Expand CSV2	Expand PLL2	Main clock	

Features

- All Expand PLL has stabilization counters, which mask the clocks till they are stable.
- Expand PLL multiplication rate can be set freely.
- Clock Supervisor circuits are present for all Expand PLLs to detect if Expand PLL goes out of lock due to high jitter on the input clock.

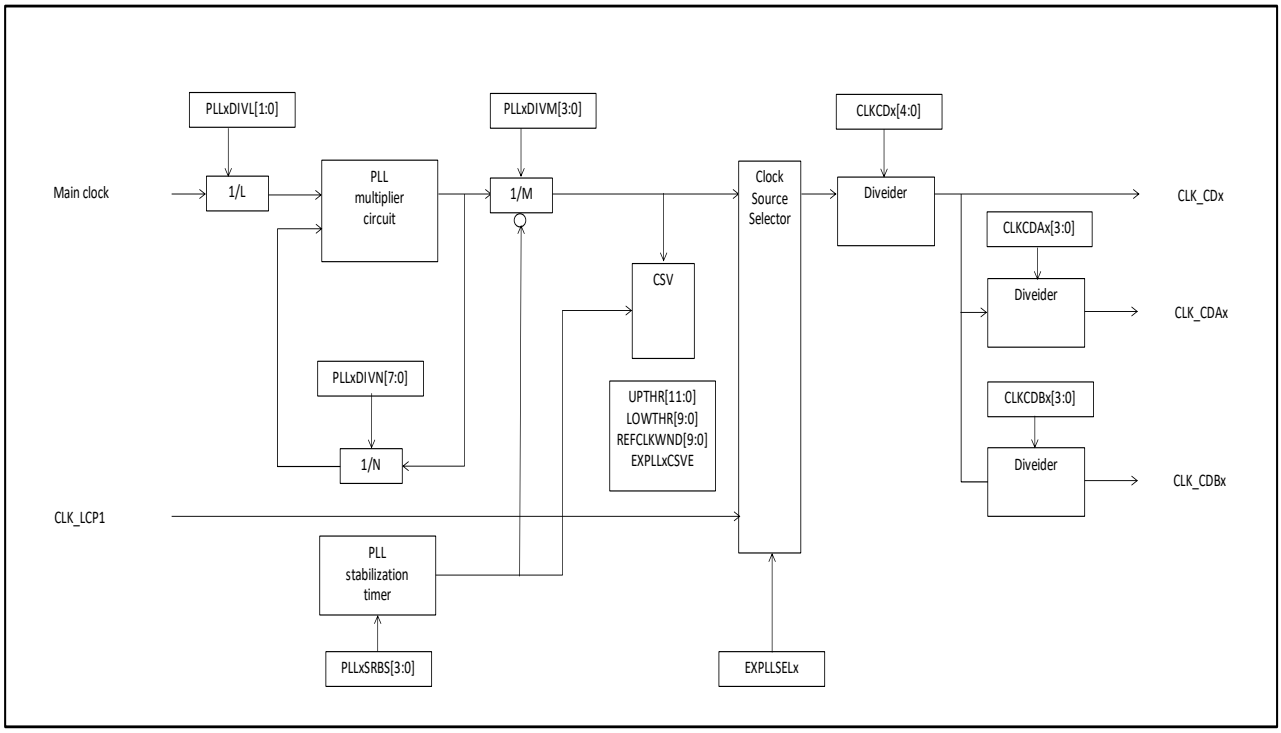
Note:

- *Expand PLL cannot be used under PSS mode. You need to stop expand PLL before transition to PSS mode. And after returning to RUN mode, you also need to configure the expand PLL following the 4.Setting Procedure.*

2. Configuration and Block Diagram

This section explains the block diagram of the Expand PLL.

Figure 2-2-1 Expand PLLx Block Diagram



3. Operation of the Expand PLL

This section shows the configuration of the Expand PLL.

3.1. Expand PLL Clock

Expand PLL Clock is a clock that is generate by using a PLL from the Main clock.

The Expand PLL has several PLL multiplier circuits is used to generate the PLL clock out of Main clock.

3.2. Clock Source Selector

Expand_CLK_CDx (Expand_CLK_CDAx, Expand_CLKCD_Bx) can select clock source.

After initialization, the default clock source of Expand PLL is CLK_LP1.

3.3. Clock Divider

There are 3 dividers on each PLL clock path.

See the Clock Configuration chapter about a relationship between output clock and destination system.

3.4. Clock Stabilization Time

The lock up time of Expand PLL is 100us.

Stabilization time settings should be set the value that satisfies the following formula.

Main clock period[s] × EXPLLxCNTR.PLLxSTABS setting cycle > 100-6[s]

<For example>

Main clock period[s] = 250-9[s]

PLLxSTABS = 1000(29 [cycle])

250-9[s] ×29 [cycle] = 128-6[s] > 100-6[s]

3.5. Expand PLL Clock Supervisor

This section explains Expand PLL clock supervisor.

- Expand PLL clock supervisor configuration register 0 (EXCSVPLLxCFGR0) and Expand PLL clock supervisor configuration register 1 (EXCSVPLLxCFGR1) are used for settings
- Expand PLL clock supervisor generate NMI when PLL clock detects abnormal state.

To get NMI information of Expand PLL, see the 5.8 EXCSVERR register.

4. Setting Procedure

This section shows the Expand PLL setting procedure.

Figure 4-1 shows Expand PLL clock setting procedure, and Figure 4-2 shows Expand PLL clock disabled Procedures.

Take the procedure of Figure 4-1 and Figure 4-2.

It may be in following conditions when a setting procedure is not kept.

- Intended PLL clock is not output.
- Clock supervisor detects NMI when PLL clock exceed a range set by EXCSVPLLxCF0.

Figure 4-1 Expand PLL Clock Setting Procedures

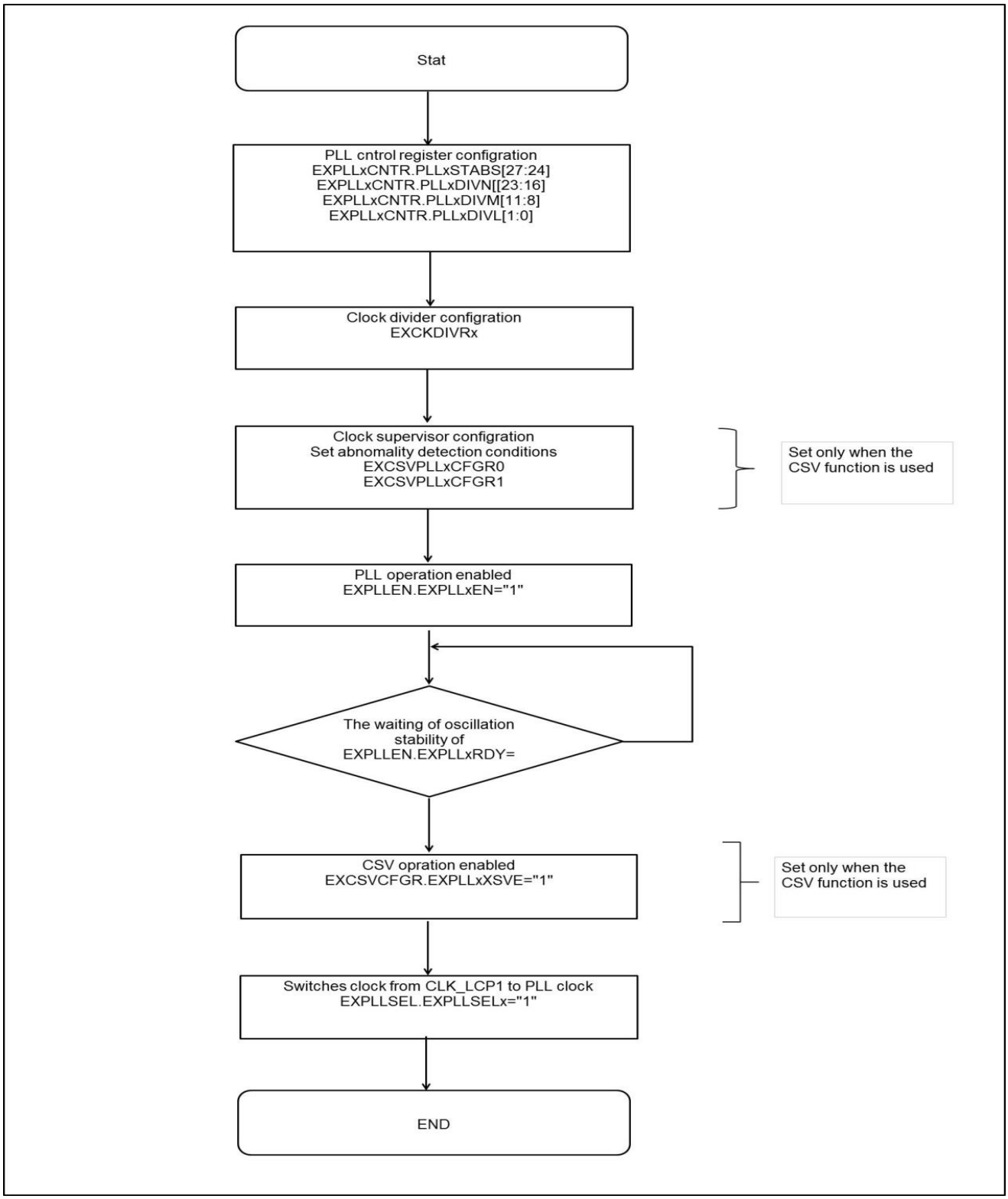
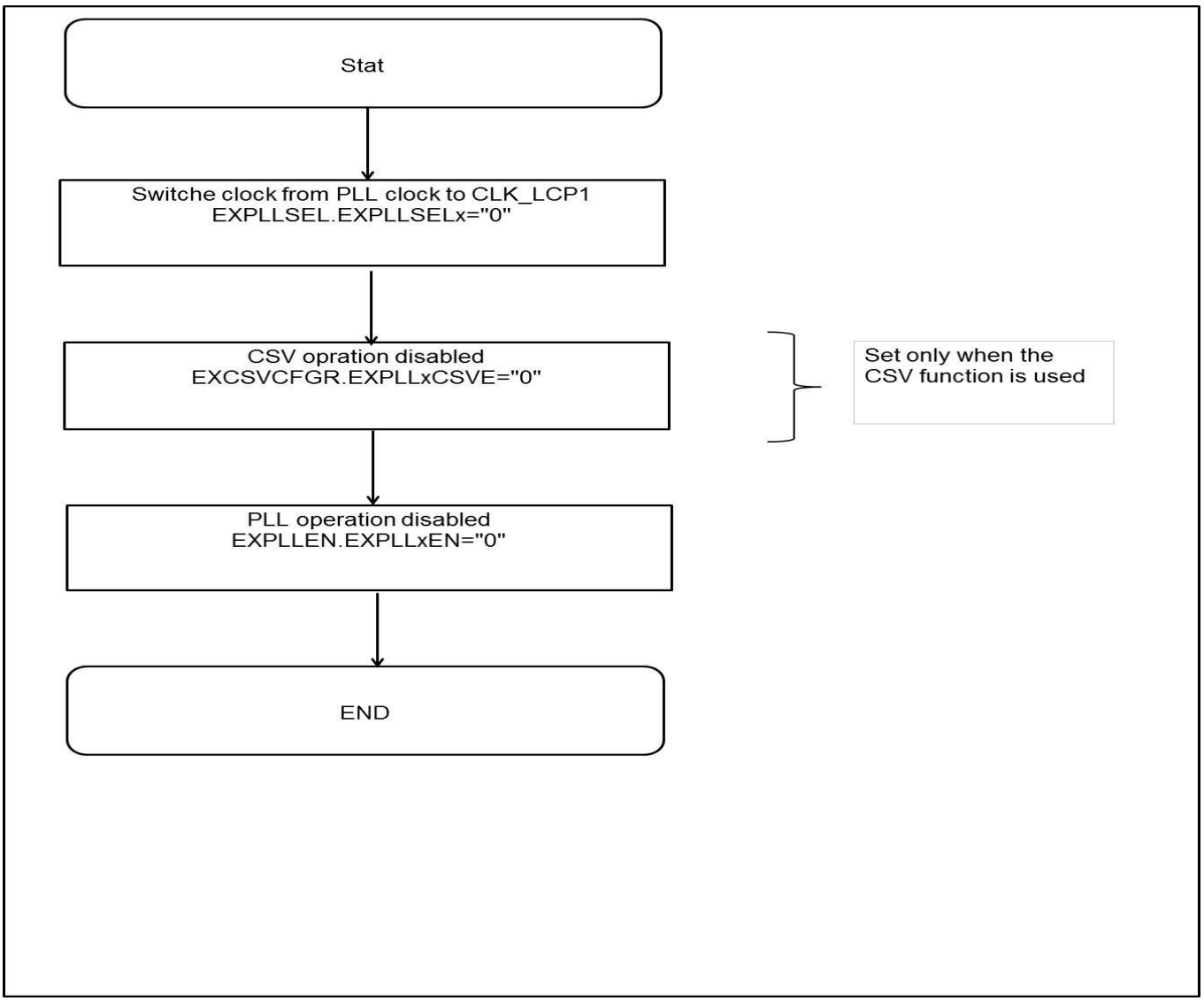


Figure 4-2 Expand PLL Clock Disabled Procedures



5. Registers

This section explains the CSV registers and PLL registers.

Table 5-1 CSV Registers

Abbreviated Register Name	Register Name	Reference
EXCSVPLL0CFGR0	Expand PLL0 clock supervisor setting register 0	5.1
EXCSVPLL0CFGR1	Expand PLL0 clock supervisor setting register 1	5.2
EXCSVPLL1CFGR0	Expand PLL1 clock supervisor setting register 0	5.1
EXCSVPLL1CFGR1	Expand PLL1 clock supervisor setting register 1	5.2
EXCSVPLL2CFGR0	Expand PLL2 clock supervisor setting register 0	5.1
EXCSVPLL2CFGR1	Expand PLL2 clock supervisor setting register 1	5.2
EXCSVCFGR	Expand clock supervisor setting register	5.3
EXCSVERR	Expand clock supervisor error register	5.4
EXCSVERRCLR	Expand clock supervisor error register clear register	5.5

Table 5-2 PLL Registers

Abbreviated Register Name	Register Name	Reference
EXPLL0CNTR	Expand PLL 0control register	5.6
EXPLL1CNTR	Expand PLL 1control register	5.6
EXPLL2CNTR	Expand PLL 2control register	5.6
EXPLLEN	Expand PLL EN register	5.7
EXCKDIVR0	Expand Clock Divider Register0	5.8
EXCKDIVR1	Expand Clock Divider Register1	5.8
EXCKDIVR2	Expand Clock Divider Register2	5.8
EXPLLSEL	Expand PLL Select Register	5.9

.

5.1. Expand PLLx Clock Supervisor Setting Register 0 (EXCSVPLLxCFGR0) (x=0 to 2)

REGISTER_NAME	Expand PLLx Clock Supervisor Setting Register 0 (EXCSVPLLxCFGR0)
OFFSET	0x00(x=0) 0x08(x=1) 0x10(x=2)
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

Expand PLLx clock supervisor setting register 0 (EXCSVPLLxCFGR0) sets the upper-limit threshold value and lower-limit threshold value of a frequency range.

Bit	31	30	29	28	27	26	25	24
Field	Reserved				UPTHR[11:8]			
R/W Attribute	R0,WX				R/W			
Protection Attribute								
Initial Value	0000				0000			

Bit	23	22	21	20	19	18	17	16
Field	UPTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved						LOWTHR[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute								
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	LOWTHR[7:0]							
R/W Attribute	R/W							
Protection Attribute								
Initial Value	00000000							

[bit31:28] Reserved: Reserved bits

[bit27:16] UPTHR[11:0]: Upper-limit threshold value bits

These bits set the clock upper-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value exceeds the upper-limit threshold value, the state is judged as abnormal.

[bit15:10] Reserved: Reserved bits

[bit9:0] LOWTHR[9:0]: Lower-limit threshold value bits

These bits set the clock lower-limit threshold value for comparison with the monitored clock count value. If the monitored clock counter value falls below the lower-limit threshold value, the state is judged as abnormal.

Note:

- *This register must not be modified during operation of the PLL clock supervisor (EXCSVCFGR.EXPLLxCSVE=1). If the setting has been executed, the setting is invalid*

5.2. Expand PLL0xClock Supervisor Setting Register 1 (EXCSVPLLxCFGR1) (X=0 to 2)

REGISTER_NAME	Expand PLL0xClock Supervisor Setting Register 1 (EXCSVPLLxCFGR1)
OFFSET	0x04(x=0) 0x0C(x=1) 0x14(x=2)
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

Expand PLLx clock supervisor setting register 1 (EXCSVPLLxCFGR1) sets a reference clock count period.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved						REFCLKWND[9:8]	
R/W Attribute	R0,WX						R/W	
Protection Attribute								
Initial Value	000000						00	

Bit	7	6	5	4	3	2	1	0
Field	REFCLKWND[7:0]							
R/W Attribute	R/W							
Protection Attribute								
Initial Value	00000000							

[bit31:10] Reserved: Reserved bits

[bit9:0] REFCLKWND[9:0]: Reference clock count period bits

These bits set the reference clock count value used to trigger a comparison by the frequency range comparator.

Note:

- *Same as 5.1*

5.3. Expand Clock Supervisor Setting Register (EXCSVCFGR)

REGISTER_NAME	Expand Clock Supervisor Setting Register (EXCSVCFGR)
OFFSET	0x18
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXCSVCFGR register sets whether to enable or disable operation of each clock supervisor.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					EXPLL2 CSVE	EXPLL1 CSVE	EXPLL0 CSVE
R/W Attribute	R0,WX					R/W	R/W	R/W
Protection Attribute								
Initial Value	00000					0	0	0

Bit	7	6	5	4	3	2	1	0
Field								
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

[bit31:11] Reserved: Reserved bits

[bit10:8] EXPLLxCSVE: EXPLL2-0 clock supervisor enable bit

This bit sets whether to enable or disable the PLLx clock supervisor.

Bit	Description
0	Disable the clock supervisor.
1	Enable the clock supervisor.

[bit7:0] Reserved: Reserved bits

5.4. Expand Clock Supervisor Error Register (EXCSVERR)

REGISTER_NAME	Expand Clock Supervisor Error Register (EXCSVERR)
OFFSET	0x1C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXCSVERR register retains the factors of Expand clock supervisor interrupt requests.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					EXPLL2IF	EXPLL1IF	EXPLL0IF
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute								
Initial Value	00000					0	0	0

Bit	7	6	5	4	3	2	1	0
Field								
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

[bit31:11] Reserved: Reserved bits

[bit10:8] EXPLLxIF: PLLx abnormality detection error NMI request bit

This bit retains the fact that a NMI request was generated for a EXPLLx abnormality detection error.

Bit	Description
0	No NMI request is detected.
1	A NMI request is detected.

[bit7:0] Reserved: Reserved bits

5.5. Expand Clock Supervisor Error Clear Register (EXCSVERRCLR)

REGISTER_NAME	Expand Clock Supervisor Error Clear Register (EXCSVERRCLR)
OFFSET	0x20
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXCSVERRCLR register clears Expand clock supervisor (interrupt) requests.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					EXPLL2 ICLR	EXPLL1 ICLR	EXPLL0 ICLR
R/W Attribute	R0,WX					R0,W	R0,W	R0,W
Protection Attribute								
Initial Value	00000					0	0	0

Bit	7	6	5	4	3	2	1	0
Field								
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

[bit31:11] Reserved: Reserved bits

[bit10:8] EXPLLxICLR: PLLx abnormality detection error interrupt factor clear bit(x=0-2)

This bit clears the interrupt factor of a EXPLLx abnormality detection error.

EXPLLxICLR	Description
During write operation	When "0" is written: Invalid When "1" is written: Clear the interrupt factor.
During read operation	The read value is always "0".

[bit7:0] Reserved: Reserved bits

5.6. Expand PLLx Control Register (EXPLLxCNTR) (x=0 to 2)

REGISTER_NAME	Expand PLLx Control Register (EXPLLxCNTR)
OFFSET	0x40(x=0) 0x44(x=1) 0x48(x=2)
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXPLLxCNTR register sets the division, multiplication rate, etc. for PLLx

Bit	31	30	29	28	27	26	25	24
Field	Reserved				PLLxSTABS[3:0]			
R/W Attribute	R0,WX				R1,WX	R/W		
Protection Attribute								
Initial Value	0	0001000						

Bit	23	22	21	20	19	18	17	16
Field	PLLxDIVN							
R/W Attribute	R/W							
Protection Attribute								
Initial Value	00001101							

Bit	15	14	13	12	11	10	9	8
Field	Reserved				PLLxDIVM			
R/W Attribute	R0,WX				R/W			
Protection Attribute								
Initial Value	0000				0001			

Bit	7	6	5	4	3	2	1	0
Field	Reserved						PLLxDIVL	
R/W Attribute	R0,WX						R/W	
Protection Attribute								
Initial Value	000000						00	

[bit31:28] Reserved: Reserved bits.

[bit27:24]PLLxSTABS[3:0]

These bits select the stabilization time for PLLx clock.

Bits	Description
1000	Stabilization time : Main clock period[s] * 2^9 [cycle]
1001	Stabilization time : Main clock period[s] * 2^10 [cycle]
1010	Stabilization time : Main clock period[s] * 2^11 [cycle]
1011	Stabilization time : Main clock period[s] * 2^12 [cycle]
1100	Stabilization time : Main clock period[s] * 2^13 [cycle]
1101	Stabilization time : Main clock period[s] * 2^14 [cycle]
1110	Stabilization time : Main clock period[s] * 2^15 [cycle]
1111	Stabilization time : Main clock period[s] * 2^16 [cycle]

Reading PLLxSTABS[3] bit always returns '1'

[bit23:16] PLLxDIVN: PLLx clock N-multiplier setting bits

These bits set the multiplication rate of the PLLx clock.

Bits	Description
0000_0000	Setting prohibited
...	...
0000_1011	Setting prohibited
0000_1100	Multiply by 12
0000_1101	Multiply by 13
...	...
0111_0110	Multiply by 118
0111_0111	Multiply by 119
0111_1000	Multiply by 120
0111_1001	Setting prohibited
...	...
1100_1111	Setting prohibited

[bit15:12] Reserved: Reserved bits

[bit11:8] PLLxDIVM: PLLx clock M-divider setting bits

These bits set the division ratio for PLLx clock output.

Bits	Description
000x	Divided by 2
0010	Divided by 4
0011	Divided by 6
0100	Divided by 8
0101	Divided by 10
0110	Divided by 12
0111	Divided by 14
1000	Divided by 16
1001	Divided by 18
1010	Divided by 20
1011	Divided by 22
1010	Divided by 24
1101	Divided by 26
1110	Divided by 28
1111	Divided by 30

[bit7:2] Reserved: Reserved bits

[bit1:0] PLLxDIVL: PLLx input clock divider setting bits

These bits set the division ratio of the PLLx input clock.

Bits	Description
00	No division
01	Divided by 2
10	Divided by 4
11	Divided by 6

Note:

- *This register must not be modified during operation of the PLL clock (EXPLLEN.EXPLLxEN=1). If the setting has been executed, the setting is invalid.*

5.7. Expand PLL Enable Register (EXPLLEN)

REGISTER_NAME	Expand PLL Enable Register (EXPLLEN)
OFFSET	0x4C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXPLLEN register sets whether to enable or disable operation of each PLL.

Bit	31	30	29	28	27	26	25	24
Field	Reserved					EXPLL2 RDY	EXPLL1 RDY	EXPLL0 RDY
R/W Attribute	R0,WX					R,WX	R,WX	R,WX
Protection Attribute								
Initial Value	00000					0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved					EXPLL2 EN	EXPLL1 EN	EXPLL0 EN
R/W Attribute	R0,WX					R/W	R/W	R/W
Protection Attribute								
Initial Value	00000					0	0	0

Bit	7	6	5	4	3	2	1	0
Field								
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

[bit31:27] Reserved: Reserved bits

[bit26:24] EXPLLxRDY : EXPLLx clock oscillation stabilization bit

This bit indicates the status of the EXPLLx clock oscillation.

Bit	Description
0	Stabilization wait or oscillation stop state
1	Stable state

[bit23:11] Reserved: Reserved bits

[bit10:8] EXPLLEN: EXPLLx clock enable bit

This bit controls oscillation of the PLLx clock.

Bit	Description
0	Disable oscillation of the PLLx clock.
1	Enable oscillation of the PLLx clock.

[bit7:0] Reserved: Reserved bits

5.8. Expand Clock Divider Register x (EXCKDIVRx)(x=0 to 2)

REGISTER_NAME	Expand Clock Divider Register x (EXCKDIVRx) (x=0 to 2)
OFFSET	0x50(x=0) 0x54(x=1) 0x58(x=2)
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The EXCKDIVRx register sets the division ratio of each internal operating clock.

Bit	31	30	29	28	27	26	25	24
Field	Reserved			Expand_CLKCDx				
R/W Attribute	R0,WX			R/W				
Protection Attribute								
Initial Value	000			00000				

Bit	23	22	21	20	19	18	17	16
Field	Reserved				Expand_CLKCDxA			
R/W Attribute	R0,WX				R/W			
Protection Attribute								
Initial Value	0000				0000			

Bit	15	14	13	12	11	10	9	8
Field	Reserved				Expand_CLKCDxB			
R/W Attribute	R0,WX				R/W			
Protection Attribute								
Initial Value	0000				0000			

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

[bit31:29] Reserved: Reserved bits

[bit28:24] Expand_CLKCDx: Expand_CLKCDx clock divider setting bits

These bits set the division ratio of the Expand_CLKCDx.

Bits	Description
00000	No division
00001	Divided by 2
00010	Divided by 3
00011	Divided by 4
...	...
11101	Divided by 30
11110	Divided by 31
11111	Divided by 32

[bit23:20] Reserved: Reserved bits

[bit19:16] Expand_CLKCDxA: Expand_CLKCDxA clock divider setting bits

These bits set the division ratio of the Expand_CLKCDxA.

Bits	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit15:12] Reserved: Reserved bits

[bit11:8] Expand_CLKCDxB: Expand_CLKCDxB clock divider setting bits

These bits set the division ratio of the Expand_CLKCDxB.

Bits	Description
0000	No division
0001	Divided by 2
0010	Divided by 3
0011	Divided by 4
...	...
1101	Divided by 14
1110	Divided by 15
1111	Divided by 16

[bit7:0] Reserved: Reserved bits

Note:

- *This register must not be modified during operation of the PLL clock (EXPLLEN.EXPLLxEN=1). If the setting has been executed, the setting is invalid.*

5.9. Expand PLL Select Register (EXPLLSEL)

REGISTER_NAME	Expand PLL Select Register (EXPLLSEL)
OFFSET	0x5C
ACCESS_SIZE	B H W
MULTIPLE	
NUMERIC_TYPE	
OTHER	

The PLLSEL register sets whether to CLK_LCP1 or PLL clock operation of output clock.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
R/W Attribute	R0,WX							
Protection Attribute								
Initial Value	00000000							

Bit	7	6	5	4	3	2	1	0
Field						EXPLL SEL2	EXPLL SEL1	EXPLL SEL0
R/W Attribute	R0,WX					R/W	R/W	R/W
Protection Attribute								
Initial Value	0000000					0	0	0

[bit31:3] Reserved: Reserved bits

[bit2] EXPLLSEL2: PLL clock select bit

This bit controls output clock.

Bit	Description
0	CLK_LCP1.
1	Expand_PLL Clock

[bit1] EXPLLSEL1: PLL clock select bit

This bit controls output clock.

Bit	Description
0	CLK_LCP1.
1	Expand_PLL Clock

[bit0] EXPLLSEL0: PLL clock select bit

This bit controls output clock.

Bit	Description
0	CLK_LCP1.
1	Expand_PLL Clock

Note:

- *This register must not be set to “1” when EXPLLEN. EXPLL2RDY = “0”. If the setting has been executed, the setting is invalid.*

CHAPTER 40: LCD Subsystem

This chapter explains the LCD Subsystem.

1. Overview
2. Configuration and Block Diagram
3. Operation

1. Overview

1.1. Scope

LCD Subsystem is defined as a subsystem that has the following functions.

- LCD Bus Interface
- DDR High Speed SPI controller
- External Bus Interface

The detailed description for the module is not included in this chapter. Functionalities, operations, restrictions and notes for the module should be referred in prepared each chapter.

1.2. Protections

The register interfaces of modules have PPU protection function to prevent a master from accessing them illegally. The PPU numbers are defined and described in memory map.

LCD Bus Interface can be an AXI master. It also has MPU, and the master access to some peripherals is protected by it.

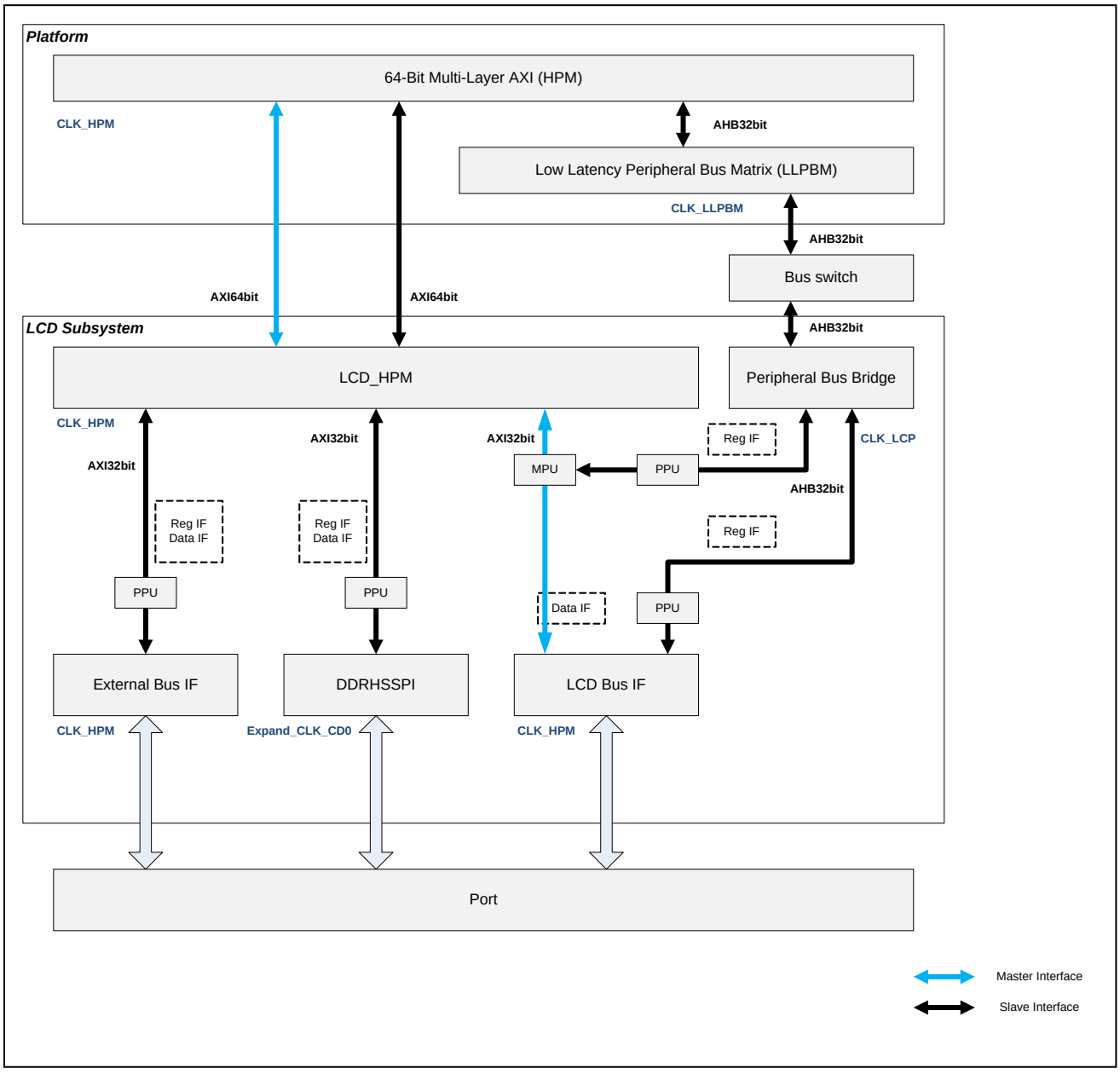
Note:

- *The detailed description of PPU should be referred in the chapter of Platform Hardware Manual.*
- *The detailed description of MPU should be referred in the chapter of Product Hardware Manual.*

2. Configuration and Block Diagram

This section shows a block diagram of LCD Subsystem.

Figure 2-1 LCD Subsystem Block Diagram



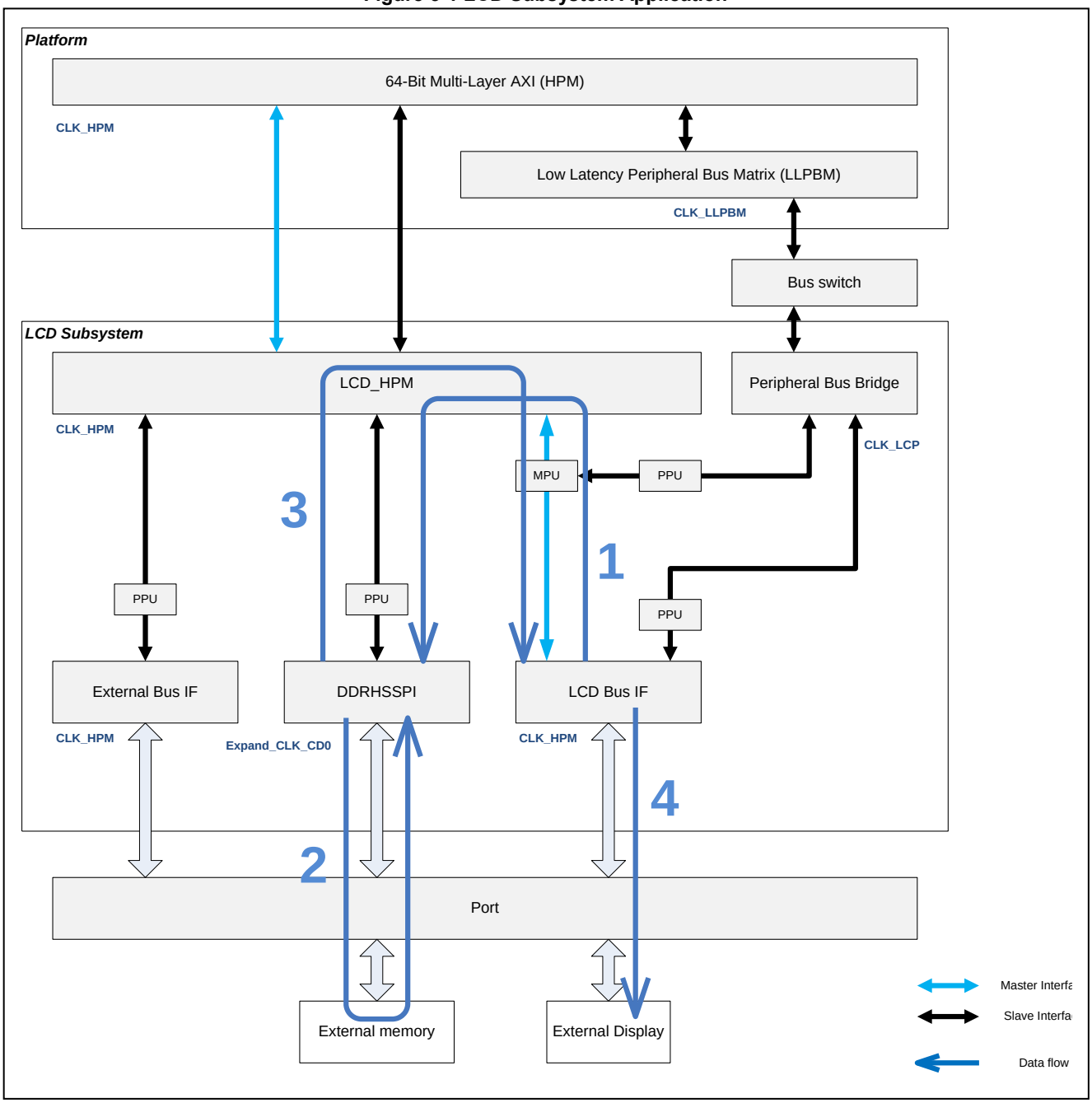
3. Operation

This section describes the assumed application of LCD Subsystem.

3.1. Application

Figure 3-1 describes a data flow on an assumed application of LCD subsystem.

Figure 3-1 LCD Subsystem Application



Notes:

- *1: Data read request from master interface of LCD Bus IF to DDRHSSPI.*
- *2: DDRHSSPI read the data from the external memory by read request of LCD Bus IF.*
- *3:.DDRHSSPI returns the data to LCD Bus IF.*
- *4: LCD Bus IF transfer the read data to an external display.*

3.1.1. Performance

The following performance is case of the RGB666 (18bit) LCD bus updating the LCD display module with 480 x 240 at 60 Hz refresh rate.

480 x 240 x 4byte x 60 Hz = 221,184,000 bit/s = 27.7 Mbyte/s

The following setting shows the conditions for ensuring the above transfer bandwidth.

CLK setting:

- CLK_HPM = 33 MHz

DDRHSSPI setting:

- DDR mode
- Dual Quad protocol
- Read access by command sequencer mode.
- Expand_CLK_CD0 = 140 MHz (provided from Expand PLL.)

LCD Bus IF setting:

The following table shows the transfer bandwidth variation according to the master transaction type of LCD Bus IF.

Figure 3-2 Transfer Rate table

External DDRHSSPI Memory	CLK_HSSPI	LCD Bus IF AXI Transaction	Transfer Rate
DDRHSSPI memory	140MHz	16 Byte (1 transaction)	23 MByte/s (*1)
		64 Byte (1 transaction)	45 MByte/s (*2)
		32 Byte x 4 transaction = 128 Byte (Burst : INCR)	51 MByte/s (*3)

Set *2 or *3 transaction type, because *1 cannot ensure transfer bandwidth.

Notes:

- *The detailed description of LCD Bus IF should be referred in the chapter LCD BUS INTERFACE.*
- *The detailed description of DDRHSSPI should be referred in the chapter DDRHSSPI.*

CHAPTER 41: APPENDIX: Major Changes

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td colspan="3">Rev.*A</td></tr>
<tr><td>19</td><td>CHAPTER 1: Overview
1.Overview</td><td>Revised the below:
Error)
S6J3360/S6J3370 is a microcontroller series which is to be applied to automotive systems representative of a body control unit.

Correct)
S6J3360/S6J3370 is a microcontroller series which is to be applied to automotive systems representative of a graphical cluster control unit on a dashboard.</td></tr>
<tr><td>20</td><td>CHAPTER 1: Overview
2.Document definition</td><td>Revised the below:
Table 2-1
Error)
<table><tr><th>Document type</th><th>Definition</th><th>Primary user</th><th>Document code</th></tr><tr><td>Datasheet</td><td>The function and its characteristics are specified quantitatively.</td><td>Investigator and hardware engineer</td><td>002-03359</td></tr><tr><td>S6J3360/S6J3370 hardware manual</td><td>The function and its operation of S6J3360/S6J3370 series are described.</td><td>Software engineer</td><td>002-15503</td></tr></table>
Correct)
<table><tr><th>Document type</th><th>Definition</th><th>Primary user</th><th>Document code</th></tr><tr><td>Datasheet</td><td>The function and its characteristics are specified quantitatively.</td><td>Investigator and hardware engineer</td><td>002-03359</td></tr><tr><td>S6J3360/S6J3370 hardware manual</td><td>The function and its operation of S6J3360/S6J3370 series are described.</td><td>Software engineer</td><td>002-18302</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>26</td><td>CHAPTER 2: Overview
1. Function list</td><td>Revised the below:

Error)
The table shows the functions which are implemented in S6H3360/S6J3370 series.
Table 1-1
<table>
<tr><th>Function</th><th>S6J33xxJxx</th><th>S6J33xxHxx</th><th>S6J33xxGxx</th><th>S6J33xxFxx</th><th>Remarks</th></tr>
<tr><td>CPU core</td><td colspan="4">ARM Cortex-R5F</td><td></td></tr>
</table>
Correct)
The table shows the functions which are implemented in S6J3360/S6J3370 series.
Table 1-1
<table>
<tr><th>Function</th><th>S6J33xxJxx
176-pin</th><th>S6J33xxHxx
144-pin</th><th>S6J33xxGxx
120-pin</th><th>S6J33xxFxx
100-pin</th><th>Remarks</th></tr>
<tr><td>CPU core</td><td colspan="4">ARM Cortex-R5F</td><td></td></tr>
</table></td></tr>
<tr><td>26</td><td>CHAPTER 2: Overview
1. Function list</td><td>Revised the below:
Table 1-1
Error)
<table>
<tr><td>PLL</td><td>PLL0,1,2,3</td><td></td></tr>
</table>
<table>
<tr><td>System-RAM</td><td>Option</td><td>See 2.1</td></tr>
<tr><td>Backup area in System-RAM</td><td>Option</td><td>See 2.1</td></tr>
</table>
Correct)
<table>
<tr><td>PLL</td><td>PLL0, Expand PLL0,1,2</td><td></td></tr>
</table>
<table>
<tr><td>System-RAM</td><td>128KB (include Backup area)</td><td>See 2.1</td></tr>
<tr><td>Backup area in System-RAM</td><td>16+8KB</td><td>See 2.1</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
27	CHAPTER 2: Overview 1. Function list	Revised the below: Table 1-1 Error)

32-bit reload timer	6 ch	6 ch	6 ch (Input: 4 ch/ Output: 4 ch)	6 ch (Input: 2 ch/ Output: 2 ch)	

Base timer	16 units (32 ch)				
Free-run timer	8 ch (Input: 8 ch)			8 ch (Input: 0 ch)	
Input Capture Unit	12 ch		9 ch	4 ch	
Output Compare Unit	12 ch (Output: 12 ch)		12 ch (Output: 8/7 ch)	12 ch (Output: 5/4 ch)	
Stepping motor controller (SMC)	6 gauges		4 gauges	3 gauges	
12bit-A/D converter	1 unit x 48 input ports	1 unit x 40 input ports	1 unit x 29 input ports	1 unit x 17 input ports	
Partial Wake Up	analog input 8 ch Trigger 1 ch		analog input 5 ch Trigger 1 ch	analog input 1 ch Trigger 0 ch	

I2S	2 ch	1ch	One only supports an output as a function of the sound system.

CAN FD RAM (ECC supported)	16 KB/ch It is equivalent to a 192 message buffer per channel of the CAN module	
Multi-function serial interface		

Correct)

32-bit reload timer	6 ch (Input: 6pin/ Output: 6pin)	6 ch (Input: 6pin/ Output: 6pin)	6 ch (Input: 4 pin/ Output: 4 pin)	6 ch (Input: 2 pin/ Output: 2 pin)	

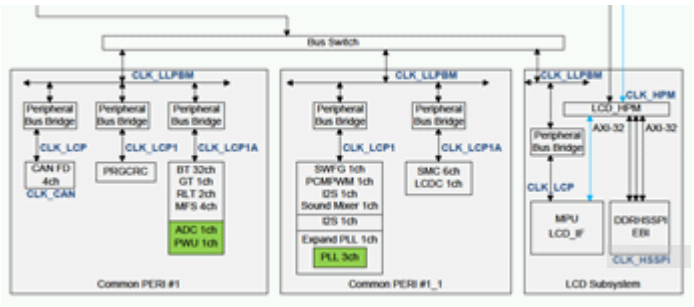
Base timer	16 units (32 ch)		15 units (30 ch)	12 unit (25ch)	
Free-run timer	8 ch (Input: 8 pin)			8 ch (Input: 0 pin)	
Input Capture Unit	12 ch		9 ch	4 ch	
Output Compare Unit	12 ch (Output: 12 pin)		12 ch (Output: 8 pin)	12 ch (Output: 5 pin)	
Stepping motor controller (SMC)	6 gauges		4 gauges	3 gauges	
12bit-A/D converter	48 ch	40 ch	29 ch	17 ch	1unit
Partial Wake Up	analog input 8 ch Trigger 1 ch		analog input 5 ch Trigger 1 ch	-	

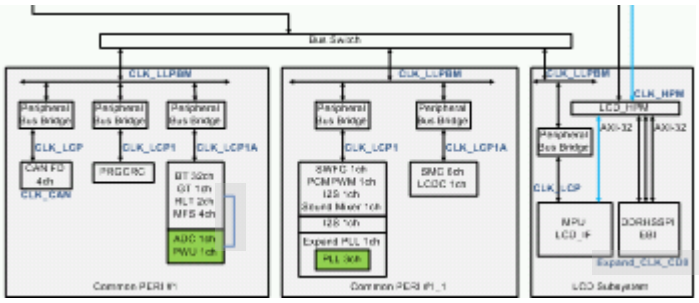
I2S	2 ch (ch0: Output only ch1: Input/ Output)	1ch (ch0: Output only)	One only supports an output as a function of the sound system.

CAN FD RAM (ECC supported)	16 KB/ch It is equivalent to a 128-message buffer per channel of the CAN module				
Multi-function serial interface	12 ch	12 ch	8 ch	5 ch	
I2C	10 ch (Fast mode support : 2 ch)	10 ch (Fast mode support : 2 ch)	6 ch (Fast mode support : 0 ch)	4 ch (Fast mode support : 0 ch)	

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>37</td><td>CHAPTER 3: Product Description
2. Product description</td><td>Revised the below:
Table 2-1
Error)
<table><tr><td>A/D Converter</td><td>12bit resolution, 2 unit
64 channels of analog input for TEQFP176 and LQFP176
56 channels of analog input for TEQFP144 and LQFP144
45 channel of analog input for TEQFP120 and LQFP120
35 channel of analog input for TEQFP100 and LQFP100
External trigger and timer trigger are available.
The description of the A/D converter function should be referred in the S6J3360/S6J3370 hardware manual.</td></tr></table>
Correct)
<table><tr><td>A/D Converter</td><td>12bit resolution, 1 unit
48 channels of analog input for TEQFP176 and LQFP176
40 channels of analog input for TEQFP144 and LQFP144
29 channel of analog input for TEQFP120 and LQFP120
17 channel of analog input for TEQFP100 and LQFP100
External trigger and timer trigger are available.
ch.0-63 of logical channels are supported.
For ch.0-47, HWTRG can be selected from combination specified on PORT CONFIGURATION.
For ch.48-63, HWTRG can be assigned on P1_09 (ADC_TRG0_0) only. Relocation or resource combination cannot be selected.
The description of the A/D converter function should be referred in the S6J3360/S6J3370 hardware manual.</td></tr></table></td></tr>
<tr><td>42</td><td>CHAPTER 3: Product Description
3.4. Restriction</td><td>Revised the below:
Error)

Correct)
<table><tr><td>PPU support of DDRHSSPI</td><td>PPU0_SR.VD,
PPU0_SR.VP,
PPU0_SR.VW,
PPU0_SR.VL[4:0]</td><td>This series has a product specific LCD subsystem and DDRHSSPI. PPU protects the DDRHSSPI against an illegal access, and then only responds bus error.
Please note the status flag bits which are served as Traveo platform function does not support the product specific PPU and DDRHSSPI, that is, the status bits on PPU0_SR are no set when the PPU detects the illegal access.</td><td>-</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>44</td><td>CHAPTER 4:
1. Block diagram</td><td>Revised the below:
Error)
 

Correct)
  </td></tr>
<tr><td>74</td><td>CHAPTER 8:
IRQ Map /
NMI Map</td><td>Revised the below:
Error)
<table>
<tr><td>194</td><td>ADC12B1 Conversion Done</td><td>IRC0_IRQPL48: IRQPL194</td><td>IRC0_IRQVA194</td></tr>
<tr><td>195</td><td>ADC12B1 Group interrupt</td><td>IRC0_IRQPL48: IRQPL195</td><td>IRC0_IRQVA195</td></tr>
<tr><td>196</td><td>ADC12B1 pulse detection function</td><td>IRC0_IRQPL49: IRQPL196</td><td>IRC0_IRQVA196</td></tr>
<tr><td>197</td><td>ADC12B1 RCO</td><td>IRC0_IRQPL49: IRQPL197</td><td>IRC0_IRQVA197</td></tr>
</table>
Correct)
<table>
<tr><td>194</td><td>ADC12B0 Conversion Done</td><td>IRC0_IRQPL48: IRQPL194</td><td>IRC0_IRQVA194</td></tr>
<tr><td>195</td><td>ADC12B0 Group interrupt</td><td>IRC0_IRQPL48: IRQPL195</td><td>IRC0_IRQVA195</td></tr>
<tr><td>196</td><td>ADC12B0 pulse detection function</td><td>IRC0_IRQPL49: IRQPL196</td><td>IRC0_IRQVA196</td></tr>
<tr><td>197</td><td>ADC12B0 RCO</td><td>IRC0_IRQPL49: IRQPL197</td><td>IRC0_IRQVA197</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>915</td><td>CHAPTER 29: Sound Generator 4. Registers</td><td>Revised the below:
Error)

Correct)
Table 4-3 Register offset address map
<table>
<tr><th rowspan="2">OFFSET_ADDRESS</th><th colspan="4">REGISTER_NAME</th><th rowspan="2">ACCESS_SIZE</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x00000000</td><td>Reserved</td><td>SGDER</td><td colspan="2">SGCR</td><td>B, H, W</td></tr>
<tr><td>0x00000004</td><td colspan="2">SGAR</td><td>SGFR</td><td>SGNR</td><td>B, H, W</td></tr>
<tr><td>0x00000008</td><td>SGTCR</td><td>SGIDR</td><td colspan="2">SGPCR</td><td>B, H, W</td></tr>
<tr><td>0x0000000C</td><td colspan="4">SGDMAR</td><td>B, H, W</td></tr>
<tr><td>0x00000010</td><td colspan="2">Reserved</td><td colspan="2">SGCCR</td><td>H, W</td></tr>
</table></td></tr>
<tr><td>933</td><td>CHAPTER 30: Sound Waveform Generator</td><td>CHAPTER 30: Sound Waveform Generator was replaced new one.</td></tr>
<tr><td>1037</td><td>CHAPTER 31: Sound Mixer</td><td>CHAPTER 31: Sound Mixer was replaced new one.</td></tr>
<tr><td>1122</td><td>CHAPTER 32: Inter-IC Sound(I2S) 1. Overview</td><td>Revised the below:
Error)
During the master mode, SCK clock can be output by dividing external clock or internal clock (it is selectable by register).

Correct)
During the master mode, SCK clock can be output by dividing external clock or internal bus clock (it is selectable by register).</td></tr>
<tr><td>1122</td><td>CHAPTER 32: Inter-IC Sound(I2S) 1. Overview</td><td>Revised the below:
Features of I2S
Error)
■ Selecting clock frequency source of serial bit clock in the master mode (internal and external clock)
■ Setting clock frequency dividing ratio in the master mode
■ Frequency of SCK = (frequency of internal clock or external clock) / (2 x I2Sn_CNTREG:CKRT[5:0])

Correct)
■ Selecting clock frequency source of serial bit clock in the master mode (internal bus clock and external clock)
■ Setting clock frequency dividing ratio in the master mode
■ Frequency of SCK = (frequency of internal bus clock or external clock) / (2 x I2Sn_CNTREG:CKRT[5:0])</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1123</td><td>CHAPTER 32: Inter-IC Sound(I2S) 2. Configuration and Block Diagram</td><td>Revised the below:
Clocking of I2S
Error)
■ The supply clock of I2S can be internal (CLK_HAPPS1B0) or external (ECLK) source.

Correct)
■ The supply clock of I2S can be internal bus clock or external (ECLK) clock source.</td></tr>
<tr><td>1146</td><td>CHAPTER 32: Inter-IC Sound(I2S) 4. Registers</td><td>Revised the below:
4.3 Control Register (I2Sn_CNTREG)
Error)
[bit31:26] CKRT : Clock Divider
This sets output clock frequency dividing ratio at master operation.
Internal clock (CLK_HAPPS1B0) is divided at I2Sn_CNTREG:ECKM = "0", and external clock (ECLK) is divided at I2Sn_CNTREG:ECKM = "1".

Correct)
[bit31:26] CKRT : Clock Divider
This sets output clock frequency dividing ratio at master operation.
Internal bus clock is divided at I2Sn_CNTREG:ECKM = "0", and external clock (ECLK) is divided at I2Sn_CNTREG:ECKM = "1".</td></tr>
<tr><td>1147</td><td>CHAPTER 32: Inter-IC Sound(I2S) 4. Registers</td><td>Revised the below:
4.3 Control Register (I2Sn_CNTREG)
Error)
[bit10] ECKM : Clock Selector
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Internal clock (CLK_HAPPS1B0)is divided and output</td></tr>
<tr><td>1</td><td>External clock (ECLK)is divided and output</td></tr>
</table>
Correct)
[bit10] ECKM : Clock Selector
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Internal bus clock is divided and output</td></tr>
<tr><td>1</td><td>External clock (ECLK)is divided and output</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1163</td><td>CHAPTER 32: Inter-IC Sound(I2S) 4. Registers</td><td>Revised the below:
4.9 Interrupt Control Register (I2Sn_INTCNT)
Error)
[bit5:4] RPTMR : Rx Completion Timer
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>00</td><td>0 (the timer is not in operation)</td></tr>
<tr><td>01</td><td>54000 CLK_HAPPS1B0 cycles</td></tr>
<tr><td>10</td><td>108000 CLK_HAPPS1B0 cycles</td></tr>
<tr><td>11</td><td>216000 CLK_HAPPS1B0 cycles</td></tr>
</table>
[bit5:4] RPTMR : Rx Completion Timer
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>00</td><td>0 (the timer is not in operation)</td></tr>
<tr><td>01</td><td>54000 internal bus clock cycles</td></tr>
<tr><td>10</td><td>108000 internal bus clock cycles</td></tr>
<tr><td>11</td><td>216000 internal bus clock cycles</td></tr>
</table></td></tr>
<tr><td>1166</td><td>CHAPTER 32: Inter-IC Sound(I2S) 4. Registers</td><td>Revised the below:
4.10 Status Register (I2Sn_STATUS)
Error)
[bit19] EOPI : Interrupt Flag for Rx Timer
The count value is automatically cleared if reception FIFO data is more than threshold or it becomes empty. When the reception FIFO is not empty and the number of data is less than or equal to the threshold, the reception timer is incremented with each CLK_HAPPS1B0 cycle.

Correct)
[bit19] EOPI : Interrupt Flag for Rx Timer
The count value is automatically cleared if reception FIFO data is more than threshold or it becomes empty. When the reception FIFO is not empty and the number of data is less than or equal to the threshold, the reception timer is incremented with each internal bus clock cycle.</td></tr>
</table>

Page	Section	Change Results
1230	CHAPTER 34: LCD Bus Interface 4.3.1.2 Interface Timing	Revised the below: Error) $$\text{SETUP} = \left(\frac{t_{AW} + t_{UNCERT}}{t_{CLK}}\right)$$ $$\text{(read) ACTIVE} = \max\left(\frac{t_{ACC} + t_{OUT2IN}}{t_{CLK}}, \frac{t_{CCLR}}{t_{CLK}}\right)$$ $$\text{(write) ACTIVE} = \max\left(\frac{t_{DS} + t_{UNCERT}}{t_{CLK}}, \frac{t_{CCLW}}{t_{CLK}}\right)$$ $$\text{(write) HOLD} = \max\left(\frac{t_{DH} + t_{UNCERT}}{t_{CLK}}, \frac{t_{CCHW}}{t_{CLK}}\right) - \text{SETUP}$$ $$\text{(read) HOLD} = \frac{t_{CCHR}}{t_{CLK}} - \text{SETUP}$$ Correct) $$\text{SETUP} = \left(\frac{t_{AW} + t_{UNCERT}}{t_{CLK}}\right)$$ $$\text{(read) ACTIVE} = \max\left(\frac{t_{ACC} + t_{OUT2IN}}{t_{CLK}}, \frac{t_{CCLR}}{t_{CLK}}\right)$$ $$\text{(write) ACTIVE} = \max\left(\frac{t_{DS} + t_{UNCERT}}{t_{CLK}}, \frac{t_{CCLW}}{t_{CLK}}\right)$$ $$\text{(write) HOLD} = \max\left(\frac{t_{AH}}{t_{CLK}}, \frac{t_{CCHW}}{t_{CLK}} - \text{SETUP}, \frac{t_{DH}}{t_{CLK}} - \text{SETUP}\right)$$ $$\text{(read) HOLD} = \max\left(\frac{t_{AH}}{t_{CLK}}, \frac{t_{CCHR}}{t_{CLK}} - \text{SETUP}\right)$$
1324	CHAPTER 36: Indicator PWM 3.5.3 Return from PSS Mode	Revised the below: Error) An interrupt of indicator PWM can be applied to a trigger of returning from PSS mode. Then the bus clock CLK_SYSC0P should be supplied during the PSS mode. Correct) The interrupt of the indicator PWM module cannot be used as a return factor from PSS mode.
1379	CHAPTER 38: Sound System Configuration 3.5. Sound system reset Operation	Revised the below: Error) The reset negating status should be confirmed when the master accesses to the sound system after negating the sound system reset. Correct) After releasing the sound system reset, confirm that the reset negation is completed, then the master can access to the sound system.

Page	Section	Change Results
1379	CHAPTER 38: Sound System Configuration 3.5.1. Disabling the Output Module	Revised the below: Error) Therefore, the output from audio-DAC, PCM-PWM and I2S are also reset asynchronously. However, regarding PCMPWM, the output level on disable status is configurable by the register in PCMPWM. To prevent from changing PCMPWM output by the sound system reset, output level selection register in PCMPWM should be initialized. Correct) Therefore, audio-DAC, PCM-PWM and I2S are stopped immediately. Notes: − Regarding PCMPWM, the output level on disable status is configurable by the register in PCMPWM. To prevent from changing the PCMPWM output by the sound system reset, output level selection register in PCMPWM should be initialized.
1379	CHAPTER 38: Sound System Configuration 3.5.2. Disabling the DMA controller	Revised the below: Error) The sound system reset is asserted asynchronously. Therefore, if the DMA access to the sound system conflict with the sound system reset, the DMA access cannot be guaranteed. To avoid this, DMA controller should be disabled. Correct) The sound system reset is asserted asynchronously. Therefore, if the sound system reset occurs during DMA transfer, the transfer data from DMA controller cannot be written into the sound system. However, the transfer data appear to be written for DMA, because any error responses are not returned from the sound system during the reset. To avoid this, DMA controller should be disabled before sound system reset asserts.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1398</td><td>CHAPTER 39: Expand 5.1. Expand PLLx Clock Supervisor Setting Register 0</td><td>Revised the below:
5.1 Expand PLLx Clock Supervisor Setting Register 0 (ECXSVPLLxCFGR0) (x=0 to 2)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand PLLx Clock Supervisor Setting Register 0 (EXCSVPLLxCFGR0)</td></tr>
<tr><td>OFFSET</td><td>0x00(x=0) 0x08(x=1) 0x10(x=2)</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1398</td><td>CHAPTER 39: Expand 5.1. Expand PLLx Clock Supervisor Setting Register 0</td><td>Revised the below:
5.1 Expand PLLx Clock Supervisor Setting Register 0 (ECXSVPLLxCFGR0) (x=0 to 2)
Error)
<table>
<tr><td>Bit</td><td>23 22 21 20 19 18 17 16</td></tr>
<tr><td>Field</td><td>UPTHR[7:0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td></td></tr>
<tr><td>Initial Value</td><td></td></tr>
</table>
Correct)
<table>
<tr><td>Bit</td><td>23 22 21 20 19 18 17 16</td></tr>
<tr><td>Field</td><td>UPTHR[7:0]</td></tr>
<tr><td>R/W Attribute</td><td>R/W</td></tr>
<tr><td>Protection Attribute</td><td></td></tr>
<tr><td>Initial Value</td><td>00000000</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1400</td><td>CHAPTER 39: Expand 5.2. Expand PLL0xClock Supervisor Setting Register 1</td><td>Revised the below:
5.2 Expand PLL0xClock Supervisor Setting Register 1 (ECXSVPLLxCFGR1) (x=0 to 2)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand PLL0xClock Supervisor Setting Register 1 (EXCSVPLLxCFGR1)</td></tr>
<tr><td>OFFSET</td><td>0x04(x=0) 0x0C(x=1) 0x14(x=2)</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1402</td><td>CHAPTER 39: Expand 5.3. Expand Clock Supervisor Setting Register</td><td>Revised the below:
5.3 Expand Clock Supervisor Setting Register (EXCSVCFGR)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand Clock Supervisor Setting Register (EXCSVCFGR)</td></tr>
<tr><td>OFFSET</td><td>0x18</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1404</td><td>CHAPTER 39: Expand 5.4. Expand Clock Supervisor Error Register</td><td>Revised the below:
5.4 Expand Clock Supervisor Error Register (EXCSVERR)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand Clock Supervisor Error Register (EXCSVERR)</td></tr>
<tr><td>OFFSET</td><td>0x1C</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1406</td><td>CHAPTER 39: Expand 5.5. Expand Clock Supervisor Error Clear Register</td><td>Revised the below:
5.5 Expand Clock Supervisor Error Clear Register (EXCSVERRCLR)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand Clock Supervisor Error Register (EXCSVERR)</td></tr>
<tr><td>OFFSET</td><td>0x1C</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1408</td><td>CHAPTER 39: Expand 5.6. Expand PLLx Control Register</td><td>Revised the below:
5.6 Expand PLLx Control Register (EXPLLxCNTR) (x=0 to 2)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand PLLx Control Register (EXPLLxCNTR)</td></tr>
<tr><td>OFFSET</td><td>0x40(x=0) 0x44(x=1) 0x48(x=2)</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1411</td><td>CHAPTER 39: Expand 5.7. Expand PLL Enable Register</td><td>Revised the below:
5.7 Expand PLL Enable Register (EXPLLEN)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand PLL Enable Register (EXPLLEN)</td></tr>
<tr><td>OFFSET</td><td>0x4C</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1413</td><td>CHAPTER 39: Expand 5.8. Expand Clock Divider Register x</td><td>Revised the below:
5.8. Expand Clock Divider Register x (EXCKDIVRx)(x=0 to 2)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand Clock Divider Register x (EXCKDIVRx)
(x=0 to 2)</td></tr>
<tr><td>OFFSET</td><td>0x50(x=0) 0x54(x=1) 0x58(x=2)</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1416</td><td>CHAPTER 39: Expand 5.9. Expand PLL Select Register</td><td>Revised the below:
5.9. Expand PLL Select Register (EXPLLSEL)
Error)

Correct)
<table>
<tr><td>REGISTER_NAME</td><td>Expand PLL Select Register (EXPLLSEL)</td></tr>
<tr><td>OFFSET</td><td>0x5C</td></tr>
<tr><td>ACCESS_SIZE</td><td>B H W</td></tr>
<tr><td>MULTIPLE</td><td></td></tr>
<tr><td>NUMERIC_TYPE</td><td></td></tr>
<tr><td>OTHER</td><td></td></tr>
</table></td></tr>
<tr><td>1418</td><td>CHAPTER 40: LCD Subsystem</td><td>Added CHAPTER 40: LCD Subsystem</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td colspan="3">Rev.*B</td></tr>
<tr><td>20</td><td>CHAPTER 1: Overview</td><td>2. Document definition
Error)

Correct)
Note:
- As for the resources specifications of this model, refer the Pin Descriptions chapter in the datasheet.</td></tr>
<tr><td>36</td><td>CHAPTER 3: Product Description</td><td>2. Product description Table 2-1
Error)
<table><tr><td>Embedded Program/Work Flash Memory</td><td>Work Flash can be accessed with 0-wait-cycle if CPU frequency is 12.5MHz or less.
7-wait-cycle: 80MHz or less.
11-wait-cycle: 132MHz or less.</td></tr></table>
Correct)
<table><tr><td>Embedded Program/Work Flash Memory</td><td>Work Flash can be accessed with 0-wait-cycle if TCFlash frequency is 12.5MHz or less.
6-wait-cycle: 80MHz or less.
10-wait-cycle: 132MHz or less.</td></tr></table></td></tr>
<tr><td>38</td><td>CHAPTER 3: Product Description</td><td>2. Product description Table 2-1
Error)

Correct)
<table><tr><td>Parallel flash programming</td><td>Parallel writer mode needs MD=0+RSTX to transit the mode correctly.</td></tr></table></td></tr>
<tr><td>62</td><td>CHAPTER 7: Memory Map</td><td>3. Base address map
Error)
<table><tr><td>B064_0000</td><td>B064_0FFF</td><td>MCU Config Group</td><td>PWM</td><td>364</td></tr></table>
Correct)
<table><tr><td>B064_0000</td><td>B064_0FFF</td><td>MCU Config Group</td><td>Indicator PWM</td><td>364</td></tr></table></td></tr>
</table>

Page	Section	Change Results
207	CHAPTER 11: Port Configuration	3.3 The Analog I/O setting Error) Correct) Note: - An analog I/O port for A/D converter performs as pull-up when the software configures pull-up and down for a port simultaneously.
212	CHAPTER 11: Port Configuration	3.5 Output Drive Capacity Setting Error) This section shows the I/O port output drive capacity settings. Correct) This section shows the I/O port output drive capacity settings. Note that the drive capacity depends on power supply voltage. The table only describes the representative value when 5V is used as standard value of power supply usage. Please see the actual characteristics in datasheet.
642	CHAPTER 23: Partial Wakeup	3.2. Flowchart Error) ***Notes:*** 'i' represents the logical channel number. i = 0, 1, 2, …, 63 In PWU mode, the number of activation channel which can be used is maximum 8. Correct) ***Notes:*** 'i' represents the logical channel number. i = 0, 1, 2, …, 63 In PWU mode, the number of activation channel which can be used is maximum 8 in i = 0, 1, 2, …, 31.
707, 732, 751	CHAPTER 26: DDR High Speed SPI Controller	Changed “Spansion” to “Cypress” in below chapters. 3.5. SPI Data Protocol 4.2. Command Sequencer Mode 4.4.1. Notice On Document Designations 4.4.2. Advance Information

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>912</td><td>CHAPTER 29: Sound Generator</td><td>
4. Registers Table 4-3 Register Offset address map
Error)
<table>
<tr><th rowspan="2">OFFSET_ADDRESS</th><th colspan="4">REGISTER_NAME</th><th rowspan="2">ACCESS_SIZE</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x00000000</td><td>Reserved</td><td>SGDER</td><td colspan="2">SGCR</td><td>B, H, W</td></tr>
<tr><td>0x00000004</td><td colspan="2">SGAR</td><td>SGFR</td><td>SGNR</td><td>B, H, W</td></tr>
<tr><td>0x00000008</td><td>SGTCR</td><td>SGIDR</td><td colspan="2">SGPCR</td><td>B, H, W</td></tr>
<tr><td>0x0000000C</td><td colspan="4">SGDMAR</td><td>B, H, W</td></tr>
<tr><td>0x00000010</td><td colspan="2">Reserved</td><td colspan="2">SGCCR</td><td>H, W</td></tr>
</table>
Correct)
<table>
<tr><th rowspan="2">OFFSET_ADDRESS</th><th colspan="4">REGISTER_NAME</th><th rowspan="2">ACCESS_SIZE</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x00000000</td><td>Reserved</td><td>SGDER</td><td colspan="2">SGCR</td><td>B, H, W</td></tr>
<tr><td>0x00000004</td><td colspan="2">SGAR</td><td>SGFR</td><td>SGNR</td><td>B, H, W</td></tr>
<tr><td>0x00000008</td><td>SGTCR</td><td>SGIDR</td><td colspan="2">SGPCR</td><td>B, H, W</td></tr>
<tr><td>0x0000000C</td><td colspan="3">Reserved *1</td><td>SGEFR</td><td>B, H, W</td></tr>
<tr><td>0x00000010</td><td colspan="4">SGDMAR</td><td>B, H, W</td></tr>
<tr><td>0x00000014</td><td colspan="2">Reserved</td><td colspan="2">SGCCR</td><td>H, W</td></tr>
</table>
*1: 0x000000 is read out from the reserved bit field of 0x0000000C.
</td></tr>
<tr><td>1024</td><td>CHAPTER 30: Sound Generator</td><td>
4.25. SWFG Channel n Control 4 Register (WGCHnCTRL4, n=0 to 4)
Error)
[bit31] CHNSTART: Channel n start mirror bit

Correct)
[bit31] CHnSTART: Channel n start mirror bit
</td></tr>
<tr><td>1037</td><td>CHAPTER 31: Sound Mixer</td><td>
3.1.1 Mixing Start
Error)

Correct)
Notes:
−In the step 5, completion of the initialization is required before executing the next step. Completion can be confirmed by reading the applicable bit of the MXBUFFCLR register, because the bit is automatically cleared to 0 following initialization.
</td></tr>
</table>

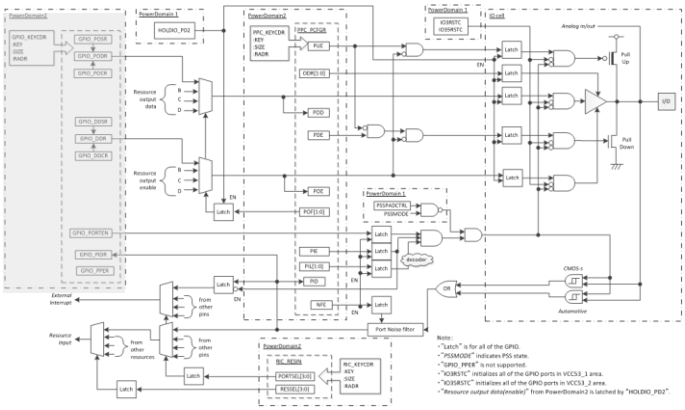
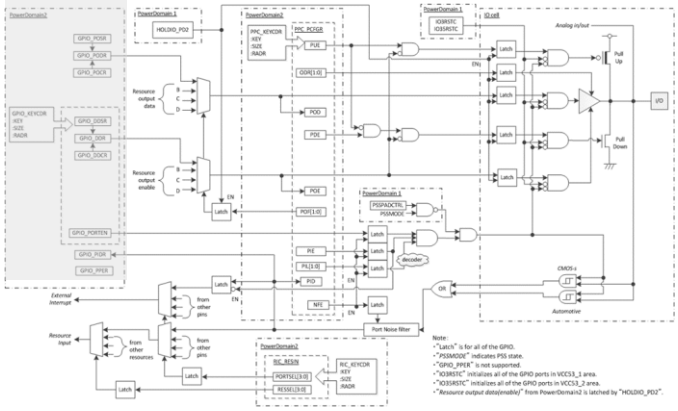
Page	Section	Change Results
1046	CHAPTER 31: Sound Mixer	3.10.2. Sampling Rate Conversion and FIR Filter Error) Notes: - Filter characteristics have been selected based on Spansion rules. Correct) Notes: - Filter characteristics have been selected based on Cypress rules.
1328	CHAPTER 36: Indicator PWM	4. Registers 4.1. Timer Control Register (ITMCR) Error) [bit23:16] Reserved. This is a reserved bit. Writing data to these bits has no effect on operation. [bit15] CKSEL: Source Clock selection bit This bit selects either Main clock or Sub clock as source clock. Stop the timer (ICNTCR:DOP[bit]=0) before changing the source clock Correct) [bit23:17] Reserved. This is a reserved bit. Writing data to these bits has no effect on operation. [bit16] CKSEL: Source Clock selection bit This bit selects either Main clock or Sub clock as source clock. Stop the timer (ICNTCR:DOP[bit]=0) before changing the source clock [bit15] Reserved. This is a reserved bit. Writing data to these bits has no effect on operation.
1375	CHAPTER 38: Sound System Configuration	3.3.1. Disabling the Output Module Error) Correct) Notes: − When the PCM-PWM or I2S is used as output module and the output module is changed from "enabled" to "disabled" (*1), the sound mixer should not be disabled and the output module should not be enabled during 16 sampling cycles. If this sequence is not followed, PCM data requested before disabling the output module may be transferred to the output module after enabling the output module. This may cause the DMA block error, Tx Block Size Error or FIFO overflow error. − (*1) In PCM-PWM case, this means that the global enable bit (PCMPWMi_CONTROL,EN) is changed from 1 to 0. − (*1) In I2S case, this means that the transmission enable bit (I2Sn_OPRREG:TXENB) or the I2S enable bit (I2Sn_OPRREG.START) is changed from 1 to 0.

Page	Section	Change Results
Rev.*C		
111	CHAPTER 11: Port Configuration	2. Configuration and Block Diagram Error) This chapter doesn't have a block diagram. Correct) Added Figure 2-1 and added below description. There are five latches in I/O cell area. The power supply for them is I/O power VCC5, VCC53_1, or VCC53_2. The state latches are only expected under these power supply turning-on. The power for other latches is internal regulated 1.2V.
938	CHAPTER 30: Sound Waveform Generator	3.1.7 Initial gain Error) Correct) Notes: − Even if the minimum gain is applied to the sound source, the SWFG does not always output ALL 0 data. In that case, positive data become ALL 0, but negative data become ALL 1 (-1 in decimal). Due to this, slight sound may be heard in accordance with your output environment.
961	CHAPTER 30: Sound Waveform Generator	3.8 Filter Error) Cut off frequency 400 Hz : refer to 3.8.1 Cut off frequency 800 Hz : refer to 3.8.2 Cut off frequency 1.2 kHz : refer to 3.8.3 Cut off frequency 2 kHz : refer to 3.8.4 Cut off frequency 5 kHz : refer to 3.8.5 Correct)
962	CHAPTER 30: Sound Waveform Generator	3.8.1. Example.1 Error) Cut off frequency is 400 Hz. Correct) Figure 3-31 shows frequency response when the filter coefficients are set as Table 3-7. Revised Figure 3-31

Page	Section	Change Results
963	CHAPTER 30: Sound Waveform Generator	3.8.2. Example.2 Error) Cut off frequency is 800 Hz. Correct) Figure 3-32 shows frequency response when the filter coefficients are set as Table 3-8. Revised Figure 3-32
964	CHAPTER 30: Sound Waveform Generator	3.8.3. Example.3 Error) Cut off frequency is 1.2 kHz. Correct) Figure 3-33 shows frequency response when the filter coefficients are set as Table 3-9. Revised Figure 3-33
965	CHAPTER 30: Sound Waveform Generator	3.8.4. Example.4 Error) Cut off frequency is 2 kHz. Correct) Figure 3-34 shows frequency response when the filter coefficients are set as Table 3-10. Revised Figure 3-34
966	CHAPTER 30: Sound Waveform Generator	3.8.5. Example.5 Error) Cut off frequency is 5 kHz. Correct) Figure 3-35 shows frequency response when the filter coefficients are set as Table 3-11. Revised Figure 3-35

Page	Section	Change Results
369	CHAPTER 30: Sound Waveform Generator	3.10 Smooth connection Error) Because of smooth connection, time period of sound source generation may become long in certain cases. If sound source generations are continuously generated, extended time period is deleted from next sound source generation. This operation is illustrated in Figure 3 37. Correct) When the smooth connection is enabled, phase of the next sound source is started from 0 degree regardless of end phase of the previous sound source. This may cause a noise by discontinuity of sound between the sound sources. ….. If parameter of frequency between the sound sources is changed, a noise by discontinuity of sound cannot be decreased by this function. Because, the discontinuity of sound cannot be improved even if the sound sources are connected at 0 degree of phase of the sound source. Because of smooth connection, time period of sound source generation may become long in certain cases. If sound source generations are continuously generated, extended time period is deleted from next sound source generation. In this case, the next sound source becomes 2 PCM data (2/48000 sec) longer than calculated sound source time. This operation is illustrated in Figure 3 37.
971	CHAPTER 30: Sound Waveform Generator	3.10 Smooth connection Error) Correct) Notes: − Sound source time of the second sound source must be longer than one period of frequency of the first sound source. "One period of 1st sound source defined by FSREQ" < "2nd sound source time (FSLEN)"
1035	CHAPTER 31: Sound Mixer	3.2 Volume Effects Error) Correct) Notes: − Even if the minimum gain or mute is applied to the sound source, the mixer does not always output ALL 0 data. In that case, positive data become ALL 0, but negative data become ALL 1 (-1 in decimal). Due to this, slight sound may be heard in accordance with your output environment.
1217 to 1247	CHAPTER 34: LCD Bus Interface	Chapter numbers were revised after chapters as below. Error) 3. Data Master Interface Correct) 2.10 Data Master Interface

Page	Section	Change Results
1231 to 1247	CHAPTER 34: LCD Bus Interface	Register descriptions were revised. 3.9.2 Register Overview to 3.9.45 ColorLookup Table[0..255].

Page	Section	Change Results
Rev.*D		
31	CHAPTER 2: Function List	2. Operation function 2.2. ID Error) Revision A Chip ID 0x10160100, 0x10168100 Correct) Revised Chip ID Revision B Chip ID 0x10160101, 0x10168101
111	CHAPTER 11: Port Configuration	2. Configuration and Block Diagram Error) Figure 2-1  Correct) Figure 2-1 
224	CHAPTER 11: Port Configuration	5. Configuration procedure Error) Correct) Added below chapters. 5. Configuration procedure.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>232</td><td>CHAPTER 11: Port Configuration</td><td>6. Note
Error)

Correct) Added below descriptions
6.2 I2C port configuration
Port configuration of I2C should be done with the following order.
(1) Set the MD (Operation Mode Setting Bits) of the SMR (Serial Mode Register) to I2C mode.
(2) Set the POF (Port Output Function Selection Bit) of the PPC_PCFGRijj (Port Setting Register) to I2C.
Otherwise, I2C fast-mode output pins "P1_02(MFS6_SDA), P1_03(MFS6_SCL), P1_05(MFS7_SDA), P1_06(MFS7_SCL)" would output unintentional "high level" from the ports during the error procedure (2) -> (1) due to I/O circuit spec.</td></tr>
<tr><td>255</td><td>CHAPTER 14: Low Voltage Detection</td><td>4. Registers
Error)
[Bit11:9] LVDH1V External low-voltage detection voltage setting bits
<table><tr><th>Bit 11:9</th><th>Voltage [V]</th></tr><tr><td>000 *</td><td>2.50</td></tr></table><table><tr><td>110 *</td><td>2.70</td></tr><tr><td>111 *</td><td>2.60 (Initial value)</td></tr></table>
Correct)
[Bit11:9] LVDH1V External low-voltage detection voltage setting bits
<table><tr><th>Bit 11:9</th><th>Voltage [V]</th></tr><tr><td>000 *</td><td>2.70</td></tr></table><table><tr><td>110 *</td><td>2.50</td></tr><tr><td>111 *</td><td>2.60 (Initial value)</td></tr></table>
The table shows the representative detected values. See the electric characteristics of LVD on datasheet in detail.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td colspan="3">Rev.*E</td></tr>
<tr><td>42</td><td>CHAPTER 3: Product Description</td><td>3. Note 3.4. Restriction
Added a restriction.
Error)

Correct)
<table>
<tr><td>DDRHSSPI</td><td>DDRHSSPIn_PCC0-3.SS2CD[1:0]
Quad Page Program</td><td>DDRHSSPIn_PCC0-3.SS2CD[1:0]
must be set 2'b11. …….</td><td>Please refer to the following figure 2-1…….</td></tr>
</table>
Fig.2-1: Quad Page Program by Quad Mode</td></tr>
<tr><td>78
to
83</td><td>CHAPTER 9: DMA Channel Activation Factors</td><td>1. Factors list
Revised Factor list table and added remarks.
Error)
<table>
<tr><th rowspan="2">Number of channels</th><th colspan="6">Peripheral functions combination</th><th rowspan="2"></th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>0 to 8</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>9</td><td>WORK FLASH</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>42 to 49</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>50</td><td>CAN_FD ch.0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>51</td><td>CAN_FD ch.1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>52</td><td>CAN_FD ch.2</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>53</td><td>CAN_FD ch.3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>54</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>55</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>56</td><td>Base Timer ch.0-0</td><td>Base Timer ch.0-1</td><td>Base Timer ch.24-0</td><td>Base Timer ch.24-1</td><td>-</td><td>-</td><td>*1</td></tr>
<tr><td>125</td><td>I2S ch.1 TX</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td>126 to 128</td><td colspan="6">Reserved</td><td></td></tr>
</table>
Correct)
<table>
<tr><th rowspan="2">Client Number</th><th colspan="6">Peripheral functions combination</th><th rowspan="2">Enable Setting of IRQ Request</th><th rowspan="2">Remarks</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>0 to 8</td><td colspan="6">Reserved</td><td>-</td><td></td></tr>
<tr><td>9</td><td>WORK FLASH</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Unnecessary</td><td>*2</td></tr>
<tr><td>42 to 55</td><td colspan="6">Reserved</td><td>-</td><td></td></tr>
<tr><td>56</td><td>Base Timer ch.0-0</td><td>Base Timer ch.0-1</td><td>Base Timer ch.24-0</td><td>Base Timer ch.24-1</td><td>-</td><td>-</td><td>Necessary</td><td>*1, *3</td></tr>
<tr><td>125</td><td>I2S ch.1 TX</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Unnecessary</td><td></td></tr>
<tr><td>126 to 128</td><td colspan="6">Reserved</td><td>-</td><td></td></tr>
</table>
*2: DSTP_ACK function is set by DMAi_CMCICm:BEHSTPACK.
*3: The interrupt factor flags of peripheral function are cleared automatically by acceptance of DMA transfer request
*4: The interrupt factor flags of peripheral function are cleared automatically after DMA transfer is started.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>100</td><td>CHAPTER 10: Port Description</td><td>1. Port description list

Error)
<table><tr><td>DDRHSSPI0_SSEL</td><td>DDR HS-SPI0 select output pin</td><td>-</td><td>25</td><td>31</td><td>39</td><td></td></tr><tr><td>DDRHSSPI1_SSEL</td><td>DDR HS-SPI1 select output pin</td><td>-</td><td>16</td><td>22</td><td>30</td><td></td></tr></table>
Correct)
<table><tr><td>DDRHSSPI0_SSEL</td><td>DDR HS-SPI select 0 output pin</td><td>-</td><td>25</td><td>31</td><td>39</td><td></td></tr><tr><td>DDRHSSPI1_SSEL</td><td>DDR HS-SPI select 1 output pin</td><td>-</td><td>16</td><td>22</td><td>30</td><td></td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td colspan="3">Rev.*F</td></tr>
<tr><td>51</td><td>CHAPTER 5: Clock Configuration</td><td>3. Remark

Added below note.
− The frequency of Sound System clocks should satisfy the following conditions.
Expand_CLK_CD1 : Expand_CLK_CD1A = 1:1
Expand_CLK_CD1 : Expand_CLK_CD1B = 1:1 or 2:1 or 3:1
Expand_CLK_CD1A >= 100MHz</td></tr>
<tr><td>96</td><td>CHAPTER 10: Port Description</td><td>1. Port description list

Error)
<table><tr><td>BT0_TIOA1_1</td><td>Base timer ch.0 TIOA output pin (1)</td><td>-</td><td>-</td><td>2</td><td>2</td><td></td></tr><tr><td>BT1_TIOA2_1</td><td>Base timer ch.1 TIOA output pin (1)</td><td>-</td><td>-</td><td>3</td><td>3</td><td></td></tr><tr><td>BT1_TIOA3_1</td><td>Base timer ch.1 TIOA output pin (1)</td><td>-</td><td>-</td><td>4</td><td>4</td><td></td></tr></table>
Correct)
<table><tr><td>BT0_TIOA1_1</td><td>Base timer ch.1 TIOA output pin (1)</td><td>-</td><td>-</td><td>2</td><td>2</td><td></td></tr><tr><td>BT1_TIOA2_1</td><td>Base timer ch.2 TIOA output pin (1)</td><td>-</td><td>-</td><td>3</td><td>3</td><td></td></tr><tr><td>BT1_TIOA3_1</td><td>Base timer ch.3 TIOA output pin (1)</td><td>-</td><td>-</td><td>4</td><td>4</td><td></td></tr></table></td></tr>
<tr><td>244</td><td>CHAPTER 12: State Transition</td><td>4. Changes to PSS and RUN
Revised “Internal Reset” signal in From PSS mode to RUN change state (Return from a power supply shutoff state) of Figure4-3 Changes to RUN Timing Chart.</td></tr>
<tr><td>405</td><td>CHAPTER 18: Base Timer Simultaneous Operation</td><td>3.1 Global Timer and Base Timers Operation Waveform
Figure 3-1 Global Timer waveform (It sets a STOP bit as “0” from “1”, and starts Global Timer.)
Revised values of GTMSCHB and GTMSCH in Figure 3-1.</td></tr>
<tr><td>409</td><td>CHAPTER 18: Base Timer Simultaneous Operation</td><td>3.2 Global Timer Match Starting Channel Buffer Operation by MODE set
Figure 3-3 The waveform in the case of adding Base Timers CH of the Global Timer Match Start enable. (MODE=0)
Revised values of GTMSCHB and GTMSCH in Figure 3-3.</td></tr>
<tr><td>410</td><td>CHAPTER 18: Base Timer Simultaneous Operation</td><td>3.2 Global Timer Match Starting Channel Buffer Operation by MODE set

Error)
① It is a start set from Base Timers CH0 to CH3.
Correct)
① It is a start set from Base Timers CH0 to CH2.</td></tr>
</table>

Page	Section	Change Results
411	CHAPTER 18: Base Timer Simultaneous Operation	3.2 Global Timer Match Starting Channel Buffer Operation by MODE set Figure 3-4 The waveform in the case of adding Base Timers CH of the Global Timer Match Start enable. (MODE=1) Revised values of GTMSCHB and GTMSCH in Figure 3-4.
411	CHAPTER 18: Base Timer Simultaneous Operation	3.2 Global Timer Match Starting Channel Buffer Operation by MODE set Error) ② It is a start set from Base Timers CH0 to CH3. Correct) ② It is a start set from Base Timers CH0 to CH2.
756	CHAPTER 26: DDR High Speed SPI Controller	4.4. General use case guidelines for DDRHSSPI Using the DDRHSSPI in Command Sequencer Mode of Operation Error) 2. DDRMODE bit shall be set same as DDRHSSPIn_DMFIFOCFG.DDRM bit. Correct) 2. DDRMODE bit shall be set same as DDRHSSPIn_DMTRP.DDRM bit.

Revision History

Document Revision History

Document Title:	32-bit Microcontroller S6J3360/S6J3370 Series Hardware Manual Traveo™ Family
Document Number:	002-18302

Revision	ECN No.	Description of Change
**	5583651	New Specification
*A	5663623	Revised CHAPTER 29: SOUND Waveform Generator and CHAPTER 30: SOUND Mixer. Added CHAPTER 40: LCD SUBSYSTEM. For detail, see “Major Changes”.
*B	5787514	Revised some chapters. Fore detail, see “Major Changes”.
*C	5909593	Revised CHAPTER11: Port Configuration, CHAPTER30 SOUND Waveform Generator, CHAPTER31 SOUND Mixer, and CHAPTER34 LCD Bus Interface. For detail, see “Major Changes”.
*D	5983070	Revised some chapters. For detail, see “Major Changes”.
*E	6224950	Revised CHAPTER 3: Product Description, CHAPTER 9: DMA ch Activation factors, and CHAPTER 10: Port Description. For detail, see “Major Changes”.
*F	6692693	Revised some chapters. For detail, see “Major Changes”.